

Reduced Basis Methods for the Solution of Parametrized PDEs in Repetitive and Complex Networks with Application to CFD

THÈSE N° 5529 (2012)

PRÉSENTÉE LE 18 OCTOBRE 2012
À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE DE MODÉLISATION ET CALCUL SCIENTIFIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Laura IAPICHINO

acceptée sur proposition du jury:

Prof. K. Hess Bellwald, présidente du jury
Prof. A. Quarteroni, Dr G. Rozza, directeurs de thèse
Prof. P. Gervasio, rapporteur
Prof. F. Nobile, rapporteur
Prof. K. Veroy Grepl, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

A mia sorella Giuliana

Acknowledgements

I would like to start my acknowledgment by thanking Prof. Alfio Quarteroni for the unique opportunity he gave me: to work with him and to be part of his excellent group of research has been for me the richest step of my education and the biggest occasion to learn what research means.

I am very thankful to Dr. Gianluigi Rozza that allowed me to carry on the very interesting subject of this thesis with enthusiasm, courage and passion in each day of these years.

I thank Prof. Kathrine Hess, Prof. Paola Gervasio, Prof. Fabio Nobile and Prof. Karen Veroy for accepting to be part of the jury of my doctoral examination and for their helpful review of this thesis.

For having provided some contributions to this work: I thank Matteo Lesinigo for his precious discussions, ideas and for being a very good friend and Christoph Jaeggli who carried out his semester project under my supervision.

I am grateful to Dr. Simone Deparis, Dr. Marco Discacciati and Dr. Luca Dedé for their kindly support within this experience.

A special acknowledgment to Ricardo for daily bringing in our office his smile and his positive and cheerful temperament.

Many thanks also to the other people of the CMCS group that represented a solid support for me: Cristiano, Matteo, Matteo, Claudia, Gwenol, Simone, Paolo, Samuel, Peng, Toni, Aymen, Federico and Radu. I can not forget to thank other special people that shared with me memorable moments in Lausanne: Elisa, Maike, Eleonora, Francesca, Paolo, Tommaso, Davide, Wibke, Andrea, Alessandro, Laura e Francesco.

Desidero ringraziare particolarmente Alessandra perché, nonostante gli anni e le distanze sempre maggiori, la sua sincera amicizia resta sempre una certezza.

Ringrazio i miei amici di sempre Gabriella, Federico e Simona perché, anche dopo lunghe assenze, ci si ritrova sempre più amici di prima. Grazie anche a due importanti amiche, Leila e Annalisa, incontrate durante il mio percorso.

Un ringraziamento speciale a Francisco per la sua disponibilità e i preziosi consigli che riserva per me. Ringrazio tantissimo Pietro, per l'importante dono di riempire ogni giorno della mia vita di così tanta gioia.

Infine, un grazie a Giuliana, per essere non solo una sorella, ma un solido punto di riferimento, esempio di coraggio e di forza, alla quale dedico il raggiungimento di questo importante traguardo.

Ringrazio la mia mamma per la sua infinita dolcezza e per avermi insegnato a superare ogni possibile ostacolo sempre con un sorriso.

Infinite grazie al mio papà, per essere stato il mio primo maestro capace di trasmettermi con passione il fascino della matematica e per rappresentare il mio esempio costante di vita.

Lausanne, 1st August 2012

Laura Iapichino

Abstract

The objective of this work is to develop a numerical framework to perform rapid and reliable simulations for solving parametric problems in domains represented by networks and to extend the classical reduced basis method. Aimed at this scope, we propose two original methodological approaches for the approximation of partial differential equations in domains made up by repetitive parametrized geometries where topological features are recurrent: the reduced basis hybrid method (RBHM) and the reduced basis-domain decomposition-finite element (RDF) method.

The common paradigm of these methods is the consideration that the blocks composing the computational domain are topologically similar to a few reference shapes. On the latter, we compute representative solutions, corresponding to the same governing partial differential equations, but for different values of some parameters of interest and representing, for example, the deformation of the blocks. A new desired solution for a new deformed domain is recovered as projection on the reduced spaces built by the previously precomputed solutions and the continuity of the solution across subdomain interfaces is guaranteed by suitable coupling conditions. The different choices for the reduced spaces and coupling conditions adopted characterize one method with respect to the other one.

The geometrical parametrization of the considered domains, by transfinite maps, induces non-affine parameter dependence: an empirical interpolation technique is used to recover an approximate affine parameter dependence and a sub-sequent offline/online decomposition of the reduced basis procedure. This computational decomposition yields a considerable reduction of the problem complexity. Results computed on some combinations of 2D and 3D geometries, representing cardiovascular networks, show the flexibility and the advantages of the proposed methods in terms of reduced computational costs and complexities. The computational time with these new approaches is, in general, much reduced with respect to a classical finite element method on the whole domain but also only marginally slower than a classical reduced basis approach on the whole domain. However, these approaches decrease drastically the offline time to pre-compute the reduced basis by splitting the total number of parameters characterizing the problem into smaller subsets for each reference block, moreover they allow to considerably increase the geometrical flexibility and versatility.

Keywords: reduced basis methods, empirical interpolation method, transfinite maps, reduced order modelling, geometrical parametrization, online/offline computational decoupling, domain decomposition, Stokes equations.

Version abrégée

L'objectif de ce travail est de résoudre des problèmes paramétriques dans des domaines représentés par des réseaux par des simulations rapides et fiables et d'étendre la méthode classique des bases réduites. Nous proposons deux approches méthodologiques originales pour l'approximation des équations aux dérivées partielles dans des domaines constitués par des géométries paramétrées répétitives, où les caractéristiques topologiques sont récurrentes : la méthode RBHM et la méthode RDE.

Le paradigme commun de ces méthodes repose sur le fait que les blocs qui composent le domaine de calcul sont topologiquement semblable à des formes de référence. Sur ces dernières, nous calculons des solutions représentatives, ce qui correspond au même système d'équations aux dérivées différentielles partielles, mais pour différentes valeurs des paramètres d'intérêt représentant, par exemple, la déformation des blocs. Chaque nouvelle solution souhaitée pour chaque nouveau domaine déformé est trouvée grâce à la projection sur les espaces réduits construits par les solutions précédemment précalculées et la continuité de la solution au travers des interfaces internes est garantie par des conditions de couplage appropriées. Les différents choix pour les espaces réduits et pour les conditions de couplage caractérisent une méthode par rapport à l'autre.

La paramétrisation géométrique des domaines considérés, obtenue par les *transfinite maps*, induit une dépendance non-affine des paramètres : une technique d'interpolation empirique est utilisée pour récupérer une dépendance affine des paramètres et une décomposition online/offline typique de la méthode des bases réduites. Cette décomposition computationnelle implique une réduction considérable de la complexité du problème. Des résultats calculés sur certaines combinaisons de géométries 2D et 3D, représentant des réseaux cardio-vasculaires, montrent la souplesse et les avantages de ces méthodes en termes de réduction des coûts informatiques et de complexité. Le temps de calcul avec ces nouvelles approches est, en général, beaucoup plus faible par rapport à une méthode classique d'éléments finis appliquée au domaine tout entier, et légèrement plus grand qu'avec une approche classique des bases réduites appliquée sur tout le domaine. Cependant, ces approches permettent de diminuer considérablement le temps offline pour pré-calculer la base réduite en divisant le nombre total de paramètres caractérisant le problème en petits sous-ensembles pour chaque bloc de référence, en outre, ils permettent d'augmenter considérablement la flexibilité géométrique du problème considéré.

Mots clés : méthodes des bases réduites, méthode d'interpolation empirique, applications transfinies, modélisation d'ordre réduit, paramétrisation géométrique, décomposition computationnelle online/offline, décomposition de domaine, équations de Stokes.

Estratto

L'obiettivo di questa tesi è la risoluzione di problemi parametrici in domini rappresentati da network tramite rapide ed accurate simulazioni numeriche e di estendere il metodo alle basi ridotte classico. Per questo scopo, proponiamo due approcci metodologici originali per la risoluzione di equazioni differenziali alle derivate parziali in domini composti da geometrie ripetitive e parametrizzate in cui le caratteristiche topologiche sono ricorrenti: il metodo RBHM e il metodo RDE.

Il paradigma comune di questi metodi è la considerazione che i blocchi che compongono il dominio computazionale sono topologicamente simili ad una qualche forma di riferimento. Su quest'ultima, vengono calcolate soluzioni rappresentative, corrispondenti allo stesso sistema di equazioni differenziali parziali, ma per diversi valori dei parametri di interesse, che rappresentano, per esempio, la deformazione geometrica del blocco. Ogni nuova soluzione desiderata per ogni nuovo dominio deformato viene trovata come proiezione sugli spazi ridotti costituiti dalle soluzioni precedentemente calcolate, mentre la continuità della soluzione lungo le interfacce interne è garantita da particolari condizioni di accoppiamento. La scelta degli spazi ridotti e delle condizioni di accoppiamento caratterizza un metodo rispetto all'altro.

La parametrizzazione geometrica dei domini considerati è ottenuta da mappe transfinite ed induce una dipendenza non affine dal parametro: opportune tecniche di interpolazione empirica vengono utilizzate per recuperare una dipendenza affine dal parametro e successivamente la decomposizione computazionale tra una fase offline e una online, tipica delle basi ridotte. Tale decomposizione computazionale produce una notevole riduzione della complessità computazionale del problema.

I risultati numerici ottenuti su alcune combinazioni di geometrie 2D e 3D, rappresentanti network cardiovascolari, dimostrano la flessibilità ed i vantaggi dei metodi proposti in termini di riduzione dei costi e delle complessità computazionale. Il tempo di calcolo con questi nuovi approcci è, in generale, molto ridotto rispetto ad un classico metodo a elementi finiti applicato al dominio globale e solo marginalmente più lento di un approccio classico alle basi ridotte applicato anch'esso al dominio globale. Tuttavia, questi approcci diminuiscono drasticamente il tempo di calcolo offline delle basi dividendo il numero totale di parametri che caratterizzano il problema in sottoinsiemi più piccoli per ciascun blocco di riferimento, permettendo, inoltre, di migliorare considerevolmente la flessibilità e la versatilità geometrica del problema trattato.

Parole chiave: metodi alle basi ridotte, metodo di interpolazione empirica, mappe transfinite, modellizzazione di ordine ridotto, parametrizzazione geometrica, online/offline decomposizione computazionale, decomposizione del dominio, equazioni di Stokes.

Contents

Acknowledgements	v
Abstract (English/Français/Italian)	vii
Introduction	1
1 The reduced basis method for a single domain setting	5
1.1 Introduction	5
1.2 The reduced basis method for parametrized PDEs	5
1.2.1 Overview	6
1.2.2 Problem description	7
1.2.3 The parametrized problem in the reference domain and its affine decomposition	8
1.2.3.1 The empirical interpolation method	9
1.2.4 Finite element approximation of the problem	11
1.2.5 The reduced basis formulation	11
1.2.5.1 The greedy algorithm	12
1.2.5.2 A posteriori error estimation	13
1.3 Transfinite maps	15
1.3.1 Generalized transfinite map	15
1.3.1.1 Offline stage	15
1.3.1.2 Online stage	17
1.3.1.3 Numerical examples	17
1.3.2 The boundary displacement dependent transfinite map	19
1.3.2.1 Offline/Online decoupling	19
1.3.2.2 An example	20
1.3.3 Transfinite maps for 3D geometries	20
1.3.3.1 Offline stage	20
1.3.3.2 Online stage	21
1.3.3.3 Numerical examples	22
1.4 Other available geometrical transformations	24
1.5 Concluding remarks	24
2 The reduced basis hybrid method for the parametrized steady Stokes equations	25
2.1 The steady Stokes problem	26
2.2 The parametrized steady Stokes problem in a local subdomain	27
2.3 The reduced basis formulation for the parametrized steady Stokes problem	31
2.4 The RBHM for parametrized steady Stokes equations	33
2.4.1 Two domains with single internal interface	33

2.4.2	Several subdomains with many internal interfaces	35
2.4.3	Computational strategy	38
2.5	Numerical tests on 2D domains	40
2.5.1	A first example: stenosed arterial vessel	40
2.5.2	A portion of a network with heterogeneous block domains	45
2.6	Computational costs	46
2.7	Comparison between RBHM and RB	47
2.8	Concluding remarks	49
3	Reduced basis, Domain Decomposition and Finite element methods in a combined perspective	51
3.1	Problem Setting	52
3.1.1	Geometrical parametrization of the problem	53
3.2	Finite element approximation of the problem	55
3.3	The non-overlapping domain decomposition method for FE	55
3.3.1	The Steklov-Poincaré operator for the FE problem	59
3.4	The Reduced basis-Domain decomposition-Finite element method (RDF)	61
3.4.1	FE region defined by the internal interfaces	61
3.4.2	Extending the FE regions	66
3.4.3	Precomputing the basis solutions	72
3.5	Offline/Online decomposition and computational complexity	74
3.5.1	FE region represented only by the internal interfaces	74
3.5.2	Extending the FE region	76
3.6	Numerical results	78
3.6.1	The local BCs and the local greedy algorithms	78
3.6.1.1	The Lagrangian piecewise interface functions	78
3.6.1.2	The Fourier interface functions	83
3.6.2	Global solution and accuracy results	86
3.6.2.1	Varying the set of local BCs	87
3.6.2.2	Extending the finite element region	88
3.6.2.3	Increasing the number of blocks	90
3.7	Concluding remarks	93
4	The RDF method for the solution of steady parametrized Stokes equations	95
4.1	Problem Setting	95
4.1.1	Geometrical affine parametrization	96
4.2	Finite element approximation of the problem	98
4.3	The non-overlapping Domain Decomposition Method	99
4.4	The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations	102
4.4.1	Extending the FE regions	105
4.4.2	Precomputing the basis solutions	109
4.5	Offline/Online decomposition and computational complexity	112
4.5.1	FE region represented only by the internal interfaces	112
4.5.2	Extending the FE region	113
4.6	Numerical results	115
4.6.1	The local BCs and the local greedy algorithms	115

4.6.1.1	The Dirichlet Fourier interface functions	115
4.6.1.2	The Neumann Fourier interface functions	118
4.6.2	Global solution and accuracy results	120
4.6.3	Extending the FE region	129
4.6.4	Computational times	132
4.7	Concluding remarks	133
5	Some applications to more complex parametrized geometrical configurations	135
5.1	Transfinite mapping on 3D parametrized configurations	136
5.1.1	A bifurcating pipe	136
5.1.2	Curved pipe geometry	140
5.2	An application of the RBHM on a 3D domain	144
5.3	RDF method applied to complex 2D networks	147
5.3.1	A bifurcated geometry	148
5.3.1.1	The NBC function set	148
5.3.1.2	The DBC function set	151
5.3.2	Stenosed geometry	152
5.3.2.1	The NBC function set	152
5.3.2.2	The DBC function set	154
5.3.3	A curved channel	156
5.3.4	Bifurcated tree	159
5.3.5	Bypass geometry	161
5.4	Comparison between the proposed methods	164
5.4.1	Some considerations	166
5.5	Concluding remarks	168
	Conclusions	171
	Curriculum Vitae	183

Introduction

Partial differential equations (PDEs) are commonly used to describe a wide variety of physical phenomena, e.g. sound propagation, heat conduction, fluid dynamic, elastic deformation, etc. A special class is that of parametrized partial differential equations (μ PDEs) in which physical properties of the system can be addressed to some parameters included in a vector μ . By varying the values of μ , a set of μ PDEs is able to model the same physical system of interest for several configurations, like different material properties, geometrical deformations or boundary conditions. For instance, an example of μ PDE can be represented by a model of heat conduction, where the parameter μ defines the conductivity of the domain material. Many engineering applications require a *rapid* and *reliable* solution of μ PDEs for *several* parameter values. For this reason a considerable part of the numerical analysis involves the research of methodologies able to solve μ PDEs in the *many-query* and *real-time* context, without compromising the accuracy of the solution [77, 71, 85]. Classical numerical techniques (like finite element (FE) method) guarantee a very high accuracy of the solution of μ PDEs, however, for problems with a large number of degrees of freedom, their use may require very long computational time that becomes prohibitive if we are interested in finding the solution corresponding to many different parameter values. These full order methods are able to recover a high fidelity solution through an accurate FE spatial discretization of the μ PDEs. Usually the more accurate is the discretization used with the FE method, the more accurate are the solutions recovered by the approximate solution, but unfortunately affected by heavier computational times and costs.

The rise of reduced order modelling (ROM) techniques is aimed at avoiding long times in numerical simulations by reducing the computational complexity of the problem still preserving its accuracy and stability [63].

The reduced basis (RB) method is a very effective ROM technique to approximate the solution of μ PDEs. It is particularly useful for solving problems in which the solution varies smoothly with the parameters. Due to the fact that we know the range \mathcal{D} of possible parameter values and if small variations of the parameter values induce small variations of the problem solution, the RB method consists in approximating the solution corresponding to every parameter values in the predefined range as a suitable combination of precomputed selected expensive FE solutions (snapshots), associated to adaptively chosen parameter values¹. Even if this technique requires an exploration of parameter values, and the expensive evaluation of the associated FE solution, these computations can be done only once during a unique expensive step (the *offline stage*). Then, thanks to an inexpensive *online stage*, the data set built offline can be exploited to find a solution for any values $\mu \in \mathcal{D}$ with drastically reduced computational time.

The parameter μ can describe many features of the μ PDEs, however in this thesis we focus our attention on problems in which μ represents the geometrical configuration of the computational domain, as we are interested in solving problems in computational domains that can assume several deformed

¹ More details regarding the RB method are introduced in Chapter 1.

configurations.

The first original contribution of this work is the analysis of the transfinite maps for dealing with suitable deformations of the computational domain. The use of these maps in the reduced basis context was introduced in the context of the reduced basis element method [58, 52], a combination between RB and domain decomposition. Here we present some improvements regarding critical aspects of these maps and we introduce possible 3D extensions. Transfinite maps are used to parametrize the computational domain in which we are interested in solving viscous flows problems using the RB framework.

Chapter 1 recalls the basic ingredients of the RB method, with emphasis on the greedy algorithm [81, 96], but also on some collateral tools like the empirical interpolation method [7] that allows an affine representation of the parametrized operator, together with a detailed description of the transfinite map and its development and improvement.

Being able to simulate the solution in several deformations of the same geometry with very small computational time represents a significant advantage with respect to classical numerical techniques. However the RB method can deal with only deformations of the same geometry, and that can represent a limit, especially if we need a quite relevant number of parameters to describe the computational domain. The scheme represented in Figure 1 gives a very general idea on the way the classical RB method deals with the geometrical parametrized configurations.

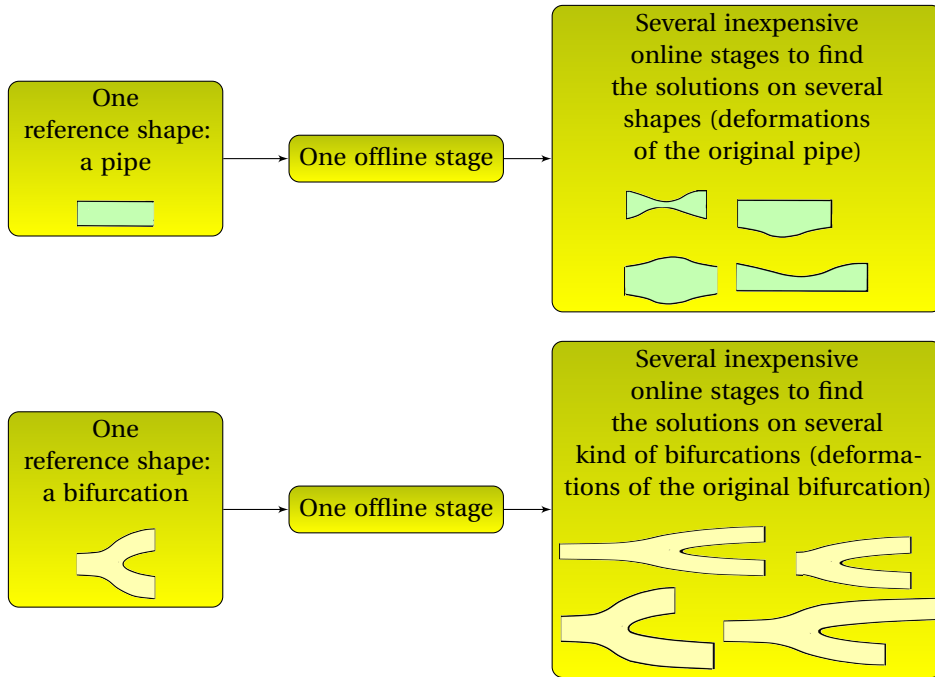


Figure 1: General scheme illustrating the way the classical RB method deals with different geometrical deformations.

In the classical RB method only a reference shape is admitted and it is possible to find accurate and inexpensive solutions on several deformations of the same reference shape. The limit of the RB method is that for each new shape (not recoverable by a deformation of the previous one, like pipes and bifurcations) we have to consider another offline stage and therefore a separate application of the RB method. Moreover the two problems cannot be linked anymore, so that if we want to deal with a geometry composed by two successive bifurcations, we have to define a shape able to recover that

geometry and its deformations and perform a further independent offline stage. This procedure has to be done for any possible new desired geometrical configuration. In addition to the fact that this implies more computational offline steps (one for each new geometry), we point out the difficulty to parametrize a complex domain and the fact that the performance (and efficiency) of the classical RB method is strongly affected if the number of parameters involved in the problem increases.

The aim of this thesis is to find a suitable methodology in order to overcome this limit and being able to reach a very high flexibility by combining virtually any kind of deformable geometries and to not have geometrical limitations in terms of repetitive complex networks. We present some convenient strategies to combine the RB technique with a domain decomposition (DD) technique in order to link different and independent RB classical offline stages, and to be able to perform simulations not only in possible deformed shapes but also in several network combinations of possible deformed shapes, as generally shown by the scheme in Figure 2.

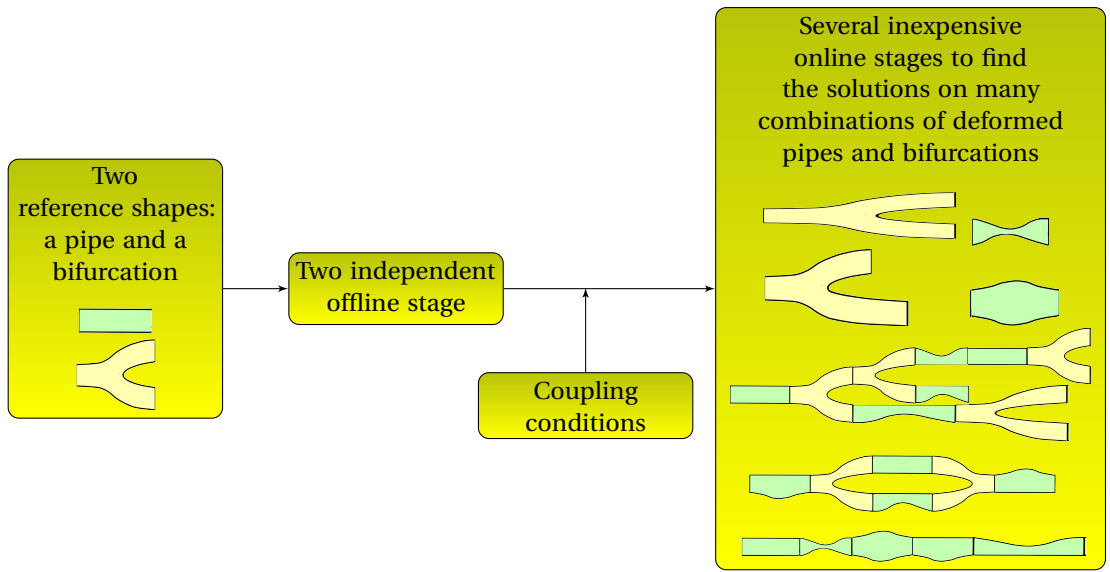


Figure 2: General scheme to combine RB with DD approaches to deal with different geometrical deformations.

The challenge is to find appropriate conditions to couple the snapshots (solutions) defined separately and independently in order to ensure the continuity of the global solution along every possible final geometrical configuration (network). This task is especially difficult for viscous flow problems (that involve velocity and pressure simultaneously) for which we have to guarantee the divergence free condition in the whole domain together with the continuity of the normal stresses at subdomain interfaces [76].

After an introduction regarding the classical RB method and the transfinite map adopted to address the geometrical transformation of the computational domain in Chapter 1, **Chapter 2** is devoted to the first strategy that we propose to couple DD and RB methods: the *reduced basis hybrid method* (RBHM). The RBHM is defined upon some ideas introduced in the reduced basis element method (RBEM) [58], widely investigated for linear elliptic coercive problems [59] and for the Stokes equations [52]. The common idea is the use of Lagrange multipliers to guarantee the continuity of the velocity at the subdomains interfaces: the original contribution of the newly introduced RBHM is the computation of a global coarse solution, in order to recover the continuity of the normal stresses, as well as the continuity of the velocity at the subdomain interfaces. As it will be explained in the method

presentation, the adopted computation for the local snapshots depends on the relative position of the corresponding subdomain (inflow, central or outflow subdomain) in the given network. Because it represents an extension of the already existing RBEM, the RBHM has been introduced directly on parametrized steady Stokes problems. RBHM represents a global enrichment by a coarse solution, computed online of local fine ones computed offline.

The second new developed methodology is the *reduced basis- domain decomposition-finite element* combined approach (RDF) and it will be presented in **Chapter 3**. We start dealing with a simple Poisson problem in order to introduce all the necessary ingredients. The idea underlying this method is to use the DD analytical approach to decompose the problem into suitable subdomains and to use the Steklov-Poincaré interface operators computed w.r.t. the variable sitting at the interfaces or at a region that is extended around the interfaces. This decomposition allows the application of a ROM technique in the subdomain and to automatically obtain the FE equations involving the interface degrees of freedom that will be used to fulfill the coupling conditions satisfying by the global solution. Then the approximate solution is found as linear combination of precomputed local snapshot defined in each subdomain plus some finite element bases defined along the interface regions of the domain. The computation of the local snapshots is performed in a more general way with respect to the previous proposed method (RBHM) and it is independent of the relative position of the subdomain in the given composite network.

The application of the RDF method to the steady Stokes problem is presented in **Chapter 4** and several numerical tests are carried out.

The proposed methodologies are particularly effective when we deal with computational domain represented by repetitive similar shapes that can define different fluidic networks. A possible field of application of these approaches is in the framework of cardiovascular fluid-dynamics, where we consider small fluidic districts or sequential networks. During the last decades, a growing importance has been devoted to this discipline as a key factor in describing some pathologies affecting the cardiovascular system [3, 24]. Being able to perform simulations “almost” in real time and in the many query context with a reasonable level of accuracy represents one of the main features of cardiovascular simulations in daily diagnosis or risk evaluation procedures. Indeed, in this range of applications, a big challenge is to speed up the computational time with rapid and efficient strategies that allow to approximate numerically fluid flows in complex and realistic configurations where specific topological features are recurrent and similar.

Chapter 5 contains some numerical applications of the proposed techniques dealing with possible models of 3D portions of the cardiovascular network and more complex 2D configurations.

Finally some concluding remarks and perspectives on future developments of this work will follow in the final chapter of **Conclusions**.

1 The reduced basis method for a single domain setting

1.1 Introduction

The classical reduced basis method is a common framework and an essential ingredient for the techniques that we present in this thesis.

In this chapter we introduce the reduced basis method for a single domain setting, together with the geometrical reduction that locally we adopt for almost all the cases in this work. The aim of this chapter is both to define a common notation useful for the next chapters and to introduce the first thesis contribution regarding the geometrical domain parametrization. The technique used to parametrize the computational domain is in general arbitrary, we exploit the generalized transfinite map (TM) and we introduce a new extension that maintains the advantage of the TM and improves its versatility.

1.2 The reduced basis method for parametrized PDEs

Reduced basis (RB) method is a very effective technique to efficiently compute numerical solutions for parametrized problems for which not only a single simulation has to be performed, but solutions for a range of different parameter configurations of the same problem are desired.

The reduced basis method was first introduced in the late 1970s for nonlinear structural analysis [2, 66], and subsequently developed in the 1980s and 1990s [6, 8, 23, 70, 78] for a much larger class of parametrized partial differential equations.

The three main ingredients that have a significant role in the computational gains of the method (the global Lagrangian approximation spaces, a rigorous a posteriori error estimators and the offline/online computational decompositions are exploited) have been introduced in [31, 57, 60, 64, 68, 71, 77, 94, 96, 95].

During the last decades, further expansion involving different applications, such as fluid dynamics, and classes of equations, such as Stokes and Navier-Stokes equations, have been presented in [19, 25, 26, 32, 41, 42, 43, 69]. General non-affine problems have been established by using the empirical interpolation method in [7, 30, 83]. Finally, reduced basis approximations and error estimators have also been developed in parabolic partial differential equations by dealing with non-affine and non-linear problems in [29, 79, 80]. Here we consider the reduced basis method built upon finite element “truth” discretization, however there are many alternative “truth” settings: a systematic finite volume framework for RB approximation and a posteriori error estimation is proposed and developed in [34], while spectral element approaches are proposed in [52, 53, 67].

1.2.1 Overview

The reduced basis method is built upon a high fidelity discretization method, as we already mentioned, the option that we adopt here is the finite element (FE) method. Let us note the dimension of the FE space with \mathcal{N} .

The generic idea of the RB method consists in finding an approximate solution of a parametric PDE (μ PDE) as Galerkin projection on a small dimensional space made of precomputed FE solutions.

The parameters involved in the PDE can be related to either geometry and/or physical properties. We consider problems that involve at least one geometrical parameter. For this reason, it is a crucial point to find a suitable map that allows us to deal with reasonable and large enough domain deformations. The choice of the map is arbitrary, we focus our attention on the *transfinite maps*, but many other options are available [49, 55, 61, 62]. Once the choice of the map is done, we can define a suitable reference domain and recast the PDE on it. The geometrical parametric tensors permit to address the parametric dependence on them and to maintain fixed the reference domain. Moreover, if the considered map is affine, it is possible to decouple these tensors in a *parameter independent part* and a *parameter dependent part* [85]. If the map is not affine, an affine approximation can be retrieved through the *empirical interpolation method* [7]. This computational decoupling allows to compute once the FE matrix ($\in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$) during an *offline FE step* and to find a solution for every value of the parameter in a easier way in an *online FE step*. Despite we can allocate many computations of the FE method in the offline step, finding the FE solution is still expensive in terms of computational cost, it depends on the dimension (typically very large) of the linear system that has to be solved. The *affine decomposition* of the parametrized problem is crucial and very helpful during the offline step of the reduced basis. The central part of this step consists in the *greedy algorithm* [81, 96] that is used in order to define the reduced basis space where the final solution will be found. During this step a (typically small) set of N parameter values is properly selected and in correspondence of such set a basis of N solutions of the problem is computed. We note that $N \ll \mathcal{N}$. The *a posteriori error estimation* is a fundamental ingredient of the greedy selection, it permits an efficient and quick parameter space exploration and a reliable reduced space construction. Moreover it is helpful in the online step in order to ensure the accuracy of the approximate RB solution. Once the N snapshot solutions have been computed, after an orthogonalization process, it is possible to define the reduced problem, that consists in finding the approximate solution in the previously built reduced basis space. Since the RB functions are special FE solutions, an advantageous matrix assembling can be done, by pre and post multiplying the FE matrix by the matrices containing the RB functions, following that procedure we can store, during the *offline RB step*, a small (parameter independent) dimensional matrix ($\in \mathbb{R}^{N \times N}$) useful in the final step of the method.

Finally, the *online RB step* consists in assembling the RB matrix and the geometrical tensors (depending on the desired parameter configuration) and to find the solution of a small dimensional linear system. Figure 1.1 shows a synthetic scheme representing the online/offline computational decomposition of both FE and RB methods. In Chapter 5 some numerical results compare the effectiveness of the RB solutions in comparison of the FE ones, often in terms of computational time. Note that the compared computational times refer to both methods the online step.

The next sections contain more details about the different ingredients mentioned above, as well as the steps depicted in Figure 1.1.

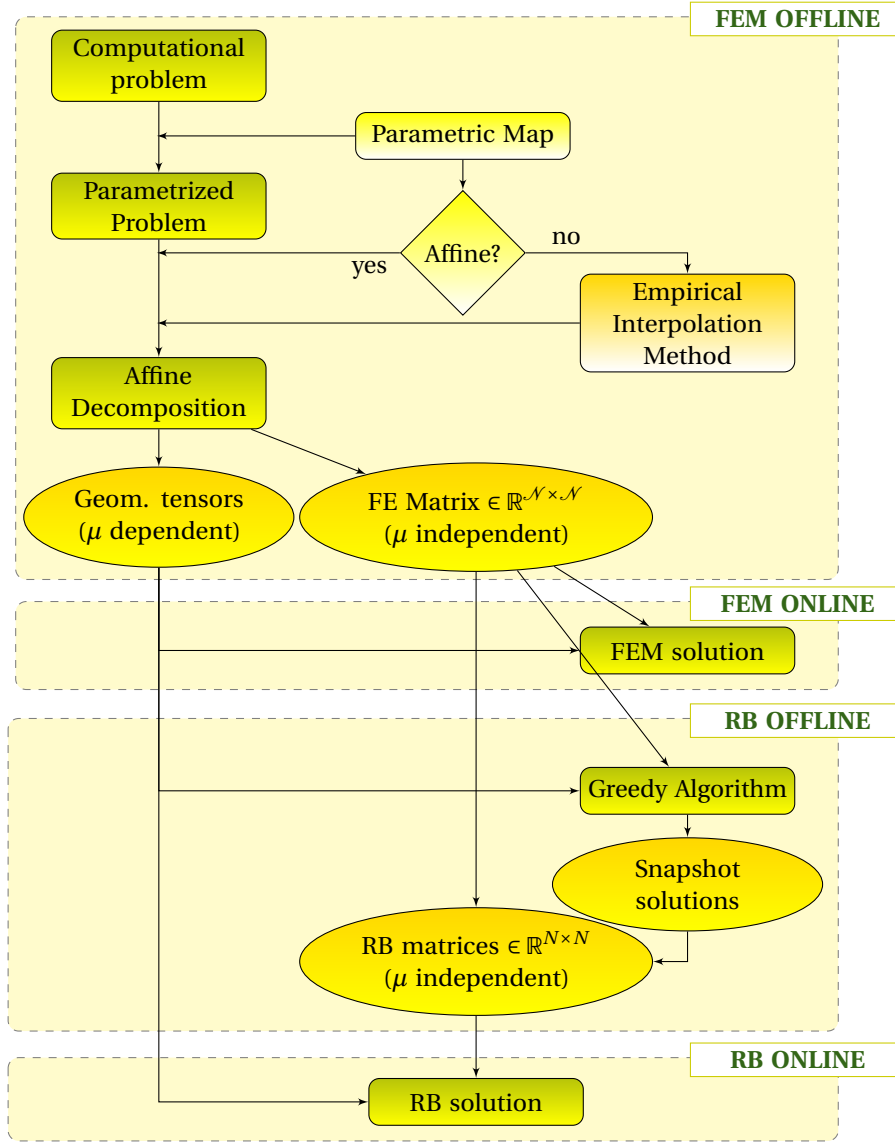


Figure 1.1: Scheme of the offline/online decoupling strategy for both FE and RB methods

1.2.2 Problem description

We consider the computational domain $\Omega(\boldsymbol{\mu})$ as a parametric geometrical deformation of a reference one $\hat{\Omega}$ and we denote with $\boldsymbol{\mu} = (\mu_1, \dots, \mu_R)$ the vector containing the R geometrical parameters. We introduce a map $T(\hat{\mathbf{x}}, \boldsymbol{\mu}) : \hat{\Omega} \rightarrow \Omega(\boldsymbol{\mu})$ that links the domain $\Omega(\boldsymbol{\mu})$ to its corresponding reference one $\hat{\Omega}$. Let us consider two Hilbert spaces $V(\Omega(\boldsymbol{\mu}))$ and $W(\Omega(\boldsymbol{\mu}))$ on $\Omega(\boldsymbol{\mu})$ and their respective duals (denoted by $*$) such that:

$$V(\Omega(\boldsymbol{\mu})) \subset W(\Omega(\boldsymbol{\mu})) = W^*(\Omega(\boldsymbol{\mu})) \subset V^*(\Omega(\boldsymbol{\mu})),$$

where we identified $W(\Omega(\boldsymbol{\mu}))$ with its dual space $W^*(\Omega(\boldsymbol{\mu}))$ thanks to the Riesz theorem [89, 98]. Analogously, we consider two Hilbert spaces $X(\partial\Omega(\boldsymbol{\mu}))$ and $Y(\partial\Omega(\boldsymbol{\mu}))$ on the boundary $\partial\Omega(\boldsymbol{\mu})$ and their

duals such that:

$$X(\partial\Omega(\boldsymbol{\mu})) \subset Y(\partial\Omega(\boldsymbol{\mu})) = Y^*(\partial\Omega(\boldsymbol{\mu})) \subset X^*(\partial\Omega(\boldsymbol{\mu})),$$

where we identified $Y(\partial\Omega(\boldsymbol{\mu}))$ with its dual space $Y^*(\partial\Omega(\boldsymbol{\mu}))$.

We are interested in considering a linear parametric partial differential equation ($\boldsymbol{\mu}$ PDE) in $\Omega(\boldsymbol{\mu})$ of the form:

$$\begin{cases} \mathcal{A}(\mathbf{u}(\boldsymbol{\mu})) &= \mathbf{f}(\boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}) \\ \mathcal{B}(\mathbf{u}(\boldsymbol{\mu})) &= \mathbf{g}(\boldsymbol{\mu}) & \text{on } \partial\Omega(\boldsymbol{\mu}) \end{cases}, \quad (1.2.1)$$

where we denote by: $\mathbf{u}(\boldsymbol{\mu})$ the solution of the problem, $\mathcal{A}(\cdot) : V(\Omega(\boldsymbol{\mu})) \rightarrow V^*(\Omega(\boldsymbol{\mu}))$ the linear differential operator, $\mathcal{B}(\cdot) : V(\Omega(\boldsymbol{\mu})) \rightarrow X(\partial\Omega(\boldsymbol{\mu}))$ represents the Dirichlet boundary conditions operator, $\mathbf{f}(\boldsymbol{\mu})$ and $\mathbf{g}(\boldsymbol{\mu})$ the source term and the boundary data, respectively.

In order to assure the well posedness of the $\boldsymbol{\mu}$ PDE (1.2.1), additional problem specific assumptions may be needed (for example for the Stokes problem an inf-sup condition should be satisfied [27, 10, 9]).

For second order elliptic problems we can choose $V(\Omega(\boldsymbol{\mu})) = H^1(\Omega(\boldsymbol{\mu}))$, $W(\Omega(\boldsymbol{\mu})) = L^2(\Omega(\boldsymbol{\mu}))$, $X(\partial\Omega(\boldsymbol{\mu})) = H^{1/2}(\partial\Omega(\boldsymbol{\mu}))$ and $Y(\partial\Omega(\boldsymbol{\mu})) = L^2(\partial\Omega(\boldsymbol{\mu}))$.

Introducing a lift $\mathbf{R}_g(\boldsymbol{\mu}) \in V(\Omega(\boldsymbol{\mu}))$ of the Dirichlet boundary condition we may decompose $\mathbf{u}(\boldsymbol{\mu})$ as:

$$\mathbf{u}(\boldsymbol{\mu}) = \mathbf{u}^0(\boldsymbol{\mu}) + \mathbf{R}_g(\boldsymbol{\mu}), \quad (1.2.2)$$

where $\mathbf{u}^0(\boldsymbol{\mu}) \in V_0(\Omega(\boldsymbol{\mu})) \equiv \{\mathbf{v} : \mathbf{v} \in V(\Omega(\boldsymbol{\mu})) \text{ s.t. } \mathcal{B}(\mathbf{v}) = \mathbf{0} \text{ on } \partial\Omega(\boldsymbol{\mu})\}$.

In the variational form, the problem of finding $\mathbf{u}^0(\boldsymbol{\mu})$ can be recasted in compact form: find $\mathbf{u}^0(\boldsymbol{\mu}) \in V_0(\Omega(\boldsymbol{\mu}))$ such that $\forall \mathbf{v} \in V_0(\Omega(\boldsymbol{\mu}))$

$$A(\mathbf{u}^0(\boldsymbol{\mu}), \mathbf{v}; \boldsymbol{\mu}) = F(\mathbf{v}; \boldsymbol{\mu}), \quad (1.2.3)$$

where, by referring to the Laplace problem, we define:

$$A(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) = \int_{\Omega(\boldsymbol{\mu})} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\mathbf{x}, \quad F(\mathbf{v}; \boldsymbol{\mu}) = - \int_{\Omega(\boldsymbol{\mu})} \nabla \mathbf{R}_g(\boldsymbol{\mu}) \cdot \nabla \mathbf{v} \, d\mathbf{x}$$

Note that, the formulation (1.2.3) is general enough to embrace second order elliptic problem [15, 47, 97] presented in Chapter 3, while saddle point problems, like the Stokes one [27, 10, 9] will be widely detailed in Chapters 2 and 4.

1.2.3 The parametrized problem in the reference domain and its affine decomposition

Problem (1.2.1) may be easily reformulated on the reference domain $\hat{\Omega}$. We are interested in considering linear $\boldsymbol{\mu}$ PDE in $\Omega(\boldsymbol{\mu})$ of the form:

$$\begin{cases} \hat{\mathcal{A}}(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \hat{\mathbf{f}}(\boldsymbol{\mu}) & \text{in } \hat{\Omega} \\ \hat{\mathcal{B}}(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \hat{\mathbf{g}}(\boldsymbol{\mu}) & \text{on } \partial\hat{\Omega} \end{cases}, \quad (1.2.4)$$

where we used the symbol $\hat{\cdot}$ to denote the operators and variables referred to the reference domain. In fact, by using the map T it is possible to rewrite the bilinear form and the linear functional in the reference domain where the domain is independent of the geometrical parametrization whose effects

are taken into account by the operators. As described in the previous section, by introducing the lift operator, the problem (1.2.4) may be recasted in the compact form: find $\hat{\mathbf{u}}^0(\boldsymbol{\mu}) \in V_0(\hat{\Omega})$ such that $\forall \hat{\mathbf{v}} \in V_0(\hat{\Omega})$:

$$\hat{A}(\hat{\mathbf{u}}^0(\boldsymbol{\mu}), \hat{\mathbf{v}}; \boldsymbol{\mu}) = \hat{F}(\hat{\mathbf{v}}; \boldsymbol{\mu}). \quad (1.2.5)$$

Here we detail an example regarding how the geometrical parameters may be embedded into the variational form. We refer to the classical bilinear form appearing in the Laplace problem:

$$A(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) = \int_{\Omega(\boldsymbol{\mu})} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\mathbf{x}$$

We suppose that the map $T(\hat{\mathbf{x}}, \boldsymbol{\mu})$ is affine and it has the form: $T(\hat{\mathbf{x}}, \boldsymbol{\mu}) = C(\boldsymbol{\mu})\hat{\mathbf{x}} + \mathbf{d}(\boldsymbol{\mu})$.

Integrating by substitution we obtain:

$$\begin{aligned} A(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) &= \int_{\Omega(\boldsymbol{\mu})} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\mathbf{x} \\ &= \int_{\hat{\Omega}} \nu C(\boldsymbol{\mu})^{-T} \hat{\nabla} \hat{\mathbf{u}} \cdot C(\boldsymbol{\mu})^{-T} \hat{\nabla} \hat{\mathbf{v}} |\det C(\boldsymbol{\mu})| \, d\hat{\mathbf{x}} \\ &= \sum_{i,j} \underbrace{[C(\boldsymbol{\mu})^{-1} C(\boldsymbol{\mu})^{-T}]_{ij}}_{\Theta^{ij}(\boldsymbol{\mu})} |\det C(\boldsymbol{\mu})| \nu \underbrace{\int_{\hat{\Omega}} \frac{\partial \hat{\mathbf{u}}}{\partial \hat{x}_j} \frac{\partial \hat{\mathbf{v}}}{\partial \hat{x}_i} \, d\hat{\mathbf{x}}}_{\hat{a}^{ij}(\hat{\mathbf{u}}, \hat{\mathbf{v}})} \\ &= \sum_{i,j} \Theta^{ij}(\boldsymbol{\mu}) \hat{a}^{ij}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) \\ &= \hat{A}(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \boldsymbol{\mu}). \end{aligned}$$

The same procedure may be applied to all the other dualities or internal products appearing in the variational form of the problem and is valid for a generic affine transformation. In the case of a piecewise affine map the procedure should be applied separately on each subregion where the map is affine [85]. More generally, if $T(\hat{\mathbf{x}}, \boldsymbol{\mu})$ is not affine with respect to $\boldsymbol{\mu}$, such that $\mathbf{x} = T(\hat{\mathbf{x}}, \boldsymbol{\mu}) = C(\hat{\mathbf{x}}, \boldsymbol{\mu}) + \mathbf{d}(\boldsymbol{\mu})$, it is possible to approximate it with an affine map, through the empirical interpolation method (see below). This decomposition of the variational form can be applied for an online/offline decomposition even for the FE method.

1.2.3.1 The empirical interpolation method

We provide here an explanation of the principal steps involved in the Empirical Interpolation Method [7]. This technique is adopted when the parametric problem presents a non-affine parametric dependence, in order to recover an affine decoupling of the linear and bilinear forms associated to the equations of the problem¹. As already mentioned, the affine decomposition is a fundamental ingredient of the RB method since it allows to exploit the offline/online procedure and to be able to perform rapid computations during the online stage.

We consider a function $f = f(\mathbf{x}, \boldsymbol{\mu})$ which depends smoothly on a parameter $\boldsymbol{\mu} \in \mathcal{D}$. The idea is to define a discrete and finite subset $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{N_{max}}\} \subset \mathcal{D}$ and the associated approximation spaces $W_n = \text{span}\{\xi_m(\cdot) = f(\cdot, \boldsymbol{\mu}_m), 1 \leq m \leq n\}$ for $n = 1, \dots, N_{max}$ such that for all $\boldsymbol{\mu} \in \mathcal{D}$, $f(\cdot, \boldsymbol{\mu})$ can be well approximated in the space $W_{N_{max}}$. This means, for any value of $\boldsymbol{\mu}$, to find $\theta_i(\boldsymbol{\mu}), i = 1, \dots, N_{max}$ such

¹Many geometrical parametrizations naturally involve non-affine transformation mappings, for example the ones involving geometrical curvatures [82], or approaches involving Free Form Deformation [49], Radial Basis Functions [45] and Transfinite Mapping [54].

that

$$g(\cdot, \boldsymbol{\mu}) = \sum_{i=1}^{N_{max}} \theta_i(\boldsymbol{\mu}) q_i(\cdot), \quad q_i \in W_{N_{max}}$$

is a good approximation of $f(\cdot, \boldsymbol{\mu})$.

We assume to have a suitably large parameter sample over \mathcal{D} denoted by Ξ , and a function $f(\cdot, \boldsymbol{\mu}) \in L^\infty$ of sufficient regularity. We remark that $W_{N_{max}}$ and Ξ depend on the function f . The choice of the initial $\boldsymbol{\mu}_1$ can be arbitrary, we define the first function $\xi_1 := f(\cdot, \boldsymbol{\mu}_1)$ with the corresponding function space $W_1 := \text{span}\{\xi_1\}$.

The identification of the n -th basis function, with $n \geq 2$, is performed in two main steps:

$$\begin{aligned} \epsilon &:= \max_{\boldsymbol{\mu} \in \Xi} \inf_{g \in W_{n-1}} \|f(\cdot, \boldsymbol{\mu}) - g(\cdot, \boldsymbol{\mu})\|_{L^\infty} \\ \boldsymbol{\mu}_n &\leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi} \inf_{g \in W_{n-1}} \|f(\cdot, \boldsymbol{\mu}) - g(\cdot, \boldsymbol{\mu})\|_{L^\infty} \\ \xi_n &:= f(\cdot, \boldsymbol{\mu}_n) \end{aligned}$$

and we define the corresponding function space as $W_{N_{max}} := \text{span}\{\xi_m, m = 1, \dots, N_{max}\}$, where N_{max} is such that ϵ is smaller than a given tolerance ϵ_{tol}^{EIM} .

In the second part of the algorithm we compute the set of interpolation nodes $T_{N_{max}} = \{\mathbf{t}_1, \dots, \mathbf{t}_{N_{max}}\}$. We define $\mathbf{t}_1 := \arg\sup_{\mathbf{x} \in \Omega} |\xi_1(\mathbf{x})|$ and $q_1 := \xi_1(\mathbf{x})/\xi_1(\mathbf{t}_1)$. For $2 \leq n \leq N_{max}$ we define the following matrix:

$$A^{n-1} = \begin{pmatrix} q_1(\mathbf{t}_1) & \cdots & q_{n-1}(\mathbf{t}_1) \\ \vdots & \ddots & \vdots \\ q_1(\mathbf{t}_{n-1}) & \cdots & q_{n-1}(\mathbf{t}_{n-1}) \end{pmatrix},$$

we compute the vector $\mathbf{b}^n = (\xi_n(\mathbf{t}_1), \dots, \xi_n(\mathbf{t}_{n-1}))^T$ and the coefficients vector $\boldsymbol{\sigma}^{n-1}$ given by solving the system $A^{n-1} \boldsymbol{\sigma}^{n-1} = \mathbf{b}^n$.

Each \mathbf{t}_n is the point in Ω where the maximum residual $r_n(\mathbf{x}) := \xi_n - \sum_{j=1}^{n-1} \boldsymbol{\sigma}_j^{n-1} q_j$ is reached. Then we define the function $q_n(\mathbf{x}) = r_n(\mathbf{x})/r_n(\mathbf{t}_n)$. We can see that the basis functions q_i are not the same as the basis functions ξ_i for the spaces W_n , however it can be shown that [7] $\forall n = 1, \dots, N_{max}$: $\text{span}\{q_1, \dots, q_n\} = \text{span}\{\xi_1, \dots, \xi_n\} = W_n$. The detailed procedure is reported in the algorithm 1.2.1.

The algorithm 1.2.1 provides the interpolation points \mathbf{t}_i and the matrix A defined as follow:

$$A = \begin{pmatrix} q_1(\mathbf{t}_1) & \cdots & q_{N_{max}}(\mathbf{t}_1) \\ \vdots & \ddots & \vdots \\ q_1(\mathbf{t}_{N_{max}}) & \cdots & q_{N_{max}}(\mathbf{t}_{N_{max}}) \end{pmatrix}.$$

Hence for any $\boldsymbol{\mu} \in \mathcal{D}$ we compute the vector $F(\boldsymbol{\mu}) = (f(\mathbf{t}_1, \boldsymbol{\mu}), \dots, f(\mathbf{t}_{N_{max}}, \boldsymbol{\mu}))^T$, then the required coefficients $\Theta(\boldsymbol{\mu}) = (\Theta_1(\boldsymbol{\mu}), \dots, \Theta_N(\boldsymbol{\mu}))^T$ of the affine approximation are found by solving the problem

$$A\Theta(\boldsymbol{\mu}) = F(\boldsymbol{\mu}) \tag{1.2.6}$$

The affine approximation of $f(\mathbf{x}, \boldsymbol{\mu})$ is finally given by $g(\mathbf{x}, \boldsymbol{\mu}) = \sum_{i=1}^{N_{max}} \Theta_i(\boldsymbol{\mu}) q_i(\mathbf{x})$.

Algorithm 1.2.1 Empirical Interpolation Procedure

Require: function $f(\mathbf{x}, \boldsymbol{\mu})$, parameter sample Ξ , convergence tolerance ϵ_{tol}^{EIM}
 $\boldsymbol{\mu}_1 \leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi} \|f(\cdot, \boldsymbol{\mu})\|_{L^\infty}$
 $\xi_1 := f(\cdot, \boldsymbol{\mu}_1)$
 $W_1 := \text{span}\{\xi_1\}$
 $err = \max_{\boldsymbol{\mu} \in \Xi} \inf_{g \in W_1} \|f(\cdot, \boldsymbol{\mu}) - g\|_{L^\infty}$
 $n = 1$
while $err > \epsilon_{tol}^{EIM}$ **do**
 $n \leftarrow n + 1$
 $\boldsymbol{\mu}_n \leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi} \inf_{g \in W_{n-1}} \|f(\cdot, \boldsymbol{\mu}) - g\|_{L^\infty}$
 $\xi_n := f(\cdot, \boldsymbol{\mu}_n)$
 $W_n := \text{span}\{\xi_m, m = 1, \dots, n\}$
 $err = \max_{\boldsymbol{\mu} \in \Xi} \inf_{g \in W_{n-1}} \|f(\cdot, \boldsymbol{\mu}) - g\|_{L^\infty}$
end while
 $N_{max} = n$
 $\mathbf{t}_1 \leftarrow \arg\max_{\mathbf{x} \in \Omega} |\xi_1(\mathbf{x})|$
 $q_1 := \xi_1 / \xi_1(\mathbf{t}_1)$
 $A^1 = (A_{11}^1) := (q_1(\mathbf{t}_1))$
for $n = 2, \dots, N_{max}$ **do**
 $\mathbf{b}^n := (\xi_n(\mathbf{t}_1), \dots, \xi_n(\mathbf{t}_{n-1}))^T$
 solve: $A^{n-1} \boldsymbol{\sigma}^{n-1} = \mathbf{b}^n$
 $\mathbf{r}_n := \xi_n - \sum_{j=1}^{n-1} \sigma_j^{n-1} q_j$
 $\mathbf{t}_n \leftarrow \arg\max_{\mathbf{x} \in \Omega} |r_n(\mathbf{x})|$
 $q_n := r_n / r_n(\mathbf{t}_n)$
 $A^n = (A_{ij}^n) := (q_j(\mathbf{t}_i)), 1 \leq i, j \leq n$
end for

1.2.4 Finite element approximation of the problem

The Galerkin method to numerically solve problem (1.2.3) consists in finding an approximate solution $\mathbf{u}_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$, where $V_{\mathcal{N}}$ is a set of subspaces of $V_0(\hat{\Omega})$ with finite dimension \mathcal{N} (typically very large). Therefore the approximated problem becomes: find $\mathbf{u}_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$ such that $\forall \mathbf{v}_{\mathcal{N}} \in V_{\mathcal{N}}$

$$\mathcal{A}(\mathbf{u}_{\mathcal{N}}(\boldsymbol{\mu}), \mathbf{v}_{\mathcal{N}}; \boldsymbol{\mu}) = \mathcal{F}(\mathbf{v}_{\mathcal{N}}; \boldsymbol{\mu}). \quad (1.2.7)$$

Problem (1.2.7) is usually called the Galerkin formulation of problem (1.2.3).

Approximating the resulting problem with the finite element method consists in a particular choice for the subspace $V_{\mathcal{N}}$. We consider a triangulation \mathcal{T}_h of $\hat{\Omega}$ and let K be the generic element of \mathcal{T}_h . An example of FE space can be represented by the space of piecewise linear functions defined as follows:

$$V_{\mathcal{N}} = [X_h^1 \equiv \{v_h \in C^0(\hat{\Omega}) : v_h|_K \in \mathbb{P}_1 \ \forall K \in \mathcal{T}_h\}] \cap H_0^1(\hat{\Omega}). \quad (1.2.8)$$

1.2.5 The reduced basis formulation

The reduced basis formulation consists in finding the solution of the problem (1.2.5) in a subspace $V_0(\hat{\Omega})$ with dimension N much lower than \mathcal{N} .

We consider the RB space defined by the RB function selected by the greedy algorithm:

$$V(\hat{\Omega})_N = \text{span}\{\boldsymbol{\varphi}^i, i = 1, \dots, N\} \subset V_{\mathcal{N}}. \quad (1.2.9)$$

Note that the dimension of the RB space may be larger than N , for example in saddle point problems the reduced space must be enriched in order to fulfill the inf-sup condition [84].

The reduced basis method for the local problem (1.2.5) on the reference block $\hat{\Omega}$ is: find $\mathbf{u}(\boldsymbol{\mu})_N \in V(\hat{\Omega})_N$ such that $\forall \mathbf{v}_N \in V(\hat{\Omega})_N$

$$\hat{\mathcal{A}}(\mathbf{u}(\boldsymbol{\mu})_N, \mathbf{v}_N; \boldsymbol{\mu}) = \hat{\mathcal{F}}(\mathbf{v}_N; \boldsymbol{\mu}), \quad (1.2.10)$$

where $\hat{\mathcal{A}}$ and $\hat{\mathcal{F}}$ represent the bilinear form and linear functional associated to problem (1.2.5).

We suppose that $\boldsymbol{\varphi}_N^i, i = 1, \dots, N$ is a basis of $V(\hat{\Omega})_N$. We recall that in order to improve the choice of the basis, a convenient procedure is to orthonormalize the functions through, for instance, the classical Gram-Schmidt orthogonalization procedure [93]. The Galerkin projection applied also for the reduced basis method guarantees the best-fit approximation property, as well as Galerkin orthogonality.

The local reduced basis problem (1.2.10) can be formulated as a linear system:

$$\mathbf{A}(\boldsymbol{\mu})_N \mathbf{u}(\boldsymbol{\mu})_N = \mathbf{f}(\boldsymbol{\mu})_N, \quad (1.2.11)$$

where $[\mathbf{A}(\boldsymbol{\mu})_N]_{ij} = \hat{\mathcal{A}}(\boldsymbol{\varphi}_N^j, \boldsymbol{\varphi}_N^i; \boldsymbol{\mu})$, $[\mathbf{f}(\boldsymbol{\mu})_N]_i = \hat{\mathcal{F}}(\boldsymbol{\varphi}_N^i; \boldsymbol{\mu})$ and $\mathbf{u}(\boldsymbol{\mu})_N$ represents the vector of coefficients of the RB solution. Moreover, as the functions $\boldsymbol{\varphi}_N^i$ belong to the finite element space $V_0(\hat{\Omega})$, by denoting $\boldsymbol{\varphi}_h^i, i = 1, \dots, \mathcal{N}$ a basis of $V(\hat{\Omega})_{\mathcal{N}}$, we can write them as:

$$\boldsymbol{\varphi}_N^i = \sum_{j=1}^{\mathcal{N}} \varphi_j^i \boldsymbol{\varphi}_h^j, \quad i = 1, \dots, N, \quad (1.2.12)$$

therefore:

$$\mathbf{u}(\boldsymbol{\mu})_N = \sum_{i=1}^N [\mathbf{u}(\boldsymbol{\mu})_N]_i \boldsymbol{\varphi}_N^i = \sum_{i=1}^N \sum_{j=1}^{\mathcal{N}} [\mathbf{u}(\boldsymbol{\mu})_N]_i \varphi_j^i \boldsymbol{\varphi}_h^j, \quad (1.2.13)$$

whence the linear system (1.2.11) can be reformulated with respect to the original FE matrix $\mathbf{A}(\boldsymbol{\mu})_h$ associated to (1.2.7) and source term $\mathbf{f}(\boldsymbol{\mu})_h$ as:

$$[\Phi^T \mathbf{A}(\boldsymbol{\mu})_h \Phi] \mathbf{u}(\boldsymbol{\mu})_N = \Phi^T \mathbf{f}(\boldsymbol{\mu})_h, \quad (1.2.14)$$

where $[\Phi]_{ij} = \varphi_j^i$, $[\Phi] \in \mathbb{R}^{\mathcal{N} \times N}$.

The “pre” and “post” multiplications involved in (1.2.14) permit to drastically reduce the size N of the reduced basis system (1.2.11) that is much smaller than the size \mathcal{N} of the corresponding FE system. However, the RB matrix is, in principle, full while the FE one is sparse. In general, as rule of thumb, the RB method becomes effective if the resolution time required for a new query of $\boldsymbol{\mu}$ is much smaller than the one required for the solution of the original FE system.

It is worth noting that the RB matrix and source term depend on the actual parameter (online computation). The linear and bilinear form decomposition (like the one detailed in example in Section 1.2.3) is used to efficiently perform this step by summing pre-built matrices, computed offline.

1.2.5.1 The greedy algorithm

The greedy algorithm is an efficient technique for the selection of the basis functions $\{\boldsymbol{\varphi}^1, \dots, \boldsymbol{\varphi}^N\}$ of the reduced basis space V_N (as subset of the FE space $V_{\mathcal{N}}$).

We suppose that we have defined the first N basis functions and we look for the value of $\boldsymbol{\mu}$ that defines the next basis function. We distinguish two key computational tasks in the greedy algorithm that can

be encapsulated in two subroutines `AssemblingData` and `ErrorEvaluation` [46]:

- taking the current set of N basis functions V_N as input and develop the *online dataset* needed to evaluate the RB approximation and associated error bounds (such as the matrix assembling and the normalization procedure through the projector Π_N onto V_N), such that $\text{OnlineSet} := \text{AssemblingData}(V_N)$;
- taking a discrete parameter set Ξ_{train} of \mathcal{D} as input and returns the parameter $\boldsymbol{\mu}^*$ which *maximizes* the prediction of the error between the RB solution (by using the previous selected RB basis) and the FE solution. This prediction is represented by the posteriori error bound $\Delta_N(\boldsymbol{\mu})$. So that: $\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \Xi} \Delta_N(\boldsymbol{\mu})$, $\text{err} = \Delta_N(\boldsymbol{\mu}^*)$, $[\boldsymbol{\mu}^*, \text{err}] := \text{ErrorEvaluation}(\Xi_{train}, \text{OnlineSet})$.

Since the evaluation of the error bound is inexpensive, we are usually able to use relatively large training sets and obtain good exploration of \mathcal{D} . Nevertheless, if the problem presents a large number of parameters, we need to choose very large training sets in order to obtain a reasonable exploration of the parameter domain and the error bound sampling tends to be very expensive in terms of computational costs and times.

Algorithm 1.2.2 Greedy algorithm

Require: Specify $\Xi_{train} \subset \mathcal{D}$ of size n_{train} and a tolerance ϵ , select $\boldsymbol{\mu}^* \in \mathcal{D}$ (arbitrary)

$N \leftarrow 0$

$V_0 = \emptyset$

while $\text{err} > \epsilon$ **do**

$\boldsymbol{\varphi}^{N+1} := (I - \Pi_N) \mathbf{u}^{\mathcal{N}}(\boldsymbol{\mu}^*)$ (normalized)

$V_{N+1} \leftarrow V_N \oplus \text{span}\{\boldsymbol{\varphi}^{N+1}\}$

$N \leftarrow N + 1$

$\text{OnlineSet} := \text{AssemblingData}(V_N)$

$[\boldsymbol{\mu}^*, \text{err}] := \text{ErrorEvaluation}(\Xi_{train}, \text{OnlineSet})$

end while

$N_{max} \leftarrow N$

We note that we always use the greedy sampling methods for the selection of the reduced space in the steady case, an alternative technique such as the Proper Orthogonal Decomposition (POD) method [48, 5] (most often applied in the temporal domain) has been exploited in literature. POD techniques can be and have successfully been applied within the parametric RB context in the time dependent case [34, 65]. Other POD applications in reduced order modelling are discussed in [14, 16, 33, 56].

1.2.5.2 A posteriori error estimation

A rigorous error estimation has two main roles in the RB method: to control the error between the approximate RB solution and the FE solution and to drive, during the greedy algorithm, the choice of the $(n + 1)$ -th basis function once the first n are already available. The calculation of the RB error bound admits an offline/online decomposition. The offline stage, performed once, is very expensive and \mathcal{N} -dependent, while the online evaluation, performed many times for each new desired $\boldsymbol{\mu}$, is very inexpensive and \mathcal{N} -independent. This efficient and reliable error estimation permits to predict the RB error with respect to the FE solution without computing the latter. It is crucial during the greedy algorithm to speed up the efficient selection of the snapshots.

We report here a brief introduction of the principal consideration involved in the error estimation theory.

We denote with $\mathbf{u}_{\mathcal{N}}(\boldsymbol{\mu})$ the FE solution of the problem (1.2.3) (i.e. the solution of (1.2.7)) and $\mathbf{u}_N(\boldsymbol{\mu})$, its RB approximation (i.e. the solution of (1.2.11)), so that the error $e(\boldsymbol{\mu}) := \mathbf{u}_{\mathcal{N}}(\boldsymbol{\mu}) - \mathbf{u}_N(\boldsymbol{\mu}) \in V_{\mathcal{N}}(\Omega)$ satisfies

$$\mathcal{A}(e(\boldsymbol{\mu}), \mathbf{v}; \boldsymbol{\mu}) = r(\mathbf{v}; \boldsymbol{\mu}), \quad \forall \mathbf{v} \in V_{\mathcal{N}}(\Omega). \quad (1.2.15)$$

where $r(\cdot; \boldsymbol{\mu}) \in V_{\mathcal{N}}(\Omega)^*$ is the residual defined as follows

$$r(\mathbf{v}; \boldsymbol{\mu}) := \mathcal{F}(\mathbf{v}; \boldsymbol{\mu}) - \mathcal{A}(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{v}; \boldsymbol{\mu}), \quad \forall \mathbf{v} \in V_{\mathcal{N}}(\Omega). \quad (1.2.16)$$

We introduce [68] $\hat{e}(\boldsymbol{\mu}) \in V_{\mathcal{N}}(\Omega)$, the Riesz representation of $r(\cdot; \boldsymbol{\mu})$, satisfying

$$(\hat{e}(\boldsymbol{\mu}), \mathbf{v})_V = r(\mathbf{v}; \boldsymbol{\mu}), \quad \forall \mathbf{v} \in V_{\mathcal{N}}(\Omega). \quad (1.2.17)$$

This allows us to write the error residual equation (1.2.15) as

$$\mathcal{A}(e(\boldsymbol{\mu}), \mathbf{v}; \boldsymbol{\mu}) = (\hat{e}(\boldsymbol{\mu}), \mathbf{v})_V, \quad \forall \mathbf{v} \in V_{\mathcal{N}}(\Omega) \quad (1.2.18)$$

and it follows that the dual norm of the residual can be evaluated through the Riesz representation:

$$\|r(\cdot; \boldsymbol{\mu})\|_{V(\Omega)^*} := \sup_{\mathbf{v} \in V(\Omega)} \frac{r(\mathbf{v}; \boldsymbol{\mu})}{\|\mathbf{v}\|_V} = \|\hat{e}(\boldsymbol{\mu})\|_V; \quad (1.2.19)$$

We define a positive, parametric lower bound function $\alpha_{\text{LB}}(\boldsymbol{\mu})$ for the FE coercivity constant defined as

$$\alpha(\boldsymbol{\mu}) = \inf_{\mathbf{w} \in V_{\mathcal{N}}} \frac{A(\mathbf{w}, \mathbf{w}; \boldsymbol{\mu})}{\|\mathbf{w}\|_V^2} \quad (1.2.20)$$

such that: $0 < \alpha_{\text{LB}}(\boldsymbol{\mu}) \leq \alpha(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}$.

In more general non-coercive problems, the stability factor is represented by the inf-sup constant $\beta(\boldsymbol{\mu})$ as it will be defined in Chapter 2, Section 2.3.

So that, the a posteriori error bound requires an estimate for the stability factor (coercivity constant or inf-sup constant) associated with the partial differential operator. This stability factor estimate must satisfy several requirements: (i) it must be a provably strict lower bound for the true stability factor (the one associated with the FE discretization); (ii) it must be a reasonably accurate approximation ($O(1)$ relative error) of the true stability factor; (iii) it must admit an offline/online computational treatment, based on the affine decomposition of the partial differential equation, for which the Online effort is independent of \mathcal{N} . There are several approaches to provide rigorous lower bounds for the stability factor. A natural norm method is proposed in [17, 91]. A Successive Constraint Method (SCM) is proposed in [13, 35, 85]. A combination of the linearized inf-sup statement introduced in [91] with the SCM lower bound procedure is proposed in [37].

After the introduction of the dual norm of the residual (1.2.19) and a lower bound for the stability factor, we now define error estimators for the solution as

$$\Delta_N(\boldsymbol{\mu}) := \frac{\|\hat{e}(\boldsymbol{\mu})\|_V}{(\alpha_{\text{LB}}(\boldsymbol{\mu}))^{1/2}}. \quad (1.2.21)$$

Some applications and properties concerning a posteriori error estimation, such as effectivities, are presented e.g. in [85, 71, 72].

1.3 Transfinite maps

In this section we introduce the method to generate parametrized transfinite maps (TM) which can be seen as a generalization of the Gordon-Hall transfinite interpolation approach for quadrilaterals [28]. As already mentioned, these maps can be used to deform the computational domain in which we want to solve the parametrized problem.

The transfinite maps induce a non-affine geometrical parametrization so that the empirical interpolation method is necessary to recover the affinity of the linear and bilinear forms of the considered problems.

We first recall the generalized transfinite map proposed in [53, 54] and used recently in [40], which however are not suitable to deal with particular configurations, e.g. when the domain is not centered around the axis origin or when we want to consider an edge of the domain parametrized by subparts (as shown in Section 1.3.1.3). For that reason, we propose a new version to overcome these problems.

1.3.1 Generalized transfinite map

The idea behind the transfinite map (TM) is to deform the interior points of the physical domain through a linear combinations of deformations of the points belonging to the boundaries, that are easily parametrized through one dimensional functions.

We assume a general two-dimensional domain Ω and a general reference domain $\hat{\Omega}$, we suppose that both are curved polygons with the same number n of curved edges. Γ_i denotes the generic edge in Ω , $\hat{\Gamma}_i$ denotes the corresponding edge in $\hat{\Omega}$; the edges are numbered clockwise.

The ingredients of the TM are three functions that have to be found on each edge of the domain Ω : the *weight function*, the *projection function* and the *parametric function* describing, through a parameter $\mu \in \mathcal{D}$, the deformation of the edge. The computation of the first two types of functions are quite expensive but are independent of the deformations and can be obtained from the solution of proper Laplace problems on the reference domain $\hat{\Omega}$. Due to this advantageous computational features, an offline/online computational decoupling procedure can be applied also for the evaluation of the maps.

1.3.1.1 Offline stage

For each edge $\hat{\Gamma}_i, i = 1, \dots, n$ of the reference domain $\hat{\Omega}$ (with n -side) we define a weight function φ_i by solving the following Laplace problem:

$$\begin{cases} \Delta \varphi_i = 0 & \text{in } \hat{\Omega}, \\ \varphi_i = 1 & \text{on } \hat{\Gamma}_i, \\ \frac{\partial \varphi_i}{\partial n} = 0 & \text{on } \hat{\Gamma}_j, j = i-1, i+1, \\ \varphi_i = 0 & \text{on } \hat{\Gamma}_j, j \neq i-1, i, i+1. \end{cases} \quad (1.3.1)$$

We represent in Figure 1.2 a scheme concerning the boundary conditions for the case of a reference bifurcation domain, we use the notational convention that if $i = 1$, $\hat{\Gamma}_{i-1} = \hat{\Gamma}_n$, and if $i = n$, $\hat{\Gamma}_{i+1} = \hat{\Gamma}_1$.

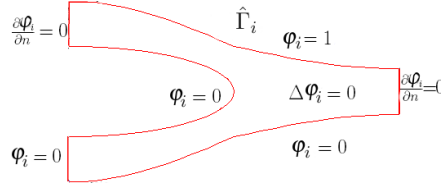


Figure 1.2: Graphical representation of the boundary conditions for problem (1.3.1) in a reference bifurcated domain.

To define the generalized transfinite map we also need to define an operator that “projects” the internal part of the reference domain onto each side $\hat{\Gamma}_i$. For that, we compute the projection function π_i associated to the side $\hat{\Gamma}_i$, by solving the Laplace problem:

$$\begin{cases} \Delta \pi_i = 0 & \text{in } \hat{\Omega}, \\ \pi_i = t & \text{on } \hat{\Gamma}_i, \\ \pi_i = 0 & \text{on } \hat{\Gamma}_{i-1}, \\ \pi_i = 1 & \text{on } \hat{\Gamma}_{i+1}, \\ \frac{\partial \pi_i}{\partial n} = 0 & \text{on } \hat{\Gamma}_j, j \neq i-1, i, i+1, \end{cases} \quad (1.3.2)$$

the Dirichlet boundary condition along $\hat{\Gamma}_i$ corresponds to the arc-length t ranging from 0 to 1, such that $t = 0$ in \mathbf{x}_{i-1} , the vertex shared by $\hat{\Gamma}_{i-1}$ and $\hat{\Gamma}_i$, $t = 1$ in \mathbf{x}_i , the vertex shared by $\hat{\Gamma}_i$ and $\hat{\Gamma}_{i+1}$. On the sides adjacent to $\hat{\Gamma}_i$ we set π_i equal to either 0 or 1, and on the remaining sides we impose homogeneous Neumann boundary conditions (Figure 1.3).

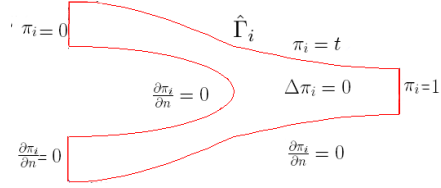


Figure 1.3: Graphical representation of the boundary condition for problem (1.3.2) in a reference bifurcated domain.

Thus, for each side of the reference domain, we associate one weight function and one projection function by solving the problems (1.3.1) and (1.3.2) respectively (Figure 1.4). For a domain with n sides, we have to solve $2n$ elliptic problems, however these computations are independent of the deformation (and of the parameter μ) and are included in the offline stage of the reduced basis method.

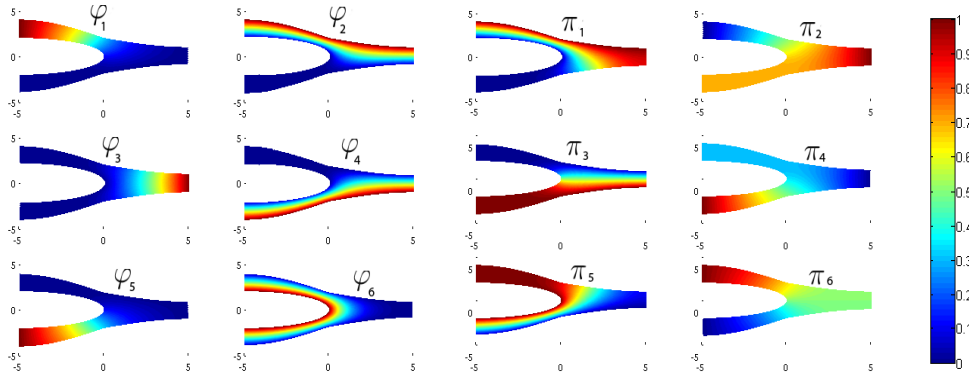


Figure 1.4: Weight functions φ_i , solutions of (1.3.5) (left) and projection functions π_i , solutions of (1.3.2) (right) for a reference bifurcation; $1 \leq i \leq 6$.

1.3.1.2 Online stage

We assume that each edge Γ_i is parametrized through the parameter $\boldsymbol{\mu} \in \mathcal{D}$ by a bijective map $\boldsymbol{\psi}_i$ from $[0, 1] \times \mathcal{D}$ into Γ_i , so that $\boldsymbol{\psi}_i(1, \boldsymbol{\mu}) = \mathbf{x}_i$, where \mathbf{x}_i denotes the vertex shared by Γ_i and Γ_{i+1} and $\boldsymbol{\psi}_i(0, \boldsymbol{\mu}) = \mathbf{x}_{i-1}$. We denote by $\hat{\mathbf{x}}$ a generic point of the reference domain $\hat{\Omega}$ and by \mathbf{x} a generic point of the deformed domain Ω . The transfinite map is then defined as follows:

$$\mathbf{T}(\hat{\mathbf{x}}, \boldsymbol{\mu}) = \sum_{i=1}^n \{ \varphi_i(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(\pi_i(\hat{\mathbf{x}}), \boldsymbol{\mu}) - \varphi_i(\hat{\mathbf{x}}) \varphi_{i+1}(\hat{\mathbf{x}}) \mathbf{x}_i \}. \quad (1.3.3)$$

The advantage of using such a map in the context of a RB approach, is that in the online stage we need only to compute the inexpensive boundary functions $\boldsymbol{\psi}_i(\pi_i(\hat{\mathbf{x}}), \boldsymbol{\mu})$ and perform the linear combination of the harmonic functions in (1.3.3) [28, 52].

1.3.1.3 Numerical examples

In this section we present some numerical tests obtained with the generalized TM, on the other side we explain the critical problem that we can notice for instance with a stenosis domain. First of all, we consider the weight and projection functions obtained for a reference bifurcation (represented in Figure 1.4) and we consider the edges described by three different parameters: μ_1 represents the length of the bifurcation, μ_2 the thickness of each branch and μ_3 the span between the branches. Figure 1.5 shows some examples of bifurcation deformations obtained by applying the transformation (1.3.3).

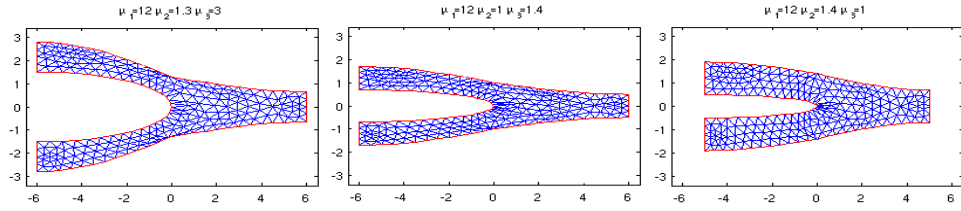


Figure 1.5: Different bifurcation deformations.

A deformed pipe (which may be regarded as the longitudinal section of an arterial vessel) represents a suitable example to show that the described map could induce some problems. We parametrized upper and lower walls with two parameters, μ_1 and $\mu_2 \in [-1, 1]$, that represent the dilatation and the contraction of the pipe. If the upper and lower boundary curves are parametrized through just two functions $\boldsymbol{\psi}_i$, we are able to obtain good deformations of the domain. Figure 1.6 shows the weight functions φ_i and projection functions π_i for the reference domain, represented by a straight pipe ($\mu_1 = 0, \mu_2 = 0$), by setting $n = 4$, while Figure 1.7 shows some possible pipe deformations.

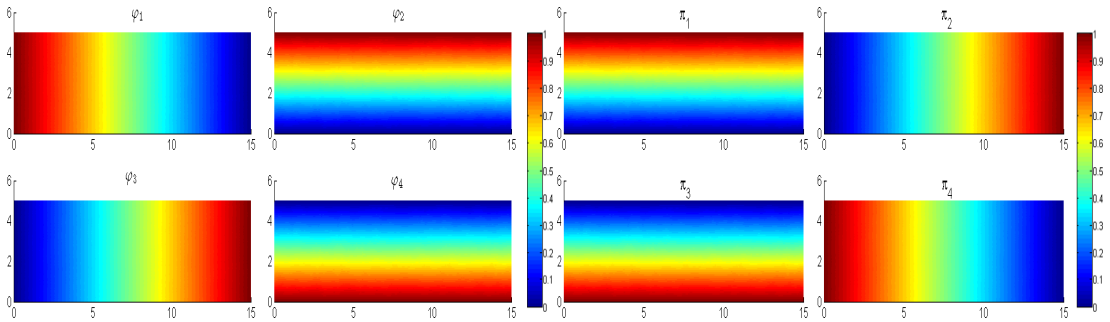


Figure 1.6: Weight functions φ_i (left) and projection functions π_i (right) for the reference straight pipe; $1 \leq i \leq 4$.

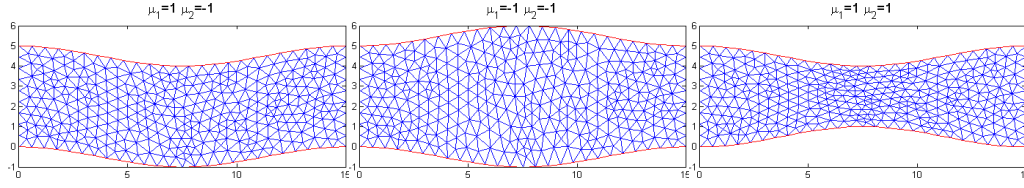


Figure 1.7: Examples of different pipe deformations.

If we want to parametrize the upper or the lower edges with more than one function ψ_i we need to define both the reference domain $\hat{\Omega}$ and the computational one Ω with a larger number of edges. This imposition means that we need to compute more weight and projection functions. We consider for instance the more complex stenosis parametrization represented in Figure 1.8, where, in particular, the four parameters μ_4, μ_5, μ_7 and μ_8 determine the length and the arterial thickness. Moreover μ_5 and μ_8 can also be negative, this can be interpreted as a blown blood vessel, we address to the Figure 1.8 for the other parameters representations.

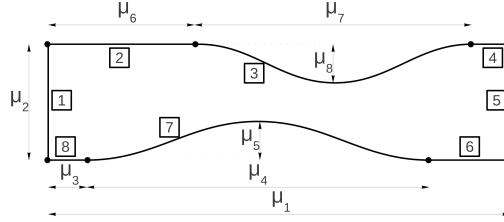


Figure 1.8: Stenosis Geometry

We still consider a straight pipe as reference domain (corresponding to $\mu_5 = 0, \mu_8 = 0$), the weight and projection functions are represented in Figure 1.9.

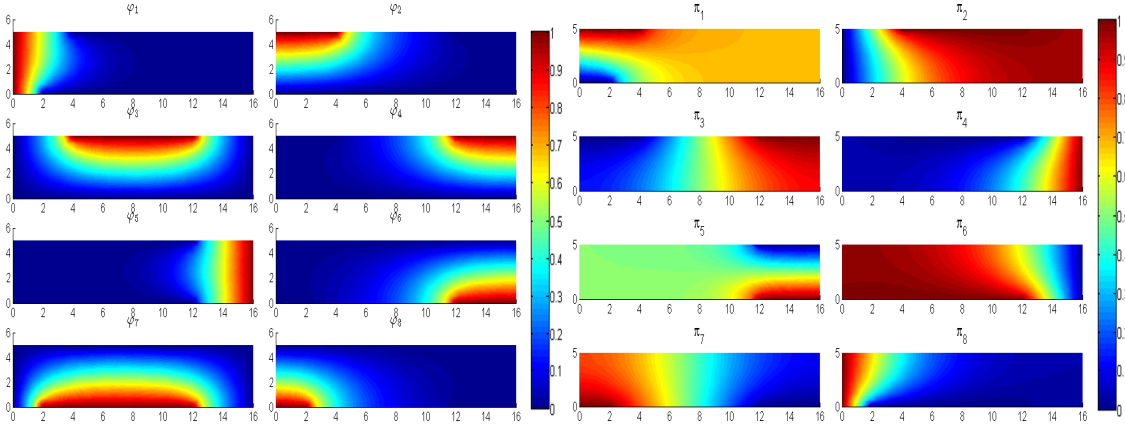


Figure 1.9: Weight functions φ_i , solutions of (1.3.5) (on the left) and projection functions π_i , solutions of (1.3.2) (on the right) for the reference straight pipe; $1 \leq i \leq 8$.

By applying the generalized transfinite map (1.3.3) to a new configuration of μ , we obtain a badly deformed geometry, (see Figure 1.10, at left). We can observe that we can improve the result by centering the geometry in the origin of the axis. The deformation is still not suitable, but it keeps the internal nodes at least inside the geometry (see Figure 1.10, at right).

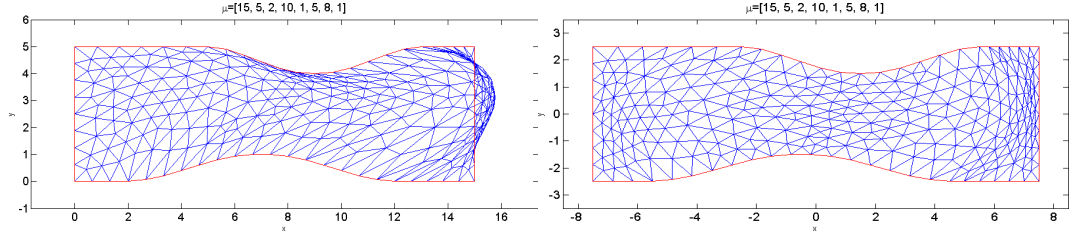


Figure 1.10: Stenosis deformations corresponding to $\mu = [15, 5, 2, 10, 1, 5, 8, 1]$, on the left it is not centered around the axes, on the right it is centered around the point (0,0).

Due to the fact that the map (1.3.3) does not perturb the reference points of the geometry $\hat{\Omega}$, but it reassigns them a position on the boundary and then repositions them on the geometry through the weights, it is strongly affected by the coordinates of the points.

We propose in the next section an improvement of the generalized map that takes in account this issue.

1.3.2 The boundary displacement dependent transfinite map

Motivated by the results in the previous sections we introduce here an extension of the generalized TM, with the aim of keeping the good properties while solving the critical issues. More precisely we want the TM to become independent of the position of the geometry in the plane \mathbb{R}^2 .

The basic idea of the *Boundary Displacement Dependent Transfinite Map (BDD TM)* [44] is to keep into account the original positions of the points in the reference domain $\hat{\Omega}$ and to move them by weighting only the difference between the reference boundaries of $\hat{\Omega}$ and the deformed boundaries of Ω . The convenient online/offline computational decoupling can still be maintained as we will explain in the next section.

1.3.2.1 Offline/Online decoupling

The offline computations coincide exactly with the offline part of the previous detailed transfinite map and it consists in computing the weight functions φ_i and the projection functions π_i for each edge of the reference domain.

In the online stage, together with the parametrized boundary functions ψ_i (as in the generalized transfinite map), we define the displacement function. As before, each boundary of the domain is parametrized by a function $\psi_i : [0, 1] \times \mathcal{D} \rightarrow \Gamma_i$, so that if we fix the reference parameter μ_{ref} it is possible to define the boundary of the reference domain ($\psi_i : [0, 1] \times \{\mu_{ref}\} \rightarrow \hat{\Gamma}_i$).

Thus we define the displacement function as follows:

$$\mathbf{d}_i(t, \mu) = \psi_i(t, \mu) - \psi_i(t, \mu_{ref}), \forall t \in [0, 1].$$

To each point on the boundary, this function associates the “distance” between the deformed position and the reference one.

Finally the BDD transfinite map is defined as:

$$\mathbf{T}(\hat{\mathbf{x}}, \mu) = \hat{\mathbf{x}} + \sum_{i=1}^n \{ \varphi_i(\hat{\mathbf{x}}) \mathbf{d}_i(\pi_i(\hat{\mathbf{x}}), \mu) - \varphi_i(\hat{\mathbf{x}}) \varphi_{i+1}(\hat{\mathbf{x}}) \mathbf{d}_i(1, \mu) \}. \quad (1.3.4)$$

1.3.2.2 An example

In order to compare the quality of two different deformations obtained with the original TM and the BDD TM, we consider the second example of stenosed geometry (represented in Figure 1.8).

In Figure 1.10 the deformations obtained with the TM are shown. Figure 1.11 shows on the left the deformation with the same parameters selection used in 1.10, we observe a considerable improvement of the points deformation. In the same figure we report other two examples of even bigger deformations. The BDD transfinite map allows to deal with more general parametrized geometries positioned arbitrarily in the axis plane.

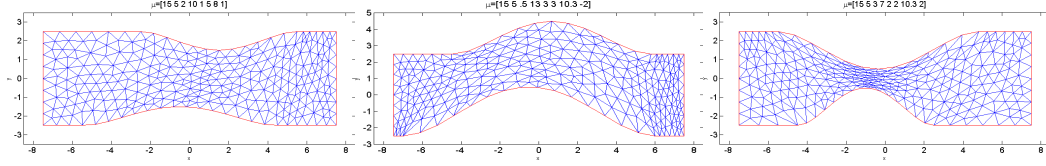


Figure 1.11: Different stenosis deformations obtained with the BDD TM.

1.3.3 Transfinite maps for 3D geometries

We introduce in this section a first exploration of the transfinite map extended to 3D geometries. The idea that we propose is to maintain the same structure of the two dimensional map in terms of offline/online decomposition, here the role played by the boundaries in the 2D domain is substituted by the edge of the 3D domain.

1.3.3.1 Offline stage

We consider a general domain Ω and a general reference domain $\hat{\Omega}$, that are polyhedrons with the same number n of edges. Γ_i denotes the generic edge of Ω , $\hat{\Gamma}_i$ denotes that of $\hat{\Omega}$. For every edge $\hat{\Gamma}_i, i = 1, \dots, n$, we associate one weight function φ_i and one projection function π_i computed on the reference domain $\hat{\Omega}$. These functions solve Laplace problems with specific boundary conditions, for simplicity we describe two cases reported in Figure 1.12 that refer to parallelepipedal geometries. The Laplace problems for the definition of the weight and projection functions are the following:

$$\Delta \varphi_i = 0 \quad \text{in } \hat{\Omega}, \quad \Delta \pi_i = 0 \quad \text{in } \hat{\Omega}. \quad (1.3.5)$$

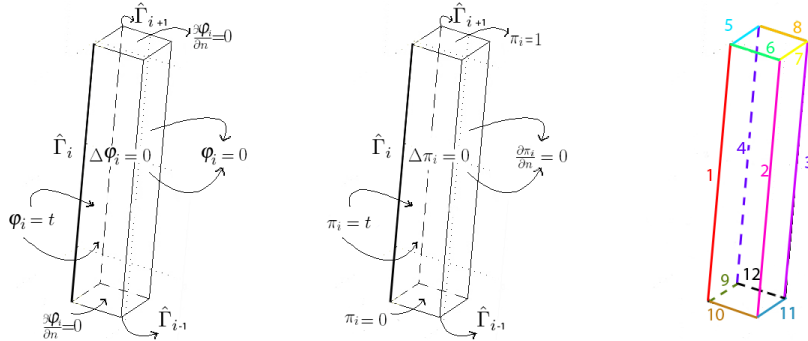


Figure 1.12: Graphical representation of the edge classification and of the BC for weight and projection functions in a reference domain for a 3D TM setting.

The boundary condition $\varphi_i = t$, shown in Figure 1.12, represents the Dirichlet boundary condition equal to the function t that represents a linear function equal to 0 on $\hat{\Gamma}_{i-1}$ and $\hat{\Gamma}_{i+1}$ and equal to 1 on $\hat{\Gamma}_i$. While the boundary condition $\pi_i = t$ indicates Dirichlet boundary condition equal to the function t that represents a linear function equal to 0 on the base of $\hat{\Omega}$ and equal to 1 on the upper face of $\hat{\Omega}$. Figures 1.13 and 1.14 show the weight and the projection functions associated to each edge of the reference domain numbered as in Figure 1.12.

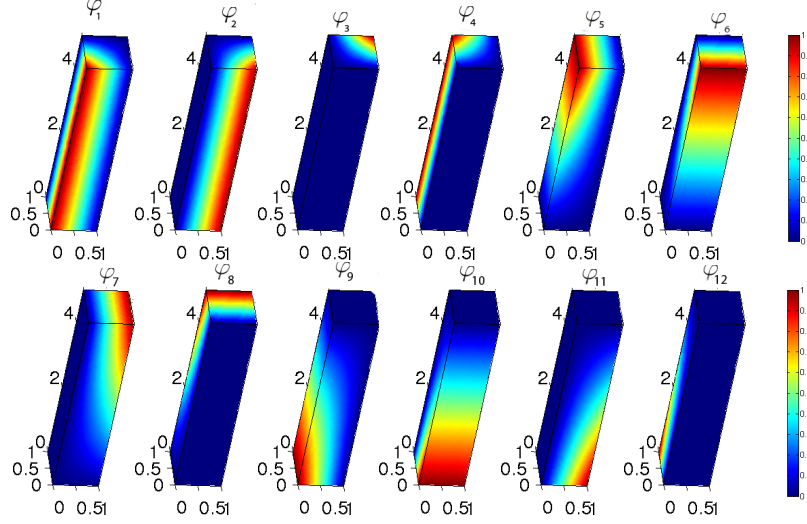


Figure 1.13: Weight functions of the edges in the reference domain.

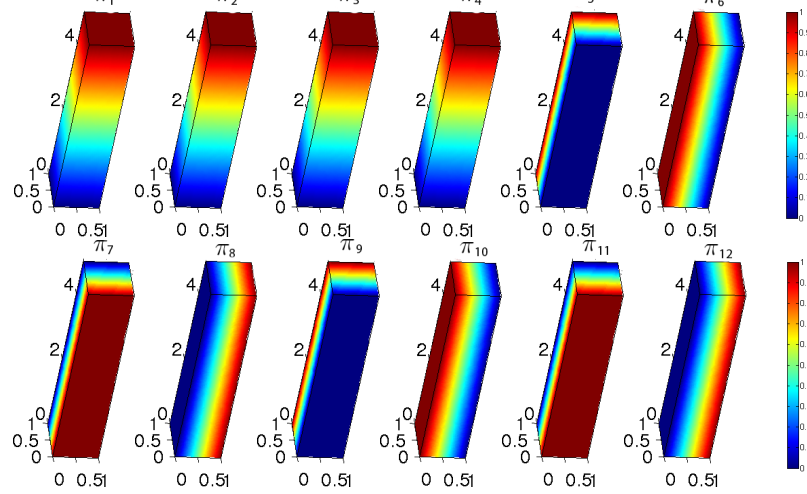


Figure 1.14: Projection functions of the edges in the reference domain.

1.3.3.2 Online stage

We assume that each edge Γ_i is parametrized by a one to one mapping ψ_i from $[0, 1] \times \mathcal{D}$ into Γ_i , so that $\psi_i(0, \mu^i)$ and $\psi_i(1, \mu^i)$ are the extreme vertices of Γ_i .

The transfinite map is then defined as follows:

$$T(\hat{x}, \mu) = \sum_{i=1}^n \left\{ \varphi_i(\hat{x}) \psi_i(\pi_i(\hat{x}), \mu^i) - \varphi_i(\hat{x}) \varphi_{i+1}(\hat{x}) \psi_i(1, \mu^i) \right\}, \quad (1.3.6)$$

where $\boldsymbol{\mu} = (\mu^1, \mu^2, \dots, \mu^n)$, μ^i contains the parameters of Γ_i and n is the number of the domain edges.

1.3.3.3 Numerical examples

We consider the parallelepiped example, the first edge of the domain can be described by the following parametrization of $\boldsymbol{\psi}_1 = (\psi_{1x}, \psi_{1y}, \psi_{1z})$ with $\boldsymbol{\mu}^1 = (\mu_1^1, \mu_2^1)$:

$$x = \psi_{1x}(t, \boldsymbol{\mu}) = \frac{\mu_1^1(\cos(2\pi t) - 1)}{2}; \quad (1.3.7)$$

$$y = \psi_{1y}(t, \boldsymbol{\mu}) = \frac{\mu_1^1(\cos(2\pi t) - 1)}{2}; \quad (1.3.8)$$

$$z = \psi_{1z}(t, \boldsymbol{\mu}) = \mu_2^1 t. \quad (1.3.9)$$

Figure 1.15 shows some domains deformed with the 3D transfinite map, the parameter μ_1^1 represents the dilatation and the contraction of a single edge Γ_1 , that induces the dilatation and the contraction of the whole domain, the parameter μ_2^1 assumes the role of defining the length of the domain.

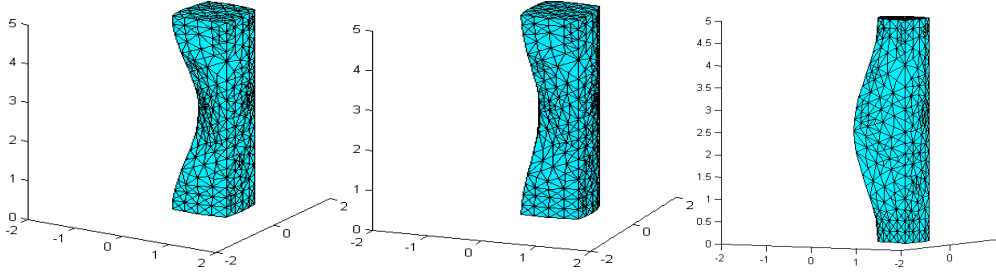


Figure 1.15: Deformed domains with $\mu_1^1 = \{-0.5, -0.2, 0.5\}$ and fixed $\mu_2^1 = 5$ respectively.

We can use now the precomputed weight and projection functions for each new deformation induced by the deformation of all the edges of the domain. Figure 1.16 shows how we can obtain a curved cylinder by deforming all the edges of the reference parallelepiped through proper parametric functions.

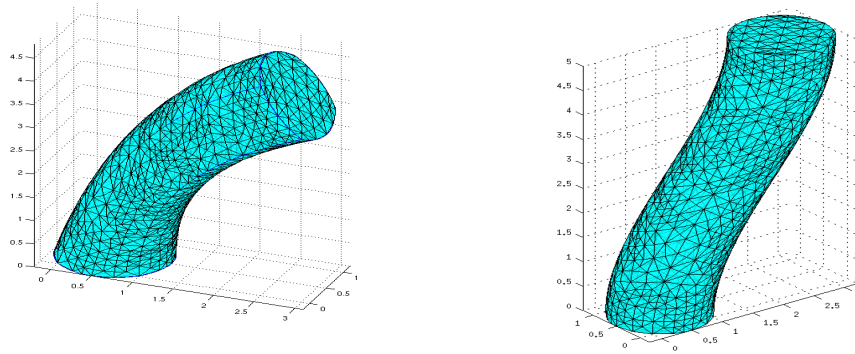


Figure 1.16: Examples of deformed cylinders by 3D TM.

Due to the complexity of three dimensional domains, we deform more complex geometries by com-

binning local transfinite maps. For example in order to deal with a bifurcated domain, we define the reference domain as union of three parallelepiped blocks (Figure 1.17). The weight and projection functions are defined separately on the three blocks following the same description of before.

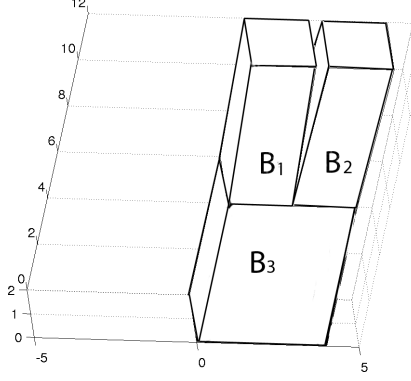


Figure 1.17: An example of reference bifurcation

By maintaining the same order of edges classification of the single block, the block B_1 includes the edges Γ_i with $i = 1, \dots, 12$, the block B_2 includes the edges Γ_i with $i = 13, \dots, 24$ and the block B_3 includes the edges Γ_i with $i = 25, \dots, 36$.

The transfinite map for the bifurcated domain is finally defined as follows:

$$T(\hat{\mathbf{x}}, \boldsymbol{\mu}) = \begin{cases} \sum_{i=1}^{12} \left\{ \varphi_i(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(\pi_i(\hat{\mathbf{x}}), \boldsymbol{\mu}^i) - \varphi_i(\hat{\mathbf{x}}) \varphi_{i+1}(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(1, \boldsymbol{\mu}^i) \right\}, & \hat{\mathbf{x}} \in B_1, \\ \sum_{i=13}^{24} \left\{ \varphi_i(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(\pi_i(\hat{\mathbf{x}}), \boldsymbol{\mu}^i) - \varphi_i(\hat{\mathbf{x}}) \varphi_{i+1}(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(1, \boldsymbol{\mu}^i) \right\}, & \hat{\mathbf{x}} \in B_2, \\ \sum_{i=25}^{36} \left\{ \varphi_i(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(\pi_i(\hat{\mathbf{x}}), \boldsymbol{\mu}^i) - \varphi_i(\hat{\mathbf{x}}) \varphi_{i+1}(\hat{\mathbf{x}}) \boldsymbol{\psi}_i(1, \boldsymbol{\mu}^i) \right\}, & \hat{\mathbf{x}} \in B_3. \end{cases} \quad (1.3.10)$$

Figure 1.18 shows some deformed bifurcations obtained through the transfinite map described in (1.3.10).

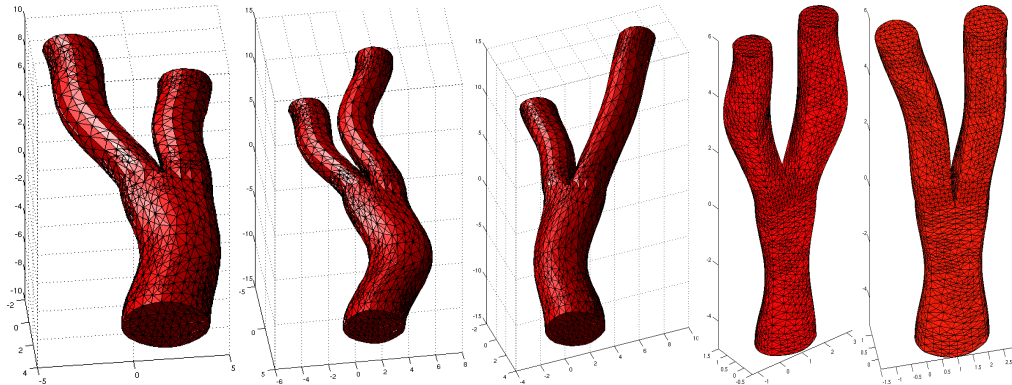


Figure 1.18: Examples of deformed bifurcations.

1.4 Other available geometrical transformations

Other different geometrical parametrization techniques, well suited for RB method, could be adopted depending on the geometrical domain configuration used in the RB context. The geometrical parametrization represents the first reduction step of the problem, it has to model a suitable and accurate deformation of the domain by maintaining a reduced computational complexity. *Affine maps* represent the simplest example (in terms of computational complexity) to transform the problem of interest into a parametrized problem defined on the reference domain. Their definitions are elementary on a simple geometry where the geometrical parameters are shape properties such as lengths, thicknesses, diameters or angles. In this case the affinity requirements for the efficacy of the Offline/Online RB strategy is automatically satisfied. Moreover, the generation of this kind of maps can be made automatic thanks to a suitable domain decomposition paradigm also on curved and more general geometries as, for instance, implemented in the software `rbMIT` [36].

Together with the transfinite maps, other options of non-affine map dealing with more complex geometries are represented by the so-called *free shape representations*. The most popular technique within this group is the Free-Form Deformation (FFD) technique which is based on tensor products of splines [90]. Another technique is the Radial Basis Function (RBF), which is a general paradigm for interpolation of scattered data in high dimensions [11]. In the RB context, FFD has been proposed as a parametrization technique in potential flows [61, 49], in fluid-structure interaction problems [50] and in thermal flows control in [88]. We remark that in order to deal with more general transformations (when the orientations of the inflow and outflow boundaries change with the domain deformation), the Piola transformation [10] process is necessary to correctly express the boundaries forces in the reference system.

1.5 Concluding remarks

In this chapter we have introduced the main ingredients we are going to use in this thesis. We have recalled a very general formulation of the reduced basis method with some emphasis on the greedy algorithm, used to perform an exploration of the parametric space, and the empirical interpolation method, used to recover the affine decomposition of the parametrized operators, holding our problems. Then we have recalled the geometrical parametrization techniques we are going to apply in this work, based on transfinite maps, well-suited to represent shape deformations introduced in the examples (Sections 1.3.1.3 and 1.3.2.2). In this framework we have also proposed some variations with respect to the standard transfinite map in order to improve its versatility and suitability (Section 1.3.2).

2 The reduced basis hybrid method for the parametrized steady Stokes equations

In this chapter we deal with the reduced basis hybrid method (RBHM) that represents the first adopted approach in this thesis for the approximation of parametrized partial differential equations in domains composed by networks, where topological features are recurrent. Some extensions of the reduced basis method are combined here with decomposed domains to solve incompressible fluid flows problems modeled by steady Stokes equations. We start revisiting previous ideas about the reduced basis element method (RBEM) [52, 53, 58] by considering the computational domain as an arbitrary union of non-overlapping subdomains (blocks) which can be obtained as deformations of reference domains (reference blocks). Then, we present the reduced basis hybrid method by maintaining an offline and online computational splitting of the problem. In particular the aim of this method is to guarantee the continuity of velocity and stresses at the interfaces by proper coupling and gluing conditions, by the help of a coarse finite element solution. This aspect was not previously addressed with RBEM, where the recovery of the continuity of the velocity at the interface of the blocks is achieved by Lagrange multipliers, but normal stresses are not guaranteed to match continuously.

The adjective *hybrid* is used here to underline that with respect to classical reduced basis methods (RB and RBEM) in this proposed approach we are using a coarse finite element solution as online correction to ensure continuity and consistency of the normal stress.

The proposed RBHM is built upon the reduced basis element method (RBEM) and it takes advantage from both the reduced basis methods (RB) and the domain decomposition method [75]. We move from the consideration that the blocks composing the computational domain are topologically similar to a few reference shapes. On the latter, representative solutions, corresponding to the same governing partial differential equations, are computed for different values of some parameters of interest, representing, for example, the deformation of the blocks. The desired solution on the given original computational domain is recovered as projection of the previously precomputed solutions and then glued across subdomain interfaces by suitable coupling conditions.

As for RBEM the construction of the map from the reference subdomain to each reference block of the computational domain is carried out by the generalized transfinite map introduced in Chapter 1, Section 1.3. The empirical interpolation procedure proposed in [7] and recalled in Chapter 1, Section 1.2.3.1 has been applied to the geometrical non-affine transformation terms to recover an approximate affine parameter dependence and a sub-sequent offline/online decomposition of the reduced basis procedure. This computational decomposition yields a considerable reduction of the problem complexity. Results computed on some combinations of 2D geometries representing cardiovascular

networks show the advantage of the method in terms of reduced computational costs and the quality of the coupling to guarantee continuity of both stresses, pressure and velocity at subdomain interfaces. As in the RBEM, dealing with viscous flows, the velocity continuity across block interfaces is guaranteed through the introduction of Lagrangian multipliers [53]. The original concept behind the RBHM is that the global solution, found by solving the Stokes problem, ensures not only the velocity continuity but also the continuity of normal stresses across block interfaces. Indeed, the final solution is a projection of local reduced basis with zero normal stress along the interfaces and a finite element solution computed in the whole computational domain with a very coarse grid, in order to guarantee the normal stress continuity at the interfaces. The coarse solution is computed by using an automatic assembling blocks algorithm, which is inexpensive and fast due to the small size of the coarse meshes in the global network. This work is motivated by the fact that in several applications (microfluidics and cardiovascular problems) also the pressure is a quantity of interest and the capability to represent extended fluidic network is of growing importance in several potential field of applications. The contents of this chapter have been based on the work [40].

2.1 The steady Stokes problem

We consider the following steady Stokes problem [73] in a domain $\Omega \subset \mathbb{R}^2$ with mixed boundary conditions on $\Gamma = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_w$:

$$\begin{cases} -\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{on } \Gamma_w, \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \bar{\boldsymbol{\sigma}}_n^{in} & \text{on } \Gamma_{in}, \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \bar{\boldsymbol{\sigma}}_n^{out} & \text{on } \Gamma_{out}, \end{cases} \quad (2.1.1)$$

for a fluid of constant density; \mathbf{u} is the fluid velocity, p the pressure, \mathbf{f} a force field (e.g. gravity), ν a kinematic viscosity and \mathbf{n} the normal outward unit vector to the domain boundary; Γ_{in} and Γ_{out} represent the inflow and outflow, respectively, while Γ_w is a boundary-wall. Here $\bar{\boldsymbol{\sigma}}_n^{in}$ and $\bar{\boldsymbol{\sigma}}_n^{out}$ represent imposed stresses on inflow and outflow, respectively.

On Ω we introduce the velocity space and the pressure space, respectively, as:

$$Y = \{\mathbf{v} \in (H^1(\Omega))^2 : \mathbf{v}|_{\Gamma_w} = \mathbf{0}\}, \quad M = L^2(\Omega).$$

Problem (2.1.1) in weak formulation reads: find $\mathbf{u} \in Y$, $p \in M$:

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = F(\mathbf{v}) & \forall \mathbf{v} \in Y, \\ b(\mathbf{u}, q) = 0 & \forall q \in M, \end{cases} \quad (2.1.2)$$

where

$$a(\mathbf{v}, \mathbf{w}) = \nu \int_{\Omega} \nabla \mathbf{v} : \nabla \mathbf{w} \, d\Omega = \nu \sum_{i,j=1}^2 \int_{\Omega} \frac{\partial v_i}{\partial x_j} \frac{\partial w_i}{\partial x_j} \, d\Omega, \quad (2.1.3)$$

$$b(\mathbf{v}, q) = - \int_{\Omega} q(\nabla \cdot \mathbf{v}) d\Omega = - \sum_{i=1}^2 \int_{\Omega} q \frac{\partial v_i}{\partial x_i} d\Omega, \quad (2.1.4)$$

$$F(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_{in}} \bar{\boldsymbol{\sigma}}_n^{in} \cdot \mathbf{v} d\Gamma + \int_{\Gamma_{out}} \bar{\boldsymbol{\sigma}}_n^{out} \cdot \mathbf{v} d\Gamma. \quad (2.1.5)$$

The continuity of the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$, the coercivity condition on $a(\cdot, \cdot)$

$$a(\mathbf{w}, \mathbf{w}) \geq \alpha \|\mathbf{w}\|_{H^1(\Omega)}^2, \quad \forall \mathbf{w} \in Y, \quad \alpha > 0,$$

and the inf-sup condition on $b(\cdot, \cdot)$

$$\beta = \inf_{q \in M} \sup_{\mathbf{v} \in Y} \frac{b(\mathbf{v}, q)}{\|q\|_{L^2(\Omega)} \|\mathbf{v}\|_{H^1(\Omega)}} > 0, \quad (2.1.6)$$

allow to have the well posedness of problem (4.1.2) and ensure, thanks to the Brezzi theorem, the existence and uniqueness of the solution, see [75, 73].

2.2 The parametrized steady Stokes problem in a local subdomain

In this section we recall the parametrized Stokes problem by focusing on geometrical maps and the treatment of non-affine parametrization as already recalled in Chapter 1, Section 1.4. In particular we introduce local Stokes problems in every subdomain of Ω , this “decomposition” represents a crucial ingredient of the RBHM.

We assume that the domain Ω can be partitioned into a non-overlapping union of R subdomains Ω_r and that each Ω_r is a deformation of a reference domain $\hat{\Omega}_{k(r)}$ through a regular enough, non-affine, map $T_{\boldsymbol{\mu}_r}^{k(r)} : \hat{\Omega}_{k(r)} \rightarrow \Omega_r$ so that:

$$\forall \hat{\mathbf{x}} \in \hat{\Omega}_{k(r)}, \mathbf{x} = T_{\boldsymbol{\mu}_r}^{k(r)}(\hat{\mathbf{x}}), \mathbf{x} \in \Omega_r \subset \Omega, \quad \bar{\Omega} = \bigcup_{r=1}^R \bar{\Omega}_r = \bigcup_{r=1}^R \overline{T_{\boldsymbol{\mu}_r}^{k(r)}(\hat{\Omega}_{k(r)})}.$$

See Figure 2.1 for an example. A possible choice of these maps has been introduced in Chapter 1, Section 1.3. The number of reference domain is $K \leq R$ (otherwise said, the map $k : \mathbb{N} \rightarrow \mathbb{N}$, $r \rightarrow k(r)$ is not necessarily injective). The same reference domain can serve the purpose for different subdomains, thanks to different choices of the parameter $\boldsymbol{\mu}_r \in \mathcal{D} \subset \mathbb{R}^P$ ($P \geq 1$), so that we can characterize different deformations of the same reference domain. In this sense we define a parametric map for each reference domain.

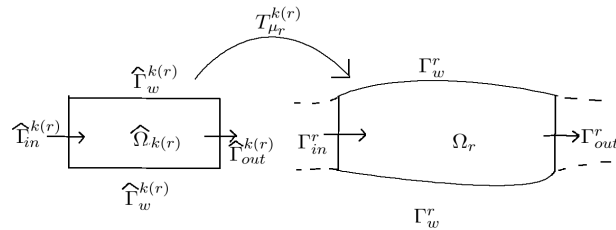


Figure 2.1: Scheme for a geometrical transformation from a reference domain.

For instance, on the example shown in Figure 2.2, we need only two reference domains, $\hat{\Omega}_1$ and $\hat{\Omega}_2$ and therefore only two parametric maps, $T_{\mu_r}^1$ and $T_{\mu_r}^2$. Then for any $\hat{\mathbf{x}} \in \hat{\Omega}_1$ its image can define the four deformed bifurcations in Ω , which are $\Omega_4, \Omega_5, \Omega_6$ and Ω_7 , through different choices of the parameter, respectively, μ_4, μ_5, μ_6 and μ_7 . Any $\mathbf{x} \in \Omega_4$ is given by $\mathbf{x} = T_{\mu_4}^1(\hat{\mathbf{x}}) := T^1(\hat{\mathbf{x}}, \mu_4), \forall \hat{\mathbf{x}} \in \hat{\Omega}_1$, while through $T_{\mu_r}^2(\hat{\mathbf{x}})$ we can map the straight pipe reference domain $\hat{\Omega}_2$ in the deformed pipes $\Omega_1, \Omega_2, \Omega_3$ and Ω_8 , for suitable choices of the parameter μ_r , i.e. $\mu_1, \mu_2, \mu_3, \mu_8$ respectively.

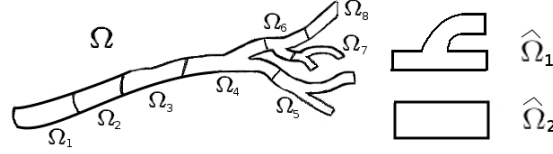


Figure 2.2: A domain composed by R=8 blocks that can be obtained as deformation of K=2 reference domains.

For every Ω_r we denote by Γ_{in}^r its inflow boundary and by Γ_{out}^r its outflow boundary, see Figure 2.1. We call Ω_r an “inflow” element if $\Gamma_{in}^r \subset \Gamma_{in}$, “outflow” element if $\Gamma_{out}^r \subset \Gamma_{out}$, and “central” element when $\Gamma^r \cap \Gamma = \Gamma_w^r$, here $\Gamma_{in}, \Gamma_{out}$ and Γ denote the global inflow, outflow and the boundaries walls of Ω . In the example illustrated in Figure 2.2, Ω_1 is an “inflow” element, Ω_r , with $r = 2, 3, 4, 6$ are “central” elements and $\Omega_8, \Omega_7, \Omega_5$ are “outflow” elements. Note that for more complex configurations involving central subdomains with either inflow or outflow boundaries, more elements should be considered. We denote with σ^r and $\hat{\sigma}^{k(r)}$ the Cauchy stress tensors on Ω_r and $\hat{\Omega}_{k(r)}$, respectively, and with $\sigma_n^r = \sigma^r \mathbf{n}^r$ and $\hat{\sigma}_n^{k(r)} = \hat{\sigma}^{k(r)} \hat{\mathbf{n}}^{k(r)}$ the imposed stresses on inflow and outflow boundaries, where \mathbf{n}^r and $\hat{\mathbf{n}}^{k(r)}$ are the unit outward normal vectors on $\partial\Omega_r$ and $\partial\hat{\Omega}_{k(r)}$, respectively.

We introduce now the formulation of the Stokes problem in every generic deformation of $\hat{\Omega}_{k(r)}$. As shown in [51, 52] and as mentioned in Chapter 1, Section 1.2.3, the parametric maps $T_{\mu_r}^{k(r)}$ and their Jacobians $J_{k(r)}$ allow the definition of the bilinear and linear forms on a deformed domain, $\Omega_r (= T_{\mu_r}^{k(r)}(\hat{\Omega}_{k(r)}))$, through the evaluation of the corresponding forms in the reference domain $\hat{\Omega}_{k(r)}$. For the viscous, pressure (and divergence) terms we consider respectively:

$$a^{k(r)}(\hat{\mathbf{v}}, \hat{\mathbf{w}}, \mu_r) = \nu \int_{\hat{\Omega}_{k(r)}} J_{k(r)}^{-T} \nabla \hat{\mathbf{v}} : J_{k(r)}^{-T} \nabla \hat{\mathbf{w}} |J_{k(r)}| d\hat{\Omega}_{k(r)}, \quad (2.2.1)$$

$$b^{k(r)}(\hat{\mathbf{v}}, \hat{q}, \mu_r) = - \int_{\hat{\Omega}_{k(r)}} \hat{q} \nabla \cdot (J_{k(r)}^{-1} \hat{\mathbf{v}}) |J_{k(r)}| d\hat{\Omega}_{k(r)}, \quad (2.2.2)$$

where $|J_{k(r)}|$ denotes the determinant of $J_{k(r)}$. For the right-hand-side, we consider:

$$F^{k(r)}(\hat{\mathbf{v}}, \mu_r, \hat{\sigma}_n^{k(r)}) = \int_{\hat{\Omega}_{k(r)}} \hat{\mathbf{f}} \cdot \hat{\mathbf{v}} |J_{k(r)}| d\hat{\Omega}_{k(r)} + \int_{\hat{\Gamma}_{in}^{k(r)} \cup \hat{\Gamma}_{out}^{k(r)}} \hat{\sigma}_n^{k(r)} \cdot \hat{\mathbf{v}} |J_{k(r)}| d\hat{\Gamma}_{\hat{\Omega}_{k(r)}}. \quad (2.2.3)$$

We impose no-slip boundary condition on $\hat{\Gamma}_w^{k(r)}$ and Neumann boundary conditions on $\hat{\Gamma}_{in}^{k(r)}$ and $\hat{\Gamma}_{out}^{k(r)}$ with:

- $\hat{\sigma}_n^{k(r)}|_{\hat{\Gamma}_{in}^{k(r)}} = \bar{\sigma}_n^{in}$ and $\hat{\sigma}_n^{k(r)}|_{\hat{\Gamma}_{out}^{k(r)}} = \mathbf{0}$, if $\Omega_r = T_{\mu_r}^{k(r)}(\hat{\Omega}_{k(r)})$ is an “inflow” element;
- $\hat{\sigma}_n^{k(r)}|_{\hat{\Gamma}_{in}^{k(r)} \cup \hat{\Gamma}_{out}^{k(r)}} = \mathbf{0}$, if Ω_r is a “central” element;
- $\hat{\sigma}_n^{k(r)}|_{\hat{\Gamma}_{in}^{k(r)}} = \mathbf{0}$ and $\hat{\sigma}_n^{k(r)}|_{\hat{\Gamma}_{out}^{k(r)}} = \bar{\sigma}_n^{out}$, if Ω_r is an “outflow” element.

2.2. The parametrized steady Stokes problem in a local subdomain

In case of homogeneous boundary data and null force term, a local Stokes problem might feature the null solution.

The Piola transformation can be used to transform the velocity field in order to maintain the orientation of the velocity with respect to the inflow and outflow boundaries, [10, 18]. Alternatively, we can get rid of this transformation by collecting the contributions from the transposed inverse Jacobians and the Jacobian determinant in the tensors $\mathbf{v}^{k(r)}$ and $\boldsymbol{\chi}^{k(r)}$, for viscous and pressure terms, respectively, and use the elements of this tensor as the parameter dependent functions. This mentioned procedure is adopted here and it is convenient as long as the outward normal vectors on the inflow and outflow boundaries are preserved after the geometrical deformations of the domain. The transformation tensors for the bilinear forms $a^{k(r)}(\cdot, \cdot, \boldsymbol{\mu}_r)$ and $b^{k(r)}(\cdot, \cdot, \boldsymbol{\mu}_r)$ are:

$$\mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) = \mathbf{J}_{k(r)}^{-1} \mathbf{J}_{k(r)}^T |\mathbf{J}_{k(r)}| \quad (2.2.4)$$

and

$$\boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) = \mathbf{J}_{k(r)}^{-1} |\mathbf{J}_{k(r)}|. \quad (2.2.5)$$

We can express (2.2.1) and (2.2.2) as

$$a^{k(r)}(\hat{\mathbf{v}}, \hat{\mathbf{w}}, \boldsymbol{\mu}_r) = \nu \int_{\hat{\Omega}_{k(r)}} \mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) \nabla \hat{\mathbf{v}} : \nabla \hat{\mathbf{w}} d\hat{\Omega}_{k(r)}, \quad (2.2.6)$$

$$b^{k(r)}(\hat{\mathbf{v}}, \hat{q}, \boldsymbol{\mu}_r) = \int_{\hat{\Omega}_{k(r)}} \boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) \hat{q} \nabla \cdot \hat{\mathbf{v}} d\hat{\Omega}_{k(r)}. \quad (2.2.7)$$

Since the tensors $\mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})$, $\boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})$ and the determinant $|\mathbf{J}_{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})|$ are non-affine for $k(r) = 1, \dots, K$, we apply the empirical interpolation procedure introduced in Chapter 1, Section 1.2.3.1 to decompose each component of these tensors in parameter dependent contribution and other parts depending only on spatial coordinates [83]. The idea is to approximate the elements of parameter dependent tensors $[\mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij}$, $[\boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij}$ as well as the determinant $|\mathbf{J}_{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})|$, as linear combination of a few “basis” functions, $\tilde{\mathbf{v}}_{ij}^{k(r)m}(\hat{\mathbf{x}}) = [\mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_m)]_{ij}$, $\tilde{\boldsymbol{\chi}}_{ij}^{k(r)n}(\hat{\mathbf{x}}) = [\boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_n)]_{ij}$ and $\tilde{J}_{k(r)s}(\hat{\mathbf{x}}) = |\mathbf{J}_{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}_s)|$, where $\boldsymbol{\mu}_m$, $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_s$ are properly selected within a predefined set of sampling parameters, as detailed presented in Section 1.2.3.1.

We can decouple parameter dependent coefficients from a parameter independent part thanks to the following expansions (no summation on repeated indices here):

$$\begin{aligned} [\mathbf{v}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij} &= \sum_{m=1}^{M_{ij}^{ak(r)}} \Theta_{ij}^{k(r)m}(\boldsymbol{\mu}) \tilde{\mathbf{v}}_{ij}^{k(r)m}(\hat{\mathbf{x}}) + \boldsymbol{\epsilon}_{ij}^{ak(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}), \\ [\boldsymbol{\chi}^{k(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij} &= \sum_{n=1}^{M_{ij}^{bk(r)}} \Phi_{ij}^{k(r)n}(\boldsymbol{\mu}) \tilde{\boldsymbol{\chi}}_{ij}^{k(r)n}(\hat{\mathbf{x}}) + \boldsymbol{\epsilon}_{ij}^{bk(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}), \\ |\mathbf{J}_k(\hat{\mathbf{x}}, \boldsymbol{\mu})| &= \sum_{s=1}^{M^{sk(r)}} \Psi^{k(r)s}(\boldsymbol{\mu}) \tilde{J}_{k(r)s}(\hat{\mathbf{x}}) + \boldsymbol{\epsilon}^{sk(r)}(\hat{\mathbf{x}}, \boldsymbol{\mu}). \end{aligned}$$

In the previous expressions $\Theta_{ij}^{k(r)m}$, $\Phi_{ij}^{k(r)n}$, $\Psi^{k(r)s} : \mathcal{D} \rightarrow \mathbb{R}$ are weighing quantities depending on the parameters; $\tilde{\mathbf{v}}_{ij}^{k(r)m}$, $\tilde{\boldsymbol{\chi}}_{ij}^{k(r)n}$, $\tilde{J}_{k(r)s}$ are interpolation functions used as basis, M refers to the number of interpolation functions we use for each form and it is related with the maximum interpolation error

ϵ_{tol}^{EIM} , such that:

$$\begin{aligned} \|\epsilon_{ij}^{ak(r)}(\cdot, \cdot)\|_{\infty} &\leq \epsilon_{tol}^{EIM}, \quad i, j = 1, 2, \\ \|\epsilon_{ij}^{bk(r)}(\cdot, \cdot)\|_{\infty} &\leq \epsilon_{tol}^{EIM}, \quad i, j = 1, 2, \\ \|\epsilon^{sk(r)}(\cdot, \cdot)\|_{\infty} &\leq \epsilon_{tol}^{EIM}. \end{aligned}$$

By applying this affine decomposition to the terms (2.2.3), (2.2.6) and (2.2.7) we define the following linear and bilinear forms as their respective approximations:

$$\mathcal{F}^{k(r)}(\hat{\mathbf{v}}, \boldsymbol{\mu}, \hat{\boldsymbol{\sigma}}_n^{k(r)}) = \sum_{s=1}^{M^{sk(r)}} \Psi^{k(r)s}(\boldsymbol{\mu}) \mathcal{F}^{k(r)s}(\hat{\mathbf{v}}, \hat{\boldsymbol{\sigma}}_n^{k(r)}), \quad (2.2.8)$$

$$\mathcal{A}^{k(r)}(\hat{\mathbf{v}}, \hat{\mathbf{w}}, \boldsymbol{\mu}) = \nu \sum_{i=1}^2 \sum_{j=1}^2 \sum_{m=1}^{M_{ij}^{ak(r)}} \Theta_{ij}^{k(r)m}(\boldsymbol{\mu}) \mathcal{A}_{ij}^{k(r)m}(\hat{\mathbf{v}}, \hat{\mathbf{w}}), \quad (2.2.9)$$

$$\mathcal{B}^{k(r)}(\hat{\mathbf{v}}, \hat{q}, \boldsymbol{\mu}) = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{n=1}^{M_{ij}^{bk(r)}} \Phi_{ij}^{k(r)n}(\boldsymbol{\mu}) \mathcal{B}_{ij}^{k(r)n}(\hat{\mathbf{v}}, \hat{q}), \quad (2.2.10)$$

where

$$\begin{aligned} \mathcal{F}^{k(r)m}(\hat{\mathbf{v}}, \hat{\boldsymbol{\sigma}}_n^{k(r)}) &= \int_{\hat{\Omega}_{k(r)}} \tilde{J}_{k(r)m}(\hat{\mathbf{x}}) \hat{\mathbf{f}} \cdot \hat{\mathbf{v}} d\hat{\Omega}_{k(r)} + \int_{\hat{\Gamma}_{in}^{k(r)} \cup \hat{\Gamma}_{out}^{k(r)}} \tilde{J}_{km}(\hat{\mathbf{x}}) \hat{\boldsymbol{\sigma}}_n^{k(r)} \hat{\mathbf{v}} d\hat{\Gamma}_{\hat{\Omega}_{k(r)}}, \\ \mathcal{A}_{ij}^{k(r)m}(\hat{\mathbf{v}}, \hat{\mathbf{w}}) &= \int_{\hat{\Omega}_{k(r)}} \hat{\mathbf{v}}_{ij}^{k(r)m}(\hat{\mathbf{x}}) \frac{\partial \hat{\mathbf{v}}}{\partial \hat{x}_i} \frac{\partial \hat{\mathbf{w}}}{\partial \hat{x}_j} d\hat{\Omega}_{k(r)}, \\ \mathcal{B}_{ij}^{k(r)m}(\hat{\mathbf{v}}, \hat{q}) &= - \int_{\hat{\Omega}_{k(r)}} \tilde{\chi}_{ij}^{k(r)m}(\hat{\mathbf{x}}) \hat{q} \frac{\partial \hat{v}_i}{\partial \hat{x}_j} d\hat{\Omega}_{k(r)}. \end{aligned}$$

As pointed out in Chapter 1, this recovered affine decomposition is crucial in the RB method to split all the heavy computation involving high resolution (concerning discretization) in an offline stage and, then during an online stage, to solve efficiently the problem for each new choice of the parameters and for each subdomain that we want to consider in the network configuration.

For a new $\boldsymbol{\mu}_r$ and for the proper reference domain $\hat{\Omega}_{k(r)}$ the Stokes problem can be rewritten as: find $(\hat{\mathbf{u}}(\boldsymbol{\mu}_r), \hat{p}(\boldsymbol{\mu}_r)) \in Y^{k(r)} \times M^{k(r)}$ such that

$$\begin{cases} \mathcal{A}^{k(r)}(\hat{\mathbf{u}}(\boldsymbol{\mu}_r), \hat{\mathbf{w}}, \boldsymbol{\mu}_r) + \mathcal{B}^{k(r)}(\hat{\mathbf{w}}, \hat{p}(\boldsymbol{\mu}_r), \boldsymbol{\mu}_r) = \mathcal{F}^{k(r)}(\hat{\mathbf{w}}, \boldsymbol{\mu}_r, \hat{\boldsymbol{\sigma}}_n^{k(r)}) & \forall \hat{\mathbf{w}} \in Y^{k(r)}, \\ \mathcal{B}^{k(r)}(\hat{\mathbf{u}}(\boldsymbol{\mu}_r), \hat{q}, \boldsymbol{\mu}_r) = 0 & \forall \hat{q} \in M^{k(r)}, \end{cases} \quad (2.2.11)$$

where

$$\begin{aligned} Y^{k(r)} &= \left\{ \hat{\mathbf{v}} \in (H^1(\hat{\Omega}_{k(r)}))^2 : \hat{\mathbf{v}}|_{\hat{\Gamma}_w^{k(r)}} = 0 \right\}, \quad \hat{\Gamma}_w^{k(r)} = T_{\boldsymbol{\mu}_r}^{k(r)-1}(\Gamma_w \cap \partial\Omega_r), \\ M^{k(r)} &= L^2(\hat{\Omega}_{k(r)}). \end{aligned}$$

2.3. The reduced basis formulation for the parametrized steady Stokes problem

The solution of this problem satisfies the inf-sup condition [75], expressed by the following inequality:

$$\exists \beta_0^{k(r)} > 0 : \beta^{k(r)}(\boldsymbol{\mu}_r) := \inf_{\hat{\mathbf{q}} \in M^{k(r)}} \sup_{\hat{\mathbf{w}} \in Y^{k(r)}} \frac{\mathcal{B}^{k(r)}(\hat{\mathbf{w}}, \hat{\mathbf{q}}, \boldsymbol{\mu}_r)}{\|\hat{\mathbf{w}}\|_{Y^{k(r)}} \|\hat{\mathbf{q}}\|_{M^{k(r)}}} \geq \beta_0^{k(r)},$$

$$\forall \boldsymbol{\mu}_r \in \mathcal{D}, \forall k(r) = 1, \dots, K. \quad (2.2.12)$$

In particular, starting from some reference domains, problem (2.2.11) represents a (local) well-posed Stokes problem in each deformed block of the computational domain, which accounts for imposing proper Neumann boundary conditions that are dictated by the relative “position” of the deformed subdomain (inflow, outflow or central).

Once we have the local Stokes formulations, we can find the Stokes solution in the global domain Ω through suitable assumptions that will be introduced in the next sections together with the reduced basis formulation.

2.3 The reduced basis formulation for the parametrized steady Stokes problem

After recalling the classical RB formulation for a single domain case in Chapter 1, Section 1.2.5, here we give the brief reduced basis formulation for the Stokes problem still for a single domain case and in the following Sections we extend it to a multi-domain case. Since we are considering only one reference domain $\hat{\Omega}$, we can omit the k index and, from now we omit the “hat” to further simplify the notations, however we warn the reader that we are always referring to the reference subdomain. For given Neumann boundary condition, we look for a reduced basis formulation of problem (2.2.11). With this aim we build a set of parameter samples $S_N^\mu = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ and correspondingly a set of pairs $(\mathbf{u}_h(\boldsymbol{\mu}^i), p_h(\boldsymbol{\mu}^i))$ which are approximate solutions of the Stokes problem using Galerkin Finite Element method¹ on an accurate fine mesh \mathcal{T}_h , where as customary h indicates the maximum edge size of \mathcal{T}_h . The choice of the parameter set S_N^μ has been done using a greedy algorithm, like those proposed in [82, 83].

Following [82, 83, 84], an approximation of problem (2.2.11) is computed as a Galerkin projection onto the following low dimensional subspaces Z_N and M_N for velocity and pressure, respectively:

$$Z_N = \text{span}\{\mathbf{u}_h(\boldsymbol{\mu}^i), i = 1, \dots, N\}, \quad (2.3.1)$$

$$M_N = \text{span}\{p_h(\boldsymbol{\mu}^i), i = 1, \dots, N\}. \quad (2.3.2)$$

In order to guarantee the approximation stability of the reduced basis method for Stokes problem, we fulfill the inf-sup condition (2.2.12) by enriching the velocity basis as follows. For every pressure solution $p_h(\boldsymbol{\mu}^i)$ spanning M_N , we define:

$$\mathbf{v}_h(\boldsymbol{\mu}^i) = \arg \sup_{\mathbf{w} \in Y} \frac{\mathcal{B}(\mathbf{w}, p_h(\boldsymbol{\mu}^i), \boldsymbol{\mu}^i)}{\|\mathbf{w}\|_Y}, \quad (2.3.3)$$

and then

$$X_N = \text{span}\{\mathbf{v}_h(\boldsymbol{\mu}^i), i = 1, \dots, N\}.$$

¹ \mathbb{P}_2 elements for velocity, \mathbb{P}_1 for pressure, respectively [76].

Finally, the enriched velocity space is defined by:

$$Y_N = Z_N \oplus X_N. \quad (2.3.4)$$

By setting

$$\beta_N(\boldsymbol{\mu}) := \inf_{q \in M_N} \sup_{\mathbf{w} \in Y_N} \frac{\mathcal{B}(\mathbf{w}, q, \boldsymbol{\mu})}{\|\mathbf{w}\|_Y \|q\|_M} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \quad (2.3.5)$$

as shown in [84] and more recently in [25, 87], the following condition, binding (2.2.12) and (2.3.5), is fulfilled:

$$\beta_N(\boldsymbol{\mu}) \geq \beta(\boldsymbol{\mu}) \geq \beta_0 > 0 \quad \forall \boldsymbol{\mu} \in \mathcal{D},$$

where $\beta(\boldsymbol{\mu})$ is the inf-sup constant appearing in (2.2.12) related to the Galerkin Finite Element method.

The reduced basis approximation of problem (2.2.11) reads: find $(\mathbf{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in (Y_N \times M_N)$ such that

$$\begin{cases} \mathcal{A}(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{w}, \boldsymbol{\mu}) + \mathcal{B}(\mathbf{w}, p_N(\boldsymbol{\mu}), \boldsymbol{\mu}) = \mathcal{F}(\mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}_n) & \forall \mathbf{w} \in Y_N \\ \mathcal{B}(\mathbf{u}_N(\boldsymbol{\mu}), q, \boldsymbol{\mu}) = 0 & \forall q \in M_N. \end{cases} \quad (2.3.6)$$

Note that this represents the generic RB formulation of the Stokes problem (2.2.11), with $\boldsymbol{\sigma}_n$ representing the proper imposed stress on the inflow and on the outflow. By writing:

$$\begin{aligned} \mathbf{u}_N(\boldsymbol{\mu}) &= \sum_{i=1}^N u_{Ni}(\boldsymbol{\mu}) \mathbf{u}_h(\boldsymbol{\mu}^i) + \sum_{i=N+1}^{2N} u_{Ni}(\boldsymbol{\mu}) \mathbf{v}_h(\boldsymbol{\mu}^{i-N}) \\ p_N(\boldsymbol{\mu}) &= \sum_{i=1}^N p_{Ni}(\boldsymbol{\mu}) p_h(\boldsymbol{\mu}^i), \end{aligned}$$

we find that the coefficients u_{Ni} and p_{Ni} are obtained by solving the following linear system:

$$\begin{cases} \sum_{i=1}^{2N} A_{ji}^{\boldsymbol{\mu}} u_{Ni}(\boldsymbol{\mu}) + \sum_{i=1}^N B_{ji}^{\boldsymbol{\mu}} p_{Ni}(\boldsymbol{\mu}) = F_j^{\boldsymbol{\mu}} & 1 \leq j \leq 2N \\ \sum_{j=1}^{2N} B_{ij}^{\boldsymbol{\mu}} u_{Ni}(\boldsymbol{\mu}) = 0 & 1 \leq i \leq N \end{cases} \quad (2.3.7)$$

where

$$\begin{aligned} A_{ij}^{\boldsymbol{\mu}} &= \mathcal{A}(\mathbf{w}_j, \mathbf{w}_i, \boldsymbol{\mu}), \quad B_{il}^{\boldsymbol{\mu}} = \mathcal{B}(\mathbf{w}_l, p_i, \boldsymbol{\mu}), \quad F_j^{\boldsymbol{\mu}} = \mathcal{F}(\mathbf{w}_j, \boldsymbol{\mu}, \boldsymbol{\sigma}_n), \\ 1 \leq i, j \leq 2N, \quad 1 \leq l \leq N, \quad \mathbf{w}_i, \mathbf{w}_j &\in Y_N, \quad p_l \in M_N. \end{aligned}$$

Since the bilinear forms are now affinely parametrized (after the empirical interpolation treatment), in an offline expensive stage we can compute the parameter independent parts of the matrices $A^{\boldsymbol{\mu}}$, $B^{\boldsymbol{\mu}}$ and the vector $F^{\boldsymbol{\mu}}$ (that include FE matrices, basis functions and pre and post multiplication procedures of the FE matrices for the basis functions computed). Then in an online stage, for each new parameter value the parametric coefficients of the system can be quickly evaluated. Finally, a small linear system can be solved efficiently during the online stage many times to find the coefficients u_{Ni} and p_{Ni} that will give the final reduced basis solution for each new value of $\boldsymbol{\mu}$ [87].

2.4 The RBHM for parametrized steady Stokes equations

What recalled in Sections 2.2 and 2.3 is a framework common to both RB method and RBEM, in this section we formulate the proposed variant of these methods: the reduced basis *hybrid* method (RBHM) for computational domains with rigid boundaries. Before providing a general description of the method and its computationally efficient realization, we illustrate the basic concept on a simplified case.

2.4.1 Two domains with single internal interface

We consider a domain Ω parametrized through $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and its subdomain decomposition in Ω_1 , parametrized through $\boldsymbol{\mu}_1$, and Ω_2 parametrized through $\boldsymbol{\mu}_2$. Γ_{12} is the common interface.

We want to solve the following Stokes problem on Ω : find $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))$ such that

$$\begin{cases} \mathcal{A}(\mathbf{u}(\boldsymbol{\mu}), \mathbf{w}, \boldsymbol{\mu}) + \mathcal{B}(\mathbf{w}, p(\boldsymbol{\mu}), \boldsymbol{\mu}) = \mathcal{F}(\mathbf{w}, \boldsymbol{\mu}) & \forall \mathbf{w} \in Y, \\ \mathcal{B}(\mathbf{u}(\boldsymbol{\mu}), q, \boldsymbol{\mu}) = 0 & \forall q \in M, \end{cases} \quad (2.4.1)$$

where the source term $\mathcal{F}(\mathbf{w}, \boldsymbol{\mu})$ accounts for the normal stresses $\bar{\sigma}_n^{in}$ on Γ_{in} and $\bar{\sigma}_n^{out}$ on Γ_{out} , i.e. on the inflow and the outflow, respectively. Thus $\mathcal{F}(\mathbf{w}, \boldsymbol{\mu}) = \mathcal{F}(\mathbf{w}, \boldsymbol{\mu}, \bar{\sigma}_n)$. We use for (2.4.1) the compact notation $St_{\Omega}(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$.

We consider now the restriction of (2.4.1) to Ω_1 and Ω_2 by maintaining the boundary condition on the part of $\partial\Omega$ included in $\partial\Omega_1$ and in $\partial\Omega_2$. Besides, we impose zero normal stress on the interface boundary $\Gamma_{12} = \partial\Omega_1 \cap \partial\Omega_2$ for either problem. We denote the two corresponding independent Stokes problems with $St_{\Omega_1}(\mathbf{u}^1(\boldsymbol{\mu}_1), p^1(\boldsymbol{\mu}_1), \boldsymbol{\mu}_1) = 0$ and $St_{\Omega_2}(\mathbf{u}^2(\boldsymbol{\mu}_2), p^2(\boldsymbol{\mu}_2), \boldsymbol{\mu}_2) = 0$.

For either problem, according to a standard RB procedure, we select a set of representative sampling parameters, say $\boldsymbol{\mu}_1^j, j = 1, \dots, N_1$ and $\boldsymbol{\mu}_2^j, j = 1, \dots, N_2$, respectively. Correspondingly, we compute (offline) the following set of basis functions on the two respective fine accurate meshes $(\mathcal{T}_h)_1$ and $(\mathcal{T}_h)_2$ on Ω_1 and Ω_2 , respectively:

$$\{\mathbf{u}_h^1(\boldsymbol{\mu}_1^j), \mathbf{v}_h^1(\boldsymbol{\mu}_1^j), p_h^1(\boldsymbol{\mu}_1^j), j = 1, \dots, N_1\} \text{ on } \Omega_1,$$

$$\{\mathbf{u}_h^2(\boldsymbol{\mu}_2^j), \mathbf{v}_h^2(\boldsymbol{\mu}_2^j), p_h^2(\boldsymbol{\mu}_2^j), j = 1, \dots, N_2\} \text{ on } \Omega_2,$$

where $\mathbf{u}_h^1(\boldsymbol{\mu}_1^j), p_h^1(\boldsymbol{\mu}_1^j)$ is the FE solution of $St_{\Omega_1}(\mathbf{u}_h^1(\boldsymbol{\mu}_1^j), p_h^1(\boldsymbol{\mu}_1^j), \boldsymbol{\mu}_1^j) = 0$, $\mathbf{u}_h^2(\boldsymbol{\mu}_2^j), p_h^2(\boldsymbol{\mu}_2^j)$ the FE solution of $St_{\Omega_2}(\mathbf{u}_h^2(\boldsymbol{\mu}_2^j), p_h^2(\boldsymbol{\mu}_2^j), \boldsymbol{\mu}_2^j) = 0$, and $\mathbf{v}_h^1(\boldsymbol{\mu}_1^j), \mathbf{v}_h^2(\boldsymbol{\mu}_2^j)$ the associated velocity supremizers. Note that the normal stresses vanish (by construction) on Γ_{12} (whence they match continuously, that is their jump is zero) whereas the velocities are (in principle) discontinuous across Γ_{12} .

We move now *online*, that is we consider a specific given value of the parameter $\boldsymbol{\mu}$. Correspondingly, we compute an approximation of the whole global problem (2.4.1), say $St_{\Omega}(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$, on a (very) coarse grid \mathcal{T}_H on Ω . Let us indicate the corresponding FE coarse solution as $\mathbf{u}_H(\boldsymbol{\mu}), p_H(\boldsymbol{\mu})$, and by $\mathbf{v}_H(\boldsymbol{\mu})$ the corresponding velocity supremizer.

Clearly, this coarse solution fulfills all the desired continuities at the interface Γ_{12} that is both $\mathbf{u}_H(\boldsymbol{\mu})$ and $\mathbf{v}_H(\boldsymbol{\mu})$, as well as their associated normal stresses, are continuous.

At this stage, our RBHM solution is sought (online) by proceeding as follows.

We look for a suitable linear combination of the previously computed solutions, that is:

$$\mathbf{u}_{N|\Omega_1}(\boldsymbol{\mu}) = \sum_{j=1}^{N_1} (\alpha_j^1 \mathbf{u}_h^1(\boldsymbol{\mu}_1^j) + \beta_j^1 \mathbf{v}_h^1(\boldsymbol{\mu}_1^j)) + \eta \mathbf{u}_H^1(\boldsymbol{\mu}) + \delta \mathbf{v}_H^1(\boldsymbol{\mu}),$$

$$p_{N|\Omega_1}(\boldsymbol{\mu}) = \sum_{j=1}^{N_1} \gamma_j^1 p_h^1(\boldsymbol{\mu}_1^j) + \epsilon p_H^1(\boldsymbol{\mu}),$$

$$\mathbf{u}_{N|\Omega_2}(\boldsymbol{\mu}) = \sum_{j=1}^{N_2} (\alpha_j^2 \mathbf{u}_h^2(\boldsymbol{\mu}_2^j) + \beta_j^2 \mathbf{v}_h^2(\boldsymbol{\mu}_2^j)) + \eta \mathbf{u}_H^2(\boldsymbol{\mu}) + \delta \mathbf{v}_H^2(\boldsymbol{\mu}),$$

$$p_{N|\Omega_2}(\boldsymbol{\mu}) = \sum_{j=1}^{N_2} \gamma_j^2 p_h^2(\boldsymbol{\mu}_2^j) + \epsilon p_H^2(\boldsymbol{\mu}).$$

where

$$\mathbf{u}_H^i(\boldsymbol{\mu}) = \mathbf{u}_H(\boldsymbol{\mu})|_{\Omega_i}, \mathbf{v}_H^i(\boldsymbol{\mu}) = \mathbf{v}_H(\boldsymbol{\mu})|_{\Omega_i}, p_H^i(\boldsymbol{\mu}) = p_H(\boldsymbol{\mu})|_{\Omega_i}, i = 1, 2.$$

The coefficients $\{\alpha_j^1, \beta_j^1, \gamma_j^1, \alpha_k^2, \beta_k^2, \gamma_k^2, j = 1, \dots, N_1, k = 1, \dots, N_2\}, \eta, \delta, \epsilon$ are of course to be determined. Note that the stresses associated with these functions are continuous across Γ_{12} , whereas the velocities are not. The continuity of velocities is imposed by solving the original Stokes problem on the whole domain, by using the Galerkin method on the spaces

$$Y_{N_1} = \text{span}\{\mathbf{u}_h^1(\boldsymbol{\mu}_1^j), \mathbf{v}_h^1(\boldsymbol{\mu}_1^j), \mathbf{u}_H^1(\boldsymbol{\mu}), \mathbf{v}_H^1(\boldsymbol{\mu}), j = 1, \dots, N_1\},$$

$$Y_{N_2} = \text{span}\{\mathbf{u}_h^2(\boldsymbol{\mu}_2^j), \mathbf{v}_h^2(\boldsymbol{\mu}_2^j), \mathbf{u}_H^2(\boldsymbol{\mu}), \mathbf{v}_H^2(\boldsymbol{\mu}), j = 1, \dots, N_2\},$$

$$M_{N_1} = \text{span}\{p_h^1(\boldsymbol{\mu}_1^j), p_H^1(\boldsymbol{\mu}), j = 1, \dots, N_1\},$$

$$M_{N_2} = \text{span}\{p_h^2(\boldsymbol{\mu}_2^j), p_H^2(\boldsymbol{\mu}), j = 1, \dots, N_2\}$$

and using Lagrange multipliers.

More precisely we set

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\psi}) = \int_{\Gamma_{12}} \llbracket \mathbf{v} \rrbracket \cdot \boldsymbol{\psi} ds, \forall \mathbf{v} \in Y_{N_1} \times Y_{N_2}, \forall \boldsymbol{\psi} \in W_{12}, \quad (2.4.2)$$

where $\llbracket \mathbf{v} \rrbracket = \mathbf{v}^1 - \mathbf{v}^2$ denotes the jump of \mathbf{v} across Γ_{12} and W_{12} is a low order polynomial space defined on Γ_{12} .

Then we consider the problem: find $\mathbf{u}_N(\boldsymbol{\mu}) \in Y_{N_1} \times Y_{N_2}, p_N(\boldsymbol{\mu}) \in M_{N_1} \times M_{N_2}, \boldsymbol{\lambda}_N \in W_{12}$ such that

$$\begin{cases} \mathcal{A}(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{w}, \boldsymbol{\mu}) + \mathcal{B}(\mathbf{w}, p_N(\boldsymbol{\mu}), \boldsymbol{\mu}) + \mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}_N) = \mathcal{F}(\mathbf{w}, \boldsymbol{\mu}) & \forall \mathbf{w} \in Y_{N_1} \times Y_{N_2}, \\ \mathcal{B}(\mathbf{u}_N(\boldsymbol{\mu}), q, \boldsymbol{\mu}) = 0 & \forall q \in M_{N_1} \times M_{N_2}, \\ \mathcal{L}(\mathbf{u}_N(\boldsymbol{\mu}), \boldsymbol{\psi}) = 0 & \forall \boldsymbol{\psi} \in W_{12}. \end{cases} \quad (2.4.3)$$

where $\boldsymbol{\lambda}_N$ represents the normal stresses on the internal interface.

In general, if the velocities are approximated with piecewise continuous polynomials of order k and pressures with polynomials of order $k - 1$, then the normal stresses are polynomials of degree $k - 1$. For this reason, the space of the Lagrange multipliers has been chosen as $W_{12} = [\mathbb{P}^{k-1}(\Gamma_{12})]^2$.

The problem (2.4.3) can be written in compact form as:

$$\begin{pmatrix} S & L \\ L^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \quad (2.4.4)$$

where:

$$S = \begin{pmatrix} A^1 & 0 & B^1 & 0 \\ 0 & A^2 & 0 & B^2 \\ (B^1)^T & 0 & 0 & 0 \\ 0 & (B^2)^T & 0 & 0 \end{pmatrix}, L = \begin{pmatrix} L_{12}^1 \\ -L_{12}^2 \end{pmatrix}, F = \begin{pmatrix} F^1 \\ F^2 \end{pmatrix}, \mathbf{U} = \begin{pmatrix} \mathbf{u}_N^1 \\ \eta \\ \delta \\ \mathbf{u}_N^2 \\ \mathbf{p}_N^1 \\ \epsilon \\ \mathbf{p}_N^2 \end{pmatrix}, \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{Q_{12}} \end{pmatrix},$$

$$A_{ij}^k = \mathcal{A}(\mathbf{w}_i^k, \mathbf{w}_j^k, \boldsymbol{\mu}_k), \quad B_{il}^k = \mathcal{B}(\mathbf{w}_i^k, p_l^k, \boldsymbol{\mu}_k), \quad F_j^k = \mathcal{F}(\mathbf{w}_j^k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_n^k), \quad (2.4.5)$$

$$(L_{12}^k)_{iq} = \int_{\Gamma_{12}} \mathbf{w}_i^k \cdot \boldsymbol{\psi}_q ds, \quad \mathbf{w}_i^k, \mathbf{w}_j^k \in Y_{N_k}, \quad p_l^k \in M_{N_k}, \quad \boldsymbol{\psi}_q \in W_{12},$$

$$\mathbf{u}_N^1 = \begin{pmatrix} \alpha_1^1 \\ \vdots \\ \alpha_{N_1}^1 \\ \beta_1^1 \\ \vdots \\ \beta_{N_1}^1 \end{pmatrix}, \mathbf{u}_N^2 = \begin{pmatrix} \alpha_1^2 \\ \vdots \\ \alpha_{N_2}^2 \\ \beta_1^2 \\ \vdots \\ \beta_{N_2}^2 \end{pmatrix}, \mathbf{p}_N^1 = \begin{pmatrix} \gamma_1^1 \\ \vdots \\ \gamma_{N_1}^1 \end{pmatrix}, \mathbf{p}_N^2 = \begin{pmatrix} \gamma_1^2 \\ \vdots \\ \gamma_{N_2}^2 \end{pmatrix},$$

$$1 \leq i, j \leq 2N_k + 2, \quad 1 \leq l \leq N_k + 1, \quad k = 1, 2, \quad 1 \leq q \leq 2Q_{12},$$

here Q_{12} is the number of nodes considered on the interface Γ_{12} .

2.4.2 Several subdomains with many internal interfaces

In the most general case, we now consider a computational domain Ω parametrized through $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_R)$ and decomposed into a non-overlapping union of R subdomains Ω_r , each one parametrized through the corresponding $\boldsymbol{\mu}_r$, see for example the domain shown in Figure 2.2. We make the assumption that each inflow boundary is included in one subdomain, each outflow boundary in another one, so that any other subdomain Ω_r has two internal interfaces. Our methodology can however be extended to more general subdomain partitions.

We still want to solve problem (2.4.1) in Ω . The difference, with respect to what we have done in Section 2.4.1, is that now Ω is partitioned into R subdomains Ω_r and each Ω_r is a deformation of a reference domain $\hat{\Omega}_{k(r)}$, for a suitable $k(r) \in \{1, \dots, K\}$, K being the number of reference shapes.

We focus now on computational strategy: as in the previous case of a portion with only two subdomains, see Section 2.4.1, we split it in two main stages. The *offline stage* involves the reference blocks and it consists in the computation of independent reduced basis spaces and matrices (by using an accurate fine mesh \mathcal{T}_h). During the *online stage*, we use the results of the previous stage and we consider some proper computations and additional gluing conditions through the internal interfaces, in order to find a continuous global solution in the domain Ω . With the term *hybrid* we want also underline that the online stage is not only depending on the dimensions N_i of the reduced spaces, but it also includes some FE computation on a very coarse mesh \mathcal{T}_H in Ω (however it does not involve any FE computations on \mathcal{T}_h , but only an interpolation procedure between the coarse and the fine discretization). The final goal is to have a fine solution at the cost of a coarse one, after proper pre-calculation (performed offline).

More precisely the following steps are performed in the offline stage.

- For every reference domain $\hat{\Omega}_{k(r)}$ we define the index $pos(r) \in \{c, in, out\}$ and three reference “elements”: $\hat{\Omega}_{k(r)}^{in}$ (“inflow” element), $\hat{\Omega}_{k(r)}^c$ (“central” element), $\hat{\Omega}_{k(r)}^{out}$ (“outflow” element). In every “element”, we formulate a Stokes problem in the form (2.2.11) : find $(\mathbf{u}(\boldsymbol{\mu}_r), p(\boldsymbol{\mu}_r)) \in Y^{k(r)} \times M^{k(r)}$ such that

$$\begin{cases} \mathcal{A}^{k(r)}(\mathbf{u}(\boldsymbol{\mu}_r), \mathbf{w}, \boldsymbol{\mu}_r) + \mathcal{B}^{k(r)}(\mathbf{w}, p(\boldsymbol{\mu}_r), \boldsymbol{\mu}_r) = \mathcal{F}^{k(r)}(\mathbf{w}, \boldsymbol{\mu}_r, \boldsymbol{\sigma}_n^{k(r)}) & \forall \mathbf{w} \in Y^{k(r)}, \\ \mathcal{B}^{k(r)}(\mathbf{u}(\boldsymbol{\mu}_r), q, \boldsymbol{\mu}_r) = 0 & \forall q \in M^{k(r)}. \end{cases} \quad (2.4.6)$$

These local Stokes problems are defined in each $\hat{\Omega}_{k(r)}^{pos(k)}$, $r = 1, \dots, R$ and the corresponding boundary conditions, as well as the spaces $Y^{k(r)}$ and $M^{k(r)}$, have been set as proposed in Section 2.2.

- We solve problems (2.4.6) for several values of the (sampling) parameter $\boldsymbol{\mu}_r$ and we define the corresponding reduced spaces. More precisely, since $k(r) \in \{1, \dots, K\}$ and $pos(r) \in \{c, in, out\}$, we define $6K$ spaces:

$$\begin{aligned} Y_{k(r)}^{pos(r)} &= span \left\{ \mathbf{u}_h^{k(r)}(\boldsymbol{\mu}_r^i), \mathbf{v}_h^{k(r)}(\boldsymbol{\mu}_r^i), i = 1, \dots, N_{k(r)}^{pos(r)} \right\}, \\ M_{k(r)}^{pos(r)} &= span \left\{ p_h^{k(r)}(\boldsymbol{\mu}_r^i), i = 1, \dots, N_{k(r)}^{pos(r)} \right\}, \end{aligned} \quad (2.4.7)$$

where $\boldsymbol{\mu}_r^i$ are the samples chosen by the greedy algorithm in $\hat{\Omega}_{k(r)}$; $N_{k(r)}^{pos(r)}$ represents the number of these samples and also of the precomputed basis functions: in general it may be different for each reference domain and for each Neumann boundary condition set. With this strategy, we can have proper spaces for each possible position of the corresponding deformed subdomain Ω_r (inflow, central, outflow element) on Ω .

In the online stage, for every given new parameter $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_r)$ and for every generic combination of the deformed subdomains $\Omega_r \subset \Omega$, $r = 1, \dots, R$, we have the following steps.

- For every Ω_r , we select the proper corresponding precomputed reduced basis spaces, depending on $k(r)$ and on the “position” of Ω_r in Ω (inflow, central or outflow element). We denote with $\mathbf{u}_h^{\mathbf{r}}(\boldsymbol{\mu}_r^i)$, $\mathbf{v}_h^{\mathbf{r}}(\boldsymbol{\mu}_r^i)$ and $p_h^{\mathbf{r}}(\boldsymbol{\mu}_r^i)$ the reduced basis functions (snapshots) of the corresponding spaces $Y_{k(r)}^{pos(r)}$ and $M_{k(r)}^{pos(r)}$ computed in the domain $\hat{\Omega}_{k(r)}$, where the index \mathbf{r} is a couple of integers: $k(r)$

and the position index $pos(r)$, $\mathbf{r} = (k(r), pos(r))$.

- A Finite Element solution $\mathbf{u}_H(\boldsymbol{\mu}), p_H(\boldsymbol{\mu})$ of the Stokes problem (2.4.1) is computed in a fast way by using a coarse mesh \mathcal{T}_H on the whole domain Ω^2 . Together with the associated supremizer solution $\mathbf{v}_H(\boldsymbol{\mu})$, we restrict these functions to each subdomain Ω_r , then we map them in the corresponding reference domain $\hat{\Omega}_{k(r)}$. We denote with $\mathbf{u}_H^r(\boldsymbol{\mu}), \mathbf{v}_H^r(\boldsymbol{\mu}), p_H^r(\boldsymbol{\mu})$ the interpolation of these coarse solutions on the fine mesh $\mathcal{T}_h \cap \hat{\Omega}_{k(r)}$. These functions are obviously continuous along the internal interfaces and they bring important physical information.

- We define in $\hat{\Omega}_{k(r)}$ the following spaces:

$$\begin{aligned} Y^{\mathbf{r}} &= span \left\{ \mathbf{u}_h^r(\boldsymbol{\mu}_r^i), \mathbf{v}_h^r(\boldsymbol{\mu}_r^i), \mathbf{u}_H^r(\boldsymbol{\mu}), \mathbf{v}_H^r(\boldsymbol{\mu}), i = 1, \dots, N^{\mathbf{r}} \right\}, \\ M^{\mathbf{r}} &= span \left\{ p_h^r(\boldsymbol{\mu}_r^i), p_H^r(\boldsymbol{\mu}), i = 1, \dots, N^{\mathbf{r}} \right\}, \end{aligned}$$

where $\mathbf{r} = (k(r), pos(r))$, $r = 1, \dots, R$, $k \in \{1, \dots, K\}$, $pos(r) \in \{in, c, out\}$ and $N^{\mathbf{r}} = N_{k(r)}^{pos(r)}$.

- Recalling that $\boldsymbol{\mu}$ is the selection of the parameters $\boldsymbol{\mu}_r$, $r = 1, \dots, R$ which define the computational domain $\bar{\Omega} = \bigcup_{r=1}^R \bar{\Omega}_r$, the global reduced basis “hybrid” solution is defined by a suitable local linear combination of the precomputed functions that define $(\mathbf{u}_N(\boldsymbol{\mu}_r), p_N(\boldsymbol{\mu}_r))$ on each subdomain Ω_r as follows, for $r = 1, \dots, R$:

$$\begin{aligned} \mathbf{u}_N|_{\Omega_r}(\boldsymbol{\mu}) &= \sum_{i=1}^{N^{\mathbf{r}}} \alpha_i^r \mathbf{u}_h^r(\boldsymbol{\mu}_r^i) + \sum_{i=1}^{N^{\mathbf{r}}} \beta_i^r \mathbf{v}_h^r(\boldsymbol{\mu}_r^i) + \eta \mathbf{u}_H^r(\boldsymbol{\mu}) + \delta \mathbf{v}_H^r(\boldsymbol{\mu}) \\ p_N|_{\Omega_r}(\boldsymbol{\mu}) &= \sum_{i=1}^{N^{\mathbf{r}}} \gamma_i^r p_h^r(\boldsymbol{\mu}_r^i) + \epsilon p_H^r(\boldsymbol{\mu}). \end{aligned} \quad (2.4.8)$$

The coefficients $\alpha_j^r, \beta_j^r, \gamma_j^r$ and η, δ, ϵ are determined by solving the original problem (2.4.1) on Ω by using the Galerkin method on the reduced spaces $Y^{\mathbf{r}}$ and $M^{\mathbf{r}}$. Since the spaces $Y^{\mathbf{r}}$ are made of discontinuous velocities, their jumps across the interfaces are minimized through the use of Lagrange multipliers. With this aim, we define $\forall \boldsymbol{\psi} \in W_{ml}$, $m, l \in \{1, \dots, R\}$,

$$\mathcal{L}^{ml}(\mathbf{u}_N(\boldsymbol{\mu}), \boldsymbol{\psi}) = \int_{\Gamma_{ml}} (\mathbf{u}_N|_{\Omega_m}(\boldsymbol{\mu}) - \mathbf{u}_N|_{\Omega_l}(\boldsymbol{\mu})) \boldsymbol{\psi} ds, \quad (2.4.9)$$

where Γ_{ml} is the interface between two adjacent subdomains denoted with the indices m and l respectively and W_{ml} is a low order polynomial space defined on this interface, see [12, 59]. A basis for W_{ml} is provided by the characteristic Lagrange polynomials $\boldsymbol{\psi}_q$, $q = 1, \dots, Q_{ml}$ associated with the Q_{ml} nodes of Γ_{ml} .

If we suppose that Ω has $R - 1$ internal interfaces, Γ_{ii+1} , $i = 1, \dots, R - 1$, we therefore solve the following problem (which generalizes problem (2.4.3) that was set for the two subdomains case):

²The coarse solution is inexpensive from a computational point of view and also from practical point of view since it is computed on a combination and repetition of only reference subdomains, which are easily constructed by translation starting by the reference sub-blocks.

find $\mathbf{u}_N(\boldsymbol{\mu}) \in Y^1 \times \dots \times Y^R$, $p_N(\boldsymbol{\mu}) \in M^1 \times \dots \times M^R$, $\boldsymbol{\lambda}_N \in W_{ii+1}$, $i = 1, \dots, R-1$, such that

$$\begin{cases} \mathcal{A}(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{w}, \boldsymbol{\mu}) + \mathcal{B}(\mathbf{w}, p_N(\boldsymbol{\mu}), \boldsymbol{\mu}) + \sum_{i=1}^{R-1} \mathcal{L}^{ii+1}(\mathbf{w}, \boldsymbol{\lambda}_N) = \mathcal{F}(\mathbf{w}, \boldsymbol{\mu}) & \forall \mathbf{w} \in Y^1 \times \dots \times Y^R, \\ \mathcal{B}(\mathbf{u}_N(\boldsymbol{\mu}), q, \boldsymbol{\mu}) = 0 & \forall q \in M^1 \times \dots \times M^R, \\ \mathcal{L}^{ii+1}(\mathbf{u}_N(\boldsymbol{\mu}), \boldsymbol{\psi}) = 0 & i = 1, \dots, R-1, \forall \boldsymbol{\psi} \in W_{ii+1}. \end{cases} \quad (2.4.10)$$

As before, on each interface $\boldsymbol{\lambda}_N$ represents the normal stresses.

In this case the subdomains on Ω are “consecutive”, for every Ω_r , $r = 2, \dots, R-1$, there are just two internal interfaces Γ_{r-1r} and Γ_{rr+1} , but in the most general case, we have to consider in problem (2.4.10) every internal interface Γ_{ml} and every corresponding bilinear form $\mathcal{L}^{ml}(\cdot, \cdot)$. As already mentioned in Section 2.2, in case of null force term, the local Stokes problem for the central domains will generate a null solution. In this particular case, the global solution will coincide with the interpolation of the coarse one. However, every tests has been carried out by imposing a non-zero force term.

2.4.3 Computational strategy

In order to illustrate and build the final linear system that has to be solved, we consider another example made up by three subdomains:

$$\begin{aligned} \overline{\Omega}(\boldsymbol{\mu}) &= \overline{\Omega_1} \cup \overline{\Omega_2} \cup \overline{\Omega_3} = \overline{T_{\boldsymbol{\mu}_1}^1(\hat{\Omega}_1)} \cup \overline{T_{\boldsymbol{\mu}_2}^1(\hat{\Omega}_1)} \cup \overline{T_{\boldsymbol{\mu}_3}^2(\hat{\Omega}_2)}, \quad \Omega_1 \cap \Omega_2 \cap \Omega_3 = \emptyset, \\ \Gamma_{12} &= \tilde{\Omega}_1 \cap \tilde{\Omega}_2, \quad \Gamma_{23} = \tilde{\Omega}_2 \cap \tilde{\Omega}_3. \end{aligned}$$

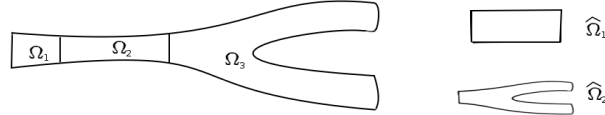


Figure 2.3: Geometrical scheme for a domain composed by R=3 deformations of K=2 reference domains.

In this case the three subdomains can be obtained as deformations of two different reference domains $\hat{\Omega}_1$ and $\hat{\Omega}_2$.

As described in the Section 2.4.2 (offline stage), we have three local problems in the form (2.4.6), and we build independently (i.e. in parallel computations) the three couples of reduced basis spaces $Y_1^{in}, M_1^{in}, Y_1^c, M_1^c, Y_2^{out}, M_2^{out}$.³

In the online stage, for the parameter $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3)$, we compute the coarse global solution, we interpolate it in the fine mesh and we add the restrictions to the reduced spaces. The interpolation process of the coarse mesh on the finer one is still a reasonable operation in terms of online computational cost, but as shown in the numerical tests (next section), it is always substantially less expensive than computing a true fine finite element solution.

³ In order to solve many other possible configurations in the online step, we may have to compute also the spaces $Y_1^{out}, M_1^{out}, Y_2^{in}, M_2^{in}, Y_2^c, M_2^c$.

We define:

$$\begin{aligned}
 Y^1 &= \text{span} \left\{ \mathbf{u}_H|_{\hat{\Omega}_1}, \mathbf{v}_H|_{\hat{\Omega}_1}, \mathbf{u}_h^1(\boldsymbol{\mu}_1^i), \mathbf{v}_h^1(\boldsymbol{\mu}_1^i), i = 1, \dots, N^1 \right\}, \\
 M^1 &= \text{span} \left\{ p_H|_{\hat{\Omega}_1}, p_h^1(\boldsymbol{\mu}_1^i), i = 1, \dots, N^1 \right\}, \\
 Y^2 &= \text{span} \left\{ \mathbf{u}_H|_{\hat{\Omega}_1}, \mathbf{v}_H|_{\hat{\Omega}_1}, \mathbf{u}_h^2(\boldsymbol{\mu}_2^i), \mathbf{v}_h^2(\boldsymbol{\mu}_2^i), i = 1, \dots, N^2 \right\}, \\
 M^2 &= \text{span} \left\{ p_H|_{\hat{\Omega}_1}, p_h^2(\boldsymbol{\mu}_2^i), i = 1, \dots, N^2 \right\}, \\
 Y^3 &= \text{span} \left\{ \mathbf{u}_H|_{\hat{\Omega}_2}, \mathbf{v}_H|_{\hat{\Omega}_2}, \mathbf{u}_h^3(\boldsymbol{\mu}_3^i), \mathbf{v}_h^3(\boldsymbol{\mu}_3^i), i = 1, \dots, N^3 \right\}, \\
 M^3 &= \text{span} \left\{ p_H|_{\hat{\Omega}_2}, p_h^3(\boldsymbol{\mu}_3^i), i = 1, \dots, N^3 \right\}.
 \end{aligned}$$

Here the index $\mathbf{r} = 1$ refers to $\text{pos}(1) = in$ and $k(1) = 1$, the index $\mathbf{r} = 2$ to $\text{pos}(2) = c$ and $k(2) = 1$ and the index $\mathbf{r} = 3$ to $\text{pos}(3) = out$ and $k(3) = 2$.

The final linear system of the reduced basis hybrid problem includes the matrices of the three problems defined in the offline stage (that correspond to the matrices A^r and B^r and the vectors F^r , $r = 1, 2, 3$ defined in (2.4.5)) and two matching conditions (corresponding to the two internal interfaces) described by (2.4.9) (involving the matrices L_{lm}^r , $l = 1, 2, m = 2, 3$ defined in (2.4.12)):

$$\begin{pmatrix}
 A^1 & 0 & 0 & B^1 & 0 & 0 & L_{12}^1 & 0 \\
 0 & A^2 & 0 & 0 & B^2 & 0 & -L_{12}^2 & L_{23}^2 \\
 0 & 0 & A^3 & 0 & 0 & B^3 & 0 & -L_{23}^3 \\
 B^{1T} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & B^{2T} & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & B^{3T} & 0 & 0 & 0 & 0 & 0 \\
 L_{12}^{1T} & -L_{12}^{2T} & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & L_{23}^{2T} & -L_{23}^{3T} & 0 & 0 & 0 & 0 & 0
 \end{pmatrix}
 \begin{pmatrix}
 \mathbf{u}_N \\
 \mathbf{p}_N \\
 \boldsymbol{\lambda}_1 \\
 \boldsymbol{\lambda}_2
 \end{pmatrix}
 =
 \begin{pmatrix}
 F^1 \\
 F^2 \\
 F^3 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{pmatrix} \quad (2.4.11)$$

where

$$\mathbf{u}_N = \begin{pmatrix} \boldsymbol{\alpha}^1 \\ \boldsymbol{\beta}^1 \\ \boldsymbol{\alpha}^2 \\ \boldsymbol{\beta}^2 \\ \boldsymbol{\alpha}^3 \\ \boldsymbol{\beta}^3 \\ \eta \\ \delta \end{pmatrix} \mathbf{p}_N = \begin{pmatrix} \boldsymbol{\gamma}^1 \\ \boldsymbol{\gamma}^2 \\ \boldsymbol{\gamma}^3 \\ \epsilon \end{pmatrix} \boldsymbol{\alpha}^i = \begin{pmatrix} \alpha_1^i \\ \vdots \\ \alpha_{N_i}^i \end{pmatrix} \boldsymbol{\beta}^i = \begin{pmatrix} \beta_1^i \\ \vdots \\ \beta_{N_i}^i \end{pmatrix} \boldsymbol{\gamma}^i = \begin{pmatrix} \gamma_1^i \\ \vdots \\ \gamma_{N_i}^i \end{pmatrix} \quad i = 1, 2, 3,$$

$$(L_{ml}^m)_{iq} = \int_{\Gamma_{ml}} \mathbf{u}_h^m(\boldsymbol{\mu}_m^i) \boldsymbol{\psi}_q ds, \quad m, l \in \{1, \dots, R\}, 1 \leq q \leq 2Q_{ml}, 1 \leq i \leq 2N^m. \quad (2.4.12)$$

We note that for the example at hand reported schematically in Figure 2.3, the dimension of the linear system is determined by two quantities: the dimensions of the reduced basis spaces, N^1 , N^2 and N^3 , the corresponding dimensions of A^1 , A^2 , A^3 (respectively $2(N^1 + 1) \times 2(N^1 + 1)$, $2(N^2 + 1) \times 2(N^2 + 1)$ and $2(N^3 + 1) \times 2(N^3 + 1)$) and B^1 , B^2 , B^3 (respectively $2(N^1 + 1) \times (N^1 + 1)$, $2(N^2 + 1) \times (N^2 + 1)$ and

$2(N^3 + 1) \times (N^3 + 1)$); the number of nodes K_{rm} on the internal interfaces Γ_{rm} affects the dimension of $L_{12}^1, L_{12}^2, L_{23}^2$ and L_{23}^3 (given respectively by $2N^1 \times K_{12}, 2(N^2 + 1) \times K_{12}, 2(N^2 + 1) \times K_{23}$ and $2(N^3 + 1) \times K_{23}$). Without any further assumption on the basis for the velocity and the pressure spaces, in order to match the nQ_{ml} Lagrange multipliers equations with the N^m basis functions of each subdomain, we have always imposed $N^m > Q_{ml}$.

2.5 Numerical tests on 2D domains

Numerical tests were carried out on different problems such as stenosis and bifurcation to solve Stokes⁴ flows with the reduced basis hybrid method using the non-affine transfinite map. Taylor-Hood Finite Element Method has been used to compute the approximated basis functions, \mathbb{P}_2 elements for velocity and supremizer, \mathbb{P}_1 for pressure, respectively [76].

2.5.1 A first example: stenosed arterial vessel

As first proposed test we model the blood flow through an artery occluded by stenoses [24]. Every stenosis represents a subdomain and it has curved and parametrized walls.

In particular, the deformations of the single stenosis domain depend on two parameters: the amplitudes, $\mu_1 \in [0, 2]$ and $\mu_2 \in [0, 2]$, on the upper and lower walls representing two independent contractions, respectively.

The deformed single block domain Ω_i in Figure 2.4 is mapped starting from the straight reference pipe $\hat{\Omega}_1$ of length L and height D through a transfinite map.

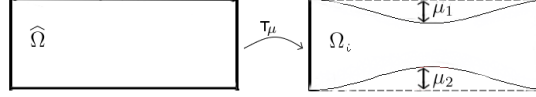


Figure 2.4: Geometrical scheme for the stenosis block.

The computational domain Ω for the model has been composed by three stenosed blocks so that $\bar{\Omega} = \cup_{i=1}^3 \bar{\Omega}_i$, as shown in Figure 2.5.

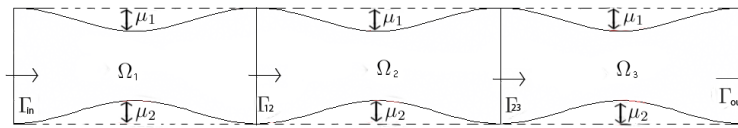


Figure 2.5: Geometrical scheme for the computational domain.

We consider a parametrized Stokes problem (2.3.6) for each subdomain, where the force field \mathbf{f} is the gravity. For the inflow subdomain, we compute the reduced basis imposing zero Dirichlet condition on the wall, Neumann boundary conditions given by imposing $\boldsymbol{\sigma}_n = \boldsymbol{\sigma}\mathbf{n} = \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n}$ to be $\boldsymbol{\sigma}_n^{in} = [-1, 0]^T$ on Γ_{in} and $\boldsymbol{\sigma}_n^{out} = \mathbf{0}$ on the internal interface Γ_{12} . For the outflow subdomain, we compute the reduced basis imposing zero Dirichlet condition on the wall, Neumann boundary conditions imposing $\boldsymbol{\sigma}_n^{in} = \mathbf{0}$ on the internal interface Γ_{23} and $\boldsymbol{\sigma}_n^{out} = [1, 0]^T$ on the outflow interface Γ_{out} . When we consider the internal subdomain, we impose zero Dirichlet condition on the walls and homogeneous Neumann

⁴Under the assumption we are considering low Reynolds number, with low average blood velocity and in mid-sized arteries, this is an acceptable approximation (for example in coronary arteries), [1, 74].

boundary conditions on Γ_{12} and Γ_{23} .

We apply the transfinite map (Section 1.3) to transform the problem in terms of reference coordinates. By referring to a single stenosed block we expand each geometrical component in order to deal with an affine decomposition. We use the empirical interpolation to decompose the terms (2.2.3), (2.2.6) and (2.2.7). The maximum interpolation error is set to $\epsilon_{tol}^{EIM} = 10^{-6}$. By applying the offline stage of the reduced basis method to the single stenosis blocks, a set of $N^i = 40$, with $i = 1, 2, 3$, combinations of parameters is selected by the greedy algorithm [82] using a tolerance $\epsilon_{greedy} = 10^{-7}$. Figure 2.6 shows the clustered distribution of these parameters used to store the basis functions [84]. Note that the classical a posteriori error bounds is valid in this single block case [87]. Coarse and fine grids have been chosen in order to deal with respectively 200 and 1583 nodes in the whole domain Ω , so that $H/h \approx 3$.

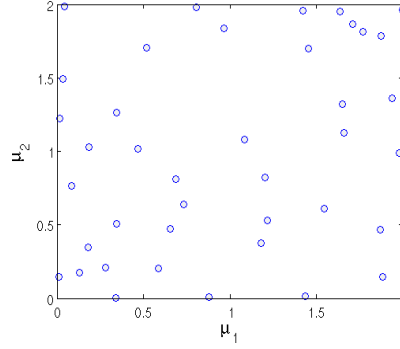


Figure 2.6: Parameter distribution representing the parameters combinations selected to generate the basis functions in a single block.

Figure 2.7 shows an example of a representative flow solution, found with the RBHM, for certain parameters combination ($\mu_1 = (0.1, 2)$, $\mu_2 = (0.1, 2)$, $\mu_3 = (2, 0.1)$), to be compared with the finite element solution. The same comparison, regarding the pressure solutions, is shown in Figure 2.8.

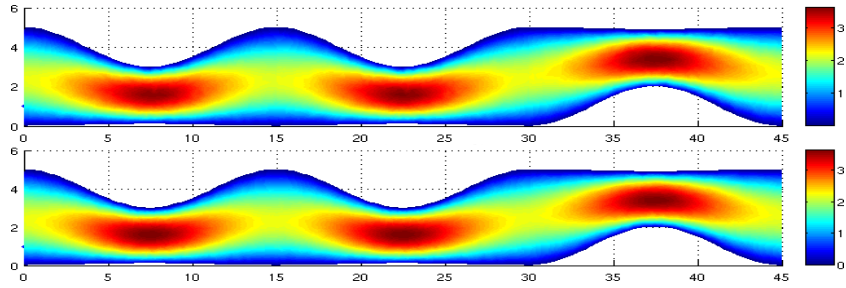


Figure 2.7: Velocity intensity representative solutions using RBHM (with $N_1 = N_2 = N_3 = 10$) (top) and solving a global FEM using the triangulation \mathcal{T}_h (bottom), ($\mu_1 = (0.1, 2)$, $\mu_2 = (0.1, 2)$, $\mu_3 = (2, 0.1)$).

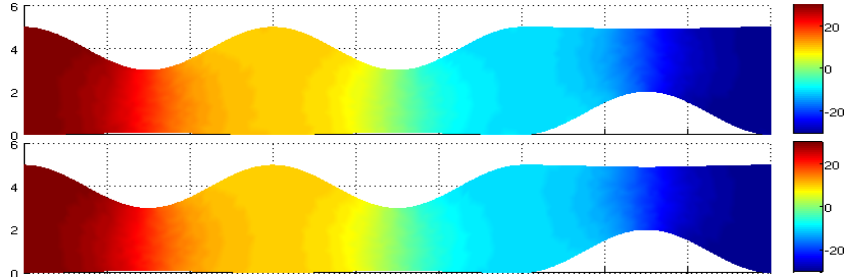


Figure 2.8: Pressure representative solutions using RBHM (with $N_1 = N_2 = N_3 = 10$) (top) and solving a global FEM in \mathcal{T}_h (bottom), ($\mu_1 = (0.1, 2)$, $\mu_2 = (0.1, 2)$, $\mu_3 = (2, 0.1)$).

Figure 2.9 shows the reduction of the H^1 and L^2 relative errors, respectively, on velocity and pressure between the “truth” FE solution and the RBHM approximation. We plot the minimum, the maximum and the average of the errors obtained with a set of 1000 samples, increasing the number N of basis functions.

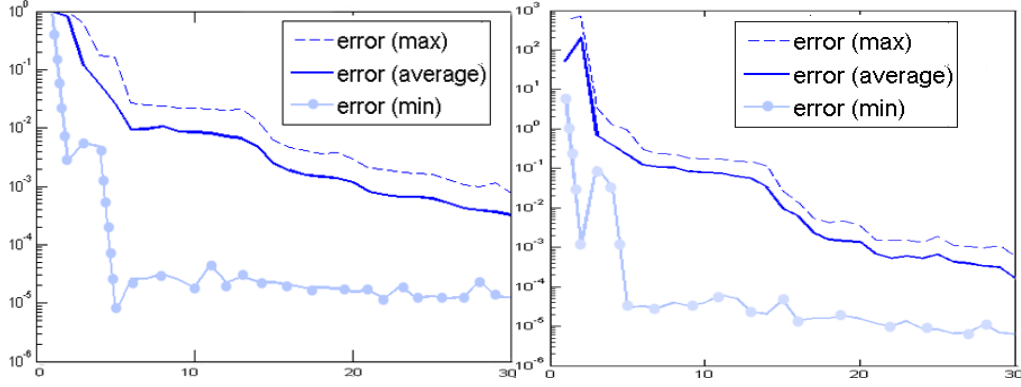


Figure 2.9: H^1 and L^2 relative error (minimum, maximum and average in a set of 1000 samples) on velocity (left) and pressure (right).

Figures 2.10 and 2.11 show the pressure profiles on the internal interfaces Γ_{12} and Γ_{23} obtained by solving the Stokes problem by using the Lagrange multipliers but not including the coarse correction to the reduced spaces (so without guaranteeing the continuity and the physical correctness of stresses), then, as second option, including the coarse correction and not using the Lagrange multipliers correction (not guaranteeing the continuity of velocity) and finally by using the RBHM method (that includes both velocity and coarse corrections). The profiles of the corresponding fine FEM solution computed in the whole network has been plotted in the same figure in order to compare the quality of the solutions.

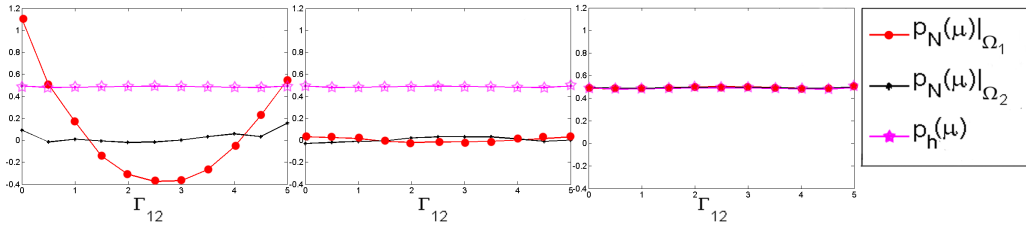


Figure 2.10: Pressure profiles along the internal interface Γ_{12} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

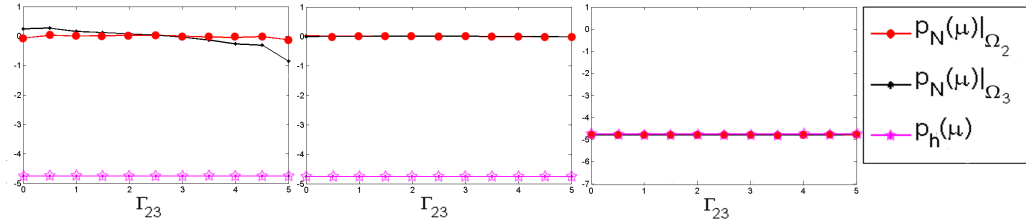


Figure 2.11: Pressure profiles along the internal interface Γ_{23} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

The same comparison on the velocity profiles is shown in Figure 2.12 and 2.13, while in Figures 2.14, 2.15 2.16 and 2.17 is shown the comparison of the normal and tangential component profiles of the

normal stress, defined respectively as $\sigma_n \mathbf{n} = \nu \frac{\partial(\mathbf{u} \cdot \mathbf{n})}{\partial \mathbf{n}} - p$ and $\sigma_n \mathbf{t} = \nu \frac{\partial(\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}}$.

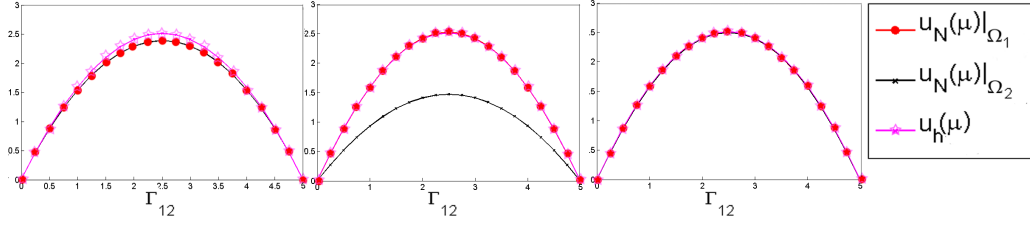


Figure 2.12: Velocity profiles along the internal interface Γ_{12} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

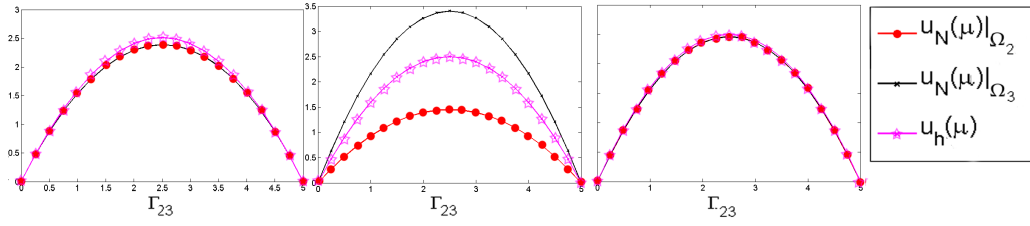


Figure 2.13: Velocity profiles along the internal interface Γ_{23} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

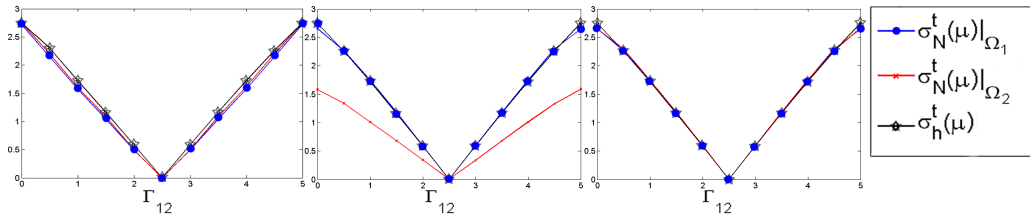


Figure 2.14: Tangential component of normal stress profiles along the internal interface Γ_{12} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

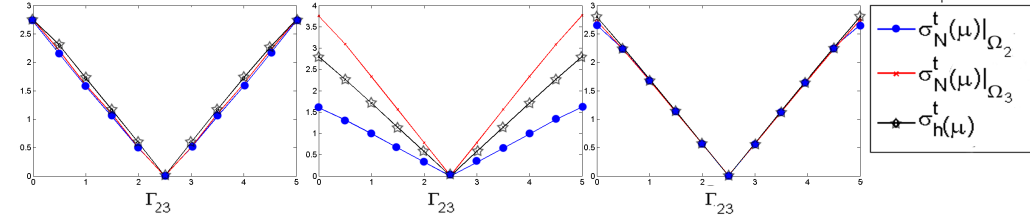


Figure 2.15: Tangential component of normal stress profiles along the internal interface Γ_{23} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

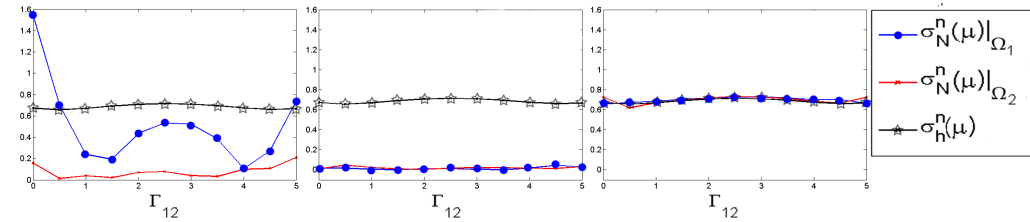


Figure 2.16: Normal component of normal stress profiles along the internal interface Γ_{12} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

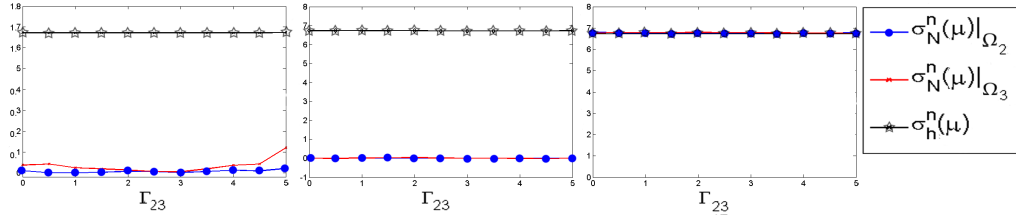


Figure 2.17: Normal component of normal stress profiles along the internal interface Γ_{23} without using the coarse correction (left), without using the Lagrange multipliers (center) and solving the complete RBHM problem (right).

In the figures shown, it is evident that the imposition of continuity of the velocities at the interfaces by Lagrange multipliers guarantees, by construction, the continuity of the tangential component of the normal stresses too, while that of the normal component of the normal stresses is indeed satisfied thanks to the coarse correction.

The RBHM, by including the correction by Lagrange multipliers for the velocity and the coarse correction, provides an approximation of the solution that recovers the continuity of velocity and that of both components of normal stresses at the interfaces.

We remark that the use of Lagrange multipliers for velocity correction as proposed in RBEM is already able to guarantee the continuity of velocity with a good accuracy, lightly improved in the test with the use of RBHM and the continuity of the tangential component of normal stress, but not the continuity of the normal component of the normal stress.

Figure 2.18 shows the maximum error between the RBHM solution and the (true) FE fine solution. We note that when $H = h$, we add to the reduced space the couple (\mathbf{u}_H, p_H) that is the “exact” solution of the interpolated problem. In this case this special basis will give all the contribution to the final solution and the error will be the smallest (10^{-6} in our case, since the empirical interpolation tolerance is set to this value). By increasing the value of H , the accuracy of the solution is given principally by that of the fine basis functions. Note however that, when increasing H , the error does not deteriorate. Otherwise said, the coarse solution guarantees the correct recovery of the physical normal stresses at the interfaces, without strongly affecting the accuracy yielded by the reduced basis functions.

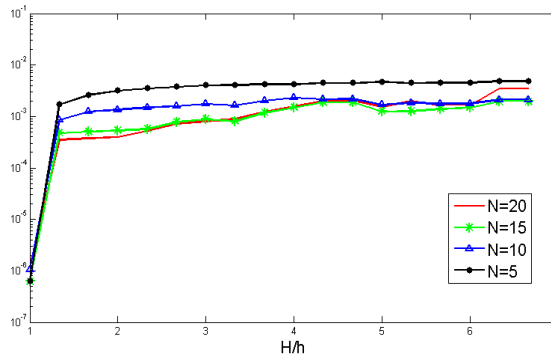


Figure 2.18: The effect of the coarse mesh on the overall accuracy of the RBHM.

2.5.2 A portion of a network with heterogeneous block domains

In the second test case, the RBHM has been applied to the solution of the Stokes problem in a configuration that can be referred as a union of two subdomains, see Figure 2.19, that simulate respectively a bifurcation and a stenosis.

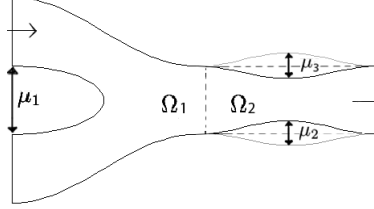


Figure 2.19: Geometrical scheme for the computational domain.

The two subdomains are obtained through two non-affine transfinite maps. The map that defines Ω_1 depends on the bifurcation span $\mu_1 \in [0.8, 3.5]$. Ω_2 is defined through the maps that depends on the parameter $\mu_2 = (\mu_2, \mu_3)$ that represents the amplitudes: $\mu_2 \in [-1, 1]$ and $\mu_3 \in [-1, 1]$, on the upper and lower walls representing the dilatation and the contraction of the pipe. This configuration can be used, for example, to model a carotid artery bifurcation, [24].

We consider the bifurcation as the inflow subdomain in which we set homogeneous Dirichlet condition on the wall and on the lower branch of the bifurcation, Neumann boundary conditions given by imposing $\sigma_n^{in} = [-10, 0]^T$ on Γ_{in} and $\sigma_n^{out} = \mathbf{0}$ on the internal interface Γ_{12} . We assume that the stenosis block is the outflow domain and we set zero Dirichlet condition on the wall, while we impose Neumann boundary conditions $\sigma_n^{in} = \mathbf{0}$ on the internal interface Γ_{12} and $\sigma_n^{out} = [10, 0]^T$ on the outflow interface Γ_{out} .

We apply the transfinite map to transform the problem in terms of reference coordinates.

By referring to a single block we expand each geometrical component in order to deal with an affine decomposition. The terms (2.2.3), (2.2.6) and (2.2.7) are treated by the empirical interpolation method (EIM). The maximum interpolation error is set $\epsilon_{tol}^{EIM} = 10^{-6}$.

By applying the greedy algorithm, we select $N_1 = 13$ parameters for μ_1 and $N_2 = 15$ parameter combinations for $\mu_2 = (\mu_2, \mu_3)$. Figure 2.20 shows the distribution of these parameters used to generate the basis functions.

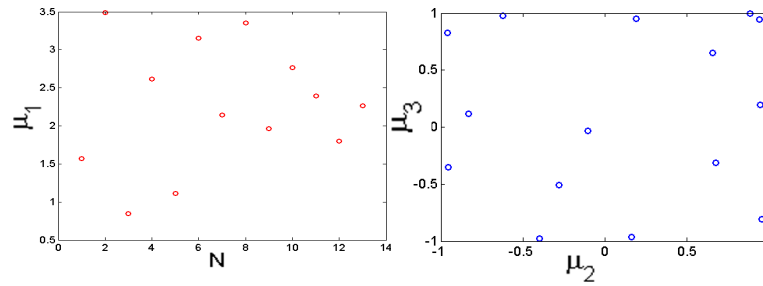


Figure 2.20: Parameter distribution representing the parameters combinations selected to generate the basis functions in the two blocks by greedy algorithm.

Coarse and fine grids have been chosen in order to deal with respectively 269 and 1006 nodes in the whole domain Ω . Figures 2.21 and 2.22 show an example of flow solution, obtained using the reduced basis hybrid method, for a certain parameters combination ($\mu_1 = 2.5$, $\mu_2 = (\mu_2, \mu_3) = (1, 0.9)$), which can be compared with the solutions obtained with finite element method.

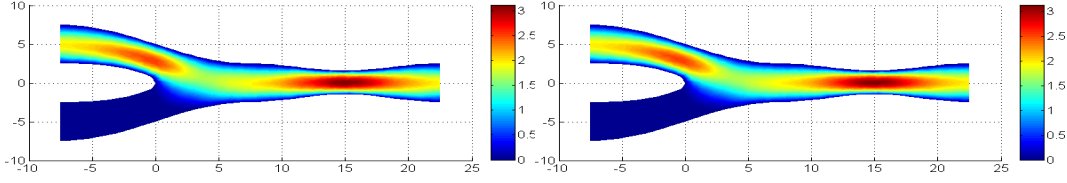


Figure 2.21: Velocity intensity representative solutions using RBHM (with $N_1 = N_2 = 10$) (left) and by a global computed FEM solution (right), ($\mu_1 = 2.5$, $\mu_2 = (1, 0.9)$).

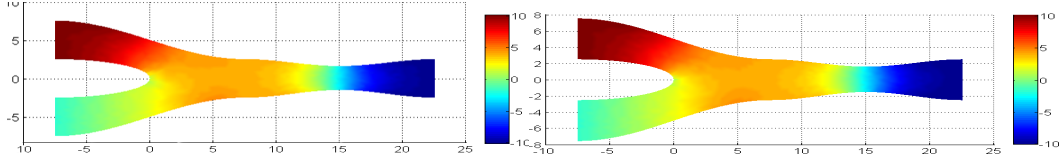


Figure 2.22: Pressure representative solutions using RBHM (with $N_1 = N_2 = 10$) (left) and by a global computed FEM solution (right), ($\mu_1 = 2.5$, $\mu_2 = (1, 0.9)$).

2.6 Computational costs

As already anticipated, the main feature of RBHM (as well as RBEM) is its capability to perform, thanks to the heavy computation done once in an offline stage, simulations on different combinations of the block domains by guaranteeing a certain versatility in combining several configurations and networks. The goal is to provide a method with lower complexity and lower computational times than the finite element method but able to guarantee an accurate solution and a certain physical reliability for velocity, pressure and stresses.

As already mentioned, the RBHM consists in two main steps. A parallel computational strategy can abate computational times, mainly in the offline step. If we have K reference domains and we want to deal, for example, with “inflow”, “central” and “outflow” elements, we can have a significant computational time reduction by computing the offline step with $3K$ parallel processors, one for each reference element. During the online step, only the matrices assembling process can be partitioned in a parallel computation and, in this case, we need R processors, one for each subdomain. In Figure 2.23 the computational times required by the global finite elements solutions and by the online stage of the RBHM are represented by increasing the number of stenosis blocks of Section 2.5.1. The CPU time of RBHM breaks down into three components, respectively due to: the FEM coarse solution, the matrix assembling for each block, and the resolution of the reduced basis linear system. The second part can be computed independently for each block, on a parallel computational architecture where every processor deals with a single block.

The computational time of the online step of RBHM is not expensive and we have not big advantages by dealing with parallel computation in this step, nonetheless we have done it in order to show how every ingredient of the method contributes to the total computational time. In particular, in Figure 2.23 we can observe that if we treat each block by using parallel computation, the CPU time is mostly due to the FEM coarse solution. Thus we can obtain an online solution with an accuracy comparable with the one of the fine finite element solution at the cost of a coarse finite element solution. This achievement is gotten by reduced basis techniques and proper coupling conditions, where the coarse FEM solution is playing a crucial role (i.e. a lift) in guaranteeing the continuity of stresses. We also underline that the computational advantages are more evident for extended networks and an increasing number of blocks.

In order to visualize the different computational loads and the advantage of the reduced model pro-

posed, we report in Table 2.1 the values of the CPU times in details for different number R of subdomains, the computational time for the matrix assembling is 0.76s for each of the subdomains considered. The last column underlines the computational costs of RBHM compared with the fine FEM solution. We can see that, in terms of computational cost, the fine solution computed with RBHM is comparable to the one obtained on the coarser grid with standard FEM, and in general the computational savings are of two orders of magnitude with RBHM compared with FEM. Moreover the CPU time spent for a FEM simulation with 5 blocks is on par with RBHM simulation dealing with a domain defined by 27 blocks, still retaining the continuity of velocities and stresses at the interfaces.

R	Fine FEM solution	Coarse FEM solution	Reduced Linear System	RBHM solution	RBHM vs Fine FEM(%)
5	31.13	1.73	0.06	2.72	8.76
10	132.18	4.86	0.14	5.68	4.30
15	311.44	10.18	0.23	11.08	3.56
20	557.57	16.77	0.28	17.81	3.19
25	880.54	23.86	0.60	25.22	2.86
30	1183.5	34.81	0.78	36.35	3.07
35	1895.7	49.74	1.02	51.52	2.71
40	2484.6	70.44	1.56	72.76	2.92

Table 2.1: Computational times (in seconds) of FEM and RBHM for different number of subdomains R .

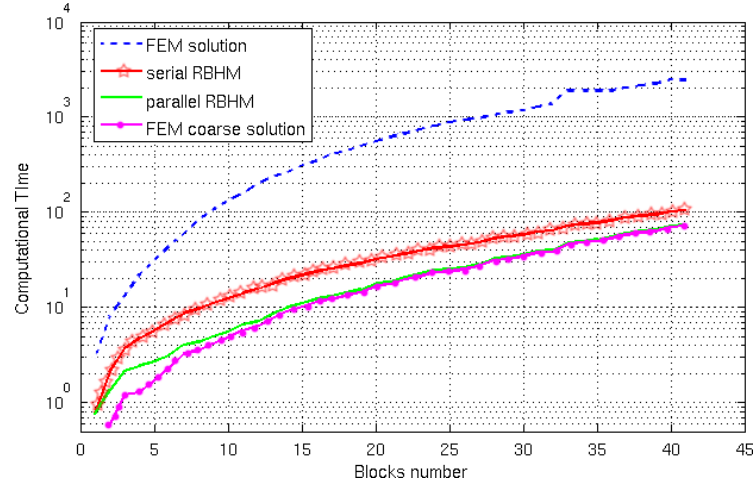


Figure 2.23: Computational times (in seconds) of FEM and RBHM

2.7 Comparison between RBHM and RB

As widely discussed in Chapter 1, the classical RB method is used when we want to solve rapidly a large number of problems governed by the same partial differential equation that depends on parameters [83, 84]. It has been developed in a mono-domain case and it is highly efficient when we deal with geometry endowed with topological similarities [61, 62]. If we want to consider repetitive and heterogeneous geometries composing a network, an offline computation for each new combination of domain configuration has to be performed. The reduced basis element method RBEM avoids this problem and allows dealing with every kind of combinations of a certain number of blocks, for which few offline stages can be computed independently. The proposed hybrid version, RBHM, combines the previous approach with a FEM coarse solution to guarantee the continuity of both velocity and stresses solutions across interfaces. We want now to consider a three stenoses configuration (presented in Section 2.5.1)

for a comparative analysis between the RB method and the RBHM. By using the classical RB method, we have to perform the offline stage in the whole domain characterized by six parameters (two for each stenoses), coupling conditions are automatically satisfied. During this step, the empirical interpolation generates $Q_a = 90$ terms for the affine decomposition of the bilinear forms and the greedy algorithm needs 49 basis functions in order to reach an error with 10^{-4} of tolerance.

Using the proposed RBHM we compute the offline stage in a single stenosed domain, we consider smaller number of parameters (only two) allowing to deal with just 29 terms for the affine decomposition of the bilinear forms representing the problems (always carried out by the empirical interpolation) and 14 basis functions to reach an error of order 10^{-3} . The comparison between the features of the two methods is shown in Table 2.2. The complexity reduction is very important also for the offline step.

Figure 2.24 shows how the number of parameters affects the choice of basis functions. We can observe that, during the greedy RB spaces assembling, in the case of three stenosed domains (6 parameters), we need more than three times the number of basis functions compared with the number we need in the case of a single stenosis in order to reach the same convergence relative error (2 parameters). In the single domain case, for a tolerance on the greedy algorithm of $\epsilon = 10^{-3}$ we need just $N = 7$, in the three stenosis domain $N = 26$, while for $\epsilon = 10^{-4}$, respectively, $N = 11$ and $N = 45$. In both cases we can conclude that three times the number of basis for the single domain case is less than the number of basis that we have in the three stenosis domain⁵, $3N = 21 < 26$ and $3N = 33 < 45$. With the application of RBHM we can reduce also the complexity of the operators representing the problem ($M^{ak} = \sum_{ij} M_{ij}^{ak}$ and $M^{bk} = \sum_{ij} M_{ij}^{bk}$ of (2.2.9) and (2.2.10)) and the dimension of the RB spaces. This is useful also for a good performance of a posteriori error bounds [87].

	RB	RBHM
H^1 rel. vel. error	1E-04	1E-04
N	45	3×14
ϵ_{EIM}	1E-06	1E-06
M^{ak}	71	24
M^{bk}	19	5

Table 2.2: Computational complexity of RBM and RBHM.

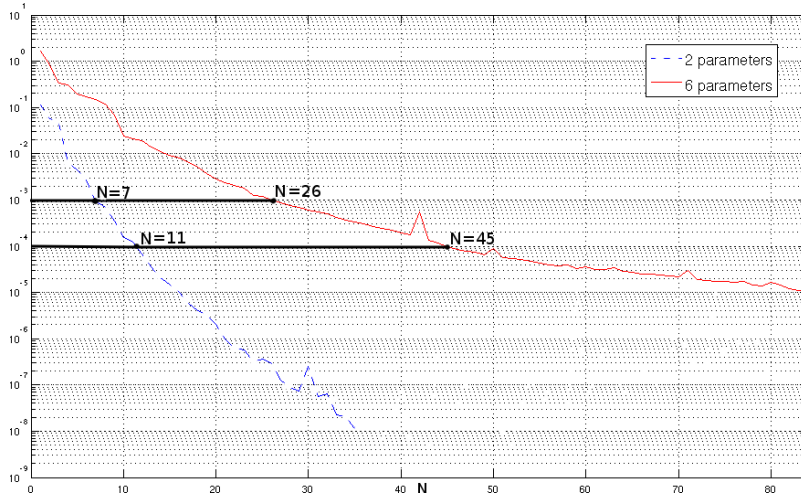


Figure 2.24: Relative H^1 velocity errors during the greedy RB spaces assembling dealing with 2 parameters and 6 parameters.

⁵Of course we have to take into account the further costs and effort of the coupling condition in the use of RBHM, but the proposed method still keeps reasonable computational advantages.

2.8 Concluding remarks

We have presented an extension of the RBEM [52] in a multi-domain flow network, based on an online correction of the solution, given by a coarse FE solution. The RBHM is a suitable method for solving the viscous fluid flow problems in complex geometries, in order to deal with a computational domain decomposed by several combinations of repetitive blocks, on which the solution can be computed locally and quickly, thanks to the classical RB method, and then properly coupled and glued guaranteeing the continuity of velocity and stresses at the subdomains interfaces.

The geometrical deformations are computed through a non-affine transfinite map and an empirical interpolation method has been used to perform a complete offline/online computational decoupling of the reduced basis problem. Results dealing with the complexity reduction and computational performances have been provided in comparison with classical FE techniques and classical RB method on two test cases of interest.

A three-dimensional application will be presented in Chapter 5 by dealing with a possible geometrical configuration suitable for cardiovascular problems and providing an efficient, accurate and real-time framework of application.

We point out that, in order to guarantee a very high versatility, being able to represent a wide range of possible fluidic networks, avoiding the computation of a coarse FE solution and, at the same time, the continuity of all the physical quantities involved, a further possible alternative methodology is presented in the following chapters. The latter is based on local basis functions enrichment at the interfaces or in the nearby surrounding area.

We remark that the a posteriori error bounds available for the classical RB method [26, 87] can be used during the local greedy procedures for the bases construction in each subdomain. During the online procedure the same framework can be adopted but considering global (on the whole domain) residual and stability constants lower bounds.

3 Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

In this chapter we present an alternative method to combine finite element method (FE), reduced basis method (RB) and domain decomposition techniques (DD) for solving parametric problems in computational domains, represented by networks made up by repetitive geometries (blocks). Each block can be considered as parametric deformation of a reference shape. The main idea behind this approach, that we call RDF method, is always to build a reduced basis space by computing (with classical FE techniques) once, locally, and for few reference domains, some representative solutions for different values of the parameters with also a joint set of different suitable conditions on the block boundaries. Once we have defined the local reduced basis spaces, we select a set of finite element functions in correspondence of a region of the domain containing the internal interfaces and we use these functions to enrich locally the reduced basis spaces. The global solution is then recovered by a Galerkin projection on the space built by the reduced basis functions and the finite element functions. The continuity of the global solution is assured by a classical domain decomposition approach.

In order to facilitate the explanation of the method and to explore the quality of some options that can be adopted, in this chapter we start with a simple Poisson model problem since we need to introduce several ingredients and this approach is quite different with respect to what is already available in literature (e.g. RBEM [58]). However the method is applicable to a general time-independent parametrized linear partial differential equation, the application of the method to the steady Stokes equations will be introduced in Chapter 4. Several initial results show the flexibility of this approach in which accuracy and computational time can be tuned by varying (i) the extension of the region involving the finite element functions, (ii) the number of reduced basis functions, (iii) the set of local BCs used to compute the reduced basis functions. We anticipate that the same problem of the local enrichment for the basis functions at the interfaces is under investigation also in the recent works by Patera et al. [22, 38] where they propose a static condensation method (i.e. *Lego*) for repetitive systems. In particular, they consider a standard static condensation formulation based on a "truth" finite element (FE) discretization of the global spatial domain. The static condensation process yields a relatively small Schur complement system for the degrees of freedom associated only with the internal interfaces of the domain (ports). However, the assembly of this system is very expensive due to the large number of local FE bubble solves required in each component: one local FE solve is required for each degree of freedom on each port. The key idea of the Static Condensation Reduced Basis Element (scrBE) method is to replace each of the local FE bubbles by their inexpensive reduced basis (RB) approximation tailored to the local parameter dependence; then these RB bubbles yield a RB approximate Schur complement system which may be assembled and solved much more rapidly than the corresponding FE Schur complement

system [20, 21, 22].

The idea of the RDF method is to solve a global reduced problem by using some local bases (comparable to the bubbles of the scRBE). The difference is that these bases are not computed for each degree of freedom of the internal interfaces and locally for the desired parameter value, but the interface profiles of these bases are interpreted as additional parameters (unphysical) to consider in the RB local problems. Finally, thanks to the local greedy selection, a set of local bases (selected by varying the physical and unphysical parameter) can be used to approximate the global solution for each desired physical parameters combination in the computational domain.

The computational effort required by this approach is slightly larger than that requested by using an online classical reduced basis approach on the whole domain. However this approach, as well as the RBHM presented in Chapter 2, decreases drastically the offline time to pre-compute the reduced basis subspaces thanks to the fact that the total number of parameters characterizing the problem at hand is split into smaller subsets for each reference block and, at the same time, it allows to deal with more complex and versatile networks and to increase the number of parameters considered. This method introduces some alternative ingredients with respect to the RBHM by replacing, for example, the necessity to compute a coarse FE global solution. After the presentation of the RDF method for both Poisson problem (in this chapter) and steady Stokes problem (in Chapter 4), a detailed comparison between the proposed methodologies (RBHM and RDF) and the already existing ones (RB and RBEM) will be discussed in Chapter 5. The contents of this chapter have been based on work [39].

3.1 Problem Setting

To illustrate the method, we consider the model problem of steady heat conduction in a thermal fin. This problem has previously been used as a test problem in the framework of the classical reduced basis method in [86] and of the reduced basis element method in [59].

The heat conduction problem described by a parametric Laplace problem in a domain composed by a combination of a repetitive geometry, is a suitable and simple framework to describe all the ingredients and the options of the method.

We consider the computational domain composed by the union of R non-overlapping subfins Ω_r , $r = 1, \dots, R$. Each block is constituted by a different material, with its own constant thermal conductivity that represents, through μ_1^r , the physical parameter of the equation. Moreover each block can be seen as a deformation of a reference one $\hat{\Omega}_r$ through a suitable affine map and a geometrical parameter μ_2^r , that defines the length of the fins of the blocks, so that $\Omega_r = \Omega_{\mu_2^r}$ and $\mathbf{x} = T(\hat{\mathbf{x}}, \mu_2^r)$, $\mathbf{x} \in \Omega_r$, $\hat{\mathbf{x}} \in \hat{\Omega}_r$, see Figure 3.1.

Figure 3.2 shows a possible configuration for the computational domain with $R = 4$. In this case there are 4 physical parameters $\boldsymbol{\mu}_1 = [\mu_1^1, \dots, \mu_1^4]$ and 4 geometrical parameters $\boldsymbol{\mu}_2 = [\mu_2^1, \dots, \mu_2^4]$, so that $\bar{\Omega}_{\boldsymbol{\mu}_2} = \cup_{r=1}^R \bar{\Omega}_r$. We assume that $\mu_1^i \in [0.1, 10]$ and $\mu_2^i \in [3, 9]$.

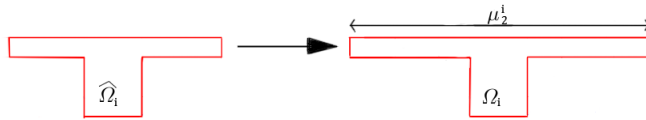


Figure 3.1: Scheme for the geometrical transformation of a single block.

We impose a non-zero uniform temperature at the bottom of the multiblock (Γ_{in}), zero temperature on the vertical surfaces of the spreaders and at the top of the domain (Γ_{out}), and zero heat flux (con-

servative) on the remaining boundaries of the fins (Γ). We are interested in studying the temperature distribution inside the whole computational domain by varying the number of blocks of the domain and the values of the parameters.

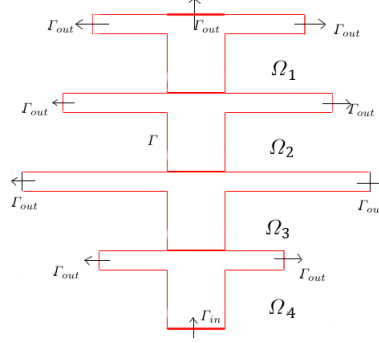


Figure 3.2: Computational domain composed by R=4 blocks.

The governing steady conduction problem for the temperature u in the thermal fin is:

$$\left\{ \begin{array}{ll} \mu_1 \Delta u = 0 & \text{in } \Omega_{\mu_2} \\ u = 1 & \text{on } \Gamma_{in} \\ u = 0 & \text{on } \Gamma_{out} \\ \frac{\partial u}{\partial n} = 0 & \text{on } \Gamma \end{array} \right. , \quad (3.1.1)$$

where $\mu_1 = \mu_1(\mathbf{x}) = \mu_1^i$ if $\mathbf{x} \in \Omega_{\mu_2^i}$ $i = 1, \dots, 4$.

Let us introduce a lift R_g of the Dirichlet BC on Γ_{in} . We define u as the sum of $u_0 \in V_0(\Omega) \equiv \{v : v \in V(\Omega) : v = 0 \text{ on } \Gamma_D = \Gamma_{in} \cup \Gamma_{out}\}$, and the lift R_g .

The weak formulation of the problem (3.1.1) reads as follows: find $u_0 \in V_0(\Omega)$ such that $\forall v \in V_0(\Omega)$:

$$a_{\mu}(u_0, v) = F_{\mu}(v), \quad (3.1.2)$$

where $\mu = (\mu_1, \mu_2)$ and

$$\begin{aligned} a_{\mu}(u_0, v) &= \int_{\Omega_{\mu_2}} \mu_1 \nabla u_0 \cdot \nabla v \, d\mathbf{x} = \sum_{r=1}^R \int_{\Omega_{\mu_2^r}} \mu_1^r \nabla u_0 \cdot \nabla v \, d\mathbf{x} \\ F_{\mu}(v) &= - \int_{\Omega_{\mu_2}} \mu_1 \nabla R_g \cdot \nabla v \, d\mathbf{x} = - \sum_{r=1}^R \int_{\Omega_{\mu_2^r}} \mu_1^r \nabla R_g \cdot \nabla v \, d\mathbf{x}. \end{aligned} \quad (3.1.3)$$

3.1.1 Geometrical parametrization of the problem

We define here the detailed geometrical parametrization introduced in Chapter 1, Section 1.2.3, for the case of the elliptic problem with affine parametric dependence. We define the integrals of (3.1.3) from the generic block Ω_r dependent on the parameter μ_2^r to the reference geometry $\hat{\Omega}_r$. Let us introduce a piecewise affine map $T(\hat{\mathbf{x}}, \mu_2^r) : \hat{\Omega}_r \rightarrow \Omega_r$ such that $\mathbf{x} = C_{\mu_2^r} \hat{\mathbf{x}} + \mathbf{c}_{\mu_2^r}$, as shown in Figure 3.3.

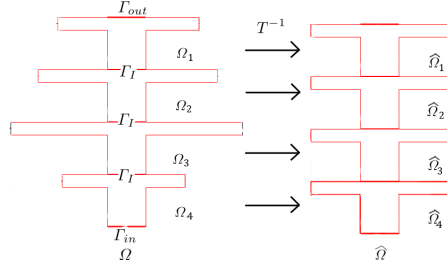


Figure 3.3: Geometrical scheme: from the deformed configuration to the reference blocks.

With this notation we have:

$$\begin{aligned}
 a_{\boldsymbol{\mu}}(u_0, v) &= \sum_{r=1}^R \int_{\Omega_r} \mu_1^r \nabla u_0 \cdot \nabla v \, d\mathbf{x} = \sum_{r=1}^R \int_{\hat{\Omega}_r} \mu_1^r C_{\mu_2^r}^{-T} \hat{\nabla} \hat{u}_0 \cdot C_{\mu_2^r}^{-T} \hat{\nabla} \hat{v} |\det C_{\mu_2^r}| \, d\hat{\mathbf{x}} \\
 &= \sum_{r=1}^R \int_{\hat{\Omega}_r} \mu_1^r \hat{\nabla} \hat{u}_0^T C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T} \hat{\nabla} \hat{v} |\det C_{\mu_2^r}| \, d\hat{\mathbf{x}} = \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{11}}_{\Theta_1^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{u}_0}{\partial \hat{x}_1} \frac{\partial \hat{v}}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{a_1^r(\hat{u}_0, \hat{v})} + \\
 &\quad \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{12}}_{\Theta_2^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{u}_0}{\partial \hat{x}_1} \frac{\partial \hat{v}}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{a_2^r(\hat{u}_0, \hat{v})} + \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{21}}_{\Theta_3^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{u}_0}{\partial \hat{x}_2} \frac{\partial \hat{v}}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{a_3^r(\hat{u}_0, \hat{v})} + \\
 &\quad \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{22}}_{\Theta_4^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{u}_0}{\partial \hat{x}_2} \frac{\partial \hat{v}}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{a_4^r(\hat{u}_0, \hat{v})} = \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\hat{u}_0, \hat{v}).
 \end{aligned} \tag{3.1.4}$$

The map $T(\hat{\mathbf{x}}, \mu_2^r)|_{\hat{\Omega}_r}$ is affine, therefore the matrix $(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}|$ depends only on μ_2^r and we denote with $\left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{lm}$ its generic element. With this notation it is possible to decouple the terms $\Theta_i^r(\boldsymbol{\mu})$ that depend on the parameters from the integrals defined on the references domain. The same procedure can be applied without relevant modifications to the right-hand-side of (3.1.2) leading to the definition of the linear functional $F_{\boldsymbol{\mu}}(v)$.

$$\begin{aligned}
 F_{\boldsymbol{\mu}}(v) &= - \sum_{r=1}^R \int_{\Omega_r} \mu_1^r \nabla R_g \cdot \nabla v \, d\mathbf{x} = - \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{11}}_{\Psi_1^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{R}_g}{\partial \hat{x}_1} \frac{\partial \hat{v}}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{F_1^r(\hat{v})} \\
 &\quad - \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{12}}_{\Psi_2^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{R}_g}{\partial \hat{x}_1} \frac{\partial \hat{v}}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{F_2^r(\hat{v})} - \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{21}}_{\Psi_3^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{R}_g}{\partial \hat{x}_2} \frac{\partial \hat{v}}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{F_3^r(\hat{v})} \\
 &\quad - \sum_{r=1}^R \underbrace{\mu_1^r \left[(C_{\mu_2^r}^{-1} C_{\mu_2^r}^{-T}) |\det C_{\mu_2^r}| \right]_{22}}_{\Psi_4^r(\boldsymbol{\mu})} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{R}_g}{\partial \hat{x}_2} \frac{\partial \hat{v}}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{F_4^r(\hat{v})} = - \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\hat{v}).
 \end{aligned} \tag{3.1.5}$$

We denote with $\bar{\Omega} = \cup_{r=1}^R \bar{\Omega}_r$ the reference domain of $\bar{\Omega} = \cup_{r=1}^R \bar{\Omega}_r$. Finally the Laplace problem reads: find $\hat{u}_0 \in V_0(\hat{\Omega})$ such that $\forall \hat{v} \in V_0(\hat{\Omega})$:

$$a_{\boldsymbol{\mu}}(\hat{u}_0, \hat{v}) = \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\hat{u}_0, \hat{v}) = - \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\hat{v}) = F_{\boldsymbol{\mu}}(\hat{v}). \quad (3.1.6)$$

With a little abuse of notation, from now, for sake of simplicity, we maintain the “hat” symbol only to characterize the reference domains.

3.2 Finite element approximation of the problem

The Galerkin method to numerically solve problem (3.1.6) consists in finding an approximated solution $u_h \in V_h$, where V_h is a set of subspaces of $V_0(\hat{\Omega})$, dependent on a positive parameter h , with finite dimension \mathcal{N} . The final solution \tilde{u} is then recovered by adding the lift function $\tilde{u} = u_h + Rg_h$. Therefore the approximated problem becomes: find $u_h \in V_h$ such that $\forall v_h \in V_h$

$$a_{\boldsymbol{\mu}}(u_h, v_h) = \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u_h, v_h) = - \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_h) = F_{\boldsymbol{\mu}}(v_h). \quad (3.2.1)$$

Problem (3.2.1) is usually called the Galerkin formulation of problem (3.1.6). If we denote with $\{\phi_j, j = 1, \dots, \mathcal{N}\}$ a basis of V_h and we project u_h on that basis we have:

$$u_h = \sum_{j=1}^{\mathcal{N}} u_j \phi_j, \quad (3.2.2)$$

where u_j are the projection coefficients of u_h on the basis of V_h . Problem (3.2.1) is therefore equivalent to: find $u_j, j = 1, \dots, \mathcal{N}$, such that for $i = 1, \dots, \mathcal{N}$

$$\sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \sum_{j=1}^{\mathcal{N}} u_j a_q^r(\phi_j, \phi_i) = - \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_i), \quad (3.2.3)$$

which leads to the linear system:

$$\mathbb{A} \mathbf{u} = \mathbb{F}, \quad (3.2.4)$$

where $[\mathbb{A}]_{ij} = \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j, \phi_i)$ and $[\mathbb{F}]_i = - \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_i)$. Approximating the resulting problem with the Finite Element Method consists in a particular choice for the subspace V_h . We consider a triangulation \mathcal{T}_h of $\hat{\Omega}$ and let K be the generic element of \mathcal{T}_h . In particular we use piecewise linear functions and we define:

$$V_h = X_h^1 \equiv \{v_h \in C^0(\hat{\Omega}) : v_h|_K \in \mathbb{P}_1 \quad \forall K \in \mathcal{T}_h\} \cap H_{0,\Gamma_D}^1(\hat{\Omega}). \quad (3.2.5)$$

3.3 The non-overlapping domain decomposition method for FE

Domain decomposition methods may be used as an efficient paradigm to solve PDEs on parallel computing platforms, see [3,4]. The idea is to subdivide the computational domain into a series of subdomains where local problems have to be solved. Here we focus on the so called non-overlapping

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

domain decomposition method.

We have seen, in the previous section, that problem (3.1.1) defined in Ω can be rewritten, through a suitable map and its geometrical tensors, in the reference domain $\hat{\Omega}$. The fact that the PDE is parametrized is not relevant for the domain decomposition definition, therefore we suppose at the moment, for sake of simplicity, that the problem does not depend on any parameter, that it is defined on $\hat{\Omega}$ and we will omit to write the parametric dependence for the remaining part of this section.

We recall the domain decomposition method for the case of a non-overlapping partition of $\hat{\Omega}$ into two subdomains $\hat{\Omega}_1$ and $\hat{\Omega}_2$. Let $\hat{\Gamma}_I$ be the interface separating $\hat{\Omega}_1$ and $\hat{\Omega}_2$, $\hat{\Gamma}_I = \partial\hat{\Omega}_1 \cap \partial\hat{\Omega}_2$, while $\hat{\Gamma}_1 = \partial\hat{\Omega}_1 \setminus (\hat{\Gamma}_I \cup \hat{\Gamma}_{in})$ and $\hat{\Gamma}_2 = \partial\hat{\Omega}_2 \setminus (\hat{\Gamma}_I \cup \hat{\Gamma}_{out})$. Then the solution u of (3.1.1) is such that $u|_{\hat{\Omega}_i} = u_i$ for $i = 1, 2$ where u_i is the solution of the problem:

$$\left\{ \begin{array}{ll} \Delta u_1 = 0 & \text{in } \hat{\Omega}_1, \\ u_1 = 1 & \text{on } \hat{\Gamma}_{in}, \\ \frac{\partial u_1}{\partial \mathbf{n}_1} = 0 & \text{on } \hat{\Gamma}_1, \\ \Delta u_2 = 0 & \text{in } \hat{\Omega}_2, \\ u_2 = 0 & \text{on } \hat{\Gamma}_{out}, \\ \frac{\partial u_2}{\partial \mathbf{n}_2} = 0 & \text{on } \hat{\Gamma}_2, \\ u_1 = u_2 & \text{on } \hat{\Gamma}_I, \\ \frac{\partial u_1}{\partial \mathbf{n}_1} = -\frac{\partial u_2}{\partial \mathbf{n}_2} & \text{on } \hat{\Gamma}_I, \end{array} \right. \quad (3.3.1)$$

where \mathbf{n}_i is the outgoing normal unit vector to $\hat{\Omega}_i$, in Figure 3.4 a geometrical scheme of the domain is represented.

In order to describe the finite element approximation of the problem in a domain decomposition setting, we introduce the partition of the nodes of the triangulation as represented in Figure 3.4:

- $\{\mathbf{x}_i^1, 1 \leq i \leq N_1\}$ denote the nodes in the subdomain $\overline{\hat{\Omega}_1} \setminus \hat{\Gamma}_I$,
- $\{\mathbf{x}_j^2, 1 \leq j \leq N_2\}$ denote the nodes in the subdomain $\overline{\hat{\Omega}_2} \setminus \hat{\Gamma}_I$,
- $\{\mathbf{x}_k^\Gamma, 1 \leq k \leq N_\Gamma\}$ denote the nodes on the interface $\hat{\Gamma}_I$,

where $N_1 + N_2 + N_\Gamma = \mathcal{N}$ and we split the basis functions accordingly:

- ϕ_i^1 are the Lagrange functions associated to the nodes \mathbf{x}_i^1 ,
- ϕ_j^2 are the Lagrange functions associated to the nodes \mathbf{x}_j^2 ,
- ϕ_k^Γ are the Lagrange functions associated to the nodes \mathbf{x}_k^Γ .

The Galerkin approximation of the problem (3.3.1) on $\hat{\Omega}$ is find $u_h \in V_h$:

$$\left\{ \begin{array}{ll} a(u_h, \phi_i^1) = F(\phi_i^1) & \forall i = 1, \dots, N_1, \\ a(u_h, \phi_j^2) = F(\phi_j^2) & \forall j = 1, \dots, N_2, \\ a(u_h, \phi_k^\Gamma) = F(\phi_k^\Gamma) & \forall k = 1, \dots, N_\Gamma, \end{array} \right. \quad (3.3.2)$$

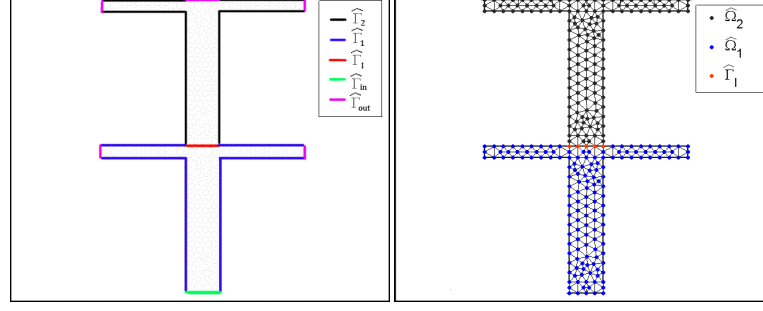


Figure 3.4: Computational domain decomposed by R=2 thermal blocks.

where $a(\cdot, \cdot)$ and $F(\cdot)$ are the bilinear and linear forms defined in (3.1.3) without the parametric dependence. We introduce the linear and bilinear form on $\hat{\Omega}_i, \forall v, w \in V_h, i = 1, 2$:

$$\begin{aligned} a_i(v, w) &= \int_{\hat{\Omega}_i} \nabla v \cdot \nabla w \, d\mathbf{x}, \\ F_i(v) &= - \int_{\hat{\Omega}_i} \nabla R_g \cdot \nabla v \, d\mathbf{x}. \end{aligned} \quad (3.3.3)$$

Problem (3.3.2) can be written as: find $u_h \in V_h$ such that

$$\begin{cases} a_1(u_h, \phi_i^1) = F_1(\phi_i^1) & \forall i = 1, \dots, N_1, \\ a_2(u_h, \phi_j^2) = F_2(\phi_j^2) & \forall j = 1, \dots, N_2, \\ a_1(u_h, \phi_k^\Gamma|_{\hat{\Omega}_1}) + a_2(u_h, \phi_k^\Gamma|_{\hat{\Omega}_2}), \\ = F_1(\phi_k^\Gamma|_{\hat{\Omega}_1}) + F_2(\phi_k^\Gamma|_{\hat{\Omega}_2}) & \forall k = 1, \dots, N_\Gamma. \end{cases} \quad (3.3.4)$$

Problem (3.3.4) corresponds to the finite element approximation of the multi-domain formulation (3.3.1). By construction, u_h is continuous on $\hat{\Gamma}_I$, therefore the discretized solution automatically satisfies the continuity condition of the multi-domain formulation.

We can consider the following decomposition of u_h :

$$u_h(\mathbf{x}) = \sum_{j=1}^{N_1} u_h(\mathbf{x}_j^1) \phi_j^1(\mathbf{x}) + \sum_{j=1}^{N_2} u_h(\mathbf{x}_j^2) \phi_j^2(\mathbf{x}) + \sum_{j=1}^{N_\Gamma} u_h(\mathbf{x}_j^\Gamma) \phi_j^\Gamma(\mathbf{x}), \quad (3.3.5)$$

and by exploiting this decomposition in problem (3.3.4), we obtain:

$$\begin{cases} \sum_{j=1}^{N_1} u_h(\mathbf{x}_j^1) a_1(\phi_j^1, \phi_i^1) + \sum_{j=1}^{N_\Gamma} u_h(\mathbf{x}_j^\Gamma) a_1(\phi_j^\Gamma, \phi_i^1) = F_1(\phi_i^1) & \forall i = 1, \dots, N_1 \\ \sum_{j=1}^{N_2} u_h(\mathbf{x}_j^2) a_2(\phi_j^2, \phi_i^2) + \sum_{j=1}^{N_\Gamma} u_h(\mathbf{x}_j^\Gamma) a_2(\phi_j^\Gamma, \phi_i^2) = F_2(\phi_i^2) & \forall j = 1, \dots, N_2 \\ \sum_{j=1}^{N_1} u_h(\mathbf{x}_j^1) a_1(\phi_j^1, \phi_k^\Gamma) + \sum_{j=1}^{N_2} u_h(\mathbf{x}_j^2) a_2(\phi_j^2, \phi_k^\Gamma) + \\ \sum_{j=1}^{N_\Gamma} u_h(\mathbf{x}_j^\Gamma) [a_1(\phi_j^\Gamma, \phi_k^\Gamma) + a_2(\phi_j^\Gamma, \phi_k^\Gamma)] = F_1(\phi_k^\Gamma) + F_2(\phi_k^\Gamma) & \forall k = 1, \dots, N_\Gamma, \end{cases} \quad (3.3.6)$$

so that the algebraic form is:

$$\begin{cases} \mathbb{A}_1 \mathbf{u}_1 + \mathbb{A}_{1\Gamma} \boldsymbol{\lambda} = \mathbb{F}_1 \\ \mathbb{A}_2 \mathbf{u}_2 + \mathbb{A}_{2\Gamma} \boldsymbol{\lambda} = \mathbb{F}_2 \\ \mathbb{A}_{\Gamma 1} \mathbf{u}_1 + \mathbb{A}_{\Gamma 2} \mathbf{u}_2 + [\mathbb{A}_{\Gamma\Gamma}^1 + \mathbb{A}_{\Gamma\Gamma}^2] \boldsymbol{\lambda} = \mathbb{F}_\Gamma^1 + \mathbb{F}_\Gamma^2 \end{cases} \quad (3.3.7)$$

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

where $[\mathbf{u}_k]_j = u_h(\mathbf{x}_j^k)$, $[\boldsymbol{\lambda}]_j = u_h(\mathbf{x}_j^\Gamma)$, $[\mathbb{A}_k]_{i,j} = a_k(\phi_j^k, \phi_i^k)$, $[\mathbb{A}_{k\Gamma}]_{i,j} = a_k(\phi_j^k, \phi_i^\Gamma)$, $[\mathbb{A}_{\Gamma k}]_{i,j} = a_k(\phi_j^\Gamma, \phi_i^k)$, $[\mathbb{A}_{\Gamma\Gamma}^k]_{i,j} = a_k(\phi_j^\Gamma, \phi_i^\Gamma)$, $[\mathbb{F}_k]_j = F_k(\phi_j^k)$, $[\mathbb{F}_\Gamma]_j = F_k(\phi_j^\Gamma)$, $k = 1, 2$. The final FE linear system is:

$$\begin{bmatrix} \mathbb{A}_1 & 0 & \mathbb{A}_{1\Gamma} \\ 0 & \mathbb{A}_2 & \mathbb{A}_{2\Gamma} \\ \mathbb{A}_{\Gamma 1} & \mathbb{A}_{\Gamma 2} & \mathbb{A}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbb{F}_1 \\ \mathbb{F}_2 \\ \mathbb{F}_\Gamma \end{bmatrix}. \quad (3.3.8)$$

Here $\mathbb{A}_{\Gamma\Gamma}$ and \mathbf{f}_Γ are split into the parts coming from the integration of the variational form and the linear functional on $\hat{\Omega}_1$ and $\hat{\Omega}_2$, yielding $\mathbb{A}_{\Gamma\Gamma} = \mathbb{A}_{\Gamma\Gamma}^1 + \mathbb{A}_{\Gamma\Gamma}^2$ and $\mathbb{F}_\Gamma = \mathbb{F}_\Gamma^1 + \mathbb{F}_\Gamma^2$.

In the general case in which the thermal fin is partitioned into an arbitrary number R of subdomains $\hat{\Omega}_r$, $R-1$ internal interfaces $\hat{\Gamma}_r = \hat{\Omega}_r \cap \hat{\Omega}_{r+1}$, $r = 1, \dots, R-1$ and a parametric dependence in the problem, we find the following linear system:

$$\begin{bmatrix} \mathbb{A}_{rr} & \mathbb{A}_{r\Gamma} \\ \mathbb{A}_{r\Gamma}^T & \mathbb{A}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbb{F}_{rr} \\ \mathbb{F}_\Gamma \end{bmatrix}, \quad (3.3.9)$$

where

$$\mathbb{A}_{rr} = \begin{bmatrix} \mathbb{A}_1 & & & \\ & \mathbb{A}_2 & & \\ & & \ddots & \\ & & & \mathbb{A}_R \end{bmatrix}, \mathbb{A}_{r\Gamma} = \begin{bmatrix} \mathbb{A}_{1\Gamma_1} & 0 & & & \\ \mathbb{A}_{2\Gamma_1} & \mathbb{A}_{2\Gamma_2} & 0 & & \\ 0 & \mathbb{A}_{3\Gamma_2} & \mathbb{A}_{3\Gamma_3} & 0 & \\ & \ddots & \ddots & \ddots & 0 \\ & & 0 & \mathbb{A}_{R-1\Gamma_{R-2}} & \mathbb{A}_{R-1\Gamma_{R-1}} \\ & & & 0 & \mathbb{A}_{R\Gamma_{R-1}} \end{bmatrix},$$

$$\mathbb{A}_{\Gamma\Gamma} = \begin{bmatrix} \mathbb{A}_{\Gamma_1\Gamma_1}^1 + \mathbb{A}_{\Gamma_1\Gamma_1}^2 & & & \\ & \mathbb{A}_{\Gamma_2\Gamma_2}^2 + \mathbb{A}_{\Gamma_2\Gamma_2}^3 & & \\ & & \ddots & \\ & & & \mathbb{A}_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + \mathbb{A}_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{R-1} \end{bmatrix}, \mathbb{F}_{rr} = \begin{bmatrix} \mathbb{F}_1 \\ \vdots \\ \mathbb{F}_R \end{bmatrix}, \mathbb{F}_\Gamma = \begin{bmatrix} \mathbb{F}_{\Gamma_1}^1 + \mathbb{F}_{\Gamma_1}^2 \\ \vdots \\ \mathbb{F}_{\Gamma_{R-1}}^{R-1} + \mathbb{F}_{\Gamma_{R-1}}^R \end{bmatrix}.$$

By recalling the affine decompositions (3.1.4) and (3.1.5) (and by considering a parametrized problem), we define for $r = 1, \dots, R$ and for $k = 1, \dots, R - 1$:

$$\begin{aligned} [\mathbf{u}_r]_j &= u_h(\mathbf{x}_j^r), \quad [\mathbb{A}_r]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^r), \quad [\mathbb{F}_r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^r), \quad i, j = 1, \dots, N_j \\ [\mathbb{A}_{r\Gamma_k}]_{i,j} &= \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^{\Gamma_k}), \quad j = 1, \dots, N_j, \quad i = 1, \dots, N_{\Gamma_r} \\ [\mathbb{A}_{\Gamma_k\Gamma_r}^r]_{i,j} &= \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_k}, \phi_i^{\Gamma_r}), \quad [\boldsymbol{\lambda}_k]_j = u_h(\mathbf{x}_j^{\Gamma_k}), \quad [\mathbb{F}_{\Gamma_k}^r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_k}), \quad i, j = 1, \dots, N_{\Gamma_r} \end{aligned}$$

where, N_j is the number of Lagrangian bases on $\widehat{\Omega}_j \setminus \widehat{\Gamma}_k$ and N_{Γ_k} the number of Lagrangian bases on $\widehat{\Gamma}_k$. As before, we split the matrix A into 4 sub-matrices $\mathbb{A}_{\Gamma\Gamma}$, $\mathbb{A}_{\Gamma\Gamma}$, $\mathbb{A}_{\Gamma\Gamma}$ and $\mathbb{A}_{\Gamma\Gamma} = \mathbb{A}_{\Gamma\Gamma}^T$ containing, respectively, the parts coming from the integration of the variational form on the subdomains, on the internal interface and partially on the internal interfaces and partially on the subdomains.

3.3.1 The Steklov-Poincaré operator for the FE problem

We want to introduce here the Steklov-Poincaré operator that represents a fundamental part of the classical domain decomposition technique. This is not currently considered in the RDF method, but it can be included in the methodology and represent a possible option to extend it.

Recalling problem (3.3.1), with two subdomains and without parametric dependence, we denote $\lambda = u_h|_{\widehat{\Gamma}_I}$ the unknown value of u_h on $\widehat{\Gamma}_I$ and $u_i = u_h|_{\widehat{\Omega}_i}$. Let us split u_i as follows:

$$u_i = u_i^0 + u_i^{\widehat{\Gamma}_I} \quad (3.3.10)$$

where u_i^0 and $u_i^{\widehat{\Gamma}_I}$, $i = 1, 2$, represent the solutions of the following problems:

$$\left\{ \begin{array}{l} \mu_1^1 \Delta u_1^0 = 0 \quad \text{in } \widehat{\Omega}_1 \\ u_1^0 = 1 \quad \text{on } \widehat{\Gamma}_{in} \\ \frac{\partial u_1}{\partial \mathbf{n}_1} = 0 \quad \text{on } \widehat{\Gamma}_1 \\ u_1^0 = 0 \quad \text{on } \widehat{\Gamma}_I \end{array} \right\}, \left\{ \begin{array}{l} \mu_1^2 \Delta u_2^0 = 0 \quad \text{in } \widehat{\Omega}_2 \\ u_2^0 = 0 \quad \text{on } \widehat{\Gamma}_{out} \\ \frac{\partial u_2}{\partial \mathbf{n}_2} = 0 \quad \text{on } \widehat{\Gamma}_2 \\ u_2^0 = 0 \quad \text{on } \widehat{\Gamma}_I \end{array} \right\}, \left\{ \begin{array}{l} \mu_1^i \Delta u_i^{\widehat{\Gamma}_I} = 0 \quad \text{in } \widehat{\Omega}_i \\ u_i^{\widehat{\Gamma}_I} = 0 \quad \text{on } \partial \widehat{\Omega}_i \setminus (\Gamma_i \cup \Gamma_I) \\ u_i^{\widehat{\Gamma}_I} = \lambda \quad \text{on } \widehat{\Gamma}_I \\ \frac{\partial u_i^{\widehat{\Gamma}_I}}{\partial \mathbf{n}_i} = 0 \quad \text{on } \widehat{\Gamma}_i \end{array} \right\}. \quad (3.3.11)$$

As the functions u_i^0 depend only on the BC data and the operators we may solve the problems formally obtaining $u_i^0 = G_i([g])$ where g is the function describing the BCs on $\widehat{\Gamma}_{in}$, $\widehat{\Gamma}_{out}$ and $\widehat{\Gamma}_i$, G_i is a linear continuous operator. On the other side we have that $u_i^{\widehat{\Gamma}_I}$ depends only on the value of λ . Therefore we may write formally $u_i^{\widehat{\Gamma}_I} = H_i(\lambda)$ where H_i is the extension operator of the considered problem. Finally, by imposing the continuity of the conormal derivative on the internal interface, we have:

$$\begin{aligned} \Leftrightarrow \quad \frac{\frac{\partial u_1}{\partial \mathbf{n}_1}}{\frac{\partial (G_1(g) + H_1(\lambda))}{\partial \mathbf{n}_1}} &= - \frac{\frac{\partial u_2}{\partial \mathbf{n}_2}}{\frac{\partial (G_2(g) + H_2(\lambda))}{\partial \mathbf{n}_2}} \quad \text{on } \widehat{\Gamma}_I, \\ \Leftrightarrow \quad \left(\frac{\partial H_1}{\partial \mathbf{n}_1} + \frac{\partial H_2}{\partial \mathbf{n}_2} \right)(\lambda) &= - \left(\frac{\partial G_1}{\partial \mathbf{n}_1} + \frac{\partial G_2}{\partial \mathbf{n}_2} \right)(g) \quad \text{on } \widehat{\Gamma}_I, \end{aligned} \quad (3.3.12)$$

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

which represents the so-called *Steklov-Poincaré* equation:

$$S\lambda = \chi \quad \text{on } \hat{\Gamma}_I. \quad (3.3.13)$$

$\chi = -\sum_i \frac{\partial}{\partial n_i} G_i g$ is a linear functional that depends on g and $S = \sum_i S_i$ is the so called local *Steklov-Poincaré* pseudo-differential operators (Dirichlet to Neumann map):

$$S_i : \eta \rightarrow S_i \eta = \frac{\partial H_i(\eta)}{\partial n_i}. \quad (3.3.14)$$

Many iterative *Domain Decomposition* schemes like Dirichlet-Neumann, Neumann-Neumann or Robin-Robin may be seen as *preconditioned Richardson* algorithms to solve the *Steklov-Poincaré* interface equation [92, 75]. By discretizing the problem with the finite element method the Steklov-Poincaré interface equation becomes an algebraic system for the interface values.

By considering the linear system (3.3.8), we can obtain \mathbf{u}_1 and \mathbf{u}_2 as:

$$\mathbf{u}_1 = \mathbb{A}_1^{-1}(\mathbb{F}_1 - \mathbb{A}_{1\Gamma}\lambda),$$

$$\mathbf{u}_2 = \mathbb{A}_2^{-1}(\mathbb{F}_2 - \mathbb{A}_{2\Gamma}\lambda),$$

and by considering these variables into the equation for λ we obtain:

$$\sum_{i=1}^2 \left(\mathbb{A}_{\Gamma\Gamma}^{(i)} - \mathbb{A}_{\Gamma\Gamma_i} \mathbb{A}_i^{-1} \mathbb{A}_{i\Gamma} \right) \lambda = \mathbb{F}_{\hat{\Gamma}_I} - \sum_{i=1}^2 \left(\mathbb{A}_{\Gamma\Gamma_i} \mathbb{A}_i^{-1} \mathbb{F}_i \right). \quad (3.3.15)$$

Note that even if (3.3.15) is derived in the case of two subdomains, it can be extended to an arbitrary number of subdomains¹.

By defining $\Sigma = \sum_i \Sigma_i$, where $\Sigma_i = \mathbb{A}_{\Gamma\Gamma}^{(i)} - \mathbb{A}_{\Gamma\Gamma_i} \mathbb{A}_i^{-1} \mathbb{A}_{i\Gamma}$ and χ_Γ the right hand side of equation (3.3.15) we obtain the discrete counterpart of the Steklov-Poincaré equation:

$$\Sigma \lambda = \chi_\Gamma, \quad (3.3.16)$$

where Σ is called *Schur complement* of \mathbb{A} with respect to λ and represents the approximation of S . Solving the Schur complement system iteratively with the preconditioned Richardson method is equivalent to use the classical iterative domain decomposition strategies. In particular, in the case of two subdomains:

- preconditioning with $P = \Sigma_2$ corresponds to the Dirichlet-Neumann scheme;
- preconditioning with $P = \Sigma_1$ corresponds to the Neumann-Dirichlet scheme;
- preconditioning with $P = (\sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1})^{-1}$ where σ_1 and σ_2 are positive acceleration parameters corresponds to the Neumann-Neumann scheme;
- preconditioning with $P = (\gamma_1 + \gamma_2)^{-1}(\gamma_1 I + \Sigma_1)(\gamma_2 I + \Sigma_2)$ where γ_1 and γ_2 are positive acceleration parameters corresponds to the Robin-Robin scheme.

We refer to [75] for more details.

¹The matrices \mathbb{A}_i have to be non-singular.

3.4 The Reduced basis-Domain decomposition-Finite element method (RDF)

We propose now a new strategy to couple the finite element method and the reduced basis method in the framework of non-overlapping domain decomposition. The proposed method consists in splitting the problem (3.1.1) into subproblems by following the DD approach of Section 3.3.

Let us consider the two domains case for an introductory description. The described DD approach, that we denote as the *full order model*, is obtained as a Galerkin projection of problem (3.3.1) onto a finite element subspace V_h defined as follows:

$$V_h = V_{FE}^1 \oplus V_{FE}^2 \oplus V_{FE}^\Gamma \quad (3.4.1)$$

where

$$V_{FE}^j = \text{span}\{\phi_i^j, i = 1, \dots, N_j\}, j = 1, 2, \quad V_{FE}^\Gamma = \text{span}\{\phi_k^\Gamma, k = 1, \dots, N_\Gamma\}.$$

The idea of the RDF method is to consider a reduced order model to solve the problem (3.3.1). The solution is found through a Galerkin projection of (3.3.1) on a space built by some proper precomputed functions (RB basis functions) and a subspace of the FE space V_h , more particularly, the subspace built by the FE functions associated to the nodes belonging to the internal interface $\hat{\Gamma}_I$. In other words, we replace V_h defined in (3.4.1) by the following composed space:

$$V_{RDF} = V_{RB}^1 \oplus V_{RB}^2 \oplus V_{FE}^\Gamma, \quad (3.4.2)$$

with V_{RB}^i RB spaces in $\hat{\Omega}_i, i = 1, 2$, yielding a distributed reduced order model (ROM) in $\hat{\Omega}_1$ and $\hat{\Omega}_2$ and a local full order model on $\hat{\Gamma}_I$.

A possible extension of the method consists in considering a larger subset of the FE space, by including also the FE functions associated to a larger number of nodes of the domain. This case is introduced in Section 3.4.2.

The main difference with respect to the other RB approaches coupled with DD techniques (i.e. RBEM [58], RBHM [40]) is that here we do not need further equations, such as those involving the Lagrange multipliers for imposing the continuity conditions, but we maintain the same equations of the problem involving the interface nodes that automatically ensure the continuity of the solution.

There are some ingredients of the RB method regarding the selection of the local snapshots, such as the local greedy algorithm, that represent a common paradigm for the RBEM, the RBHM and the RDF method. We address to the Section 5.4 for a more accurate comparison between the methods. As it will be deeply discussed in the following sections, a fundamental topic of the RDF approach is represented by the introduction of local parametric BCs imposed in the local problems defining the reduced basis functions.

3.4.1 FE region defined by the internal interfaces

Let us consider the problem (3.1.1), with the effective dependence on the parameter vector μ . As before, we consider the computational domain Ω composed by R non-overlapped subdomains Ω_r , we denote with Γ_I^r the internal interface between Ω_r and Ω_{r+1} . Each subdomain Ω_r can be seen as deformations of reference subdomains $\hat{\Omega}_r$ through the map T_r and $\hat{\Gamma}_I^r$ represents the internal interface between $\hat{\Omega}_r$ and $\hat{\Omega}_{r+1}$. Moreover, the reference subdomains $\hat{\Omega}_r$ may be interpreted as translation of few reference shapes Λ_k through the map \mathcal{T}_{kr} , such that $\hat{\mathbf{x}} = \mathcal{T}_{kr}(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}} + \tau_r, \tilde{\mathbf{x}} \in \Lambda_k, \hat{\mathbf{x}} \in \hat{\Omega}_r$, Figure 3.5 shows an example of these transformations.

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

In the particular case of the thermal fin we have only one reference shape Λ and we consider the map $\mathcal{T}_r : \Lambda \rightarrow \hat{\Omega}_r$.

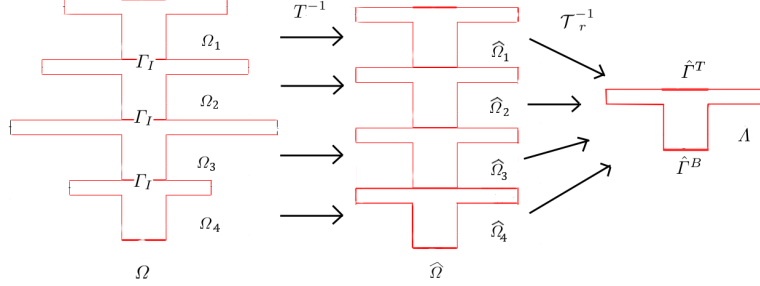


Figure 3.5: Computational domain Ω composed by $R=4$ deformed blocks Ω_r (left) and reference domain $\hat{\Omega}$ composed by $R=4$ reference blocks $\hat{\Omega}_r$ (center) and reference shape Λ (right).

We recall that with a little abuse of notation, due to the fact that we do not deal with the computational domain Ω , we are maintaining the “hat” symbol only to characterize the reference domains $\hat{\Omega}_r$. We define the reduced basis space V_{RB} , built by some solution v_i of N local heat conduction problems on the reference shape Λ .

$$V_{RB} = \{v_i, i = 1, \dots, N\}. \quad (3.4.3)$$

The theoretical description regarding the strategy adopted to define the local heat conduction problems will follow in Section 3.4.3. From now we denote their solutions as RB functions.

The functions v_i are defined locally on the mesh of Λ , we map them into the mesh of $\hat{\Omega}_r$ through the map \mathcal{T}_r and extend them equal to zero on the nodes belonging to $\hat{\Omega}_l, l \neq r$ and on the nodes belonging to the internal interfaces $\hat{\Omega}_{FE} = \cup_{r=1}^{R-1} \hat{\Gamma}_r$, so that for $r = 1, \dots, R$ and $i = 1, \dots, N$:

$$v_i^r(\mathbf{x}_j^r) = v_i(\mathcal{T}_r(\hat{\mathbf{x}}_j^r)), j = 1, \dots, N_j, \quad v_i^r(\mathbf{x}_k^\Gamma) = 0, k = 1, \dots, N_{\Gamma_k}, \quad v_i^r(\mathbf{x}_l^l) = 0, l \neq r, j = 1, \dots, N_l. \quad (3.4.4)$$

Figure 3.6 shows some particular functions $v_i^r, r = 1, \dots, 4$ associated to the solutions v_i of local heat conduction problems on Λ by imposing non-homogeneous constant Dirichlet BCs on $\hat{\Gamma}^T$ and $\hat{\Gamma}^B$ and then defined onto the subdomains $\hat{\Omega}_r$ as described in (3.4.4).

We address to Section 3.6.1 for a detailed numerical description on the way the local BCs are set in the local problems together with the representation of further local solutions presenting other interface profiles.

Then, depending on the BCs of the global problem that we have to solve we set: $v_i^r(\mathbf{x}) = 0, \forall \mathbf{x} \in \Gamma_D$, where Γ_D denotes the boundary of Ω with Dirichlet BCs.

By considering problem (3.1.1), we consider $\Gamma_D = \Gamma_{in} \cup \Gamma_{out}$. The functions v_i^r are linearly defined between the nodes of the mesh in Ω , such that $v_i \in V_h$ defined in (3.2.5).

We associate to each $\hat{\Omega}_r$ the following space:

$$V_{RB}^r = \{v_i^r, i = 1, \dots, N\}, r = 1, \dots, R. \quad (3.4.5)$$

²In general, in the case of K reference shapes, we define K local reduced basis spaces build by N_k reduced basis functions.

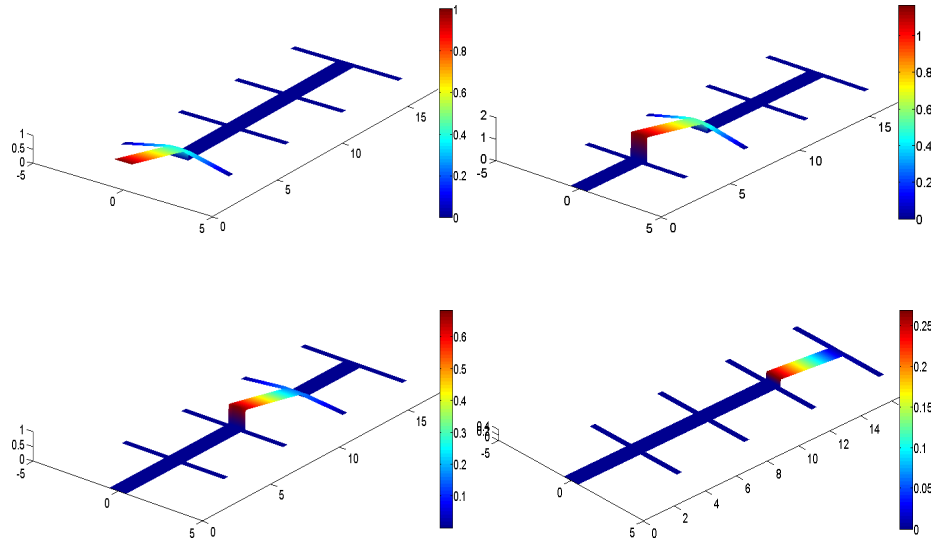


Figure 3.6: Example of RB functions associated to each subdomain $\hat{\Omega}_r, r = 1, \dots, 4$.

In this particular case, we need only one reference domain and the number of reduced basis functions computed on it is N , consequently the number of RB bases defined in each subdomain is the same for $r = 1, \dots, R$.

We consider now a subspace of the finite element space (3.2.5) composed by the finite element functions corresponding to the nodes $\mathbf{x}_j^{\Gamma_r}$ belonging to the internal interfaces $\hat{\Gamma}_r, \mathbf{x}_j^{\Gamma_r} \in \hat{\Omega}_{FE}$:

$$V_{FE}^{\Gamma} \equiv \{\phi_j^{\Gamma_r} \in V_h, j = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1\}, \quad (3.4.6)$$

where N_{Γ_r} is the number of nodes on the internal interface $\hat{\Gamma}_r$, we denote N_{Γ} as $N_{\Gamma} = \sum_{r=1}^{R-1} N_{\Gamma_r}$.

Figure 3.7 shows in the same plot the FE functions associated to each node belonging to the internal interfaces. In order to be able to graphically visualize all the functions involved in the considered FE subspace, we used a very coarse mesh that deals only with 15 interface nodes. We denote these selected functions as FE functions.

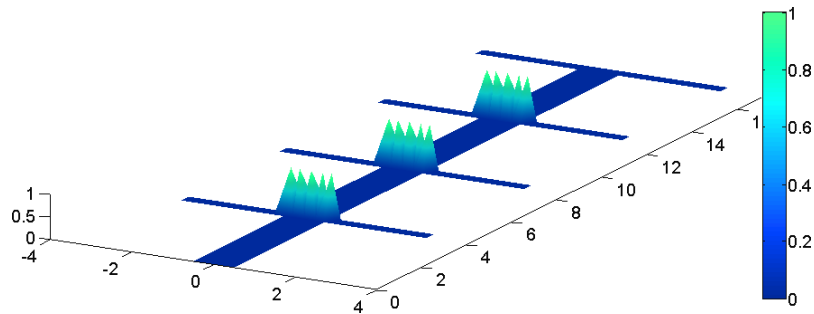


Figure 3.7: Finite element functions associated to every node belonging to the internal interfaces $\hat{\Gamma}_r$ and plotted in the same graph.

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

Finally we define the space:

$$V_{RDF} = V_{RB}^1 \oplus \cdots \oplus V_{RB}^R \oplus V_{FE}^\Gamma. \quad (3.4.7)$$

We recall now the same procedure of Section 3.3 and the parametric affine decomposition of the problem (3.1.4) and (3.1.5), by considering R subdomains and a K reference shapes, so that the number of local reduced basis functions on each $\hat{\Omega}_r$ can be different and depends on the reference shape, it is denoted by N_k , where $k = k(r)$.

We classify the basis functions as follows:

- $v_j^r, j = 1, \dots, N_k$ are the RB functions associated to the domain $\hat{\Omega}_r, r = 1, \dots, R$, as defined in (3.4.4) (an example of RB functions is represented in Figure 3.6, $r = 1, \dots, 4$);
- $\phi_j^{\Gamma_r}$ are the FE functions associated to the nodes $\mathbf{x}_j^{\Gamma_r} \in \hat{\Gamma}_r, j = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1$ (represented in Figure 3.7).

The problem (3.1.6) can be read as find $u(\boldsymbol{\mu}) \in V$:

$$\begin{cases} \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), v_i^r) = \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) & i = 1, \dots, RN_k, \\ \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), \phi_j^{\Gamma_r}) = \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_r}) & j = 1, \dots, N_{\Gamma}. \end{cases} \quad (3.4.8)$$

Problem (3.4.8) can be written as decomposed in the subdomains: find $u(\boldsymbol{\mu}) \in V$ such that

$$\begin{cases} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), v_i^r) = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) & i = 1, \dots, N_k, r = 1, \dots, R, \\ \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), \phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Theta_q^{r+1}(\boldsymbol{\mu}) a_q^{r+1}(u(\boldsymbol{\mu}), \phi_j^{\Gamma_r}) \\ = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Psi_q^{r+1}(\boldsymbol{\mu}) F_q^{r+1}(\phi_j^{\Gamma_r}) & j = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1. \end{cases} \quad (3.4.9)$$

The approximate solution $u(\boldsymbol{\mu})$ is defined in V and we can write it as linear combination of the basis functions of V :

$$u(\mathbf{x}, \boldsymbol{\mu}) = \sum_{r=1}^R \sum_{j=1}^{N_k} u_j^r(\boldsymbol{\mu}) v_j^r(\mathbf{x}) + \sum_{r=1}^{R-1} \sum_{j=1}^{N_{\Gamma_r}} \lambda_j(\boldsymbol{\mu}) \phi_j^{\Gamma_r}(\mathbf{x}). \quad (3.4.10)$$

Problem (3.4.9) becomes:

$$\left\{ \begin{array}{l} \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^1(\boldsymbol{\mu}) u_j^1(\boldsymbol{\mu}) a_q^1(v_j^1, v_i^1) + \sum_{j=1}^{N_{\Gamma_1}} \sum_{q=1}^4 \Theta_q^1(\boldsymbol{\mu}) \lambda_j(\boldsymbol{\mu}) a_q^1(\phi_j^{\Gamma_1}, v_i^1) = \\ \sum_{q=1}^4 \Psi_q^1(\boldsymbol{\mu}) F_q^1(v_i^1) \quad i = 1, \dots, N_k, \\ \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) u_j^r(\boldsymbol{\mu}) a_q^r(v_j^r, v_i^r) + \sum_{j=1}^{N_{\Gamma_{r-1}}} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_{r-1}}, v_i^r) = \\ \sum_{j=1}^{N_{\Gamma_r}} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_r}, v_i^r) + \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) \quad i = 1, \dots, N_k, r = 2, \dots, R-1, \\ \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^R(\boldsymbol{\mu}) u_j^R(\boldsymbol{\mu}) a_q^R(v_j^R, v_i^R) + \sum_{j=1}^{N_{\Gamma_{R-1}}} \sum_{q=1}^4 \Theta_q^R(\boldsymbol{\mu}) \lambda_j(\boldsymbol{\mu}) a_q^R(\phi_j^{\Gamma_{R-1}}, v_i^R) = \\ \sum_{q=1}^4 \Psi_q^R(\boldsymbol{\mu}) F_q^R(v_i^R) \quad i = 1, \dots, N_k, \\ \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) u_j^r(\boldsymbol{\mu}) a_q^r(v_j^r, \phi_i^{\Gamma_r}) + \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^{r+1}(\boldsymbol{\mu}) u_j^{r+1}(\boldsymbol{\mu}) a_q^{r+1}(v_j^{r+1}, \phi_i^{\Gamma_r}) \\ + \sum_{j=1}^{N_{\Gamma_r}} \lambda_j^r(\boldsymbol{\mu}) \left[\sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_r}, \phi_i^{\Gamma_r}) + \sum_{q=1}^4 \Theta_q^{r+1}(\boldsymbol{\mu}) a_q^{r+1}(\phi_j^{\Gamma_r}, \phi_i^{\Gamma_r}) \right] \\ = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Psi_q^{r+1}(\boldsymbol{\mu}) F_q^{r+1}(\phi_j^{\Gamma_r}) \\ i = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1. \end{array} \right. \quad (3.4.11)$$

where $\lambda^r(\boldsymbol{\mu}) = \lambda(\boldsymbol{\mu})|_{\Gamma_r}$ so that we obtain the algebraic form

$$\left\{ \begin{array}{ll} A_r \mathbf{u}_r + A_{r\Gamma} \boldsymbol{\lambda} & = \mathbf{f}_r \quad r = 1, \dots, R, \\ A_{\Gamma r} \mathbf{u}_r + A_{\Gamma r+1} \mathbf{u}_{r+1} + [A_{\Gamma\Gamma}^r + A_{\Gamma\Gamma}^{r+1}] \boldsymbol{\lambda} & = \mathbf{f}_{\Gamma}^r + \mathbf{f}_{\Gamma}^{r+1} \quad r = 1, \dots, R-1. \end{array} \right. \quad (3.4.12)$$

The linear system of the RDF method is the following:

$$\left[\begin{array}{cc} M_{rr} & M_{r\Gamma} \\ M_{r\Gamma}^T & M_{\Gamma\Gamma} \end{array} \right] \left[\begin{array}{c} \mathbf{U} \\ \boldsymbol{\lambda} \end{array} \right] = \left[\begin{array}{c} \mathbf{f}_{rr} \\ \mathbf{f}_{\Gamma} \end{array} \right], \quad (3.4.13)$$

where

$$M_{rr} = \left[\begin{array}{cccc} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_R \end{array} \right], M_{r\Gamma} = \left[\begin{array}{cccccc} A_{1\Gamma_1} & 0 & & & & \\ A_{2\Gamma_1} & A_{2\Gamma_2} & 0 & & & \\ 0 & A_{3\Gamma_2} & A_{3\Gamma_3} & 0 & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & A_{R-1\Gamma_{R-2}} & A_{R-1\Gamma_{R-1}} & \\ & & & 0 & A_{R\Gamma_{R-1}} & \end{array} \right],$$

$$M_{\Gamma\Gamma} = \begin{bmatrix} A_{\Gamma_1\Gamma_1}^1 + A_{\Gamma_1\Gamma_1}^2 & & & \\ & A_{\Gamma_2\Gamma_2}^2 + A_{\Gamma_2\Gamma_2}^3 & & \\ & & \ddots & \\ & & & A_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + A_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{R-1} \end{bmatrix}, \mathbf{f}_{rr} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_R \end{bmatrix}, \mathbf{f}_{\Gamma} = \begin{bmatrix} \mathbf{f}_{\Gamma_1}^1 + \mathbf{f}_{\Gamma_1}^2 \\ \vdots \\ \mathbf{f}_{\Gamma_{R-1}}^{R-1} + \mathbf{f}_{\Gamma_{R-1}}^R \end{bmatrix},$$

and, by recalling the affine decompositions (3.1.4) and (3.1.5), for $r = 1, \dots, R$ and for $k = 1, \dots, R-1$:

$$[\mathbf{u}_r]_j = u_j^r(\boldsymbol{\mu}), [A_r]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(v_j^r, v_i^r), \quad (3.4.14)$$

$$[\mathbf{f}_r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_j^r), [\boldsymbol{\lambda}_k]_j = u(\mathbf{x}_j^{\Gamma_k}, \boldsymbol{\mu}), \quad (3.4.15)$$

$$[A_{r\Gamma_k}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(v_j^r, \phi_i^{\Gamma_k}), \quad (3.4.16)$$

$$[A_{\Gamma_k\Gamma_k}^r]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_k}, \phi_i^{\Gamma_k}), \quad (3.4.17)$$

$$[\mathbf{f}_{\Gamma_k}^r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_k}). \quad (3.4.18)$$

As before, we split the matrix of the linear system (3.4.13) into 4 sub-blocks M_{tr} containing the contributions of the internal nodes of the R subdomains, the block $M_{\Gamma\Gamma}$ containing all the interface nodes between different subdomains and the mixed $M_{\Gamma\text{tr}}$ and $M_{\text{tr}\Gamma} = M_{\Gamma\text{tr}}^T$ coming partially from internal nodes and partially from interface nodes. The final RDF solution is defined by using the solution of the linear system as vector of coefficients to define $u(\mathbf{x}, \boldsymbol{\mu})$ of (3.4.10) and by adding the lift operator R_{g_h} , $u_{RDF}(\mathbf{x}, \boldsymbol{\mu}) = u(\mathbf{x}, \boldsymbol{\mu}) + R_{g_h}$.

System (3.4.13) represents the natural extension of the domain decomposition method in which we consider some bases coming from local properly selected solutions (RB part) and some elementary bases coming from the finite element functions (FE part).

3.4.2 Extending the FE regions

In general, the model order reduction can be applied in a smaller and local portions of the domain by extending the domain portion (including the internal interfaces) described by the full order model.

In literature, similar ideas have been presented, for example, in shape optimization when only a small portion of the shape can be modified or in parameter identification problems where the parameters are associated with spatially localized material properties [4].

We define now $\hat{\Omega}_{FE}$ as a subdomain of $\hat{\Omega}$ containing the internal interfaces. Figure 3.8 shows, through the help of a partition of the nodes of the triangulation, three example of $\hat{\Omega}_{FE}$. From now, we denote with FE nodes the nodes belonging to $\hat{\Omega}_{FE}$ and with RB nodes the nodes belonging to the remaining portion of the domain.

3.4. The Reduced basis-Domain decomposition-Finite element method (RDF)

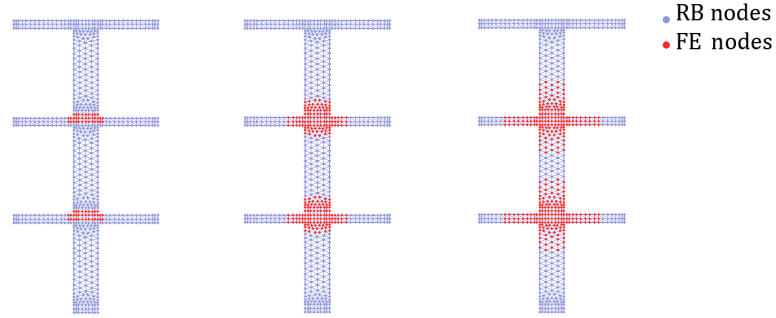


Figure 3.8: Examples of three different options for extending the FE regions in a domain composed by 3 blocks.

We still consider only a reference shape, as seen in Figure 3.5, and we define the local space V_{RB} similarly to the previous case³ in which the functions v_i are defined locally on Λ , their computations follows the same procedure of the case described in the previous section. We map them into $\hat{\Omega}_r$ and, this time, we extend them to zero on the other subdomains of $\hat{\Omega}_j$, $j \neq r$ and on the FE region $\hat{\Omega}_{FE}$. By referring to the FE nodes of the mesh in the subdomains of Ω , we define for $r = 1, \dots, R$ and $i = 1, \dots, N$:

$$v_i^r(\mathbf{x}_j) = v_i(\mathcal{T}_r(\hat{\mathbf{x}}_j)), \mathbf{x}_j \in \hat{\Omega}_r \setminus \hat{\Omega}_{FE}, \quad v_i^r(\mathbf{x}_j) = 0, \mathbf{x}_j \in \hat{\Omega}_j, j \neq r, \quad v_i^r(\mathbf{x}_j) = 0, \mathbf{x}_j \in \hat{\Omega}_{FE}. \quad (3.4.19)$$

As before, we show in Figure 3.9 some particular functions v_i^r , $r = 1, \dots, 4$ this time defined onto the subdomains $\hat{\Omega}_r$, $r = 1, \dots, R$ as described in (3.4.19), where the functions v_i are still solutions of local heat conduction problems on Λ by imposing non-homogeneous constant Dirichlet BCs on $\hat{\Gamma}^T$ and $\hat{\Gamma}^B$.

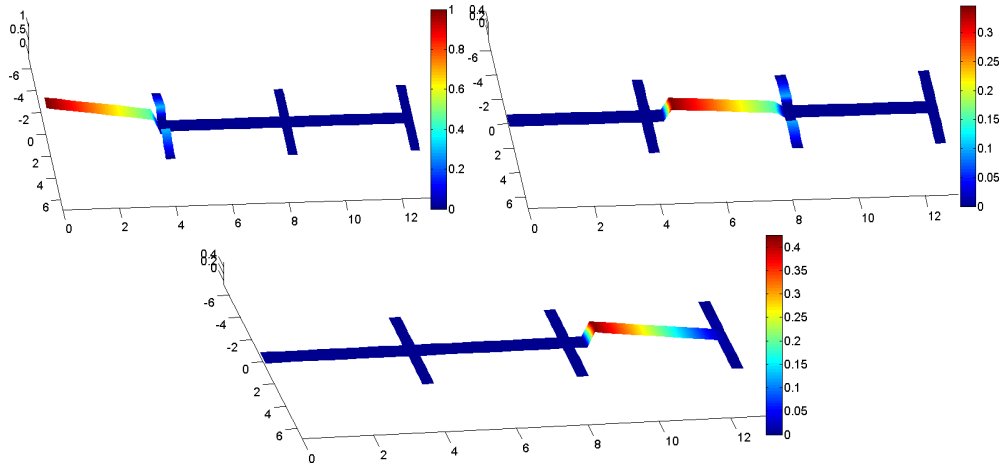


Figure 3.9: Example of a RB functions associated to each subdomain $\hat{\Omega}_r$, $r = 1, 2, 3$.

Then, depending on the BCs of the global problem that we have to solve we set: $v_i^r(\mathbf{x}) = 0, \forall \mathbf{x} \in \Gamma_D$,

³In general, in the case of K reference shapes, we define K local reduced basis spaces build by N_k reduced basis functions.

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

where Γ_D denotes the boundary of Ω with Dirichlet BCs. By considering problem (3.1.1), we consider $\Gamma_D = \Gamma_{in} \cup \Gamma_{out}$.

The functions v_i^r are linearly defined between the nodes of the mesh in Ω , such that $v_i \in V_h$ defined in (3.2.5).

We associate to each $\hat{\Omega}_r$ the following space:

$$V_{RB}^r = \{v_i^r, i = 1, \dots, N\}, r = 1, \dots, R. \quad (3.4.20)$$

We consider now a subspace of the finite element space (3.2.5) composed by the finite element functions corresponding to the nodes \mathbf{x}_j on $\hat{\Omega}_{FE}$:

$$V_{FE} \equiv \{\phi_j \in V_h, j = 1, \dots, N_{FE}\}, \quad (3.4.21)$$

where N_{FE} is the number of FE nodes on $\hat{\Omega}$ and N_Γ is the number of FE nodes on $\hat{\Gamma}$, $N_{FE} > N_\Gamma$.

As before we show in Figure 3.10 the graph containing the plot of all the FE functions associated to the subdomain $\hat{\Omega}_{FE}$ by using a very coarse grid for the scope of the visualization.

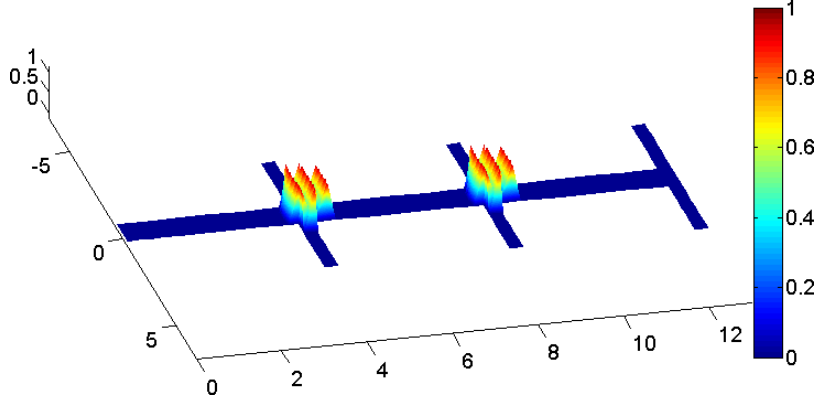


Figure 3.10: Finite element functions associated to each node belonging to the subdomain $\hat{\Omega}_{FE}$ and plotted in the same graph.

Finally we define the composite space:

$$V_{RDF} = V_{RB}^1 \oplus \dots \oplus V_{RB}^R \oplus V_{FE}. \quad (3.4.22)$$

We denote with N_{FE}^r the number of FE nodes on $\hat{\Omega}_r \cap \hat{\Omega}_{FE}$, with N_{Γ_r} the number of FE nodes on $\hat{\Gamma}_r$ and with N_F^r the number of FE nodes on $\hat{\Omega}_{FE}$ without counting the nodes in the internal interfaces.

$$N_{FE} = \sum_{r=1}^R N_{FE}^r, \quad N_\Gamma = \sum_{r=1}^{R-1} N_{\Gamma_r}, \quad N_F = \sum_{r=1}^R N_F^r.$$

We move to the case of K reference shapes, so that the number of local reduced basis functions on each $\hat{\Omega}_r$ can be different and depends on the reference shape, it is denoted by N_k , where $k = k(r)$.

Then, we consider the following basis functions:

- $v_j^r, j = 1, \dots, N_k$ are the functions associated to the domain $\hat{\Omega}_r$, as defined in (3.4.19) (an example of a RB function is represented in Figure 3.9, $r = 1, 2, 3$),

3.4. The Reduced basis-Domain decomposition-Finite element method (RDF)

- ϕ_j^r are the finite element functions associated to the FE nodes on $(\hat{\Omega}_{FE} \cap \hat{\Omega}_r) \setminus \hat{\Gamma}_I, \hat{\Gamma}_I = \cup_{r=1}^{R-1} \hat{\Gamma}_r, j = 1, \dots, N_F^r$,
- $\phi_j^{\Gamma_r}$ are the finite element functions associated to the FE nodes on the internal interfaces $\hat{\Gamma}_r \subset \hat{\Omega}_{FE}, j = 1, \dots, N_{\Gamma_r}$.

The problem (3.1.6) can be read as find $u(\boldsymbol{\mu}) \in V$:

$$\begin{cases} \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), v_i^r) = \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) & i = 1, \dots, RN_k, \\ \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), \phi_j) = \sum_{r=1}^R \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j) & j = 1, \dots, N_{FE}, \end{cases} \quad (3.4.23)$$

where ϕ_j are the FE function of V_{FE} , we are not distinguishing here ϕ_j^r and $\phi_j^{\Gamma_r}$. Problem (3.4.23) can be rewritten as composition of part defined in the subdomains: find $u(\boldsymbol{\mu}) \in V$ such that

$$\begin{cases} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), v_i^r) = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) & i = 1, \dots, N_k, r = 1, \dots, R, \\ \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), \phi_i^r) = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_i^r) & i = 1, \dots, N_F^r, r = 1, \dots, R, \\ \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(u(\boldsymbol{\mu}), \phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Theta_q^{r+1}(\boldsymbol{\mu}) a_q^{r+1}(u(\boldsymbol{\mu}), \phi_j^{\Gamma_r}) \\ = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Psi_q^{r+1}(\boldsymbol{\mu}) F_q^{r+1}(\phi_j^{\Gamma_r}) & j = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1. \end{cases} \quad (3.4.24)$$

The approximate solution $u(\boldsymbol{\mu})$ is defined in V , we can write it as linear combination of the basis functions of V .

$$u(\mathbf{x}, \boldsymbol{\mu}) = \sum_{r=1}^R \sum_{j=1}^{N_k} u_j^r(\boldsymbol{\mu}) v_j^r(\mathbf{x}) + \sum_{r=1}^R \sum_{j=1}^{N_F^r} \lambda_j^r(\boldsymbol{\mu}) \phi_j^r(\mathbf{x}) + \sum_{r=1}^{R-1} \sum_{j=1}^{N_{\Gamma_r}} \lambda_j^{\Gamma_r}(\boldsymbol{\mu}) \phi_j^{\Gamma_r}(\mathbf{x}). \quad (3.4.25)$$

Problem (3.4.24) becomes:

$$\begin{cases} \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) u_j^r(\boldsymbol{\mu}) a_q^r(v_j^r, v_i^r) + \sum_{j=1}^{N_F^r} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, v_i^r) = \\ \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_i^r) & i = 1, \dots, N_k, r = 1, \dots, R, \\ \sum_{j=1}^{N_k} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) u_j^r(\boldsymbol{\mu}) a_q^r(v_j^r, \phi_i^r) + \sum_{j=1}^{N_F^r} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^r) \\ + \sum_{j=1}^{N_{\Gamma_r}} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j^{\Gamma_r}(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_r}, \phi_i^r) = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_i^r) & i = 1, \dots, N_F^r, r = 1, \dots, R, \\ \sum_{j=1}^{N_F^r} \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \lambda_j^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^{\Gamma_r}) + \sum_{j=1}^{N_F^{r+1}} \sum_{q=1}^4 \Theta_q^{r+1}(\boldsymbol{\mu}) \lambda_j^{r+1}(\boldsymbol{\mu}) a_q^{r+1}(\phi_j^{r+1}, \phi_i^{\Gamma_r}) \\ + \sum_{j=1}^{N_{\Gamma_r}} \lambda_j^{\Gamma_r}(\boldsymbol{\mu}) \left[\sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_r}, \phi_i^{\Gamma_r}) + \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^{r+1}(\phi_j^{\Gamma_r}, \phi_i^{\Gamma_r}) \right] \\ = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_r}) + \sum_{q=1}^4 \Psi_q^{r+1}(\boldsymbol{\mu}) F_q^{r+1}(\phi_j^{\Gamma_r}) & i = 1, \dots, N_{\Gamma_r}, r = 1, \dots, R-1, \end{cases} \quad (3.4.26)$$

so that we obtain the algebraic formulation:

$$\begin{bmatrix} M_{rr} & M_{rF} & 0 \\ M_{rF}^T & M_{FF} & M_{F\Gamma} \\ 0 & M_{F\Gamma}^T & M_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{rr} \\ \mathbf{f}_F \\ \mathbf{f}_\Gamma \end{bmatrix}, \quad (3.4.27)$$

where

$$M_{rr} = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_R \end{bmatrix}, M_{rF} = \begin{bmatrix} A_{1F_1} & & & \\ & A_{2F_2} & & \\ & & \ddots & \\ & & & A_{RF_R} \end{bmatrix}, M_{FF} = \begin{bmatrix} A_{F_1} & & & \\ & A_{F_2} & & \\ & & \ddots & \\ & & & A_{F_R} \end{bmatrix},$$

$$M_{F\Gamma} = \begin{bmatrix} A_{F_1\Gamma_1} & 0 & & & \\ A_{F_2\Gamma_1} & A_{F_2\Gamma_2} & 0 & & \\ 0 & A_{F_3\Gamma_2} & A_{F_3\Gamma_3} & 0 & \\ & \ddots & \ddots & \ddots & 0 \\ & & 0 & A_{F_{R-1}\Gamma_{R-2}} & A_{F_{R-1}\Gamma_{R-1}} \\ & & & 0 & A_{F_R\Gamma_{R-1}} \end{bmatrix},$$

$$M_{\Gamma\Gamma} = \begin{bmatrix} A_{\Gamma_1\Gamma_1}^1 + A_{\Gamma_1\Gamma_1}^2 & & & \\ & A_{\Gamma_2\Gamma_2}^2 + A_{\Gamma_2\Gamma_2}^3 & & \\ & & \ddots & \\ & & & A_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + A_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_R \end{bmatrix}, \boldsymbol{\lambda}^\Gamma = \begin{bmatrix} \boldsymbol{\lambda}^{\Gamma_1} \\ \vdots \\ \boldsymbol{\lambda}^{\Gamma_{R-1}} \end{bmatrix},$$

$$\mathbf{f}_{rr} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_R \end{bmatrix}, \mathbf{f}_F = \begin{bmatrix} \mathbf{f}_{F_1} \\ \vdots \\ \mathbf{f}_{F_R} \end{bmatrix}, \mathbf{f}_\Gamma = \begin{bmatrix} \mathbf{f}_{\Gamma_1}^1 + \mathbf{f}_{\Gamma_1}^2 \\ \vdots \\ \mathbf{f}_{\Gamma_{R-1}}^{R-1} + \mathbf{f}_{\Gamma_{R-1}}^R \end{bmatrix},$$

3.4. The Reduced basis-Domain decomposition-Finite element method (RDF)

and, by recalling the affine decompositions (3.1.4) and (3.1.5), we define for $r = 1, \dots, R$ and for $k = 1, \dots, R - 1$:

$$[\mathbf{U}]_j = u_j^r(\boldsymbol{\mu}), \quad (3.4.28)$$

$$[A_r]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(v_j^r, v_i^r), \quad (3.4.29)$$

$$[\mathbf{f}_r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(v_j^r), \quad (3.4.30)$$

$$[\boldsymbol{\lambda}_r]_j = u(\mathbf{x}_j, \boldsymbol{\mu}), \mathbf{x}_j \in (\hat{\Omega}_{FE} \cap \hat{\Omega}_r) \setminus \Gamma_I, \quad (3.4.31)$$

$$[A_{Fr}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^r), \quad (3.4.32)$$

$$[A_{rFr}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(v_j^r, \phi_i^r), \quad (3.4.33)$$

$$[\mathbf{f}_{Fr}]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^r), \quad (3.4.34)$$

$$[\boldsymbol{\lambda}_k^\Gamma]_j = u(\mathbf{x}_j, \boldsymbol{\mu}), \mathbf{x}_j \in \hat{\Gamma}_k, \quad (3.4.35)$$

$$[A_{Fr\Gamma_k}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^{\Gamma_k}), \quad (3.4.36)$$

$$[A_{\Gamma_k\Gamma_k}^r]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) a_q^r(\phi_j^{\Gamma_k}, \phi_i^{\Gamma_k}), \quad (3.4.37)$$

$$[\mathbf{f}_{\Gamma_k}^r]_j = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F_q^r(\phi_j^{\Gamma_k}). \quad (3.4.38)$$

As before, the matrix A is split into different sub-blocks A_{rr} , $A_{\Gamma\Gamma}$, $A_{\Gamma F}$ and A_{FF} that contain the contributions of, respectively, the internal nodes of the R subdomains, the interface nodes between different subdomains and their interactions. The final RDF solution is defined by using the solution of the linear system as vector of coefficients to define $u(\mathbf{x}, \boldsymbol{\mu})$ of (3.4.25) and by adding the lift operator R_{g_h} , $u_{RDF}(\mathbf{x}, \boldsymbol{\mu}) = u(\mathbf{x}, \boldsymbol{\mu}) + R_{g_h}$.

System (3.4.27) still represents the natural extension of the domain decomposition method in which some bases arise from local solutions (RB part) and some from the local finite element interface functions (FE part).

This methodology automatically guarantees the continuity of the variables of interest at the interfaces which are left free, i.e. a projection on the finite elements basis. However in order to be able to approximate the solution at the interface it is necessary to define the local RB spaces by taking into account the fact that the solution on $\hat{\Gamma}_I$ is unknown. Some strategies for the computation of the local reduced basis functions and the corresponding numerical results are detailed in the following sections.

System (3.4.27) may be recast in the Schur complement form for $\boldsymbol{\lambda}$. Particularly, in the case of two domains all the classical preconditioning strategies like Dirichlet-Neumann, Neumann-Neumann and Robin-Robin work as expected. For more subdomains the Neumann-Neumann preconditioner or other problem specific preconditioners may be used to solve the Schur complement system in an effective way.

3.4.3 Precomputing the basis solutions

The computation of the RB basis functions is performed locally in the reference shape Λ and during an offline step. It is independent of the number of subdomains that will compose the whole computational domain. The way to define these basis functions is arbitrary and can depend on the physical characteristics of the problem. The way to compute these functions affects strongly the numerical solution of the final system (3.4.13) (or (3.4.27)) and its accuracy with respect to the exact solution of the problem.

The goal at this stage is to define a set of basis functions that is large enough to recover locally the final solution of the problem. To perform this task, we have to take into account two main aspects, first that the functions that we define now in Λ will be used locally as basis to recover the solution on the whole domain Ω . In particular, this means that the BCs of the local problem defined in Λ will define a basis to recover the trace of the final solution on the internal interfaces. On the other hand each local problem depends on the local parameters μ_i that are unknown at this stage.

We consider the reference shape Λ and we want to define the reduced basis space V_{RB} introduced in (3.4.3).

We introduce the following parametric local Laplace problem in a generic subdomain Ω_r . We denote with $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ the two boundaries of Ω_r that may correspond to the inlet, the outlet or the internal interfaces in the whole domain Ω . We denote with the function $\lambda(\mu_\lambda)$ the Dirichlet BC on $\Gamma_D = \hat{\Gamma}_T \cup \hat{\Gamma}_B$. The parameter μ_λ characterizes the profile of the function $\lambda(\mu_\lambda)$ on the boundary Γ_D .

$$\begin{cases} \mu_1^r \Delta u &= 0 & \text{in } \Omega_{\mu_2^r} \\ u &= \lambda(\mu_\lambda) & \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= 0 & \text{on } \Gamma_r \end{cases} . \quad (3.4.39)$$

By considering the weak formulation of problem (3.4.39), we can apply the same local affine decomposition of (3.1.6) to recast problem (3.4.39) on $\hat{\Omega}_r$ and consequently on Λ , that represents a translation of $\hat{\Omega}_r$.

As seen for the FE formulation of the conduction problem in Ω , thanks to the lift operator, we define on Λ the finite element approximation \tilde{u} of problem (3.4.39) such that $\tilde{u} = u + R_{gh}$ and find $u = u(\mu, \mu_\lambda) \in V_h$ such that

$$\sum_{q=1}^Q \Theta_q(\mu) a_q(u, v_h) = - \sum_{q=1}^{QL} \Psi_q(\mu, \mu_\lambda) F_q(v_h), \quad \forall v_h \in V_h, \quad (3.4.40)$$

where

$$V_h = X_h^1 \equiv \{v_h \in C^0(\Lambda) : v_h|_K \in \mathbb{P}_1 \quad \forall K \in \mathcal{T}_h\} \cap H_{0\Gamma_D}^1. \quad (3.4.41)$$

3.4. The Reduced basis-Domain decomposition-Finite element method (RDF)

The linear and bilinear forms, together with the parametric coefficient of (3.4.40) are defined through the affine decomposition as follows:

$$a_{q(i,j)}(u, v) = \int_{\Lambda} \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_j} d\mathbf{x},$$

$$\Theta_{q(i,j)}(\boldsymbol{\mu}) = \mu_1 [C_{\mu_2}^{-1} C_{\mu_2}^{-T} |det(C_{\mu_2})|]_{ij},$$

$$F_{q(i,j,l)}(v) = - \int_{\Lambda} \frac{\partial G_{\mu_{\lambda}^l}}{\partial x_i} \frac{\partial v}{\partial x_j} d\mathbf{x},$$

$$\Psi_{q(i,j,l)}(\boldsymbol{\mu}, \mu_{\lambda}) = \mu_1 \tilde{\lambda}_l(\mu_{\lambda}) [C_{\mu_2}^{-1} C_{\mu_2}^{-T} |det(C_{\mu_2})|]_{ij},$$

where $G_{\mu_{\lambda}}$ is the lift of the Dirichlet BC and it can be an affine function described as follows:

$$G_{\mu_{\lambda}}(x, \mu_{\lambda}) = \sum_{l=1}^L \tilde{\lambda}_l(\mu_{\lambda}) G_{\mu_{\lambda}^l}(x), \quad (3.4.42)$$

or can be approximated by an affine function through the empirical interpolation method [7]:

$$G_{\mu_{\lambda}}(x, \mu_{\lambda}) \approx \sum_{l=1}^L \tilde{\lambda}_l(\mu_{\lambda}) G_{\mu_{\lambda}^l}(x, \mu_{\lambda}^l). \quad (3.4.43)$$

We apply now the offline stage of the classical reduced basis method for the parametrized problem (3.4.40) in Λ in order to select a set of suitable parameter values that define the local reduced basis functions.

Thanks to the greedy algorithm (Section 1.2.5.1) we define the set $\{(\boldsymbol{\mu}_i, \lambda_{\lambda}^i), i = 1, \dots, N\}$.

Finally the local RB space is defined as the space of the solutions of (3.4.40) corresponding to the parameters values selected by the greedy algorithm:

$$V_{RB} = \{u(\boldsymbol{\mu}^i, \mu_{\lambda}^i), i = 1, \dots, N\} \quad (3.4.44)$$

In general if there are k reference domains, we define k independent parametric problems and we perform k independent greedy algorithms (through parallel computations) and we define k reduced basis spaces V_{RB}^k with their proper number N_k of basis functions.

The choice of the function $\lambda(\mu_{\lambda})$ used as local BC strongly affects the final solution of the problem, on the other hand it influences the computational complexity of the greedy algorithm.

Due to this strong relation between the choice of λ and (i) the accuracy, (ii) the computational complexity and (iii) the performances of the method, in Section 3.6 we introduce several options for defining λ and we report the related results regarding the offline complexity, the solution accuracy reached during the online stage and the corresponding computational times.

3.5 Offline/Online decomposition and computational complexity

As already mentioned in the previous section, all the computations involving the construction of the local reduced basis functions are performed in an offline stage.

The offline/online computational decomposition regards also the matrix assembling of the linear system (3.4.12) or, in the case of extended FE region, of (3.4.27). In both cases the matrices can be assembled starting from the FE matrices associated to the original problem (4.4.19), in which $\mathbb{A} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, $\mathbb{F} \in \mathbb{R}^{\mathcal{N} \times 1}$. We note that \mathcal{N} indicates the dimension of the whole finite element space and N_k the number of local reduced basis functions, we recall that in general $N_k \ll \mathcal{N}$. We still assume that we have only a reference shape Λ ($k = 1$), and N is the number of reduced basis functions for every subdomain $\hat{\Omega}_r$ and in its translation Λ . Moreover, we assume that the number of nodes N_{Γ_r} along the internal interfaces Γ_k of the computational domain, belonging to Ω_r , while N_{Γ} is the number of nodes on each internal interface, that is the same $\forall k \in \{1, \dots, R-1\}$ and it coincides with that is the number of nodes on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ in Λ .

In particular, thanks to the introduced domain decomposition approach, can be decomposed in submatrices and subvectors that can be defined locally in the reference domains and without further assumptions on the reference shape. Due to the fact that $\hat{\Omega}_r$ is a translation of Λ , $\hat{\Omega}_r = \mathcal{T}_r(\Lambda)$, we can assume that:

$$a_r(v, w) = \int_{\hat{\Omega}_r} \nabla v \cdot \nabla w d\mathbf{x} = \int_{\Lambda} \nabla v(\mathcal{T}_r \mathbf{x}) \cdot \nabla w(\mathcal{T}_r \mathbf{x}) d\mathbf{x} = \int_{\Lambda} \nabla v \cdot \nabla w d\mathbf{x}. \quad (3.5.1)$$

Similar considerations can be made for $F_r(v)$, so that we can refer only to the bilinear and linear forms $a(v, w)$ and $F(v)$ defined in Λ .

3.5.1 FE region represented only by the internal interfaces

We introduce the matrix $U \in \mathbb{R}^{(\mathcal{N}_k - N_{\Gamma_k}) \times N}$, containing the local reduced basis functions computed in Λ and associated to the nodes of $\Lambda \setminus \Gamma_D$.

The submatrices A_r of the systems (3.4.12) are defined as follows, for $r = 1, \dots, R$:

$$A_r = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) A_r^q = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) U^T \mathbb{A}^q U, \quad (3.5.2)$$

where, by denoting with $\phi_j, j = 1, \dots, \mathcal{N}_k - N_{\Gamma_k}$ the finite element functions associated to the nodes on $\Lambda \setminus \Gamma_D$ and with $\phi_j^{\Gamma}, j = 1, \dots, N_{\Gamma_k}$ the finite element functions associated to the nodes on Γ_D (see Section 3.4.1), we define:

$$[\mathbb{A}^q]_{ij} = a_q(\phi_j, \phi_i), \quad i, j = 1, \dots, \mathcal{N}_k - N_{\Gamma_k}, \quad (3.5.3)$$

$$[A^q]_{ij} = a_q(v_j, v_i), \quad i, j = 1, \dots, N. \quad (3.5.4)$$

Once we have defined the local reduced basis, the matrices $U^T \mathbb{A}^q U, q = 1, \dots, 4$, can be assembled offline and stored as matrices of smaller dimensions, $A^q \in \mathbb{R}^{N \times N}$.

Regarding the assembling of the matrices $A_{r\Gamma_k}, r = 1, \dots, R, k = 1, \dots, R-1$, we follow a similar procedure:

$$A_{r\Gamma_k} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) A_{\Gamma}^q = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) U^T \mathbb{A}_{\Gamma}^q, \quad (3.5.5)$$

where

$$[\mathbb{A}_\Gamma^q]_{ij} = a_q(\phi_j, \phi_i^\Gamma), \quad i = 1, \dots, \mathcal{N}_k - N_{\Gamma_k}, \quad j = 1, \dots, N_{\Gamma_k}, \quad (3.5.6)$$

$$[A_\Gamma^q]_{ij} = a_q(v_j, \phi_i^\Gamma), \quad i = 1, \dots, N, \quad j = 1, \dots, N_{\Gamma_k}. \quad (3.5.7)$$

The matrices $U^T \mathbb{A}_\Gamma^q, q = 1, \dots, 4$, can be computed and stored in the smaller matrices $A_\Gamma^q \in \mathbb{R}^{N \times N_{\Gamma_k}}$. The matrix $A_{\Gamma\Gamma}$ coincides exactly with the finite element submatrix involving the nodes on the internal interfaces and it is built by the following submatrices:

$$A_{\Gamma_k \Gamma_k}^r = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \mathbb{A}_{\Gamma\Gamma}^q, \quad r = 1, \dots, R, \quad (3.5.8)$$

$$[\mathbb{A}_{\Gamma\Gamma}^q]_{ij} = a_q(\phi_j^\Gamma, \phi_i^\Gamma), \quad i, j = 1, \dots, N_{\Gamma_k}. \quad (3.5.9)$$

These small dimensional matrices $\mathbb{A}_{\Gamma\Gamma}^q \in \mathbb{R}^{N_{\Gamma} \times N_{\Gamma}}$ can be computed and stored offline.

The same decoupling strategy is considered for the right hand side of the system (3.4.12):

$$\mathbf{f}_r = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) F^q = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) U^T \mathbb{F}^q, \quad r = 1, \dots, R, \quad (3.5.10)$$

$$\mathbf{f}_{\Gamma_k}^r = \sum_{q=1}^4 \Psi_q^r(\boldsymbol{\mu}) \mathbb{F}_\Gamma^q, \quad r = 1, \dots, R, \quad k = 1, \dots, R-1, \quad (3.5.11)$$

where

$$[F^q]_i = F_q(v_i), \quad i = 1, \dots, N, \quad (3.5.12)$$

$$[\mathbb{F}^q]_i = F_q(\phi_i), \quad i = 1, \dots, \mathcal{N}_k - N_{\Gamma_k}, \quad (3.5.13)$$

$$[\mathbb{F}_\Gamma^q]_i = F_q(\phi_i^\Gamma), \quad i = 1, \dots, N_{\Gamma_k}. \quad (3.5.14)$$

The vectors $U^T \mathbb{F}^q \in \mathbb{R}^{N \times 1}$ and $\mathbb{F}_\Gamma^q \in \mathbb{R}^{N_{\Gamma_k} \times 1}, q = 1, \dots, 4$ can be computed and stored offline.

In the online stage, for a new value of $\boldsymbol{\mu}$ and for an arbitrary number R of subdomains, we assemble the matrices $A_r, A_{r\Gamma_k}, A_{\Gamma_k \Gamma_k}^r$ and the vectors \mathbf{f}_r and $\mathbf{f}_{\Gamma_k}^r$ for $r = 1, \dots, R$ and $k = 1, \dots, R-1$, that will define respectively the matrices $A_{rr} \in \mathbb{R}^{RN \times RN}, A_{r\Gamma} \in \mathbb{R}^{RN \times (R-1)N_{\Gamma}}, A_{\Gamma\Gamma} \in \mathbb{R}^{(R-1)N_{\Gamma} \times (R-1)N_{\Gamma}}$ and the vectors $\mathbf{f}_{rr} \in \mathbb{R}^{RN \times 1}$ and $\mathbf{f}_{\Gamma} \in \mathbb{R}^{(R-1)N_{\Gamma} \times 1}$. Finally we solve the linear system (3.4.12).

The quantities that determine the dimension of the final system (that has to be solved online to find the RDF solution of the problem) are:

- the number of subdomains R ,
- the number of nodes along each internal interface N_{Γ} ,
- the number of local reduced basis function N ,

so that the dimension of the linear system (3.4.12) is $(RN + (R-1)N_{\Gamma}) \times (RN + (R-1)N_{\Gamma})$.

3.5.2 Extending the FE region

As introduced in the last section, in this case we extend the region $\hat{\Omega}_{FE}$ in which we use the finite element functions as basis for the solution of the problem, so that $\hat{\Gamma}_I \subset \hat{\Omega}_{FE} \subset \hat{\Omega}$. We have introduced the quantity $N_{FE} \ll \mathcal{N}$ equal to the nodes of the domain $\hat{\Omega}$ involved in the region $\hat{\Omega}_{FE}$, such that $N_{FE} = \sum_{r=1}^R N_F^r + (R-1)N_\Gamma$, we assume that the quantity N_F^r is the same $\forall r \in \{1, \dots, R\}$. We can consider on Λ the corresponding region $\Lambda_{FE} = \mathcal{T}_r^{-1}(\hat{\Omega}_{FE} \cap \hat{\Omega}_r \setminus \Gamma_I)$ and denote with $N_F = N_F^r$ the number of nodes involved in this region.

We define the matrix $\tilde{U} \in \mathbb{R}^{(\mathcal{N}-N_F) \times N}$ containing the local reduced basis functions associated to the subdomain $\Lambda \setminus \Lambda_{FE}$. The submatrices A_r of the system (3.4.27) are defined as before, for $r = 1, \dots, R$:

$$A_r = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) A_r^q = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \tilde{U}_r^T \mathbb{A}_r^q \tilde{U}_r, \quad (3.5.15)$$

where, by denoting with $\phi_j^r, j = 1, \dots, \mathcal{N}_k - N_F$ the finite element functions associated to the nodes on $\Lambda \setminus \Lambda_{FE}$, we define:

$$[\mathbb{A}_r^q]_{ij} = a_q^r(\phi_j, \phi_i), \quad i, j = 1, \dots, \mathcal{N}_k - N_F, \quad (3.5.16)$$

$$[A_r^q]_{ij} = a_q^r(v_j^r, v_i^r), \quad i, j = 1, \dots, N. \quad (3.5.17)$$

Once we have defined the local reduced basis, the matrices $U_r^T \mathbb{A}_r^q U_r$ can be assembled offline and stored as matrices of smaller dimensions, $A_r^q \in \mathbb{R}^{N \times N}$.

Regarding the assembling of the matrices A_{rF_r} , we have:

$$A_{rF_r} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) A_F^q = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \tilde{U}_r^T \mathbb{A}_F^q, \quad (3.5.18)$$

where, by denoting with $\phi_j^F, j = 1, \dots, N_F$ the finite element functions associated to the nodes on Λ_{FE} , we define:

$$[\mathbb{A}_F^q]_{ij} = a_q(\phi_j, \phi_i^F), \quad i = 1, \dots, \mathcal{N}_k - N_F, \quad j = 1, \dots, N_F, \quad (3.5.19)$$

$$[A_F^q]_{ij} = a_q(v_j, \phi_i^F), \quad i = 1, \dots, N, \quad j = 1, \dots, N_F. \quad (3.5.20)$$

The matrices $\tilde{U}_r^T \mathbb{A}_F^q$ can be computed and stored offline in the smaller matrices $A_F^q \in \mathbb{R}^{N \times N_F}, q = 1, \dots, 4$. The matrices $A_{FF}, A_{F\Gamma}$ and $A_{\Gamma\Gamma}$ coincide with the FE submatrices corresponding to the nodes on the FE region of the domain. More precisely they are defined starting from the following submatrices:

$$A_{F_r} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \mathbb{A}_{FF}^q, \quad (3.5.21)$$

$$A_{F_r\Gamma_k} = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \mathbb{A}_{F\Gamma}^q, \quad (3.5.22)$$

$$A_{\Gamma_k\Gamma_k}^r = \sum_{q=1}^4 \Theta_q^r(\boldsymbol{\mu}) \mathbb{A}_{\Gamma\Gamma}^q, \quad (3.5.23)$$

where

$$[\mathbb{A}_{FF}^q]_{ij} = a_q(\phi_j^F, \phi_i^F), \quad i, j = 1, \dots, N_F, \quad (3.5.24)$$

$$[\mathbb{A}_{F\Gamma}^q]_{ij} = a_q(\phi_j^F, \phi_i^\Gamma), \quad i = 1, \dots, N_F, \quad j = 1, \dots, N_\Gamma, \quad (3.5.25)$$

$$[\mathbb{A}_{\Gamma\Gamma}^q]_{ij} = a_q(\phi_j^\Gamma, \phi_i^\Gamma), \quad i, j = 1, \dots, N_\Gamma. \quad (3.5.26)$$

$$(3.5.27)$$

Also these small dimensional matrices $\mathbb{A}_{FF}^q \in \mathbb{R}^{N_F \times N_F}$, $\mathbb{A}_{F\Gamma}^q \in \mathbb{R}^{N_F \times N_\Gamma}$, $\mathbb{A}_{\Gamma\Gamma}^q \in \mathbb{R}^{N_\Gamma \times N_\Gamma}$, can be computed and stored offline. Analogous considerations regard the right hand side term of system (3.4.27).

$$\mathbf{f}_r = \sum_{q=1}^4 \Psi_r^q(\boldsymbol{\mu}), \quad F^q = \sum_{q=1}^4 \Psi_r^q(\boldsymbol{\mu}) \tilde{U}^T \mathbb{F}^q, \quad r = 1, \dots, R, \quad (3.5.28)$$

$$\mathbf{f}_{F_r} = \sum_{q=1}^4 \Psi_r^q(\boldsymbol{\mu}) \mathbb{F}_F^q, \quad r = 1, \dots, R, \quad (3.5.29)$$

$$\mathbf{f}_{\Gamma_k}^r = \sum_{q=1}^4 \Psi_r^q(\boldsymbol{\mu}) \mathbb{F}_\Gamma^q, \quad r = 1, \dots, R, \quad k = 1, \dots, R-1, \quad (3.5.30)$$

where

$$[F^q]_i = F_q(v_i), \quad i = 1, \dots, N, \quad (3.5.31)$$

$$[\mathbb{F}^q]_i = F_q(\phi_i), \quad i = 1, \dots, \mathcal{N}_k - N_F, \quad (3.5.32)$$

$$[\mathbb{F}_F^q]_i = F_q(\phi_i^F), \quad i = 1, \dots, N_F. \quad (3.5.33)$$

$$[\mathbb{F}_\Gamma^q]_i = F_q(\phi_i^\Gamma), \quad i = 1, \dots, N_\Gamma. \quad (3.5.34)$$

The vectors $\tilde{U}^T \mathbb{F}^q \in \mathbb{R}^{N \times 1}$, $\mathbb{F}_F^q \in \mathbb{R}^{N_F \times 1}$ and $\mathbb{F}_\Gamma^q \in \mathbb{R}^{N_\Gamma \times 1}$, $q = 1, \dots, 4$ can be computed and stored offline. In the online stage, as in the previous case, for a new value of $\boldsymbol{\mu}$ and for an arbitrary number R of subdomains, we assemble the matrices $A_r, A_{rF_r}, A_{F_r}, A_{F_r\Gamma_k}, A_{\Gamma_k\Gamma_k}^r$ and the vectors $\mathbf{f}_r, \mathbf{f}_{F_r}$ and $\mathbf{f}_{\Gamma_k}^r$ for $r = 1, \dots, R$ and $k = 1, \dots, R-1$, that will define respectively the matrices $A_{rr} \in \mathbb{R}^{RN \times RN}$, $A_{rF} \in \mathbb{R}^{RN \times RN_F}$, $A_{FF} \in \mathbb{R}^{RN_F \times RN_F}$, $A_{F\Gamma} \in \mathbb{R}^{RN_F \times (R-1)N_\Gamma}$, $A_{\Gamma\Gamma} \in \mathbb{R}^{(R-1)N_\Gamma \times (R-1)N_\Gamma}$ and the vectors $\mathbf{f}_{rr} \in \mathbb{R}^{RN \times 1}$, $\mathbf{f}_F \in \mathbb{R}^{RN_F \times 1}$ and $\mathbf{f}_\Gamma \in \mathbb{R}^{(R-1)N_\Gamma \times 1}$. Finally we solve the linear system (3.4.27).

Summarizing the quantities that determine the dimension of the final system that has to be solved to find the RDF solution of the problem are:

- the number of subdomains R ,
- the number of nodes along each internal interface N_Γ ,
- the number of nodes involved in the FE region N_F ,
- the number of local reduced basis function N ,

so that the dimension of the linear system (3.4.27) is $(R(N + N_F) + (R-1)N_\Gamma) \times (R(N + N_F) + (R-1)N_\Gamma)$. We note that the dimension of the system (3.4.27) is bigger than the dimension of the system (3.4.12) (depends on RN_F), leading to a greater computational cost that is counterbalanced by a better accuracy of the solution.

In the following section, several numerical results show how to achieve the desired accuracy of the solution in a competitive computational time (respect to the classical FE method) by tuning the

ingredients of the RDF method: the extension of the FE region, the number of the reduced basis functions and the set of local BCs used to compute the reduced basis space.

3.6 Numerical results

We present in this section some numerical results dealing with the RDF approximated solution of problem (3.1.1). We consider two different main sets of reduced basis functions associated to two different kinds of functions describing the parametric Dirichlet BCs of the local problem (3.4.40): the Lagrange functions and the Fourier functions. We show in this section how the relative errors between the RDF and the FE solution (always in H^1 norm) are affected by these different local choices. Moreover, we present numerical tests involving different FE portions of the domain together with a discussion about the computational time savings with respect to the FE full order method.

3.6.1 The local BCs and the local greedy algorithms

The choice of functions associated to the local parametric BCs for the problem (3.4.40) has the role to create a large enough set of local space of basis functions to recover the value of the final solution along the internal interfaces for different combinations of subdomains and parameters that define the whole computational domain. We introduce here two possible main options to define the set of local BCs of the problem (3.4.40) defined in Λ . In particular, we consider, first, that $\lambda(\mu_\lambda^i)$, $i = 1, \dots, M$, are the M Lagrangian piecewise functions defined on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ (see Figure 3.5). Then, we explore the possibility to define $\lambda(\mu_\lambda^i)$, $i = 1, \dots, M$ as M different Fourier functions on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$. The next subsections describe in detail the role of both sets of bases in the offline stage of method.

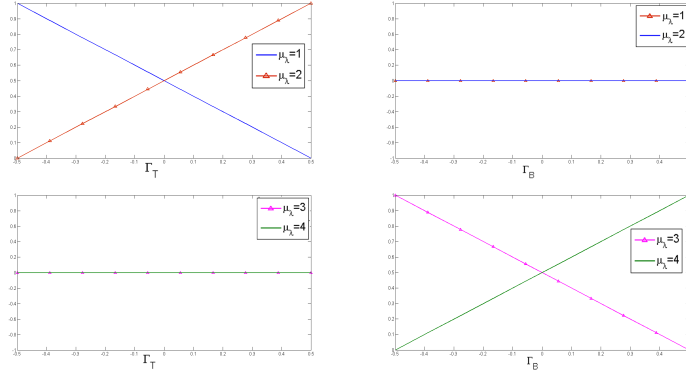
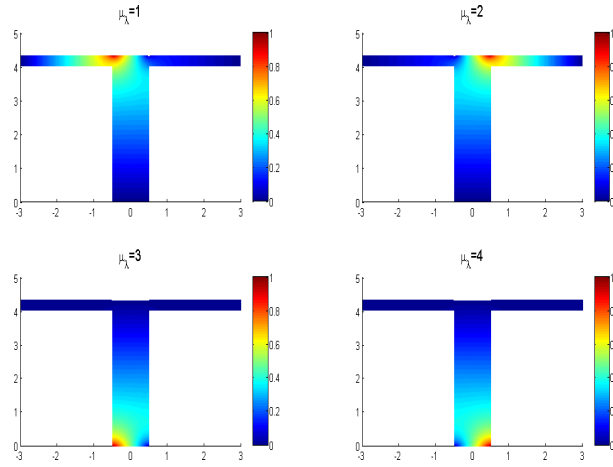
3.6.1.1 The Lagrangian piecewise interface functions

The first adopted option is the assumption that $\lambda(\mu_\lambda)$ represents, for different values μ_λ , a Lagrangian piecewise basis function defined on some nodes of the interfaces $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$. Furthermore, we have more sub-options in the range of this kind of functions that depend on the dimension of the support of the Lagrangian functions, that we denote with H .

In this case the parameter μ_λ that characterizes the function $\lambda(\mu_\lambda)$ indicates the node where $\lambda(\mu_\lambda)$ is equal to 1.

We suppose that $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ have the same dimension and N_Γ is the number of nodes on these boundaries. We assume that the nodes of the mesh belonging to $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ are equipartitioned and h is the distance between them. The support of $\lambda(\mu_\lambda)$ can be smaller or equal to the interface length, $H \leq (N_\Gamma - 1)h$, so that the number of the Lagrangian possible functions is equal or smaller than the number of nodes N_Γ . We associate a value of $\mu_\lambda \in \{1, 2, \dots, 2(\frac{(N_\Gamma - 1)h}{H} + 1)\}$ to each Lagrangian basis functions.

Let us consider, as first case, that the support coincides with the interface $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$, so that $H = (N_\Gamma - 1)h$ (in these numerical tests we have $(N_\Gamma - 1)h = 1$) and $\mu_\lambda = 1, \dots, 4$. The functions $\lambda(1)$ and $\lambda(2)$ are defined as the Lagrangian piecewise functions associated to the two extreme nodes of $\hat{\Gamma}_T$ and equal to zero on $\hat{\Gamma}_B$ and $\lambda(3)$ and $\lambda(4)$ are the Lagrangian piecewise functions associated to the two extreme nodes of $\hat{\Gamma}_B$ and equal to zero on $\hat{\Gamma}_T$, see Figure 3.11. Figure 3.12 shows, for an arbitrary parameter μ , the solutions $u(\mu, \mu_\lambda)$ of (3.4.40) by fixing $H = (N_\Gamma - 1)h$ and in correspondence of the ‘‘Lagrange BCs’’ $\lambda(\mu_\lambda)$ with $\mu_\lambda = 1, 2, 3, 4$.

Figure 3.11: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $H=1$ and $\mu_\lambda = 1, 2, 3, 4$.Figure 3.12: Solutions $u(\mu, \mu_\lambda)$ of (3.4.40) $\mu_\lambda = 1, 2, 3, 4, H = 1$.

By varying the value of H , we have different sets of Lagrangian functions $\lambda(\mu_\lambda)$ to use as BCs and consequently we have different sets of possible values of μ_λ . The parameter μ_λ is involved, together with μ , in the greedy algorithm and in the selection of the local reduced basis space.

By recalling that in the local parametric problem (3.4.40) the parameters are $\mu = (\mu_1, \mu_2)$ and μ_λ , we observe that the parameter μ_1 (for the specific considered problem) does not influence the solution $u(\mu, \mu_\lambda)$ of the problem (3.4.40), so that we can set $\mu_1 = 1$ and consider only two parameters involved in the greedy algorithm: μ_2 and μ_λ .

Figures 3.13 and 3.14 show the distribution of the parameter values μ_2^i and μ_λ^i selected during the greedy algorithm, with $H = 1$.

In general, we can reduce the value of H in order to consider a larger space of Lagrangian functions along the interface that may approximate better the final solution of the problem.

We consider a second example with $H = 1/2$, in this case $\mu_\lambda \in \{1, \dots, 6\}$ and the profiles of the function λ dependent on μ_λ are shown in the Figure 3.15 (here we report only the interface in which λ is non-zero).

Figure 3.16 shows, for fixed parameter μ , the solutions $u(\mu, \mu_\lambda)$ of (3.4.40) associated to the values $\mu_\lambda = 1, \dots, 6$ for $H = 1/2$.

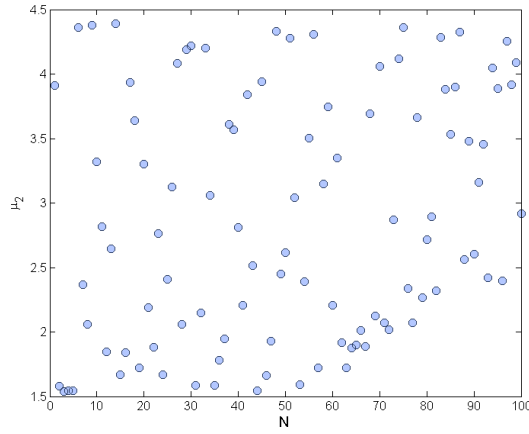


Figure 3.13: Parameter values of the selected μ_2^i , $i = 1, \dots, N$ during the greedy algorithm for $H = 1$.

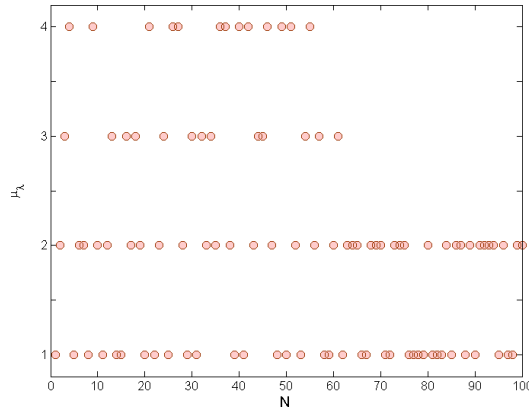


Figure 3.14: Parameter values of the selected μ_λ^i , $i = 1, \dots, N$ during the greedy algorithm for $H = 1$.

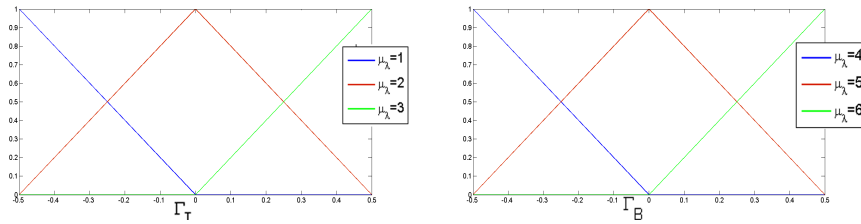


Figure 3.15: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $H=1/2$ and $\mu_\lambda = 1, 2, 3, 4, 5, 6$.

Figure 3.17 shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm, with $H = 1/2$. The distribution of the parameter values μ_2^i is similar to the previous case.

We report in detail a last case in which $H = 1/4$ and $\mu_\lambda \in \{1, \dots, 10\}$ and the displacements of the function λ in dependence of μ_λ are shown in the Figure 3.18.

Figure 3.16 shows, for fixed parameter μ , the solutions $u(\mu, \mu_\lambda)$ of (3.4.40) associated to the values $\mu_\lambda = 1, \dots, 10$ for $H = 1/4$.

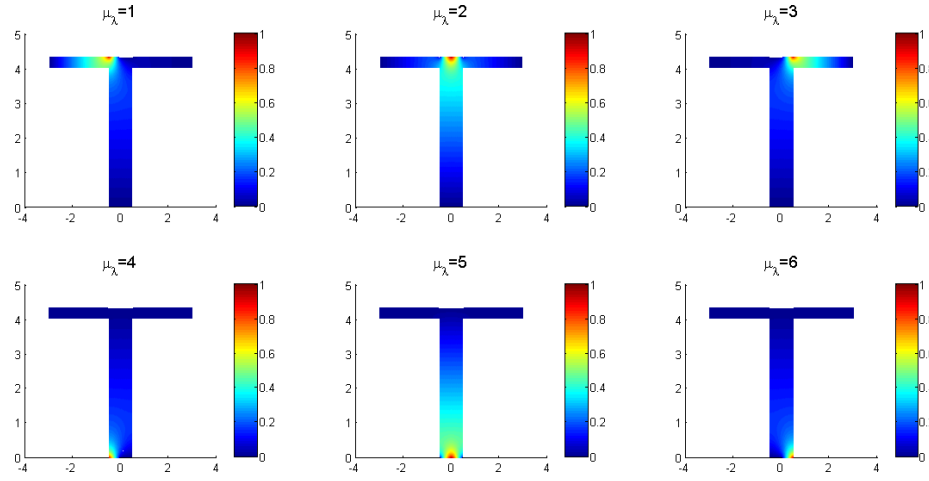
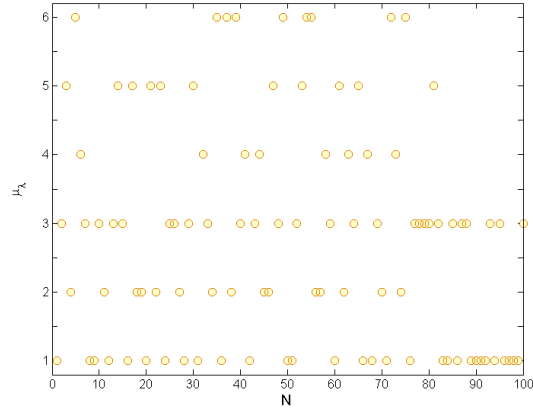
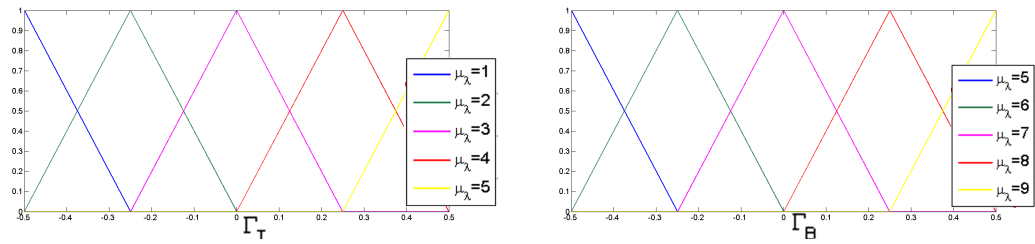

 Figure 3.16: Solutions $u(\mu, \mu_\lambda)$ of (3.4.40), $\mu_\lambda = 1, 2, 3, 4, 5, 6$, $H = 1/2$.

 Figure 3.17: Parameter values of the selected μ_λ^i , $i = 1, \dots, N$ during the greedy algorithm for $H = 1/2$.

 Figure 3.18: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $H=1/4$ and $\mu_\lambda = 1, \dots, 10$.

Figure 3.20 shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm, $i = 1, \dots, N$. Also in this case the distribution of the parameter values μ_2^i is similar to the previous cases.

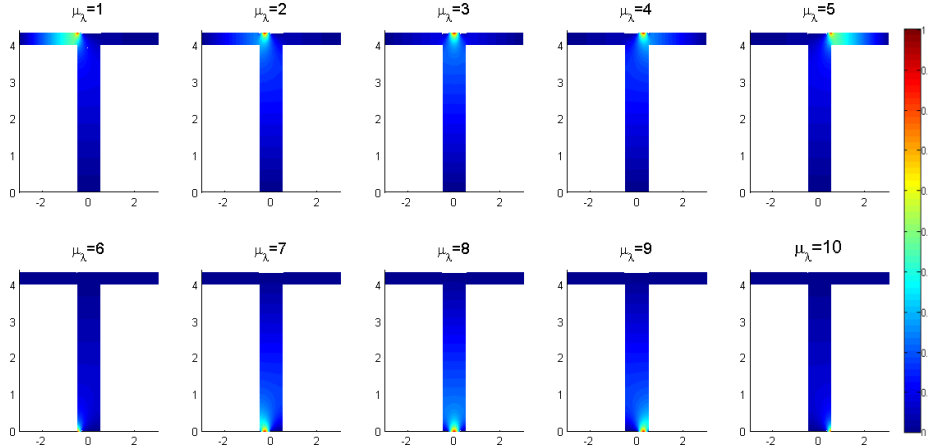


Figure 3.19: Solutions $u(\mu, \mu_\lambda)$ of (3.4.40), $\mu_\lambda = 1, 2, \dots, 10$, $H = 1/4$.

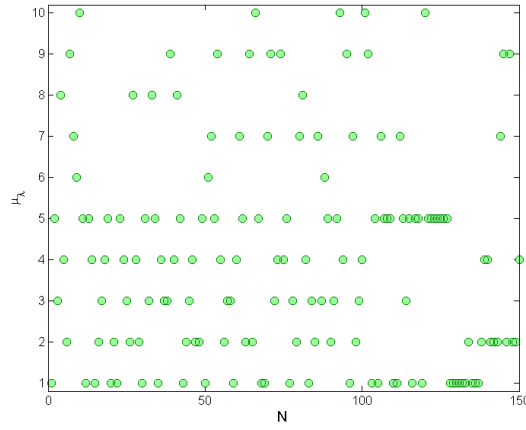


Figure 3.20: Parameter values of the selected μ_λ^i , $i = 1, \dots, N$ during the greedy algorithm for $H = 1/4$.

We note that in the three cases, the selected values of μ_λ have a bigger concentration in the interval $[1, \frac{\max(\mu_\lambda)}{2}]$, for which $\lambda(\mu_\lambda)$ is not zero along $\hat{\Gamma}_T$, that is the boundary close to the geometrical deformation (see the geometrical representation of Λ in Figure 3.5). In particular, in this set of functions, the number of selected basis with BCs equal to the Lagrangian basis corresponding to the extreme nodes of $\hat{\Gamma}_T$ is larger with respect to the others. These solutions are obviously more affected to the geometrical deformation of the domain, so that the greedy algorithm needs to select more basis referred to these value of μ_λ in correspondence of different values of μ_2 .

We can decrease the value of H progressively until we consider $H = h$, in this case we take one Lagrangian basis for each node on the interface and $\mu_\lambda \in \{1, \dots, 2N_\Gamma\}$. In theory, this is the case in which we can approximate in the best way the final solution, but we have a large range of the parameter values of μ_λ , this means that we increase both the computational complexity of the greedy algorithm and the number of the necessary reduced basis functions to reach a suitable accuracy. For these numerical tests we set 65 FE nodes belonging to the interface $\hat{\Gamma}_T$ as well as to the interface $\hat{\Gamma}_B$, so that if $H = h$ the maximum number of possible Lagrangian functions is 130. Figure 3.21 shows the average of the relative errors between the RB and FE solutions obtained by solving the local problem (3.4.40) and by using different values of H for the setting of the BCs. The errors are reported by varying the number of

reduced basis (computed by the greedy algorithm). We observe that by enlarging the range of possible values of μ_λ , the number of needed reduced basis to reach a suitable accuracy become bigger. For instance, we note that in the case of $\max \mu_\lambda = 130$, i.e. $H=h$ (the case of one Lagrange function for each interface nodes), the curve of the error drastically decrease when $N > 130$. This is due to the fact that a basis corresponding to a particular Lagrangian basis can not be well approximate by a combination of the solutions corresponding to the other independent Lagrangian basis (centered on the other different nodes of the interfaces). So that the minimum number of local reduced basis functions corresponds to the number of local Lagrangian basis functions used as BC.

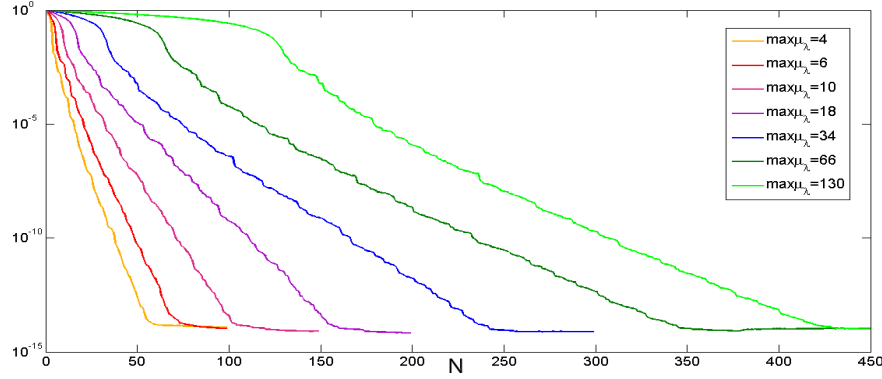


Figure 3.21: Average of relative error between the RB local solution of (3.4.40) and the FE one on a test of 1000 samples set.

3.6.1.2 The Fourier interface functions

With the aim of improving the offline stage in terms of complexity reduction and for decreasing the number of needed basis functions to reach a proper tolerance, we explore a second main option: the use of the Fourier functions associated to the local parametric BCs for the problem (3.4.40). Also in this case we have to fix a value (M) that defines which is the number of possible functions to consider on $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$. The function λ , when it is different from zero, represents the Fourier basis function defined on $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$. In particular, if we consider a set of M different local BCs, we define λ as follows: if $\mu_\lambda = 1, \dots, M/2$

$$\lambda(\mathbf{x}, \mu_\lambda) = \begin{cases} \cos(m(\mu_\lambda)\pi \mathbf{x}) & \mathbf{x} \in \hat{\Gamma}_T, m(\mu_\lambda) = \mu_\lambda - 1, \\ 0 & \mathbf{x} \in \hat{\Gamma}_B, \end{cases} \quad (3.6.1)$$

and if $\mu_\lambda = M/2 + 1, \dots, M$

$$\lambda(\mathbf{x}, \mu_\lambda) = \begin{cases} 0 & \mathbf{x} \in \hat{\Gamma}_T \\ \cos(m(\mu_\lambda)\pi \mathbf{x}) & \mathbf{x} \in \hat{\Gamma}_B, m(\mu_\lambda) = \mu_\lambda - M/2 - 1, \end{cases} \quad (3.6.2)$$

Figure 3.22 shows the function λ by varying the values of $\mu_\lambda \in \{1, \dots, 6\}$.

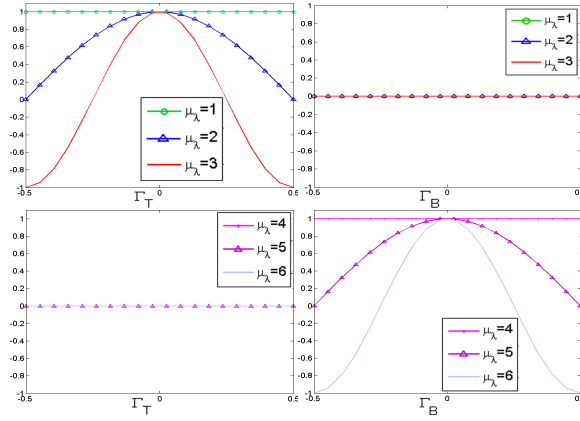


Figure 3.22: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $M=6$ and $\mu_\lambda = 1, \dots, 6$.

Figure 3.23 shows, for a fixed parameter μ , the solutions $u(\mu, \mu_\lambda)$ of (3.4.40) associated to the value $M = 6$ and $\mu_\lambda = 1, \dots, 6$.

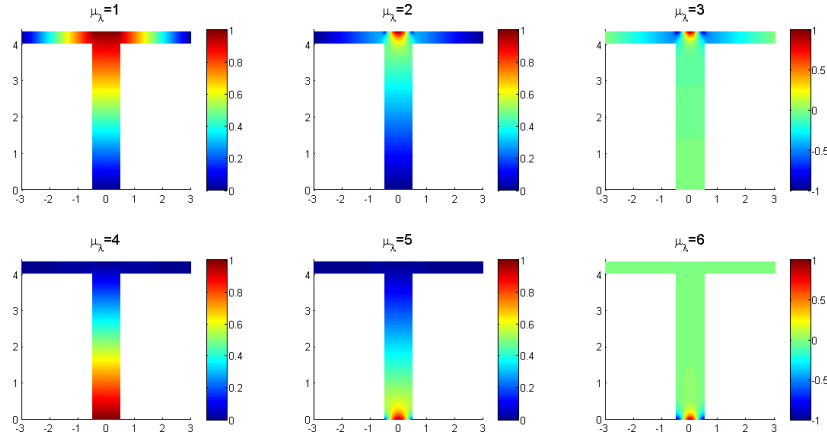


Figure 3.23: Solutions $u(\mu, \mu_\lambda)$ of (3.4.40) in correspondence of $\mu_\lambda = 1, \dots, 6$.

Figure 3.24 shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm with $M = 6$.

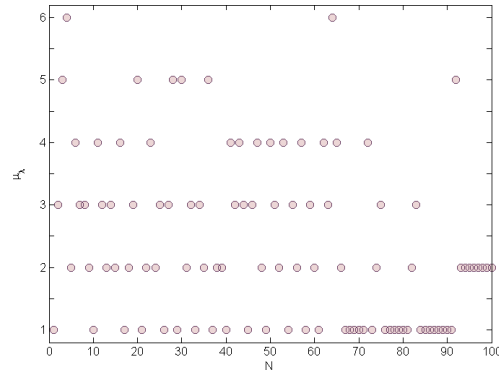


Figure 3.24: Parameter values of the selected μ_λ^i , $i = 1, \dots, 100$ during the greedy algorithm.

By increasing the quantity M , we add more functions to the previous set of BCs, as shown in Figure 3.25 by the plot of the function $\lambda(\mathbf{x}, \mu_\lambda)$, when it is non-zero, for $M = 10$ and consequently $\mu_\lambda = 1, \dots, M/2$. By following the same descriptions of the previous case, we show, for $M = 10$, both the set of basis functions and the distributions of the parameter values μ_λ^i , $i = 1, \dots, N$, in Figures 3.26 and 3.27.

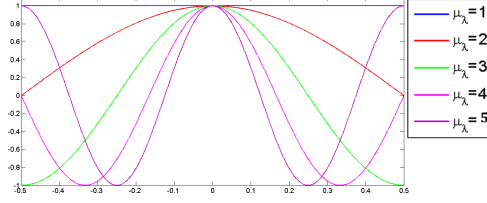


Figure 3.25: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $M=10$ and $\mu_\lambda = 1, \dots, 5$.

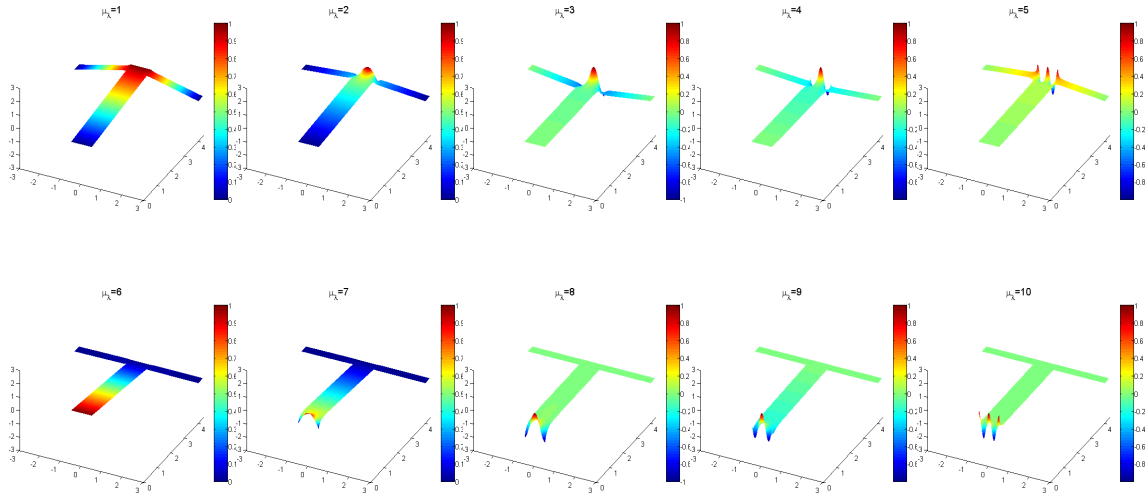


Figure 3.26: Solutions $u(\mu, \mu_\lambda)$ of (3.4.40) in correspondence of $\mu_\lambda = 1, \dots, 10$.

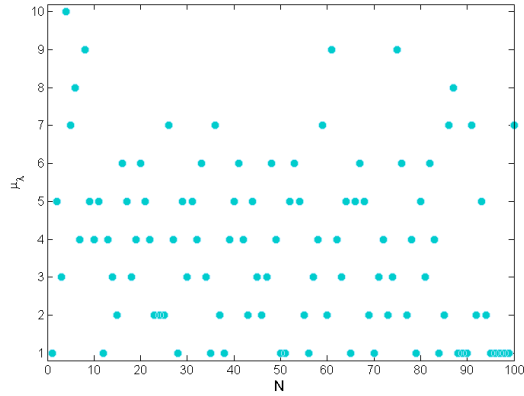


Figure 3.27: Parameter values of the selected μ_λ^i , $i = 1, \dots, 100$ during the greedy algorithm, $M=10$.

As last example, we show in Figure 3.28 the profiles of λ for $M = 16$ and in Figure 3.29 the parameter values distribution during the greedy algorithm in this case.

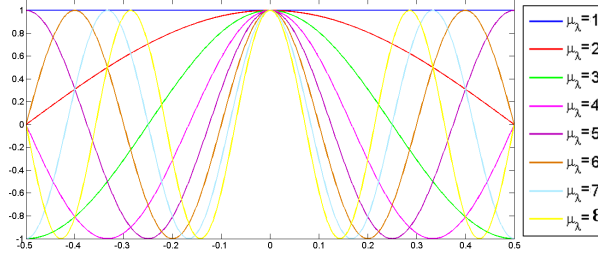


Figure 3.28: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $M=16$ and $\mu_\lambda = 1, \dots, 8$.

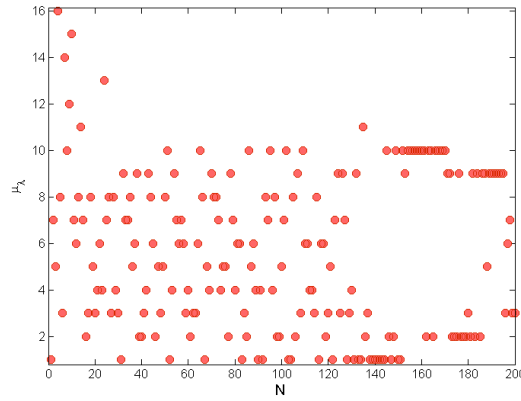


Figure 3.29: Parameter values of the selected μ_λ^i , $i = 1, \dots, 100$ during the greedy algorithm, $M=16$.

Also for that choice of Fourier boundary functions, if we increase the number M , we are considering a larger set of possible Fourier basis functions. Figure 3.30 shows the average of the relative errors between the RB local solution and the FE one obtained by solving the local problem (3.4.40) and by using different boundary “Fourier” conditions by varying the number of reduced basis (computed by the greedy algorithm). The same considerations of the Lagrangian case about the offline computation complexity hold this time; we observe that, due to the different description of $u(\boldsymbol{\mu}, \mu_\lambda)$ with respect to this kind of BCs, the number of reduced basis solutions that we need to reach an accurate solution is smaller than the case of the Lagrangian functions.

3.6.2 Global solution and accuracy results

Several numerical tests have been carried out in order to test all the different options introduced in the previous section. In the following subsections we report how the choice of λ (and M or H) affects the accuracy of the global approximated solution. The results will be shown in relation to different combination of (i) the set of local BCs, (ii) the number of nodes involved in the FE region and (iii) the number of used basis functions. Moreover, some considerations regarding the computational saving time gained by using the online RDF method with respect to the online FEM will follow.

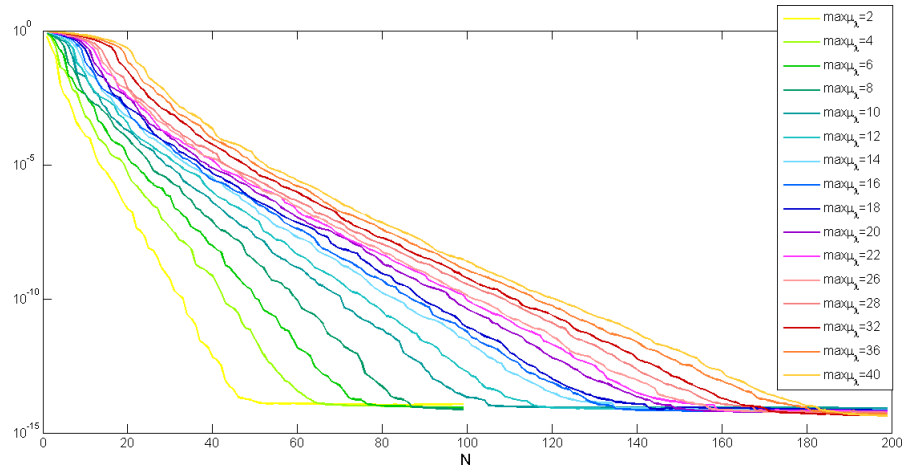


Figure 3.30: Average of relative errors between the RB local solution of (3.4.40) and the FE one on a test of 1000 samples set.

3.6.2.1 Varying the set of local BCs

We have performed, in different and independent offline stages, several greedy algorithms in order to build many reduced basis spaces. As mentioned, each reduced basis space represents a possible choice for applying the described RDF method. In particular, we have always used a fine mesh of 37505 nodes and 65 local interface nodes and we have considered 7 options for the Lagrangian piecewise functions, by setting $H = 1, 1/2, 1/4, 1/16, 1/32, 1/64$ and 20 options of the Fourier functions, by setting $M = 4, 6, 8, \dots, 40$. Then we have applied the RDF method by using the 26 different local reduced basis spaces on the 3 blocks network configuration to approximate the Laplace problem (3.1.1) by using 100 random parametric samples.

Figure 3.31 shows the average relative errors between the RDF solution and the finite element solution obtained as function of the number of local BCs (dependent on H) and the number of reduced basis functions used in each block (N), by using the Lagrangian piecewise functions set.

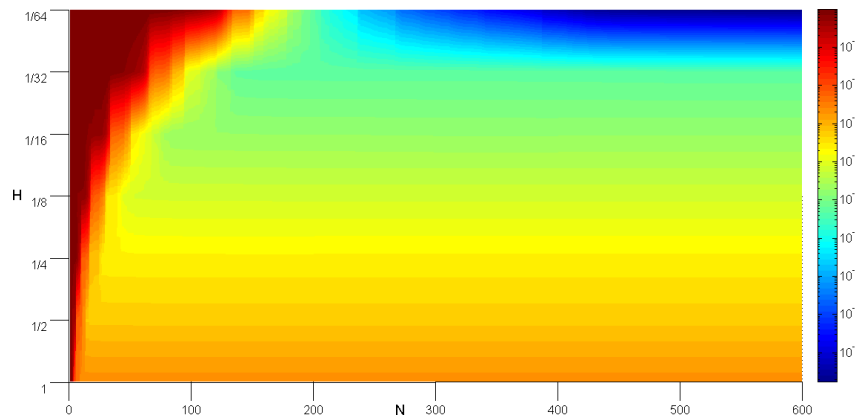


Figure 3.31: Average relative error between the RDF the FE solutions on a samples set of 100 by using different Lagrangian BCs.

The same comparison has been performed with the Fourier set of BCs. Precisely, we have built in different and independent offline stages 19 reduced basis spaces computed through the greedy algorithm and each time by imposing a different value of $M = \{4, \dots, 40\}$. Figure 3.32 shows the average relative errors between the RDF solution and the finite element solution obtained again as function of

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

the number of local BCs (dependent on M) and the number of reduced basis functions (N).

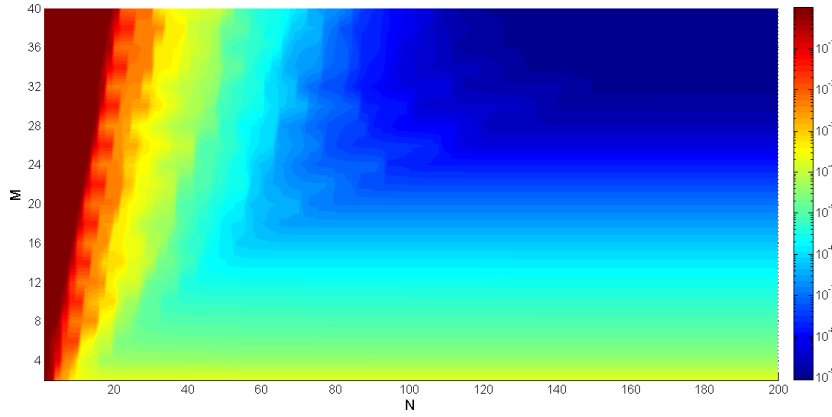


Figure 3.32: Average relative error between the RDF and the FE solutions on a samples set of 100 by using different Fourier BCs.

We note that, in both cases, for a fixed number of local BCs, the error decreases by increasing the number of local reduced basis functions and reaches a plateau that depends on the number M or H characterizing the choice of the reduced basis space. On the other hand, for a fixed number of local reduced basis functions the error diminishes by increasing the number of local BCs, however, if the number of local reduced basis functions is too small, the error increases as the RB is not large enough to accurately approximate solutions corresponding to all the different local BCs. Here the finite element region involves only the nodal value on the internal interfaces. Moreover, we observe that, at least for the problem considered, the Fourier basis used to recover the trace of the solution along the internal interfaces seems a more practical and effective choice for the RDF method. This choice allows to reach a reasonable level of accuracy by using a smaller number of basis functions respect to the choice of the Lagrangian piecewise functions. For instance, in order to achieve an accuracy of the order 10^{-6} , by using the Fourier option, we have to choose $M=16$ and $N=50$, while by using the Lagrange option the same accuracy is recoverable only by considering more than 200 basis functions and, at least, $H=1/32$. Figure 3.33 shows the RDF solution obtained by using Fourier BCs, $M = 6$ and $N = 40$ corresponding to the parameter values $\mu_1 = (4.3725, 9.7089)$, $\mu_2 = (4.3947, 9.5760)$, $\mu_3 = (2.0272, 4.9052)$.

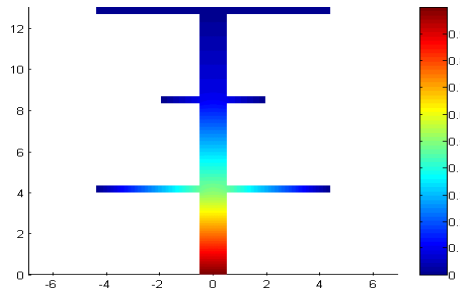


Figure 3.33: RDF Solution for a three blocks configuration corresponding to $\mu_1 = (4.3725, 9.7089)$, $\mu_2 = (4.3947, 9.5760)$, $\mu_3 = (2.0272, 4.9052)$, obtained by using Fourier BCs, $M = 6$, $N=40$.

3.6.2.2 Extending the finite element region

The second series of tests that we have performed is concerned with several extensions of the FE regions introduced in section 3.4.2. Figure 3.34 reports the average relative error between the RDF solution and

the FE solution by including in the FE region comprising about 4.32% nodes of the total number of mesh nodes of the domain. The values of the error have been computed by using the spaces of Fourier BCs and by varying the number of RB functions and the local BCs.

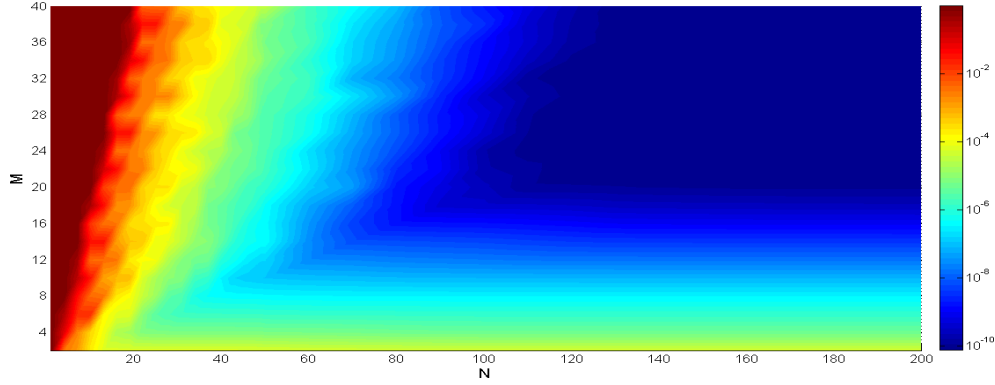


Figure 3.34: Average relative error between the RDF and FE solutions on a samples set of 100 by using local Fourier BCs and by including in the FE region 4.32% nodes of the total number of mesh nodes.

We observe that the error plot has a similar pattern than the one obtained with FE nodes considered only on the internal interfaces, with the difference that here the error values have a lower order of magnitude. For instance, this time, in order to reach an accuracy error of 10^{-6} it is necessary to choose $M=5$ and $N=25$. In the next section we discuss how the addition of further FE functions affects the computational time.

In order to visualize clearly (still in the case of Fourier option) the decreasing of the error by increasing the number of nodes involved in the FE region we report the Figure 3.35, it shows the relative error between the RDF solution and the FE solution by increasing the percentage of FE nodes with respect to the total and by varying the number of local BCs. This result shows that the expansion of FE nodes by adding nodes proximal to the internal interfaces at fixed number of local RB functions is particularly effective for a specific range of the number of local BCs ($5 \leq M \leq 22$).

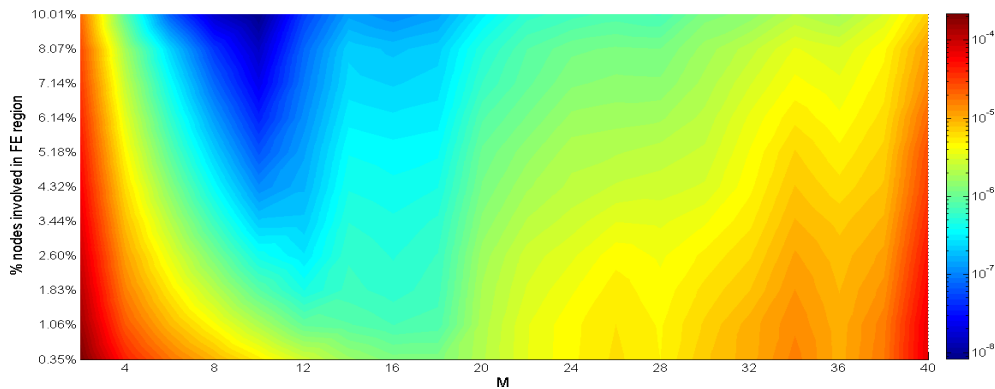


Figure 3.35: Average relative error between the RDF and FE solution on a samples set of 100 by using local Fourier BCs by fixing $N=50$ RB functions and by varying the number of BCs and the percentage of nodes involved in the FE regions.

Figure 3.36 shows the relative error between the RDF solution and the FE solution by increasing both the number of local RB functions and the number of FE nodes (by using the Fourier option and $M=10$). Therefore, by increasing the percentage of FE nodes is a viable strategy to decrease the error. When FE nodes are 100% of the total (limit case), RDF correspond to the full FE method on the global domain. The graphs relative to different number of local BC present similar patterns.

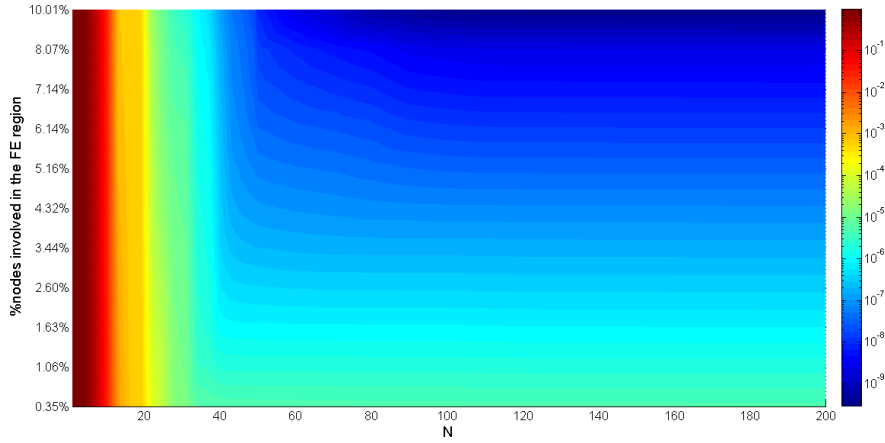


Figure 3.36: Average relative error between the RDF and FE solutions on a samples set of 100 by using local Fourier BCs, by fixing $M = 10$ and by varying the number of RB functions and the percentage of nodes involved in the FE regions.

Figure 3.37 shows the computational time of the RDF method as percentage of the FE method applied to the global problem. The computational time increases when increasing either the number of local RB functions and the number of FE functions.

We observe that it is possible to choose a good combination of the number of FE nodes and that of local RB functions in order to have a advantageous computational time and suitable accuracy results. The type of reduced basis solutions does not affect the computational time.

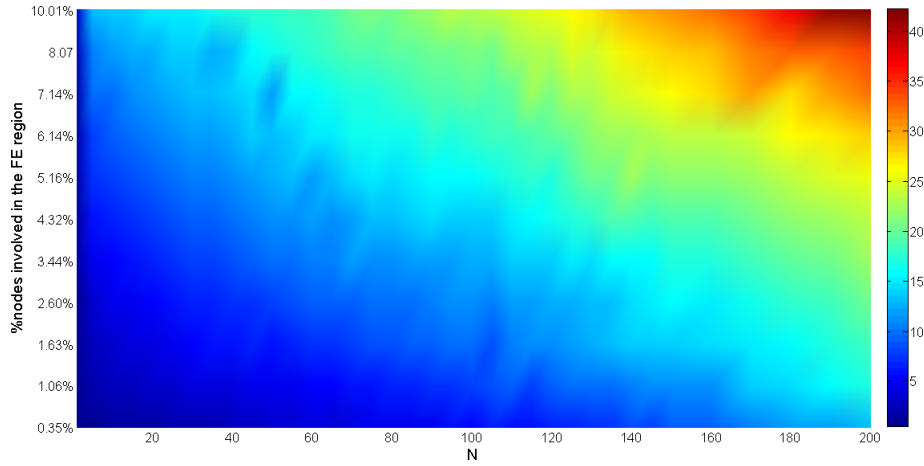


Figure 3.37: Computational time of the RDF method as percentage of the FE method applied to the global problem for a set of 100 tests by varying the number of FE nodes and the number of RB functions.

3.6.2.3 Increasing the number of blocks

The last train of tests has been performed by increasing the number of blocks on the computational domain, by maintaining the FE region equal only to the internal interfaces.

Figures 3.38, 3.39 and 3.40 represent the relative errors between the RDF solution and the FE solution by using the Fourier local BC with values $M = 6$, $M = 20$ and $M = 36$, respectively.

In the three Figures, the errors are related to several computational domains composed by $R = 3, \dots, 50$ domains and in dependence of the number of reduced basis functions.

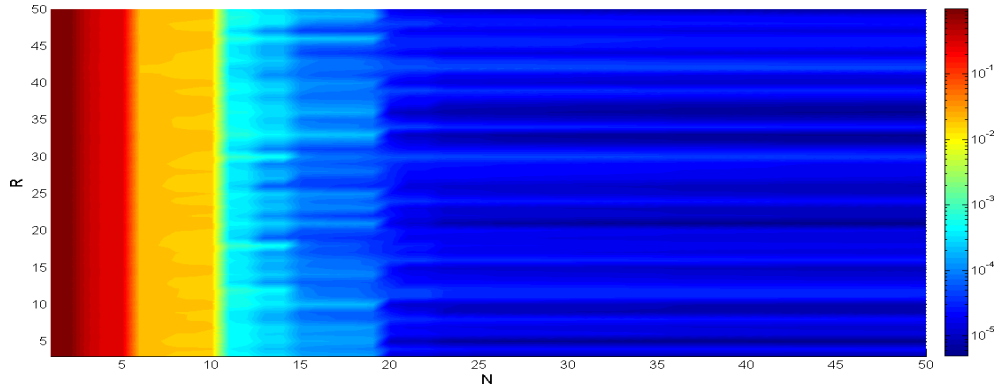


Figure 3.38: Relative error between the RDF and the FE solutions by varying the number of blocks (R) on the domain and the number of RB functions (N) with $M = 6$.

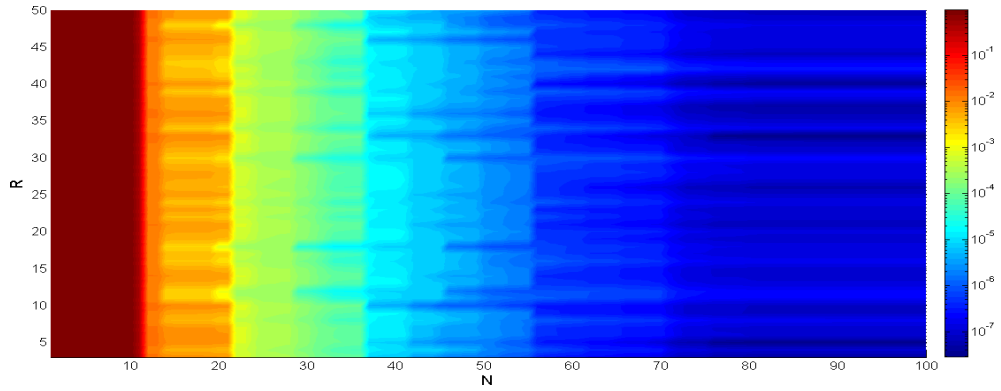


Figure 3.39: Relative error between the RDF solution and the FE solution by varying the number of blocks (R) on the domain and the number of RB functions (N) with $M = 20$.

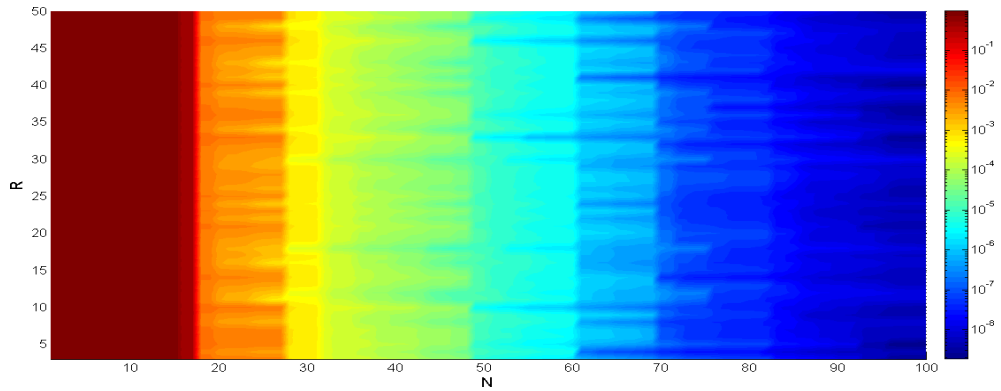


Figure 3.40: Relative error between the RDF solution and the FE solution by varying the number of blocks (R) on the domain and the number of RB functions (N) with $M = 32$.

We note that the order of magnitude of the error is not affected from the number of blocks composing the domain. Moreover, as already seen in the previous tests, in order to reach the same accuracy, if we use a larger set of local BCs, we need to increase the number of reduced basis functions. For instance, for reaching the relative error equal to 10^{-5} , we need 25 reduced basis functions if $M = 6$, 40 basis

Chapter 3. Reduced basis, Domain Decomposition and Finite element methods in a combined perspective

functions if $M = 20$, more than 50 basis functions if $M = 32$. On the other hand only if we use a larger set of local BCs we are able to reach a very low relative error.

Finally, we can observe that the attractive performances of the RDF method, regarding both the accuracy and the computational time, are maintained also in the case in which several blocks compose the computational domain.

Figure 3.41 shows the percentage of the computational time of the RDF time with respect to the FE method by varying both the number of blocks (R) and the number of RB functions (N).

Figure 3.42 shows the computational time in seconds of both methods. We remark that the type of reduced basis solutions (so the options of local BCs) does not affect the computational time.

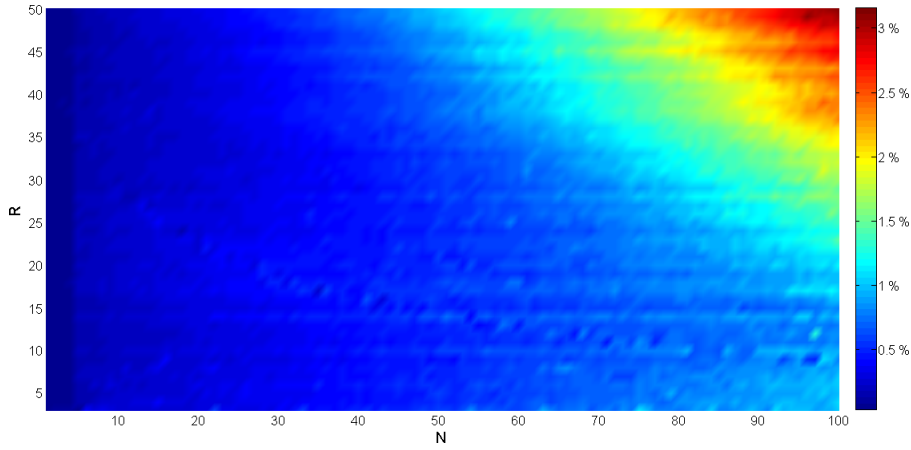


Figure 3.41: Computational time of the RDF method as percentage of the FE method applied to an increasing number of blocks on the domain and varying the number of reduced basis functions.

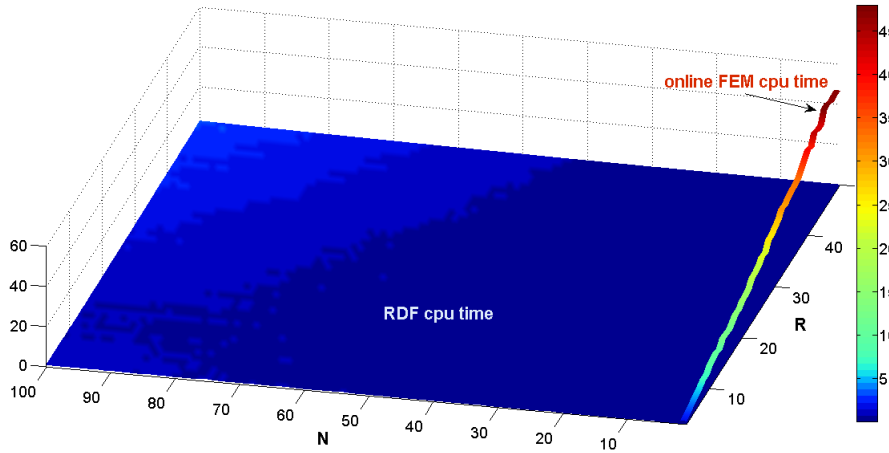


Figure 3.42: Computational time of the RDF method and of the FE method applied to an increasing number of blocks on the domain and varying the number of reduced basis functions.

3.7 Concluding remarks

In order to apply efficiently the RDF methodology, we started considering the number and the type of local BCs determining the space in which we are looking for the trace of the solution on the internal interfaces. We performed simulations both by choosing a piecewise linear Lagrangian basis and a Fourier hierarchical basis. Local greedy algorithm convergence is affected by the number of local BCs while the type of BC (Fourier or Lagrangian) does not affect sensibly the procedure. Particularly the number of local reduced basis N (necessary to achieve a prefixed tolerance) increases sensibly if we increase the number of local BCs. This behavior is due to the fact that by increasing the number of possible parametric BC profiles, the range of the parameter μ_λ increases as well, hence we need a bigger space of reduced basis functions to recover the possible solutions.

However, the kind of BCs chosen may affect the global RDF error. In this respect, the Fourier basis functions seem to be more efficient than the piecewise Lagrangian one in the sense that the global RDF accuracy is much better for small values of μ_λ .

From the computational point of view, if we consider a large range of μ_λ values, the *offline* computational time slows down the rate of convergence of the greedy algorithm. On the other hand, it does not affect directly the *online* computational time, but it influences the number of needed basis functions (that affects the *online* computational time).

In general, by increasing the number of reduced basis functions the error of the RDF method decreases but increases the *online* computational time (the size of the local RB submatrices leading to a larger algebraic linear system). Moreover we note that the RB submatrices are full.

The dimension of the FE region also affects both the accuracy and the *online* computational time of the method. In general, by increasing this dimension, the error of the RDF method decreases, but increases the *online* computational time (it means that we add to final linear system sparse matrices of dimensions equal to the number of nodes in the FE region).

Although the indication about the behavior of the method in response to a change of its parameters are valid in general it is clear from Section 3.6 that some caution should be taken when tuning the method. For example, it may be inefficient to increase the number of local BCs without increasing the number of reduced basis functions, as it would not produce any improvement on the accuracy of the RDF solution. The challenge of the method is to find the best combination of parameters (i. e. a reasonable trade-off) to reach a desired accuracy minimizing the *online* and *offline* computational times.

4 The RDF method for the solution of steady parametrized Stokes equations

In this chapter we extend the RDF method introduced in Chapter 3 for a scalar elliptic problem in order to solve incompressible viscous fluid flow problems modeled by the steady Stokes equations. The investigation and the method description carried out in Chapter 3 for a simple model governed by the Laplace equation represent an essential step to investigate the properties of the coupling approach and the role of the two types of local reduced basis space (*Fourier vs Lagrange*). The previous results induced us to select the Fourier option, since it allows to reach a better accuracy by using a reduced number of local basis functions and to deal with a less complex offline stage.

In order to increase gradually, step by step, the problem complexity, we describe the RDF method for the steady Stokes equation by considering a geometry similar to the one treated in Chapter 3 where the deformations are still obtained through an affine map. Moreover we show some preliminary results regarding the application of the RDF method for solving the steady Stokes problem. Several applications of the RDF method will be introduced in Chapter 5 by dealing with more complex geometries and three dimensional examples.

4.1 Problem Setting

Being interested in solving incompressible viscous fluid flow problems modeled by the steady Stokes equations in a domain with a geometrical parametrization given by a combination of repetitive geometries, we consider the computational domain composed of the union of non-overlapping R subdomains Ω_r , $r = 1, \dots, R$. For this problem we consider only the geometrical parameters, in the sense that each block can be seen as a deformation of a reference one $\hat{\Omega}_r$ through a suitable choice of a geometrical parameter μ^r , that defines the length of the fins of the blocks, so that, as before, $\Omega_r = \Omega_{\mu^r}$ and $\mathbf{x} = T(\hat{\mathbf{x}}, \mu^r)$, $\mathbf{x} \in \Omega_r, \hat{\mathbf{x}} \in \hat{\Omega}_r$, see Figure 4.1.

Figure 4.2 shows a possible configuration for the computational domain with $R = 3$, in this case the problem presents 3 physical parameters $\boldsymbol{\mu} = [\mu^1, \mu^2, \mu^3], \mu^i \in [3, 9]$, $\bar{\Omega}_{\boldsymbol{\mu}} = \cup_{r=1}^R \bar{\Omega}_r$.

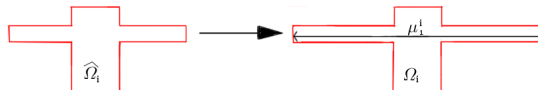


Figure 4.1: Scheme for the geometrical transformation of a single block.

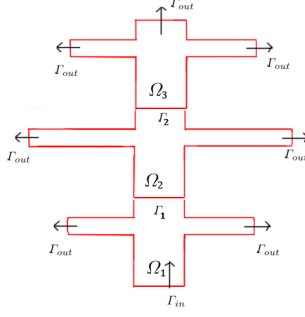


Figure 4.2: Computational domain composed by R=3 blocks.

We consider the following steady Stokes problem:

$$\begin{cases} -\nu \Delta \mathbf{u} + \nabla p = 0 & \text{in } \Omega_\mu, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega_\mu, \\ \mathbf{u} = 0 & \text{on } \Gamma_D, \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \boldsymbol{\sigma} & \text{on } \Gamma_{in}, \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{0} & \text{on } \Gamma_{out}, \end{cases} \quad (4.1.1)$$

where $\Gamma_D = \partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out})$, \mathbf{u} is the velocity field, p is the pressure, ν is the fluid viscosity and \mathbf{n} the normal unit vector to the domain boundary and $\boldsymbol{\sigma} = (0, -1)$.

We are interested in studying at least the velocity field and the pressure inside the whole computational domain by changing both the number of blocks composing the domain and the values of the parameters.

As already seen for the Stokes equations described in Chapter 2, Section 2.1, we introduce on Ω_μ the velocity space and the pressure space, respectively, as:

$$Y = \{\mathbf{v} \in (H^1(\Omega_\mu))^2 : \mathbf{v}|_{\Gamma_D} = \mathbf{0}\}, \quad M = L^2(\Omega_\mu).$$

Problem (4.1.1) in weak formulation reads: find $\mathbf{u} \in Y$, $p \in M$:

$$\begin{cases} a_\mu(\mathbf{u}, \mathbf{v}) + b_\mu(\mathbf{v}, p) = F_\mu(\mathbf{v}) & \forall \mathbf{v} \in Y, \\ b_\mu(\mathbf{u}, q) = 0 & \forall q \in M, \end{cases} \quad (4.1.2)$$

where

$$a_\mu(\mathbf{v}, \mathbf{w}) = \nu \int_{\Omega_\mu} \nabla \mathbf{v} \cdot \nabla \mathbf{w} \, d\Omega_\mu, \quad b_\mu(\mathbf{v}, q) = - \int_{\Omega_\mu} q(\nabla \cdot \mathbf{v}) \, d\Omega_\mu, \quad F_\mu(\mathbf{v}) = \int_{\Gamma_{in}} \boldsymbol{\sigma} \cdot \mathbf{v} \, d\Gamma. \quad (4.1.3)$$

4.1.1 Geometrical affine parametrization

As developed in the Chapter 3, Section 3.1.1 in the case of a scalar elliptic problem, also for the steady Stokes equations we can introduce a decoupling of the equations in the parameter dependent and independent parts. So that, we define the linear and bilinear forms on the reference geometry $\hat{\Omega}_r$. We consider the following piecewise affine map $T(\hat{\mathbf{x}}, \mu^r) : \hat{\Omega}_r \rightarrow \Omega_r$ such that $\mathbf{x} = C_{\mu^r} \hat{\mathbf{x}} + \mathbf{c}_{\mu^r}$, as shown in Figure 4.3. We denote with $\bar{\Omega} = \cup_{r=1}^R \bar{\Omega}_r$ the reference whole domain of $\bar{\Omega}_\mu = \cup_{r=1}^R \bar{\Omega}_r$.

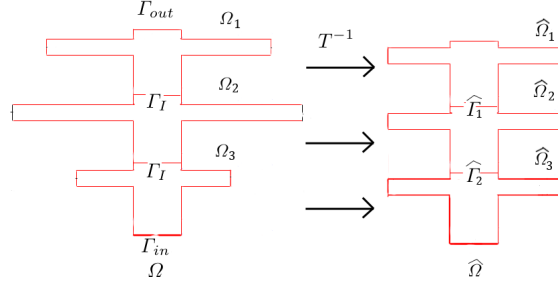


Figure 4.3: Geometrical scheme: from the deformed configuration to the reference blocks.

We can operate the following decompositions for the linear and bilinear forms:

$$\begin{aligned}
 a_{\boldsymbol{\mu}}(\mathbf{v}, \mathbf{w}) &= \int_{\Omega_{\boldsymbol{\mu}}} \nu \nabla \mathbf{v} \cdot \nabla \mathbf{w} \, d\mathbf{x} \\
 &= \sum_{r=1}^R \sum_{i,j=1}^2 \underbrace{\nu \left[(C_{\mu^r}^{-1} C_{\mu^r}^{-T}) |\det C_{\mu^r}| \right]_{ij}}_{\Theta_{q(i,j)}^r(\mu^r)} \underbrace{\int_{\hat{\Omega}_r} \frac{\partial \hat{v}_i}{\partial \hat{x}_j} \frac{\partial \hat{w}_i}{\partial \hat{x}_j} \, d\hat{\mathbf{x}}}_{a_{q(i,j)}^r(\hat{\mathbf{v}}, \hat{\mathbf{w}})} \, d\hat{\mathbf{x}} \\
 &= \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\hat{\mathbf{v}}, \hat{\mathbf{w}}).
 \end{aligned} \tag{4.1.4}$$

$$\begin{aligned}
 b_{\boldsymbol{\mu}}(\mathbf{v}, q) &= - \int_{\Omega_{\boldsymbol{\mu}}} q \nabla \cdot \mathbf{v} \, d\mathbf{x} = - \sum_{r=1}^R \int_{\hat{\Omega}_r} \hat{q} \hat{\nabla} \cdot (C_{\mu^r}^{-1} \hat{\mathbf{v}}) |\det C_{\mu^r}| \, d\hat{\mathbf{x}} \\
 &= - \sum_{r=1}^R \underbrace{\left[C_{\mu^r}^{-1} |\det C_{\mu^r}| \right]_{11}}_{\Phi_1^r(\mu^r)} \underbrace{\int_{\hat{\Omega}_r} \hat{q} \frac{\partial \hat{v}_1}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{b_1^r(\hat{\mathbf{v}}, \hat{q})} - \sum_{r=1}^R \underbrace{\left[C_{\mu^r}^{-1} |\det C_{\mu^r}| \right]_{12}}_{\Phi_2^r(\mu^r)} \underbrace{\int_{\hat{\Omega}_r} \hat{q} \frac{\partial \hat{v}_1}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{b_2^r(\hat{\mathbf{v}}, \hat{q})} \\
 &\quad - \sum_{r=1}^R \underbrace{\left[C_{\mu^r}^{-1} |\det C_{\mu^r}| \right]_{21}}_{\Phi_3^r(\mu^r)} \underbrace{\int_{\hat{\Omega}_r} \hat{q} \frac{\partial \hat{v}_2}{\partial \hat{x}_1} \, d\hat{\mathbf{x}}}_{b_3^r(\hat{\mathbf{v}}, \hat{q})} - \sum_{r=1}^R \underbrace{\left[C_{\mu^r}^{-1} |\det C_{\mu^r}| \right]_{22}}_{\Phi_4^r(\mu^r)} \underbrace{\int_{\hat{\Omega}_r} \hat{q} \frac{\partial \hat{v}_2}{\partial \hat{x}_2} \, d\hat{\mathbf{x}}}_{b_4^r(\hat{\mathbf{v}}, \hat{q})} \\
 &= \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\hat{q}, \hat{\mathbf{v}})
 \end{aligned} \tag{4.1.5}$$

$$F_{\boldsymbol{\mu}}(\mathbf{v}) = \int_{\Gamma_{\boldsymbol{\mu}}} \boldsymbol{\sigma} \cdot \mathbf{v} \, d\mathbf{x} = \underbrace{|\det C_{\mu^1}|}_{\Psi(\mu^1)} \underbrace{\int_{\hat{\Gamma}_{in}} \hat{\boldsymbol{\sigma}} \cdot \hat{\mathbf{v}} \, d\hat{\mathbf{x}}}_{F(\hat{\mathbf{v}})}. \tag{4.1.6}$$

Due to the affine deformation, the matrices $C_{\mu^r}^{-1}, C_{\mu^r}^{-T}$ and the determinant $|\det C_{\mu^r}|$ are dependent only on μ^r , thus it is possible to separate the terms $\Theta_q^r(\mu^r), \Phi_q^r(\mu^r), \Psi(\mu^1), r = 1, \dots, R; q = 1, \dots, 4$ that depend on the parameters, from the integrals.

Finally we define the following spaces:

$$\hat{Y} = \left\{ \mathbf{v} \in (H^1(\hat{\Omega}_{\boldsymbol{\mu}}))^2 : \mathbf{v}|_{\hat{\Gamma}_D} = \mathbf{0} \right\}, \quad \hat{M} = L^2(\hat{\Omega}_{\boldsymbol{\mu}}).$$

and the Stokes problem reads: find $\hat{\mathbf{u}} \in \hat{Y}$ and $\hat{p} \in \hat{M}$ such that $\forall \hat{\mathbf{v}} \in \hat{Y}$ and $\forall \hat{q} \in \hat{M}$:

$$\begin{cases} \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\hat{\mathbf{u}}, \hat{\mathbf{v}}) + \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\hat{\mathbf{v}}, \hat{p}) = \Psi(\mu^1) F(\hat{\mathbf{v}}), \\ \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\hat{\mathbf{u}}, \hat{q}) = 0. \end{cases} \quad (4.1.7)$$

4.2 Finite element approximation of the problem

We give here a brief introduction about the finite element approximation of the solution $(\hat{\mathbf{u}}, \hat{p})$ of the problem (4.1.7). From now, for sake of simplicity, we omit the hat above the functions, but we keep it only to distinguish the deformed domain from the reference one. We consider the two set of subspaces $Y_h \subset \hat{Y}$ and $M_h \subset \hat{M}$, dependent on a positive parameter h , with finite dimension N_h^Y and N_h^M . The approximation of problem (4.1.7) is: find $\mathbf{u}_h \in Y_h$, $p_h \in M_h$ such that $\forall \mathbf{v}_h \in Y_h$ and $\forall q_h \in M_h$:

$$\begin{cases} \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\mathbf{u}_h, \mathbf{v}_h) + \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{v}_h, p_h) = \Psi(\mu^1) F(\mathbf{v}_h), \\ \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{u}_h, q_h) = 0. \end{cases} \quad (4.2.1)$$

We denote with $\{\boldsymbol{\phi}_j, j = 1, \dots, N_h^Y\}$ a basis of Y_h and $\{\xi_j, j = 1, \dots, N_h^M\}$ one of M_h . We project \mathbf{u}_h and p_h on these bases:

$$\mathbf{u}_h = \sum_{j=1}^{N_h^Y} u_j \boldsymbol{\phi}_j, \quad p_h = \sum_{j=1}^{N_h^M} p_j \xi_j, \quad (4.2.2)$$

where u_j and p_j are the unknown coefficients of \mathbf{u}_h and p_h on bases of Y_h and M_h . Problem (4.2.1) is therefore equivalent to: find $u_j, j = 1, \dots, N_h^Y$ and $p_j, j = 1, \dots, N_h^M$ such that for $i = 1, \dots, N_h^Y$, $m = 1, \dots, N_h^M$:

$$\begin{cases} \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\mu^r) \sum_{j=1}^{N_h^Y} u_j a_q^r(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i) - \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) \sum_{j=1}^{N_h^M} p_j b_q^r(\boldsymbol{\phi}_i, \xi_j) = -\Psi(\mu^1) F(\boldsymbol{\phi}_i), \\ \sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) \sum_{j=1}^{N_h^Y} u_j b_q^r(\boldsymbol{\phi}_j, \xi_m) = 0, \end{cases} \quad (4.2.3)$$

which leads to the linear system:

$$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbb{F} \\ 0 \end{bmatrix}, \quad (4.2.4)$$

where $[\mathbb{A}]_{ij} = \sum_{r=1}^R \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i)$, $[\mathbb{B}]_{im} = -\sum_{r=1}^R \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_i, \xi_m)$, $\mathbb{D} = \mathbb{B}^T$ and $[\mathbb{F}]_i = -\Psi(\mu^1) F(\boldsymbol{\phi}_i)$.

The finite element spaces are built on a triangulation \mathcal{T}_h of $\hat{\Omega}$ with K being the generic element of \mathcal{T}_h . We use piecewise quadratic and linear functions, respectively, to construct the subspaces Y_h and M_h [73]:

$$Y_h = (X_h^2)^2, \quad X_h^2 = \{v_h \in C^0(\hat{\Omega}) : v_h|_K \in \mathbb{P}_2 \quad \forall K \in \mathcal{T}_h\} \cap H_{0\Gamma_D}^1(\hat{\Omega}), \quad (4.2.5)$$

$$M_h = X_h^1 \equiv \{v_h \in C^0(\hat{\Omega}) : v_h|_K \in \mathbb{P}_1 \quad \forall K \in \mathcal{T}_h\}. \quad (4.2.6)$$

4.3 The non-overlapping Domain Decomposition Method

Since we have already investigated the domain decomposition approach for the Laplace problem in Chapter 3, here we consider directly the problem in a domain composed by a generic number R of subdomains $\hat{\Omega}_r$ (with parametric dependence), by assuming that the maximum number of internal interfaces involved in each subdomain is 2. Let us refer, for example, to the Figure 4.3 and denote $\hat{\Gamma}_r$ be the interface separating $\hat{\Omega}_r$ and $\hat{\Omega}_{r+1}$ and $\hat{\Gamma}_I = \cup_{r=1}^{R-1} \hat{\Gamma}_r$.

We start by considering the finite element approximation of problem (4.1.1), by introducing a partition of the nodes triangulation of the global mesh on $\hat{\Omega}$ and the corresponding finite element functions defined on $\hat{\Omega}$:

- ϕ_j^r , $j = 1, \dots, N_r^Y$, and ξ_i^r , $i = 1, \dots, N_r^M$, are the basis Lagrangian functions associated to the nodes of the domain $\hat{\Omega}_r \setminus \hat{\Gamma}_I$, $r = 1, \dots, R$.
- $\phi_j^{\Gamma_r}$, $j = 1, \dots, N_{\Gamma_r}^Y$, and $\xi_j^{\Gamma_r}$, $i = 1, \dots, N_{\Gamma_r}^M$, are those associated to the nodes on the interfaces $\hat{\Gamma}_r$, $r = 1, \dots, R-1$.

The Galerkin approximation of problem (4.1.1) on $\hat{\Omega}$ can be written as: find $(\mathbf{u}_h, p_h) \in Y_h \times M_h$ such that

$$\begin{cases} a_\mu(\mathbf{u}_h, \phi_i^r) + b_\mu(\phi_i^r, p_h) &= F_\mu(\phi_i^r) & i = 1, \dots, N_r^Y, r = 1, \dots, R, \\ b_\mu(\mathbf{u}_h, \xi_i^r) &= 0 & i = 1, \dots, N_r^M, r = 1, \dots, R, \\ a_\mu(\mathbf{u}_h, \phi_i^{\Gamma_r}) + b_\mu(\phi_i^{\Gamma_r}, p_h) &= F_\mu(\phi_i^{\Gamma_r}) & i = 1, \dots, N_{\Gamma_r}^Y, r = 1, \dots, R-1, \\ b_\mu(\mathbf{u}_h, \xi_i^{\Gamma_r}) &= 0 & i = 1, \dots, N_{\Gamma_r}^M, r = 1, \dots, R-1. \end{cases} \quad (4.3.1)$$

Where \mathbf{u}_h and p_h can be defined as:

$$\begin{aligned} \mathbf{u}_h(\mathbf{x}) &= \sum_{r=1}^R \sum_{j=1}^{N_r^Y} \mathbf{u}_h(\mathbf{x}_j^r) \phi_j^r(\mathbf{x}) + \sum_{r=1}^{R-1} \sum_{j=1}^{N_{\Gamma_r}^Y} \mathbf{u}_h(\mathbf{x}_j^{\Gamma_r}) \phi_j^{\Gamma_r}(\mathbf{x}), \\ p_h(\mathbf{x}) &= \sum_{r=1}^R \sum_{j=1}^{N_r^M} p_h(\mathbf{x}_j^r) \xi_j^r(\mathbf{x}) + \sum_{r=1}^{R-1} \sum_{j=1}^{N_{\Gamma_r}^M} p_h(\mathbf{x}_j^{\Gamma_r}) \xi_j^{\Gamma_r}(\mathbf{x}), \end{aligned} \quad (4.3.2)$$

and \mathbf{x}_j^r and $\mathbf{x}_j^{\Gamma_r}$ are the nodes belonging to $\hat{\Omega}_r \setminus \hat{\Gamma}_I$ and Γ_r , respectively.

Then, by following the DD approach, the solution (\mathbf{u}_h, p_h) of (4.1.1) is such that $\mathbf{u}_h|_{\hat{\Omega}_r} = \mathbf{u}_r$, $p_h|_{\hat{\Omega}_r} = p_r$ for $r = 1, \dots, R$ where (\mathbf{u}_r, p_r) is the solution of the problem

$$\begin{cases} -\nu \Delta \mathbf{u}_r + \nabla p_r &= 0 & \text{in } \hat{\Omega}_r, \\ \nabla \cdot \mathbf{u}_r &= 0 & \text{in } \hat{\Omega}_r, \\ \nu \frac{\partial \mathbf{u}_r}{\partial \mathbf{n}} - p_r \mathbf{n} &= \boldsymbol{\sigma} & \text{on } \hat{\Gamma}_{in}, \\ \nu \frac{\partial \mathbf{u}_r}{\partial \mathbf{n}} - p_r \mathbf{n} &= \mathbf{0} & \text{on } \hat{\Gamma}_{out}, \end{cases} \quad (4.3.3)$$

with interface conditions:

$$\mathbf{u}_r = \mathbf{u}_{r+1} \quad \text{and} \quad v \frac{\partial \mathbf{u}_r}{\partial \mathbf{n}_r} - p_r \mathbf{n}_r = -v \frac{\partial \mathbf{u}_{r+1}}{\partial \mathbf{n}_{r+1}} + p_{r+1} \mathbf{n}_{r+1} \quad \text{on } \hat{\Gamma}_{Ir}, r = 1, \dots, R, \quad (4.3.4)$$

where \mathbf{n}_i is the outgoing normal unit vector to $\hat{\Omega}_i$ on $\hat{\Gamma}_i$ and

$$\hat{\Gamma}_{Ir} = \begin{cases} \hat{\Gamma}_r, & \text{if } r = 1, \\ \hat{\Gamma}_{r-1} \cup \hat{\Gamma}_r, & \text{if } r = 1, \dots, R-1, \\ \hat{\Gamma}_{r-1}, & \text{if } r = R. \end{cases} \quad (4.3.5)$$

Thanks to the affine decompositions (4.1.4), (4.1.5) and (4.1.6) and the nodes division of the mesh, problem (4.3.1) can be written as: find $(\mathbf{u}_h, p_h) \in Y_h \times M_h$:

$$\left\{ \begin{array}{ll} \sum_{q=1}^4 [\Theta_q^r(\mu^r) a_q^r(\mathbf{u}_h, \boldsymbol{\phi}_i^r) + \Phi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_i^r, p_h)] & = \Psi(\mu^1) F(\boldsymbol{\phi}_i^r) \quad i = 1, \dots, N_r^Y, \\ \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{u}_h, \boldsymbol{\xi}_i^r) & = 0 \quad i = 1, \dots, N_r^M, \\ & r = 1, \dots, R, \\ \sum_{k=r}^{r+1} \left[\sum_{q=1}^4 [\Theta_q^k(\mu^k) a_q^k(\mathbf{u}_h, \boldsymbol{\phi}_i^{\Gamma_r}) + \Phi_q^k(\mu^k) b_q^k(\boldsymbol{\phi}_i^{\Gamma_r}, p_h)] \right] & = \Psi(\mu^1) F(\boldsymbol{\phi}_i^{\Gamma_r}) \quad i = 1, \dots, N_{\Gamma_r}^Y, \\ \sum_{k=r}^{r+1} \left[\sum_{q=1}^4 \Phi_q^k(\mu^k) b_q^k(\mathbf{u}_h, \boldsymbol{\xi}_i^{\Gamma_r}) \right] & = 0 \quad i = 1, \dots, N_{\Gamma_r}^M, \\ & r = 1, \dots, R-1. \end{array} \right. \quad (4.3.6)$$

By definition, the velocity and the stresses are continuous on $\hat{\Gamma}_I$, therefore the discretized solution automatically satisfies the continuity condition of the multi-domain formulation.

By including this decomposition in problem (4.3.6), we obtain the algebraic form, for $r = 1, \dots, R, k = 1, \dots, R-1$:

$$\left\{ \begin{array}{ll} \mathbb{A}_{rr} \mathbf{u}_r + \mathbb{B}_{rr} \mathbf{p}_r + \mathbb{A}_{r\Gamma_k} \boldsymbol{\lambda}_u^k + \mathbb{B}_{r\Gamma_k} \boldsymbol{\lambda}_p^k & = \mathbb{F}_r, \\ \mathbb{D}_{rr} \mathbf{u}_r + \mathbb{D}_{r\Gamma_k} \boldsymbol{\lambda}_u^k & = 0, \\ \mathbb{A}_{\Gamma_k r} \mathbf{u}_r + \mathbb{A}_{\Gamma_k r+1} \mathbf{u}_{r+1} + \mathbb{B}_{\Gamma_k r} \mathbf{p}_r + \mathbb{B}_{\Gamma_k r+1} \mathbf{p}_{r+1} \\ + [\mathbb{A}_{\Gamma_k \Gamma_k}^r + \mathbb{A}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_u^k + [\mathbb{B}_{\Gamma_k \Gamma_k}^r + \mathbb{B}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_p^k & = \mathbb{F}_{\Gamma_k}^r + \mathbb{F}_{\Gamma_k}^{r+1}, \\ \mathbb{D}_{\Gamma_k r} \mathbf{u}_r + \mathbb{D}_{\Gamma_k r+1} \mathbf{u}_{r+1} + [\mathbb{D}_{\Gamma_k \Gamma_k}^r + \mathbb{D}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_u^k & = 0, \end{array} \right. \quad (4.3.7)$$

where for $r = 1, \dots, R$, for $k = 1, \dots, R-1$ we define for $i, j = 1, \dots, N_r^Y$, $m = 1, \dots, N_r^M$, $\tilde{i}, \tilde{j} = 1, \dots, N_{\Gamma_k}^Y$, $\tilde{m}, \tilde{n} = 1, \dots, N_{\Gamma_k}^M$:

$$[\mathbf{u}_r]_j = \mathbf{u}_h(\mathbf{x}_j^r), [\mathbf{p}_r]_m = p_h(\mathbf{x}_m^r), [\boldsymbol{\lambda}_u^k]_{\tilde{j}} = \mathbf{u}_h(\mathbf{x}_{\tilde{j}}^{\Gamma_k}), [\boldsymbol{\lambda}_p^k]_{\tilde{m}} = p_h(\mathbf{x}_{\tilde{m}}^{\Gamma_k}),$$

$$[\mathbb{A}_{rr}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_j^r, \boldsymbol{\phi}_i^r), [\mathbb{B}_{rr}]_{i,m} = \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_i^r, \boldsymbol{\xi}_m^r), \mathbb{D}_{rr} = \mathbb{B}_{rr}^T,$$

$$\begin{aligned}
 [\mathbb{A}_{r\Gamma_k}]_{i,\tilde{j}} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_i^{\Gamma_k}, \boldsymbol{\phi}_{\tilde{j}}^r), \quad [\mathbb{B}_{r\Gamma_k}]_{i,\tilde{m}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_i^r, \xi_{\tilde{m}}^{\Gamma_k}), \\
 [\mathbb{D}_{r\Gamma_k}]_{\tilde{i},m} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \xi_m^r), \quad \mathbb{D}_{\Gamma_k r} = \mathbb{B}_{r\Gamma_k}^T, \mathbb{B}_{\Gamma_k r} = \mathbb{D}_{r\Gamma_k}^T, \\
 [\mathbb{A}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{j}} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}), \quad [\mathbb{B}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{m}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \xi_{\tilde{m}}^{\Gamma_k}), \\
 \mathbb{D}_{\Gamma_k \Gamma_k}^r &= \mathbb{B}_{\Gamma_k \Gamma_k}^{rT}, \quad [\mathbb{F}_r]_j = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\boldsymbol{\phi}_j^r), \quad [\mathbb{F}_{\Gamma_k}^r]_{\tilde{j}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}).
 \end{aligned}$$

By defining:

$$\mathbb{M}_r = \begin{bmatrix} \mathbb{A}_{rr} & \mathbb{B}_{rr} \\ \mathbb{D}_{rr} & \mathbf{0} \end{bmatrix}, \mathbb{M}_{r\Gamma_k} = \begin{bmatrix} \mathbb{A}_{r\Gamma_k} & \mathbb{B}_{r\Gamma_k} \\ \mathbb{D}_{r\Gamma_k} & \mathbf{0} \end{bmatrix}, \mathbb{M}_{\Gamma_k \Gamma_k} = \begin{bmatrix} \mathbb{A}_{\Gamma_k \Gamma_k} & \mathbb{B}_{\Gamma_k \Gamma_k} \\ \mathbb{D}_{\Gamma_k \Gamma_k} & \mathbf{0} \end{bmatrix}, \quad (4.3.8)$$

the final linear system is:

$$\begin{bmatrix} \mathbb{M}_{rr} & \mathbb{M}_{r\Gamma} \\ \mathbb{M}_{r\Gamma}^T & \mathbb{M}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbb{F}_{rr} \\ \mathbb{F}_{\Gamma} \end{bmatrix}, \quad (4.3.9)$$

where

$$\begin{aligned}
 \mathbb{M}_{rr} &= \begin{bmatrix} \mathbb{M}_1 & & & \\ & \mathbb{M}_2 & & \\ & & \ddots & \\ & & & \mathbb{M}_R \end{bmatrix}, \quad \mathbb{M}_{r\Gamma} = \begin{bmatrix} \mathbb{M}_{1\Gamma_1} & \mathbf{0} & & & \\ \mathbb{M}_{2\Gamma_1} & \mathbb{M}_{2\Gamma_2} & \mathbf{0} & & \\ \mathbf{0} & \mathbb{M}_{3\Gamma_2} & \mathbb{M}_{3\Gamma_3} & \mathbf{0} & \\ & \ddots & \ddots & \ddots & \mathbf{0} \\ & & \mathbf{0} & \mathbb{M}_{R-1\Gamma_{R-2}} & \mathbb{M}_{R-1\Gamma_{R-1}} \\ & & & \mathbf{0} & \mathbb{M}_{R\Gamma_{R-1}} \end{bmatrix}, \\
 \mathbb{M}_{\Gamma\Gamma} &= \begin{bmatrix} \mathbb{M}_{\Gamma_1\Gamma_1}^1 + \mathbb{M}_{\Gamma_1\Gamma_1}^2 & & & \\ & \mathbb{M}_{\Gamma_2\Gamma_2}^2 + \mathbb{M}_{\Gamma_2\Gamma_2}^3 & & \\ & & \ddots & \\ & & & \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix},
 \end{aligned}$$

$$\mathbf{s}_r = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \lambda_u^1 \\ \vdots \\ \lambda_u^{R-1} \\ \lambda_p^1 \\ \vdots \\ \lambda_p^{R-1} \end{bmatrix}, \mathbb{F}_{rr} = \begin{bmatrix} \mathbb{F}_1 \\ \vdots \\ \mathbb{F}_R \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbb{F}_\Gamma = \begin{bmatrix} \mathbb{F}_{\Gamma_1}^1 + \mathbb{F}_{\Gamma_1}^2 \\ \vdots \\ \mathbb{F}_{\Gamma_{R-1}}^{R-1} + \mathbb{F}_{\Gamma_{R-1}}^R \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The matrix of the linear system (4.3.9) is constituted by 4 sub-blocks: \mathbb{M}_{rr} , $\mathbb{M}_{\Gamma\Gamma}$, $\mathbb{M}_{r\Gamma}$ and $\mathbb{M}_{\Gamma r}^T$ coming from, respectively, the equations regarding the internal nodes of the R subdomains, the interface nodes between different subdomains and partially the internal nodes and partially the interface nodes.

4.4 The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

We extend here the strategy to couple the finite element method and the reduced basis method in the framework of non-overlapping domain decomposition for the Stokes equations.

As before, by referring to Figure 3.5, we consider the computational domain $\hat{\Omega}$ composed by disjoint non-overlapped R subdomains $\hat{\Omega}_r$, we denote with $\hat{\Gamma}_r$ the internal interface between $\hat{\Omega}_r$ and $\hat{\Omega}_{r+1}$.

We can define the maps that link the subdomain Ω_r with the reference subdomains $\hat{\Omega}_r$ and their reference shapes Λ_k (see Figure 3.5 in Chapter 3).

We define $\hat{\mathbf{x}} = \mathcal{T}_{kr}(\hat{\mathbf{x}}) = \hat{\mathbf{x}} + \tau_r$, $\hat{\mathbf{x}} \in \Lambda_k$, $\hat{\mathbf{x}} \in \hat{\Omega}_r$ (in the case of the geometry depicted in Figure 3.5 there is only one reference shape Λ).

We define the velocity reduced basis space \hat{Y}_{RB}^k and the pressure reduced space \hat{M}_{RB}^k for each Λ_k , $k = 1, \dots, K$, built by the solution of proper local Stokes problems defined on Λ_k (we address the detailed description to Section 4.4.2).

$$\hat{Y}_{RB}^k = \{\mathbf{w}_i^k, i = 1, \dots, N_k^Y\}, \hat{M}_{RB}^k = \{p_i^k, i = 1, \dots, N_k^M\}. \quad (4.4.1)$$

where $N_k^Y \ll N_h^Y$, $N_k^M \ll N_h^M$. The functions \mathbf{w}_i^k , p_i^k are defined locally on each Λ_k , we map them into $\hat{\Omega}_r$ and extend them to zero on $\hat{\Omega}_l$, if $l \neq r$, and on the internal interfaces $\hat{\Omega}_{FE} = \cup_{r=1}^{R-1} \hat{\Gamma}_r$, so that for $r = 1, \dots, R$ and $i = 1, \dots, N_k^Y$, $m = 1, \dots, N_k^M$, $\mathbf{v}_i^r \in V_h$, $q_m^r \in M_h$ and:

$$\mathbf{v}_i^r(\mathbf{x}_j) = \mathbf{w}_i^k(\mathcal{T}_{kr}(\hat{\mathbf{x}}_j)), \text{ if } \mathbf{x}_j \in \hat{\Omega}_r \setminus \hat{\Omega}_{FE} \quad \mathbf{v}_i^r(\mathbf{x}_j) = 0, \text{ if } \mathbf{x}_j \in \hat{\Omega}_l \cup \hat{\Omega}_{FE}, l \neq r. \quad (4.4.2)$$

$$q_m^r(\mathbf{x}_j) = p_m^k(\mathcal{T}_{kr}(\hat{\mathbf{x}}_j)), \text{ if } \mathbf{x}_j \in \hat{\Omega}_r \setminus \hat{\Omega}_{FE} \quad q_m^r(\mathbf{x}_j) = 0, \text{ if } \mathbf{x}_j \in \hat{\Omega}_l \cup \hat{\Omega}_{FE}, l \neq r. \quad (4.4.3)$$

We can define on each $\hat{\Omega}_r$ the following spaces:

$$Y_{RB}^r = \{\mathbf{v}_i^r, i = 1, \dots, N_k^Y\}, M_{RB}^r = \{q_i^r, i = 1, \dots, N_k^M\}. \quad (4.4.4)$$

In this case we have only one reference shape, so we denote with N^Y and N^M the number of velocity and pressure functions.

We consider now a subspace of the finite element spaces (4.2.5) composed by the finite element

4.4. The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

functions corresponding to the nodes \mathbf{x}_j^Γ on $\hat{\Omega}_{FE}$:

$$Y_{FE}^\Gamma \equiv \{\boldsymbol{\phi}_j^\Gamma \in Y_h, j = 1, \dots, N_\Gamma^Y\}, M_{FE}^\Gamma \equiv \{\xi_j^\Gamma \in M_h, j = 1, \dots, N_\Gamma^M\}, \quad (4.4.5)$$

where $N_\Gamma^Y = \sum_{r=1}^{R-1} N_{\Gamma_r}^Y$, $N_\Gamma^M = \sum_{r=1}^{R-1} N_{\Gamma_r}^M$ and $N_{\Gamma_r}^Y, N_{\Gamma_r}^M$ are, respectively, the number of \mathbb{P}_2 nodes and \mathbb{P}_1 nodes on the internal interface $\hat{\Gamma}_r$.

Finally we define the spaces

$$Y = Y_{RB}^1 \oplus \dots \oplus Y_{RB}^R \oplus Y_{FE}^\Gamma, M = M_{RB}^1 \oplus \dots \oplus M_{RB}^R \oplus M_{FE}^\Gamma. \quad (4.4.6)$$

We consider the following basis functions:

- \mathbf{v}_j^r and q_m^r are the RB functions associated to the domain $\hat{\Omega}_r$, as defined in (4.4.2), $j = 1, \dots, N^Y, m = 1, \dots, N^M$;
- $\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_r}$ and $\xi_{\tilde{m}}^{\Gamma_r}$ are the Lagrangian functions associated to the nodes on $\hat{\Gamma}_r^r, \tilde{j} = 1, \dots, N_{\Gamma_r}^Y, \tilde{m} = 1, \dots, N_{\Gamma_r}^M, r = 1, \dots, R-1$.

The RDF approximation of problem (4.1.7) is find $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in Y \times M$ such that:

$$\left\{ \begin{array}{ll} \sum_{q=1}^4 [\Theta_q^r(\mu^r) a_q^r(\mathbf{u}(\boldsymbol{\mu}), \mathbf{v}_i^r) - \Phi_q^r(\mu^r) b_q^r(\mathbf{v}_i^r, p(\boldsymbol{\mu}))] &= -\Psi(\mu^1) F(\mathbf{v}_i^r) \quad i = 1, \dots, N^Y, \\ \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{u}(\boldsymbol{\mu}), q_i^r) &= 0 \quad i = 1, \dots, N^M, \\ & \quad r = 1, \dots, R, \\ \sum_{q=1}^4 [\Theta_q^r(\mu^r) a_q^r(\mathbf{u}(\boldsymbol{\mu}), \boldsymbol{\phi}_i^{\Gamma_r}) + \Phi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_i^{\Gamma_r}, p(\boldsymbol{\mu}))] &+ \\ \sum_{q=1}^4 [\Theta_q^{r+1}(\mu^{r+1}) a_q^{r+1}(\mathbf{u}(\boldsymbol{\mu}), \boldsymbol{\phi}_i^{\Gamma_r}) + \Phi_q^{r+1}(\mu^{r+1}) b_q^r(\boldsymbol{\phi}_i^{\Gamma_r}, p(\boldsymbol{\mu}))] &= \Psi(\mu^1) F(\boldsymbol{\phi}_i^{\Gamma_r}) \quad i = 1, \dots, N_{\Gamma_r}^Y, \\ \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{u}(\boldsymbol{\mu}), \xi_i^{\Gamma_r}) + \sum_{q=1}^4 \Phi_q^{r+1}(\mu^{r+1}) b_q^{r+1}(\mathbf{u}(\boldsymbol{\mu}), \xi_i^{\Gamma_r}) &= 0 \quad i = 1, \dots, N_{\Gamma_r}^M, \\ & \quad r = 1, \dots, R-1. \end{array} \right. \quad (4.4.7)$$

The approximate solutions $\mathbf{u}(\boldsymbol{\mu})$ and $p(\boldsymbol{\mu})$ are defined in $Y \times M$, we can write them as linear combination of the basis functions of $Y \times M$.

$$\begin{aligned} \mathbf{u}(\mathbf{x}, \boldsymbol{\mu}) &= \sum_{r=1}^R \sum_{j=1}^{N^Y} u_j^r(\mu^r) \mathbf{v}_j^r(\mathbf{x}) + \sum_{j=1}^{N_\Gamma^Y} \lambda_{uj}(\mu^r) \boldsymbol{\phi}_j^\Gamma(\mathbf{x}), \\ p(\mathbf{x}, \boldsymbol{\mu}) &= \sum_{r=1}^R \sum_{j=1}^{N^M} p_j^r(\mu^r) q_j^r(\mathbf{x}) + \sum_{j=1}^{N_\Gamma^M} \lambda_{pj}(\mu^r) \xi_j^\Gamma(\mathbf{x}). \end{aligned} \quad (4.4.8)$$

By including this decomposition in problem (4.4.7), we obtain the algebraic form:

$$\left\{ \begin{array}{lcl} A_{rr} \mathbf{u}_r + B_{rr} \mathbf{p}_r + A_{r\Gamma_k} \boldsymbol{\lambda}_u^k + B_{r\Gamma_k} \boldsymbol{\lambda}_p^k & = & F_r, \\ D_{rr} \mathbf{u}_r + D_{r\Gamma_k} \boldsymbol{\lambda}_u^k & = & 0, \\ A_{\Gamma_k r} \mathbf{u}_r + A_{\Gamma_k r+1} \mathbf{u}_{r+1} + B_{\Gamma_k r} \mathbf{p}_r + B_{\Gamma_k r+1} \mathbf{p}_{r+1} \\ + [\mathbb{A}_{\Gamma_k \Gamma_k}^r + \mathbb{A}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_u^k + [\mathbb{B}_{\Gamma_k \Gamma_k}^r + \mathbb{B}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_p^k & = & \mathbb{F}_{\Gamma_k}^r + \mathbb{F}_{\Gamma_k}^{r+1}, \\ D_{\Gamma_k r} \mathbf{u}_r + D_{\Gamma_k r+1} \mathbf{u}_{r+1} + [\mathbb{D}_{\Gamma_k \Gamma_k}^r + \mathbb{D}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_u^k & = & 0. \end{array} \right. \quad (4.4.9)$$

where for $r = 1, \dots, R$, for $k = 1, \dots, R-1$ we define for $i, j = 1, \dots, N^Y$, $m = 1, \dots, N^M$, $\tilde{i}, \tilde{j} = 1, \dots, N_{\Gamma_k}^Y$, $\tilde{m}, \tilde{n} = 1, \dots, N_{\Gamma_k}^M$:

$$[\mathbf{u}_r]_j = u_j^r(\mu^r), [\mathbf{p}_r]_m = p_j^r(\mu^r), [\boldsymbol{\lambda}_u^k]_{\tilde{j}} = \lambda_{u\tilde{j}}(\mu^k), [\boldsymbol{\lambda}_p^k]_{\tilde{m}} = \lambda_{p\tilde{m}}(\mu^k),$$

$$[A_{rr}]_{i,j} = \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\mathbf{v}_j^r, \mathbf{v}_i^r), [B_{rr}]_{i,m} = \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{v}_i^r, q_m^r), D_{rr} = B_{rr}^T,$$

$$[A_{r\Gamma_k}]_{i,\tilde{j}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}, \mathbf{v}_i^r), [B_{r\Gamma_k}]_{i,\tilde{m}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\mathbf{v}_i^r, \xi_{\tilde{m}}^{\Gamma_k}),$$

$$[D_{r\Gamma_k}]_{\tilde{i},m} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, q_m^r), D_{\Gamma_k r} = B_{r\Gamma_k}^T, B_{\Gamma_k r} = D_{r\Gamma_k}^T,$$

$$[\mathbb{A}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{j}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}), [\mathbb{B}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{m}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \xi_{\tilde{m}}^{\Gamma_k}),$$

$$\mathbb{D}_{\Gamma_k \Gamma_k}^r = \mathbb{B}_{\Gamma_k \Gamma_k}^{rT}, [F_r]_j = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\mathbf{v}_j^r), [\mathbb{F}_{\Gamma_k}^r]_{\tilde{j}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}).$$

By defining:

$$M_r = \begin{bmatrix} A_{rr} & B_{rr} \\ D_{rr}^T & 0 \end{bmatrix}, M_{r\Gamma_k} = \begin{bmatrix} A_{r\Gamma_k} & B_{r\Gamma_k} \\ D_{r\Gamma_k}^T & 0 \end{bmatrix}, \mathbb{M}_{\Gamma_k \Gamma_k} = \begin{bmatrix} \mathbb{A}_{\Gamma_k \Gamma_k} & \mathbb{B}_{\Gamma_k \Gamma_k} \\ \mathbb{D}_{\Gamma_k \Gamma_k}^T & 0 \end{bmatrix}, \quad (4.4.10)$$

the final linear system is:

$$\begin{bmatrix} M_{rr} & M_{r\Gamma} \\ M_{r\Gamma}^T & \mathbb{M}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} F_{rr} \\ \mathbb{F}_{\Gamma} \end{bmatrix}, \quad (4.4.11)$$

4.4. The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

where

$$\begin{aligned}
 M_{rr} &= \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_R \end{bmatrix}, M_{r\Gamma} = \begin{bmatrix} M_{1\Gamma_1} & 0 & & & & \\ M_{2\Gamma_1} & M_{2\Gamma_2} & 0 & & & \\ & 0 & M_{3\Gamma_2} & M_{3\Gamma_3} & 0 & \\ & & \ddots & \ddots & \ddots & 0 \\ & & & 0 & M_{R-1\Gamma_{R-2}} & M_{R-1\Gamma_{R-1}} \\ & & & & 0 & M_{R\Gamma_{R-1}} \end{bmatrix}, \\
 M_{\Gamma\Gamma} &= \begin{bmatrix} \mathbb{M}_{\Gamma_1\Gamma_1}^1 + \mathbb{M}_{\Gamma_1\Gamma_1}^2 & & & & \\ & \mathbb{M}_{\Gamma_2\Gamma_2}^2 + \mathbb{M}_{\Gamma_2\Gamma_2}^3 & & & \\ & & \ddots & & \\ & & & \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix}, \\
 \mathbf{s}_r &= \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_u^1 \\ \vdots \\ \boldsymbol{\lambda}_u^{R-1} \\ \boldsymbol{\lambda}_p^1 \\ \vdots \\ \boldsymbol{\lambda}_p^{R-1} \end{bmatrix}, F_{rr} = \begin{bmatrix} F_1 \\ \vdots \\ F_R \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbb{F}_\Gamma = \begin{bmatrix} \mathbb{F}_{\Gamma_1}^1 + \mathbb{F}_{\Gamma_1}^2 \\ \vdots \\ \mathbb{F}_{\Gamma_{R-1}}^{R-1} + \mathbb{F}_{\Gamma_{R-1}}^R \\ 0 \\ \vdots \\ 0 \end{bmatrix}.
 \end{aligned}$$

System (4.4.11) represents the reduced version of the system (4.3.9) in which we consider some bases coming from local proper solutions (RB part) instead of the finite element bases and the same bases coming from the finite element functions in the FE region of the domain. In (4.4.11) the dimension of the linear system is $R(N^Y + N^M) + (R-1)(N_{\Gamma_r}^Y + N_{\Gamma_r}^M)$ that is much smaller than the dimension of system (4.3.9) that is $\dim(Y_h) + \dim(M_h)$.

4.4.1 Extending the FE regions

As we have done in Chapter 3, Section 3.4.2, we extend the region in which we consider the finite element functions as bases of the problem, we denote that region as $\hat{\Omega}_{FE}$ and it includes also the internal interfaces, see Figure 3.8 in Section 3.4.2. We consider again the local spaces V_{RB}^k defined in (4.4.1) by functions \mathbf{v}_i^k and p_i^k that are defined locally on each Λ_k , we map them into $\hat{\Omega}_r$ and we extend them to zero on the other subdomains of $\hat{\Omega}_j$, $j \neq r$ and on the FE region $\hat{\Omega}_{FE}$. We define for $r = 1, \dots, R$ and $i = 1, \dots, N_k^Y$, $m = 1, \dots, N_k^M$, $\mathbf{v}_i^r \in V_h$, $q_m^r \in M_h$ and:

$$\mathbf{v}_i^r(\mathbf{x}_j) = \mathbf{w}_i^k(\mathcal{T}_{kr}(\hat{\mathbf{x}}_j)), \text{ if } \mathbf{x}_j \in \hat{\Omega}_r \setminus \hat{\Omega}_{FE} \quad \mathbf{v}_i^r(\mathbf{x}_j) = 0, \text{ if } \mathbf{x}_j \in \hat{\Omega}_l \cup \hat{\Omega}_{FE}, l \neq r. \quad (4.4.12)$$

$$q_m^r(\mathbf{x}_j) = p_m^k(\mathcal{T}_{kr}(\hat{\mathbf{x}}_j)), \text{ if } \mathbf{x}_j \in \hat{\Omega}_r \setminus \hat{\Omega}_{FE} \quad q_m^r(\mathbf{x}_j) = 0, \text{ if } \mathbf{x}_j \in \hat{\Omega}_l \cup \hat{\Omega}_{FE}, l \neq r. \quad (4.4.13)$$

We can define on each $\hat{\Omega}_r$ the following spaces:

$$Y_{RB}^r = \{\mathbf{v}_i^r, i = 1, \dots, N_k^Y\}, M_{RB}^r = \{q_i^r, i = 1, \dots, N_k^M\}, \quad (4.4.14)$$

where $N_k^Y < N_h^Y$ and $N_k^M < N_h^M$. We consider now a subspace of the finite element space (4.2.5) composed by the finite element functions corresponding to the nodes \mathbf{x}_j on $\hat{\Omega}_{FE}$:

$$Y_{FE} \equiv \{\boldsymbol{\phi}_j \in Y_h, j = 1, \dots, N_{FE}^Y\}, M_{FE} \equiv \{\xi_j \in Y_h, j = 1, \dots, N_{FE}^M\}, \quad (4.4.15)$$

where $N_{FE}^Y > N_\Gamma^Y$ is the number of finite element \mathbb{P}_2 nodes on $\hat{\Omega}_{FE}$, $N_{FE}^M > N_\Gamma^M$ is the number of finite element \mathbb{P}_1 nodes on $\hat{\Omega}_{FE}$.

Finally we define the spaces:

$$Y = Y_{RB}^1 \oplus \dots \oplus Y_{RB}^R \oplus Y_{FE}, M = M_{RB}^1 \oplus \dots \oplus M_{RB}^R \oplus M_{FE}, \quad (4.4.16)$$

and we consider the following basis functions:

- \mathbf{v}_j^r and q_m^r are the RB functions associated to the domain $\hat{\Omega}_r \setminus \hat{\Omega}_{FE}$, $j = 1, \dots, N^Y$, $m = 1, \dots, N^M$;
- $\boldsymbol{\phi}_l^r$ and ξ_n^r are the Lagrangian functions associated to the FE nodes on $(\hat{\Omega}_{FE} \cap \hat{\Omega}_r) \setminus \hat{\Gamma}_l^r$, $l = 1, \dots, N_{rF}^Y$, $n = 1, \dots, N_{rF}^M$, $N_{rF}^Y = N_{rFE}^Y - N_{\Gamma_r}^Y$, $N_{rF}^M = N_{rFE}^M - N_{\Gamma_r}^M$;
- $\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_r}$ and $\xi_{\tilde{m}}^{\Gamma_r}$ are the Lagrangian functions associated to the FE nodes on the internal interfaces, $\hat{\Gamma}_l^r \subset \hat{\Omega}_{FE}$, $\tilde{j} = 1, \dots, N_{\Gamma_r}^Y$, $\tilde{m} = 1, \dots, N_{\Gamma_r}^M$.

In this case, the RDF approximation of problem (4.1.7) is find $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in Y \times M$ such that, for $r = 1, \dots, R$:

$$\left\{ \begin{array}{ll} \sum_{q=1}^4 [\Theta_q^r(\mu_1^r) a_q^r(\mathbf{u}(\boldsymbol{\mu}), \mathbf{v}_i^r) - \Phi_q^r(\mu_1^r) b_q^r(\mathbf{v}_i^r, p(\boldsymbol{\mu}))] = \Psi(\mu_1^r) F(\mathbf{v}_i^r) & i = 1, \dots, N^Y, \\ \sum_{q=1}^4 \Phi_q^r(\mu_1^r) b_q^r(\mathbf{u}(\boldsymbol{\mu}), q_i^r) = 0 & i = 1, \dots, N^M, \\ & r = 1, \dots, R, \\ \sum_{k=r}^{r+1} \left[\sum_{q=1}^4 [\Theta_q^k(\mu_1^k) a_q^k(\mathbf{u}(\boldsymbol{\mu}), \boldsymbol{\phi}_i) - \Phi_q^k(\mu_1^k) b_q^k(\boldsymbol{\phi}_i, p(\boldsymbol{\mu}))] \right] = \Psi(\mu_1^1) F(\boldsymbol{\phi}_i) & i = 1, \dots, N_{FE}^Y, \\ \sum_{k=r}^{r+1} \left[\sum_{q=1}^4 \Phi_q^k(\mu_1^k) b_q^k(\mathbf{u}(\boldsymbol{\mu}), \xi_i) \right] = 0 & i = 1, \dots, N_{FE}^M, \\ & r = 1, \dots, R-1. \end{array} \right. \quad (4.4.17)$$

The approximate solutions $\mathbf{u}(\boldsymbol{\mu})$ and $p(\boldsymbol{\mu})$ are defined in $Y \times M$, we can write it as linear combination of the basis functions of $Y \times M$.

$$\begin{aligned} \mathbf{u}(\mathbf{x}, \boldsymbol{\mu}) &= \sum_{r=1}^R \sum_{j=1}^{N^Y} u_j^r(\mu^r) \mathbf{v}_j^r(\mathbf{x}) + \sum_{r=1}^R \sum_{j=1}^{N_{rF}^Y} \lambda_{uj}^r(\mu^r) \boldsymbol{\phi}_j^r(\mathbf{x}) + \sum_{k=1}^{R-1} \sum_{j=1}^{N_{\Gamma_k}^Y} \lambda_{uj}^{\Gamma_k}(\mu^r) \boldsymbol{\phi}_j^{\Gamma_k}(\mathbf{x}), \\ p(\mathbf{x}, \boldsymbol{\mu}) &= \sum_{r=1}^R \sum_{j=1}^{N^M} p_j^r(\mu^r) q_j^r(\mathbf{x}) + \sum_{r=1}^R \sum_{j=1}^{N_{rF}^M} \lambda_{pj}^r(\mu^r) \xi_j^r(\mathbf{x}) + \sum_{k=1}^{R-1} \sum_{j=1}^{N_{\Gamma_k}^M} \lambda_{pj}^{\Gamma_k}(\mu^r) \xi_j^{\Gamma_k}(\mathbf{x}). \end{aligned}$$

4.4. The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

The algebraic formulation of the problem (4.4.17) becomes:

$$\left\{ \begin{array}{l} A_{rr} \mathbf{u}_r + B_{rr} \mathbf{p}_r + A_{rF} \boldsymbol{\lambda}_u^r + B_{rF} \boldsymbol{\lambda}_p^r = F_r, \\ D_{rr} \mathbf{u}_r + D_{rF} \boldsymbol{\lambda}_u^r = 0, \\ A_{Fr} \mathbf{u}_r + B_{Fr} \mathbf{p}_r + \mathbb{A}_{FF}^r \boldsymbol{\lambda}_u^r + \mathbb{B}_{FF}^r \boldsymbol{\lambda}_p^r + \mathbb{A}_{F\Gamma_k}^r \boldsymbol{\lambda}_u^{\Gamma_k} + \mathbb{B}_{F\Gamma_k}^r \boldsymbol{\lambda}_p^{\Gamma_k} = \mathbb{F}_F^r, \\ D_{Fr} \mathbf{u}_r + \mathbb{D}_{FF}^r \boldsymbol{\lambda}_u^r + \mathbb{D}_{F\Gamma_k}^r \boldsymbol{\lambda}_p^{\Gamma_k} = 0, \quad r = 1, \dots, R, \\ \mathbb{A}_{\Gamma_k F}^r \boldsymbol{\lambda}_u^r + \mathbb{A}_{\Gamma_k F}^{r+1} \boldsymbol{\lambda}_u^{r+1} + \mathbb{B}_{\Gamma_k F}^r \boldsymbol{\lambda}_p^r + \mathbb{B}_{\Gamma_k F+1}^{r+1} \boldsymbol{\lambda}_p^{r+1} \\ + [\mathbb{A}_{\Gamma_k \Gamma_k}^r + \mathbb{A}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_u^{\Gamma_k} + [\mathbb{B}_{\Gamma_k \Gamma_k}^r + \mathbb{B}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_p^{\Gamma_k} = \mathbb{F}_{\Gamma_k}^r + \mathbb{F}_{\Gamma_k}^{r+1}, \\ \mathbb{D}_{\Gamma_k F} \boldsymbol{\lambda}_u^r + \mathbb{D}_{\Gamma_k F+1} \boldsymbol{\lambda}_u^{r+1} + [\mathbb{D}_{\Gamma_k \Gamma_k}^r + \mathbb{D}_{\Gamma_k \Gamma_k}^{r+1}] \boldsymbol{\lambda}_p^{\Gamma_k} = 0, \quad r = 1, \dots, R-1, \end{array} \right. \quad (4.4.18)$$

where for $r = 1, \dots, R$, for $k = 1, \dots, R-1$ and $i, j = 1, \dots, N^Y$, $m = 1, \dots, N^M$, $l, f = 1, \dots, N_{rF}^Y$, $n = 1, \dots, N_{rF}^M$, $\tilde{i}, \tilde{j} = 1, \dots, N_{\Gamma_r}^Y$, $\tilde{m} = 1, \dots, N_{\Gamma_r}^M$:

$$[\mathbf{u}_r]_j = u_j^r(\mu^r), [\boldsymbol{\lambda}_u^r]_l = \lambda_{ul}^r(\mu^r), [\boldsymbol{\lambda}_u^{\Gamma_k}]_{\tilde{j}} = \lambda_{u\tilde{j}}^{\Gamma_k}(\mu^k),$$

$$[\mathbf{p}_r]_m = p_m^r(\mu^r), [\boldsymbol{\lambda}_p^r]_n = \lambda_{pm}^r(\mu^r), [\boldsymbol{\lambda}_p^{\Gamma_k}]_{\tilde{m}} = \lambda_{p\tilde{m}}^{\Gamma_k}(\mu^k)$$

$$\begin{aligned} [A_{rr}]_{i,j} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\mathbf{v}_j^r, \mathbf{v}_i^r), [B_{rr}]_{i,m} = \sum_{q=1}^4 \Phi_q^r(\mu^r) b_q^r(\mathbf{v}_i^r, q_m^r), D_{rr} = B_{rr}, \\ [A_{rF}]_{i,l} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_l^r, \mathbf{v}_i^r), [B_{rF}]_{i,n} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\mathbf{v}_i^r, \xi_n^r), \\ [\mathbb{A}_{F\Gamma_k}]_{l,\tilde{j}} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}, \boldsymbol{\phi}_l^r), [\mathbb{B}_{F\Gamma_k}]_{l,\tilde{m}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_l^r, \xi_{\tilde{m}}^{\Gamma_k}), \\ [D_{rF}]_{m,l} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_l^r, q_m^r), [\mathbb{D}_{F\Gamma_k}]_{n,\tilde{j}} = \sum_{q=1}^4 \Theta_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}, \xi_n^{\Gamma_k}), \\ [\mathbb{A}_{FF}^r]_{l,f} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_f^r, \boldsymbol{\phi}_l^r), [\mathbb{B}_{FF}^r]_{l,n} = \sum_{q=1}^4 \Psi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_l^r, \xi_n^r), \\ [\mathbb{A}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{j}} &= \sum_{q=1}^4 \Theta_q^r(\mu^r) a_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}), [\mathbb{B}_{\Gamma_k \Gamma_k}^r]_{\tilde{i},\tilde{m}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) b_q^r(\boldsymbol{\phi}_{\tilde{i}}^{\Gamma_k}, \xi_{\tilde{m}}^{\Gamma_k}), \\ \mathbb{D}_{F\Gamma_k} &= \mathbb{B}_{F\Gamma_k}^T, \mathbb{D}_{FF}^r = \mathbb{B}_{FF}^{rT}, \mathbb{D}_{\Gamma_k \Gamma_k}^r = \mathbb{B}_{\Gamma_k \Gamma_k}^{rT} \\ [F_r]_j &= \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\mathbf{v}_j^r), [\mathbb{F}_F^r]_l = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\boldsymbol{\phi}_l^r), [\mathbb{F}_{\Gamma_k}^r]_{\tilde{j}} = \sum_{q=1}^4 \Psi_q^r(\mu^r) F_q^r(\boldsymbol{\phi}_{\tilde{j}}^{\Gamma_k}). \end{aligned}$$

By defining:

$$\begin{aligned} M_r &= \begin{bmatrix} A_{rr} & B_{rr} \\ D_{rr} & 0 \end{bmatrix}, M_{rF} = \begin{bmatrix} A_{rF} & B_{rF} \\ D_{rF} & 0 \end{bmatrix}, \\ \mathbb{M}_{F\Gamma_k} &= \begin{bmatrix} \mathbb{A}_{F\Gamma_k} & \mathbb{B}_{F\Gamma_k} \\ \mathbb{D}_{F\Gamma_k} & 0 \end{bmatrix}, \mathbb{M}_{\Gamma_k \Gamma_k} = \begin{bmatrix} \mathbb{A}_{\Gamma_k \Gamma_k} & \mathbb{B}_{\Gamma_k \Gamma_k} \\ \mathbb{D}_{\Gamma_k \Gamma_k} & 0 \end{bmatrix}, \mathbb{M}_{FF} = \begin{bmatrix} \mathbb{A}_{FF} & \mathbb{B}_{FF} \\ \mathbb{D}_{FF} & 0 \end{bmatrix}, \end{aligned} \quad (4.4.19)$$

the final linear system is:

$$\begin{bmatrix} M_{rr} & M_{rF} & 0 \\ M_{rF}^T & \mathbb{M}_{FF} & \mathbb{M}_{F\Gamma} \\ 0 & \mathbb{M}_{F\Gamma}^T & \mathbb{M}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r \\ \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{rr} \\ \mathbf{f}_F \\ \mathbf{f}_\Gamma \end{bmatrix}, \quad (4.4.20)$$

where

$$\begin{aligned} M_{rr} &= \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_R \end{bmatrix}, M_{rF} = \begin{bmatrix} M_{1F_1} & & & \\ & M_{2F_2} & & \\ & & \ddots & \\ & & & M_{RF_R} \end{bmatrix}, \mathbb{M}_{FF} = \begin{bmatrix} \mathbb{M}_{F_1} & & & \\ & \mathbb{M}_{F_2} & & \\ & & \ddots & \\ & & & \mathbb{M}_{F_R} \end{bmatrix}, \\ \mathbb{M}_{F\Gamma} &= \begin{bmatrix} \mathbb{M}_{F_1\Gamma_1} & 0 & & & \\ \mathbb{M}_{F_2\Gamma_1} & \mathbb{M}_{F_2\Gamma_2} & 0 & & \\ 0 & \mathbb{M}_{F_3\Gamma_2} & \mathbb{M}_{F_3\Gamma_3} & 0 & \\ & \ddots & \ddots & \ddots & 0 \\ & & 0 & \mathbb{M}_{F_{R-1}\Gamma_{R-2}} & \mathbb{M}_{F_{R-1}\Gamma_{R-1}} \\ & & & 0 & \mathbb{M}_{F_R\Gamma_{R-1}} \end{bmatrix}, \\ \mathbb{M}_{\Gamma\Gamma} &= \begin{bmatrix} \mathbb{M}_{\Gamma_1\Gamma_1}^1 + \mathbb{M}_{\Gamma_1\Gamma_1}^2 & & & \\ & \mathbb{M}_{\Gamma_2\Gamma_2}^2 + \mathbb{M}_{\Gamma_2\Gamma_2}^3 & & \\ & & \ddots & \\ & & & \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^{R-1} + \mathbb{M}_{\Gamma_{R-1}\Gamma_{R-1}}^R \end{bmatrix}, \\ \mathbf{s}_r &= \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_R \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_R \end{bmatrix}, \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_u^1 \\ \vdots \\ \boldsymbol{\lambda}_u^R \\ \boldsymbol{\lambda}_p^1 \\ \vdots \\ \boldsymbol{\lambda}_p^R \end{bmatrix}, \boldsymbol{\lambda}^\Gamma = \begin{bmatrix} \boldsymbol{\lambda}_u^{\Gamma_1} \\ \vdots \\ \boldsymbol{\lambda}_u^{\Gamma_{R-1}} \\ \boldsymbol{\lambda}_p^{\Gamma_1} \\ \vdots \\ \boldsymbol{\lambda}_p^{\Gamma_{R-1}} \end{bmatrix}, \mathbf{f}_{rr} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_R \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{f}_F = \begin{bmatrix} \mathbf{f}_{F_1} \\ \vdots \\ \mathbf{f}_{F_R} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{f}_\Gamma = \begin{bmatrix} \mathbf{f}_{\Gamma_1}^1 + \mathbf{f}_{\Gamma_1}^2 \\ \vdots \\ \mathbf{f}_{\Gamma_{R-1}}^{R-1} + \mathbf{f}_{\Gamma_{R-1}}^R \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned}$$

As before, we split the matrix of the linear system (4.4.20) into different sub-blocks M_{rr} containing the informations coming from the internal nodes of the R subdomains, the block $\mathbb{M}_{\Gamma\Gamma}$ from the interface nodes between different subdomains, $\mathbb{M}_{F\Gamma}$ and $M_{F\Gamma}$ partially from the internal nodes, partially from internal FE nodes and partially from interface nodes.

System (4.4.20) still represents a reduced version of the system (4.3.9), its dimension is bigger than the reduced system (4.4.11) and it depends on the dimension of the subdomain $\hat{\Omega}_{FE}$.

4.4. The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

As in the case of the Laplace equation, see Chapter 3, the use of the FE basis functions allows to add more degrees of freedom in the most critical portion of the domain (in terms of difficulty to recover the approximate solution of the problem), i.e. the portions containing the internal interfaces. Despite that, due to the fact that the RB local functions are computed independently and embed the informations about the local BCs, the procedure to compute them is crucial to guarantee a smooth solution. In the next section, this procedure is introduced.

4.4.2 Precomputing the basis solutions

While the description of the method for the Stokes equations follows step by step the description given for the Laplace equation, the strategy for computation of the basis functions is different in this case, due to the different features of the Stokes problem. This computation is performed locally in the reference shapes $\Lambda_k, k = 1, \dots, K$, and it is independent of the number of subdomains that will compose the whole computational domain. The definition of these functions may affect strongly the numerical solution of the final system (4.4.11) or (4.4.20) and therefore its accuracy.

Also in this case, the goal at this stage is to define a set of basis functions that is large enough to recover locally the final solution of the problem. We consider now not only the case of parametric Dirichlet BCs, but also the case of parametric Neumann BCs.

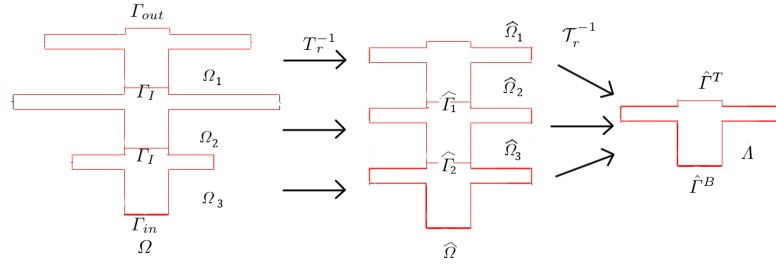


Figure 4.4: Computational domain Ω (left), reference domain $\hat{\Omega}$ (center) and reference shape Λ (right).

We consider only one reference shape Λ , as shown in Figure 4.4 and we want to define the reduced basis spaces Y_{RB} and M_{RB} introduced in (4.4.1). We denote with $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ the two boundaries of Λ that may be mapped into the inlet, the outlet or the internal interfaces in the whole domain $\hat{\Omega}$.

We introduce two type of parametric local Stokes problems in a generic subdomain Ω_r , we define Γ_λ as, depending of the position of Ω_r , $\Gamma_\lambda = \Gamma_{in} \cup \Gamma_I$ or $\Gamma_\lambda = \Gamma_I$ or $\Gamma_\lambda = \Gamma_I \cup \Gamma_{out}$.

The first Stokes problem includes Dirichlet BCs on Γ_λ defined through the parametric function $\lambda(\mu_\lambda)$:

$$\begin{cases} -\nu \Delta \mathbf{u}(\mu^r, \mu_\lambda) + \nabla p(\mu^r, \mu_\lambda) = \mathbf{0} & \text{in } \Omega_r, \\ \nabla \cdot \mathbf{u}(\mu^r, \mu_\lambda) = 0 & \text{in } \Omega_r, \\ \mathbf{u}(\mu^r, \mu_\lambda) = \mathbf{0} & \text{on } \partial\Omega_r \setminus \Gamma_\lambda, \\ \mathbf{u}(\mu^r, \mu_\lambda) = \lambda(\mu_\lambda) & \text{on } \Gamma_\lambda. \end{cases} \quad (4.4.21)$$

Let us introduce a lift R_λ of the Dirichlet BC on Γ_λ . R_λ can be an affine function described as follows:

$$R_\lambda(\mathbf{x}, \mu_\lambda) = \sum_{q=1}^{Q_l} \tilde{\lambda}_l(\mu_\lambda) R_\lambda^l(\mathbf{x}), \quad (4.4.22)$$

or can be approximated by an affine function through the empirical interpolation method (Section 1.2.3.1):

$$R_\lambda(\mathbf{x}, \mu_\lambda) \approx \sum_{q=1}^{Q_l} \tilde{\lambda}_l(\mu_\lambda) R_\lambda^l(\mathbf{x}, \mu_\lambda^l). \quad (4.4.23)$$

We consider the weak formulation of the problem (4.4.21) and by applying the same local affine decomposition presented in Section 4.1.1, we can recast the weak formulation of problem (4.4.21) on $\hat{\Omega}_r$ and consequently on Λ that represents its translation.

We define the following finite element spaces:

$$Y_h(\Lambda) = (X_h^2(\Lambda))^2, \quad X_h^2(\Lambda) \equiv \{\mathbf{v}_h \in C^0(\Lambda) : \mathbf{v}_h|_K \in \mathbb{P}_2 \quad \forall K \in \mathcal{T}_h\} \cap [H_0^1(\Lambda)]^2, \quad (4.4.24)$$

$$M_h(\Lambda) = X_h^1(\Lambda) \equiv \{p_h \in C^0(\Lambda) : p_h|_K \in \mathbb{P}_1 \quad \forall K \in \mathcal{T}_h\}. \quad (4.4.25)$$

The finite element approximation problem (4.4.21) is: find $\mathbf{u}_h \in Y_h(\Lambda)$, $p_h \in M_h(\Lambda)$ such that $\forall \mathbf{v}_h \in Y_h(\Lambda)$ and $\forall q_h \in M_h(\Lambda)$:

$$\begin{cases} \sum_{q=1}^Q \Theta_q(\boldsymbol{\mu}) a_q(\mathbf{u}_h, \mathbf{v}_h) + \sum_{q=1}^Q \Phi_q(\boldsymbol{\mu}) b_q(\mathbf{v}_h, p_h) = \sum_{q=1}^{Q_l} \Psi_q(\boldsymbol{\mu}) F_q(\mathbf{v}_h), \\ \sum_{q=1}^Q \Phi_q(\boldsymbol{\mu}) b_q(\mathbf{u}_h, q_h) = - \sum_{q=1}^{Q_l} \chi_q(\boldsymbol{\mu}) G_q(q_h). \end{cases} \quad (4.4.26)$$

where

$$a_{q(i,j)}(\mathbf{v}, \mathbf{w}) = \int_{\Lambda} \frac{\partial \mathbf{v}}{\partial x_i} \frac{\partial \mathbf{w}}{\partial x_j} d\mathbf{x}, \quad \Theta_{q(i,j)}(\boldsymbol{\mu}) = v \left[C_{\boldsymbol{\mu}}^{-1} C_{\boldsymbol{\mu}}^{-T} |det C_{\boldsymbol{\mu}}| \right]_{ij}, \quad (4.4.27)$$

$$b_{q(i,j)}(\mathbf{v}, q) = \int_{\Lambda} q \frac{\partial w_i}{\partial x_j} d\mathbf{x}, \quad \Phi_{q(i,j)}(\boldsymbol{\mu}) = \left[C_{\boldsymbol{\mu}}^{-1} |det C_{\boldsymbol{\mu}}| \right]_{ij}, \quad (4.4.28)$$

$$F_{q(i,j,l)}(\mathbf{v}) = - \int_{\Lambda} \frac{\partial R_{\lambda}^l}{\partial x_i} \frac{\partial \mathbf{v}}{\partial x_j} d\mathbf{x}, \quad \chi_{q(i,j,l)}(\boldsymbol{\mu}) = v \tilde{\lambda}_l(\mu_\lambda) \left[C_{\boldsymbol{\mu}}^{-1} C_{\boldsymbol{\mu}}^{-T} |det C_{\boldsymbol{\mu}}| \right]_{ij}, \quad (4.4.29)$$

$$G_{q(i,j,l)}(q) = \int_{\Lambda} q \frac{\partial R_{\lambda}^l}{\partial x_j} d\mathbf{x}, \quad \Psi_{q(i,j,l)}(\boldsymbol{\mu}) = v \tilde{\lambda}_l(\mu_\lambda) \left[C_{\boldsymbol{\mu}}^{-1} |det C_{\boldsymbol{\mu}}| \right]_{ij}. \quad (4.4.30)$$

In the second Stokes problem we use the parametric function $\boldsymbol{\lambda}(\mu_\lambda)$ to define the Neumann BC on Γ_λ :

$$\begin{cases} -v \Delta \mathbf{u}(\mu^r, \mu_\lambda) + \nabla p(\mu^r, \mu_\lambda) = \mathbf{0} & \text{in } \Omega_r, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega_r, \\ \mathbf{u}(\mu^r, \mu_\lambda) = \mathbf{0} & \text{on } \partial\Omega_r \setminus \Gamma_\lambda, \\ v \frac{\partial \mathbf{u}(\mu^r, \mu_\lambda)}{\partial \mathbf{n}} - p(\mu^r, \mu_\lambda) \mathbf{n} = \boldsymbol{\lambda}(\mu_\lambda) & \text{on } \Gamma_\lambda. \end{cases} \quad (4.4.31)$$

4.4. The Reduced basis-Domain decomposition-Finite element method (RDF) for Stokes equations

The function $\lambda(\mu_\lambda)$ can be an affine function described as follows:

$$\lambda(\mathbf{x}, \mu_\lambda) = \sum_{l=1}^L \tilde{\lambda}_l(\mu_\lambda) \lambda^l(\mathbf{x}), \quad (4.4.32)$$

or can be approximated by an affine function through the empirical interpolation method:

$$\lambda(\mathbf{x}, \mu_\lambda) \approx \sum_{l=1}^L \tilde{\lambda}_l(\mu_\lambda) \lambda^l(\mathbf{x}, \mu_\lambda^l). \quad (4.4.33)$$

The finite element approximation problem (4.4.31) is: find $\mathbf{u}_h \in Y_h(\Lambda)$, $p_h \in M_h(\Lambda)$ such that $\forall \mathbf{v}_h \in Y_h(\Lambda)$ and $\forall q_h \in M_h(\Lambda)$:

$$\begin{cases} \sum_{q=1}^Q \Theta_q(\boldsymbol{\mu}) a_q(\mathbf{u}_h, \mathbf{v}_h) + \sum_{q=1}^Q \Phi_q(\boldsymbol{\mu}) b_q(\mathbf{v}_h, p_h) = \sum_{l=1}^L \Psi_l(\boldsymbol{\mu}) F_l(\mathbf{v}_h), \\ - \sum_{q=1}^Q \Phi_q(\boldsymbol{\mu}) b_q(\mathbf{u}_h, q_h) = 0, \end{cases} \quad (4.4.34)$$

where $a_{q(i,j)}(\mathbf{v}, \mathbf{w})$ and $b_{q(i,j)}(\mathbf{v}, q)$ are defined as in (4.4.27) and (4.4.28) and

$$F_l(\mathbf{v}) = - \int_{\Gamma_\lambda} \lambda^l \cdot \mathbf{v} d\mathbf{x}, \quad \Psi_l(\boldsymbol{\mu}) = \tilde{\lambda}_l(\mu_\lambda) |\det C_\mu|. \quad (4.4.35)$$

The parameter μ^r indicates the geometrical transformation from Λ to $\hat{\Omega}_r$, from now, in order to consider every generic transformation of Λ we use only the symbol μ .

The two problems (4.4.34) and (4.4.26) represent two independent parametric problems and the two different possible ways to compute the local bases.

For both options, we follow the same general procedure to select the local basis functions presented in Chapter 3.

In the following sections we will refer to the problem (4.4.21) and (4.4.31) in order to specify which type of parametric BCs we use.

We perform now the offline stage of the classical reduced basis method applied to the parametrized problem in Λ in order to select a set of suitable parameter values that define the local reduced basis functions.

Thanks to the greedy algorithm, described in Section 1.2.5.1, we define the set $\{(\boldsymbol{\mu}^i, \mu_\lambda^i), i = 1, \dots, N\}$ and the space of the solutions the local problem (4.4.34) (or (4.4.26)) corresponding to these parameter values:

$$Z_{RB} = \{\mathbf{u}(\boldsymbol{\mu}^i, \mu_\lambda^i), i = 1, \dots, N_{RB}\}, \quad M_{RB} = \{p(\boldsymbol{\mu}^i, \mu_\lambda^i), i = 1, \dots, N_{RB}\}. \quad (4.4.36)$$

As seen in Chapter 2, in order to guarantee the approximation stability of the reduced basis method for the Stokes problem, we enrich the velocity space with the supremizer functions associated to each solution $p(\boldsymbol{\mu}^i, \mu_\lambda^i)$:

$$X_{RB} = \{\mathbf{v}(\boldsymbol{\mu}^i, \mu_\lambda^i), i = 1, \dots, N_{RB}\}, \quad Y_{RB} = Z_{RB} \oplus X_{RB}. \quad (4.4.37)$$

In general if there are k reference domains, we define k independent parametric problems and we perform k independent greedy algorithms (in parallel) and we define k reduced basis spaces Y_{RB}^k, M_{RB}^k

with a proper number N_k of basis functions.

As already mentioned the procedure to compute the basis functions could involve the problem (4.4.26) and/or (4.4.34): in Section 4.6 the numerical results will show the quality of the RDF solution by using both local problems. Moreover, the choice of the function $\lambda(\mathbf{x})$ defined in this step strongly affects the final solution of the problem, on the other hand it influences the computational complexity of the greedy algorithm. The obtained numerical results regarding the choice of λ (either Fourier or Lagrange functions) and the accuracy and computational complexity of the method feature performances are similar to those of the Laplace problem. For this reason we bound ourselves to show only the tests obtained by using the Fourier option. Furthermore we have analyzed in detail the results computed by using locally the RB spaces obtained by using the function λ as Dirichlet or Neumann condition. Section 4.6 reports the related results regarding both the offline complexity and the solution accuracy reached in the online stage.

4.5 Offline/Online decomposition and computational complexity

The offline/online decomposition of the problem in the case of the Stokes equations is an easy generalization of the Laplace one. So that here we only recall the basic ideas. All the computations involving the definition of the local reduced basis functions are performed in an offline stage.

The offline/online computational decomposition regards also the matrix assembling of the linear system (4.4.11) or, in the case of extended FE region, (4.4.20). In both cases the matrices can be assembled starting from the FE matrices associated to the original problem (4.4.19), in which $\mathbb{M} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, $\mathbb{F} \in \mathbb{R}^{\mathcal{N} \times 1}$. We note that \mathcal{N} indicates the dimension of the whole finite element space (\mathbb{P}_1 and \mathbb{P}_2 nodes) and N_k the number of local basis functions (velocities and pressures), we recall that in general $N_k \ll \mathcal{N}$. We still assume that we have only a reference shape Λ ($k = 1$), and $N (= 3N_{RB})$ is the number of reduced basis functions for every subdomain $\hat{\Omega}_r$ and in its translation Λ , in which $\mathcal{N}_k = \mathcal{N}_r$. Moreover, we assume that the number of nodes (\mathbb{P}_1 and \mathbb{P}_2) N_{Γ_k} along the internal interfaces Γ_k , $k = 1, \dots, R-1$ of the computational domain is the same $\forall k \in \{1, \dots, R-1\}$ and it coincides with the number N_{Γ} of nodes (\mathbb{P}_1 and \mathbb{P}_2) on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ in Λ .

In particular, thanks to the introduced domain decomposition approach, \mathbb{M} and \mathbb{F} can be decomposed in submatrices $\mathbb{M}^r \in \mathbb{R}^{\mathcal{N}_r \times \mathcal{N}_r}$ and subvectors $\mathbb{F}^r \in \mathbb{R}^{\mathcal{N}_r \times 1}$ that can be defined locally in the reference domains and without further assumptions on the reference shape. Due to the fact that Λ is a translation of $\hat{\Omega}_r = \mathcal{T}_r(\Lambda)$.

4.5.1 FE region represented only by the internal interfaces

We introduce the matrix $S \in \mathbb{R}^{(\mathcal{N}_k - N_{\Gamma_k}) \times N}$, containing the local reduced basis functions $\mathbf{v}_i \in Y_{RB}$, $q_i \in M_{RB}$ computed in Λ and associated to the nodes of $\Lambda \setminus \Gamma_D$.

$$S = \begin{bmatrix} U & 0 \\ 0 & P \end{bmatrix}, U = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_{2N_{RB}} \end{bmatrix}, P = \begin{bmatrix} q_1 & \dots & q_{N_{RB}} \end{bmatrix}. \quad (4.5.1)$$

The submatrices M_r of the systems (4.4.11) are defined as follows, for $r = 1, \dots, R$:

$$M_r = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) M_r^q = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) S^T \mathbb{M}^q S, \quad (4.5.2)$$

where, $\tilde{\Theta}_r^q(\boldsymbol{\mu})$ denote the coefficient regarding both the matrices A and B . Once we have defined the local reduced basis functions, the matrices $S^T \mathbb{M}^q S$, $q = 1, \dots, Q$, can be assembled offline and stored as matrices of smaller dimensions.

Regarding the assembling of the matrices $M_{r\Gamma_k}$, $r = 1, \dots, R$, $k = 1, \dots, R-1$, we follow a similar procedure:

$$M_{r\Gamma_k} = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) M_{\Gamma}^q = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) S^T \mathbb{M}_{\Gamma}^q. \quad (4.5.3)$$

The matrices $S^T \mathbb{M}_{\Gamma}^q$, $q = 1, \dots, 4$, can be computed and stored into smaller matrices.

The matrix \mathbb{M}_{Γ} coincides exactly with the finite element submatrix involving the nodes on the internal interfaces. These small dimensional matrices $\mathbb{M}_{\Gamma}^q \in \mathbb{R}^{N_{\Gamma} \times N_{\Gamma}}$ can be computed and stored offline.

The same decoupling strategy is considered for the right hand side of the system.

$$\mathbf{f}_r = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) F^q = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) S^T \mathbb{F}^q, \quad r = 1, \dots, R, \quad (4.5.4)$$

$$\mathbf{f}_{\Gamma_k}^r = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) \mathbb{F}_{\Gamma}^q, \quad r = 1, \dots, R, \quad k = 1, \dots, R-1. \quad (4.5.5)$$

The vectors $S^T \mathbb{F}^q \in \mathbb{R}^{N \times 1}$ and $\mathbb{F}_{\Gamma}^q \in \mathbb{R}^{N_{\Gamma} \times 1}$, $q = 1, \dots, Q$ can be computed and stored offline.

Finally in the online stage, for a new value of $\boldsymbol{\mu}$ and for an arbitrary number R of subdomains, we assemble the matrices M_r , $M_{r\Gamma_k}$, $\mathbb{M}_{\Gamma_k}^r$ and the vectors \mathbf{f}_r and $\mathbf{f}_{\Gamma_k}^r$ for $r = 1, \dots, R$ and $k = 1, \dots, R-1$, that will define respectively the matrices $M_{rr} \in \mathbb{R}^{RN \times RN}$, $M_{r\Gamma} \in \mathbb{R}^{RN \times (R-1)N_{\Gamma}}$, $\mathbb{M}_{\Gamma\Gamma} \in \mathbb{R}^{(R-1)N_{\Gamma} \times (R-1)N_{\Gamma}}$ and the vectors $\mathbf{f}_{rr} \in \mathbb{R}^{RN \times 1}$ and $\mathbf{f}_{\Gamma} \in \mathbb{R}^{(R-1)N_{\Gamma} \times 1}$ and we solve the linear system (4.4.11).

Summarizing the quantities that determine the dimension of the final system (that has to be solved to find the RDF solution of the problem) are:

- the number of subdomains R ,
- the number of nodes along each internal interface N_{Γ} (\mathbb{P}_1 and \mathbb{P}_2 nodes),
- the number of local reduced basis function $N = 3N_{RB}$,

so that the dimension of the linear system (4.4.11) is $(RN + (R-1)N_{\Gamma}) \times (RN + (R-1)N_{\Gamma})$.

4.5.2 Extending the FE region

As introduced in the previous section, in this case we extend the region $\hat{\Omega}_{FE}$ in which we use the finite element functions as bases for the solution of the problem, so that $\hat{\Gamma}_I \subset \hat{\Omega}_{FE} \subset \hat{\Omega}$, see for example Figure 3.8 in Section 3.4.2. We have introduced the quantity $N_{FE} < \mathcal{N}$ equal to the nodes of the domain $\hat{\Omega}$ involved in the region $\hat{\Omega}_{FE}$, such that $N_{FE} = \sum_{r=1}^R N_F^r + (R-1)N_{\Gamma}$, we assume that the quantity N_F^r is the same $\forall r \in \{1, \dots, R\}$. We can consider on Λ the corresponding region $\Lambda_{FE} = \mathcal{T}_r^{-1}(\hat{\Omega}_{FE} \cap \hat{\Omega}_r \setminus \Gamma_I)$ and denote with $N_F = N_F^r$ the number of nodes (\mathbb{P}_1 and \mathbb{P}_2) involved in this region.

We define the matrix $\tilde{S} \in \mathbb{R}^{(\mathcal{N}-N_F) \times N}$ containing the local reduced basis functions associated to the subdomain $\Lambda \setminus \Lambda_{FE}$.

The submatrices M_r of the system (4.5.1) are defined as before, for $r = 1, \dots, R$:

$$M_r = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}), \quad M_r^q = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) \tilde{S}_r^T \mathbb{M}_r^q \tilde{S}_r. \quad (4.5.6)$$

Once we have defined the local reduced basis functions, the matrices $\tilde{S}_r^T \mathbb{M}_r^q \tilde{S}_r$ can be assembled offline and stored as matrices of smaller dimensions.

Regarding the assembling of the matrices M_{rF_r} , we have:

$$M_{rF_k} = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}), \quad M_F^q = \sum_{q=1}^Q \tilde{\Theta}_r^q(\boldsymbol{\mu}) \tilde{S}_r^T \mathbb{M}_F^q. \quad (4.5.7)$$

The matrices $\tilde{S}_r^T \mathbb{M}_F^q$ can be computed and stored offline in the smaller matrices. The matrices $\mathbb{M}_{FF}, \mathbb{M}_{F\Gamma}$ and $\mathbb{M}_{\Gamma\Gamma}$ coincide with the FE submatrices corresponding to the nodes on the FE region of the domain. Also these small dimensional matrices $\mathbb{M}_{FF}^q \in \mathbb{R}^{N_F \times N_F}, \mathbb{M}_{F\Gamma}^q \in \mathbb{R}^{N_F \times N_\Gamma}, \mathbb{M}_{\Gamma\Gamma}^q \in \mathbb{R}^{N_\Gamma \times N_\Gamma}$, can be computed and stored offline. Analogous considerations regarding the right hand side term of system (4.5.1) would follow.

$$\mathbf{f}_r = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) F^q = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) \tilde{S}_r^T \mathbb{F}^q, \quad r = 1, \dots, R, \quad (4.5.8)$$

$$\mathbf{f}_{F_r} = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) \mathbb{F}_F^q, \quad r = 1, \dots, R, \quad (4.5.9)$$

$$\mathbf{f}_{\Gamma_k}^r = \sum_{q=1}^Q \tilde{\Psi}_r^q(\boldsymbol{\mu}) \mathbb{F}_\Gamma^q, \quad r = 1, \dots, R, \quad k = 1, \dots, R-1. \quad (4.5.10)$$

The vectors $\tilde{S}_r^T \mathbb{F}^q \in \mathbb{R}^{N \times 1}, \mathbb{F}_F^q \in \mathbb{R}^{N_F \times 1}$ and $\mathbb{F}_\Gamma^q \in \mathbb{R}^{N_\Gamma \times 1}, q = 1, \dots, 4$ can be computed and stored offline. Finally in the online stage, as in the previous case, for a new value of $\boldsymbol{\mu}$ and for an arbitrary number R of subdomains, we assemble the matrices $M_r, M_{rF_r}, \mathbb{M}_{F_r}, \mathbb{M}_{F_r\Gamma_k}, \mathbb{M}_{\Gamma_k\Gamma_k}^r$ and the vectors $\mathbf{f}_r, \mathbf{f}_{F_r}$ and $\mathbf{f}_{\Gamma_k}^r$ for $r = 1, \dots, R$ and $k = 1, \dots, R-1$, that will define respectively the matrices $M_{rr} \in \mathbb{R}^{RN \times RN}, M_{rF} \in \mathbb{R}^{RN \times RN_F}, A_{FF} \in \mathbb{R}^{RN_F \times RN_F}, M_{F\Gamma} \in \mathbb{R}^{RN_F \times (R-1)N_\Gamma}, M_{\Gamma\Gamma} \in \mathbb{R}^{(R-1)N_\Gamma \times (R-1)N_\Gamma}$ and the vectors $\mathbf{f}_{rr} \in \mathbb{R}^{RN \times 1}, \mathbf{f}_F \in \mathbb{R}^{RN_F \times 1}$ and $\mathbf{f}_\Gamma \in \mathbb{R}^{(R-1)N_\Gamma \times 1}$ and we solve the linear system (4.4.20).

Summarizing the quantities that determinate the dimension of the final system (4.4.20) (to be solved in order to find the RDF solution of the problem) are:

- the number of subdomains R ,
- the number of nodes along each internal interface N_Γ (\mathbb{P}_1 and \mathbb{P}_2 nodes),
- the number of nodes involved in the FE region N_F (\mathbb{P}_1 and \mathbb{P}_2 nodes),
- the number of local reduced basis function $N = 3N_{RB}$,

such that the dimension of the linear system (4.4.20) is $(R(N + N_F) + (R-1)N_\Gamma) \times (R(N + N_F) + (R-1)N_\Gamma)$. We note that the dimension of the system (4.4.20) is a bit bigger than the dimension of the system (4.4.11) (depends on RN_F), it deals a bigger online computational cost but, at fixed N , by increasing N_F we reach a better accuracy of the solution.

4.6 Numerical results

We present in this section some numerical results dealing with the RDF approximated solution of the Stokes problem 4.1.1. We consider several sets of reduced basis functions (Fourier functions for both Dirichlet or Neumann BCs) and the corresponding results. Moreover, numerical tests involving different FE portions of the domain will follow, together with a discussion about the computational time savings with respect to the FE full order method.

4.6.1 The local BCs and the local greedy algorithms

We introduce here some possible choices of the local BCs set for the problems (4.4.21) or (4.4.31). The spaces built by the solutions of these two possible problems constitute the spaces in which we find the RDF solution of problem (4.1.1). We discuss how each space affects the approximate solution for different combinations of parameters that define the whole computational domain.

As already mentioned the numerical results performed for both the Laplace and Stokes problems, induced us to prefer the Fourier interface conditions. Despite that, in this section we show which choice of parametric BC (Dirichlet, problem (4.4.21), or Neumann, problem (4.4.31)) appears more convenient to compute the local reduced spaces.

Due to the physical features of this particular problem, we observe that principal direction of the flux flow in the region of the internal interfaces is normal to the interface. So that along the internal interfaces the velocity and stress solution is mainly defined by their normal component. For this reason in both cases (Dirichlet or Neumann) the tangential component of parametric function λ_t is assumed to be equal to zero and the Fourier functions are assigned only on the normal component λ_n .

4.6.1.1 The Dirichlet Fourier interface functions

The first main option that we have adopted is the use of the Fourier functions as Dirichlet BCs, or in other words to consider the problem (4.4.21) for the computation of the local reduced basis functions. As in the Laplace case we fix number M of possible functions to consider. The function λ , when it is different from zero, represents the Fourier basis function defined on $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$. In particular, if we consider a set of M different local BCs, we define λ_n as follows: if $\mu_\lambda = 1, \dots, M/2$

$$\lambda_n(\mathbf{x}, \mu_\lambda) = \begin{cases} \cos(m(\mu_\lambda)\pi\mathbf{x}) & \mathbf{x} \in \hat{\Gamma}_T, m(\mu_\lambda) = \mu_\lambda - 1, \\ 0 & \mathbf{x} \in \hat{\Gamma}_B, \end{cases} \quad (4.6.1)$$

if $\mu_\lambda = M/2 + 1, \dots, M$

$$\lambda_n(\mathbf{x}, \mu_\lambda) = \begin{cases} 0 & \mathbf{x} \in \hat{\Gamma}_B, \\ \cos(m(\mu_\lambda)\pi\mathbf{x}) & \mathbf{x} \in \hat{\Gamma}_T, m(\mu_\lambda) = \mu_\lambda - M/2 - 1. \end{cases} \quad (4.6.2)$$

Figure 4.5 shows the normal component of the function λ_n for the values of $\mu_\lambda = 1, \dots, 6$.

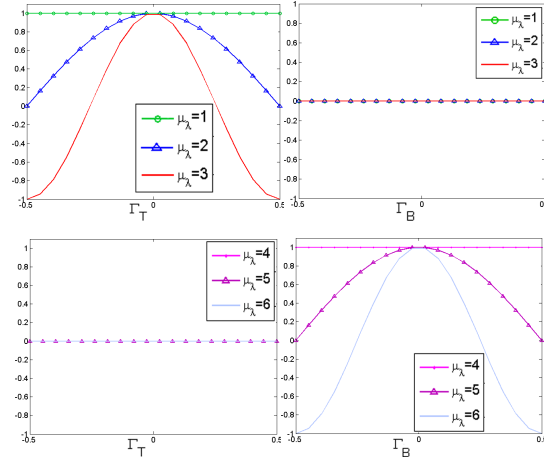


Figure 4.5: Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for $M=6$ and $\mu_\lambda = 1, \dots, 6$.

Figures 4.6 and 4.10 show, for a fixed geometrical parameter μ , the solutions $(\mathbf{u}(\mu, \mu_\lambda), p(\mu, \mu_\lambda))$ (and the velocity streamlines) of (4.4.21) associated to the value $M = 6$ and $\mu_\lambda = 1, \dots, 6$.

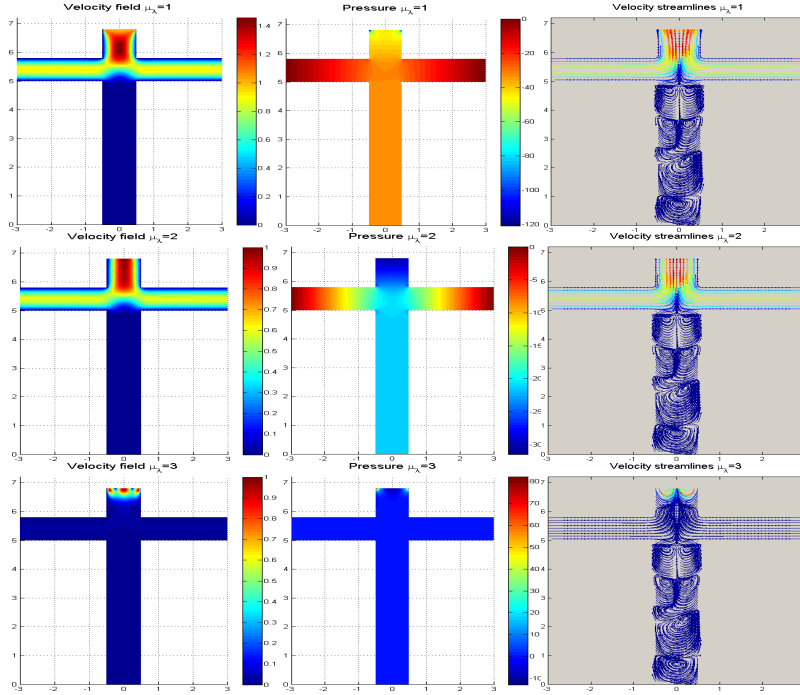


Figure 4.6: Velocity field $\mathbf{u}(\mu, \mu_\lambda)$, pressure $p(\mu, \mu_\lambda)$ and velocity streamlines obtained solving problem (4.4.21) with $\mu_\lambda = 1, 2, 3$.

Figure 4.8 shows the distribution of the parameter values (μ^i, μ_λ^i) selected during the greedy algorithm, by using the Dirichlet parametric BCs and $M = 6$. We note that the values of the parameter μ_λ have a bigger concentration when they refer to the parametric BC different from zero in $\hat{\Gamma}_T$ that is the boundary close to the geometrical deformation of the domain.

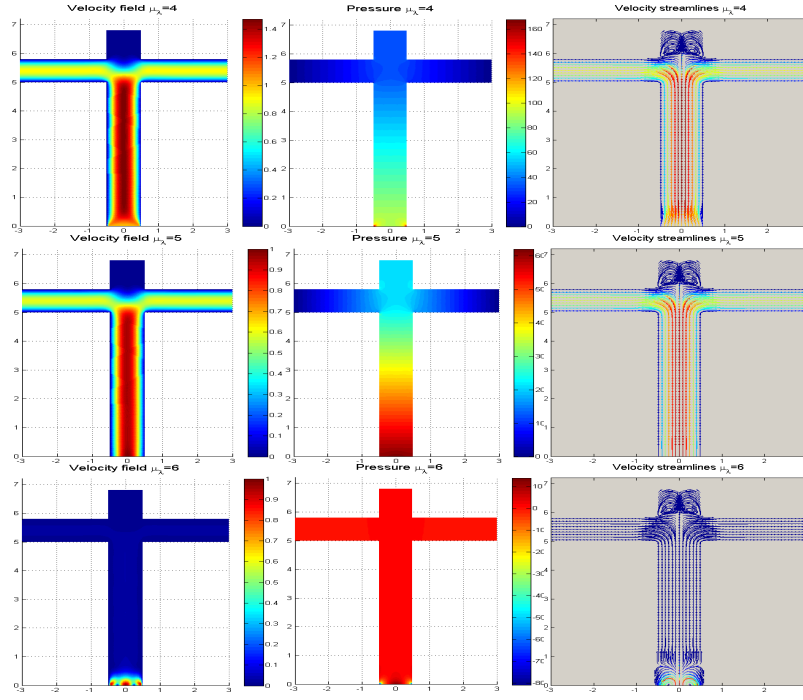


Figure 4.7: Velocity field $\mathbf{u}(\mu, \mu_\lambda)$, pressure $p(\mu, \mu_\lambda)$ and velocity streamlines obtained solving problem (4.4.21) with $\mu_\lambda = 4, 5, 6$.

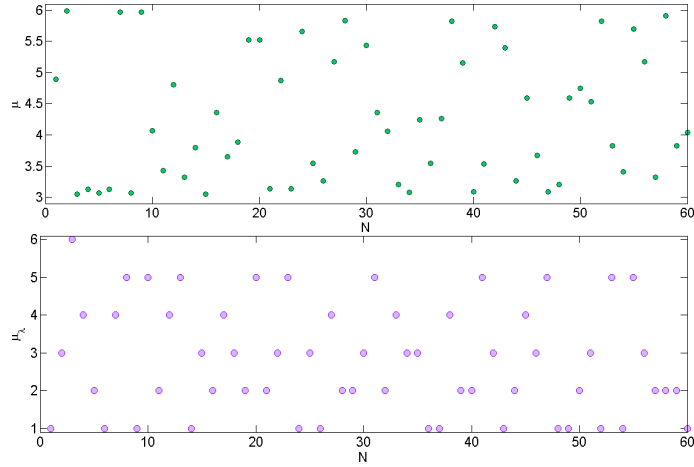


Figure 4.8: Parameter values of the selected μ^i and μ_λ^i , $i = 1, \dots, 60$ during the greedy algorithm.

Figure 4.9 shows the average of the relative errors between the RB local solution and the FE one obtained by solving the local problem (4.4.21) (for 100 parameter values (μ^i, μ_λ^i)) and by using the different Fourier functions as Dirichlet boundary conditions in dependence of number of reduced basis functions, computed by the greedy algorithm. We note that in correspondence of a larger range of possible values of μ_λ (larger M , bigger value of $\max \mu_\lambda$), we need a larger set of basis functions to reach a selected accuracy of the solution.

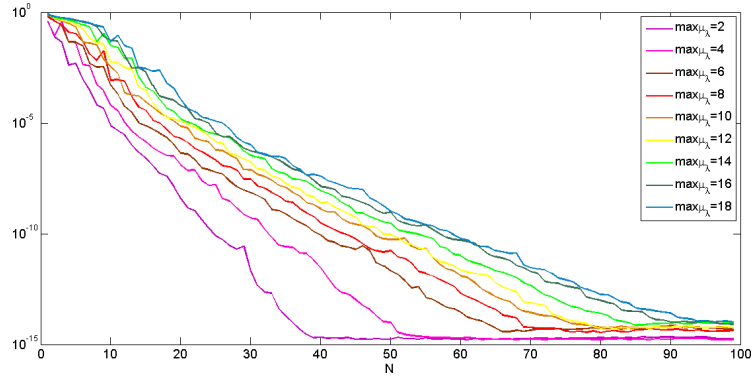


Figure 4.9: Average of relative error between the RB local solution and the FE one of problem (4.4.21) on a test of 100 samples set, by using parametric Dirichlet BC.

4.6.1.2 The Neumann Fourier interface functions

The second main option that we have adopted is the use of the Fourier functions as Neumann BCs, that means we are considering the problem (4.4.31) for the computation of the local reduced basis functions. We refer to the function λ_n defined in (4.6.1) and (4.6.2).

Figures 4.10 and 4.11 show, for a fixed geometrical parameter μ , the solutions $(\mathbf{u}(\mu, \mu_\lambda), p(\mu, \mu_\lambda))$ (and the velocity streamlines) of (4.4.31) associated to the value $M = 6$ and $\mu_\lambda = 1, \dots, 6$.

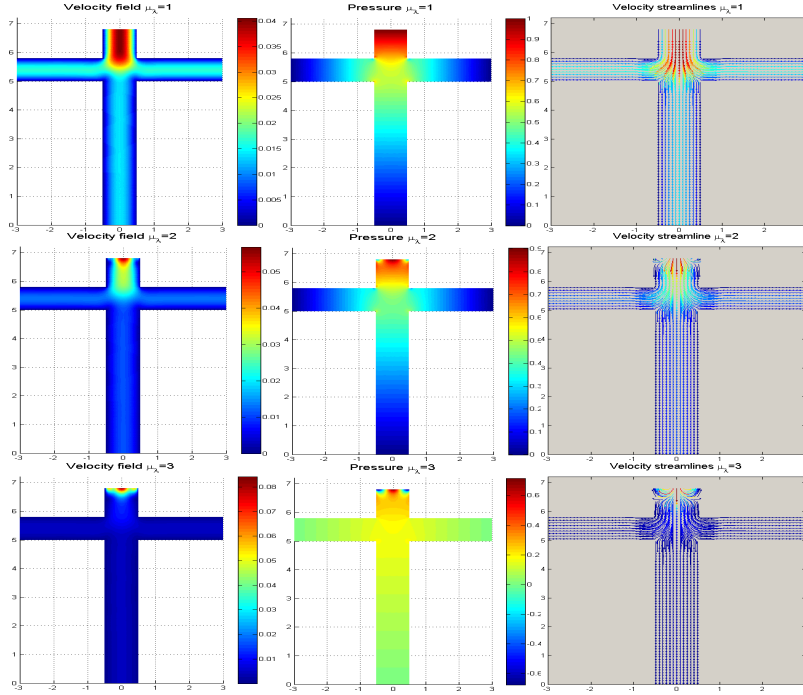


Figure 4.10: Velocity field $\mathbf{u}(\mu, \mu_\lambda)$, pressure $p(\mu, \mu_\lambda)$ and velocity streamlines of the solutions of problem (4.4.21) with $\mu_\lambda = 1, 2, 3$.

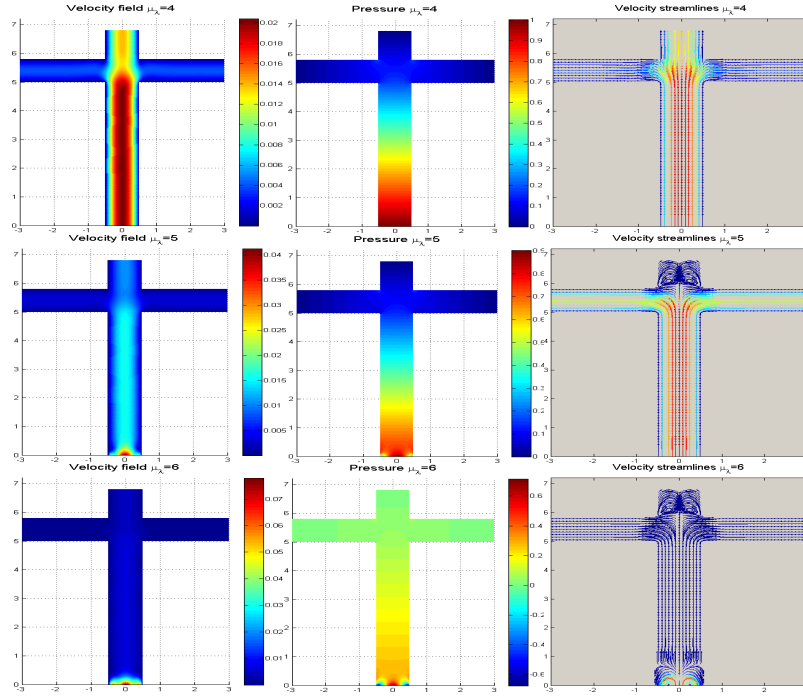


Figure 4.11: Velocity field $\mathbf{u}(\mu, \mu_\lambda)$, pressure $p(\mu, \mu_\lambda)$ and velocity streamlines of the solutions of problem (4.4.21) with $\mu_\lambda = 4, 5, 6$.

Figure 4.12 shows the distribution of the parameter values (μ^i, μ_λ^i) selected during the greedy algorithm, by fixing $M = 6$. The same considerations provided for the Dirichlet case, about the parameter values concentration of μ_λ , can be observed here.

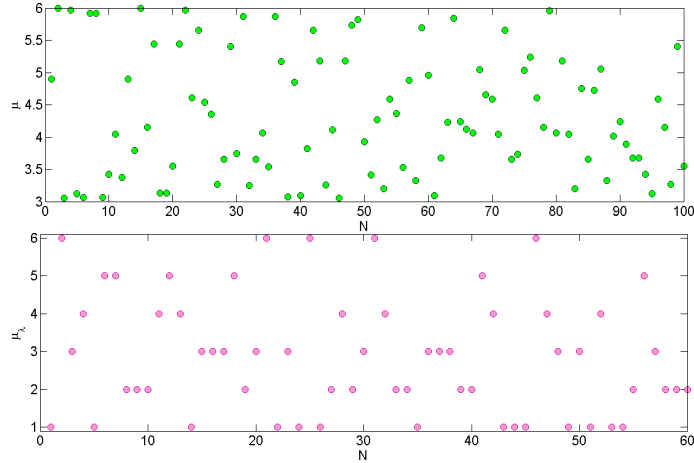


Figure 4.12: Parameter values of the selected μ^i and μ_λ^i , $i = 1, \dots, 60$ during the greedy algorithm.

Figure 4.13 shows the average of the relative errors between the solution obtained by solving the RB local problem (4.4.31) (by using the basis functions selected by the greedy algorithm) and the FE local solution on a test of 100 parameter samples. The case of different values of M are reported in the same figure, we note that for larger range of μ_λ values (bigger value of $\max \mu_\lambda$), a larger set of basis functions

is necessary to reach a selected accuracy error.

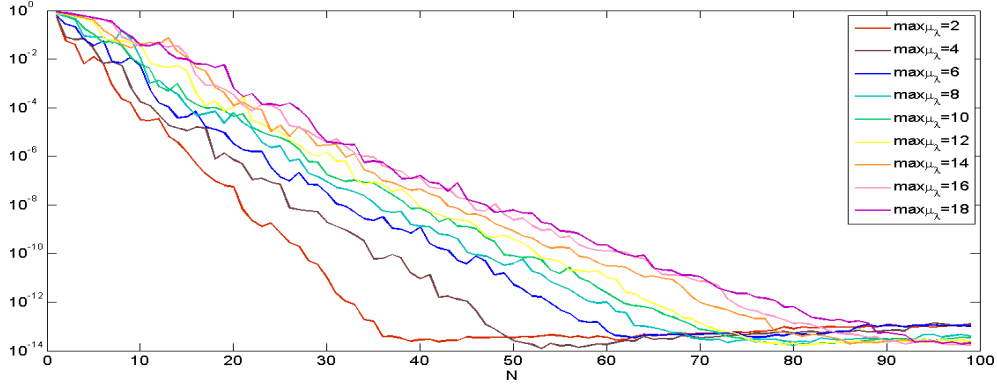


Figure 4.13: Average of relative error between the RB local solution and the FE one of problem (4.4.31) for different sets of local basis function, by setting λ as Neumann BC.

4.6.2 Global solution and accuracy results

In the previous section we have introduced two different main ways to compute the local reduced basis functions, the former by considering the normal component of the velocity along the internal interfaces as parameter, and the latter by considering the normal component of the normal stress along the internal interfaces as parameter. These two approaches (regarding problem (4.4.21) and problem (4.4.31), respectively) provide (through the greedy algorithm) two independent sets of local basis functions that can be used separately or not for solving the final RDF Stokes problem.

In order to investigate the quality of these basis functions we define the reduced basis spaces in three different ways:

- the spaces, denoted by DBC, built by the solutions of problem (4.4.21) selected by the greedy algorithm (μ_λ involved only in the local Dirichlet BCs),
- the spaces, denoted by NBC, built by the solutions of problem (4.4.31) selected by the greedy algorithm (μ_λ involved only in the local Neumann BCs),
- the spaces, denoted by DNBC, built by the functions of the spaces DBC and NBC.

We have performed, in different and independent offline stages, several greedy algorithms, in order to collect the different reduced basis spaces. In particular, we have always used a fine mesh of 3537 nodes used for \mathbb{P}_1 finite elements and 13729 nodes used for \mathbb{P}_2 finite elements, that involves 17 interface nodes \mathbb{P}_1 and 31 interface nodes \mathbb{P}_2 , respectively. We have considered 9 options of the Fourier functions used as Dirichlet BC and 9 options of the Fourier functions used as Neumann BC, by imposing $M = 2, 4, 6, \dots, 18$. Then we have collected 27 different local reduced basis spaces: 9 spaces of the first type (DBC), 9 spaces of the second type (NBC) and 9 spaces of the third type (DNBC).

Finally we tested these spaces for solving the RDF method on the 3 blocks test network configuration to approximate the Stokes problem for a set of 100 parameter values μ by reporting the average of the errors between the approximate RDF solution and the FE one, always in norm H^1 for the velocity and L^2 for the pressure.

Figures 4.14 and 4.15 show the average relative errors (velocity and pressure respectively) between the RDF solution and the finite element solution obtained as function of the set of local bases considered and the number of reduced basis functions, by using the Dirichlet Fourier functions set.

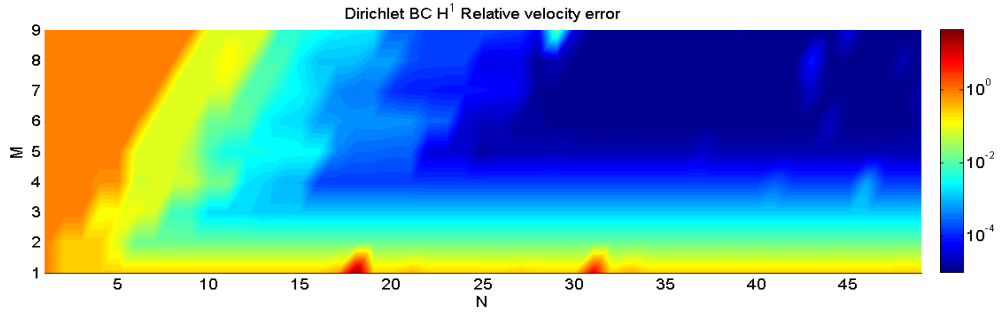


Figure 4.14: Velocity relative error between the RDF solution and the FE solution by using the DBC spaces.

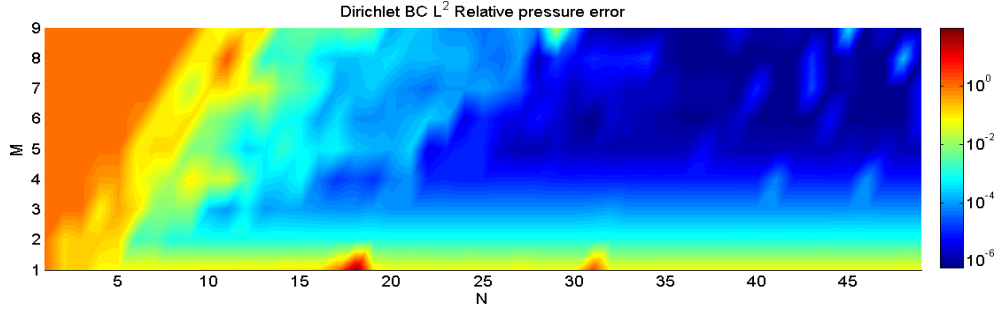


Figure 4.15: Pressure relative error between the RDF solution and the FE solution by using the DBC spaces.

Figures 4.16 and 4.17 show the relative errors (velocity and pressure, respectively) between the RDF solution and the FE one as function of M and the number of RB functions, by using the NBC sets.

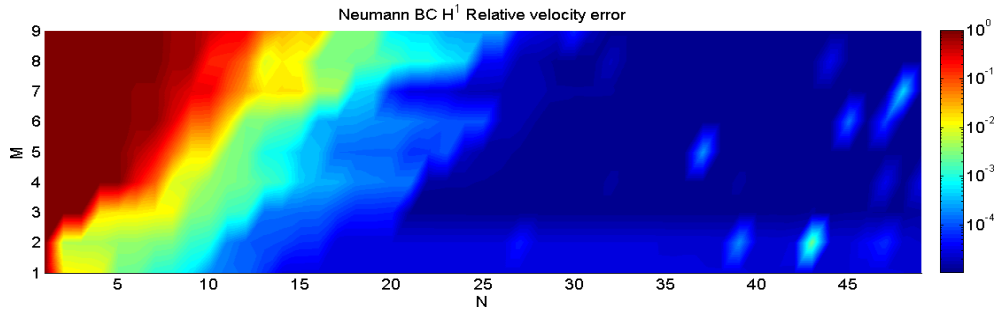


Figure 4.16: Velocity relative error between the RDF solution and the FE solution by using the NBC spaces.

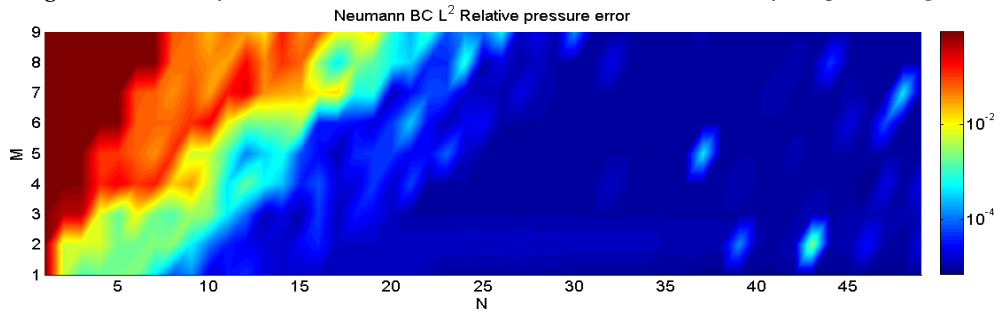


Figure 4.17: Pressure relative error between the RDF solution and the FE solution by using the NBC spaces.

Figures 4.18 and 4.19 show the same relative errors between the RDF solution and the FE one as function of M and the number of RB functions, by using the DNBC spaces.

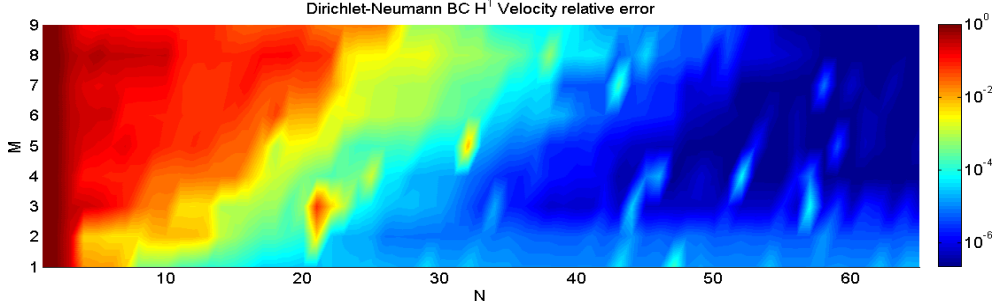


Figure 4.18: Velocity relative error between the RDF solution and the FE solution by using the DNBC spaces.

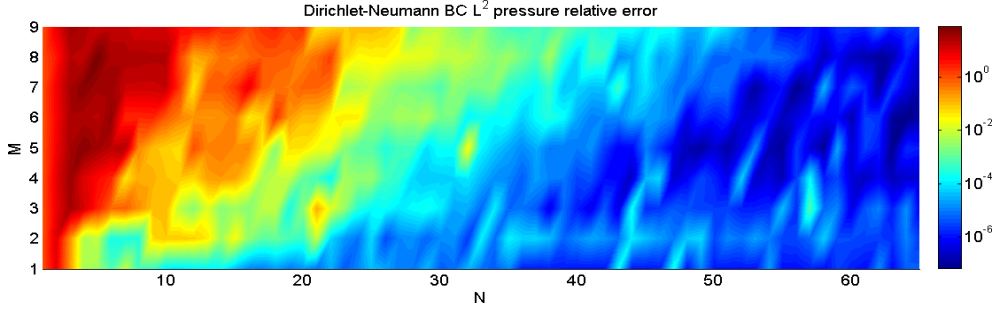


Figure 4.19: Pressure relative error between the RDF solution and the FE solution by using the DNBC spaces.

We note that by using the NBC option we reach a lower error value with respect to the DBC option with the same number of N basis functions. For instance, by using the DBC option, we recover an accuracy of 10^{-4} (for velocity and pressure) with $M \geq 5$ and $N \geq 25$, while by using the NBC the needed number of basis functions is 15 with $M \geq 1$. In the case of DNBC we have for $N=15$ and $M \geq 1$ the same level of accuracy of the NBC option, moreover it is possible to reach even a lower relative error for both velocity and pressure by increasing further on the number of basis functions ($\sim 10^{-7}$).

Figure 4.20 shows the velocity field, the pressure and the velocity streamlines of the RDF solution with the DNBC functions set for $\mu = [8.35, 4.78, 3.17]$ and by choosing $N=35$ and $M=5$.

In order to better understand the quality of the obtained solutions, we compare in Figure 4.21 the error along the first internal interface, denoted by Γ_1 (see Figure 4.2) by plotting the velocity field absolute error and the pressure absolute errors between the RDF solution and the FE one. We take for this visualization, the RDF solution found by using different number of basis functions and the sets of DNBC functions with $M=2$.

The plots presented in Figure 4.21 show how, by increasing the number of the basis functions (N), the RDF solution approximates the FEM solution and which is the profile of the corresponding errors. Due to the fact that, for this comparison, we used the third set of basis functions (DNBC) with $M=2$, we note that the minimum error presents a profile strongly affected by the chosen Fourier functions.

Further tests, regarding similar comparisons along the internal interface Γ_1 , by varying the set of local basis functions, are presented in Figures 4.22, 4.23, 4.24, 4.25, 4.26, 4.27 this time in order to visualize the final error profiles we start directly by plotting the errors found with an enough large number of basis functions ($N \geq 30$).

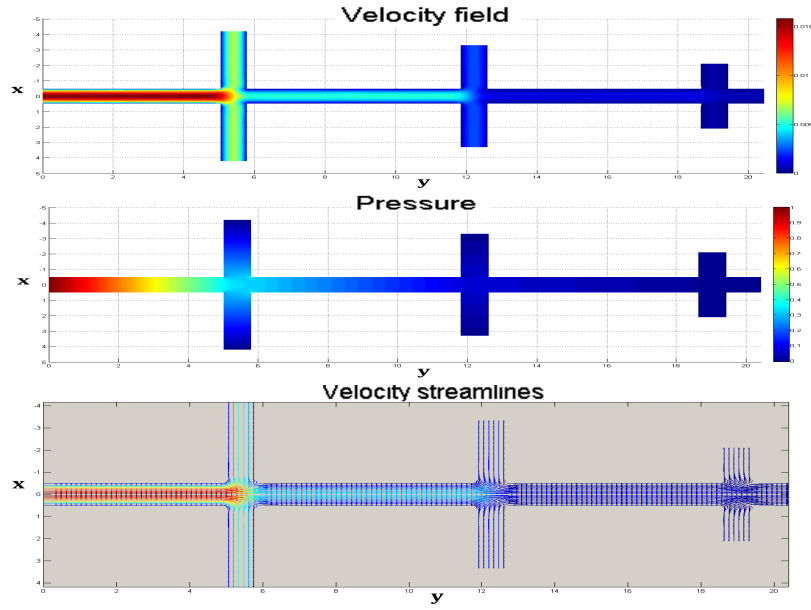


Figure 4.20: Representative RDF solution (velocity field, pressure and velocity streamlines) by using $N=35$, $M=5$ and NDBC set.

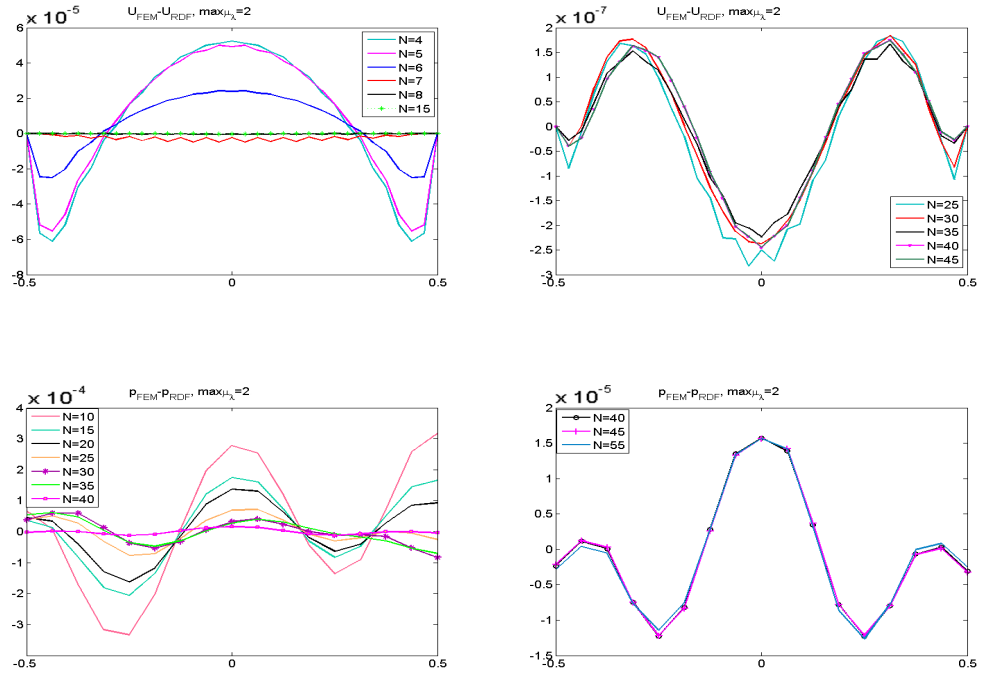


Figure 4.21: Plot of the functions $u_{FEM} - u_{RDF}$ and $p_{FEM} - p_{RDF}$ along the internal interface Γ_1 by varying the number of basis function and by using the set of Neumann BC with $M=2$.

In particular, Figures 4.22 shows the error plots of the tangential and normal components of the velocity field with respect to the FE solution by using different DBC sets of RB functions, with $M = 2, 4, \dots, 14, 18$. We omit the case $M = 16$ that is similar to the case $M = 18$.

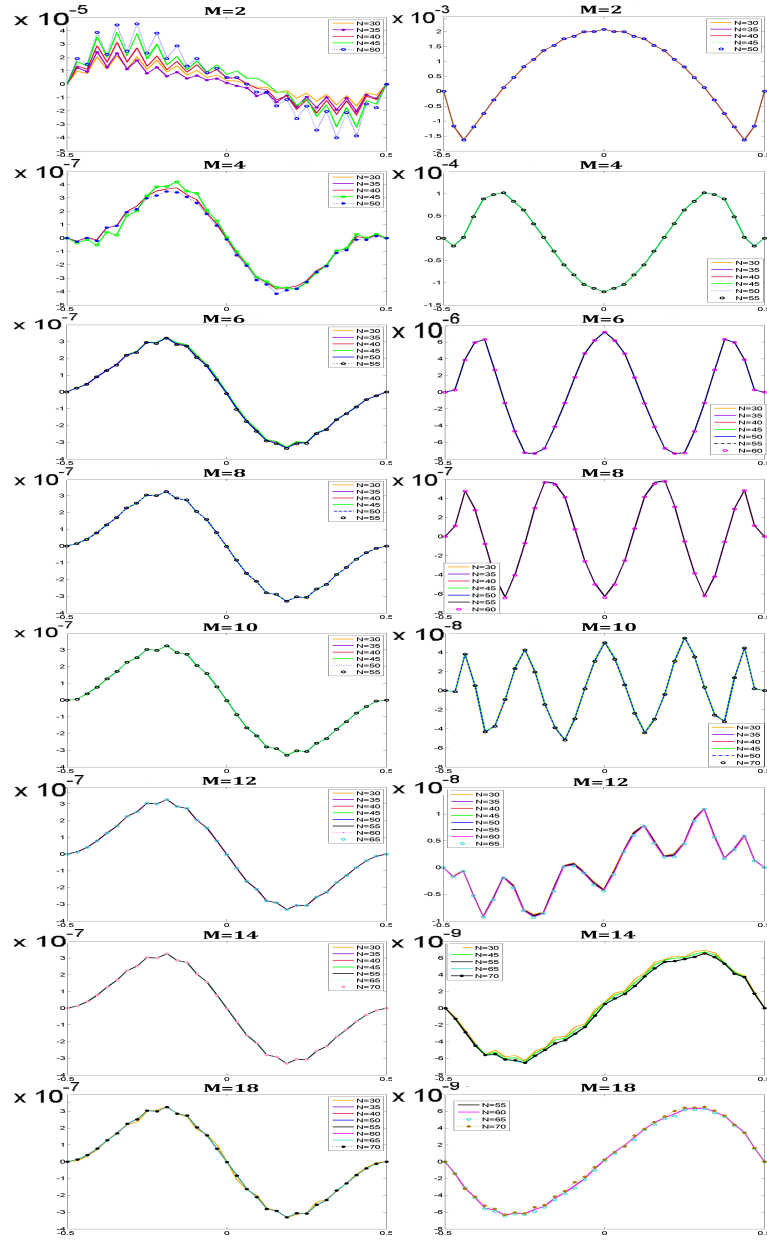


Figure 4.22: Plot of the functions $u_{xFEM} - u_{xRDF}$ and $u_{yFEM} - u_{yRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Dirichlet BC with $M=2, \dots, 18$.

We recall that the BCs are set equal to zero along the tangential component of the velocity, this justifies the fact that the plots on the left part of Figure 4.23 are similar and the error does not decrease by increasing the number of Fourier functions (M), because the tangential component of the velocity (x component in the considered test case) remains equal to zero. On the other hand, the plots on the right part of Figure 4.23 show that, by increasing the value of M , the errors regarding normal component of the velocity decrease, because we are using a larger number of Fourier functions to approximate this component. We note that the error profiles are affected by the Fourier bases selected.

In Figure 4.23 we can observe how the different values of M affect the error plots of the normal and tangential components of the normal stress respect with the FE solution, by using the DBC set of basis functions and by varying $M = 2, 4, \dots, 18$.

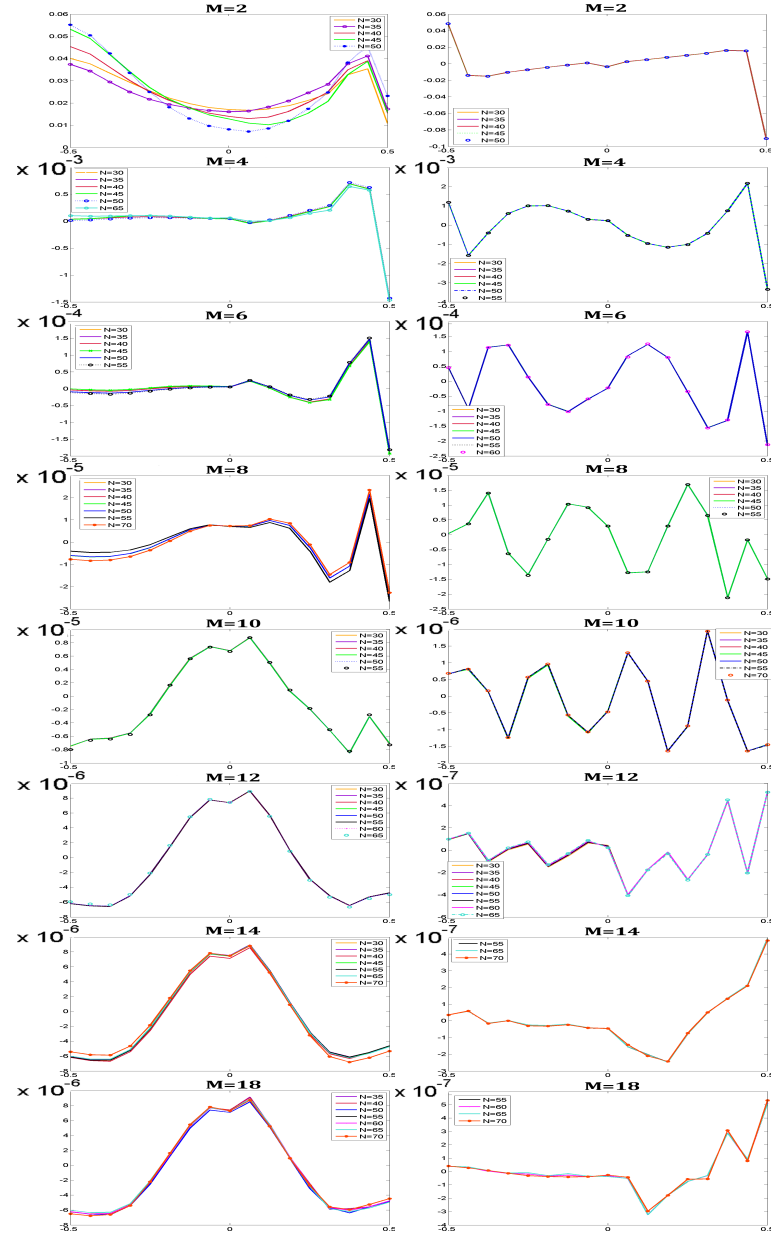


Figure 4.23: Plot of the functions $\sigma_{nFEM} - \sigma_{nRDF}$ and $\sigma_{tFEM} - \sigma_{tRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Dirichlet BC with $M=2, \dots, 18$.

Figures 4.24, 4.25 show the same plots errors by using the NBC sets of basis functions, with $M = 2, 4, \dots, 14, 18$. We note that the Fourier local BCs strongly affect the normal component of the normal stress errors.

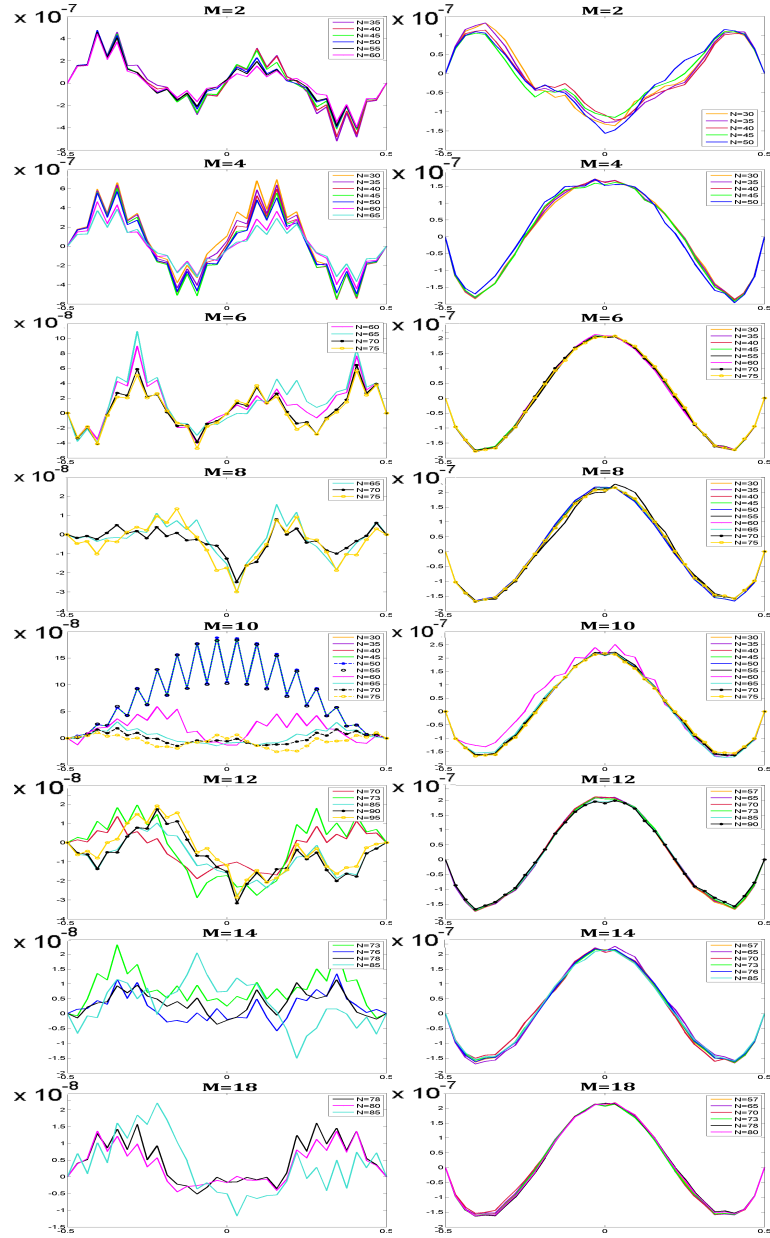


Figure 4.24: Plot of the functions $u_{xFEM} - u_{xRDF}$ and $u_{yFEM} - u_{yRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Neumann BC with $M=2, \dots, 18$.

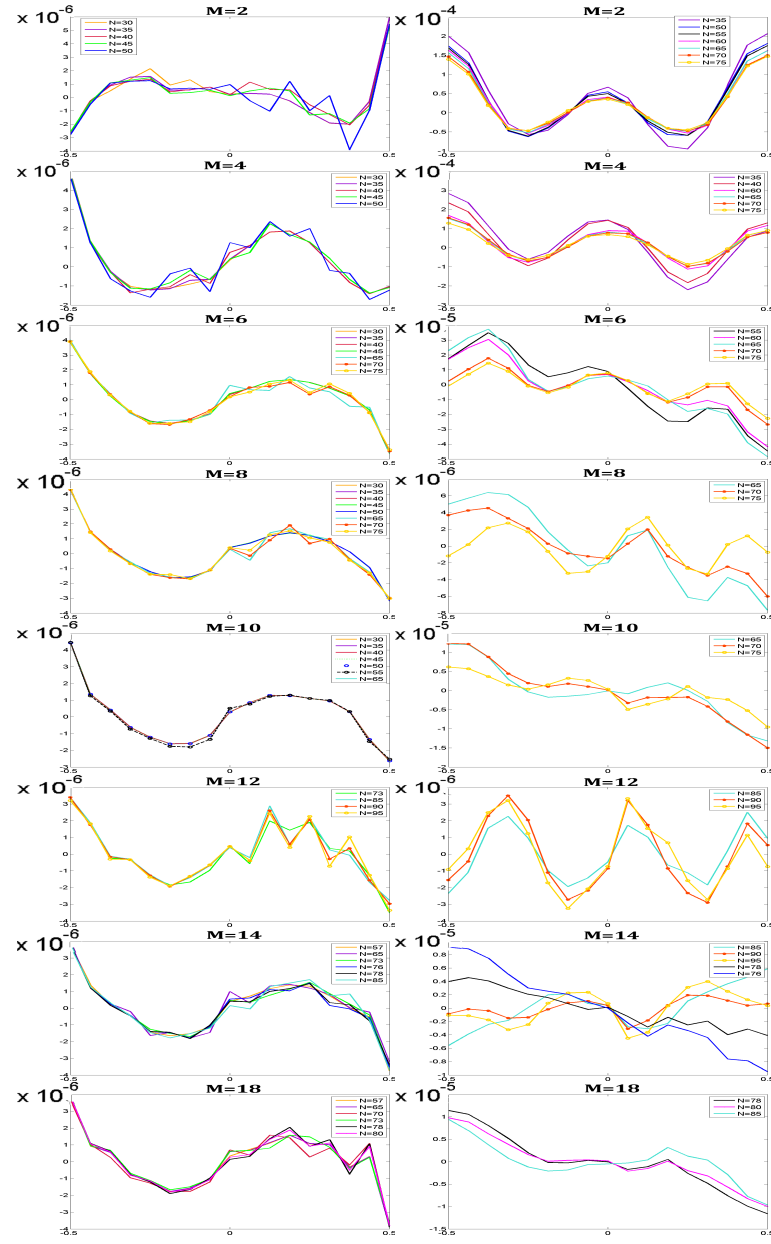


Figure 4.25: Plot of the functions $\sigma_{nFEM} - \sigma_{nRDF}$ and $\sigma_{tFEM} - \sigma_{tRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Neumann BC with $M=2, \dots, 18$.

Figures 4.26, 4.27 show the plots errors of the velocity field and the normal stress by using the DNBC sets of basis functions, with $M = 2, 4, \dots, 14, 18$. We note here that the previous considerations can be observed for the normal components of both velocity and normal stress.

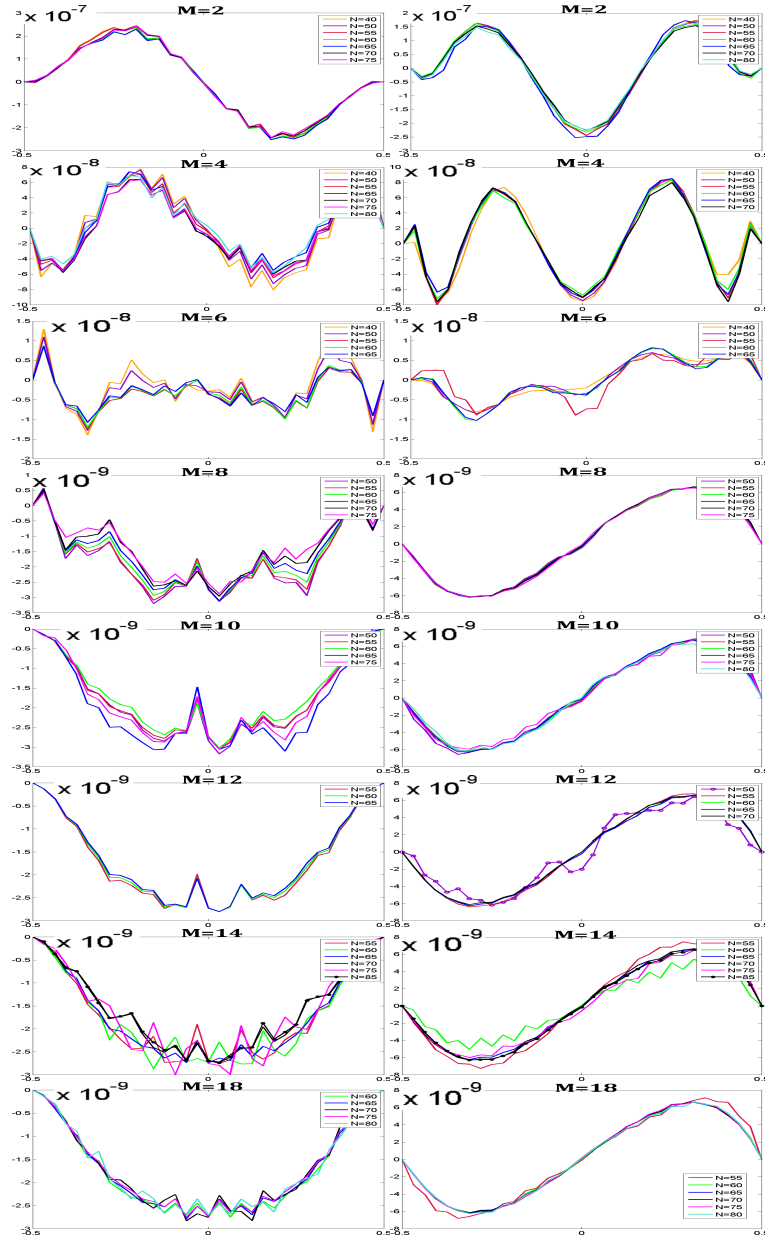


Figure 4.26: Plot of the functions $u_{xFEM} - u_{xRDF}$ and $u_{yFEM} - u_{yRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Dirichlet-Neumann BC with $M=2, \dots, 18$.

We can observe that we are able to well approximate the profiles of the FE solution along the internal interfaces when we use local RB functions associated to the solution of the local problems in which the corresponding profiles are imposed as parametric BCs. Due to the geometrical configuration of the considered domain, we have imposed the parametric function $\lambda(\mu_\lambda)$ in correspondence to the vertical velocity or of the normal component of the stress, having the most important influence on the solution. For different domain configurations or in order to further improve the accuracy of the method, we may enrich offline the reduced space with functions associated to problems involving a parametric profile also in correspondence to that other components. However, we recall that the range of values of the local parameters increases as well as the complexity of the offline stage.

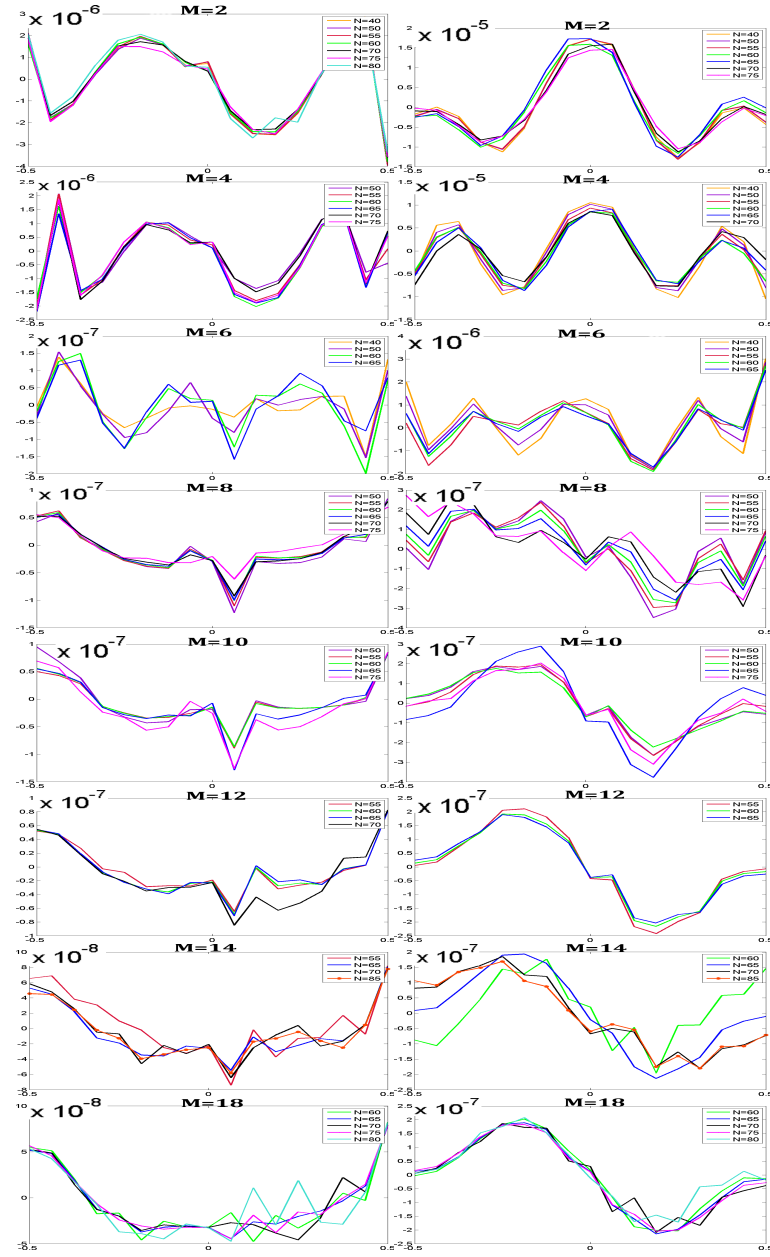


Figure 4.27: Plot of the functions $\sigma_{nFEM} - \sigma_{nRDF}$ and $\sigma_{tFEM} - \sigma_{tRDF}$ along the internal interface Γ_1 by varying the number of basis functions and by using the set of Dirichlet-Neumann BC with $M=2, \dots, 18$.

4.6.3 Extending the FE region

A range of tests regarding different extensions of the finite element regions has been tested. Figures 4.28 and 4.29 report the average relative error (velocity and pressure) between the RDF solution and the finite element solution by considering different extensions of $\hat{\Omega}_{FE}$.

The errors have been computed by varying the number of basis functions and by using respectively the spaces DBC and NBC. In these error plots we have denoted with L a value corresponding to the “layers” of nodes proximal to the internal interface that we consider in the FE region of the domain. Table 4.1 reports the exact number of \mathbb{P}_1 and \mathbb{P}_2 nodes corresponding to different values of L .

L	\mathbb{P}_1 nodes	\mathbb{P}_2 nodes	% \mathbb{P}_1 FE nodes	% \mathbb{P}_2 FE nodes
0	17	31	0.48	0.23
1	68	186	1.92	1.43
2	102	310	2.88	2.38
3	136	434	3.84	3.33
4	170	558	4.80	4.28
5	204	682	5.76	5.23
6	238	806	6.72	6.19
7	272	930	7.69	7.14

Table 4.1: Number of \mathbb{P}_1 and \mathbb{P}_2 nodes involved in different layers of FE regions (L).

Figure 4.28 shows the average error between the RDF solution and the FE solution by using the DBC sets of basis functions, by fixing $M=10$ and varying the number of basis functions (N) and the number of “layers” involved in the FE regions.

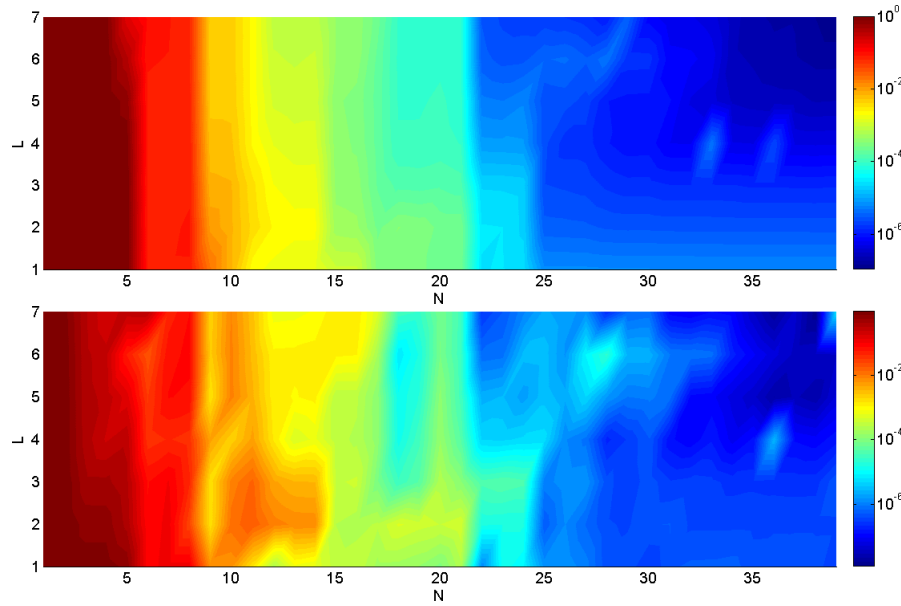


Figure 4.28: Relative error (velocity top, pressure bottom) between the RDF solution and the finite element solution by using Dirichlet BC ($M=10$) and by including different layers on the FE region (L).

Figure 4.29 shows the average error between the RDF solution and the FE solution by using the NBC sets of basis functions, by fixing $M=10$ and varying the number of basis functions (N) and the number of “layers” involved in the FE regions.

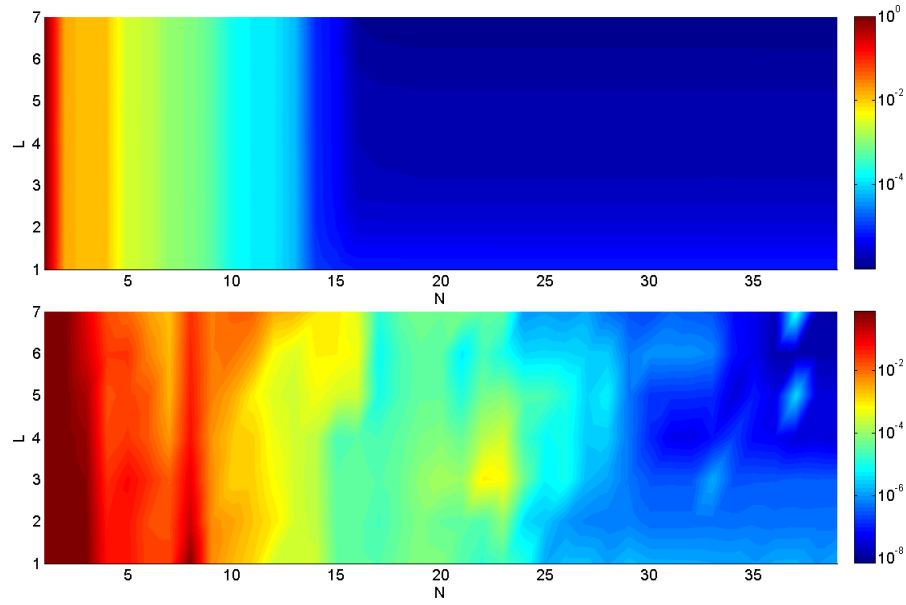


Figure 4.29: Relative error (velocity top, pressure bottom) between the RDF solution and the finite element solution by using Neumann BC ($M=10$) and by including different layers on the FE region (L).

These results show that by considering more FE nodes proximal to the internal interfaces at fixed number of local RB functions is not particularly effective for the DBC set of reduced space, indeed we can observe in Figure 4.28 that the error does not decrease considerably along the y-axes (by increasing the value of L). On the other hand, Figure 4.29 shows that in the case of NBC set, it is clearly visible that the velocity error decrease progressively by increasing the value of L when $N \geq 15$.

The influence of considering a bigger number of FE nodes is shown also in the Figures 4.30 and 4.31. Here we consider a fixed number of basis functions ($N=50$) and we vary the values of M and L , Figure 4.30 shows the relative average error between the RDF solution and the FE solution by using the set of DBC functions. The error decreases by increasing the value of L when $M \geq 5$.

Figure 4.31 shows the average error for the same configurations, by choosing the set of NBC basis functions, here the error decrease by increasing the value of L for each value of M .

These results, together with the results of the previous section show that if we use the DBC set, in order to have enough accuracy by varying the number of RB functions or the number of layers, a reasonable choice of M is $M \geq 5$, while for the NBC it is possible, by tuning the ingredients of the method, to reach a suitable accuracy for each value of M .

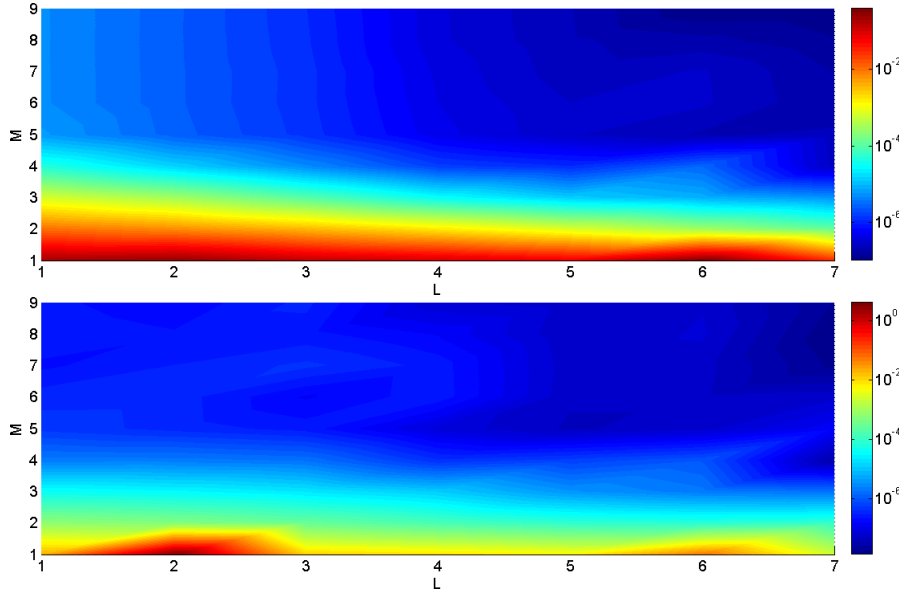


Figure 4.30: Relative error (velocity top, pressure bottom) between the RDF solution and the finite element solution by varying the number of local BCs and the percentage of nodes involved in the FE regions (L), by using the Dirichlet BC.

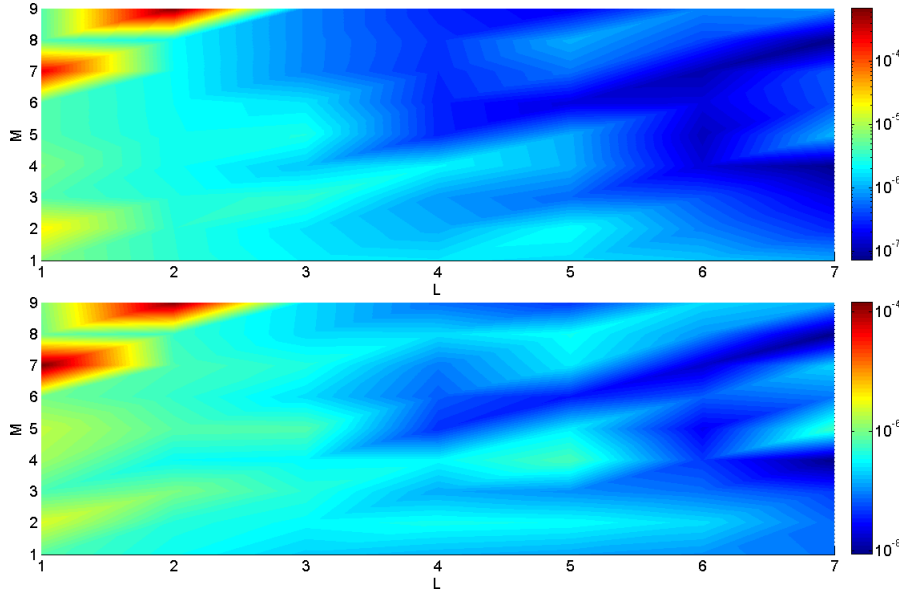


Figure 4.31: Relative error (velocity top, pressure bottom) between the RDF solution and the finite element solution by varying the number of local BCs and the percentage of nodes involved in the FE regions, by using the Neumann BC.

4.6.4 Computational times

Table 4.2 reports the computational times for computing the RDF solution and the percentage with respect to the *cpu time* for computing the FE solutions. The table contains the values for all the tested layers and two particular choices of N . We note that the number of FE solutions involved in the reduced basis space strongly affects the computational time needed to solve the RDF problem.

As observed in Section 4.5.2, the nodes involved in the FE region (N_F , here indicates in terms of layers L) affects the dimension of the linear system (4.4.11) of the RDF method, moreover its dimension depends also of the number (N) of RB functions used to approximate the solution. The case $L = 0$ indicates the situation in which the nodes involved in the FE region are only those associated to the internal interfaces.

A more general scenario regarding the variations of these variables (L and N) and the corresponding computational times of the RDF method as percentage of the FE method applied to the global problem is given by Figure 4.32 that shows the increasing of the computational time by varying both the number of local RB functions and the number of FE nodes. We observe that it is possible to combine the different options of the method in order to have a reasonable trade-off between the computational time and the accuracy of the solution. The selected set of reduced basis solutions does not affect the computational time.

L	N	cpu time (s)	% of FE cpu time	N	cpu time (s)	% of FE cpu time
0	20	0.11	1.55	50	0.26	3.6
1	20	0.48	6.73	50	0.94	13.15
2	20	0.92	12.88	50	1.22	17.15
3	20	1.07	15.01	50	1.44	20.25
4	20	1.08	15.14	50	1.63	22.9
5	20	1.25	17.58	50	1.66	23.3
6	20	1.42	19.99	50	1.83	25.71
7	20	1.68	23.55	50	1.87	26.18

Table 4.2: Cpu time (in seconds) for computing the RDF solutions, by varying the number of layers and by using $N=20$ and $N=50$ and comparison with the cpu time needed to compute the FE solution.

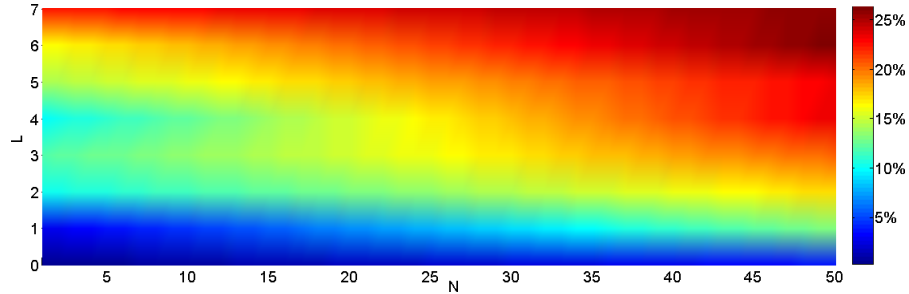


Figure 4.32: Computational time of the online RDF method as percentage of the cpu time needed to solve the online FE method applied to the global problem for a set of 100 tests, by varying the number of the considered FE functions (L) and the number of RB functions.

4.7 Concluding remarks

As we have seen in this chapter and in Chapter 3, the RDF method is particularly suitable for parametrized PDEs in networks of repetitive blocks with heterogeneous parametrization and has been successfully applied to second order elliptic problems, like the Laplace one and saddle point problems, like the Stokes one.

By exploiting the repetitiveness of few reference block the method reduces drastically the *offline* computational time with respect to global reduced basis approaches, especially for large networks.

In fact the reduced basis functions is pre-built locally on the few reference blocks whose space of

parameters is much reduced with respect to the global one. The method is quite flexible in the sense that modifying (i) the dimension of the local reduced bases, (ii) the number and the type of bases for the trace of the solution on the internal interface and (iii) the dimension of $\hat{\Omega}_{FE}$ it is possible to tune and balance the accuracy and the computational time. In general, by increasing the value of these parameters the accuracy of the numerical RDF solution is increased, however, the drawback is that also *online* and *offline* computational times may be increased.

In Chapter 3 we have investigated a first application of the RDF method to the thermal fin problem, by exploring the effectiveness and the computational convenience of two main type of local parametric BCs: the Lagrange and the Fourier options. It has been an important step to improve and extend the application of the RDF method to the steady Stokes problems. In this chapter we have exploited the Stokes problem with a simple geometry in order to focus our attention on the quality of the local BCs selected.

We conclude by observing that the attractive performances of the RDF method are maintained also in the case of the Stokes problems regarding both the accuracy and the computational time. This is quite interesting in view of applications to viscous flows in parametrized fluidic networks.

In the next chapter, the RDF method will be exploited by dealing with more complex parametrized geometries and with a larger number of subdomains in order to consider more realistic geometries. Together with numerical application of RB, RBHM and RDF methods, Chapter 5 will deal with a detailed comparison between the method RDF and RBHM in order to explore the effectiveness and the suitability of the proposed methodologies.

5 Some applications to more complex parametrized geometrical configurations

Flow simulations in pipelined channels and several kinds of parametrized configurations have a growing interest in many biological and industrial applications. The methodologies proposed in this thesis are particularly suitable for the study of internal flows in hierarchical parametrized geometries. Applications may be found in the analysis of the blood flow in specific compartments of the circulatory system that can be represented as a combination of few deformed vessels from reference ones (pipes and bifurcations), as shown in the sketch of Figure 5.1.

In this chapter we present some numerical examples based on the application of the proposed methods for the solution of steady Stokes equations. In particular we start by showing the application of the classical RB method in a 3D setting represented by a single domain, followed by some applications of the RBHM developed in Chapter 2 for 3D setting and the RDF method developed in Chapter 4 for complex 2D fluidic networks.

Finally, in order to provide a broad idea about RB methodologies (those already existing and those introduced in this thesis), this chapter contains a detailed comparison between the RB, RBEM, RBHM and RDF methods.

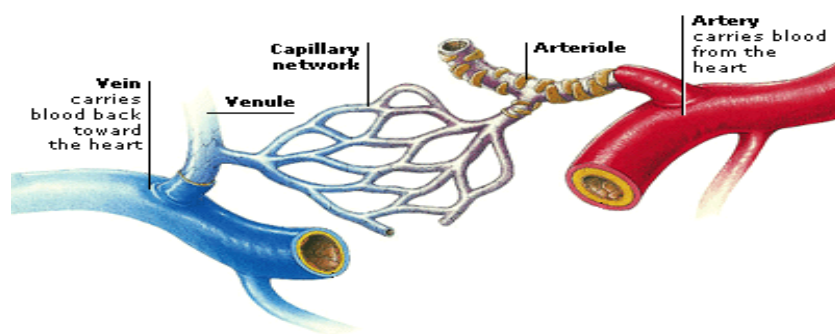


Figure 5.1: Sketch of a cardiovascular network, courtesy of <http://thecardiovascularsystem.wikispaces.com>

5.1 Transfinite mapping on 3D parametrized configurations

To begin we consider the classical reduced basis method, introduced in Chapter 1, for solving steady Stokes equations on a 3D single geometry. We show these tests in order to consider also 3D transfinite maps and to devote few considerations about the effectiveness of the classical reduced basis method applied to a single domain.

In the first test case we deal with a bifurcated geometry, in the second one with a curved pipe. Both geometries are parametrized through transfinite maps, introduced in Chapter 1, Section 1.3.3, here extended to the 3D setting. The affine decomposition of the problem is recovered thanks to the empirical interpolation method introduced in Section 1.2.3.1.

5.1.1 A bifurcating pipe

We consider a 3D bifurcation parametrized through two parameters that describe the possible deformations of one branch of the domain. In particular, $\mu_1 \in [6, 13]$ represents the length and $\mu_2 \in [-0.5, 0.5]$ the way to bend that part of the domain, as depicted in Figure 5.2. In order to grasp more insight on the meaning of the parameters, we show in Figures 5.3 and 5.4 some deformed geometries in correspondence to different values of the parameters, together with the selected reference geometry, corresponding to the values $\mu_1 = 9, \mu_2 = 0$ (shown in Figure 5.3).

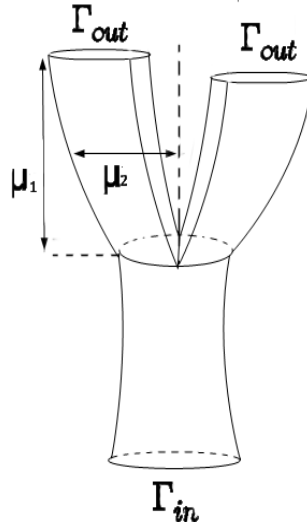
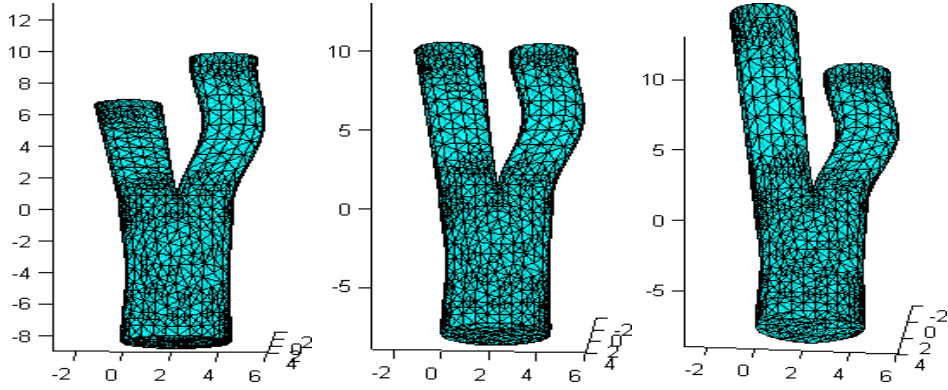
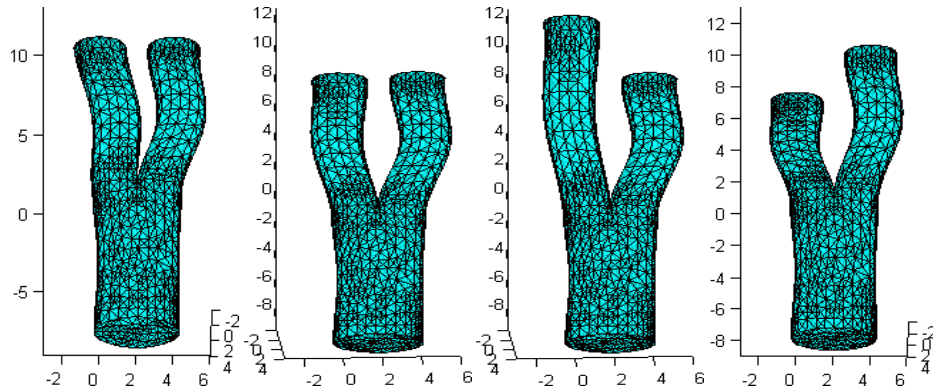


Figure 5.2: Geometrical scheme for the parametrized bifurcation.

We report here, only for this geometry, some results about the empirical interpolation method. It has been used to recover the affine decomposition property, as introduced in Chapter 1, Section 1.2.3, in particular it is applied to approximate the elements of the geometrical tensors with affine functions.


 Figure 5.3: Examples of deformed bifurcations: $\mu_1 = 6, \mu_2 = 0$ (left), reference domain (center), $\mu_1 = 13, \mu_2 = 0$ (right).

 Figure 5.4: Examples of deformed bifurcations: $\mu_1 = 9, \mu_2 = -0.5$ (first), $\mu_1 = 9, \mu_2 = 0.5$ (second), $\mu_1 = 13, \mu_2 = 0.5$ (third), $\mu_1 = 6, \mu_2 = 0.5$ (fourth).

The decoupling of the Stokes equations in the case of non-affine geometrical deformations has been introduced in Chapter 2, in Section 2.3 and we refer here to the tensors presented in (2.3.4) and (2.3.5) corresponding to the 3D setting. In particular, by denoting with \mathbf{J} the Jacobian of the geometrical transformation, we recall that the geometrical non-affine tensors are defined as follows:

$$\mathbf{v}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) = \mathbf{J}^{-1} \mathbf{J}^{-T} |\mathbf{J}|, \quad \boldsymbol{\chi}(\hat{\mathbf{x}}, \boldsymbol{\mu}_r) = \mathbf{J}^{-1} |\mathbf{J}|. \quad (5.1.1)$$

Thanks to the empirical interpolation method we can use the following approximate affine decompositions:

$$[\mathbf{v}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij} \approx \sum_{m=1}^{M_{ij}^a} \Theta_{ij}^m(\boldsymbol{\mu}) \tilde{\mathbf{v}}_{ij}^m(\hat{\mathbf{x}})$$

$$[\boldsymbol{\chi}(\hat{\mathbf{x}}, \boldsymbol{\mu})]_{ij} \approx \sum_{n=1}^{M_{ij}^b} \Phi_{ij}^n(\boldsymbol{\mu}) \tilde{\boldsymbol{\chi}}_{ij}^n(\hat{\mathbf{x}})$$

where M_{ij}^a and M_{ij}^b are the numbers of required terms that allow to approximate the function with a chosen level of accuracy. We show in Figure 5.5 how by increasing the number M_{ij}^a the error introduced by the approximations of the tensor elements of the tensor \mathbf{v} decreases.

For the tests shown in this section, we have used $M_{ij}^a = 20, i = 1, 2, 3, j = 1, 2, 3$ to approximate each element of the geometrical tensors.

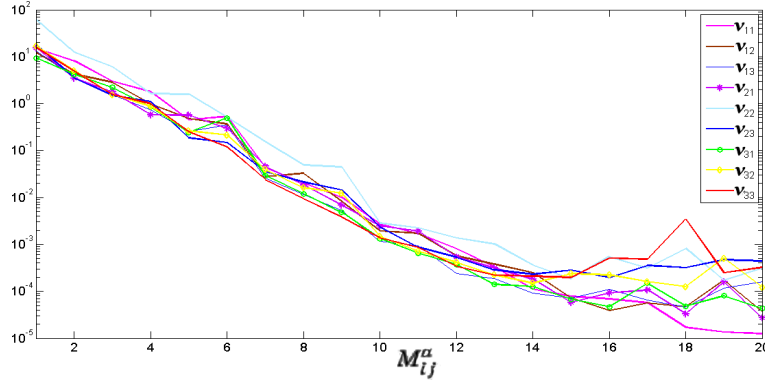


Figure 5.5: Empirical interpolation error for the approximation of the geometrical tensor elements \mathbf{v}_{ij} of (5.1.1) generated by the 3D TM.

We want now to compute the approximate solution of the Stokes problem by the reduced basis method. We impose homogeneous Dirichlet boundary conditions on the wall of the domain, homogeneous Neumann boundary conditions on the upper outflow parts (the end of the branches, indicated with Γ_{out} in Figure 5.2) and non-homogeneous Neumann boundary conditions equal to $[0, -1]$ on the bottom part, indicated with Γ_{in} in Figure 5.2.

Figure 5.12 shows the distribution of the parameter values selected during the greedy algorithm, by setting the maximum number of basis N_{max} equal to 32. We note that the values of μ_1 are uniformly distributed within the range of possible values, while the values of μ_2 have a bigger concentration close to the values 0.5 representing the maximum external curvature, as shown in Figure 5.4.

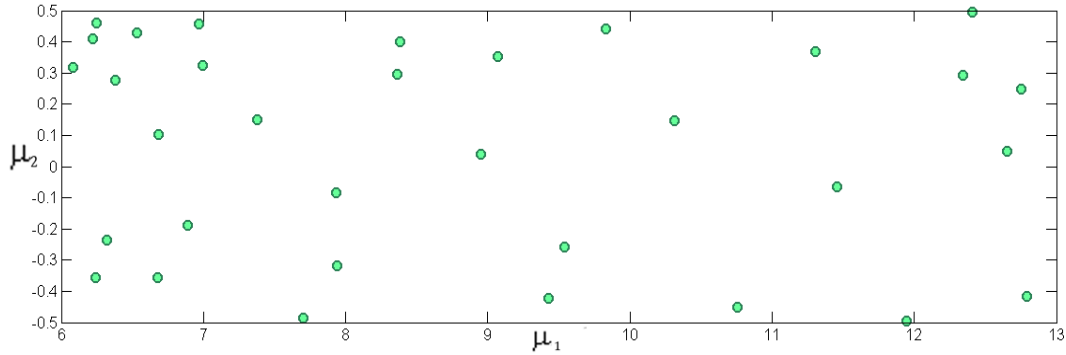


Figure 5.6: Distribution of the selected parameter values during the greedy algorithm.

Figure 5.13 shows the error (the sum of H^1 velocity relative error and L^2 pressure relative error) between the “truth” FE solution and the RB approximation obtained by solving the Stokes problem. In particular we report the minimum, the maximum and the average error obtained by solving the problem for a set of 100 different samples and by varying the number of reduced basis functions used to approximate the solution.

5.1. Transfinite mapping on 3D parametrized configurations

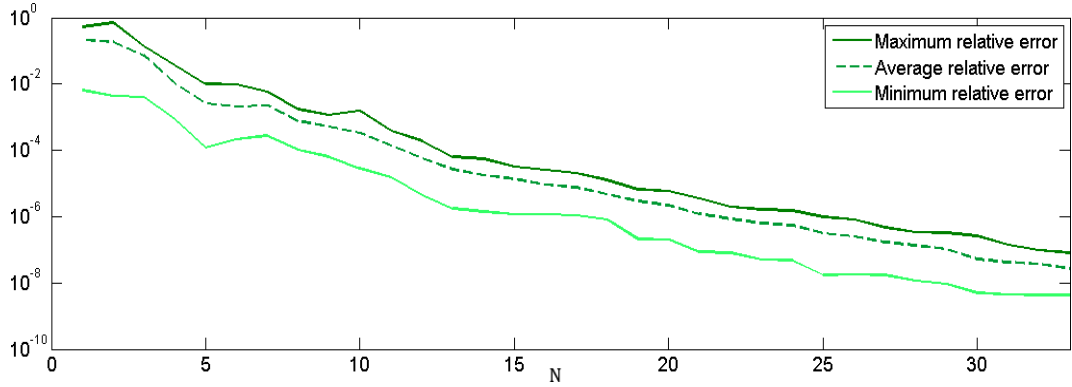


Figure 5.7: Minimum, maximum and average errors between the FE solution and the RB approximation for $N = 1, \dots, 32$, by considering a test set of 100 samples.

Figures 5.8 and 5.9 show two representative RB solutions of the Stokes problem corresponding to two parameter configurations: $\mu = (6, -0.48)$ and $\mu = (11, 0.5)$. For both figures in the upper part we report on the left the velocity field and on the right the pressure obtained by using $N=30$ basis functions. The corresponding errors with respect to the FE solutions are plotted in the bottom part of the figures, on the left the velocity error and on the right the pressure error, respectively.

We note that the dimension of the matrix that has to be solved in order to find the FE solution is $\mathcal{N} = 107803$, while by using $N=30$ basis functions in order to find the RB solution we need to solve (online) a linear system of dimension $3*N=90$ (number of velocity, supremizer and pressure snapshots). We report in Table 5.1 some information about the computational complexity of the RB method with respect to the FE method.

\mathcal{N}	cpu time of online FEM (s)	$3N$	cpu time of online RBM	time-ratio
107803	206	90	0.028	0.013%

Table 5.1: Computational details for the bifurcation geometry test.

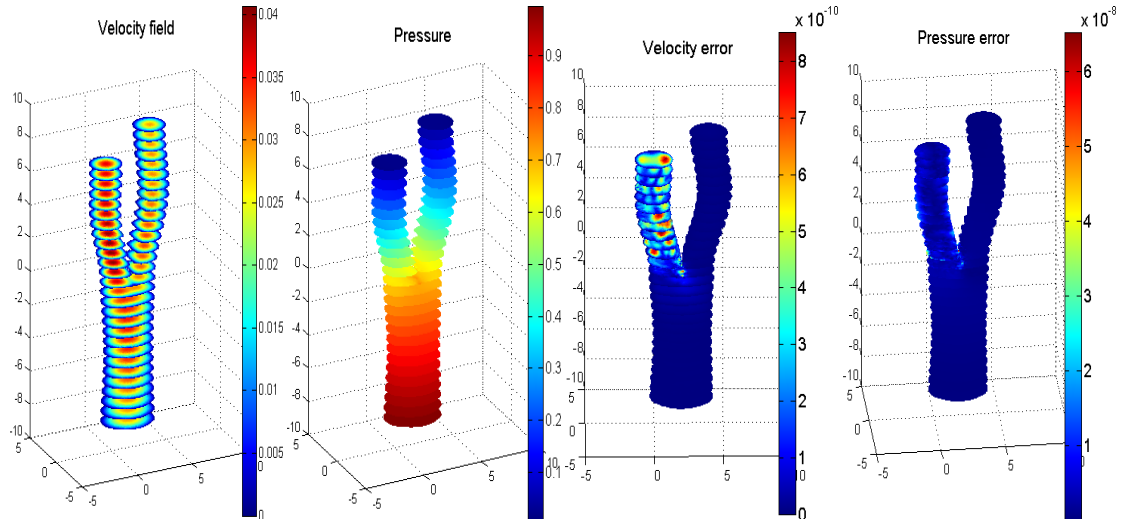


Figure 5.8: Representative RB solution (velocity field and pressure) obtained by using $N=30$ basis functions ($\mu_1 = 6, \mu_2 = -0.48$) and the corresponding error with respect to the FE solution.

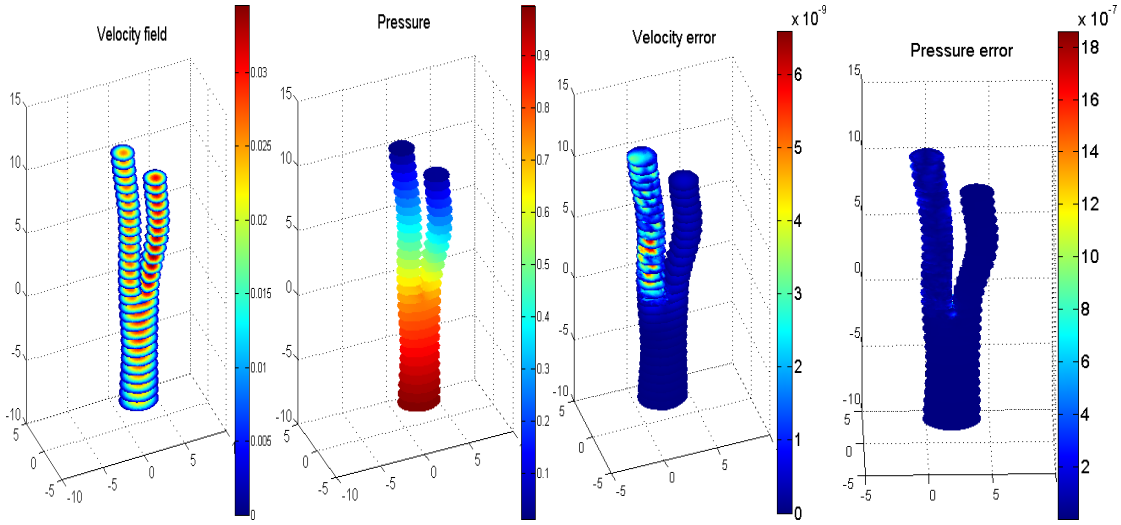


Figure 5.9: Representative RB solution (velocity field and pressure) obtained by using $N=30$ basis functions ($\mu_1 = 11, \mu_2 = 0.5$) and the corresponding error with respect to the FE solution.

5.1.2 Curved pipe geometry

We consider an example of 3D curved pipe geometry, parametrized through the 3D TM introduced in Chapter 1, Section 1.3.1.3, as the previous bifurcated geometry.

Here the two considered parameters represent the length and the bending of the pipe. In particular they refer to the position of the upper outflow surface: $\mu_1 \in [3, 5]$ indicates its position along the z-axes and $\mu_2 \in [3, 5]$ its position along the x-axes, as shown in Figure 5.10. Figure 5.11 contains the deformed pipe geometry in correspondence to some representative parameter values.

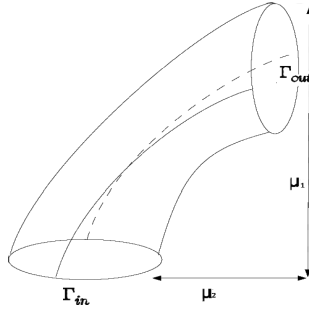


Figure 5.10: Geometrical scheme for the parametrized pipe

Also for this test case we want to approximate the solution of the steady Stokes problem by imposing homogeneous Dirichlet boundary conditions on the wall of the domain, homogeneous Neumann boundary conditions on the upper outflows surface and non-homogeneous Neumann boundary conditions equal to $[0, -1]$ on the bottom inlet surface.

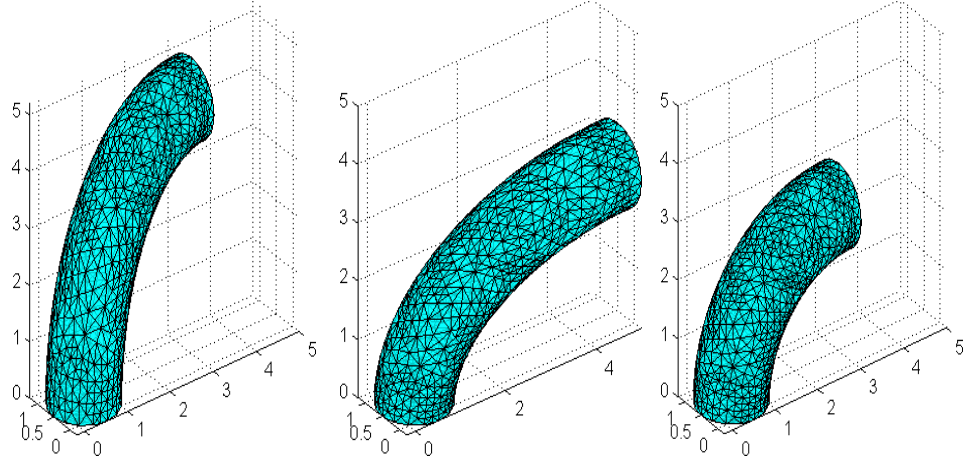


Figure 5.11: Examples of deformed pipes: $\mu_1 = 5, \mu_2 = 3$ (left), $\mu_1 = 3, \mu_2 = 5$ (center), $\mu_1 = 3, \mu_2 = 3$ (right).

Figure 5.12 shows the distribution of the parameter values selected during the greedy algorithm by choosing the maximum number of basis functions N_{max} equal to 30. We note that the distribution of the selected parameters μ_1 and μ_2 has a clustered concentration close to extreme values of the ranges.

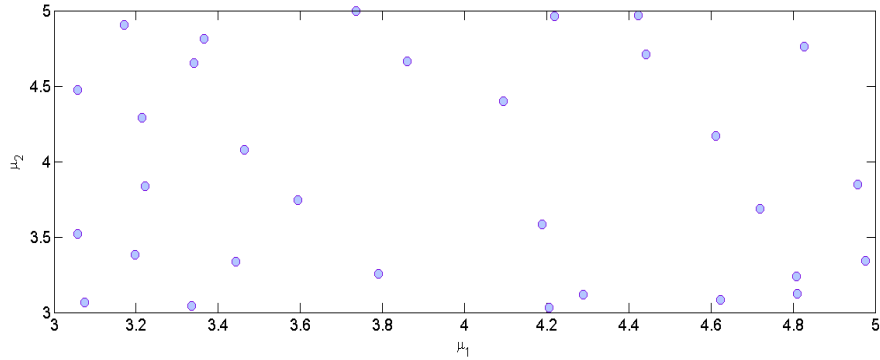


Figure 5.12: Distribution of the selected parameter values during the greedy algorithm.

Figure 5.13 shows the error (the sum of H^1 velocity relative error and L^2 pressure relative error) between the “truth” FE solution and the RB approximation obtained by solving the Stokes problem for a set of 100 samples and by varying the number of reduced basis functions used to approximate the solution.

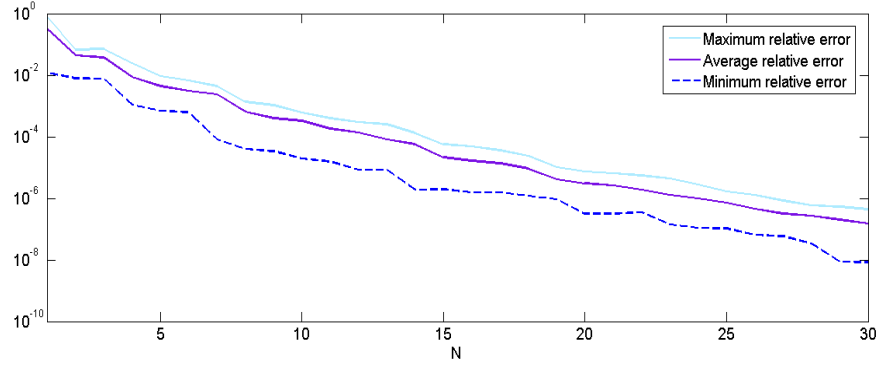


Figure 5.13: Minimum, maximum and average errors between the FE solution and the RB approximation for $N = 1, \dots, 32$, by considering a test set of 100 samples.

Figures 5.14 and 5.15 show two representative RB solutions of the Stokes problem in correspondence to two parameter configurations. As before, we report the velocity field and the pressure obtained by using $N=20$ basis functions together with the corresponding error with respect to the FE solution.

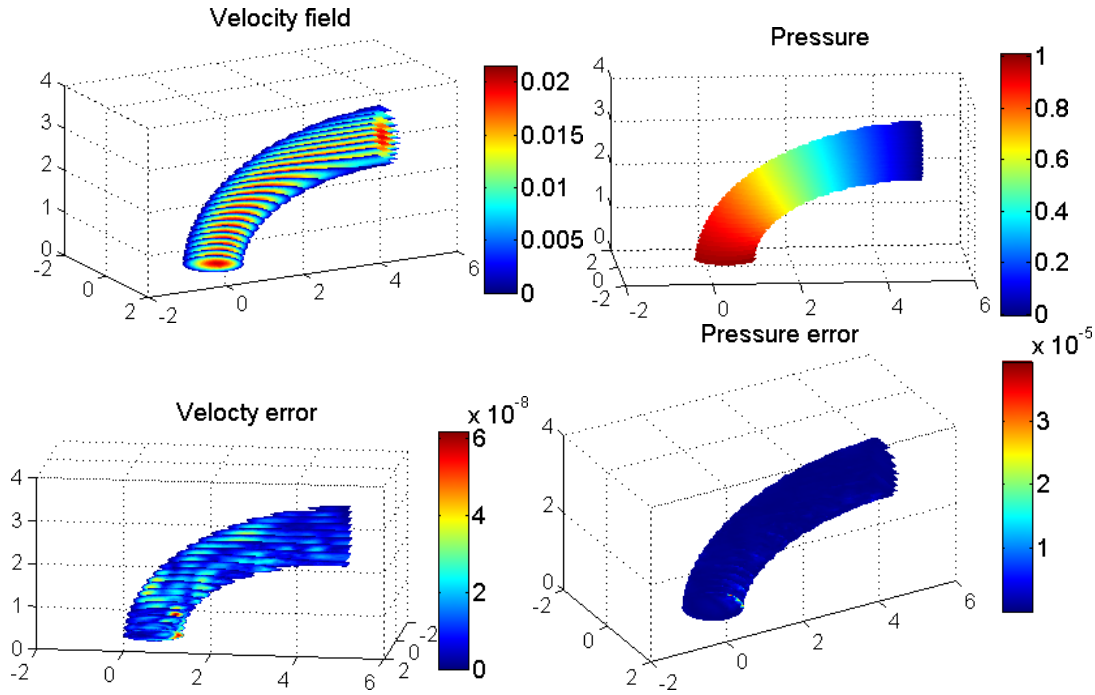


Figure 5.14: Representative RB solution (velocity field and pressure) obtained by using $N=20$ basis functions ($\mu_1 = 6, \mu_2 = -0.48$) and the corresponding error with respect to the FE solution.

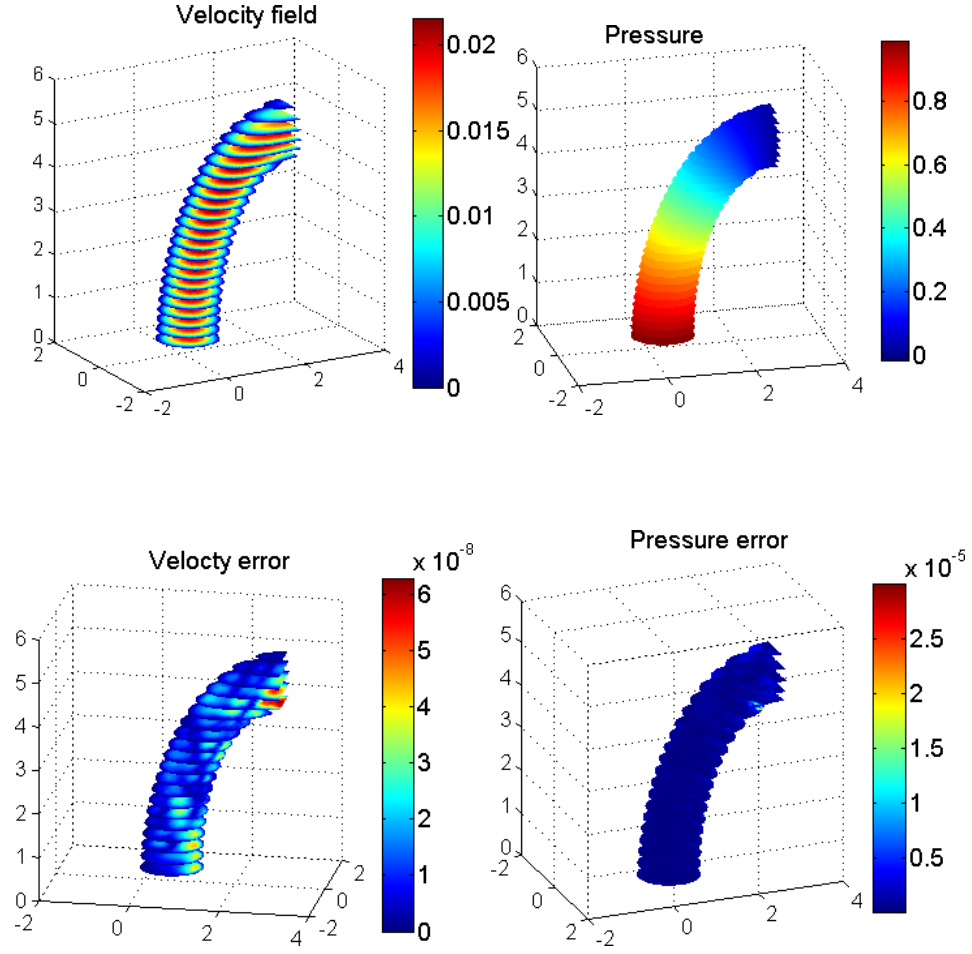


Figure 5.15: Representative RB solution (velocity field and pressure) obtained by using $N=20$ basis functions ($\mu_1 = 11, \mu_2 = 0.5$) and the corresponding error with respect to the FE solution.

The dimension of the linear system that correspond to the FE method is $\mathcal{N} = 91520$ while, by using $N=20$ basis functions, for finding the approximate RB solution we need to solve online a linear system of dimension $3*N=60$ (number of velocity, supremizer and pressure snapshots). We report in Table 5.2 some figures about the computational complexity of the RB method with respect to the FE method.

\mathcal{N}	cpu time of online FEM (s)	$3N$	cpu time of online RBM	time-ratio
91520	218	60	0.023	0.010%

Table 5.2: Computational details for the pipe geometry test.

5.2 An application of the RBHM on a 3D domain

In this section we apply the RBHM, presented in Chapter 2, to 3D domain in order to address more realistic configurations for blood flow in the study of stenosed arteries.

Here, Taylor-Hood Finite Element Method has been used to compute the basis functions, \mathbb{P}_2 elements for velocity and supremizer, \mathbb{P}_1 for pressure, respectively [76] and consequently $\mathbb{P}_1(\Gamma_{lm})$ for the Lagrange multipliers space.

The RBHM has been applied to solve the Stokes equations in a computational domain Ω composed by two stenosed blocks Ω_{μ_1} and Ω_{μ_2} (Figure 5.16), by imposing non-homogeneous BCs $\sigma_n^{in} = [0, 5]^T$ in the inlet surface ($x_1 = 10$), non-homogeneous BCs $\sigma_n^{in} = [0, -1]^T$ in the outlet surface ($x_1 = 0$) and homogeneous Dirichlet BC on the remaining boundaries of the domain.

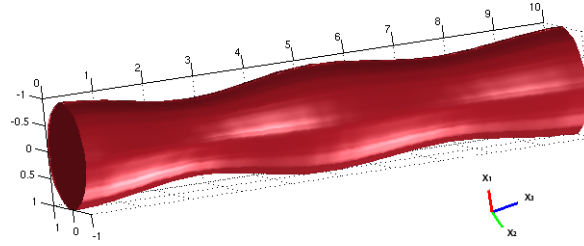


Figure 5.16: Computational domain $\mu_1 = 7, \mu_2 = 10$.

The geometry of a single stenosis is obtained by the deformation of a reference pipe through a parameter that represents the contraction in the middle of the pipe.

The deformed domain Ω_μ is mapped from the straight reference pipe $\hat{\Omega}$ of length $L = 5$ and radius $r = 1$ through the following coordinate transformation $T_\mu : \hat{\Omega} \rightarrow \Omega_\mu$ such as $\mathbf{x} = T_\mu(\hat{\mathbf{x}})$ and

$$\begin{aligned} x_1 &= \hat{x}_1 + \frac{\hat{x}_1}{\mu} \left(\cos\left(\frac{2\pi\hat{x}_3}{L}\right) - 1 \right), \\ x_2 &= \hat{x}_2 + \frac{\hat{x}_2}{\mu} \left(\cos\left(\frac{2\pi\hat{x}_3}{L}\right) - 1 \right), \\ x_3 &= \hat{x}_3. \end{aligned}$$

The range of the parameter μ is $[-20, -5] \cup [5, 20]$, Figure 5.17 shows the reference pipe and some representative deformations of the geometry.

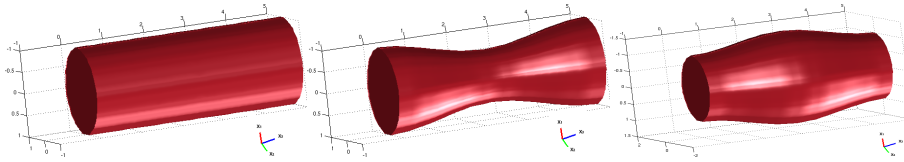


Figure 5.17: Reference pipe and two deformed pipes ($\mu = 5, \mu = -5, \mu = 5$): stenosis and aneurysm configuration.

We consider a parametrized Stokes problem for each subdomain. For the inflow subdomain, we compute the reduced basis imposing zero Dirichlet condition on the wall, Neumann boundary conditions given by imposing $\sigma_n = \sigma \cdot \mathbf{n} = \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n}$ to be $\sigma_n^{in} = [0, 5]^T$ on Γ_{in} and $\sigma_n^{out} = \mathbf{0}$ on the internal interface Γ_{12} . For the outflow subdomain, we compute the reduced basis imposing zero Dirichlet

condition on the wall, Neumann boundary conditions imposing $\sigma_n^{in} = \mathbf{0}$ on the internal interface Γ_{23} and $\sigma_n^{out} = [0, -1]^T$ on the outflow interface Γ_{out} . Figure 5.18 shows the distribution of the parameter values selected by the greedy algorithm, by applying the offline stage of the reduced basis method to the single stenosis block. By taking into account that the range $[-5, 5]$ is not admitted, we can see that the higher concentration of values is in the intervals $[-10, -5]$ and $[5, 10]$ in correspondence to larger deformation of the pipe.

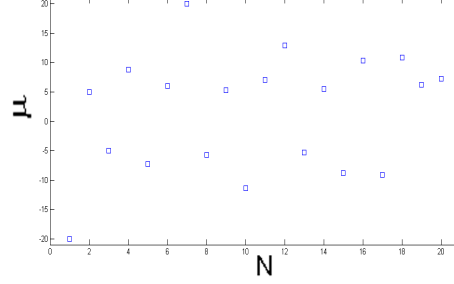


Figure 5.18: Distribution of the selected parameter values by the greedy algorithm used to generate the basis functions in a single block.

Coarse and fine grids have been chosen in order to deal with respectively 155 and 2714 nodes in a single block domain. Figure 5.19 shows a representative flow solution in Ω , found with the reduced basis hybrid method, to be compared with the finite element solution. The same comparison, regarding the pressure solutions, is shown in Figure 5.20.

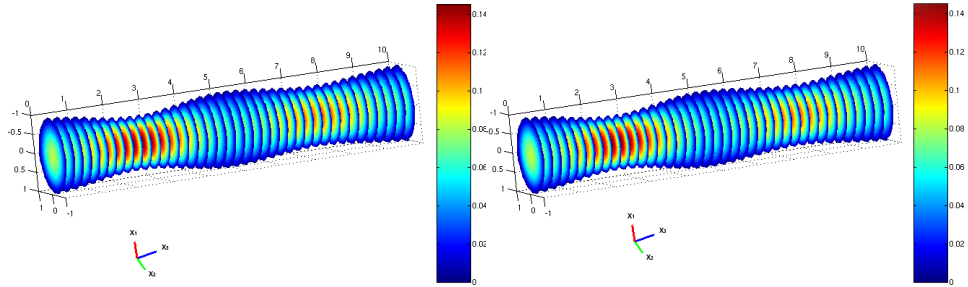


Figure 5.19: Representative solutions of velocity using RBHM (with $N_1 = N_2 = 19$) (left) and using FEM as a global solution (right), $\mu_1 = 7, \mu_2 = 10$.

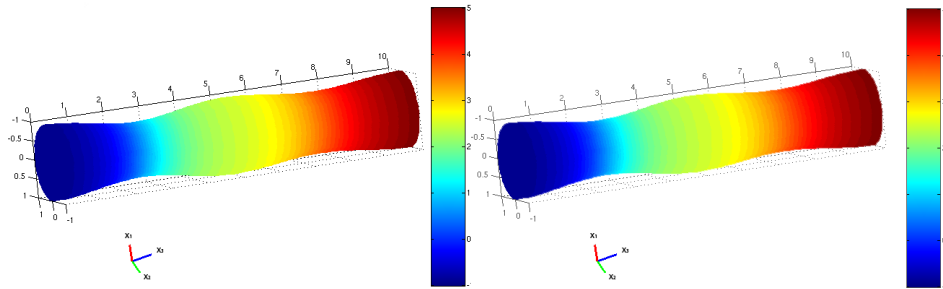


Figure 5.20: Representative solutions pressure using RBHM (with $N_1 = N_2 = 19$) (left) and using FEM as a global solution (right), $\mu_1 = 7, \mu_2 = 10$.

Figure 5.21 shows the reduction of the H^1 relative errors on velocity and L^2 relative errors on pressure, respectively, for the configuration of Figures 5.19 and 5.20, versus the number N of basis functions.

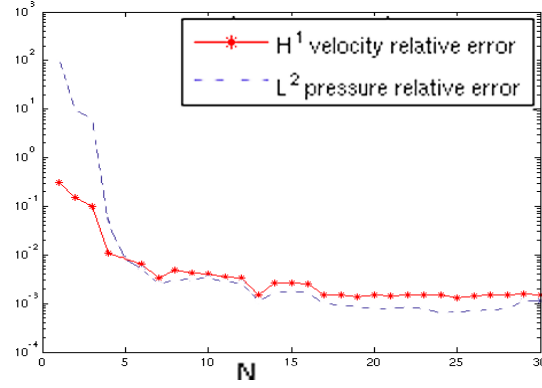


Figure 5.21: H^1 and L^2 relative errors on velocity and pressure.

As in the 2D case, presented in Chapter 2, Sections 2.5.1 and 2.5.2, we show in Figure 5.22 a comparison regarding the velocity profiles on the internal interface obtained by using the RBHM (that includes both velocity and coarse corrections). The profiles of the corresponding fine FEM global solution has been plotted in order to compare the quality of the solution. Figure 5.23 shows the velocity profiles on the internal interface Γ_{12} obtained solving the Stokes problem by using the Lagrange multipliers but not including the coarse correction to the reduced spaces (so without guaranteeing the continuity of stresses). Figure 5.24, shows the velocity profiles on the internal interface Γ_{12} obtained including the coarse correction and not using the Lagrange multipliers correction (not guaranteeing the continuity of velocity). The profiles of the corresponding fine FEM solution computed in the whole network has been plotted as well in order to compare the quality of the solutions. The solutions on Ω for both options are shown in Figures 5.25 and 5.26 in order to compare the pressure as well.

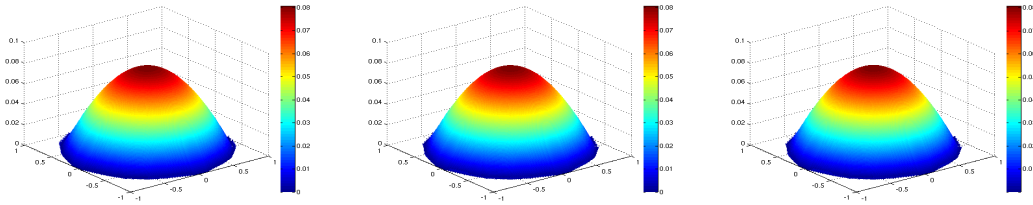


Figure 5.22: Velocity profiles along the internal interface Γ_{12} by solving the RBHM problem plotted from the first block (left), from the second block (center), compared with the velocity profile obtained by using the global FEM solution along the same internal interface (right).

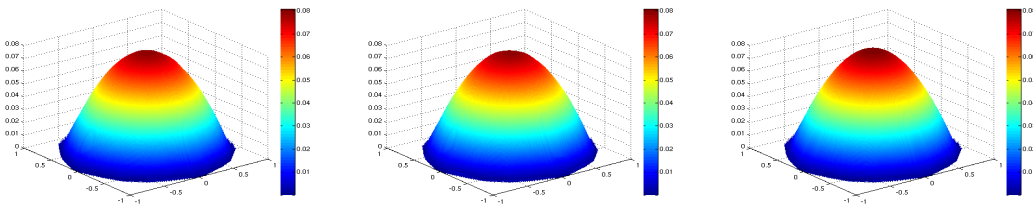


Figure 5.23: Velocity profiles along the internal interface Γ_{12} by using the velocity correction and not using the coarse correction, plotted from the first block (left), from the second block (center), compared with the velocity profile obtained by using the global FEM solution along the same internal interface (right).

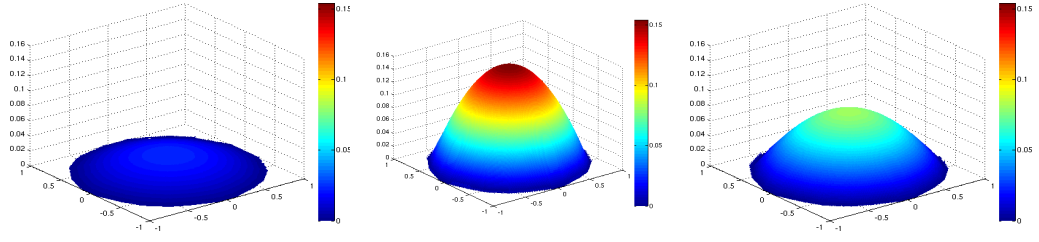


Figure 5.24: Velocity profiles along the internal interface Γ_{12} by not using the velocity correction and including the coarse correction, plotted from the first block (left), from the second block (center), compared with the velocity profile obtained by using the global FEM solution along the same internal interface (right).

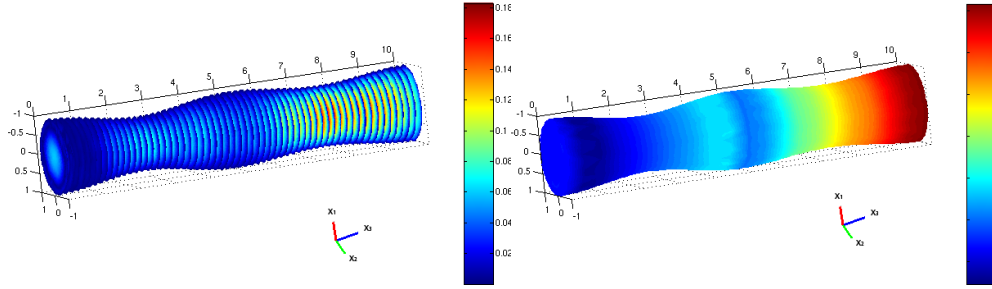


Figure 5.25: Representative solutions of velocity [ms^{-1}] and pressure [Nm^{-2}] using only the velocity correction but not using the coarse correction (with $N_1 = N_2 = 19$), $\mu_1 = 7, \mu_2 = 10$.

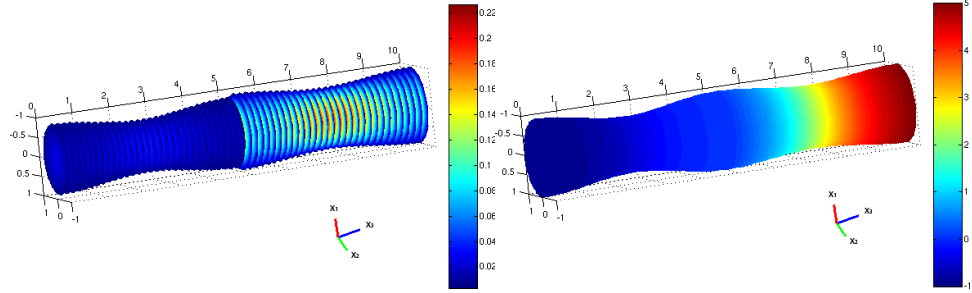


Figure 5.26: Representative solutions of velocity [ms^{-1}] and pressure [Nm^{-2}] using the coarse correction but not the velocity correction (with $N_1 = N_2 = 19$), $\mu_1 = 7, \mu_2 = 10$.

We conclude this section by pointing out that the RBHM is a suitable method to solve steady Stokes equations not only for 2D geometries but also in 3D setting. Further remarks regarding its versatility and the way it behaves compared with other RB-like approaches will be made in Section 5.4.

5.3 RDF method applied to complex 2D networks

In this section, we consider the RDF method introduced in Chapters 3 and 4 for the solution of the steady Stokes problem in 2D non-affinely parametrized geometries. We consider two reference shapes, a stenosed pipe and a bifurcation, through the combination of these geometries we are able to build several complex networks. The numerical results regarding the computation for the reduced basis spaces in the reference shapes will be reported in Sections 5.3.1 and 5.3.2, here we only deal with the offline stage of the method and it is common to all possible configurations of the two considered shapes. Three representative examples of networks are considered in Sections 5.3.3, 5.3.4 and 5.3.5 for solving Stokes problems with the RDF method.

5.3.1 A bifurcated geometry

Figure 5.27 represents the scheme of a 2D bifurcation model parametrized through the BDD TM introduced in Chapter 1, Section 1.3.2. Also in this case the affine decomposition of the problem is recovered thanks to the empirical interpolation method (Chapter 1, Section 1.2.3.1). Despite the fact that this parametrization is able to recover many different deformations, we consider some fixed parameters and that the geometry is deformed only through μ_2 and μ_3 . We consider the reference geometry defined by $\mu_{ref} = [12, 3.8, 0.75, 1, 1, 4, 2, 1]$ and the range of parameters as follows: $\mu_2 \in [2.8, 5]$; $\mu_3 \in [-0.5, 1.6]$.

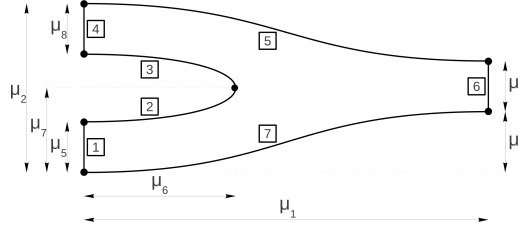


Figure 5.27: Scheme of the 2D bifurcated geometry.

5.3.1.1 The NBC function set

We perform offline the greedy algorithm with respect to the parameters μ_2, μ_3 and by choosing the parametric Fourier profile for the Neumann BC (only the normal component) with $M = 5$, as presented in Chapter 4, Section 4.6.1.2. We note that this bifurcated pipe can be used to compose a complex network and there are three boundaries that can represent the internal interfaces of that network (boundaries 1, 4, 6 numbered as shown in Figure 5.27). For this reason, we consider the local problem with the parametric profiles imposed alternatively on the three boundaries. Due to the fact that we are considering $M=5$ and we have three boundaries involved in parametric profiles, the parameter defining the parametric function λ used for the local Neumann BC is $\mu_\lambda \in \{1, 2, \dots, 15\}$. Figures 5.28 and 5.29 show three different basis functions (snapshots) associated respectively to the values $\mu_\lambda = 1, \mu_\lambda = 6, \mu_\lambda = 11$, these parameters define the constant Fourier functions equal to 1 for the normal component of σ alternatively on the three local problems. The values of the geometrical parameters are $\mu_2 = 4.52$ and $\mu_3 = -0.16$.

Figure 5.30 shows the parameter values μ_λ selected by the greedy algorithm by choosing $N_{max} = 100$. We note that the parameter values distribution is uniformly distributed within the range of possible values, we recall that $\mu_\lambda = 1, \dots, 5$ corresponds to impose the Fourier profile on the boundary 6, $\mu_\lambda = 6, \dots, 10$ corresponds to impose the Fourier profile on the boundary 1, $\mu_\lambda = 11, \dots, 15$ corresponds to impose the Fourier profile on the boundary 4. The same consideration can be done for the distribution of the geometrical parameters μ_2 and μ_3 represented in Figure 5.31.

Figure 5.32 reports the local relative error (the sum of the H^1 velocity relative error and the L^2 pressure relative error) between the RB approximation and the FE solution of the local problem with Fourier Neumann BCs, by varying the number N of the reduced basis functions. In particular, the minimum, maximum and average errors related to a set of 300 samples are reported in the same plot.

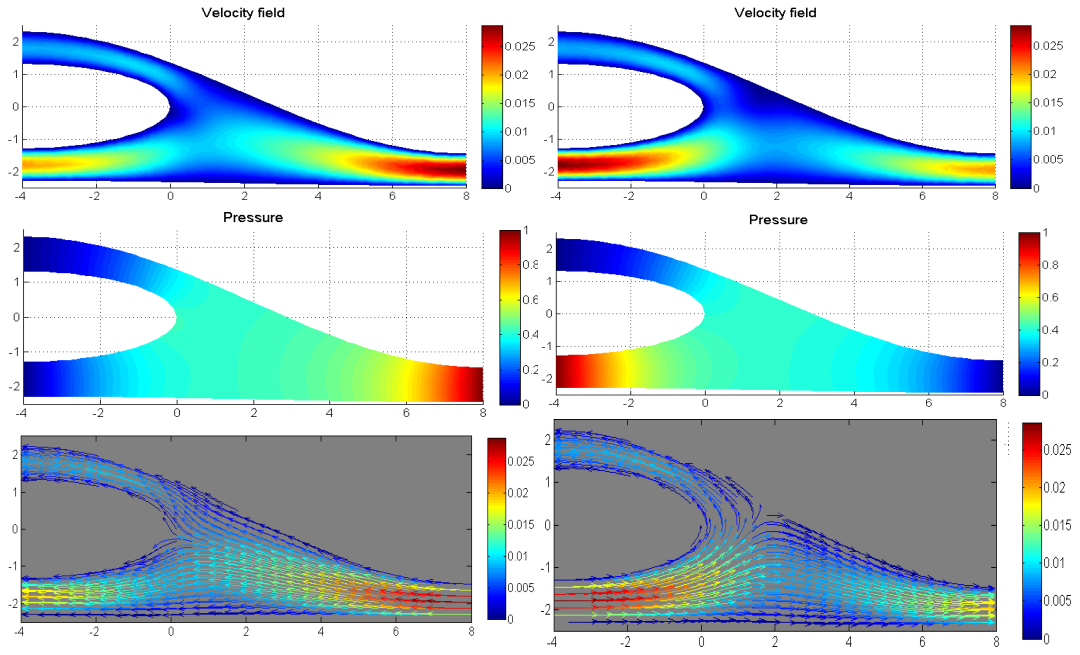


Figure 5.28: Reduced basis snapshots (velocity field on the top, pressure on the center and velocity streamlines on the bottom part) associated to a Fourier profile on the boundary 6 (left) and on the boundary 1 (right), such that the Neumann BC is $\sigma_n = (\cos(0 * \pi * y), 0)$.

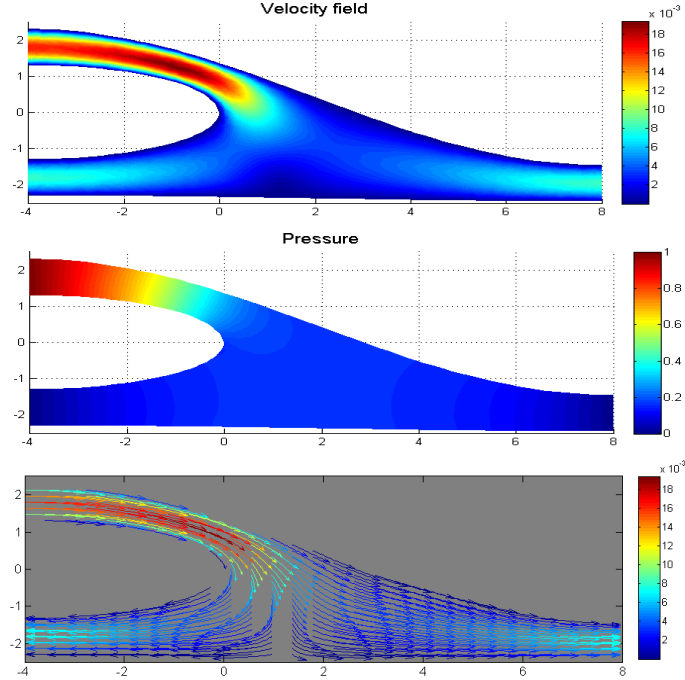


Figure 5.29: Reduced basis snapshots (velocity field on the top, pressure on the center and velocity streamlines on the bottom part) associated to a Fourier profile on the boundary 4, such that the Neumann BC is $\sigma_n = (\cos(0 * \pi * y), 0)$.

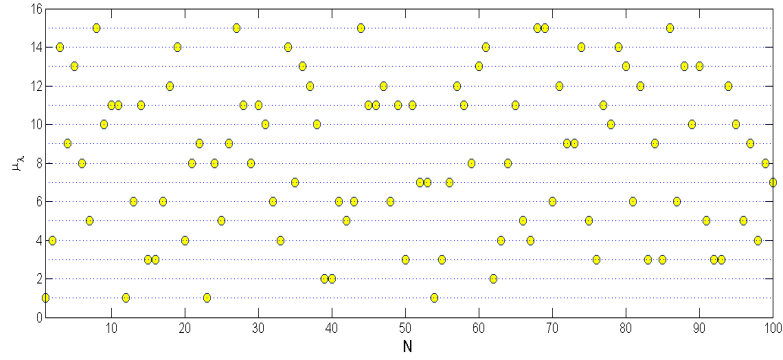


Figure 5.30: Distribution of the parameter values μ_A selected by the greedy algorithm by choosing $N_{max} = 100$.

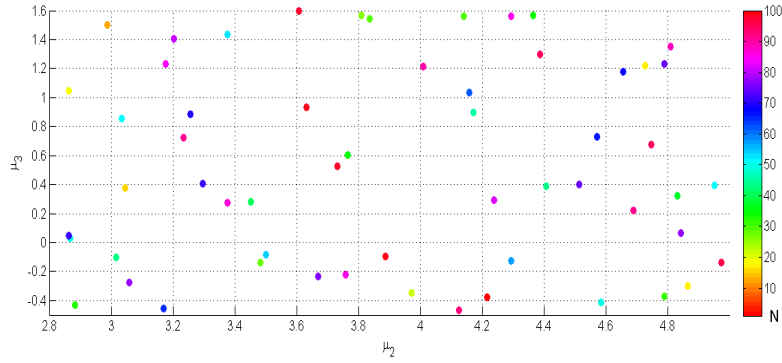


Figure 5.31: Distribution of the parameter values μ_2 and μ_3 selected by the greedy algorithm by choosing $N_{max} = 100$, the colors represent the corresponding N .

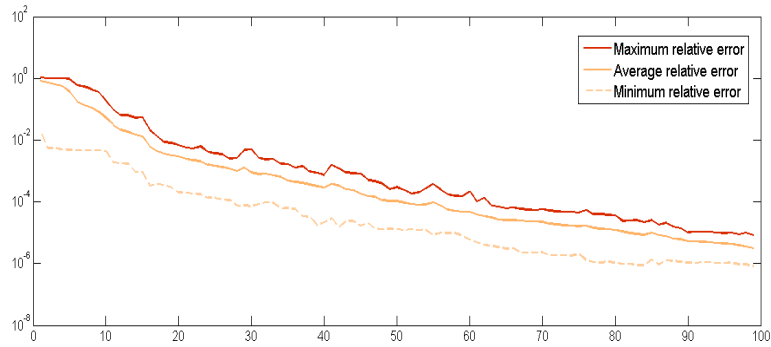


Figure 5.32: Minimum, maximum and average errors between the FE solution and the RB approximation in the case of NBC ($M=5$) by considering a test set of 300 samples.

The basis construction described in this section coincides with that of a classical RB offline stage (with the application of the greedy algorithm) and it can be used for solving the local steady Stokes problem in a single geometry obtained as deformation of the considered bifurcation.

5.3.1.2 The DBC function set

In order to enrich the space with the local basis, we compute another reduced basis space, by choosing the Fourier parametric profile for the Dirichlet BC, as introduced in Chapter 4, Section 4.6.1.1. This time we select $M=3$, so that $\mu_\lambda \in \{1, 2, \dots, 9\}$.

As before, we show in Figures 5.33 and 5.34 the parameter values distribution produced by the greedy algorithm. The considerations regarding the parameter distributions made in the previous section hold for this case too.

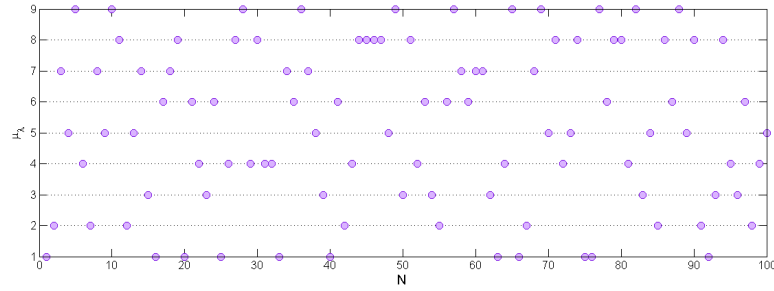


Figure 5.33: Distribution of the parameter values μ_λ selected by the greedy algorithm by choosing $N_{max} = 100$.

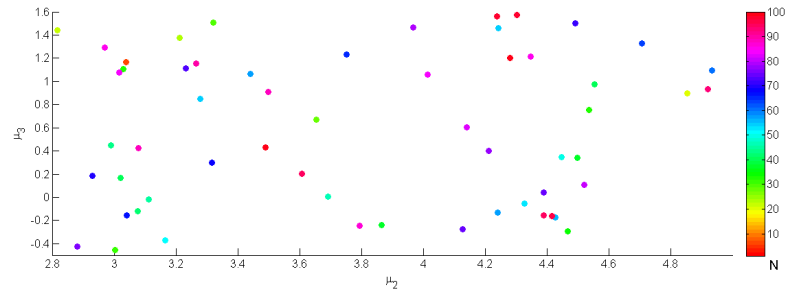


Figure 5.34: Distribution of the parameter values μ_2 and μ_3 selected by the greedy algorithm by choosing $N_{max} = 100$, the colors represent the corresponding N .

Figure 5.35 shows the minimum, maximum and average errors between the FE solution and the RB approximation of the local Stokes problem associated to the Fourier profile set for Dirichlet BCs ($M=3$) by considering a test set of 300 samples.

We can note that with respect to the previous NBC set, here we have a smaller range of possible parameter values μ_λ and by using the same number of RB functions we are able to reach a better accuracy. By observing Figure 5.32 and 5.35 we can conclude that in order to have an error of order $\sim 10^{-4}$, in the first case (NBC) we need to consider $N=65$ basis function, in the second one (DBC) $N=45$ is enough.

As in the previous offline stage, the reduced basis space built for defining the DBC set can be used for the evaluation of a classical RB solution in a single geometry defined as deformation of the considered bifurcation.

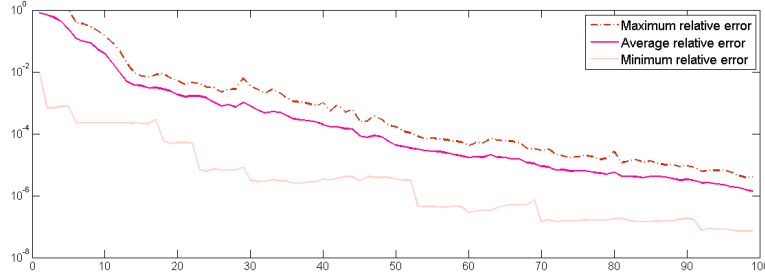


Figure 5.35: Minimum, maximum and average errors between the FE solution and the RB approximation in the case of DBC (M=3) by considering a test set of 300 samples.

5.3.2 Stenosed geometry

Here we follow the same description of the previous subsection but dealing with the stenosed geometry. Figure 5.36 shows a geometrical scheme of the parametrized geometry. Also in this case we maintain some fixed parameters and we consider that the geometry is deformed only through μ_5 and μ_8 that represent the occlusion or the dilatation of the pipe. In particular we consider $\mu_{ref} = [4, 1, 1, 2.5, 0, 1, 1.5, 0]$ and $\mu_5 \in [-0.2, 0.5]$; $\mu_8 \in [-0.2, 0.3]$.

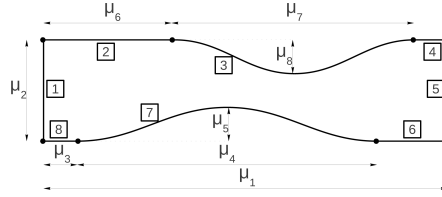


Figure 5.36: Scheme of the 2D stenosed geometry.

5.3.2.1 The NBC function set

We compute independent greedy algorithms for the selection of the local reduced basis space. As before we select the Fourier profiles by fixing $M = 5$ for the NBC option. The stenosed geometry presents two boundaries (1 and 5) that can represent the internal interfaces of the possible assembled network, thus the parameter describing the Fourier profile is $\mu_\lambda \in \{1, 2, \dots, 10\}$.

First of all, we show in Figure 5.37 some representative basis functions, corresponding to the values $\mu_\lambda = 1$ and to the two boundaries.

As in the case of the bifurcation, we show in Figures 5.38 and 5.39 the parameter values distribution selected during the greedy algorithm by choosing the maximum number of basis functions N_{max} equal to 100. We note that the values of the 3 considered parameters are uniformly distributed within the ranges of the possible parameter values.

In Figure 5.40 we show the minimum, maximum and average errors (the sum of the H^1 velocity relative error and L^2 pressure relative error) between the FE solution and the RB approximation in the case of NBC (M=5) by considering a test set of 200 samples.

5.3. RDF method applied to complex 2D networks

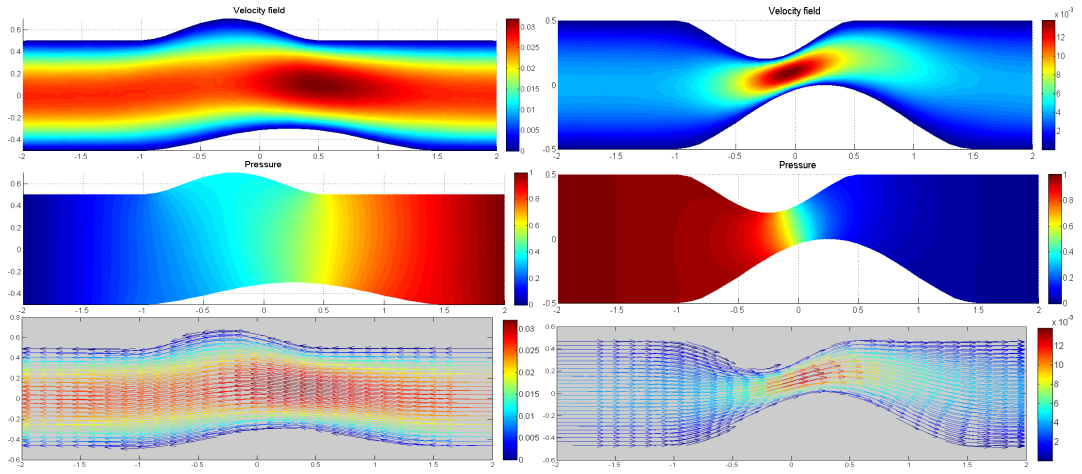


Figure 5.37: Reduced basis function (velocity field on the top, pressure on the center and velocity streamlines on the bottom part) associated to a Fourier profile on the boundary 5 (left) and on the boundary 1 (right), such that the Neumann BC is $\sigma_n = (\cos(0 * \pi * y), 0)$.

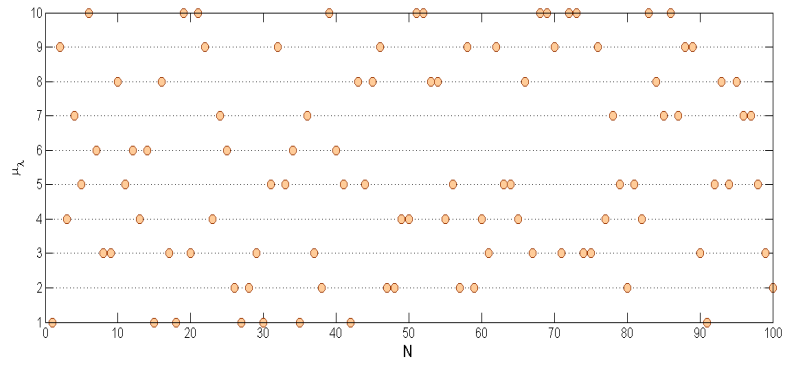


Figure 5.38: Distribution of the parameter values μ_λ selected by the greedy algorithm.

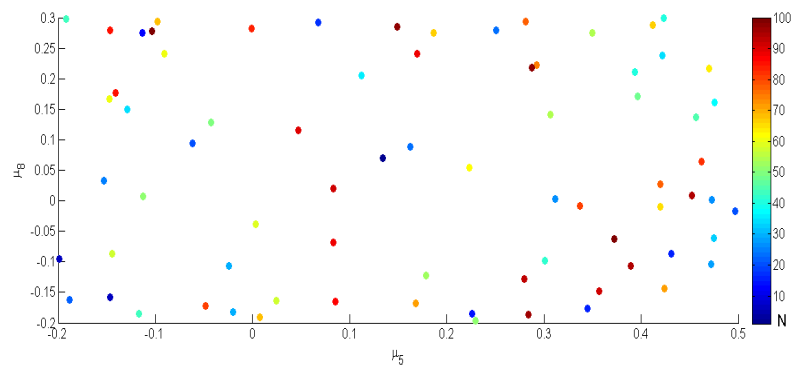


Figure 5.39: Distribution of the parameter values μ_5 and μ_8 selected by the greedy algorithm, the colors represent the corresponding N .

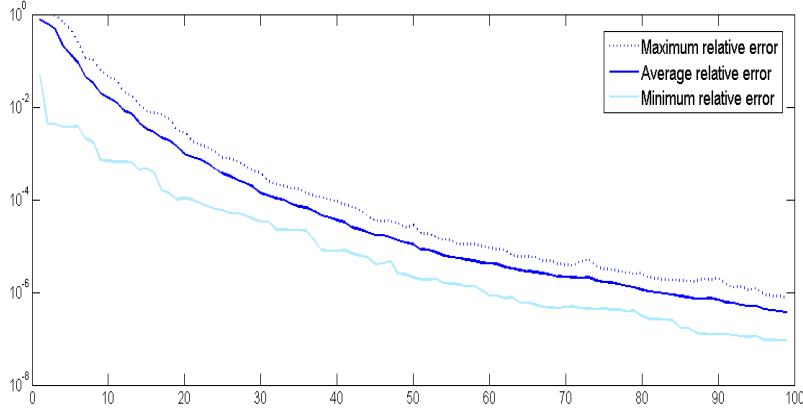


Figure 5.40: Minimum, maximum and average errors between the FE solution and the RB approximation in the case of NBC ($M=5$) by considering a test set of 200 samples.

5.3.2.2 The DBC function set

As in the case of the bifurcation geometry, we consider a second space of basis functions associated to the solutions of the local Stokes problem involving a parametric profile for the Dirichlet BCs, in this case $\mu_\lambda \in \{1, 2, \dots, 6\}$. Figure 5.41 shows some representative snapshots corresponding to the values $\mu_\lambda = 3, 6$.

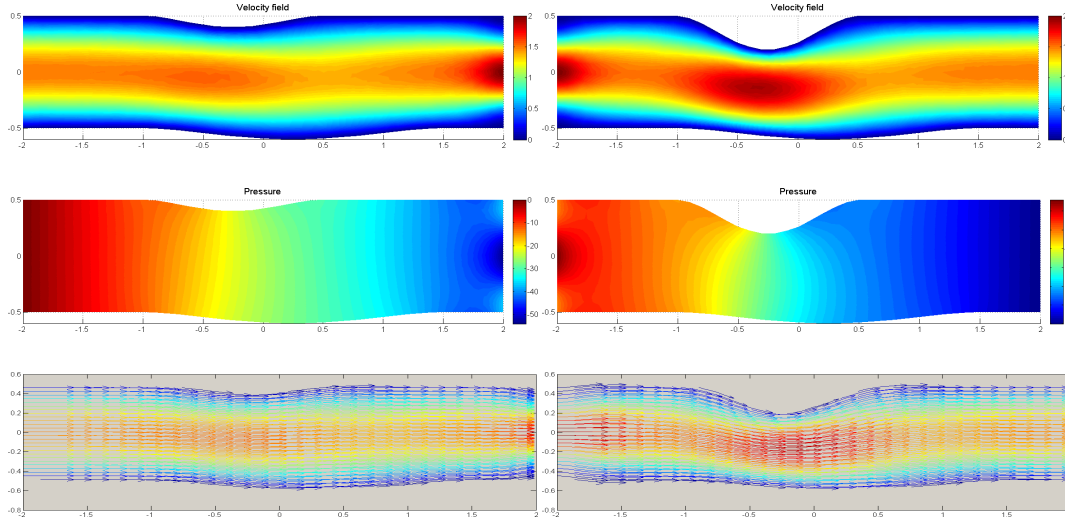
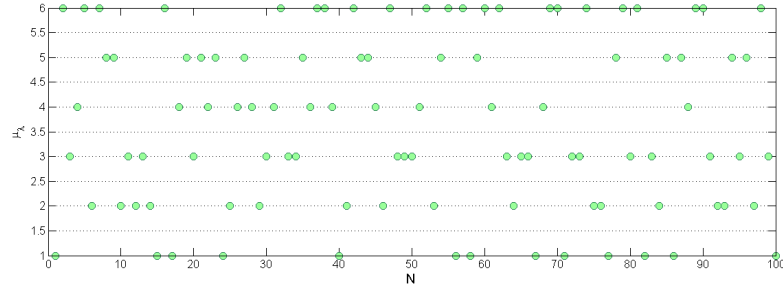
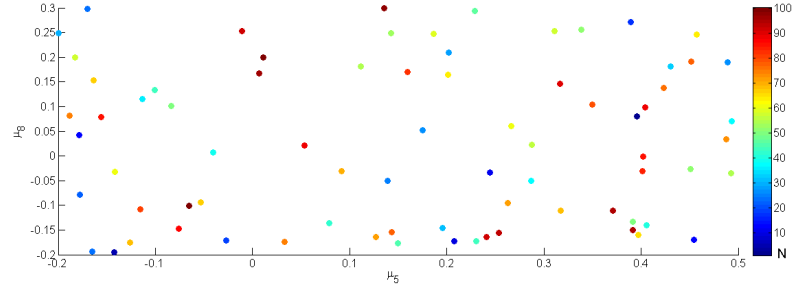
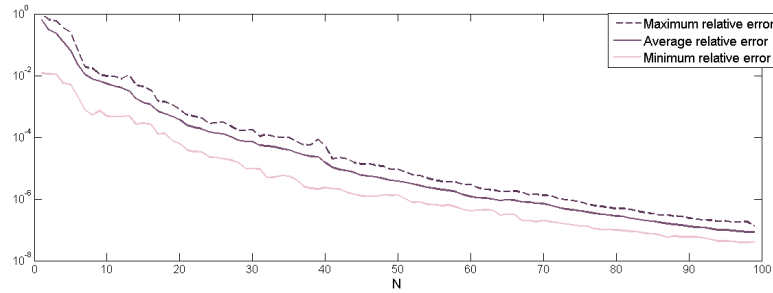


Figure 5.41: Reduced basis function (velocity field on the top, pressure on the center and velocity streamlines on the bottom part) associated to a Fourier profile on the boundary 5 (left) and on the boundary 1 (right), such that the Neumann BC is $\sigma_n = (\cos(2 * \pi * y), 0)$.

Figures 5.42 and 5.43 contain the distribution of the parameter values selected by the greedy algorithm, also in this case we have a uniform distribution within the range of possible values.


 Figure 5.42: Distribution of the parameter values μ_λ selected by the greedy algorithm.

 Figure 5.43: Distribution of the parameter values μ_5 and μ_8 selected by the greedy algorithm, the colors represent the corresponding N .

Finally Figure 5.44 shows the decrease of the error between the RDF and the FE solutions of the local Stokes problem with Dirichlet parametric BCs. As usual, we show the minimum, the maximum and the average error (sum of the H^1 velocity relative error and the L^2 pressure relative error) obtained by considering a set of 200 samples.


 Figure 5.44: Minimum, maximum and average errors between the FE solution and the RB approximation in the case of DBC ($M=3$) by considering a test set of 200 samples.

Here we have a further example of the way the range of parameter values affects the RB solution of the problem. We can see from Figures 5.40 and 5.44 that if we want to reach a relative error of about $\sim 10^{-4}$ in the case of NBC option ($M=5$) we need $N=45$ basis functions, while in the case of the DBC option ($M=3$) only $N=30$.

5.3.3 A curved channel

We consider a composed network representing a channel with curved upper and bottom walls and composed by four stenosed geometries described in the previous section. The network is parametrized through eight parameters, two for each stenosed block, $\boldsymbol{\mu} = (\boldsymbol{\mu}^1; \boldsymbol{\mu}^2; \boldsymbol{\mu}^3; \boldsymbol{\mu}^4)$, $\boldsymbol{\mu}^i = (\mu_1^i, \mu_2^i)$, $i = 1, \dots, 4$.

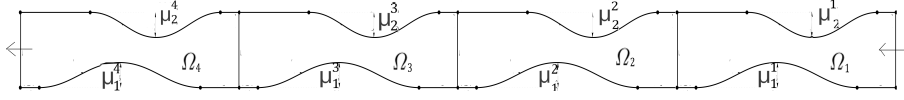


Figure 5.45: Geometrical scheme for the curved channel.

We want to solve the Stokes problem by imposing homogeneous Dirichlet BC on both the upper and bottom walls of the domain, homogeneous Neumann BC on the outflow boundary (on the left) and non-homogeneous Neumann BC equal to 1 on the inflow boundary of the channel (on the right).

For this test case we find the solution by using the DBC set of basis functions presented in the Section 5.3.2.2, the NBC set presented in the Section 5.3.2.1 and the spaces composed by both sets. Figure 5.46 shows a representative solution by setting $\boldsymbol{\mu} = (0.5, 0.3; 0.5, -0.2; -0.2, 0.3; 0.3, 0.3)$ and by using $N=40$ basis functions for each block and the DBC set. The velocity streamlines are plotted in Figure 5.47.

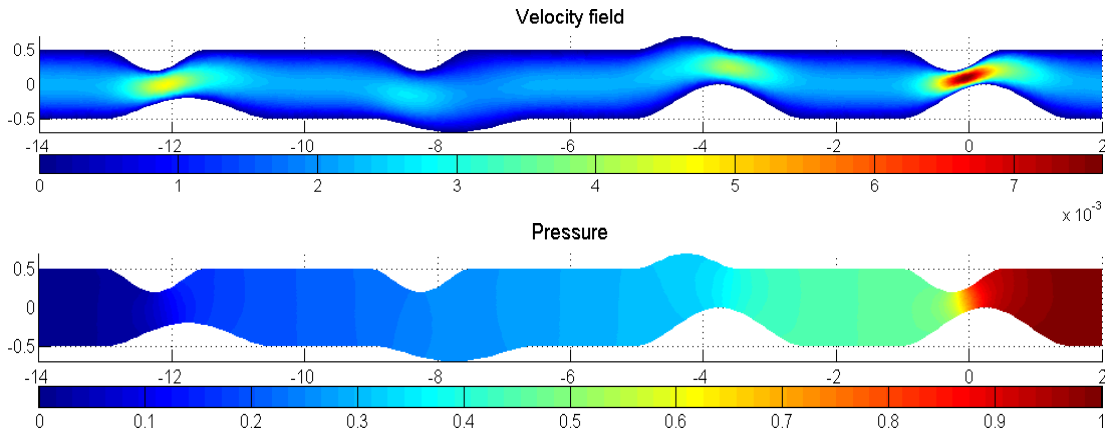


Figure 5.46: The RDF approximation (velocity field and pressure) corresponding to $\boldsymbol{\mu} = (0.5, 0.3; 0.5, -0.2; -0.2, 0.3; 0.3, 0.3)$ and by using $N=40$.

Due to the fact that the underline grid for both RDF and FE solutions is the same, the velocity and pressure absolute errors obtained as difference between the two solutions are plotted in Figure 5.48. We note that the error is bigger in the regions of the domain close to the internal interfaces. This error concentration is due to the fact that every local basis functions composing (through a linear combination) the RDF solution have fixed profiles (3 types of Fourier functions) along the internal interfaces that are different with respect to the FE “truth” solution.

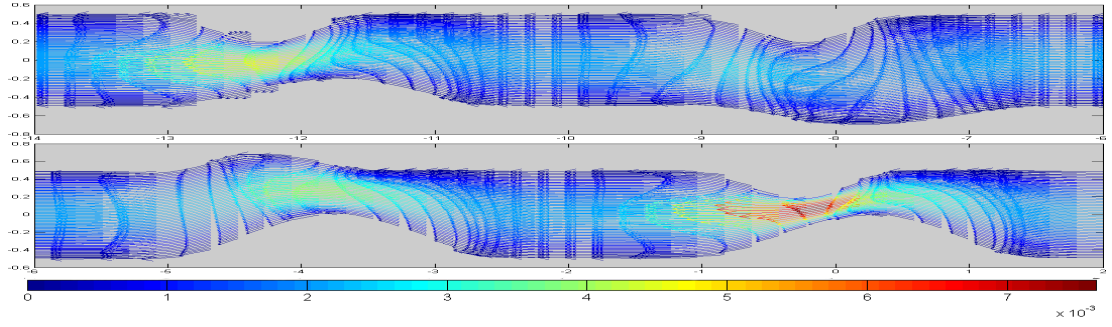


Figure 5.47: Velocity streamlines of a particular RDF approximation corresponding to $\mu = (0.5, 0.3; 0.5, -0.2; -0.2, 0.3; 0.3, 0.3)$ and by using $N=40$. The streamlines on Ω_1 and Ω_2 are in the upper part and on Ω_3 and Ω_4 in the bottom part.

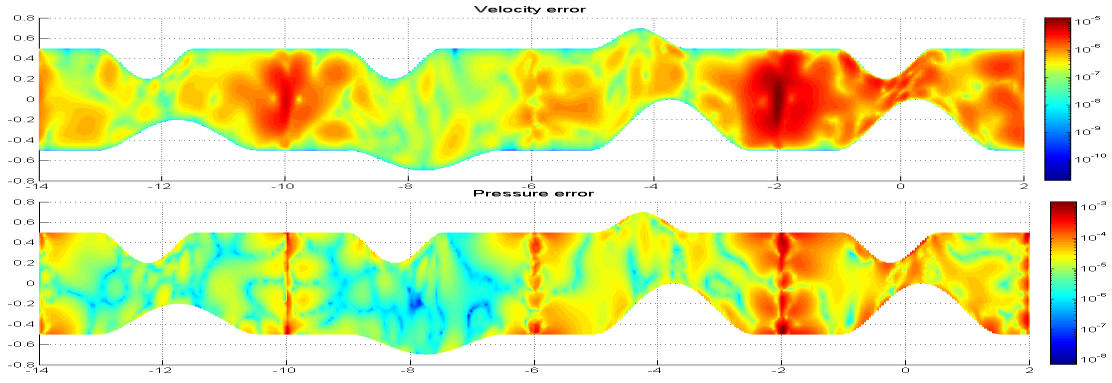


Figure 5.48: Distributed velocity error (top) and pressure error (bottom) of the RDF approximation with respect to the FE solution, by using the DBC set and $M=3$ and $N=40$.

We find now the solution of the same Stokes problem, with the same parameters configuration, by using the set of NBC with $M=5$. We report only the error plots with respect to the FE solution: the plot of the approximate velocity and pressure solutions looks qualitatively the same of the previous case. We show the error plots obtained as difference between the RDF and the FE solutions: in Figure 5.49 the RDF solution is computed by using $N=40$ basis functions and in 5.50 by using $N=70$ basis functions. In this case we are imposing locally a Neumann parametric profile, indeed we observe that by increasing the number of basis we see in particular a reduction of the error along the interfaces regarding the pressure solution. In general, a bigger number of basis functions is needed in this case with respect to the previous one to reach the same accuracy, because here the range of possible values of μ_λ is bigger than in the DBC set case.

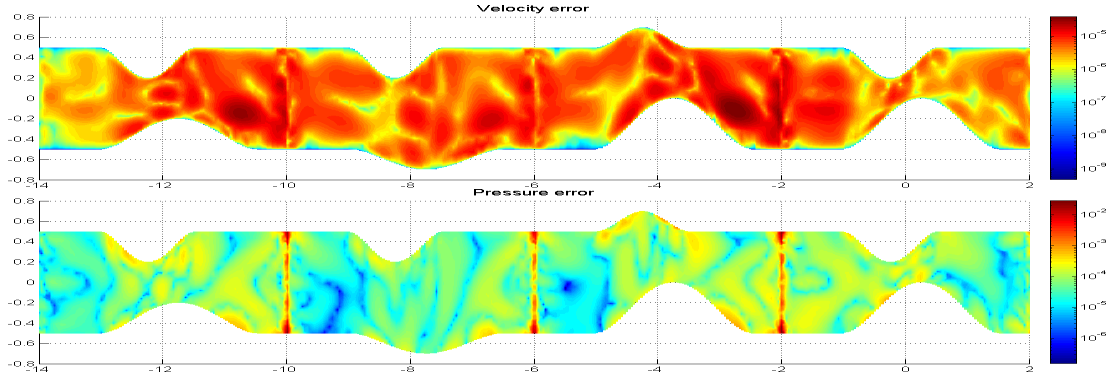


Figure 5.49: Distributed velocity error (top) and pressure error (bottom) of the RDF approximation with respect to the FE solution, by using the NBC set and $M=3$ and $N=40$.

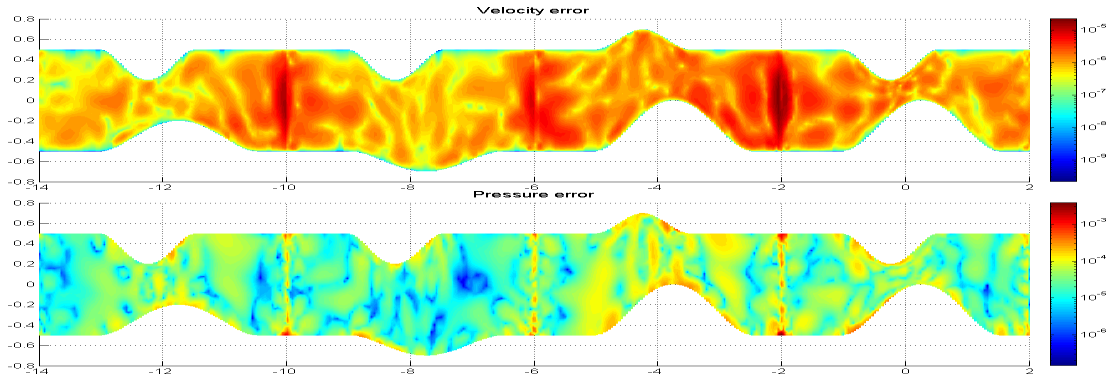


Figure 5.50: Distributed velocity error (top) and pressure error (bottom) of the RDF approximation with respect to the FE solution, by using the NBC set and $M=3$ and $N=70$.

In order to provide a global idea of the error reduction by varying the number of basis functions for both cases the DBC and NBC we show in Figures 5.51 and 5.52 the H^1 velocity relative errors and the L^2 pressure relative error between the global RDF approximation and the FE solution.

In order to visualize the error in each region of the domain we have split it into four curves, one for each subdomain of the network. In the following plots we observe that, after a certain point, by increasing the number of basis functions, the error does not decrease. This behavior is due to the fact that the error is dominated by the interpolation error along the interface and increasing the number of local Fourier BCs becomes necessary to reduce the global error.

We consider now the RDF solution of the problem by using the RB spaces built by both the functions defined with the DBC option and those defined with the NBC option. Figure and 5.53 show the errors (regarding velocity and pressure as before). We note that by using the latter set of spaces we are able to recover a better accuracy with respect to both the velocity and pressure approximation.

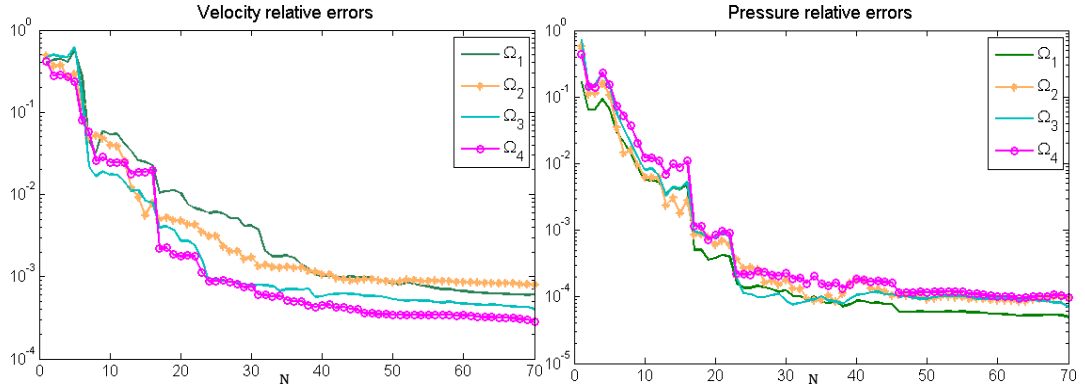


Figure 5.51: Velocity relative error (left) and pressure relative error (right) between the RDF approximation and the FE solution, by using the DBC set and $M=3$ varying the number of local basis functions.

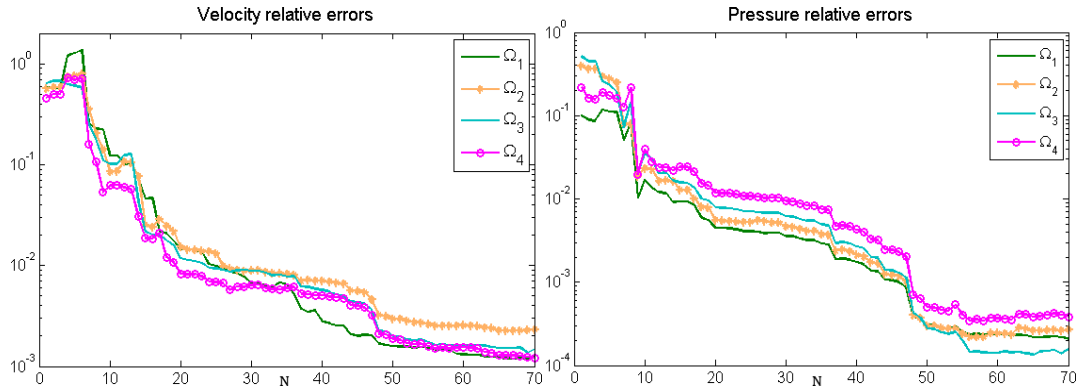


Figure 5.52: Velocity relative error (left) and pressure relative error (right) between the RDF approximation and the FE solution, by using the NBC set and $M=5$ varying the number of local basis functions.

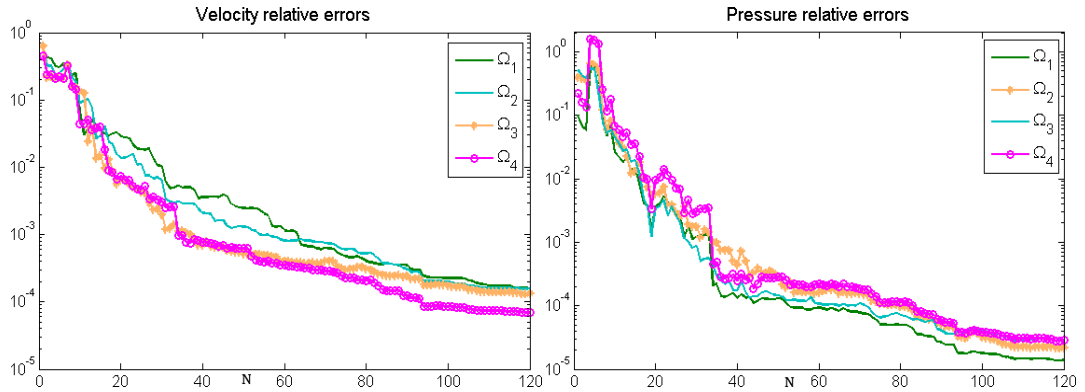


Figure 5.53: Velocity relative error (left) and pressure relative error (right) between the RDF approximation and the FE solution, by using the reduced basis space built by the DBC set with $M=3$ and the NBC set with $M=5$ and by varying the number of local basis functions.

5.3.4 Bifurcated tree

We consider a second composite example of network made up of three bifurcated domains and a stenosed one described in Sections 5.3.1 and 5.3.2 respectively. A scheme that reports the global

network and the different parameters is presented in Figure 5.54. In this case we have $\mu = (\mu^1; \mu^2; \mu^3; \mu^4)$ where $\mu^i = (\mu_1^i, \mu_2^i), i = 1, \dots, 4$.

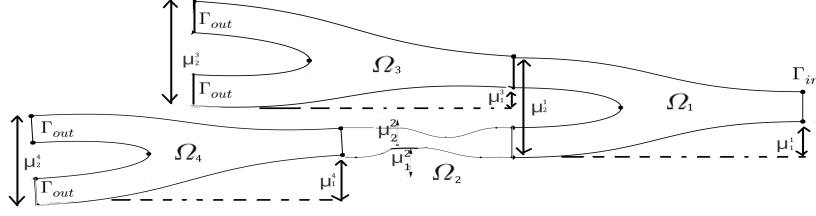


Figure 5.54: Scheme for the bifurcated tree

We want to find the solution for the steady Stokes problem by imposing homogeneous Dirichlet BC on the walls of the network, homogeneous Neumann BC on Γ_{out} and non-homogeneous Neumann BC equal to $[1,0]$ on Γ_{in} . We use the RDF method by considering directly the local reduced spaces composed by both sets DBC (presented in Section 5.3.1.2 for the bifurcation geometry and in Section 5.3.2.2 for the stenosed one) and NBC (presented in Section 5.3.1.1 for the bifurcation geometry and in Section 5.3.2.1 for the stenosed one) with respectively $M=3$ and $M=5$. Figure 5.55 shows the RDF solution corresponding to the value $\mu = (4, -0.5; 0.5, -0.2; 2.8, -0.5; 2.8, 1.6)$ and by using $N=70$ basis functions for each block. The plot of the velocity streamlines is presented in Figure 5.56.

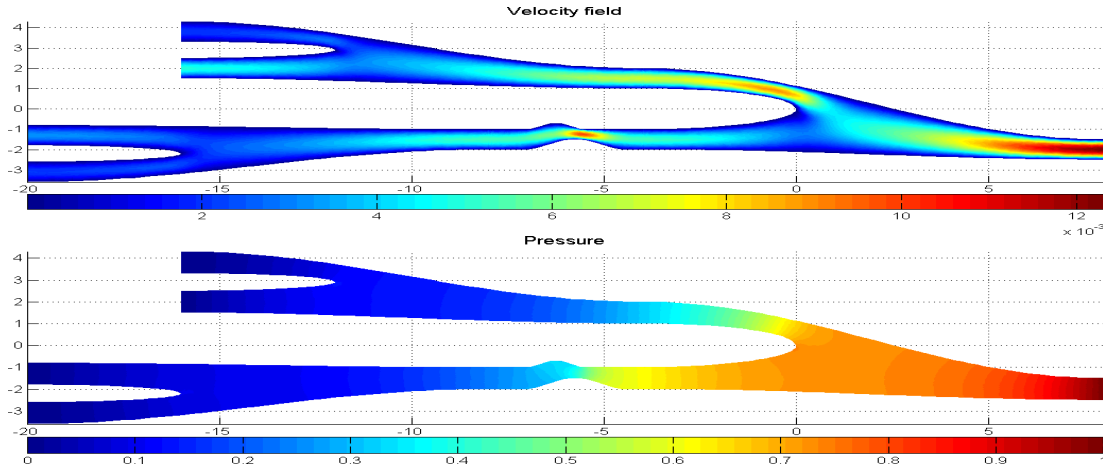


Figure 5.55: Representative RDF approximation (velocity field and pressure) by setting $\mu = (4, -0.5; 0.5, -0.2; 2.8, -0.5; 2.8, 1.6)$ and by using $N=70$.

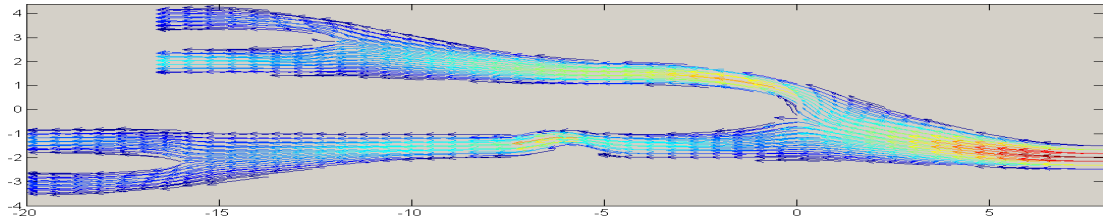


Figure 5.56: Velocity streamlines of a particular RDF approximation by setting $\mu = (4, -0.5; 0.5, -0.2; 2.8, -0.5; 2.8, 1.6)$ and by using $N=70$.

Also in this case the underline grid for both RDF and FE solutions is the same, thus we show in Figure 5.57 the velocity and pressure absolute errors of the RDF solution with respect to the FE solution.

We note that it is bigger in the regions of the domain close to the internal interfaces as already discussed in the previous test case.

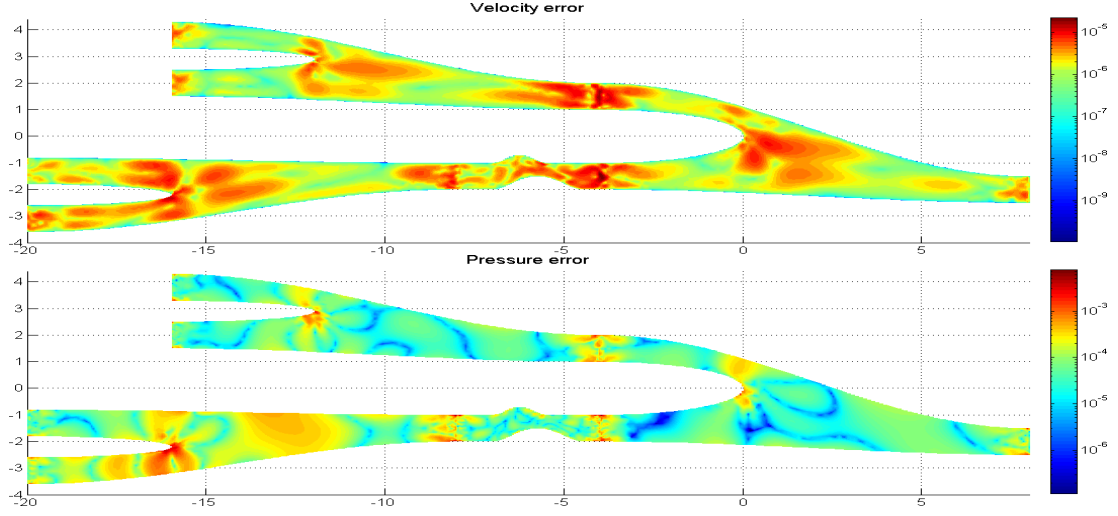


Figure 5.57: Velocity error (top) and pressure error (bottom) of the RDF approximation with respect to the FE solution, by using $N=70$.

We show in Figures 5.58 more detailed results by plotting the relative errors (H^1 for the velocity field and L^2 for the pressure) between the RDF approximation and the FE solution when increasing the number of basis functions.

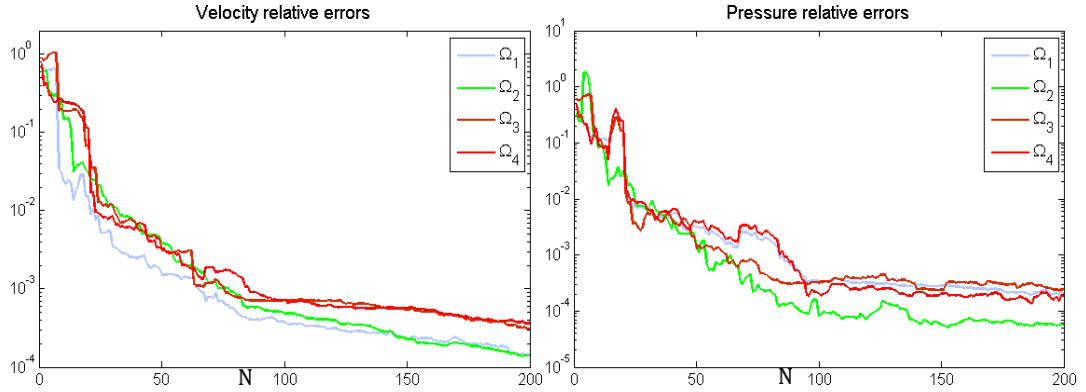


Figure 5.58: Velocity relative error (left) and pressure relative error (right) between the RDF approximation and the FE solution, by using the reduced basis space built by the DNBC set.

5.3.5 Bypass geometry

The last example of network is composed by three bifurcated domains and a stenosed one to represent a simplified network of “cascadic” (sequential) bypasses. A scheme that reports the global network and the total parameter is presented in Figure 5.59. In this case we have $\mu = (\mu^1; \mu^2; \mu^3; \mu^4; \mu^5)$ where $\mu^i = (\mu_1^i, \mu_2^i), i = 1, \dots, 5$.

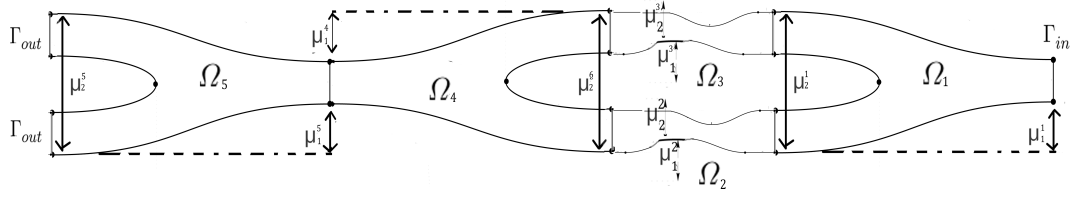


Figure 5.59: Scheme for the composite bypass geometry

We want to find the solution of the steady Stokes problem by imposing homogeneous Dirichlet BC on the walls of the network, homogeneous Neumann BC on Γ_{out} and non-homogeneous Neumann BC equal to $[1, 0]$ on Γ_{in} .

We use the RDF method by considering, as in the previous example, the local reduced spaces composed by both sets DBC and NBC with respectively $M=3$ and $M=5$. Figure 5.60 shows a representative solution corresponding to the parameter value $\mu = (4, -0.5; 0.5, -0.2; -0.2, -0.2; 4, -0.5; 2.8, -0.1)$ and by using $N=70$ basis functions for each block. The plot of the velocity streamlines is presented in Figure 5.61.

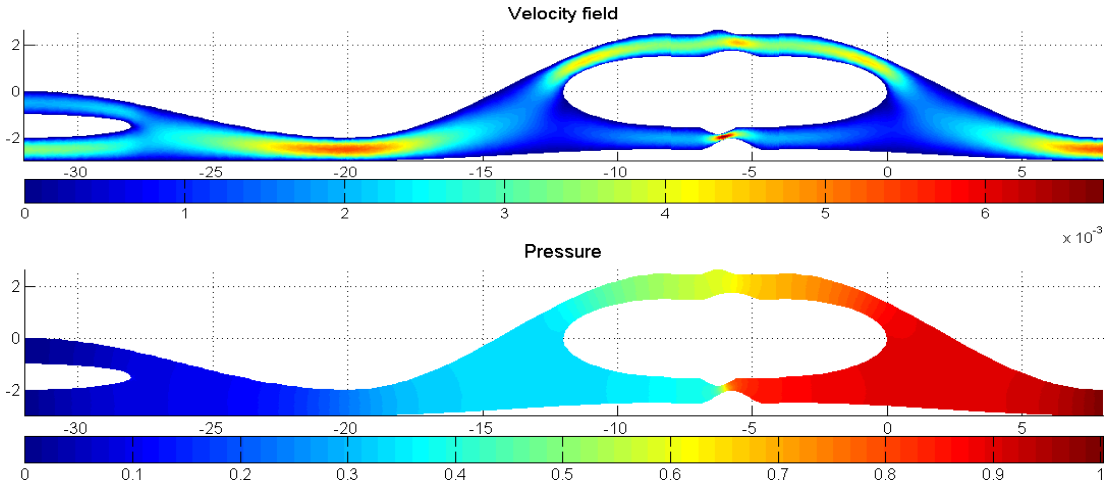


Figure 5.60: Representative RDF approximation (velocity field and pressure) by setting $\mu = (4, -0.5; 0.5, -0.2; -0.2, -0.2; 4, -0.5; 2.8, -0.1)$ and by using $N=70$.

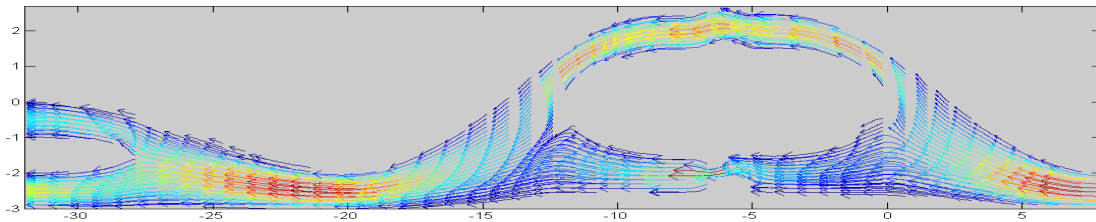


Figure 5.61: Velocity streamlines of a representative RDF approximation by setting $\mu = (4, -0.5; 0.5, -0.2; -0.2, -0.2; 4, -0.5; 2.8, -0.1)$ and by using $N=70$.

As in the two previous examples, the underline grid of the RDF solution coincides with the one of the FE solution and we show in Figure 5.62 the velocity and pressure absolute errors of the RDF solution with respect to the FE solution.

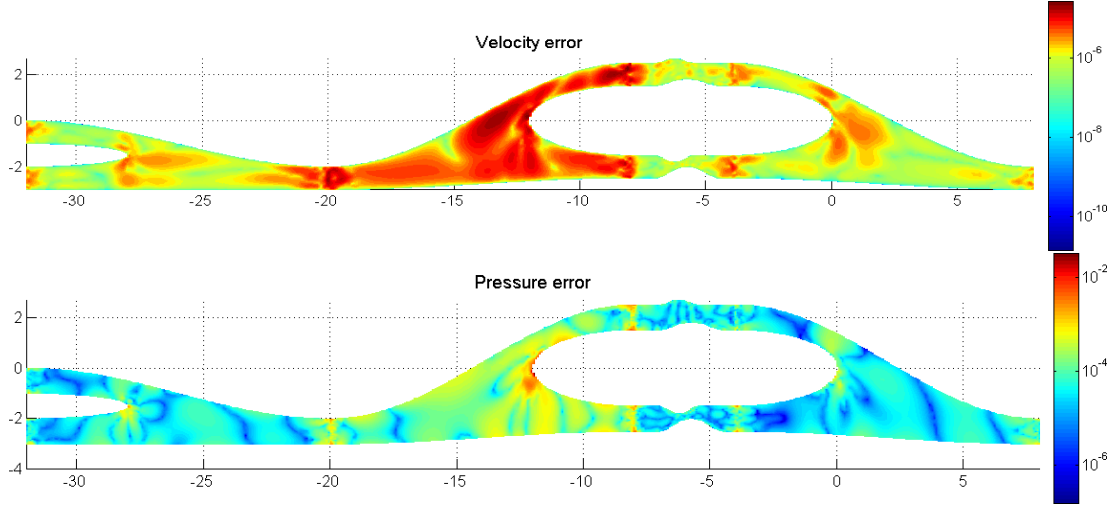


Figure 5.62: Velocity error (top) and pressure error (bottom) of the RDF approximation with respect to the FE solution, by using $N=70$.

Also for this network we show in Figures 5.63 the relative errors (H^1 for the velocity field and L^2 for the pressure) between the RDF approximation and the FE solution by increasing the number of basis functions.

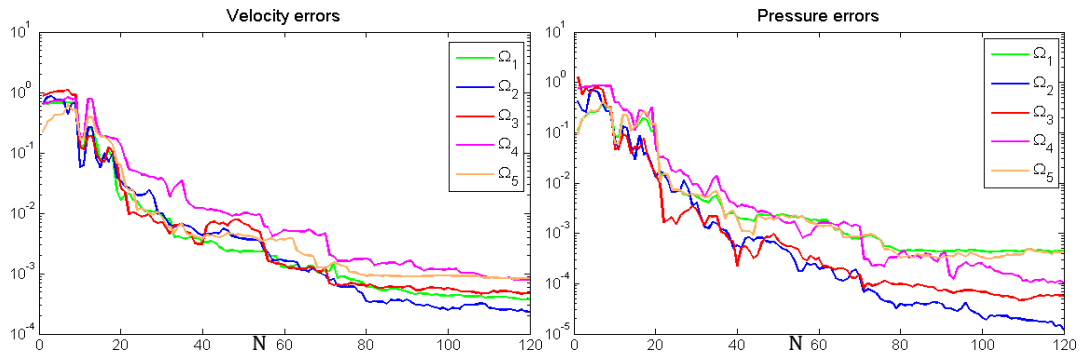


Figure 5.63: Velocity relative error (left) and pressure relative error (right) between the RDF approximation and the FE solution, by using the reduced basis space built by the DNBC set.

5.4 Comparison between the proposed methods

We summarize in this section all the ingredients of the methods introduced in this thesis (RBHM and RDF) together with those already available in literature (RB and RBEM).

As mentioned several times, the development of methods like the RBEM, RBHM and RDF aims at extending the range of applications of the classical RB method and to improve its flexibility principally in terms of more realistic geometrical configurations defining the computational domain.

First of all, we underline a feature that characterizes the classical RB method and that has already been pointed out while illustrating several numerical results: how the number of parameters or their range of possible values affects the computational complexity of the method. In order to better illustrate this issue, we consider the 3D stenosed geometry defined in Section 5.2 and parametrized through a single parameter (denoted with “1 block” configuration), then we consider the 3D pipe composed by two of those stenosed geometries (denoted with “2 blocks” configuration). We consider the same Stokes problem of Section 5.2 for both geometric configurations and we solve that problem with the classical RB method.

Figure 5.64 reports the relative errors (H^1 for the velocity field and L^2 for the pressure) between the RB and the FE solutions and shows how the number of parameters affects the accuracy of the RB method.

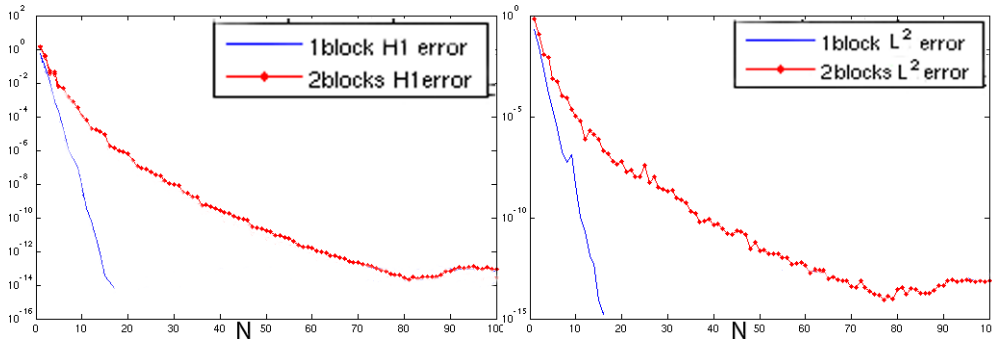


Figure 5.64: Relative H^1 velocity errors (left) and L^2 pressure errors (right) dealing with 1 block and 2 blocks configurations.

We can observe that, in the case of two stenosed domains (2 parameters), we need more than two times the number of basis functions compared with the number we need in the case of a single stenosis (1 parameter) in order to reach the same relative error. In the single domain case, in order to reach an accuracy of order $\sim 10^{-7}$ we need just $N = 9$, in the two stenosed domain $N = 22$, while for reaching a tolerance of order $\sim 10^{-11}$, respectively, $N = 12$ and $N = 45$.

This simple example helps us to explain that a complex geometry defined by a large set of parameters may raise some difficulties in the application of the classical RB method.

In the following, we make further considerations on the way the different methods compare, in particular we highlight the different reduced spaces that characterize the different methods for the solution of a steady Stokes problem in a domain made up by 2 subdomains Ω_1 and Ω_2 with one single internal interface Γ_{12} . We omit here the computations of the supremizer functions that represent a necessary ingredient for all the considered methods.

The RB method

As presented in Chapter 1, the RB method consists in finding the approximated solution of the Stokes problem as Galerkin projection on the velocity and pressure spaces built by proper velocity and pressure

snapshots defined in the global domain. We recall that by using this reduced spaces the RB method is able to deal only with the same computational domain and its geometrical deformations. A very general scheme is presented in Figure 5.65

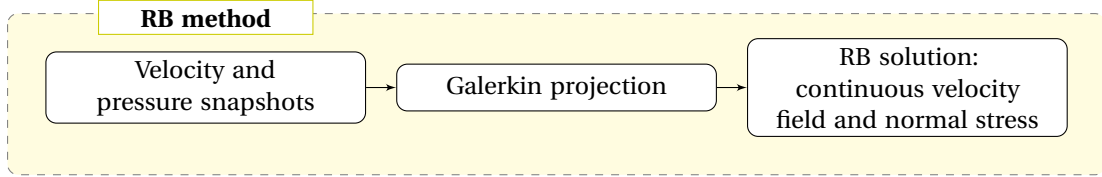


Figure 5.65: General scheme representing the RB method.

The RBEM

The RBEM considers local reduced basis spaces built by proper local snapshots associated to the solutions of proper restricted Stokes problems. Several options have been used in literature for the computation of these bases. One possibility is represented by the definition of local Stokes problem in a domain composed by one subdomain (block) and a small part of the contiguous subdomains, along the interfaces of the latter proper BCs are imposed. In this way, the restrictions of these solutions (velocities and stresses) in each block have different interface profiles not bound to particular BCs [52]. The final RBEM solution is found through a Galerkin projection on the velocity and pressure spaces and the application of Lagrangian multipliers associated to the jump of the velocity functions. This procedure, as shown in the scheme of Figure 5.66 ensures only the continuity of the velocity of the RBEM solution and not the continuity of the normal stress.

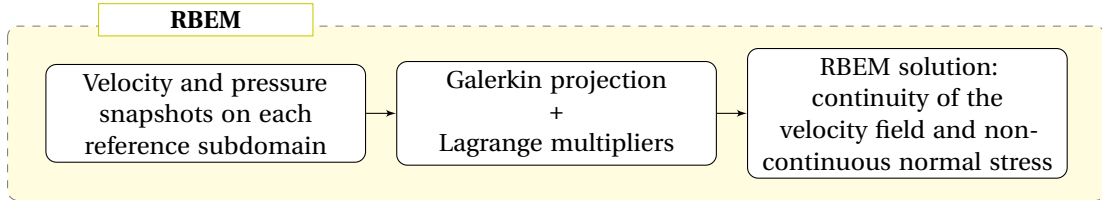


Figure 5.66: General scheme representing the RBEM method.

The RBHM

The RBHM presented in Chapter 2, is built upon the idea of the RBEM and introduces an alternative way for the computation of the local snapshots in order to deal with solutions that, due to their construction, if coupled have continuous stresses along the internal interfaces. The idea is to consider for each reference subdomain three different local Stokes problems one for each possible position of the corresponding internal interface in the whole computational domain (inflow, center or outflow). In the considered 2 subdomains examples, it is sufficient to consider two local Stokes problems: one by considering $\hat{\Gamma}_{12}$ as inflow and one by considering $\hat{\Gamma}_{12}$ as outflow.

The solution of the local Stokes problems with zero-stress BCs imposed to the boundary that correspond to the internal interface are considered as snapshots of the RBHM.

Due to the fact that all these snapshots have zero-stress on $\hat{\Gamma}_{12}$, in order to be able to recover the normal stress of the final solution we need to enrich the spaces of the snapshots. This is done by adding to the local spaces a global coarse solution of the original problem that is then restricted to

the subdomains and added to the local spaces. Finally the RBHM solution is provided, as well as the RBEM, through a Galerkin projection on the reduced velocity and pressure spaces and the use of the Lagrangian multipliers associated to the jump of the velocity functions, see the scheme in Figure 5.67,.

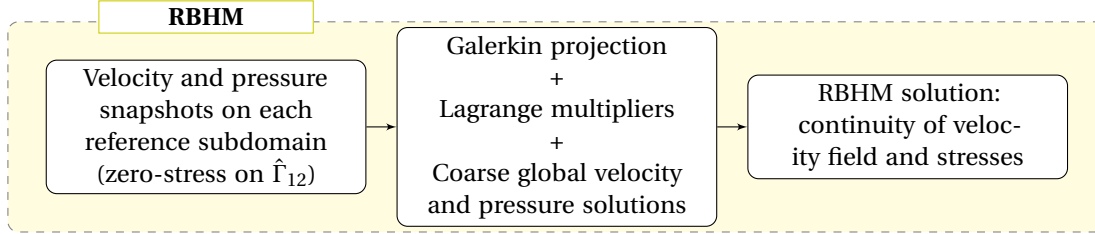


Figure 5.67: General scheme representing the RBHM method.

The RDF method

The RDF method, introduced in Chapters 3 and 4, has been developed in order to improve the performances of the RBHM, by avoiding the computation of a global coarse solution. On one hand, the RDF method permits to increase the versatility of the local snapshots, but on the other hand it requires the introduction of further parameters to the local Stokes problem, by increasing, as pointed out before, the complexity of the local problems and by enlarging the number of local snapshots necessary to solve the problem. Moreover we enrich this local space with the finite element bases defined on the internal interfaces. The RDF solution of the problem is finally provided by a Galerkin projection on these enriched velocity and pressure spaces.

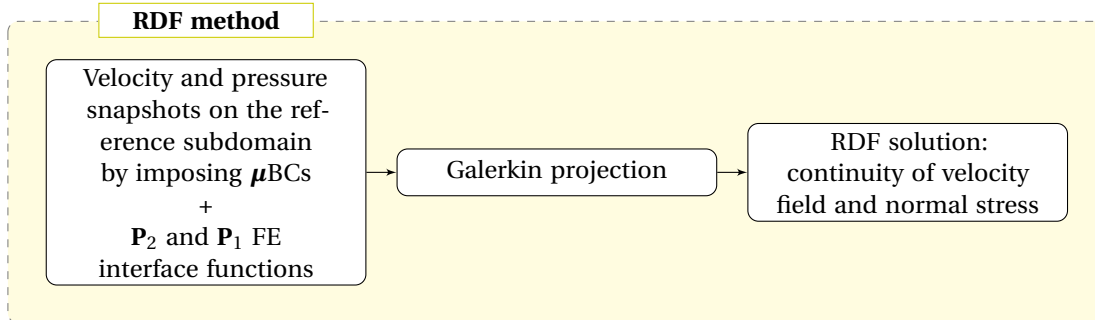


Figure 5.68: General scheme representing the RDF method.

5.4.1 Some considerations

The RBEM, the RBHM and the RDF method allow to deal with every kind of combinations of a certain number of subdomains, for which few offline stages can be computed independently. By using the classical RB method we have to perform the offline stage in the whole domain characterized by more parameters with respect to the single subdomain, but the coupling conditions are automatically satisfied.

In general, the RBEM, the RBHM and the RDF method represent more flexible approaches for the solution of μ PDEs, nevertheless we have to take into account the further costs and effort of the coupling conditions needed to ensure the continuity of the solutions. Each one of the mentioned methodologies presents some positive features and some critical issues.

5.4. Comparison between the proposed methods

Tables 5.3 and 5.4 collect an indication of all the features (either *pros and cons*) of the considered methods in order to allow a better insight of the considered model order reduction techniques.

Method	Number of basis	param. set	cont. cond.	geom. flex.*	cpu time of the offline stage	cpu time of the online stage
RB	$N_{RB} > N_{RBEM}$	small	yes	no	$T_{RB} > T_{RBEM}$	$T_{RB} < < T_{RBEM}$
	$N_{RB} > N_{RBHM}$	small	yes	no	$T_{RB} > T_{RBHM}$	$T_{RB} < < T_{RBHM}$
	$N_{RB} < N_{RDF}$	small	yes	no	$T_{RB} < T_{RDF}$	$T_{RB} < < T_{RDF}$
	$N_{RB} >> N_{RBEM}$	big	yes	no	$T_{RB} >> T_{RBEM}$	$T_{RB} < T_{RBEM}$
	$N_{RB} >> N_{RBHM}$	big	yes	no	$T_{RB} >> T_{RBHM}$	$T_{RB} < T_{RBHM}$
	$N_{RB} > N_{RDF}$	big	yes	no	$T_{RB} > T_{RDF}$	$T_{RB} < T_{RDF}$
RBEM	$N_{RBEM} < N_{RB}$	small	only vel.	yes	$T_{RBEM} < T_{RB}$	$T_{RBEM} > T_{RB}$
	$N_{RBEM} = N_{RBHM}$	small	only vel.	yes	$T_{RBEM} = T_{RBHM}$	$T_{RBEM} < T_{RBHM}$
	$N_{RBEM} < N_{RDF}$	small	only vel.	yes	$T_{RBEM} < T_{RDF}$	$T_{RBEM} < T_{RDF}$
	$N_{RBEM} < < N_{RB}$	big	only vel.	yes	$T_{RBEM} < < T_{RB}$	$T_{RBEM} > T_{RB}$
	$N_{RBEM} = N_{RBHM}$	big	only vel.	yes	$T_{RBEM} = T_{RBHM}$	$T_{RBEM} < T_{RBHM}$
	$N_{RBEM} < < N_{RDF}$	big	only vel.	yes	$T_{RBEM} < T_{RDF}$	$T_{RBEM} < T_{RDF}$

Table 5.3: Overview of the RB and RBEM methods (* geometrical flexibility in terms of the possibility to deal with more than one computational domain).

Method	Number of basis	param. set	cont. cond.	geom. flex.*	cpu time of the offline stage	cpu time of the online stage
RBHM	$N_{RBHM} < < N_{RB}$	big	yes	yes	$T_{RBHM} < < T_{RB}$	$T_{RBHM} > T_{RB}$
	$N_{RBHM} = N_{RBEM}$	big	yes	yes	$T_{RBHM} = T_{RBEM}$	$T_{RBHM} > T_{RBEM}$
	$N_{RBHM} < N_{RDF}$	big	yes	yes	$T_{RBHM} < T_{RDF}$	$T_{RBHM} \geq T_{RDF}$
RDF	$N_{RDF} < N_{RB}$	big	yes	yes	$T_{RDF} < T_{RB}$	$T_{RDF} > T_{RB}$
	$N_{RDF} > N_{RBEM}$	big	yes	yes	$T_{RDF} > T_{RBEM}$	$T_{RDF} > T_{RBEM}$
	$N_{RDF} > N_{RBHM}$	big	yes	yes	$T_{RDF} > T_{RBHM}$	$T_{RDF} \leq T_{RBHM}$

Table 5.4: Overview of the RBHM and RDF methods (* geometrical flexibility in terms of the possibility to deal with more than one computational domain).

As the tables show, when we consider a PDE with a small set of parameters, dealing with simple geometries, the classical RB method is the recommended one, because it allows to deal with small computational time (online and offline).

The main reason to prefer the other methodologies is the possibility to deal with more versatile parametrized geometries by using the same set of data, computed only once offline. The common paradigm between these method is the computation of one offline stage for each simple reference subdomain and then an online stage, to deal with problems on geometries made up by an arbitrarily combination of these subdomains.

Between the other 3 methods (RBEM, RBHM, RDF) the simplest one, in terms of computational complexity and cpu time, is the RBEM, with respect to the RB method it involves the Lagrange multipliers equations, but at the same time we have to consider the critical issue represented by the not ensured stress continuity at the internal interfaces. The RBHM maintains the advantages of the RBEM and, by introducing a coarse online global solution, it can ensure also the stress continuity. This further ingredient involves some additional online computational time, but it introduces an important improvement for the continuity of the global solution (and its quality).

Finally we have considered the RDF method: it represents a different approach that involves the introduction of further parameters into the problem (boundary conditions). This additional computational

complexity may be compared online to the computation of the online coarse solution of RBHM, depending on the problem and on the accuracy of the desired RDF solution. Indeed the introduction of the additional parametric complexity in the RDF method allows to reach a better level of accuracy and, as seen in Chapter 3 and 4, we can decrease the error approximation by slightly increasing this further complexity. We can conclude that at the same level of accuracy the RDF method and the RBHM can be considered similar in terms of cpu time. Moreover by the RDF method we can improve considerably the accuracy of the solution. Table 5.5 reports some further details about the comparison of the two new proposed methodologies.

Features	RBHM	RDF
continuity conditions	coarse solution, Lagrange multipliers	μ BCs, further FE bases
error approximation	$\epsilon_{RBHM} \approx 10^{-3}$	$\epsilon_{RBHM} \leq 10^{-3}$
cpu time if $\epsilon_{RDF} \approx \epsilon_{RBHM}$	offline $T_{RBHM} < T_{RDF}$	online $T_{RBHM} > T_{RDF}$
cpu time if $\epsilon_{RDF} < \epsilon_{RBHM}$	offline $T_{RBHM} \ll T_{RDF}$	online $T_{RBHM} < T_{RDF}$

Table 5.5: Further details of the RBHM and RDF methods.

5.5 Concluding remarks

In this chapter, we have collected numerical results concerning some simplified models of possible cardiovascular geometries and networks.

The classical RB method is very effective for solving parametrized viscous flows equations even in the three dimensional setting and with relevant geometrical deformations. The 3D transfinite maps, introduced in Chapter 1, represent a viable approach when dealing with more complex and quite realistic configurations and can be usefully introduced in the RB framework after a pre-processing operated by the empirical interpolation method.

By dealing with a single geometry, the greedy algorithm is very effective for finding an efficient reduced basis space able to give rich enough data for recovering the RB solution of the problem with a good level of accuracy (and recovering the relevant physical features). The computational time gain is very high and permits to solve the steady Stokes problem, for every new parameter value, through a very inexpensive RB online step. In particular, if we deal with a very fine grid (and especially in a 3D domain) involving a large number of FE degrees of freedom, the RB method represents a very helpful strategy to reduce the computational time required to solve the parametrized problem, specially if we are interested to find the solution for several different values of the parameters, as well as several geometrical configurations.

A different perspective regards the RB combined with the DD approaches, motivated by the fact that the geometrical topology of the domain is not available at the offline stage. In general, we are interested in defining a method able to maintain the flexibility of dealing with arbitrary combinations of subdomains and several geometrical deformations of the latter. In Chapter 2 the RBHM was presented together with 2D numerical results, in this chapter that methodology has been successfully extended to the 3D setting.

In Chapter 4 the RDF method was introduced and tested on simple networks, in this chapter we presented some applications of the RDF method to more complex and composite configurations. Among other advantages, the RDF method allows to perform simulations on extended networks by reducing the number of effective parameters. This versatility is ensured thanks to a very small number of offline stages (one for each reference shape). We have considered three relevant networks and illustrated the corresponding numerical results. We point out that in general it is possible to use the

two presented reference shapes (bifurcation and stenosis) in every kind of composed cardiovascular network, by maintaining a suitable level of accuracy and, as investigated in Chapter 4, a considerable reduction of the computational time and improved performances.

Conclusions

In this thesis we have developed some model order reduction techniques based on the reduced basis method and the domain decomposition techniques for the solution of parametrized partial differential equations. The aim of the proposed work is to exploit the advantages of the reduced basis method to deal with domains arbitrarily composed by deformed shapes, defining a wide variety of possible networks.

The reduced order framework allows to decrease drastically the computational time needed to find the numerical solution on a big range of problems. The main motivation is that there is an increasing interest for many applications requiring rapid and reliable numerical simulations in order to provide prompt and reliable mathematical models for the real situations involving complex geometries (like microfluidic networks).

A common paradigm, presented from the beginning of the thesis, and considered in all the remaining parts, is the introduction of a technique to parametrize the computational domain (or subdomains): the transfinite maps. We have seen, through its 2D and 3D descriptions, that these maps represent suitable ways to efficiently parametrize the computational domain. Consequently, it has been proved that these maps can be successfully applied in the general framework of the reduced basis method as well as in its proposed extensions.

The reduced basis technique is a very effective method for finding rapid and reliable solution of μ PDEs in geometries that can assume different deformation of a reference shapes. The two new methodologies developed in this work, and derived from the reduced basis method, permit to solve rapidly and accurately μ PDEs in geometries and networks that can be made up of several subdomains, seen as deformations of few reference shapes. This useful generalization implies the addition of some further complexities represented by proper coupling conditions and/or basis enrichment in the two approaches.

The first proposed method is the RBHM, it allows to successfully couple independent local RB problems defined in few reference shapes, through the help of the Lagrange multipliers and a coarse global solution, computed online. The second developed approach is the RDF method, as well as the previous one, it is able to recover the continuity of the solutions by exploiting some local RB problems defined in few reference shapes. The improvement of the latter consists in introducing further parameters in the local problems (μ BCs) that, together with the help of proper selected FE bases, are able to drastically increase the convergence of the coupled method with respect with the RBHM method. The introduction of several tunable ingredients make the RDF method a very flexible and suitable approach to easily fulfill a quick convergence requirement. On the other hand, as usual in the numerical analysis field, each technical improvement requires some further costs, in terms of computational complexities and times.

The two developed techniques represent an important and efficient improvements of the classical RB method when we need to perform simulation in composed networks dealing with viscous flows. Nevertheless, we point out that the RB method remains the best and very effective option to use when we are interested in problems involving a small set of parameter and defined in local geometries.

On the other hand, in order to deal with more complex networks and problems involving a large set of parameters the RBHM and the RDF methods represent two valid alternatives involving the main advantages of the RB method. Both of these approaches guarantee the continuity of the μ PDE solutions, the first one presents simpler local computation and a global coarse solution that allows to recover quite accurate solution. The second one is preferable when we need to considerably improve the accuracy of the solution and to avoid the computation of the online coarse solution on the global network.

In order to provide some examples of possible applications of the proposed approaches in the last part of the thesis we have presented several numerical results arising the study of internal flows in hierarchical parametrized coupled geometries such as the ones representing the circulatory system (e.g. bifurcations, stenosis, etc.).

At the moment the proposed techniques have allowed to solve efficiently problems characterized by a higher number of geometrical parameters, with respect to the current state of the art, and to provide a good approximation of viscous linear flows in terms of both velocity and pressure.

Perspectives and future works

For the presented work we would see some natural and straightforward developments in order to extend and improve the newly introduced methodologies. Moreover the reduced basis techniques (the existing ones and the new proposed variants) represent a very versatile framework which is finding a rich and growing field of applications and possible extensions. In the following we mention some important developments together with further original ideas to be developed on this branch of approaches.

- *A posteriori error estimate.* While in literature very useful error estimation theory has been developed for the classical reduced basis method (coercive and non-coercive problems), a posteriori error bounds for the proposed approaches represent a possible extensions, above all in the online evaluation of global solutions (classical a posteriori error theory applies to the local greedy approaches for the parameters space exploration).
- *Three dimensional relevant configurations.* The analyzed methods present a very feasible performance and a good versatility, so that they may successfully be applied to real clinical problems, for examples in simulations involving blood flows in large networks of arterial vessels, represented by suitable real 3D models, provided by medical doctors (thanks to coronagraphic computer assisted tomography (CAT) and advanced medical image reconstruction and data assimilations techniques in haemodynamic).
- *Non-linear μ PDEs.* The extension of the coupling between RB and DD approaches to a non-linear set of parametrized equation represents an important development of the proposed strategies, successfully applied for the solution of linear μ PDEs at the current state of the art. The extension to non-linear viscous fluid model (like Navier-Stokes equations) would allow to deal with many more micro-fluidic applications, characterized by higher Reynolds numbers.
- *Time dependency.* The extension to the time-dependent case is seen as more optimistic for the second proposed approach (RDF method) with respect to the first one (RBHM) dealing with

viscous flows and interface coupling conditions.

- *Optimization framework.* An interesting possible field of application involves inverse optimization problems such as parameter estimation, shape optimization for global networks (and not only local ones) and linear-quadratic optimal control problems.
- *Heterogeneous couplings.* Another possible extension is the use of the same methodologies to couple different fluid models at the interfaces (heterogeneous coupling) as, for examples, in Stokes-Darcy equations (porous media models) or in coupling potential/viscous flow models.

Bibliography

- [1] V. Agoshkov, A. Quarteroni, and G. Rozza. Shape design in aorto-coronary bypass anastomoses using perturbation theory. *SIAM J. Numer. Anal.*, 44(1):367–384, 2006.
- [2] B. O. Almroth, P. Stern, and F. A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16:525–528, 1978.
- [3] D. Ambrosi, A. Quarteroni, and G. Rozza. *Modelling of Physiological Flows*. Springer, Series MS&A, Vol. 5, 2011.
- [4] H. Antil, M. Heinkenschloss, and R.H.W. Hoppe. Domain decomposition and balanced truncation model reduction for shape optimization of the Stokes system. *Optim. Methods Softw.*, 26(4–5): 643–669, 2011.
- [5] J. A. Atwell and B. B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and Computer Modelling*, 33(1-3):1–19, 2001.
- [6] E. Balmes. Parametric families of reduced finite element models: Theory and applications. *Mechanical Systems and Signal Processing*, 10(4):381–394, 1996.
- [7] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.
- [8] A. Barrett and G. Reddien. On the reduced basis method. *Z. Angew. Math. Mech.*, 75(7):543–549, 1995.
- [9] J.H. Bramble. A proof of the inf-sup condition for the Stokes equations on Lipschitz domains. *Mathematical Models and Methods in Applied Sciences*, 13(3):361–371, 2003.
- [10] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer Verlag, 1991.
- [11] M.D. Buhmann. *Radial Basis Functions: Theory and Implementations*, volume 12 of *Cambridge monographs on applied and computational mathematics*. Cambridge University Press, UK, 2003.
- [12] C. Canuto, M.Y. Hussaini, and T. A. Quarteroni, A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer, 2007.
- [13] Y. Chen, J. Hesthaven, Y. Maday, and J. Rodriguez. A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations. *C. R. Acad. Sci. Paris, Ser. I*, 346:1295–1300, 2008.

- [14] E.A. Christensen, M. Brøns, and J.N. Sørensen. Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM J. Sci. Comput.*, 21:1419–1434, 1999.
- [15] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North Holland, January 1978.
- [16] L. Daniel, C.S. Ong, and J. White. Geometrically parametrized interconnect performance models for interconnect synthesis. In *Proceedings of the 2002 International Symposium on Physical Design*, ACM press, pages 202–207, 2002.
- [17] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. *SIAM J. Numer. Anal.*, 46(4):2039–2067, 2008.
- [18] S. Deparis and E. Løvgrén. Stabilized reduced basis approximation of incompressible three-dimensional Navier-Stokes equations in parametrized deformed domains. *Journal of Scientific Computing*, pages 1–15, 2011. ISSN 0885-7474.
- [19] S. Deparis and G. Rozza. Reduced basis method for multi-parameter-dependent steady Navier-Stokes equations: Applications to natural convection in a cavity. *Journal of Computational Physics*, 228(12):4359–4378, 2009.
- [20] J.L. Eftang, A.T. Patera, and E.M. Rønquist. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.
- [21] J.L. Eftang, D.J. Knezevic, and A.T. Patera. An “hp” certified reduced basis method for parametrized parabolic partial differential equations. *Math. Comput. Model. Dynam. Syst.*, 17(4):395–422, 2011.
- [22] J.L. Eftang, D.B.P. Huynh, D. J. Knezevic, E. M. Rønquist, and A.T. Patera. Adaptive port reduction in static condensation. In *Proceedings of 7th Vienna Conference on Mathematical Modelling*, 2012.
- [23] J.P. Fink and W.C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.*, 63(1):21–28, 1983.
- [24] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics. Modeling and simulation of the circulatory system*. Springer, Series MS&A, Vol. 1, 2009.
- [25] A.-L. Gerner and K. Veroy. Reduced basis a posteriori error bounds for the Stokes equations in parametrized domains: a penalty approach. *Math. Models Meth. Appl. Sci.*, 21(10):2103–2134, 2010.
- [26] A.-L. Gerner and K. Veroy. Certified reduced basis methods for parametrized saddle point problems. *Submitted*, 2011. Technical report AICES-2011/11-01: <http://www.aices.rwth-aachen.de:8080/aices/preprint/list2011.html>.
- [27] V. Girault and P.A. Raviart. *Finite element methods for Navier-Stokes equations. theory and algorithms*. Springer Verlag, 1986.
- [28] W. Gordon and C. Hall. Transfinite element method: blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21:109–129, 1973. ISSN 0029-599X.
- [29] M. Grepl. *Reduced-basis approximations and a posteriori error estimation for parabolic partial differential equations*. PhD thesis, Massachusetts Institute of Technology, 2005.

-
- [30] M. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 41(3):575–605, 2007.
- [31] M. Grepl, N.C. Nguyen, K. Veroy, A.T. Patera, and G.R. Liu. Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. Van Bloemen Waanders, editors, *Real-time PDE-Constrained Optimization*, pages 197–215. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007.
- [32] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows*. Academic Press, 1989.
- [33] M.D. Gunzburger, J.S. Peterson, and J.N. Shadid. Reducer-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comput. Methods Appl. Mech. Engrg.*, 196: 1030–1047, 2007.
- [34] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math. Modelling Numer. Anal.*, 42:277–302, 2008.
- [35] D.B.P Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Acad. Sci. Paris. Sér. I Math.*, 345:473–478, 2007.
- [36] D.B.P. Huynh, N.C. Nguyen, G. Rozza, and A.T. Patera. Rapid reliable solution of the parametrized partial differential equations of continuum mechanics and transport, 2008. <http://augustine.mit.edu>.
- [37] D.B.P. Huynh, D.J. Knezevic, Y. Chen, J.S. Hesthaven, and A.T. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Meth. Appl. Mech. Engrg.*, 199(29–32): 1963–1975, 2010.
- [38] D.B.P. Huynh, D.J. Knezevic, and A.T. Patera. A static condensation reduced basis element method: Approximation and a posteriori error estimation. *Mathematical Modelling and Numerical Analysis*, in press May 2012.
- [39] L. Iapichino, M. Lesinigo, A. Quarteroni, and G. Rozza. Rdf: Reduced basis, domain decomposition and finite element method: a combined perspective. *In preparation*, 2012.
- [40] L. Iapichino, A. Quarteroni, and G. Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Comput. Methods Appl. Mech. Engrg.*, 221–222:63–82, 2012.
- [41] K. Ito and S. Ravindran. A reduced-order method for simulation and control of fluid flow. *Journal of Computational Physics*, 143(2):403–425, 1998.
- [42] K. Ito and S.S. Ravindran. Reduced basis method for optimal control of unsteady viscous flows. *Int. J. Comput. Fluid Dynamics*, 15(2):97–113, 2001.
- [43] K. Ito and J. D. Schroeter. Reduced order feedback synthesis for viscous incompressible flows. *Mathematical And Computer Modelling*, 33(1-3):173–192, 2001.
- [44] C. Jaeggli, L. Iapichino, and G. Rozza. Some improvements on geometrical parametrizations by transfinite maps. *In preparation*, 2012.

Bibliography

- [45] S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski. A method for simulation based optimization using radial basis functions. *Optim. Eng.*, 11:501–532, 2010.
- [46] D. J. Knezevic and J. W. Peterson. A High-Performance Parallel Implementation of the Certified Reduced Basis Method. *Comput. Meth. Appl. Mech. Eng.*, 200(13–16):1455–1466, March 2011.
- [47] N. V. Krylov. *Lectures on Elliptic and Parabolic Equations in Sobolev Spaces*. Amer Mathematical Society, September 2008.
- [48] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2003.
- [49] T. Lassila and G. Rozza. Parametric free-form shape design with pde models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23–24):1583–1592, 2010.
- [50] T. Lassila, A. Quarteroni, and G. Rozza. A reduced basis model with parametric coupling for fluid-structure interaction problem. *SIAM Journal on Scientific Computing*, 34(2):A1187–A1213, 2012.
- [51] A. E. Løvgrén, Y. Maday, and E. M. Rønquist. The reduced basis element method: Offline-online decomposition in the nonconforming, nonaffine case. In J. S. Hesthaven and E. M. Rønquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76, pages 247–254. Springer Berlin Heidelberg, 2011.
- [52] A.E. Løvgrén, Y. Maday, and E.M. Rønquist. A reduced basis element method for complex flow systems. *Proceedings of ECCOMAS CFD, P. Wesseling, E. Onate, J. Periaux (Eds.) TU Delft, The Netherlands*, 2006.
- [53] A.E. Løvgrén, Y. Maday, and E.M. Rønquist. A reduced basis element method for the steady Stokes problem. *Mathematical Modelling and Numerical Analysis*, 40(3):529–552, 2006.
- [54] A.E. Løvgrén, Y. Maday, and E.M. Rønquist. Global C^1 maps on general domains. *Mathematical Models and Methods in Applied Sciences (M3AS)*, 19(5):803–832, 2009.
- [55] A.E. Løvgrén, Y. Maday, and E.M. Rønquist. The spectral element method used to assess the quality of a global C^1 map. In J. S. Hesthaven and E. M. Rønquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76, pages 441–448. Springer Berlin Heidelberg, 2011.
- [56] H.V. Ly and H.T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33:223–236, 2001.
- [57] L. Machiels, Y. Maday, I. B. Oliveira, A.T. Patera, and D.V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris, Série I*, 331(2):153–158, 2000.
- [58] Y. Maday and E.M. Rønquist. A reduced-basis element method. *J.Sci. Comput.*, 17:447–459, 2002.
- [59] Y. Maday and E.M. Rønquist. The reduced-basis element method: Application to a thermal fin problem. *SIAM J.Sci. Comput.*, 26:240–258, 2004.
- [60] Y. Maday, A.T. Patera, and G. Turinici. Global a priori convergence theory for reduced-basis approximation of single-parameter symmetric coercive elliptic partial differential equations. *C. R. Acad. Sci. Paris. Sér. I Math.*, 335:1–6, 2002.

-
- [61] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis method and free form deformation. *Int. J. Numer. Methods Fluids*, 2011. In press (DOI: 10.1002/flid.2712).
 - [62] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *Int. J. Numer. Methods Biomed. Engrg.*, 28(604-625), 2012.
 - [63] A. Manzoni, A. Quarteroni, and G. Rozza. Computational reduction for parametrized pdes: strategies and applications. *Milan J. Math.*, In press, 2012. Technical report MATHICSE 15.2012: <http://mathicse.epfl.ch/>.
 - [64] N.C. Nguyen, K. Veroy, and A.T. Patera. *Certified real-time solution of parametrized partial differential equations*. Handbook of Materials Modeling, S. Yip Ed., Kluwer Academic Publishing, Springer, 2005.
 - [65] N.C. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46(3):157–185, 2009.
 - [66] A.K. Noor and J.M. Peters. Recent advances in reduction methods for instability analysis of structures. *Comput. Struct.*, 16:67–80, 1983.
 - [67] A. T. Patera and E. M. Rønquist. Reduced basis approximations and a posteriori error estimation for a Boltzmann model. *Computer Methods in Applied Mechanics and Engineering*, 196:2925–2942, 2007.
 - [68] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering. Available at <http://augustine.mit.edu>.
 - [69] J.S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.*, 10(4):777–786, 1989.
 - [70] T. A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, 1985.
 - [71] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: reduced-basis output bounds methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.
 - [72] C. Prud'homme, D.V. Rovas, K. Veroy, and A.T. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 36(5):747–771, 2002.
 - [73] A. Quarteroni. *Numerical Models for Differential Problems*. Springer, Series MS&A , Vol. 2, 2009.
 - [74] A. Quarteroni and G. Rozza. Optimal control and shape optimization of aorto-coronary bypass anastomoses. *Math. Models Meth. Appl. Sci.*, 13(12):1801–1823, 2003.
 - [75] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations (1st Ed.)*. Springer-Verlag, Berlin-Heidelberg, 1994.
 - [76] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, 1999.

Bibliography

- [77] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations in industrial applications. *J. Math. Ind.*, 1(3), 2011.
- [78] W. C. Rheinboldt. On the theory and error estimation of the reduced basis method for multi-parameter problems. *Nonlinear Analysis, Theory, Methods and Applications*, 21(11):849–858, 1993.
- [79] D. Rovas, L. Machiels, and Y. Maday. Reduced basis output bounds methods for parabolic problems. *IMA J. Appl. Math.*, 2005.
- [80] D.V. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [81] G. Rozza. Reduced-basis methods for elliptic equations in sub-domains with *a posteriori* error bounds and adaptivity. *Appl. Numer. Math.*, 55(4):403–424, 2005.
- [82] G. Rozza. *Shape design by optimal flow control and reduced basis techniques: applications to bypass configurations in haemodynamics*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2005. N. 3400, <http://infoscience.epfl.ch>.
- [83] G. Rozza. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Comput. Vis. Sci.*, 12(1):23–35, 2009.
- [84] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Meth. Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.
- [85] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Engrg.*, 15:229–275, 2008.
- [86] G. Rozza, C.N. Nguyen, A.T. Patera, and S. Deparis. Reduced basis methods and a posteriori error estimators for heat transfer problems. *Proceedings of HT2009, ASME Summer Heat Transfer Conference, San Francisco, CA, USA, paper HT 2009–88211*, 2:753–762, 2009.
- [87] G. Rozza, D.B.P. Huynh, and A. Manzoni. Reduced basis approximation and a posteriori error estimation for stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Submitted*, 2010.
- [88] G. Rozza, T. Lassila, and A. Manzoni. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. In E.M. Rønquist J.S. Hesthaven, editor, *Spectral and High Order Methods for Partial Differential Equations. Selected papers from the ICOSAHOM '09 conference, June 22–26, Trondheim, Norway*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 307–315. Springer, 2011.
- [89] W. Rudin. *Functional analysis*. McGraw-Hill Science/Engineering/Math, 1991.
- [90] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *Comput. Graph.*, 20(4):151–160, 1986.
- [91] S. Sen, K. Veroy, D.B.P. Huynh, S. Deparis, N.C. Nguyen, and A.T. Patera. “Natural norm” a posteriori error estimators for reduced basis approximations. *J. Comp. Phys.*, 217(1):37–62, 2006.
- [92] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory*. Springer Verlag, 2005.

- [93] L.N. Trefethen and III D. Bau. *Numerical Linear Algebra*. SIAM: Philadelphia, PA, 1997.
- [94] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47 (8-9):773–788, 2005.
- [95] K. Veroy, D. Rovas, and A. T. Patera. *A Posteriori* error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: “convex inverse” bound conditioners. *ESAIM: Control, Optimization and Calculus of Variations*, 8:1007–1028, 2002.
- [96] K. Veroy, C. Prud’homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proc. 16th AIAA Comput. Fluid Dynamics*, 2003.
- [97] V. Volpert. *Elliptic Partial Differential Equations*. Fredholm Theory of Elliptic Problems in Unbounded Domains. Birkhauser, 2011.
- [98] K. Yoshida. *Functional Analysis*. Springer Verlag, 1980.

Laura Iapichino

Personal Information

Nationality	Italian
Date of birth	6 th March, 1984

Work Experience

01/2009 - Present	<p>Research Assistant at EPFL-MATHICSE-CMCS, Lausanne, Switzerland.</p> <p><i>Responsibilities:</i></p> <p>Carrying out research activities in the field of numerical approximation of partial differential equations for computational fluid dynamics. Investigation, development and application of new model order reduction techniques by combining the Reduced Basis Method with the Domain Decomposition technique.</p> <p>Defining and developing efficient procedures to obtain rapid and reliable numerical solution for parametrized applications by reducing both the computational costs and the geometrical complexity.</p> <p>Performing accurate real-time simulations for repeated output evaluations for different values of some inputs of interest, in particular for simulating hierarchical flow systems and to model the blood flow in complex arterial networks.</p> <p>Assistant for 7 cycles of Numerical Analysis Courses for Engineers based on theoretical and computer exercises in Matlab and Octave environment.</p> <p>Responsible of a semester project.</p>
03/2008 - 12/2008	<p>Software Developer at APL Italiana S.p.A, Milano, Italy.</p> <p><i>Responsibilities:</i></p> <p>Improving and monitoring the life cycle of SOFIA, the product provided by the APL company to the main italian insurance groups for managing their assets.</p> <p>Development and testing of new functionalities written in APL language to satisfy the specific change requests coming from company's customers.</p>
09/2007 - 12/2007	<p>Research Intern at ESA-ESTEC, Noordwijk, The Netherlands.</p> <p><i>Responsibilities:</i></p> <p>Development of a model for estimation of the risk generated by the destructive re-entry of the Automatic Transfer Vehicle (ATV). In particular, creation of a software called PRISM able to compute the footprint size of the ATV debris after its explosion or fragmentation, its location in a map of the world and the corresponding fatality and casualties probabilities of the impacts.</p>

Education

- 2009 - Present** **Ph.D. Degree in Applied Mathematics, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.**
Advisors: Prof. Alfio Quarteroni, Dr. Gianluigi Rozza.
Dissertation Topic: “Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD”.
Research Topics: Investigation, development and application of new model order reduction techniques in computational fluid-dynamics.
Ph.D. Thesis financed by: Swiss National Science Foundation under the Project 122136 and by ERC-Mathcard Project (ERC-2008-AdG-2270058)
- 2005 - 2007** **M.Sc. in Applied Mathematics, Università degli studi di Milano, Milano, Italy.**
Grade: “110/110”.
Thesis Topic: “Risk analysis of space debris impact”.
Main Subjects: Scientific Computing, Scientific Calculation, Image Elaboration, Continuum Mechanics, Partial Differential Equations, Approximations Methods I and II, Numerical Analysis II, Algorithms and Data Structures.
- 2002 - 2005** **B.Sc. in Applied Mathematics, Università degli studi di Catania, Catania, Italy.**
Grade: “110/110 cum laude”.
Thesis Topic: “The conjugate gradient method for the resolution of algebraic linear systems”.
Main Subjects: Mathematical Analysis I, II, III and IV, Physics I and II, Geometry I and II, Algebra I and II, Linear Algebra, Computer Science, Mathematical Physics, General Topology, Statistics, Probability Calculation, Economy, Financial Mathematics, Graphics Theory, Optimization, Operational Search, Numerical Analysis I and II, Dynamic Systems, Numerical Computing.

Publications

- L. Iapichino, A. Quarteroni and G. Rozza, “A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks”, *Computer Methods in Applied Mechanics and Engineering*, 221-222: 63-82, 2012.
- L. Iapichino, M. Lesinigo, A. Quarteroni and G. Rozza. RDF: Reduced basis, Domain decomposition and Finite element method: a combined perspective. In preparation, 2012.
- C. Jaeggli, L. Iapichino and G. Rozza. Some improvements on geometrical parametrizations by transfinite maps. In preparation, 2012.

Conferences

- Workshop on Reduced Basis, POD and Reduced Order Methods for model and computational reduction, Poster session, “Reduced basis, Finite element method and Domain Decomposition: a combined perspective”, EPFL, Lausanne (Switzerland), May 14-16, 2012.

Conferences (continued)

RMMM 2011, Reliable Method of Mathematical Modeling, Contributed talk, “A Reduced Basis Hybrid Method for the study of parametrized cardiovascular geometries and networks”, EPFL, Lausanne (Switzerland), July 6-8, 2011.

Workshop on Numerical Solutions of Partial Differential Equations: Novel Discretization Techniques, Institute of Mathematics and its Applications, University of Minnesota (Minneapolis, US), Poster session, “A reduced basis hybrid method for viscous flows in parametrized complex networks”, November 1-5, 2010.

MPF 2010, IV International Symposium on Modelling of Physiological flows, Centro Congressi Chia Laguna Resort, Sardinia (Italy), Contributed talk, “Reduced basis (element) methods for the study of parametrized cardiovascular geometries and networks”, June 2-5, 2010.