Stabilized reduced basis methods for the approximation of parametrized viscous flows

Author

Shafqat Ali

Supervisors

Prof. Gianluigi Rozza, SISSA

Dr. Francesco Ballarin, SISSA

A thesis submitted in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Mathematical Analysis, Modelling and Applications at SISSA, International School for Advance Studies, Trieste, Italy

September 26, 2018, Trieste, Italy

[This page intentionally left blank]

Abstract

In Reduced Basis (RB) method, the Galerkin projection on the reduced space does not guarantee the inf-sup approximation stability even if the stable Taylor-Hood Finite Element pair is chosen. Therefore in this PhD thesis we aim to build a stabilized RB method suitable for the approximation of parametrized viscous flows. Starting from the state of the art we study the residual based stabilization techniques for parametrized viscous flows in a RB setting. We are interested in the approximation of the velocity and pressure. *Offline-online* computational splitting is implemented and *offline-only stabilization*, and *offline-online stabilization* are compared (as well as without a stabilization approach). Different test cases are illustrated and several classical stabilization approaches like Brezzi-Pitkaranta, Franca-Hughes, streamline upwind Petrov-Galerkin, Galerkin Least Square are recast into a parametric reduced order setting. The RB method is introduced as a Galerkin projection into reduced spaces, generated by basis functions chosen through a greedy (steady cases) and POD-greedy (unsteady cases) algorithms. This approach is then compared with the supremizer options to guarantee the approximation stability by increasing the corresponding parametric inf-sup condition. We also implement a rectification method to correct the consistency of *offline-only stabilization* approach. Several numerical results for both steady and unsteady problems are presented and compared. The goal is two-fold: to guarantee the RB inf-sup stability and to guarantee online computational savings by reducing the dimension of the online reduced basis system.

Dedicated to my loving *parents*, my wife, *Asma*, and my cute son, *Maaz Ali*

Acknowledgement

This PhD thesis would not have been possible without the help and contributions of many people. First of all, I would like to thank my supervisor, *Prof. Gianluigi Rozza,* for the patient guidance, encouragement and advice he has provided me throughout my time as PhD student. I have been extremely lucky to have supervisor who always cared so much about my work, and who responded to my questions and queries so promptly.

I would like also to convey my deepest gratitude to my second advisor *Dr. Francesco Ballarin* for his support, constructive suggestions and kind help that paved a joyful and enthusiastic way in accomplishing this thesis. I appreciate all his contributions of time, challenging ideas, critical remarks, constant guidance and support to make my Ph.D. experience productive.

I would also like to thank all the members of our group at SISSA mathlab, who helped me in my research activities and shared new ideas. Special thanks to my colleague *Zakia Zainib* for her valuable comments and discussion with me on the topic.

I am grateful to the student secretariat office for helping and assisting me during the last four years, in particular *Riccardo Iancer*, *Federica Tuniz* and *Luisella Zecchini* deserve special mention.

Some special words of gratitude go to all my friends back in Pakistan who have always been a major source of support when things would get a bit discouraging: *Ramzan Ali, Jam Sadiq, Farooq, Taimoor, Masood, Talib, Jabbar, Waseem, Javed, Amman, Naveed and Khurram.* Thanks guys for always being there for me.

Surviving in a different world would not have been easy without a few good friends around. I would like to thank *Farhan, Tahir, Nawaz, Imran, Rafaqat, Nouman, Haris* and *Adnan* for accompanying me as close family friends in Trieste.

I must express my gratitude to *Asma*, my wife, for her love and constant support, for all the late nights and early mornings, and for keeping me sane over the past few months. Thank you for being with me, thank you for being my best friend. I owe you everything.

Last but not least, I would like to express my love and gratitude to my family: my parents, my brothers and sisters, my grandmother and mother in law. Without their support, love and encouragement, I would not have been able to pursue my dreams.

Trieste, Italy 2018

Outline

This PhD thesis is divided into five chapters. Here we give the outline of the organization of this thesis by giving a summary of each chapter:

Chapter 1
We start with a brief review of the historical development in residual based stabilization techniques for Finite Element (FE) method. For notation purposes, we give a brief introduction of Reduced Basis method (RB) for elliptic case and a review of algorithms (Greedy, POD and POD-greedy) required in the construction of RB spaces in the following chapters. We also introduce the notion of geometrical parametrization and finally we give the objective and motivation of this thesis.

Chapter 2
We study the stabilization of steady and unsteady Stokes problem in parametric reduced order setting. The goal is to give a stable and computationally less expensive RB solution for velocity and pressure. In the first part we construct a stabilized RB method for steady Stokes problem using classical stabilization techniques (Franca-Hughes in particular) for FE and, then, projecting on RB. Additional stabilization terms allows us to consider the FE pair $\mathbb{P}_k/\mathbb{P}_k; k = 1, 2$, instead of the classical Taylor-Hood FE pair $\mathbb{P}_2/\mathbb{P}_1$. We apply this method to a cavity flow problem with geometrical parametrization and compare our results with supremizer enrichment approach. We present the comparison of computational performances of different possible cases. In the second part we discuss the stabilized RB formulation of unsteady Stokes problem. Similar to steady case we apply this method to a cavity flow problem and numerical results for different possible stabilization options are discussed.

Chapter 3
This chapter extends the analysis carried out in Chapter 2 to the nonlinear problems in parametric reduced order settings. We divide this Chapter into two parts: In the first part we present the stabilized RB method for the approximation of steady Navier-Stokes problem. We apply this method to a cavity flow problem taking into consideration both physical and geometrical parametrization. Numerical results are presented for different options and computational performances of several cases are compared through tables. We also give a comparison with supremizer enrichment option. In the second part we extend the first part to the unsteady Navier-Stokes problem and repeat the same numerical tests for the physical and geometrical parametrization.

Chapter 4

In this chapter, we present the new *online* intrinsic stabilization strategy for the RB approximation of parametrized steady Stokes problem based on rectification method. The main goal of this chapter is to give a post processing technique based on rectification method that helps in correcting the consistency of *offline-only stabilization* approach. We consider the same cavity flow problem with geometrical parameter. In this chapter we also give a test case with more than one geometrical parameter (T-shape) to proof the validity of rectification method in more complex problems.

Chapter 5

This Chapter summarizes all the important findings of the thesis. We address future plans and perspectives in continuation of this work. We also discuss the impact and connection of this work with different topics, for instance, Fluid Structure Interaction (FSI) problems, optimal control problems and the problems involving high Reynolds number.

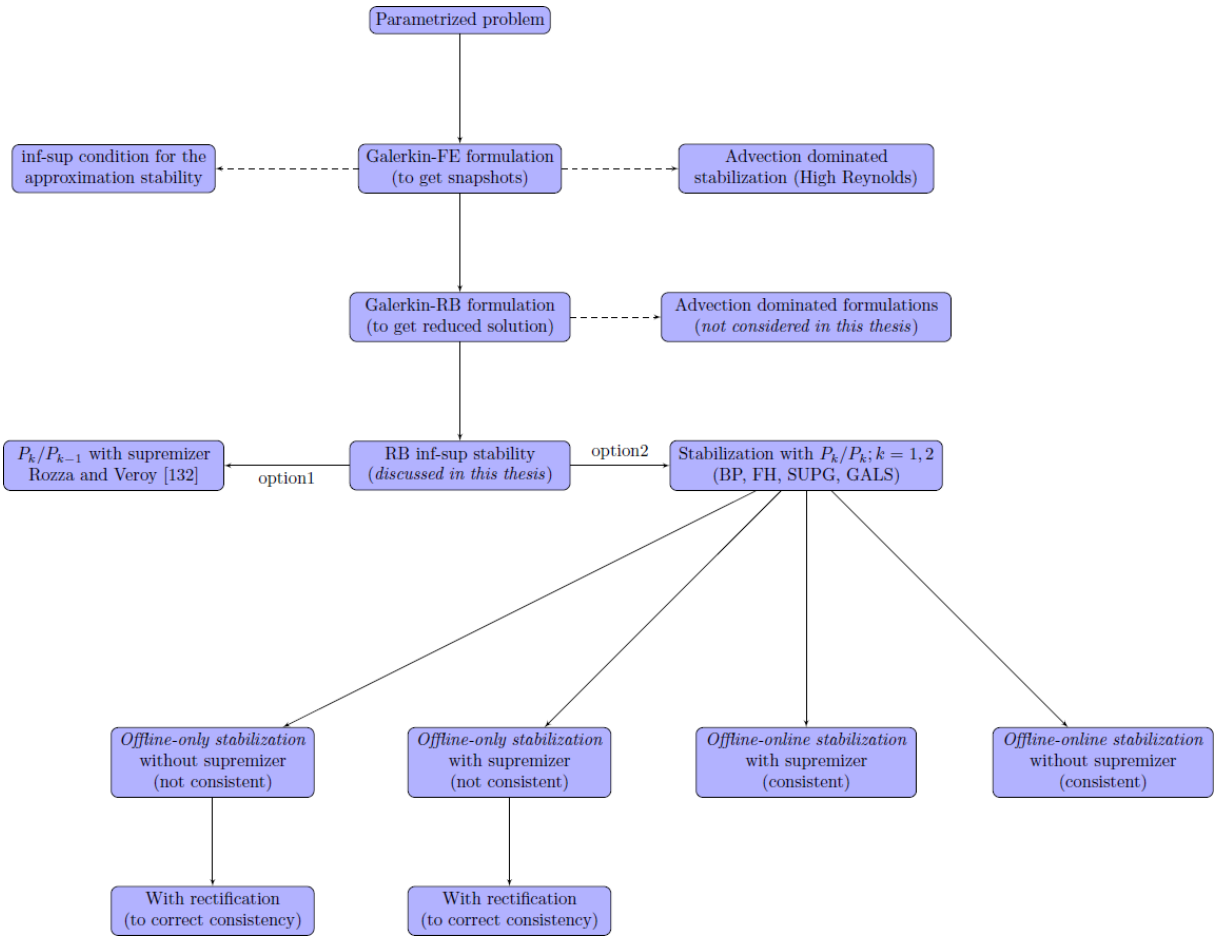


Figure 1: Scheme of thesis.

Contents

1 Introduction and Motivation **1**
- 1.1 Brief excurses literature on historical development of stabilization 1
- 1.2 Reduced Basis Methods 2
  - 1.2.1 The solution manifold and RB approximation 3
  - 1.2.2 *Offline-online* computational decoupling 7
  - 1.2.3 Greedy basis generation 8
  - 1.2.4 Proper orthogonal decomposition (POD) 9
  - 1.2.5 A *posteriori* error estimates 10
  - 1.2.6 POD-greedy sampling 12
- 1.3 Geometrical parametrization 14
- 1.4 Motivation and objective of the thesis 16

2 Reduced Basis Stabilization of Parametrized Stokes Problem **18**
- 2.1 Parametrized steady Stokes problem 19
  - 2.1.1 Finite Element formulation 20
  - 2.1.2 Stabilized Finite Element formulation 21
  - 2.1.3 Reduced Basis formulation 22
  - 2.1.4 Stabilized Reduced Basis formulation 23
- 2.2 Numerical results and discussion 25
  - 2.2.1 Numerical results for $\mathbb{P}_k/\mathbb{P}_k$ for $k = 1, 2$ 26
  - 2.2.2 Numerical results for $\mathbb{P}_1/\mathbb{P}_0$ 31
  - 2.2.3 Numerical results for stable $\mathbb{P}_2/\mathbb{P}_1$ 33
- 2.3 Unsteady parametrized Stokes problem 35
  - 2.3.1 Semi-discrete Finite Element formulation 36
  - 2.3.2 Stabilized Finite Element formulation 37
  - 2.3.3 Reduced Basis formulation 38
  - 2.3.4 Stabilized Reduced Basis formulation 39
- 2.4 Numerical results and discussion 40
  - 2.4.1 Numerical results for $\mathbb{P}_k/\mathbb{P}_k$ for $k = 2$ 41
  - 2.4.2 Numerical results for $\mathbb{P}_1/\mathbb{P}_0$ 45
  - 2.4.3 Sensitivity on Δt 46
- 2.5 Concluding remarks and perspectives 48

3 Reduced Basis Stabilization of Parametrized Navier-Stokes Problem 49
3.1 Parametrized steady Navier-Stokes problem 50
3.1.1 Finite Element formulation 51
3.1.2 Stabilized Finite Element formulation 52
3.1.3 Reduced Basis formulation 53
3.1.4 Stabilized Reduced Basis formulation 54
3.2 Numerical results and discussion 54
3.2.1 Results for physical parametrization 55
3.2.2 Results for physical and geometrical parameterization 63
3.3 Parametrized unsteady Navier-Stokes problem 67
3.3.1 Discrete Finite Element formulation 68
3.3.2 Stabilized Finite Element formulation 69
3.3.3 Reduced Basis formulation 70
3.3.4 Stabilized Reduced Basis formulation 70
3.4 Numerical results and discussion 71
3.4.1 Results for physical parameter case only 71
3.4.2 Results for physical and geometrical parameters 74
3.5 Concluding remarks 76

4 An Online Stabilization Method for Parametrized Stokes Problems 78
4.1 Rectification method for advection-diffusion problem 78
4.1.1 Online vanishing viscosity stabilization method 80
4.1.2 Rectification method 81
4.1.3 Numerical results and discussion 82
4.2 Rectification method for Stokes problem 84
4.2.1 Numerical results and discussion 87
4.3 Concluding remarks 91

5 Conclusion and Perspectives 92
5.1 Summary 92
5.2 Perspectives for future work 93

Chapter 1

Introduction and Motivation

In this chapter we start giving a brief literature review about the classical stabilization techniques for the Finite Element methods (FE) implemented to Advection-diffusion problem, Stokes problem and Navier-Stokes problem. The main goal of this thesis is to properly extend such stabilization techniques into the Reduced Basis (RB) method [63].

We have organized this chapter as follows: in section 1.1, we present a brief historical literature review on classical stabilization techniques for FE methods. In section 1.2, we introduce the RB method, followed by the definition of solution manifold and the RB approximation of a parametric PDE in subsection 1.2.1. We present the *offline-online* decomposition for RB method in subsection 1.2.2. In subsection 1.2.3 and 1.2.4, we discuss the greedy and POD algorithms, respectively while in section 1.2.6, we present the POD-greedy algorithm involved in the construction of RB space for time-dependent RB problems. A *posteriori* error estimation and error bounds for RB method are described in subsection 1.2.5. A brief review about the geometrical parametrization for the scalar case is presented in section 1.3. Finally, the objective and the final goal of this thesis is presented in section 1.4 with the focus on the inf-sup stability condition for RB method.

1.1 Brief excurses literature on historical development of stabilization

In the FE computations of incompressible flows, using a standard Galerkin formulation there are two possible sources of instabilities. One reason could be due to the presence of the convection term which for high Reynolds number creates instability in numerical solution. Other source of instability could be due to the inappropriate choice of interpolating functions for velocity and pressure.

In 1970s, many researchers [39, 60, 61, 120] proposed stable schemes to overcome the instabilities caused by increasing Reynolds number. Later on, Hughes [71] introduced a different scheme for one-dimension steady state advection diffusion problem but his scheme was only first order accurate and would not give a consistent formulation when extended to unsteady cases. Hughes and Brooks [29, 30, 66] proposed to add artificial

diffusion term acting only in the streamline direction and named this type of formulation as streamline upwind/ Petrov-Galerkin (SUPG) formulation. An extension of SUPG formulation is given by Hughes et al. [69] and is named as Galerkin Least Square (GLS) formulation. Later on Franca et al. [44] introduced a change of sign in GLS formulation.

In order to overcome the instability caused by the inappropriate choice of velocity and pressure interpolating functions, Taylor and Hood [138] employed equal-order velocity pressure interpolation not satisfying inf-sup condition [25] and noticed that velocity is good but the pressure approximation is very poor. Penalty method in which pressure is eliminated by penalizing the continuity equation, and then retained in boundary condition was introduced by Hughes et al. [70]. Penalty method is also studied by Chacón et al. [34]. Hughes et al. [68] used equal order interpolation for velocity and pressure by perturbing the pressure test function with a gradient term to achieve the stability. A symmetric version of this method was proposed by Hughes and Franca [67]. The SUPG method, first applied by Brooks and Hughes [30] to solve numerically the incompressible Navier-Stokes equations with high Reynolds number was later on extended by several researchers [44, 49, 139]. An additional term called pressure stabilizing Petrov-Galerkin (PSPG) is created by Johnson and Saranen [81]. The streamline upwind/pressure stabilizing Petrov-Galerkin (SUPG/PSPG) method is also studied in the references [40, 142]. An extension of the SUPG/PSPG stabilization to the Oseen and the Navier-Stokes problem was independently analyzed by Hansbo et al. [58], Franca et al. [48] and Tobiska et al. [141]. They have introduced an additional stabilization of the divergence constraint and named it as *grad-div stabilization*.

Another stabilization method called Streamline Diffusion method (SD) for convection-dominated convection-diffusion problems was introduced by Hughes and Brooks [72] and then Johnson [81] extended this method to time-dependent two-dimensional Navier-Stokes equations for an incompressible Newtonian fluid in the case of high Reynolds number. Stabilization techniques used to stabilize advection-diffusion problem and Stokes problem are also proposed for the Navier-Stokes equations by Behr et al. [14]. Stream function-vorticity formulation of Navier-Stokes equations has been used in flow computation, also combined with stabilization techniques [140].

In the SUPG and GLS stabilizations the stabilizating terms added involves the residual of momentum equation as a factor. Consequently, when an exact solution is substituted into the stabilized formulation, these added terms vanishes and as a result the stabilized formulation is satisfied by the exact solution in the same way as the Galerkin formulation is satisfied [117].

1.2 Reduced Basis Methods

In this section we introduce the Reduced Basis (RB) method as a cheaper method (in terms of CPU time) to perform numerical simulations of (stabilized) parametrized problems. The RB method does not replace the stabilized FE method, but rather builds upon it. For this reason, sometimes we will call the FE method as *truth* or *high fidelity* method. Indeed, RB method is based on a two stage procedure, comprising an *offline*

and an *online* stage. During the potentially very costly *offline* stage, one empirically explores the solution manifold by querying the FE solver to construct a RB that will approximates the whole solution manifold. The *online* stage consists of a Galerkin projection onto the space spanned by the RB. During this stage, one can explore the parameter space at a substantially reduced cost owing to the small size N of the RB, which is assumed to be much less than the number $\mathcal{N}$ of degree of freedom of *truth* discretization.

This *offline-online* separation is very useful in all those scenarios which are characterized by repeated queries for several different parameter values, where a direct and repeated evaluation of *truth* solution would be extremely costly. This is the case for instance in areas of optimization, design, uncertainty quantification. Moreover, the *online* procedure can be embedded in a computer environment that has only limited computational power and memory to allow rapid *online* query of the response of a complex system for control, optimization and analysis using a deployed device.

In the next subsection we start with an introduction to the concept of solution manifold and RB approximation. In this chapter we deal with the RB method only for elliptic problems to introduce the concepts and notations etc, while in the remaining part of the thesis we deal with Stokes and Navier-Stokes problems..

1.2.1 The solution manifold and RB approximation

We first introduce a physical domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$, where $d = 1, 2$, or 3 is the spatial dimension since in this chapter we are dealing with elliptic problems. We define the scalar space $V = H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$, with Γ_D is the part of $\partial\Omega$ where we impose the Dirichlet boundary condition. Let $\mathbb{P}$ be the parameter domain, a subset of $\mathbb{R}^P$. We introduce the parametrized bilinear form $a : V \times V \times \mathbb{P} \to \mathbb{R}$, where the bilinearity is with respect to first two variables; and the parametrized linear form $f : V \times \mathbb{P} \to \mathbb{R}$, where the linearity is with respect to the first variable [63, 115]. Moreover, for a selected value of $\boldsymbol{\mu} \in \mathbb{P}$, the bilinear form $a(.,.;\boldsymbol{\mu})$ enjoy the continuity and coercivity property, whereas the linear form $f(.;\boldsymbol{\mu})$ is a continuous linear operator [129].

Let us now define the problem of interest as: for any $\boldsymbol{\mu} \in \mathbb{P}$, find $u(\boldsymbol{\mu}) \in V$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in V, \tag{1.1}$$

where $\mathbb{P} \subset \mathbb{R}^P$ is a compact parameter domain. We shall refer to $u(\boldsymbol{\mu})$ as exact solution.

The energy inner product and energy norm are defined as:

$$\begin{aligned} ((w, v))_{\boldsymbol{\mu}} &\equiv a(w, v; \boldsymbol{\mu}), \quad \forall w, v \in V, \\ |||w|||_{\boldsymbol{\mu}} &\equiv ((w, w))_{\boldsymbol{\mu}}^{1/2}, \quad \forall w \in V, \end{aligned} \tag{1.2}$$

where we assume that $a(.,.;\boldsymbol{\mu})$ is symmetric [63]. Next, we also define the inner product and the norm that are not parameter dependent. Thus for a given $\bar{\boldsymbol{\mu}} \in \mathbb{P}$ and real $\tau > 0$,

$$\begin{aligned} (w, v)_V &\equiv ((w, v))_{\bar{\boldsymbol{\mu}}} + \tau(w, v)_{L^2(\Omega)}, \quad \forall w, v \in V, \\ \|w\|_V &\equiv (w, w)_V^{1/2}, \quad \forall w \in V, \end{aligned} \tag{1.3}$$

where $(w,v)_{L^2(\Omega)} \equiv \int_\Omega wv$. The coercivity and continuity constants are defined as:

$$\exists\, \alpha_0 \text{ s.t. } \alpha_0 \leq \alpha(\boldsymbol{\mu}) = \inf_{w\in V} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_V^2}, \quad \forall \boldsymbol{\mu} \in \mathbb{P} \tag{1.4}$$

and

$$+\infty > \gamma(\boldsymbol{\mu}) = \sup_{w\in V}\sup_{v\in V} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_V\|v\|_V}, \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{1.5}$$

Now we make an important assumption that has important role in *offline-online* computational decoupling in RB. This assumption is called the affine dependence of $a(.,.;\boldsymbol{\mu})$ on the parameter $\boldsymbol{\mu}$ [63]. With this assumption we mean that the bilinear form $a(.,.;\boldsymbol{\mu})$ can be written as:

$$a(w,v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w,v), \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{1.6}$$

where $\Theta_a^q(\boldsymbol{\mu}) : \mathbb{P} \to \mathbb{R}$, $q = 1, ..., Q_a$, are smooth functions, and $a^q : V \times V \to \mathbb{R}$, are $\boldsymbol{\mu}$-independent continuous bilinear forms. Similarly, we assume that the linear functional $f(.;\boldsymbol{\mu})$ also depends affinely on the parameter, i.e.

$$f(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f^q(v), \quad \forall \boldsymbol{\mu} \in \mathbb{P}, \tag{1.7}$$

where $\Theta_f^q(\boldsymbol{\mu}) : \mathbb{P} \to \mathbb{R}$, $q = 1, ..., Q_f$, are smooth functions, while $f^q : V \times V \to \mathbb{R}$, are $\boldsymbol{\mu}$-independent linear functionals.

Let us introduce the notion of solution manifold comprising of all solutions of the parametric problem under the variation of parameter, i.e.,

$$\mathcal{M} = \{u(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \mathbb{P}\} \subset V.$$

We assume that, under suitable regularity assumptions on the data [63], the solution manifold $\mathcal{M}$ is smooth.

In many cases of interest, we are not able to find the exact solution in an analytic manner. Therefore for any $\boldsymbol{\mu} \in \mathbb{P}$, we seek an approximate solution $u_h(\boldsymbol{\mu}) \in V_h$ such that

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h;\boldsymbol{\mu}), \qquad \forall\, v_h \in V_h, \tag{1.8}$$

where $u_h(\boldsymbol{\mu})$ referred to as *truth* solution and V_h is FE space (e.g. Lagrange basis in this case).

Here $V_h \subset V$ is a sequence of conforming FE approximation spaces. Therefore it follows from assumptions on a and f that (1.8) admits a unique solution [112]. Also the conditions (1.4) and (1.5) for members of V_h are fulfilled by restriction. The FE coercivity constant can be defined as

$$\alpha_h(\boldsymbol{\mu}) = \inf_{w_h\in V_h} \frac{a(w_h,w_h;\boldsymbol{\mu})}{\|w_h\|_V^2}, \quad \forall \boldsymbol{\mu} \in \mathbb{P} \tag{1.9}$$

where, by restriction

$$\alpha(\boldsymbol{\mu}) \leq \alpha_h(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathbb{P}.$$

Similarly we define the FE continuity constant as

$$+\infty > \gamma_h(\boldsymbol{\mu}) = \sup_{w_h \in V_h} \sup_{v_h \in V_h} \frac{a(w_h, w_h; \boldsymbol{\mu})}{\|w_h\|_V \|v_h\|_V}, \quad \forall \boldsymbol{\mu} \in \mathbb{P}, \tag{1.10}$$

and it follows that

$$\gamma(\boldsymbol{\mu}) \geq \gamma_h(\boldsymbol{\mu}). \quad \forall \boldsymbol{\mu} \in \mathbb{P}.$$

As the main goal of the thesis is to investigate the RB method (and not the FE one), in the subsequent discussion we assume that for any given parameter value $\boldsymbol{\mu} \in \mathbb{P}$, the quantity $||u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})||_V$ can be made arbitrarily small. This state that a computational model is available to solve the *truth* problem and hence approximate the exact solution at any required accuracy. Unfortunately, this accuracy requirement also implies that the computational cost (e.g. in terms of CPU time) of evaluating the *truth* model may be very high and depend directly on $\mathcal{N}$=dim(V_h).

Following the definition for the continuous problem, we also define the discrete version of the solution manifold [63, 115]

$$\mathcal{M}_h = \{u_h(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \mathbb{P}\} \subset V_h.$$

The main idea of the RB method is to take advantage of the smoothness of M_h to provide a computationally efficient RB method. A central assumption in the development of any reduced model is that the solution manifold is of low dimension, i.e., that the span of low number of approximately chosen basis functions represents the solution manifold with a small error [63].

Intuitively, we can represent the approximation of the truth manifold by mean of the Lagrangian RB method as sketched in Figure 1.1, when we are dealing with one dimensional parameter $\boldsymbol{\mu}$. The black line is the truth manifold in the $\mathcal{N}$-dimensional space V_h. The black dots represent the snapshot solutions, which act like Lagrangian interpolation nodes. Finally, the red dashed "interpolant" is our RB approximation, that is built by linear combination of snapshot solutions.

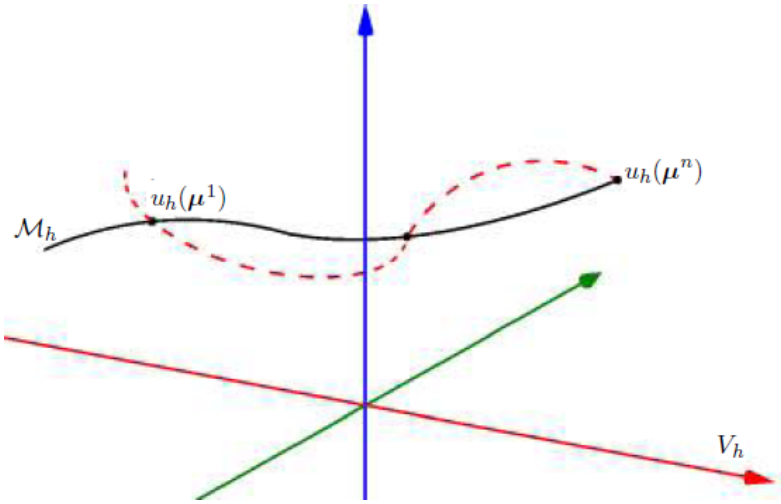


Figure 1.1: Solution Manifold

Let us introduce an integer $N_{max} \ll \mathcal{N}$, a sequence of at most N_{max} subspaces of V_h. Let us suppose that $\{V_N\}_{N=1}^{N_{max}}$ be a N-dimensional hierarchical subspace [129] of V_h such that:

$$V_1 \subset V_2 \subset ... \subset V_N \subset ... \subset V_{N_{max}}. \tag{1.11}$$

In general it is not necessary to choose these subspaces hierarchical. Nevertheless, it is very useful because it allows a better exploitation of the memory during the computation and consequently, the efficiency of the method is improved. In this thesis we will focus on Lagrange RB spaces. In order to define these spaces, we introduce a set of N_{max} parameter values:

$$\Xi = \{\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^{N_{max}}\}. \tag{1.12}$$

We shall call these basis functions as RB approximation space and it will allow us to represent the *truth* solution, $u_h(\boldsymbol{\mu})$ based on an N-dimensional subspace V_N of V_h. Let us assume that an N-dimensional RB, denoted as $\{\xi_n\}_{n=1}^{N}$ is available, then the associated RB space is given by:

$$V_N = \text{span}\{\xi_n = u_h(\boldsymbol{\mu}^n) | 1 \leq n \leq N\} \subset V_h, \tag{1.13}$$

for $N = 1, ..., N_{max}$. The assumption of the low dimensionality of the solution manifold implies that $N \ll \mathcal{N}$. The main idea of this particular definition is to approximate the *truth* manifold (1.8) with corresponding parameter values belonging to Ξ.

Let us introduce the Galerkin projection on N dimensional RB space V_N. The RB approximation is sought as: for any given $\boldsymbol{\mu} \in \mathbb{P}$, find $u_N(\boldsymbol{\mu}) \in V_N$ s.t.

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}), \qquad \forall\, v_N \in V_N, \tag{1.14}$$

where $u_N(\boldsymbol{\mu})$ represents the RB solution. We emphasize that in order to find the RB solution we only need to solve $N \times N$ linear system, instead of solving $\mathcal{N} \times \mathcal{N}$ system of

FE method. In order to proceed with the *offline-online* decoupling, first we apply the Gram-Schmidt orthogonalization process [63] to the *snapshots* $u(\boldsymbol{\mu}^n), n = 1, ..., N_{max}$, spanning the RB space V_N. The obtained mutually orthonormal basis functions are denoted by $\xi_1, \xi_2, ..., \xi_N$, therefore $u_N(\boldsymbol{\mu})$ is represented by

$$u_N(\boldsymbol{\mu}) = \sum_{n=1}^{N} u_{Nn}(\boldsymbol{\mu})\xi_n, \tag{1.15}$$

where $\{u_{Nn}(\boldsymbol{\mu})\}_{n=1}^{N}$ denote the coefficients of the RB approximation. Rewriting (1.14) we arrive at following $N \times N$ linear system [115]

$$\sum_{n=1}^{N} a(\xi_n, \xi_m; \boldsymbol{\mu})u_{Nn}(\boldsymbol{\mu}) = F(\xi_m; \boldsymbol{\mu}). \tag{1.16}$$

1.2.2 *Offline-online* computational decoupling

A key assumption necessary for an efficient RB is the capability to decouple the *offline* of the RB space from the *online* stage, called the *offline-online* decomposition. We now define the affine decomposition of the system (1.16) to construct an efficient *offline-online* procedure, thanks to the assumptions (1.6) and (1.7). The system (1.16) can be rewritten as

$$\sum_{n=1}^{N} \left(\sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(\xi_n, \xi_m) \right) u_{Nn}(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_F} \Theta_F^{q'}(\boldsymbol{\mu}) f^{q'}(\xi_m) \tag{1.17}$$

for $m = 1, ..., N$. Above system can be expressed in matrix form as:

$$\left(\sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) A_N^q \right) \boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_F} \Theta_F^{q'}(\boldsymbol{\mu}) F_N^{q'} \tag{1.18}$$

where

$$(\boldsymbol{u}_N(\boldsymbol{\mu}))_n = u_{Nn}(\boldsymbol{\mu}), \quad (A_N^q)_{mn} = a^q(\xi_n, \xi_m), \quad (F_N^q)_m = f^{q'}(\xi_m), \tag{1.19}$$

for $m, n = 1, ..., N$. The matrices A_N^q and $F_N^{q'}$ do not depend on parameter $\boldsymbol{\mu}$ and hence can be computed once and stored during the *offline* stage. Recall that ξ_n belongs to V_h for $n = 1, .., N$, therefore in order to compute the matrices A_N^q and $F_N^{q'}$ we proceed as:

$$\xi_m = \sum_{i=1}^{\mathcal{N}} \xi_m \phi_i, \quad 1 \leq m \leq N, \tag{1.20}$$

where $\mathcal{N}$ is the *truth* space dimension with basis $\{\phi_i\}_{i=1}^{\mathcal{N}}$. Let $\mathcal{Z}$ denote the $\mathcal{N} \times N$ matrix whose columns are the coordinates of $\xi_1, ..., \xi_N$ with respect to $\{\phi_i\}_{i=1}^{\mathcal{N}}$, we have

$$\begin{aligned} A_N^q &= \mathcal{Z}^T A_{\mathcal{N}}^q \mathcal{Z}, \quad 1 \leq q \leq Q_a, \\ F_N^{q'} &= \mathcal{Z}^T F_{\mathcal{N}}^{q'}, \quad 1 \leq q' \leq Q_F, \end{aligned} \tag{1.21}$$

where

$$(A^q_{\mathcal{N}})_{ij} = a^q(\phi_j, \phi_i), \quad (F^{q'}_{\mathcal{N}})_i = F(\phi_i). \tag{1.22}$$

The computation and storage of all the quantities which are $\boldsymbol{\mu}-$independent is done once in the *offline* stage. In this stage we compute and store:

- FE stiffness matrices $A^q_{\mathcal{N}}$, for $q = 1, ..., N_{max}$, and FE right-hand side terms $F^{q'}_{\mathcal{N}}$ for $q' = 1, ..., N_{max}$;
- *snapshots* solutions and the corresponding orthonormal basis $\{\xi_n\}_{m=1}^{N_{max}}$;
- RB matrices A^q_N, for $q = 1, ..., N$ and RB right hand side terms $F^{q'}_N$, for $q' = 1, ..., N$.

One of the main feature of RB method is to obtain a fast approximation of $u_h(\boldsymbol{\mu})$ by giving a new parameter value $\boldsymbol{\mu} \in \mathbb{P}$. We need to evaluate the coefficients $\Theta^q_a(\boldsymbol{\mu})$ and $\Theta^q_F(\boldsymbol{\mu})$ in order to assemble the $N \times N$ system in (1.18). Once this system has been solved, the RB solution is obtained through the relation (1.16). The operations done to perform the evaluation $u_N(\boldsymbol{\mu})$ is called the *online stage.*

Let us analyze the computational cost of the *online stage.* We start with the system (1.18), to assemble the matrix and the right hand side we need the cost of $O(Q_a N^2) + O(Q_F N)$, whereas we need $O(N^3)$ to solve this system using a direct method. Finally we need $O(N \times \mathcal{N})$ operations for the product in (1.15) to get the RB solution.

1.2.3 Greedy basis generation

In this section, we discuss the greedy algorithm for the construction of the RB space (1.13). This algorithm was introduced in early 1970s [45] for optimization problems. The greedy algorithm has been successfully employed by various researchers [95, 96, 113] to construct the RB space. Algebraic and exponential convergence of the greedy algorithm is studied by Binev et al. [19] and Buffa et al. [31], respectively.

We consider a training set Ξ_{train} as a finite subset of $\mathbb{P}$, with cardinality $|\Xi_{train}| = M$. The cardinality M is assumed to be large enough so that Ξ_{train} is a good approximation of the parameter space $\mathbb{P}$. The greedy procedure requires a sharp and computationally inexpensive a *posteriori* error estimator $\boldsymbol{\mu} \longrightarrow \Delta_N(\boldsymbol{\mu})$, i.e,

$$|||u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathbb{P}, \quad 1 \leq N \leq N_{max} \tag{1.23}$$

We will discuss this error estimator in more detail during the section 1.2.5. Here we will briefly explain the greedy algorithm which is summarized in Algorithm 1.

For the initialization of greedy algorithm, we need some initial steps to setup the initial reduced basis space (line 1-5), then each step of greedy algorithm is based on two sub-steps: first, find the value $\boldsymbol{\mu}^N \in \Xi_{train}$ for which the estimator $\Delta_N(\boldsymbol{\mu})$ is maximized (line 8), then compute the *truth* solution $u_h(\bar{\boldsymbol{\mu}})$ (line 10) and add it to the Lagrange basis (line 11). Continuing in this way, at the $(N+1) - th$ (line 7) iteration, we are adding to the already chosen N-basis the solution that is worst approximated by the

Galerkin projection onto V_N. The algorithm stops when the maximum estimated error is less then a prescribed tolerance *tol* (line 6). We also introduce a secondary stopping criteria by setting N_{max} as the maximum number of basis we are interested in. If the tolerance has been obtained with a number of basis N which is less than N_{max} (line 6), we set $N_{max} = N$. Implementation of greedy algorithm is as it follows:

Data: *tol*, $\boldsymbol{\mu}^1, N_{max}$
Result: RB space V_N
1. $N = 1$;
2. $S_1 = \{\boldsymbol{\mu}^1\}$;
3. compute $u_h(\boldsymbol{\mu}^1)$;
4. $V_1 = \text{span}\{u_h(\boldsymbol{\mu}^1)\}$;
5. compute $\Delta_1(\boldsymbol{\mu}^1)$;
6. **while** $\Delta_N(\boldsymbol{\mu}^N) > tol$ *and* $N < N_{max}$ **do**
   7. $N = N + 1$; compute $\Delta_N(\boldsymbol{\mu}) \ \forall \ \boldsymbol{\mu} \in \Xi_{train}$;
   8. $\boldsymbol{\mu}^N := \arg\max_{\boldsymbol{\mu}\in\Xi_{train}} \Delta_{N-1}(\boldsymbol{\mu})$;
   9. $S_N = S_{N-1} \cup \{\boldsymbol{\mu}^N\}$;
   10. compute $u_h(\boldsymbol{\mu}^N)$;
   11. $V_N = V_{N-1} \oplus \{u_h(\boldsymbol{\mu}^N)\}$;
end

Algorithm 1: Greedy Algorithm

1.2.4 Proper orthogonal decomposition (POD)

In this section, we introduce the Proper Orthogonal Decomposition (POD) technique for the construction of reduced spaces of problem (1.13). The POD is an explore-and-compress strategy in which one samples the parameter space, compute the corresponding *truth* solutions at all sample points and, following compression, retains only the essential information.

POD was successfully applied in a variety of fields including fluid dynamics and coherent structures [17, 33, 65, 84, 133, 145], control theory [4, 12, 83, 118].

With the POD algorithm, we obtain the $N-$dimensional POD space that minimizes the quantity

$$\sqrt{\frac{1}{M}\sum_{\boldsymbol{\mu}\in\Xi_{train}} \inf_{u_N\in V_N} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V^2}, \tag{1.24}$$

over all $N-$dimensional subspaces V_N of the span $V_{\mathcal{M}}$=span$\{u_h(\boldsymbol{\mu})|\boldsymbol{\mu} \in \Xi_{train}\}$ of the elements of manifold $\mathcal{M}_h$=$\{u_h(\boldsymbol{\mu})|\boldsymbol{\mu} \in \Xi_{train}\}$.

For the construction of POD reduced space, we introduce an ordering of parameter values $\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^M$ in Ξ_{train}, hence including an ordering $u_h(\boldsymbol{\mu}^1), ..., u_h(\boldsymbol{\mu}^M)$ of the elements of $\mathcal{M}_h$.

To construct the POD-space, we define the correlation matrix,

$$\boldsymbol{C}_{ij} = \frac{1}{M}\left(u_h(\boldsymbol{\mu}^i), u_h(\boldsymbol{\mu}^j)\right)_V, \qquad 1 \le i, j \le M, \tag{1.25}$$

and we consider eigenvalue-eigenfunction pairs $(\lambda_n, \boldsymbol{v}_n) \in \mathbb{R} \times \mathbb{R}^M$ corresponding to the correlation matrix (1.25) satisfying

$$\boldsymbol{C}\boldsymbol{v}_n = \lambda_n \boldsymbol{v}_n, \qquad 1 \leq n \leq M. \tag{1.26}$$

We assume that the positive real eigenvalues are sorted in descending order $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M > 0$. The orthogonal POD basis functions are given by the eigenfunctions $\xi_1, ..., \xi_M$ and they span $V_{\mathcal{M}}$. If we truncate the basis and only consider the first N functions $\xi_1, ..., \xi_N$, $N \leq M$, we define the POD modes that gives us the reduced POD space V_N=span$\{\xi_1, ..., \xi_N\}$ which satisfies the optimality condition (1.24). The orthogonal POD basis functions (POD modes) are defined as

$$\xi_n = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} (\boldsymbol{v}_n)_m u_h(\boldsymbol{\mu}^m), \qquad 1 \leq n \leq N, \tag{1.27}$$

where $(\boldsymbol{v}_n)_m$ denotes the m-th coefficient of the eigenvector $\boldsymbol{v}_n \in \mathbb{R}^M$. We fix a tolerance and then the value of N is chosen according to the the relation

$$\frac{\sum_{m=1}^{N} \lambda_m}{\sum_{m=1}^{M} \lambda_m} \geq 1 - \epsilon_{POD}, \tag{1.28}$$

where the tolerance ϵ_{POD} indicates that the energy carried by the last $M - N$ eigenmodes is less or equal than ϵ_{POD}.

1.2.5 A *posteriori* error estimates

A *posteriori* error estimation is one of the most important feature of the RB method. Indeed, in section (1.2.3) we have seen that the error estimators Δ_N, $N = 1, ..., N_{max}$, play a crucial role in the construction of the RB space. A good a *posteriori* error estimator have to fulfill the following characteristics:

- It has to be a *rigorous*, in the sense that the inequality

  $$|||u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N(\boldsymbol{\mu})$$

  must hold for all parameter values $\boldsymbol{\mu} \in \mathbb{P}$. It is the fundamental requirement to ensure reliability for the RB method.

- It has to be *sharp*. An overly conservative error bound can yield inefficient approximations (N too large).

- It has to be computationally *efficient*. The *online* operation count and storage to compute the RB error bounds - the marginal average cost - must be independent of $\mathcal{N}$ (and commensurate with the cost associated with the RB output prediction) [122]

Some preliminaries

Before defining the error bounds, we first review some notations. Let us consider a *posteriori* error bounds [129] for the field variable $u_N(\boldsymbol{\mu})$ in (1.14). We introduce two basic ingredients of our error bounds: the error residual relationship and the coercivity lower bounds. We observe that the error[1] $e(\boldsymbol{\mu}) := u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}) \in V_h$ satisfies for all $v_h \in V_h$:

$$a(e(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = a(u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = F(v_h; \boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}), \tag{1.29}$$

therefore we can define the RB residual $r(v_h; \boldsymbol{\mu}) \in (V_h)'$ (dual space to V_h)

$$r(v_h; \boldsymbol{\mu}) = F(v_h; \boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h. \tag{1.30}$$

We know that $r(v_h; \boldsymbol{\mu})$ is a continuous linear functional over the space V_h therefore we can apply the *Riesz representation* theorem to get $\hat{r}(\boldsymbol{\mu}) \in V_h$ such that:

$$r(v_h; \boldsymbol{\mu}) = (\hat{r}(\boldsymbol{\mu}), v_h)_V, \quad \forall v_h \in V_h, \tag{1.31}$$

and

$$\|\hat{r}(\boldsymbol{\mu})\|_V = \|r(.; \boldsymbol{\mu})\|_{(V_h)'} = \sup_{v_h \in V_h} \frac{r(v_h; \boldsymbol{\mu})}{\|v_h\|}. \tag{1.32}$$

Now, let us define the error between the FE solution and the RB solution. As we know that

$$|||e(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} = a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) \geq \alpha_h \|e(\boldsymbol{\mu})\|^2_V \quad \text{i.e.} \quad \|e(\boldsymbol{\mu})\|_V \leq \frac{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}}{\sqrt{\alpha_h}}, \tag{1.33}$$

and

$$|||e(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} = a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) = r(e(\boldsymbol{\mu})) \leq \|\hat{r}(\boldsymbol{\mu})\|_V \|e(\boldsymbol{\mu})\|_V \leq \frac{\|\hat{r}(\boldsymbol{\mu})\|_V}{\sqrt{\alpha_h}} |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}, \tag{1.34}$$

from which we arrive at the first estimator

$$|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \frac{\|\hat{r}(\boldsymbol{\mu})\|_V}{\sqrt{\alpha_h(\boldsymbol{\mu})}}. \tag{1.35}$$

Now we introduce a positive parametric lower bound function $\alpha_h^{LB}(\boldsymbol{\mu})$ for the coercivity constant α_h such that:

$$0 \leq \alpha_h^{LB}(\boldsymbol{\mu}) \leq \alpha_h(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathbb{P}, \tag{1.36}$$

where the *online* computational time to evaluate $\boldsymbol{\mu} \longrightarrow \alpha_h^{LB}(\boldsymbol{\mu})$ is independent of $\mathcal{N}$. An efficient algorithm for the computation of $\alpha_h^{LB}(\boldsymbol{\mu})$ is given by the well-known Successive Constraint Method (SCM), widely analyzed in [73, 76, 129]. Moreover, the SCM algorithm which is based on the successive solution of suitable linear optimization problems has been developed for the special requirements of the RB method; it thus features an efficient *offline-online* strategy, making the *online* calculation complexity independent of $\mathcal{N}$.

[1] $e(\boldsymbol{\mu})$ depends on both $\mathcal{N}$ and N

Error bounds

Now we are ready to define our a *posteriori* error estimator. The estimator for the energy norm of the error is:

$$\Delta_N(\boldsymbol{\mu}) := \frac{\|\hat{r}(\boldsymbol{\mu})\|_V}{\sqrt{\alpha_h^{LB}(\boldsymbol{\mu})}}. \tag{1.37}$$

In order to quantify the *sharpness* of this error estimator, we introduce the associated effectivity as:

$$\eta_N(\boldsymbol{\mu}) := \frac{\Delta_N(\boldsymbol{\mu})}{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}}. \tag{1.38}$$

We can now prove the following result [63, 129].

Proposition 1.2.1. *For any $N = 1, ..., N_{max}$ and for any $\boldsymbol{\mu} \in \mathbb{P}$, the effectivity satisfies*

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})}}. \tag{1.39}$$

Proof: The inequality given by (1.35) which corresponds to $\eta_N(\boldsymbol{\mu}) \geq 1$ states that our estimator is *rigorous*. We recall (1.30) and the continuity assumption to show that

$$\begin{aligned} |||\hat{r}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2 = a(\hat{r}(\boldsymbol{\mu}), \hat{r}(\boldsymbol{\mu}); \boldsymbol{\mu}) \leq \gamma \|\hat{r}(\boldsymbol{\mu})\|_V^2 = \gamma(\boldsymbol{\mu}) a(e(\boldsymbol{\mu}), \hat{r}(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ \leq \gamma(\boldsymbol{\mu}) |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} |||\hat{r}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}, \end{aligned} \tag{1.40}$$

now we use this result together with the definition of error estimator (1.37) and *Cauchy-Schwarz* inequality to get

$$\Delta_N^2(\boldsymbol{\mu}) = \frac{\|\hat{r}(\boldsymbol{\mu})\|_V^2}{\alpha_h^{LB}(\boldsymbol{\mu})} = \frac{a(e(\boldsymbol{\mu}), \hat{r}(\boldsymbol{\mu}); \boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})} \leq \frac{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} |||\hat{r}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}}{\alpha_h^{LB}(\boldsymbol{\mu})} \leq \frac{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2 \gamma(\boldsymbol{\mu})}{\alpha_h^{LB}}, \tag{1.41}$$

from which we get finally

$$\eta_N(\boldsymbol{\mu}) = \frac{\Delta_N(\boldsymbol{\mu})}{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}} \leq \sqrt{\frac{\gamma(\boldsymbol{\mu})}{\alpha_h^{LB}(\boldsymbol{\mu})}}. \tag{1.42}$$

1.2.6 POD-greedy sampling

In this section, we discuss the POD-greedy sampling that will be used in Chapter 2 and Chapter 3 for the construction of RB spaces in time dependent problems.

The POD-greedy strategy was proposed in [56] and has been used [103, 104] to construct the RB spaces for the time-dependent problems. This sampling technique is the combination of proper orthogonal decomposition (POD) in time with greedy approach in parameter space [46, 56]. Convergence results for the greedy algorithm [19, 31] are extended to the POD-greedy algorithm by Haasdonk [55].

We first summarize the basic POD optimality property, applied to a time trajectory: given K elements $u_h^k(\boldsymbol{\mu}) \in V_h, 1 \leq k \leq K$, the procedure $\mathrm{POD}\big(\{u_h^1(\boldsymbol{\mu}), ..., u_h^K(\boldsymbol{\mu})\}, M\big)$, with $M < K$, returns M V-orthonormal functions $\{\xi_m, 1 \leq m \leq M\}$ for which the POD space V_N=span$\{\xi_m, 1 \leq m \leq M\}$ is optimal in the sense that following holds:

$$V_N = \arg \inf_{V_M \subset span\{u_h^k(\boldsymbol{\mu}), 1 \leq k \leq K\}} \left(\frac{1}{K} \sum_{k=1}^{K} \inf_{v \in V_M} \|u_h^k(\boldsymbol{\mu}) - v\|_V^2 \right), \tag{1.43}$$

where V_M denotes a M-dimensional linear subspace of V.

In order to initiate the POD-greedy sampling algorithm, we specify the train sample set Ξ_{train}, a tolerance *tol*, and initial sample point $\boldsymbol{\mu}^1$. The algorithm depends on two suitable integers N_1 and N_2, chosen in such a way that $N_1 \geq N_2$ and N_1 satisfy an internal POD error criterion based on the usual sum of eigenvalues and the *tol*. This method is based on successive greedy cycles, so that at each iteration the new information will always be retained and redundant information rejected.

In POD-greedy algorithm, the greedy algorithm provides the outer algorithm where, for each new selected parameter $\boldsymbol{\mu}^n$, the first N_1 principal components of the time trajectory $u_h^1(\boldsymbol{\mu}^n), ..., u_h^K(\boldsymbol{\mu}^n)$ are recovered (line 1). Then this time trajectory is compressed using a POD and to retain the relevant modes (line 2). In a subsequent step, the existing N-dimensional RB space is enriched with those components to build a new $N + N_2$ dimensional basis (line 3 and 4). The a *posteriori* error estimator is used to define a new sample point that minimizes the estimated error over the training set Ξ_{train}. We assume that $\eta(t^k; \boldsymbol{\mu})$ provides a sharp and inexpensive a *posteriori* error bound for $\|u_h^k(\boldsymbol{\mu}) - u_N^k(\boldsymbol{\mu})\|_V$. We exit the POD-greedy sampling procedure at $N = N_{max}$ for which a prescribed tolerance (line 6 and 7)

$$\max_{\boldsymbol{\mu} \in \Xi_{train}} \eta(t^k; \boldsymbol{\mu}) < tol,$$

is satisfied. The reduced spaces $V_N, 1 \leq N \leq N_{max}$ generated by POD-greedy are always hierarchical and consequently computationally advantageous. The POD-greedy algorithm is summarized as follows:

Data: *tol*, $\boldsymbol{\mu}^1$, N_1 and N_2, $\mathcal{Z} = 0$.
Result: A RB space V_N.
1. Compute $\{u_h^1(\boldsymbol{\mu}^n), ..., u_h^K(\boldsymbol{\mu}^n)\}$;
2. Compress this time trajectory using a POD and retain the relevant modes: $\{\xi_1, ..., \xi_{N_1}\} = \mathrm{POD}\big(\{u_h^1(\boldsymbol{\mu}^n), ..., u_h^K(\boldsymbol{\mu}^n)\}, N_1\big)$;
3. $\mathcal{Z} \leftarrow \{\mathcal{Z}, \{\zeta_1, ..., \zeta_{N_1}\}\}$;
4. Set $N \leftarrow N + N_2$ and compute $\xi_1, ..., \xi_N = \mathrm{POD}(\mathcal{Z}, N)$;
5. $V_N = \mathrm{span}\{\xi_1, ..., \xi_N\}$;
6. $\boldsymbol{\mu}^{n+1}$=arg max $_{\boldsymbol{\mu} \in \Xi_{train}} \eta(t^k; \boldsymbol{\mu})$;
7. If $\eta(t^k; \boldsymbol{\mu}) > tol$, then set $n := n + 1$ and **go to** 1., otherwise **terminate.**

Algorithm 2: POD-greedy algorithm

1.3 Geometrical parametrization

An important feature of the RB method is that, we can use it to solve the problems involving "geometrical" parameters. For instance, RB method is applied to heat and mass transfer problems [43, 50, 128, 131], potential flows [126], micro-fluid dynamics [127], material sciences and linear elasticity [74, 75, 101, 105]. In most of these problems, the geometric parameters are considered in addition to other physical parameters [86, 99, 100, 130].

We define the original problem with notation o in the subscript, posed over the parameter-dependent physical domain $\Omega_o = \Omega_o(\boldsymbol{\mu}) \subset \mathbb{R}^d$. We denote by $V_o(\boldsymbol{\mu})$, a suitable Hilbert space defined on parameter-dependent domain $\Omega_o(\boldsymbol{\mu})$ and define the problem: find $u_o(\boldsymbol{\mu}) \in V_o(\boldsymbol{\mu})$ such that

$$a_o(u_o(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_o(v; \boldsymbol{\mu}), \qquad \forall\, v \in V_o(\boldsymbol{\mu}), \tag{1.44}$$

where $a_o(.,.;\boldsymbol{\mu})$ and $f_o(.,.;\boldsymbol{\mu})$ are respectively the bilinear and linear forms on $V_o(\boldsymbol{\mu})$. Let us choose a particular parameter value $\bar{\boldsymbol{\mu}} \in \mathbb{P}$ and a reference ($\boldsymbol{\mu}$-independent) domain Ω which is the requirement of RB framework to compare and combine discrete solutions. We set the $\Omega = \Omega_o(\bar{\boldsymbol{\mu}})$ as reference domain. We define a suitable parametric mapping $T(.;\boldsymbol{\mu})$ to relate the reference domain Ω with the original domain $\Omega_o(\boldsymbol{\mu})$ such that $\Omega_o(\boldsymbol{\mu}) = T(\Omega;\boldsymbol{\mu})$ and $T(.;\bar{\boldsymbol{\mu}})$ becomes the identity.

We now focus on a particular class of admissible geometries [63, 115, 129]. To build a parametric mapping related to geometrical properties, we introduce a domain decomposition of $\Omega_o(\bar{\boldsymbol{\mu}})$ such that

$$\Omega = \bigcup_{l=1}^{L} \Omega^l, \tag{1.45}$$

where $\Omega^l = \Omega_o^l(\bar{\boldsymbol{\mu}})$, for $l = 1, ..., L$ are mutually nonoverlapping open subdomains of Ω. The geometric input parameters, e.g. thicknesses,lengths, angles or diameters, orientations, allows for the definition of parametric mappings to be done in intuitive way.

We now suppose that the original and the reference subdomains are linked via a mapping $T^l(.,\boldsymbol{\mu}) : \Omega^l \to \Omega_o^l(\boldsymbol{\mu}), 1 \le l \le L$, as

$$\Omega_o^l(\boldsymbol{\mu}) = T^l(\Omega^l;\boldsymbol{\mu}), \qquad 1 \le l \le L.$$

Moreover we assume that for each $\boldsymbol{\mu} \in \mathbb{P}$, these maps are collectively continuous and bijective, such that $T^l(x;\boldsymbol{\mu}) = T^{l'}(x;\boldsymbol{\mu}), \forall x \in \overline{\Omega^l} \cap \overline{\Omega^{l'}}$, for $1 \le l < l' \le L$. We also assume that the maps $T^l(.,\boldsymbol{\mu})$ are affine [129]. Therefore, each local map T^l can be expressed as

$$T^l(x;\boldsymbol{\mu}) = \boldsymbol{G}^l(\boldsymbol{\mu})x + c^l(\boldsymbol{\mu}),$$

where $c^l : \mathbb{P} \to \mathbb{R}^d$, $\boldsymbol{G}^l : \mathbb{P} \to \mathbb{R}^{d\times d}$ are translation vectors and linear transformation matrices, respectively. The linear transformation matrices can enable rotation, scaling and must be invertible. The associated Jacobians of transformations are defined as

$$J^l(\boldsymbol{\mu}) = |\det\left(\boldsymbol{G}^l(\boldsymbol{\mu})\right)|, 1 \leq l \leq L.$$

Under these assumptions for many cases the problem (1.44) can easily be formulated in the form (1.1).

Let us now introduce the geometric parametrization in the operators [63]. Consider the bilinear forms

$$a_o(u_o, v_o; \boldsymbol{\mu}) = \sum_{l=1}^{L} \int_{\Omega_o^l(\boldsymbol{\mu})} D(u)^T \boldsymbol{K}_o^l(\boldsymbol{\mu}) D(v), \tag{1.46}$$

where $D(v) : \Omega \to \mathbb{R}^{d+1}$ is defined by $D(v) = \left[\frac{\partial v}{\partial x_{o1}}, ..., \frac{\partial v}{\partial x_{od}}, v_o\right]^T$, and $\boldsymbol{K}_o^l : \mathbb{P} \to \mathbb{R}^{(d+1)\times(d+1)}, 1 \leq l \leq L$, are prescribed coefficients. Here, $\boldsymbol{K}_o^l$ is a given symmetric positive definite matrix, ensuring the coercivity of bilinear form.

In a similar way, we require that $f_o(.)$ expressed as

$$f_o(v_o; \boldsymbol{\mu}) = \sum_{l=1}^{L} \int_{\Omega_o^l(\boldsymbol{\mu})} F_o^l(\boldsymbol{\mu}) v_o, \tag{1.47}$$

where $F_o^l : \mathbb{P} \to \mathbb{R}$ is prescribed coefficient. By identifying $u(\boldsymbol{\mu}) = u_o(\boldsymbol{\mu}) \circ T(.,;\boldsymbol{\mu})$ and tracing (1.46) back to the reference domain Ω by the mapping $T(.,;\boldsymbol{\mu})$, we can define a transformed bilinear form $a(.,.;\boldsymbol{\mu})$ as

$$a(u, v; \boldsymbol{\mu}) = \sum_{l=1}^{L} \int_{\Omega^l(\boldsymbol{\mu})} D(u)^T \boldsymbol{K}^l(\boldsymbol{\mu}) D(v), \tag{1.48}$$

where $\boldsymbol{K}^l : \mathbb{P} \to \mathbb{R}^{(d+1)\times(d+1)}, 1 \leq l \leq L$, is a parametrized tensor defined as

$$\boldsymbol{K}^l(\boldsymbol{\mu}) = J^l(\boldsymbol{\mu}) \hat{\boldsymbol{G}}^{\boldsymbol{l}}(\boldsymbol{\mu}) \boldsymbol{K}_o^l(\boldsymbol{\mu})) (\hat{\boldsymbol{G}}^{\boldsymbol{l}}(\boldsymbol{\mu}))^T$$

and $\hat{\boldsymbol{G}}^{\boldsymbol{l}} : \mathbb{P} \to \mathbb{R}^{(d+1)\times(d+1)}$ is given by

$$\hat{\boldsymbol{G}}^{\boldsymbol{l}}(\boldsymbol{\mu}) = \begin{pmatrix} (\boldsymbol{G}^l(\boldsymbol{\mu}))^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \quad 1 \leq l \leq L.$$

Similarly the transformed linear form can be expressed as

$$f(v; \boldsymbol{\mu}) = \sum_{l=1}^{L} \int_{\Omega^l} F^l(\boldsymbol{\mu}) v, \tag{1.49}$$

where $F^l : \mathbb{P} \to \mathbb{R}$ is defined as $F^l = J^l(\boldsymbol{\mu}) F_o^l(\boldsymbol{\mu})$, for $1 \leq l \leq L$. Now in this setting, the problem on original domain Ω_o has been recast on the reference domain Ω, which give us a parametrized problem where the parametrized transformation tensors expresses the effect of geometry variation. With all above definitions, we can now define the following problem which is equivalent to (1.44). Find $u(\boldsymbol{\mu}) \in V = V_o(\bar{\boldsymbol{\mu}})$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in V. \tag{1.50}$$

There are many ways through which the given assumptions can be relaxed and we can treat an even broader class problems. For instance, we may consider curved triangular subdomains [129].

We mention here that for notation purposes, in this chapter we have only introduced the RB for elliptic case. For the Stokes and Navier-Stokes problem, RB will be introduced in Chapter 2 and Chapter 3, respectively.

1.4 Motivation and objective of the thesis

In this section we present briefly the motivation and the final goal of the thesis. From theory of FE [117], we know that the Galerkin FE for the Stokes and the Navier-Stokes problems is subject to the fulfillment of the discrete inf-sup condition. For a successful method, the discrete spaces for the velocity and pressure should be chosen properly. The discrete inf-sup constant (will be introduced in Chapter 2) is independent of mesh properties, such as the size and shape of the elements.

Similarly the RB method for the Stokes [89] and Navier-Stokes [88] problems also requires the fulfillment of discrete inf-sup condition for reduced velocity and pressure spaces, respectively. For this reason Rovas [121] in his PhD thesis introduced a supremizer enrichment approach to fulfill the inf-sup condition for RB spaces. Rozza et al. [123, 132] also studied the supremizer enrichment approach. This approach consist in the introduction of the inner pressure supremizer for the velocity-pressure stability of the RB spaces. Several works on RB method for Stokes problem [102, 127] were developed using the pressure stabilization via the inner pressure supremizer operator. This approach was then extended to the incompressible Navier-Stokes problems [8, 41, 43, 98, 144], taking into consideration the Brezzi-Rappaz-Raviart (BRR) a *posteriori* error theory [28]. Further detail about the supremizer stabilization approach will be discuss in the Chapter 2.

However, in the supremizer enrichment approach, the RB velocity approximation space is enriched with the solutions of supremizer problem which is expensive in the sense that it increases the computational time in the online phase and, hence, the speedup. Therefore, the main motivation of this PhD thesis is to look for an alternative approach which is computationally less expensive in order to guarantee the stable RB solution.

The stabilization approach developed in this thesis is an alternative approach in which the residual based classical stabilization techniques [30, 68] already known for FE methods are implemented in a reduced order setting [2]. We compare and combine this approach with the existing supremizer approach. The idea of this particular stabilization approach follows the work of Pacciarini and Rozza [107] and Torlo et al. [143] for advection-dominated problem, where they introduced the concept of *offline-online stabilization* and the *offline-only stabilization*

The *offline-online stabilization* method is based on performing the Galerkin projections in both *offline* and *online* stage with respect to the consistent stabilized formulations. On the other hand, *offline-only stabilization* consists in using the stabilized formulations only during the *offline* stage and then projecting with respect to the stan-

dard formulation during the *online* stage.

In this thesis we aim to build a stable RB method [2] suitable for the approximation of parametrized viscous flows. We also guarantee the *online* computational savings by reducing the dimension of the *online* RB system, i.e, we show that with this approach it is possible to get the stable RB solution without the supremizer enrichment into velocity space. We extend the idea of *offline-online stabilization* [107] to the parametric Stokes and Navier-Stokes problems by introducing a generalize inf-sup condition (see Chapter 2) and we compare our approach with supremizer approach [132].

We point out that in thesis we are not dealing with the instabilities of RB solutions caused by the dominating transport field. For these kind of instabilities we refer to some recent works by the authors in [5, 7, 52, 147].

Further detailed implementation of this stabilization approach to steady [2] and unsteady [3] problems, and its comparison with the supremizer enrichment [132] will be discussed with numerical experiments in Chapter 2 and 3, respectively, for parametric Stokes and Navier-Stokes problems.

Chapter 2

Reduced Basis Stabilization of Parametrized Stokes Problem

We know that when dealing with Stokes [89] or Navier-Stokes [88] problems, the Galerkin projection on reduced basis (RB) space does not guarantee the fulfillment of discrete inf-sup condition for reduced velocity and pressure spaces. Therefore, our goal in this chapter is to develop and discuss the reduced basis methods which ensure a stable RB solution. In particular, we apply the residual based stabilization methods using equal order ($\mathbb{P}_k/\mathbb{P}_k$) velocity/pressure interpolation, first in the high order level and then project onto reduced spaces. By stabilizing the *offline* stage we are able to get the stable bases functions to construct the reduced basis space, but this does not guarantee a stable RB solution. Therefore we also properly stabilize the *online* stage. The process of stabilizing both the *offline* and *online* stages is named as *offline-online stabilization* [107], while when we stabilize only the *offline* stage, it is called *offline-only stabilization*. We compare the *offline-online stabilization* approach with the supremizer enrichment approach [132] through several numerical experiments. We want to show if the RB stabilization that we propose in this thesis is enough to guarantee the stable RB solution or we need also a supremizer enrichment for RB velocity spaces.

This chapter is divided into two parts. In the first part (section 2.1), we solve numerically the steady Stokes problem [2] in a parametrized domain. First we start defining the finite element (FE) formulation followed by the stabilized FE formulation, and then, by selecting the snapshots using greedy algorithm [63], we obtain the RB spaces for velocity and pressure. For these low-dimensional spaces we define the RB model and then, the stabilized RB model. Finally in section 2.2 we show some numerical solutions and error analysis to compare different stabilization options.

In the second part of the chapter (section 2.3) we present the stabilized reduced basis method for unsteady parametrized Stokes problem [3]. We repeat the same procedure as we did for the steady case. In this case we use POD-Greedy algorithm [63] to select the snapshots for the construction of RB spaces for velocity and pressure. Again we compare the *offline-online stabilization* approach with the the supremizer enrichment approach through numerical results in section 2.4. We also present a test case for the

lowest order ($\mathbb{P}_1/\mathbb{P}_0$) element and the sensitivity analysis on the time step Δt. Finally we summarize the main findings of this chapter in section 2.5.

2.1 Parametrized steady Stokes problem

The steady Stokes problem in a two-dimensional parametrized domain $\Omega_0(\boldsymbol{\mu}) \subset \mathbb{R}^2$ read as: find $(\boldsymbol{u}_o, p_o) \in \boldsymbol{V} \times Q$ such that

$$\begin{cases} -\nu\Delta \boldsymbol{u}_o + \nabla p_o = \boldsymbol{f} & \text{in } \Omega_o(\boldsymbol{\mu}), \\ \text{div}\boldsymbol{u}_\text{o} = 0 & \text{in } \Omega_o(\boldsymbol{\mu}), \\ \boldsymbol{u}_o = \boldsymbol{g} & \text{on } \partial\Omega_o, \end{cases} \tag{2.1}$$

where $\boldsymbol{u}_o$ is the unknown velocity and p_o is the unknown pressure, $\boldsymbol{f}$ is a given forcing function and ν is the viscosity of fluid, $\boldsymbol{\mu} \in \mathbb{P}$ (parameter domain) denotes a parameter which may be physical or geometrical. For the sake of simplicity we take $\boldsymbol{f} = \boldsymbol{0}$. The boundary $\partial\Omega_o$ is divided into two parts in such a way that $\partial\Omega_o = \Gamma_{D_0} \cup \Gamma_{D_g}$, where Γ_{D_g} is the Dirichlet boundary with non-homogeneous data and Γ_{D_0} denotes the Dirichlet boundary with zero data.

In order to write the weak formulation of problem (2.1), we introduce a reference domain, i.e. a $\boldsymbol{\mu}$-independent configuration Ω by assuming that each parametrized domain $\Omega_o(\boldsymbol{\mu})$ can be obtained as the image of $\boldsymbol{\mu}$-independent domain Ω through a parametrized map $\boldsymbol{T}(.;\boldsymbol{\mu}) : \mathbb{R}^2 \to \mathbb{R}^2$, i.e. $\Omega_o(\boldsymbol{\mu}) = \boldsymbol{T}(\Omega;\boldsymbol{\mu})$.

Now the weak formulation of (2.1) can be obtained by multiplying with the velocity, pressure test functions and using integration by parts; then by tracing everything back onto the reference domain Ω, we have the following parametrized weak formulation of problem (2.1):

$$\begin{cases} \text{Find } \boldsymbol{u} \in \boldsymbol{V},\, p \in Q : \\ a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) + b(\boldsymbol{v}, p; \boldsymbol{\mu}) = F(\boldsymbol{v}; \boldsymbol{\mu}) & \forall\, v \in \boldsymbol{V}, \\ b(\boldsymbol{u}, q; \boldsymbol{\mu}) = G(q; \boldsymbol{\mu}) & \forall\, q \in Q, \end{cases} \tag{2.2}$$

where $\boldsymbol{V} = \{\boldsymbol{v} \in H^1(\Omega)^2 : \boldsymbol{v} = 0 \text{ on } \Gamma_{D_0} \text{ and } \boldsymbol{v} = \boldsymbol{g} \text{ on } \Gamma_{D_g}\}$ and $Q = L^2_0(\Omega) = \{q \in L^2(\Omega) : \int_\Omega q = 0\}$. Bilinear forms related to diffusion and pressure-divergence operators are defined as:

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) = \int_\Omega \frac{\partial \boldsymbol{u}}{\partial x_i} \kappa_{ij}(x; \boldsymbol{\mu}) \frac{\partial \boldsymbol{v}}{\partial x_j} d\boldsymbol{x}, \qquad b(\boldsymbol{v}, q; \boldsymbol{\mu}) = -\int_\Omega q \chi_{ij}(x; \boldsymbol{\mu}) \frac{\partial v_j}{\partial x_i} d\boldsymbol{x}. \tag{2.3}$$

The transformation tensors for bilinear viscous and pressure divergence terms in (2.3) are defined as follows:

$$\begin{aligned} \boldsymbol{\kappa}(x; \boldsymbol{\mu}) &= \nu(\boldsymbol{\mu})(J_T(x; \boldsymbol{\mu}))^{-1}(J_T(x; \boldsymbol{\mu}))^{-T}|J_T(X; \boldsymbol{\mu})|, \\ \boldsymbol{\chi}(x; \boldsymbol{\mu}) &= (J_T(x; \boldsymbol{\mu}))^{-1}|J_T(X; \boldsymbol{\mu})|, \end{aligned} \tag{2.4}$$

where $|J_T|$ is the determinant of the Jacobian matrix $J_T \in \mathbb{R}^{2\times 2}$ of the map $T(.;\boldsymbol{\mu})$. F and G are terms due to non-homogeneous Dirichlet boundary condition on the boundary

are defined as:

$$\begin{aligned} F(\boldsymbol{v};\boldsymbol{\mu}) &= -a(\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{v};\boldsymbol{\mu}),\\ G(q;\boldsymbol{\mu}) &= -b(\boldsymbol{l}(\boldsymbol{\mu}),q;\boldsymbol{\mu}), \end{aligned} \tag{2.5}$$

where we denote by $\boldsymbol{l}(\boldsymbol{\mu})$ a parametrized lifting function such that $\boldsymbol{l}(\boldsymbol{\mu})|_{\Gamma_{D_g}} = \boldsymbol{g}_D$.

2.1.1 Finite Element formulation

For a given parameter $\boldsymbol{\mu} \in \mathbb{P}$, the Galerkin-FE approximation of the parametrized Stokes problem (2.2) reads as follows:

$$\begin{cases} \text{Find } \boldsymbol{u}_h(\boldsymbol{\mu}) \in \boldsymbol{V}_h, p_h(\boldsymbol{\mu}) \in Q_h : \\ a(\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}) + b(\boldsymbol{v}_h,p_h(\boldsymbol{\mu});\boldsymbol{\mu}) = F(\boldsymbol{v}_h;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h(\boldsymbol{\mu}),q_h;\boldsymbol{\mu}) = G(q_h;\boldsymbol{\mu}) & \forall\, q_h \in Q_h, \end{cases} \tag{2.6}$$

where $\boldsymbol{V}_h$ and Q_h are finite dimensional subspaces of $\boldsymbol{V}$ and Q of dimension $\mathcal{N}_u$ and $\mathcal{N}_p$, respectively with h related to the computational mesh size, $\boldsymbol{l}(\boldsymbol{\mu})$ in (2.1) is discretized by the FE interpolant.

Let $\boldsymbol{\phi}_i^h$ and $\boldsymbol{\psi}_j^h$ be the basis functions of $\boldsymbol{V}_h$ and Q_h respectively. We introduce the matrices $A(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}_u \times \mathcal{N}_u}$ and $B(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}_p \times \mathcal{N}_u}$ whose entries are

$$(A(\boldsymbol{\mu}))_{ij} = a(\boldsymbol{\phi}_j^h,\boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \quad (B(\boldsymbol{\mu}))_{ki} = b(\boldsymbol{\phi}_i^h,\boldsymbol{\psi}_k^h;\boldsymbol{\mu}), \text{ for } 1 \le i,j \le \mathcal{N}_u, 1 \le k \le \mathcal{N}_p, \tag{2.7}$$

and the algebraic form of discrete problem (2.6) problem reads

$$\begin{bmatrix} A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(\boldsymbol{\mu}) \\ \boldsymbol{P}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \tag{2.8}$$

for the vectors $\boldsymbol{U} = (u_h^{(1)},...,u_h^{(\mathcal{N}_u)})^T, \boldsymbol{P} = (p_h^{(1)},...,p_h^{(\mathcal{N}_p)})^T$, where for $1 \le i \le \mathcal{N}_u$ and $1 \le k \le \mathcal{N}_p$:

$$(\bar{\boldsymbol{f}}(\boldsymbol{\mu}))_i = -a(\boldsymbol{l}_h,\boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \quad (\bar{\boldsymbol{g}}(\boldsymbol{\mu}))_k = -b(\boldsymbol{l}_h,\boldsymbol{\psi}_k^h;\boldsymbol{\mu}), \tag{2.9}$$

with $\boldsymbol{l}_h = \boldsymbol{l}_h(\boldsymbol{\mu})$.

For a stable solution the finite element spaces $\boldsymbol{V}_h$ and Q_h have to fulfill the following parametrized version of the stability condition (LBB) [117].

$$\exists \beta_0 > 0 : \beta_h(\boldsymbol{\mu}) = \inf_{q_h \in Q_h} \sup_{\boldsymbol{v}_h \in \boldsymbol{V}_h} \frac{b(\boldsymbol{v}_h,q_h;\boldsymbol{\mu})}{\|\boldsymbol{v}_h\|_{\boldsymbol{V}_h}\|q_h\|_{Q_h}} \ge \beta_0 \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{2.10}$$

This relation holds if, e.g., the Taylor-Hood ($\mathbb{P}_2/\mathbb{P}_1$) FE spaces are chosen. Condition (2.10) do not holds in case of equal order FE spaces ($\mathbb{P}_k/\mathbb{P}_k$), $k \ge 1$ and for lowest order element ($\mathbb{P}_1/\mathbb{P}_0$).

2.1.2 Stabilized Finite Element formulation

When the finite dimensional spaces $\boldsymbol{V}_h$ and Q_h do not satisfy the inf-sup stability condition (2.10), then in order to avoid possible spurious pressure modes we use a stabilization procedure to stabilize the discrete solution. Therefore (2.6) is modified as

$$\begin{cases} \text{Find } \boldsymbol{u}_h(\boldsymbol{\mu}) \in \boldsymbol{V}_h,\ p_h(\boldsymbol{\mu}) \in Q_h : \\ a(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = F(\boldsymbol{v}_h; \boldsymbol{\mu}) + s_h^v(\boldsymbol{v}_h; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h(\boldsymbol{\mu}), q_h; \boldsymbol{\mu}) = G(q_h; \boldsymbol{\mu}) + s_h^q(q_h; \boldsymbol{\mu}) & \forall\, q_h \in Q_h, \end{cases} \tag{2.11}$$

where $s_h^v(\boldsymbol{v}_h; \boldsymbol{\mu})$ and $s_h^q(q_h; \boldsymbol{\mu})$ are the stabilization terms [117]. Dependence of these stabilization terms on parameter $\boldsymbol{\mu}$ is motivated by the fact that we are dealing with the parametrized system (2.8). These stabilization terms are defined as follows

$$s_h^v(\boldsymbol{v}_h; \boldsymbol{\mu}) := \delta \sum_K h_K^2 \int_K (-\nu \Delta \boldsymbol{u}_h + \nabla p_h - \boldsymbol{f}, -\rho\nu \Delta \boldsymbol{v}_h), \tag{2.12}$$

and

$$s_h^q(q_h; \boldsymbol{\mu}) := \delta \sum_K h_K^2 \int_K (-\nu \Delta \boldsymbol{u}_h + \nabla p_h - \boldsymbol{f}, \nabla q_h), \tag{2.13}$$

where K is an element of triangulation τ_h of the domain Ω, h_K is the diameter of element K, δ is the stabilization coefficient such that $0 < \delta \leq C$ (C is suitable constant). For $\rho = 0, 1, -1$, the method (2.12) is respectively known as the pressure-poisson stabilized Galerkin (Franca-Hughes) [68], Galerkin-least-square (GALS) [67], Douglas-Wang (DW) [44]. For simplicity we take $\boldsymbol{f} = \boldsymbol{0}$. In case of linear interpolation for velocity and pressure ($\mathbb{P}_1/\mathbb{P}_1$) the Laplacian term $-\nu\Delta\boldsymbol{u}_h$ inside the stabilization vanishes and all above choices reduce to Brezzi-Pitkãranta stabilization [27] and is written as

$$s_h^q(q_h; \boldsymbol{\mu}) := \delta \sum_K h_K^2 \int_K \nabla p_h \cdot \nabla q_h. \tag{2.14}$$

For a detail study about the possible stabilization options, we refer to [117, chapter 9] and the references therein.

In this thesis, we prefer to chose the stabilization option corresponding to $\rho = 0$ (Franca-Hughes) [68]. Which means that (2.12) vanishes and we are left with the stabilization term (2.13) only. Further investigations regarding the coercivity of bilinear forms in [68] and the error estimates are given by Brezzi and Douglas [26]. Another modification of [68] was done by Hughes and Franca [67].

The motivation to choose a *residual method* is, it improve the stability of Galerkin FE method without compromising the consistency. This choice also allows us to fulfill the LBB condition (2.10) with a supplementary terms.

In practice, the stabilization coefficient δ has to be chosen properly. It cannot be too small otherwise the stabilization will be poor and spurious modes will not be eliminated. On the other hand, a large value of parameter δ could results a poor approximation for

the pressure field near to the boundary. For possible choices of stabilization coefficient for different stabilization methods we refer to [80] and references therein.

When we use the stabilization method we have to fulfill a generalized inf-sup condition

$$\sup_{\boldsymbol{v}_h \in \boldsymbol{V}_h} \frac{b(\boldsymbol{v}_h, q_h; \boldsymbol{\mu})}{\|\nabla \boldsymbol{v}_h\|} + s_h^q(q_h; \boldsymbol{\mu}) \geq \beta_0 \|q_h\| > 0, \forall q_h \in Q_h, \tag{2.15}$$

where the term $s_h^q(q_h; \boldsymbol{\mu})$ is the additional stabilization term (2.14). This condition is the crucial part of thesis and will be discussed in detail at reduced order level in subsection 2.1.4.

After adding the stabilization effects into the system (2.8), the stabilized algebraic formulation reads

$$\begin{bmatrix} A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ \tilde{B}(\boldsymbol{\mu}) & -S(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(\boldsymbol{\mu}) \\ \boldsymbol{P}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}, \tag{2.16}$$

where $S_{ij} = s_h^q(\boldsymbol{\psi}_j^h, \boldsymbol{\psi}_i^h)$ and $\tilde{B}(\boldsymbol{\mu})$ contains the effects of stabilization terms.

2.1.3 Reduced Basis formulation

In this section we present the reduced basis formulation of steady Stokes problem. We take the parameter sample $s_N = \{\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^N\}$, where $\boldsymbol{\mu}^n \in \mathbb{P}$ and we solve the Stokes problem N times using standard Galerkin-Finite element method to obtain snapshots for velocity and pressure $((\boldsymbol{u}_h(\boldsymbol{\mu}^1), p_h(\boldsymbol{\mu}^1)), ..., (\boldsymbol{u}_h(\boldsymbol{\mu}^N), p_h(\boldsymbol{\mu}^N)))$. These snapshots are selected in a proper way using the greedy algorithm [127].

We define the reduced basis velocity space $\boldsymbol{V}_N \subset \boldsymbol{V}_h$ and pressure space $Q_N \subset Q_h$, respectively as:

$$\boldsymbol{V}_N = \text{span}\left\{\xi_n^u = \boldsymbol{u}_h(\boldsymbol{\mu}^n), 1 \leq n \leq N_u\right\}, \tag{2.17}$$

and

$$Q_N = \text{span}\left\{\xi_n^p = p_h(\boldsymbol{\mu}^n), 1 \leq n \leq N_p\right\}, \tag{2.18}$$

where N_u and N_p are the dimensions of RB velocity space $\boldsymbol{V}_N$ and RB pressure space Q_N, respectively. $\{\xi_n^u\}_{n=1}^{N_u}$ and $\{\xi_n^p\}_{n=1}^{N_p}$ are mutually orthonormal basis functions for RB velocity and pressure, respectively obtained by applying the Gram-Schmidt orthogonalization process [63]. In order to guarantee the inf-sup of the RB approximation, we enrich the RB velocity space with supremizer solutions [132]. Thus, we introduce the supremizer operator $T^{\boldsymbol{\mu}} : Q_h \to \boldsymbol{V}_h$ defined as follows:

$$(T^{\boldsymbol{\mu}} q, \boldsymbol{v}_h)_{\boldsymbol{V}} = b(\boldsymbol{v}_h, q_h; \boldsymbol{\mu}), \quad \forall \boldsymbol{v} \in \boldsymbol{V}_h. \tag{2.19}$$

Solution of the reduced system is subject to the fulfillment of an equivalent inf-sup condition (2.10) for reduced velocity and pressure spaces $\boldsymbol{V}_N$ and Q_N, respectively. Induced inf-sup condition for reduced order problem is:

$$\exists \tilde{\beta}_N > 0 : \quad \beta_N(\boldsymbol{\mu}) = \inf_{q_N \in Q_N} \sup_{\boldsymbol{v}_N \in \boldsymbol{V}_N} \frac{b(\boldsymbol{v}_N, q_N; \boldsymbol{\mu})}{\|\boldsymbol{v}_N\|_{\boldsymbol{V}_N} \|q_N\|_{Q_N}} \geq \tilde{\beta}_N \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{2.20}$$

It is important to point out that, even though the velocity basis functions are obtained through a stable full order model, a Galerkin projection over the reduced spaces does not guarantee the fulfillment of the reduced inf-sup condition (2.20).

In order to fulfill the reduced order inf-sup condition we solve a supremizer problem (2.19) to enrich the RB velocity space with additional velocities. Since our focus is not the supremizer enrichment option, therefore we skip the details about supremizer and refer the readers to [121, 123, 132] for different supremizer options.

When the RB velocity space is enriched with the supremizer solutions (2.19), we denote in this case the RB velocity space by $\tilde{\boldsymbol{V}}_N$ with dimension $N_u + N_s$ and is defined as:

$$\tilde{\boldsymbol{V}}_N = \text{span}\left\{\xi_n^u, 1 \leq n \leq N_u; T^{\boldsymbol{\mu}}\xi_n^s, 1 \leq n \leq N_s\right\}, \tag{2.21}$$

where N_s denotes the dimension of supremizer space. In order to keep the notations simple, we take $N_u = N_s = N_p = N$. Now the RB formulation corresponding to FE formulation (2.6) can be written as:

$$\begin{cases} \text{Find } (\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \tilde{\boldsymbol{V}}_N \times Q_N : \\ a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N(\boldsymbol{\mu})) = F(\boldsymbol{v}_N; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \tilde{\boldsymbol{V}}_N, \\ b(\boldsymbol{u}_N(\boldsymbol{\mu}), q_N; \boldsymbol{\mu}) = G(q_N; \boldsymbol{\mu}) & \forall\, q_N \in Q_N. \end{cases} \tag{2.22}$$

The solution $(\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \tilde{\boldsymbol{V}}_N \times Q_N$ of (2.22) can be expressed as a linear combination of the basis functions:

$$\boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{n=1}^{2N} \boldsymbol{u}_{Nn}(\boldsymbol{\mu})\xi_n^u, \qquad p_N(\boldsymbol{\mu}) = \sum_{n=1}^{N} p_{Nn}(\boldsymbol{\mu})\xi_n^p, \tag{2.23}$$

where $\{\boldsymbol{u}_{Nn}(\boldsymbol{\mu})\}_{n=1}^{N_u}$ and $\{p_{Nn}(\boldsymbol{\mu})\}_{n=1}^{N_p}$ denotes the coefficients of the reduced basis approximation for velocity and pressure. Finally, we write the system in compact form as

$$\begin{bmatrix} A_N(\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ B_N(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(\boldsymbol{\mu}) \\ \boldsymbol{P}_N(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix} \tag{2.24}$$

where the RB matrices are computed as

$$\begin{aligned} &A_N(\boldsymbol{\mu}) = Z_{u,s}^T A(\boldsymbol{\mu}) Z_{u,s}, \quad B_N(\boldsymbol{\mu}) = Z_p^T B(\boldsymbol{\mu}) Z_{u,s}, \\ &\bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) = Z_{u,s}^T \bar{\boldsymbol{f}}(\boldsymbol{\mu}), \qquad \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) = Z_p^T \bar{\boldsymbol{g}}(\boldsymbol{\mu}), \end{aligned} \tag{2.25}$$

2.1.4 Stabilized Reduced Basis formulation

In this section we present the stabilized Reduced Basis (RB) model derived from the stabilized FE problem (2.11). The computation of the reduced spaces is done through the Greedy algorithm (see algorithm 1.1). The stabilized RB problem reads

$$\begin{cases} \text{Find } (\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \boldsymbol{V}_N \times Q_N : \\ a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N(\boldsymbol{\mu})) = F(\boldsymbol{v}_N; \boldsymbol{\mu}) + s_N^v(\boldsymbol{v}_N; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \boldsymbol{V}_N, \\ b(\boldsymbol{u}_N(\boldsymbol{\mu}), q_N; \boldsymbol{\mu}) = G(q_N; \boldsymbol{\mu}) + s_N^q(q_N; \boldsymbol{\mu}) & \forall\, q_N \in Q_N, \end{cases} \tag{2.26}$$

where $s_N^v(\boldsymbol{v}_N;\boldsymbol{\mu})$ and $s_N^q(q_N;\boldsymbol{\mu})$ are the reduced order counterparts of the stabilization terms defined in (2.12) and (2.13), respectively.

We also define the reduced order version of generalized inf-sup condition (2.15) as:

$$\sup_{\boldsymbol{v}_N\in\boldsymbol{V}_N}\frac{b(\boldsymbol{v}_N,q_N;\boldsymbol{\mu})}{\|\nabla\boldsymbol{v}_N\|}+s_N^q(q_N;\boldsymbol{\mu})\geq\beta_0\|q_N\|>0,\forall q_N\in Q_N, \tag{2.27}$$

where $s_N^q(q_N;\boldsymbol{\mu})$ is due to the addition of stabilization terms in RB consistent formulation.

For simplicity and discussion purpose, we denote

$$\beta_{\inf-\sup}=\sup_{\boldsymbol{v}_N\in\boldsymbol{V}_N}\frac{b(\boldsymbol{v}_N,q_N;\boldsymbol{\mu})}{\|\nabla\boldsymbol{v}_N\|}\quad\text{and}\quad\beta_{stab}=s_N^q(q_N;\boldsymbol{\mu}), \tag{2.28}$$

so that (2.27) becomes

$$\beta_{\inf-\sup}+\beta_{stab}\geq\beta_0\|q_N\|>0,\forall q_N\in Q_N, \tag{2.29}$$

The results presented in this thesis are based on following possible options on (2.27):

- the first option is to increase the "$\beta_{\inf-\sup}$" by enriching the RB velocity space with supremizer solutions and also add "β_{stab}" in both *offline* and *online* stage, which could be computationally expensive; we call this option as *offline-online stabilization* with supremizer
- as a second option, we do not enrich the RB velocity space with supremizer solutions but instead we add only the "*stab*" in both *offline* and *online* stage; we call this option *offline-online stabilization* without supremizer
- the third option is to increase "$\beta_{\inf-\sup}$" by enriching the RB velocity space with supremizer solutions and not adding the "β_{stab}" in *online* stage; which we call *offline-only stabilization* with supremizer
- a fourth option could be to completely avoid the supremizer enrichment in order to increase "$\beta_{\inf-\sup}$" as well as skip "β_{stab}" in *online* stage but add "β_{stab}" in *offline* stage; we call this option as *offline-only stabilization* without supremizer.

We study all these options for equal order FE spaces $(\mathbb{P}_k/\mathbb{P}_k), k\geq 1$. In this thesis we only consider the first three options and we have done several test cases to compare the three options. Fourth option is the worst option because of its lack of stability and is not reported in numerical results presented in this thesis. A further option (and comparison) will be provided in Chapter 4 with rectification approach.

After adding the stabilization effects into the system (2.24), the RB stabilized algebraic system (2.26) in compact form reads

$$\begin{bmatrix} A_N(\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ \tilde{B}_N(\boldsymbol{\mu}) & -S_N(\boldsymbol{\mu}) \end{bmatrix}\begin{bmatrix} \boldsymbol{U}_N(\boldsymbol{\mu}) \\ \boldsymbol{P}_N(\boldsymbol{\mu}) \end{bmatrix}=\begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix}, \tag{2.30}$$

where $\tilde{B}_N(\boldsymbol{\mu})$ includes the effects of stabilization in the mass equation on the divergence term.

The main motivation behind this work is the combination of the supremizer approach by Rozza et al. [132] for inf-sup stability of Stokes problem and the stabilization approach by Pacciarini et al. [107], in which the *offline-online stabilization* method and *offline-only stabilization* method for advection-diffusion problem has been developed.

For the *offline-online stabilization* we have to perform the whole RB standard method solving the stabilized algebraic system (2.16) in the *offline* stage and (2.30) during the *online* stage. The *offline-only stabilized* approach consists in solving the system (2.16) during the *offline* stage, in order to obtain stable reduced basis, and to perform the *online* Galerkin projection with respect to the non-stabilized RB system (2.24).

2.2 Numerical results and discussion

In this section, we present some numerical results for stabilized reduced order model for steady Stokes problem developed in section 2.1 and subsections therein. Numerical simulations are carried out in FreeFem++ [59], and also with RBniCS [10] for comparison. In subsection 2.2.1 we present some numerical results for unstable FE pairs $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$ using classical stabilization method (2.13). In subsection 2.2.2 we show some results for lowest order unstable FE element $\mathbb{P}_1/\mathbb{P}_0$ with stabilization. We also show results for stable FE pair $\mathbb{P}_2/\mathbb{P}_1$ in subsection 2.2.3.

As a test case we consider the parametrized cavity flow problem. We set the parametrized domain $\Omega_o(\boldsymbol{\mu}) = (0, 1+\mu_2) \times (0, 1)$, where we define $\boldsymbol{\mu} = (\mu_1, \mu_2)$ such that μ_1 is physical parameter (kinematic viscosity of fluid) and μ_2 is geometrical parameter (length of domain). Main goal is to see the effect of geometrical parameter on the velocity and pressure. Problem is defined as follows: Find $\boldsymbol{u} \in \boldsymbol{V}_o$ and $p \in Q_o$:

$$\begin{cases} -\mu_1 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{0} & \text{in } \Omega_o(\mu_2), \\ \text{div}\boldsymbol{u} = 0 & \text{in } \Omega_o(\mu_2) \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \Gamma_{D_o}(\mu_2) \\ \boldsymbol{u} = (1, 0) & \text{on } \Gamma_D \end{cases} \tag{2.31}$$

where $\boldsymbol{u}$ and p are unknowns velocity and pressure, respectively. We consider a partition of domain $\partial\Omega_o = \Gamma_{D_o} \cup \Gamma_D$, where we have the homogeneous Dirichlet condition on Γ_{D_o} and non-homogeneous Dirichlet condition on Γ_D. We define the spaces $\boldsymbol{V}_o = H_0^1(\Omega_o)^2$ and $Q_o = L^2(\Omega_o)$.

We omit here the details of writing weak formulations, stabilized formulations and building reduced basis formulations for this particular problem. We refer to section 2.2 for parametrized formulation, subsection 2.1.1 for FE formulation, subsection 2.1.2 for stabilized FE formulation. Similarly, for the development of RB method, we refer to subsection 2.1.3 for the RB formulation and 2.1.4 for stabilized RB formulation. Parametrized domain is shown in Fig. 2.1.

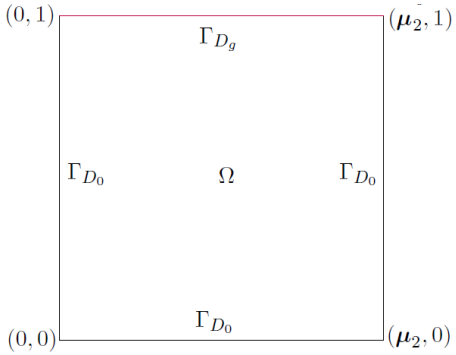


Figure 2.1: Parametrized domain

2.2.1 Numerical results for $\mathbb{P}_k/\mathbb{P}_k$ for $k = 1, 2$

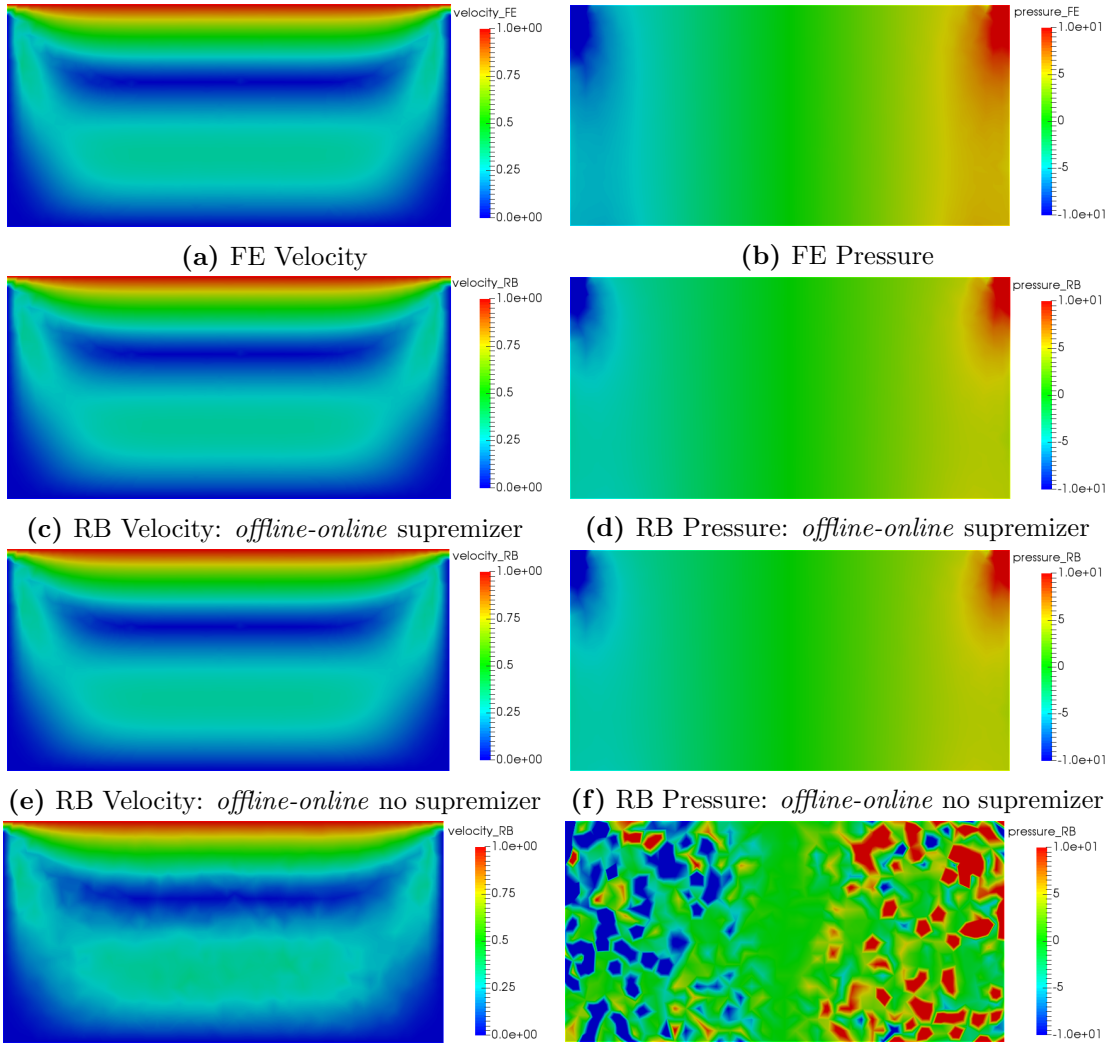
In Fig. 2.2 we show a comparison between the FE velocity solution and the RB solutions obtained for three different options; *(i) offline-online stabilization* with supremizer; *(ii) offline-online stabilization* without supremizer; *(iii) offline-only stabilization* with supremizer, for a chosen parameters value *online* $(\mu_1, \mu_2) = (0.6, 2)$. The range of parameters in the *offline* stage is $\mu_1 \in [0.25, 0.75], \mu_2 \in [1, 3]$, and we use the $\mathbb{P}_2/\mathbb{P}_2$ FE pair to plot these solutions.

From Fig. 2.2, we see that the RB velocity and pressure solutions obtained by using the *offline-online stabilization* with/without supremizer looks similar to the FE solution. However, the RB solutions obtained by the *offline-only stabilization* is poor, in particular pressure solution is highly oscillatory. Results are similar if we chose $\mathbb{P}_1/\mathbb{P}_1$ FE pair for velocity and pressure, therefore we do not report here.

In order to see the comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer we have done an error analysis. Figures 2.3 and 2.4 shows the error comparison for velocity and pressure, respectively using Brezzi-Pitkaranta stabilization. In these plots we have used $\mathbb{P}_1/\mathbb{P}_1$ FE pair for which we mentioned earlier that all other stabilization terms in (2.12) and (2.13) vanishes and we are left with Brezzi-Pitkaranta stabilization (2.14). Parameter values are same as in Fig. 2.2. These results show that *offline-online stabilization* is the best way to stabilize and *offline-only stabilization* is inaccurate. We also see that the enrichment of supremizer together with *offline-online stabilization* is improving the error in case of pressure, however supremizer does not improve the velocity error.

Figures 2.5 and 2.6 are plotted to see the error comparison for velocity and pressure, respectivley using Franca-Hughes stabilization (2.13) for $\mathbb{P}_2/\mathbb{P}_2$ FE pair and by varying the stabilization coefficient δ. If we compare these results with previous results obtained by Brezzi-Pitkaranta stabilization, we see that Franca-Hughes stabilization is able to perform better in case of *offline-online stabilization* without supremizer. Therefore, we conclude that when we use Franca-Hughes stabilization with $\mathbb{P}_2/\mathbb{P}_2$ FE pair, there is no need to enrich the RB velocity space with supremizer solutions. In this way we

can reduce also the online computational cost by decreasing the dimension of reduced velocity space.



(a) FE Velocity **(b)** FE Pressure

(c) RB Velocity: *offline-online* supremizer **(d)** RB Pressure: *offline-online* supremizer

(e) RB Velocity: *offline-online* no supremizer **(f)** RB Pressure: *offline-online* no supremizer

(g) RB Velocity: *offline-only* supremizer **(h)** RB Pressure: *offline-only* supremizer

Figure 2.2: Comparison between the FE solution and RB solutions for velocity and pressure obtained by three possible options, respectively.

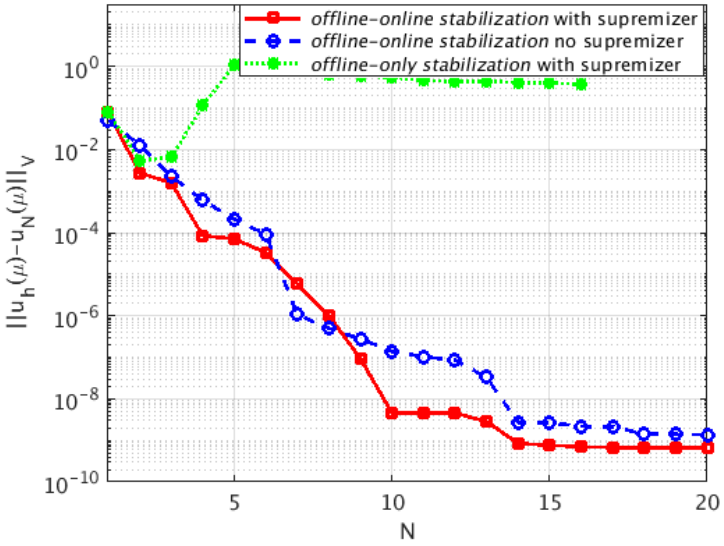


Figure 2.3: Brezzi-Pitkaranta stabilization on cavity flow: Velocity error comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer using $\mathbb{P}_1/\mathbb{P}_1$; stabilization coefficient $\delta = 0.05$; $N_u = N_p = N_s = 20$.

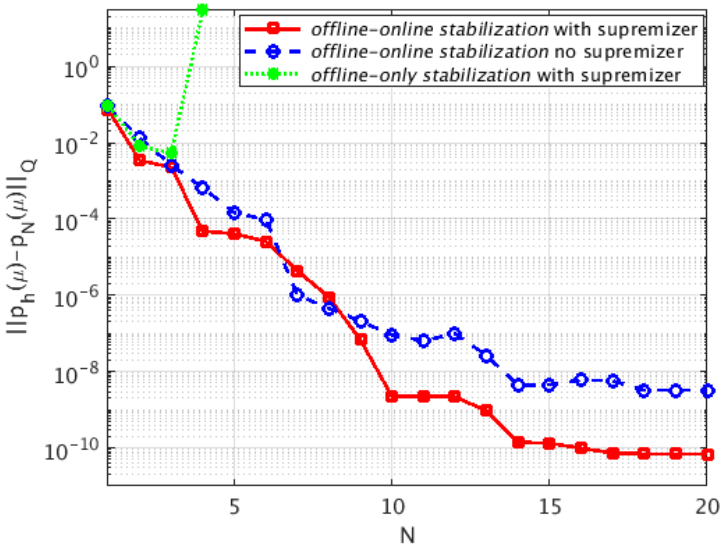


Figure 2.4: Brezzi-Pitkaranta stabilization on cavity flow: Pressure error comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer using $\mathbb{P}_1/\mathbb{P}_1$; stabilization coefficient $\delta = 0.05$; $N_u = N_p = N_s = 20$.

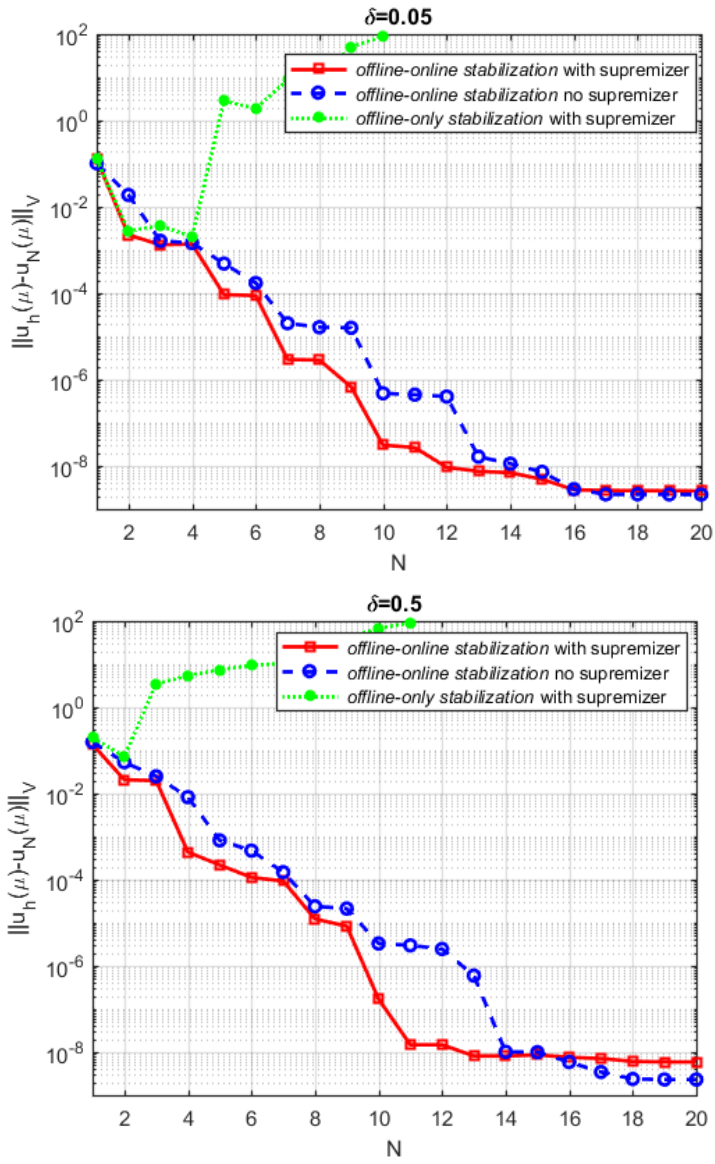


Figure 2.5: Franca-Hughes stabilization on cavity flow: Velocity error comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer using $\mathbb{P}_2/\mathbb{P}_2$; stabilization coefficient $\delta = 0.05, 0.5$; $N_u = N_p = N_s = 20$.

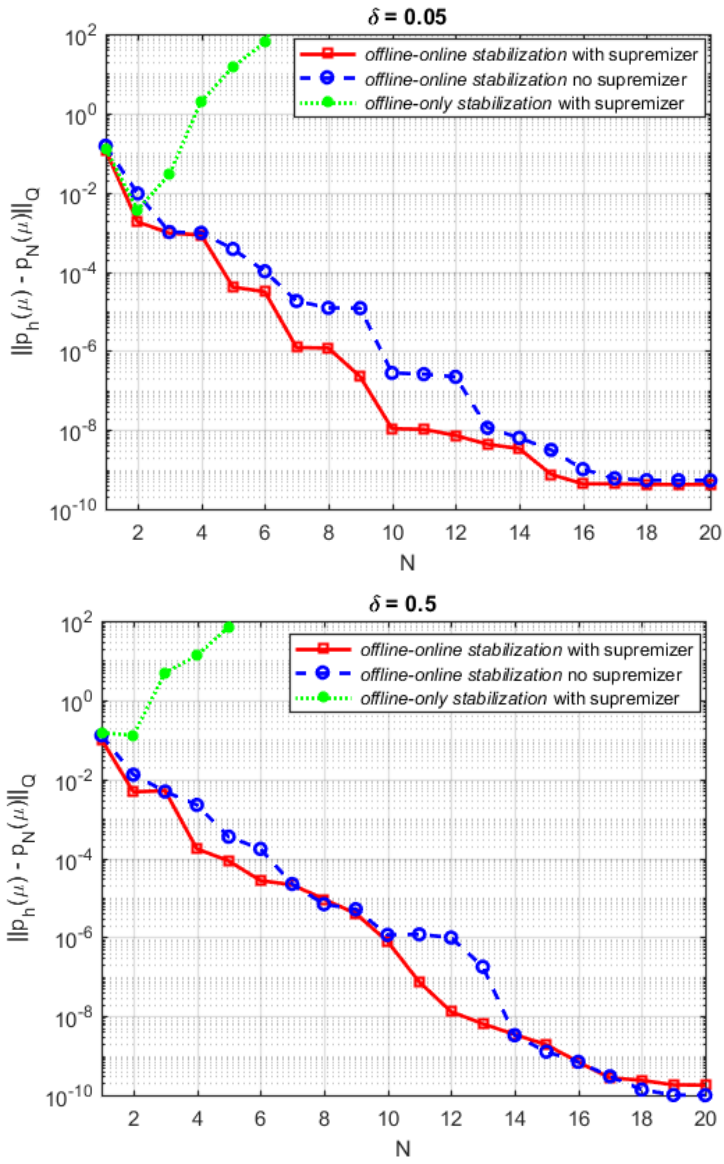


Figure 2.6: Franca-Hughes stabilization on cavity flow: Pressure error comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer using $\mathbb{P}_2/\mathbb{P}_2$; stabilization coefficient $\delta = 0.05, 0.5$; $N_u = N_p = N_s = 20$.

Keeping in mind the earlier comments on the choice of stabilization parameter δ in subsection 2.1.2, the reason here to vary this δ is to see if the properties of this coefficient in FE problem are preserved by RB problem or not. The answer is affirmative, as we can see in Figs. 2.5 and 2.6 that a suitable variation in δ does not give a significant change in the error between FE and RB solutions.

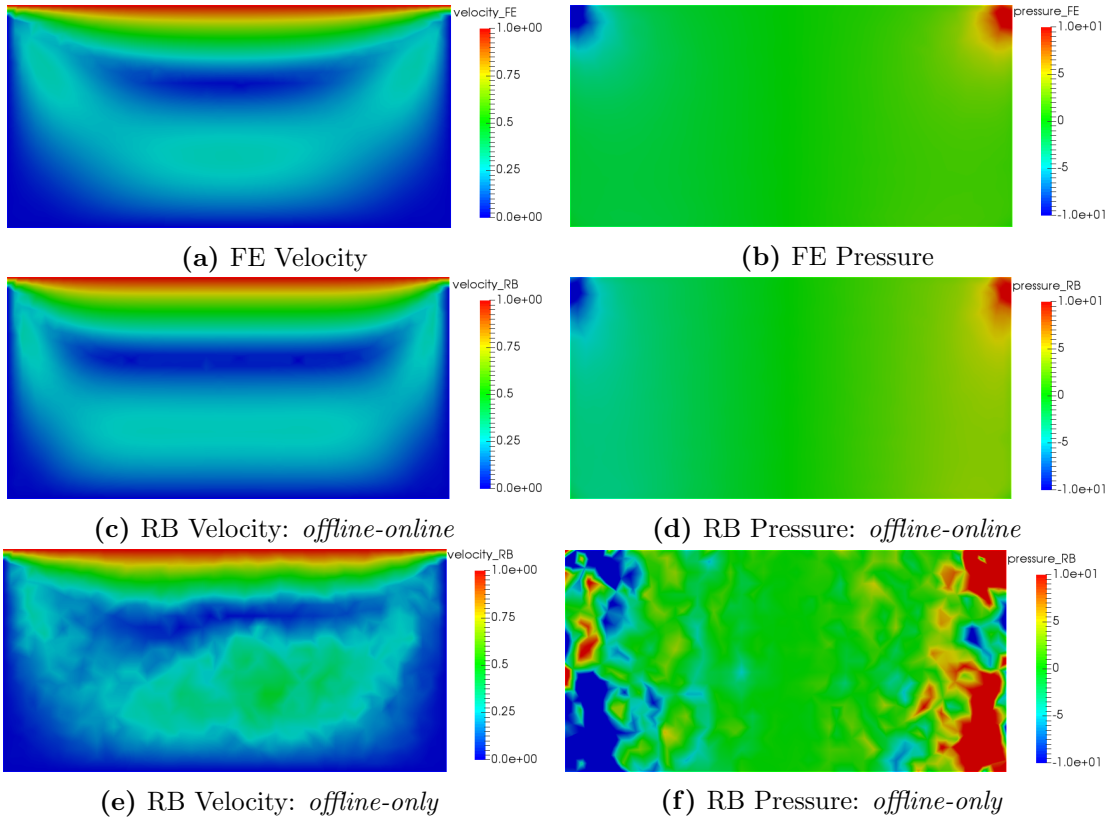
2.2.2 Numerical results for $\mathbb{P}_1/\mathbb{P}_0$

In this section we discuss the solution of steady parametrized Stokes problem using the lowest order finite element pair $\mathbb{P}_1/\mathbb{P}_0$. The choice of stabilization term $\phi_h(q_h)$ in equation (2.11) for lowest order element is as follows [117]:

$$s_h^q(q_h;\boldsymbol{\mu}) := \delta \sum_{\sigma\in\Gamma_h} h_\sigma \int_\sigma [p_h]_\sigma \, [q_h]_\sigma \tag{2.32}$$

where Γ_h is the set of all edges σ of the triangulation except for those belonging to the boundary $\partial\Omega$, h_σ is the length of σ and $[q_h]_\sigma$ denotes its jump across σ.

In Fig. 2.7 we show some snapshots for velocity and pressure fields using *stabilized* FE method and *stabilized* RB method. From these solution plots we conclude that we are able to recover a good qualitative approximation of FE solution at reduced order level using the *offline-online stabilization*, whereas the *offline-only stabilization* is not enough to recover FE approximation.



(a) FE Velocity **(b)** FE Pressure

(c) RB Velocity: *offline-online* **(d)** RB Pressure: *offline-online*

(e) RB Velocity: *offline-only* **(f)** RB Pressure: *offline-only*

Figure 2.7: Comparison between the FE solution and RB solutions for velocity and pressure obtained by three possible options: *(i) offline-online stabilization* with supremizer; *(ii) offline-online stabilization* without supremizer; *(iii) offline-only stabilization* with supremizer. Parameters range: *offline* $\mu_1 \in [0.25, 0.75], \mu_2 \in [1, 3]$; *online* $(\mu_1, \mu_2) = (0.6, 2)$; $N_u = N_p = 20$.

In Figs. 2.8 and 2.9 we plot the comparison between *offline-online stabilization* with/without supremizer and *offline-only stabilization* with supremizer for velocity and pressure, respectively for $\mathbb{P}_1/\mathbb{P}_0$. These comparison shows that the *offline-online stabilization* is the best way to stabilize and the addition of supremizer to velocity space is not necessary. Indeed, we are getting a good approximation of velocity without the supremizer. However in case of pressure, supremizer is improving the *offline-online stabilization* upto one order of magnitude.

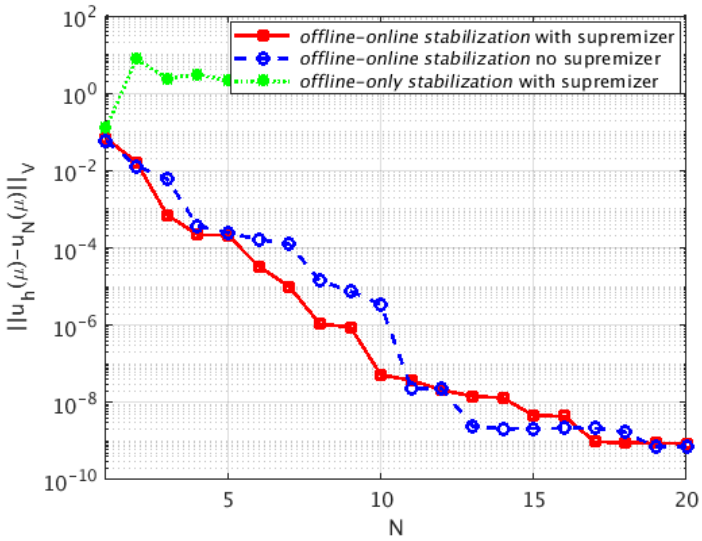


Figure 2.8: Stabilization with $\mathbb{P}_1/\mathbb{P}_0$ on cavity flow: Velocity error between FE solution and RB solution for different possible options.

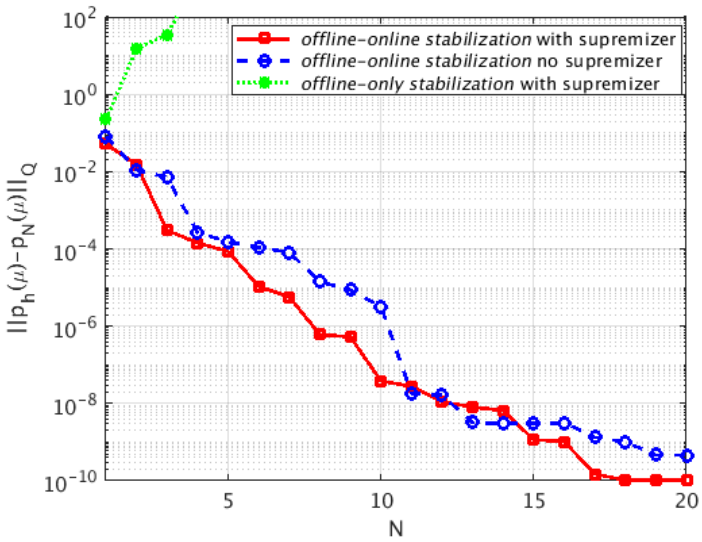


Figure 2.9: Stabilization with $\mathbb{P}_1/\mathbb{P}_0$ on cavity flow: Pressure error between FE solution and RB solution for different possible options.

2.2.3 Numerical results for stable $\mathbb{P}_2/\mathbb{P}_1$

Here we plot a comparison between the FE solution and RB solution for velocity and pressure in Figs. 2.10 and 2.11, respectively using stable FE pair $\mathbb{P}_2/\mathbb{P}_1$ [132]. We compare results with and without supremizer and conclude that supremizer is necessary to enrich the RB velocity space. In this case we are not using any *stabilization* method, therefore we need supremizer to fulfill reduced inf-sup condition.

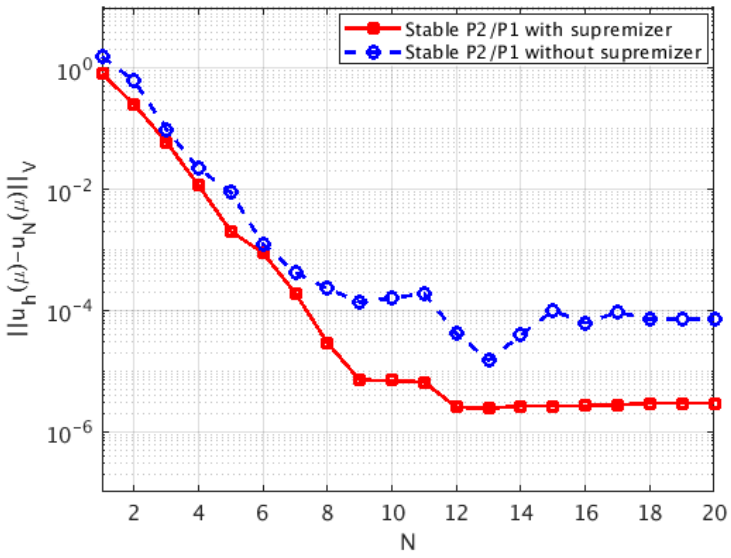


Figure 2.10: Cavity flow: Error between FE and RB solutions for velocity with/without supremizer.

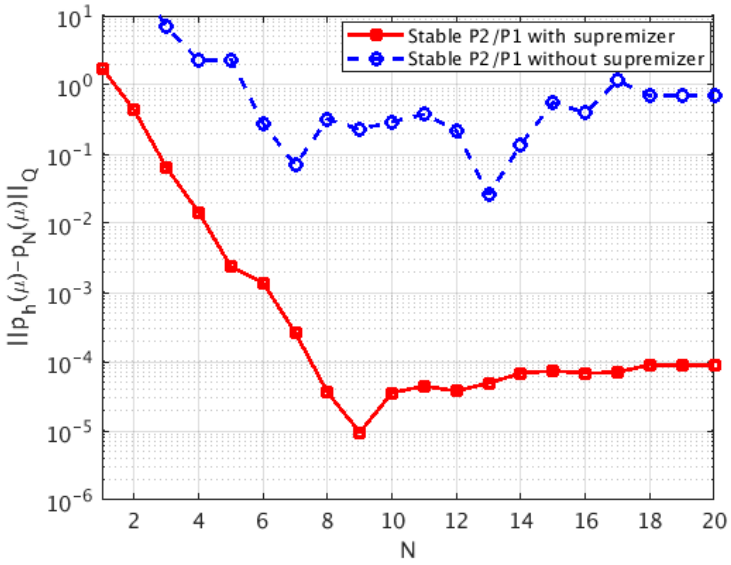


Figure 2.11: Cavity flow: Error between FE and RB solutions for velocity with/without supremizer.

2.3 Unsteady parametrized Stokes problem

In this section we extend the stabilization method introduced in section 2.1 to the case of time-dependent Stokes problem in a parametrized domain shown in Fig. 2.1. We introduce a stabilization term to the bilinear form, first in FE formulation and then, we project onto reduced order space in order to guarantee the reduced basis stability.

Let $\Omega \subset \mathbb{R}^2$, be a reference configuration, and we assume that current configuration $\Omega_o(\boldsymbol{\mu})$ can be obtained as the image of map $\boldsymbol{T}(.;\boldsymbol{\mu}) : \mathbb{R}^2 \to \mathbb{R}^2$, i.e. $\Omega_o(\boldsymbol{\mu}) = \boldsymbol{T}(\Omega;\boldsymbol{\mu})$. The unsteady parametrized Stokes problem in current configuration reads as follows: find $\boldsymbol{u}_o(t;\boldsymbol{\mu}) \in \boldsymbol{V}$ and $p_o(t;\boldsymbol{\mu}) \in Q$ such that

$$\begin{cases} \dfrac{\partial}{\partial t}\boldsymbol{u}_o - \nu\Delta\boldsymbol{u}_o + \nabla p_o = \boldsymbol{f} & \text{in } \Omega_o(\boldsymbol{\mu}) \times (0,T)\,, \\ \text{div}\boldsymbol{u}_\text{o} = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \times (0,T)\,, \\ \boldsymbol{u}_o = \boldsymbol{g} & \text{on } \partial\Omega \times (0,T)\,, \\ \boldsymbol{u}_o|_{t=0} = \boldsymbol{u_0} & \text{in } \Omega_o(\boldsymbol{\mu}), \end{cases} \tag{2.33}$$

where $(0,T)$ with $T > 0$ is the time interval of interest, $\boldsymbol{f}$ is the forcing function: $\boldsymbol{f} \in L^2(0,T;H^1(\Omega))$, $\boldsymbol{u_0} \in L^2(\Omega)$ and ν is the viscosity of fluid. We multiply (2.3) by velocity and pressure test functions $\boldsymbol{v}$ and q, respectively then integrating by parts, and tracing everything back onto the reference domain Ω, we obtain the following parametrized formulation:

for a given $\boldsymbol{\mu} \in \mathbb{P}$, find $\boldsymbol{u}(t;\boldsymbol{\mu}) \in \boldsymbol{V}$ and $p(t;\boldsymbol{\mu})) \in Q$ such that

$$\begin{cases} m(\dfrac{\partial}{\partial t}\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) + a(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) + b(\boldsymbol{v},p;\boldsymbol{\mu}) = F(\boldsymbol{v};\boldsymbol{\mu}) & \forall\, v \in \boldsymbol{V}, t > 0, \\ b(\boldsymbol{u},q;\boldsymbol{\mu}) = G(q;\boldsymbol{\mu}) & \forall\, q \in Q, t > 0, \\ \boldsymbol{u}|_{t=0} = \boldsymbol{u_0}. \end{cases} \tag{2.34}$$

We define the sobolev spaces; $\boldsymbol{V} = L^2(\mathbb{R}^+;[H^1(\Omega)]^2) \cap C^0(\mathbb{R}^+;[L^2(\Omega)]^2)$ for velocity and $Q = L^2(\mathbb{R}^+;L^2_0(\Omega))$ for pressure on reference domain. These spaces are equipped with H^1-seminorm and L^2-norm respectively. Bilinear forms in (2.34) are

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) &= \int_\Omega \frac{\partial \boldsymbol{u}}{\partial x_i}\kappa_{ij}(x;\boldsymbol{\mu})\frac{\partial \boldsymbol{v}}{\partial x_j}d\boldsymbol{x}, \\ b(\boldsymbol{v},q;\boldsymbol{\mu}) &= -\int_\Omega q\chi_{ij}(x;\boldsymbol{\mu})\frac{\partial v_j}{\partial x_i}d\boldsymbol{x}, \\ m(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) &= \int_\Omega \pi(\boldsymbol{x},\boldsymbol{\mu})\boldsymbol{u}_i\boldsymbol{v}_i d\boldsymbol{x}. \end{aligned} \tag{2.35}$$

As in steady case, assume for simplicity that $\boldsymbol{f} = \boldsymbol{0}$, so that the right hand sides in (2.34) vanishes. The tensors $\boldsymbol{\kappa}$ and $\boldsymbol{\chi}$ are defined bby (2.4). Th scalar π encoding both physical and geometrical parametrization as follows

$$\pi(\boldsymbol{x};\boldsymbol{\mu}) = |J_T(X;\boldsymbol{\mu})|, \tag{2.36}$$

where $J_T \in \mathbb{R}^{2\times 2}$ is the Jacobian matrix of the map $\boldsymbol{T}(.;\boldsymbol{\mu})$, and $|J_T|$ denotes the determinant.

2.3.1 Semi-discrete Finite Element formulation

The mixed Galerkin finite element semi-discretization [53, 54] of (2.34) is defined as follows:

for a given $\boldsymbol{\mu} \in \mathbb{P}$, find $\boldsymbol{u}_h(t;\boldsymbol{\mu}) \in \boldsymbol{V}_h \subset \boldsymbol{V}$ and $p_h(t;\boldsymbol{\mu})) \in Q_h \subset Q$ such that

$$\begin{cases} m(\dfrac{\partial}{\partial t}\boldsymbol{u}_h, \boldsymbol{v}_h;\boldsymbol{\mu}) + a(\boldsymbol{u}_h, \boldsymbol{v}_h;\boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h;\boldsymbol{\mu}) = F(\boldsymbol{v}_h;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, t > 0, \\ b(\boldsymbol{u}_h, q_h;\boldsymbol{\mu}) = G(q_h;\boldsymbol{\mu}) & \forall\, q_h \in Q_h, t > 0, \\ \boldsymbol{u}_h|_{t=0} = \boldsymbol{u}_{0,h}. \end{cases} \tag{2.37}$$

We consider a partition of the interval $[0, T]$ into K sub-intervals of equal length $\Delta t = T/K$ and $t^k = k\Delta t, 0 \le k \le K$. Applying the implicit Euler time discretization we obtain the following time discrete problem:

for a given $\boldsymbol{\mu} \in \mathbb{P}$, and $(\boldsymbol{u}_h^{k-1}(\boldsymbol{\mu}), p_h^{k-1}(\boldsymbol{\mu}))$, find $\boldsymbol{u}_h^k(t;\boldsymbol{\mu}) \in \boldsymbol{V}_h$ and $p_h^k(t;\boldsymbol{\mu}) \in Q_h$ such that

$$\begin{cases} \frac{1}{\Delta t} m(\boldsymbol{u}_h^k, \boldsymbol{v}_h;\boldsymbol{\mu}) + a(\boldsymbol{u}_h^k, \boldsymbol{v}_h;\boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h^k;\boldsymbol{\mu}) = F(\boldsymbol{v}_h;\boldsymbol{\mu}) & \\ + \frac{1}{\Delta t} m(\boldsymbol{u}_h^{k-1}, \boldsymbol{v}_h; t^{k-1};\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h^k, q_h;\boldsymbol{\mu}) = G(q_h;\boldsymbol{\mu}) & \forall\, q_h \in Q_h, \\ \boldsymbol{u}_h^0 = \boldsymbol{u}_{0,h}. \end{cases} \tag{2.38}$$

Similar to what we did for steady Stokes problem in 2.1.1, we provide the algebraic formulation of the semi-discrete problem (2.37). The resulting ODE system is as follows:

$$\begin{bmatrix} M(\boldsymbol{\mu}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{U}}(t;\boldsymbol{\mu}) \\ \dot{\boldsymbol{P}}(t;\boldsymbol{\mu}) \end{bmatrix} + \begin{bmatrix} A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t;\boldsymbol{\mu}) \\ \boldsymbol{P}(t;\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \tag{2.39}$$

for the vectors $\boldsymbol{U} = (u_h^{(1)}, ..., u_h^{(\mathcal{N}_u)})^T, \boldsymbol{P} = (p_h^{(1)}, ..., p_h^{(\mathcal{N}_p)})^T$, where for $1 \le i, j \le \mathcal{N}_u$ and $1 \le k \le \mathcal{N}_p$:

$$\begin{aligned} (M(\boldsymbol{\mu}))_{ij} &= m(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \qquad (A(\boldsymbol{\mu}))_{ij} = a(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \\ (B(\boldsymbol{\mu}))_{ki} &= b(\boldsymbol{\phi}_i^h, \boldsymbol{\psi}_k^h;\boldsymbol{\mu}), \qquad (\bar{\boldsymbol{f}}(\boldsymbol{\mu}))_i = F(\boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \\ & \qquad\qquad (\bar{\boldsymbol{g}}(\boldsymbol{\mu}))_k = G(\boldsymbol{\psi}_k^h;\boldsymbol{\mu}), \end{aligned} \tag{2.40}$$

A key assumption for an efficient ROM evaluation is the capability to decouple the construction stage of the reduced order space (*offline*) from evaluation stage (*online*). We require that the matrices and vectors appearing in (2.40) can be written as

$$\begin{aligned} M(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) M^q, \quad A(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) A^q, \qquad B(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_q^b(\boldsymbol{\mu}) B^q, \\ \bar{\boldsymbol{f}}(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_f} \Theta_q^f(\boldsymbol{\mu}) \bar{\boldsymbol{f}}^q, \qquad \bar{\boldsymbol{g}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \bar{\boldsymbol{g}}^q. \end{aligned} \tag{2.41}$$

After applying the time discretization with implicit Euler scheme, the resulting algebraic formulation of (2.38) is

$$\begin{aligned}\begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} + A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}(t^{k-1};\boldsymbol{\mu}) \end{bmatrix}\end{aligned} \tag{2.42}$$

Non-homogeneous Dirichlet boundary conditions are imposed by introducing a suitable lifting function, in a similar way as in steady Stokes case in 2.1.1.

2.3.2 Stabilized Finite Element formulation

Similar to what we discussed for steady Stokes problem, the uniqueness of the system (2.42) is subject to the fulfillment of parametrized $\inf - \sup$ condition (2.10).

There are many combinations of finite element spaces (V_h, Q_h) for which inf-sup condition (2.10) holds. However, it is well known that when we choose piecewise polynomial velocity and pressure of same degree defined w.r.t same grid, then this choice do not obey the inf-sup condition (2.10). For this reason the theory of stabilized finite element is developed. Let us modify equation (2.37) by adding the stabilization terms. We read the modified formulation as follows:

for a given $\boldsymbol{\mu} \in \mathbb{P}$, find $\boldsymbol{u}_h(t;\boldsymbol{\mu}) \in \boldsymbol{V}_h$ and $p_h(t;\boldsymbol{\mu})) \in Q_h$ such that

$$\begin{cases} m(\frac{\partial}{\partial t}\boldsymbol{u}_h, \boldsymbol{v}_h;\boldsymbol{\mu}) + a(\boldsymbol{u}_h, \boldsymbol{v}_h;\boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h;\boldsymbol{\mu}) = F(\boldsymbol{v}_h;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, t > 0, \\ b(\boldsymbol{u}_h, q_h;\boldsymbol{\mu}) = G(q_h;\boldsymbol{\mu}) + \phi_h(q_h;\boldsymbol{\mu}) & \forall\, q_h \in Q_h, t > 0, \\ \boldsymbol{u}_h|_{t=0} = \boldsymbol{u}_{0,h}. \end{cases} \tag{2.43}$$

where $\phi_h(q_h;\boldsymbol{\mu})$ is the stabilization term. We have already discussed the stabilization options in subsection 2.1.2 in detail. For unsteady Stokes problem, the stabilization term defined in (2.13) is modified as:

$$\phi_h(q_h;\boldsymbol{\mu}) = \delta \sum_K h_K^2 \int_K (\frac{\partial}{\partial t}\boldsymbol{u}_h - \nu\Delta\boldsymbol{u}_h + \nabla p_h - \boldsymbol{f}, \nabla q_h) \tag{2.44}$$

Therefore, the stabilized algebraic system for unsteady Stokes problem can be written as:

$$\begin{bmatrix} M(\boldsymbol{\mu}) & \mathbf{0} \\ \tilde{M}(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{U}}(t;\boldsymbol{\mu}) \\ \dot{\boldsymbol{P}}(t;\boldsymbol{\mu}) \end{bmatrix} + \begin{bmatrix} A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ \tilde{B}(\boldsymbol{\mu}) & -S(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t;\boldsymbol{\mu}) \\ \boldsymbol{P}(t;\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \tag{2.45}$$

where $\tilde{M}(\boldsymbol{\mu}), \tilde{B}(\boldsymbol{\mu})$ and $-S(\boldsymbol{\mu})$ contains the stabilization effects [68].

After applying the time discretization with implicit Euler scheme, the system (2.45) becomes

$$\begin{aligned}\begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} + A(\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ \tilde{B}(\boldsymbol{\mu}) + \frac{\tilde{M}(\boldsymbol{\mu})}{\Delta t} & -S(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \frac{\tilde{M}(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}(t^{k-1};\boldsymbol{\mu}) \end{bmatrix}.\end{aligned} \tag{2.46}$$

2.3.3 Reduced Basis formulation

We present a RB method for solving parametrized unsteady Stokes problem. Similar to what we discussed in steady Stokes case, in this section we present the reduced basis formulation of the unsteady Stokes problem formulated in section 2.3.1. Let us define the parameter sample $s_N = \{\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^N\}$, where $\boldsymbol{\mu}^n \in \mathbb{P}$. The reduced basis approximation is based on an N-dimensional reduced basis spaces $\boldsymbol{V}_N$ and Q_N generated by a sampling procedure which combines spatial snapshots in time and parameter space in an optimal way. In particular, in our case we have used the POD-Greedy algorithm 1.2.6 for snapshots selection to generate the reduced spaces. Reduced basis velocity and pressure spaces are

$$\boldsymbol{V}_N = \text{ span}\left\{\text{POD}(\boldsymbol{u}_h(t^k;\boldsymbol{\mu}^n)), 1 \le k \le K, 1 \le n \le N_u\right\}, \tag{2.47}$$

$$Q_N = \text{ span}\left\{\text{POD}(p_h(t^k;\boldsymbol{\mu}^n)), 1 \le k \le K, 1 \le n \le N_p\right\}. \tag{2.48}$$

When the velocity space is enriched with the supremizer solutions at each time step, the velocity space in this case is defined as:

$$\tilde{\boldsymbol{V}}_N = \text{span}\left\{\text{POD}(\boldsymbol{u}_h(t^k;\boldsymbol{\mu}^n)), 1 \le n \le N_u; T^{\boldsymbol{\mu}}\xi_n^s, 1 \le n \le N_s\right\}. \tag{2.49}$$

Now the reduced basis formulation corresponding to semi-discrete FE formulation (2.37) can be written as: find $\boldsymbol{u}_N \in \tilde{\boldsymbol{V}}_N$ and $p_N \in Q_N$ such that

$$\begin{cases} m(\frac{\partial}{\partial t}\boldsymbol{u}_N, \boldsymbol{v}_N;\boldsymbol{\mu}) + a(\boldsymbol{u}_N, \boldsymbol{v}_N;\boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N;\boldsymbol{\mu}) = F(\boldsymbol{v}_N;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \boldsymbol{V}_N, \\ b(\boldsymbol{u}_N, q_N;\boldsymbol{\mu}) = G(q_N;\boldsymbol{\mu}) & \forall\, q_N \in Q_N, \\ \boldsymbol{u}_N|_{t=0} = \boldsymbol{u}_{0,N}, \end{cases} \tag{2.50}$$

In the online stage, the algebraic formulation of resulting reduced order approximation for any $\boldsymbol{\mu} \in \mathbb{P}$ is given by

$$\begin{aligned}\begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} + A_N(\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ B_N(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^{k-1};\boldsymbol{\mu}) \end{bmatrix},\end{aligned} \tag{2.51}$$

where the reduced order matrices are defined as:

$$\begin{aligned}M_N(t;\boldsymbol{\mu}) = Z_{u,s}^T M(t;\boldsymbol{\mu}) Z_{u,s}, \quad A_N(\boldsymbol{\mu}) = Z_{u,s}^T A(\boldsymbol{\mu}) Z_{u,s}, \quad B_N(\boldsymbol{\mu}) = Z_p^T B(\boldsymbol{\mu}) Z_{u,s}, \\ \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) = Z_{u,s}^T \bar{\boldsymbol{f}}(\boldsymbol{\mu}), \quad \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) = Z_p^T \bar{\boldsymbol{g}}(\boldsymbol{\mu}),\end{aligned} \tag{2.52}$$

with $Z_{u,s}$ being the velocity snapshot matrix including the supremizer solutions, Z_p denotes the pressure snapshot matrix. Moreover, thanks to the affine parametric dependence (2.41), we need to store only the matrices and vectors

$$A_N^q = Z_{u,s}^T A^q Z_{u,s}, \quad B_N^q = Z_p^T B^q Z_{u,s}, \quad \bar{\boldsymbol{f}}_N^q = Z_{u,s}^T \bar{\boldsymbol{f}}^q, \quad \bar{\boldsymbol{g}}_N^q = Z_p^T \bar{\boldsymbol{g}}^q. \tag{2.53}$$

The store data structures do not depend explicitly on time because the temporal dependence is stored in the multiplicative factors $\Theta(t;\boldsymbol{\mu})$. Therefore, $A^q, B^q, \bar{\boldsymbol{f}}^q, \bar{\boldsymbol{g}}^q$ are independent of both $\boldsymbol{\mu}$ and t.

2.3.4 Stabilized Reduced Basis formulation

In this section we present the stabilized Reduced Basis (RB) model for unsteady Stokes problem derived from the stabilized FE problem (2.43). The computation of the reduced spaces is done through the POD-Greedy algorithm (see algorithm 1.3). The stabilized RB approximation of velocity and pressure field obtained by means of Galerkin projection on reduced spaces reads:

for any $\boldsymbol{\mu} \in \mathbb{P}$, find $\boldsymbol{u}_N(t;\boldsymbol{\mu}) \in \boldsymbol{V}_N$ and $p_N(t;\boldsymbol{\mu})) \in Q_N$ such that

$$\begin{cases} m(\dfrac{\partial}{\partial t}\boldsymbol{u}_N, \boldsymbol{v}_N;\boldsymbol{\mu}) + a(\boldsymbol{u}_N, \boldsymbol{v}_N;\boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N;\boldsymbol{\mu}) = F(\boldsymbol{v}_N;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \boldsymbol{V}_N, \\ b(\boldsymbol{u}_N, q_N;\boldsymbol{\mu}) = G(q_N;\boldsymbol{\mu}) + \phi_N(q_N;\boldsymbol{\mu}) & \forall\, q_N \in Q_N, \\ \boldsymbol{u}_N|_{t=0} = \boldsymbol{u}_{0,N}, \end{cases} \tag{2.54}$$

where $\phi_N(q_N;\boldsymbol{\mu})$ is the reduced order counterpart of the stabilization term (2.44).

Finally, we write the reduced order stabilized formulation of unsteady FE stabilized

Stokes problem (2.46) in compact form as:

$$\begin{aligned}\begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} + A_N(\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ \tilde{B}_N(\boldsymbol{\mu}) + \frac{\tilde{M}_N(\boldsymbol{\mu})}{\Delta t} & -S_N(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \frac{\tilde{M}_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^{k-1};\boldsymbol{\mu}) \end{bmatrix}. \end{aligned} \tag{2.55}$$

Similar to steady Stokes case, we discuss and compare the following options using unstable FE pair $\mathbb{P}_k/\mathbb{P}_k$:

- for *offline-online stabilization* with supremizer we solve the stabilized system (2.46) in the *offline* stage and stabilized RB system (2.55) in the *online* stage; and the velocity space in this case is enriched with supremizer solutions, given by (2.49);
- for *offline-online stabilization* without supremizer we solve the stabilized system (2.46) in the *offline* stage and stabilized RB system (2.55) in the *online* stage; but the velocity space in this case is given by (2.47);
- for *offline-only stabilization* with supremizer we solve the stabilized system (2.46) in the *offline* stage and non-stabilized RB system (2.51) in the *online* stage; and the velocity space in this case is enriched with supremizer solutions, given by (2.49);

2.4 Numerical results and discussion

In this section, we present several numerical results for stabilized reduced order model for unsteady Stokes problem developed in section 2.3 and subsections therein. In subsection 2.4.1 we show some numerical solutions and error analysis using unstable FE pair $\mathbb{P}_2/\mathbb{P}_2$ for Franca-Hughes stabilization [68]. In subsection 2.4.2 we plot error comparison using lowest order unstable element $\mathbb{P}_1/\mathbb{P}_0$ with stabilization term (2.32). Finally in subsection 2.4.3 we show the variation of time step Δt on error between FE and RB solutions.

We consider the same test case as we did for the steady case on parametrized domain in Fig. 2.1. For unsteady Stokes case, the problem is defined as:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} - \mu_1 \Delta \boldsymbol{u} + \nabla p = \mathbf{0} & \text{in } \Omega_o(\mu_2) \times (0,T), \\ \text{div}\boldsymbol{u} = 0 & \text{in } \Omega_o(\mu_2) \times (0,T), \\ \boldsymbol{u} = \mathbf{0} & \text{on } \Gamma_{D_o}(\mu_2) \times (0,T), \\ \boldsymbol{u} = (1,0) & \text{on } \Gamma_D \times (0,T), \\ \boldsymbol{u}|_{t=0} = \mathbf{0} & \text{in } \Omega. \end{cases} \tag{2.56}$$

2.4.1 Numerical results for $\mathbb{P}_k/\mathbb{P}_k$ for $k = 2$

The aim of present subsection is to show and discuss some numerical results for unsteady parametrized Stokes problem (2.4) using Franca-Hughes stabilization [68]. This subsection is the extension of subsection 2.2.1 to unsteady problem.

In Table 2.1 we show the details of parameter ranges in *offline*, *online* stages; and other information about the *offline* stage.

In Fig. 2.12 we show the RB solutions for velocity and pressure at different time steps using the *offline-online stabilization* without supremizer. We observe that as the time increases, both velocity and pressure fields are converging to steady state solutions. We have similar results with *offline-online stabilization* with supremizer that we do not show here.

Figure 2.13 shows the error between FE and RB velocity, whereas Fig. 2.14 plots the error between FE and RB pressure. From these plots we observe that the *offline-online stabilization* with and without supremizer have the same order of convergence in case of velocity but in case of pressure, supremizer is improving the *offline-online stabilization* upto one order of magnitude. THis property will be much important in case of coupling conditions in multi-physics involving pressure, for example, since we may guarantee a better accuracy. The *offline-only stabilization* with supremizer option has poor performance as it was in steady case.

Number of Parameters	2: μ_1(viscosity), μ_2(domain's length)
μ_1 range *offline*	[0.25,0.75]
μ_2 range *offline*	[1,2]
μ_1 value *online*	0.57
μ_2 value *online*	1.78
Final time	0.2
Time step Δt	0.02
N_{train}	25
N_{max}	25
Stabilization coefficient δ	0.05
FE degrees of freedom	6222 ($\mathbb{P}_1/\mathbb{P}_1$) 18300 ($\mathbb{P}_2/\mathbb{P}_2$)
RB dimension	$N_u = N_s = N_p = 30$
Computation time ($\mathbb{P}_2/\mathbb{P}_1$)	$1780s$ (*offline*), $300s$ (*online*) with supremizer
Offline time ($\mathbb{P}_1/\mathbb{P}_1$)	$1046s$ (*offline-online stabilization* with supremizer) $738s$ (*offline-online stabilization* without supremizer) 980 (*offline-only stabilization* with supremizer)
Offline time ($\mathbb{P}_2/\mathbb{P}_2$)	$2260s$ (*offline-online stabilization* with supremizer) $1945ss$ (*offline-online stabilization* without supremizer) $17300s$ (*offline-only stabilization* with supremizer)
Online time ($\mathbb{P}_1/\mathbb{P}_1$)	$103s$ (with supremizer) $82s$ (without supremizer) $81s$ (*offline-only stabilization*)
Online time ($\mathbb{P}_2/\mathbb{P}_2$)	$242s$ (with supremizer) $180s$ (without supremizer) $90s$ (*offline-only stabilization*)

Table 2.1: Computational details of unsteady Stokes problem (2.4).

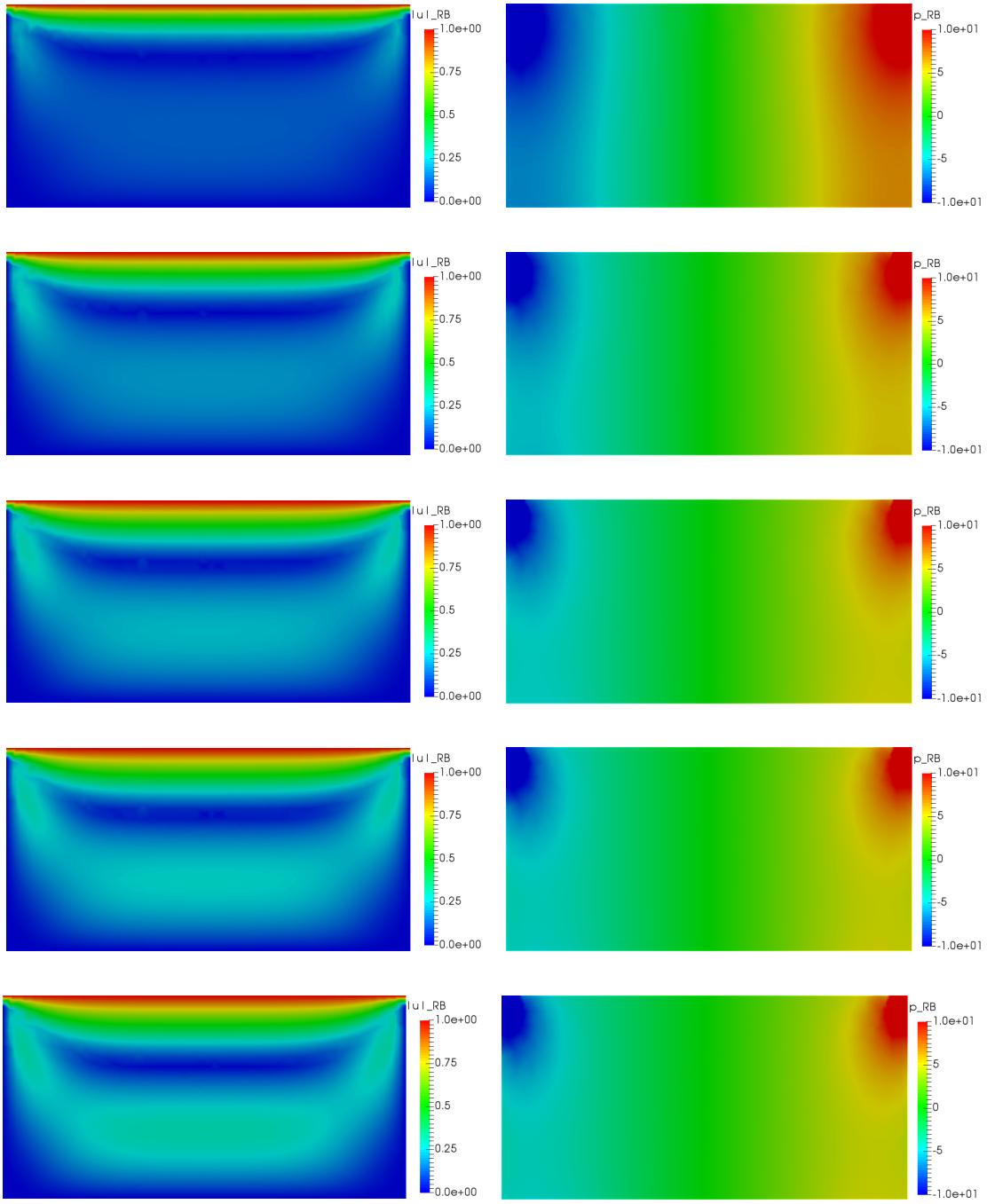


Figure 2.12: Franca-Hughes stabilization with $\mathbb{P}_2/\mathbb{P}_2$ FE pair: RB solutions for Velocity field (left) and Pressure field (right) at different time step from top to bottom; $t = 0.02, 0.04, 0.06, 0.1, 0.12$, $N_u = N_p = 30$.

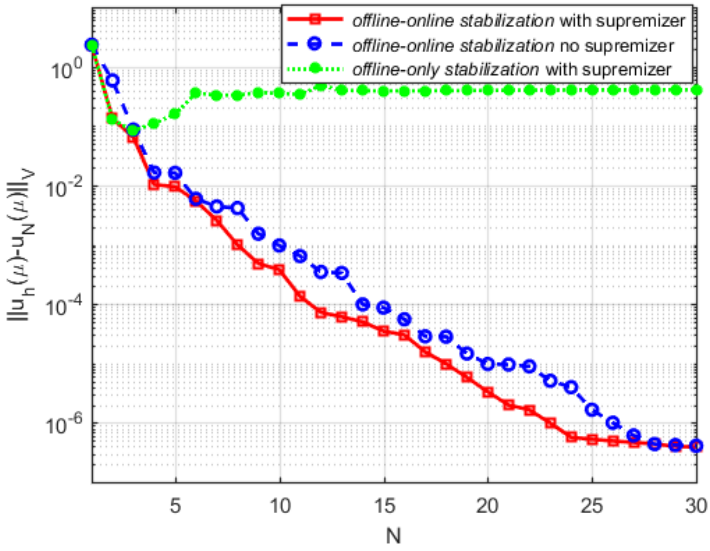


Figure 2.13: Franca-Hughes stabilization with $\mathbb{P}_2/\mathbb{P}_2$ on cavity flow: Velocity L^2-error in time for stabilization coefficient $\delta = 0.05$ and $\Delta t = 0.02$.

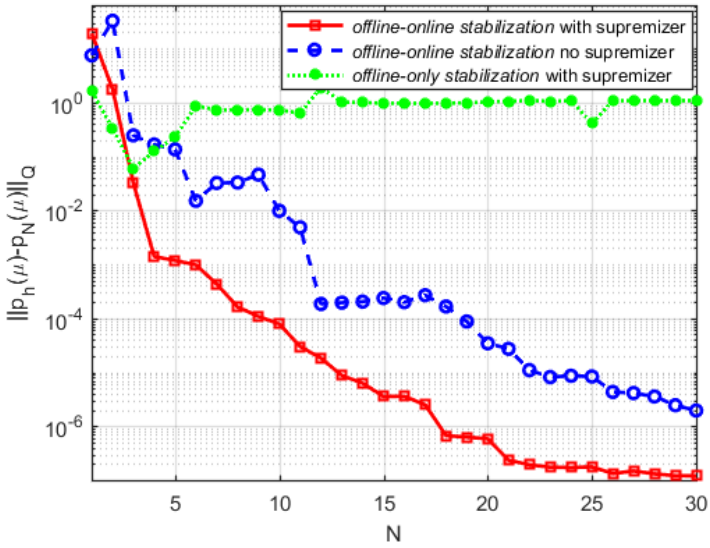


Figure 2.14: Franca-Hughes stabilization with $\mathbb{P}_2/\mathbb{P}_2$ on cavity flow: Pressure L^2-error in time for stabilization coefficient $\delta = 0.05$ and $\Delta t = 0.02$.

2.4.2 Numerical results for $\mathbb{P}_1/\mathbb{P}_0$

Similar to subsection 2.2.2 for steady Stokes case, in this subsection we show some results for the error comparison between the different stabilization options using the lowest order FE pair $\mathbb{P}_1/\mathbb{P}_0$.

The motivation in doing this case is to support the *offline-online stabilization*, i.e, we want to show, by doing different numerical experiments that the *offline-online stabilization* is the best way to stabilize whatever the stabilization we chose. For instance, in subsection 2.4.1 we chose the Franca-Hughes stabilization, which has different stabilization terms as compared to this subsection.

We plot the L^2- error in time for velocity and pressure in Figs. 2.15 and 2.16, respectively. The stabilization term in this case is defined in (2.32). Computational detail for this case is also given in Table 2.1. We have similar conclusion as before in subsection 2.4.1.

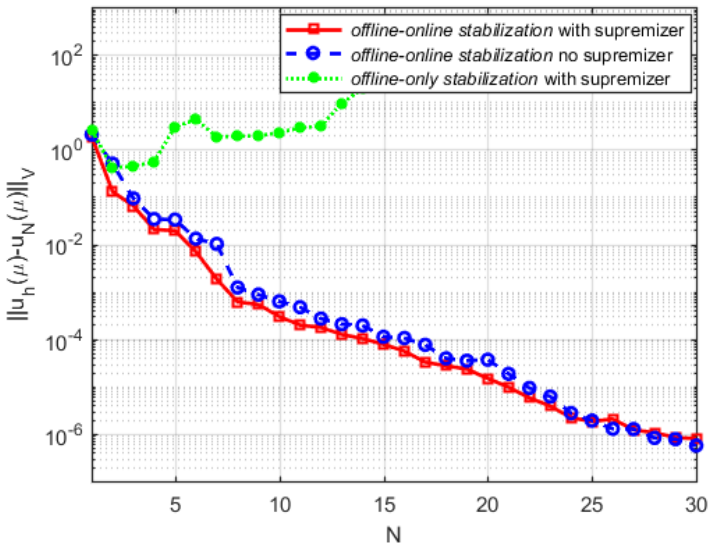


Figure 2.15: Cavity flow: Velocity L^2-error in time for stabilization coefficient $\delta = 0.05$ and $\Delta t = 0.02$. using $\mathbb{P}_1/\mathbb{P}_0$.

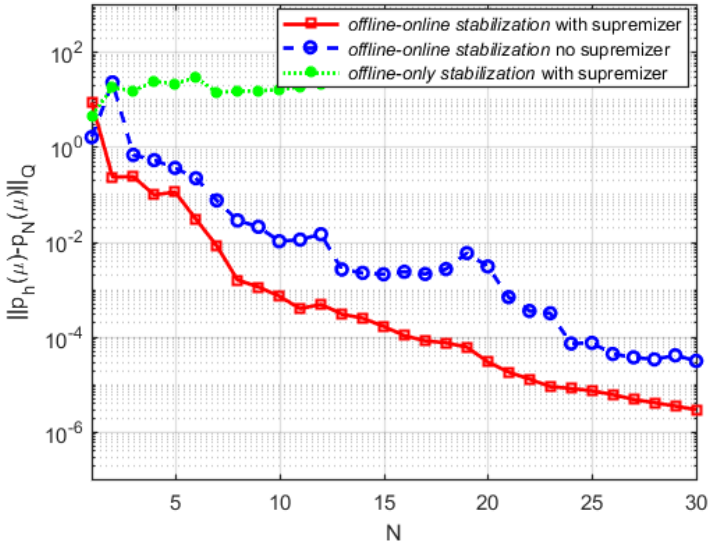


Figure 2.16: Cavity flow: Pressure L^2-error in time for stabilization coefficient $\delta = 0.05$ and $\Delta t = 0.02$. using $\mathbb{P}_1/\mathbb{P}_0$

2.4.3 Sensitivity on Δt

Consistently stabilized FE methods have complications while working with small time steps. These complications are reported in [20, 22] and references therein. The analysis found in [22] established that

$$\Delta t > \delta h^2$$

is a sufficient condition to avoid instabilities. Later on a detailed study and series of numerical experiments are performed in [21] and it is established that the fully discrete problem (2.43) is conditionally stable with the condition

$$\Delta t/\delta h^2 \geq \delta, \tag{2.57}$$

where Δt is the time step, δ is the stabilization coefficient independent of the spatial grid size h.

In this subsection we present some numerical results to see the variation of Δt on the error between FE and RB solutions. We use the *offline-online stabilization* without supremizer to plot the velocity error between FE and RB solution in Fig. 2.17 and pressure error between FE and RB solution in Fig. 2.18, respectively. We fix the value of stabilization coefficient $\delta = 0.05$

From these error plots, we observe that $\Delta t = 0.02$ (in this case, not generally) is the best value. If we decrease the value of Δt, keeping δ and h fixed, i.e, we are decreasing the left hand side of (2.57), which increases the error.

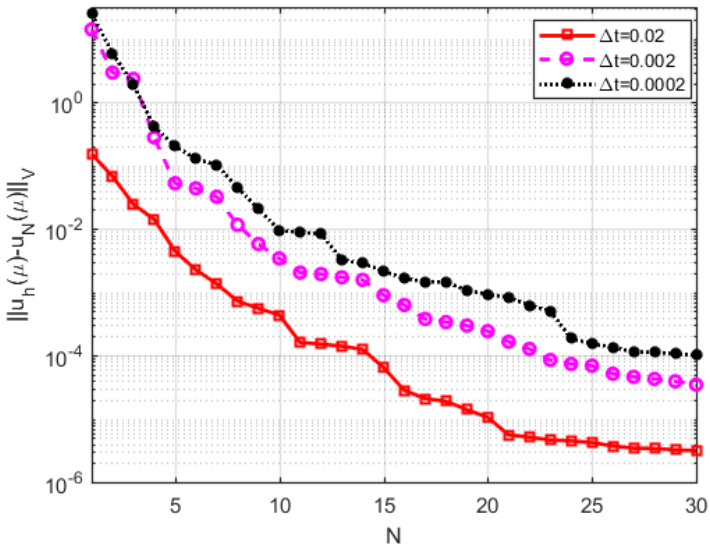


Figure 2.17: Franca-Hughes stabilization: L^2-error in time for Velocity using $\mathbb{P}_2/\mathbb{P}_2$ and $\delta = 0.05$, $\Delta t = 0.02, 0.002, 0.0002$. *offline-online stabilization* without supremizer.

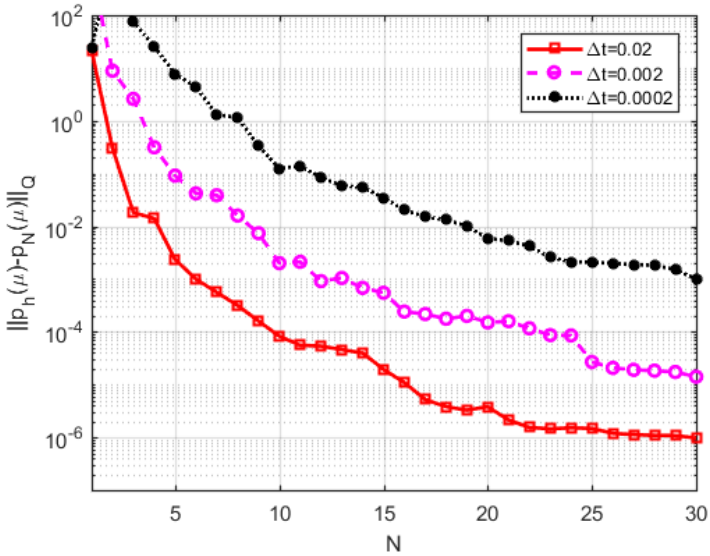


Figure 2.18: Franca-Hughes stabilization: L^2-error in time for Pressure using $\mathbb{P}_2/\mathbb{P}_2$ and $\delta = 0.05$, $\Delta t = 0.02, 0.002, 0.0002$. *offline-online stabilization* without supremizer.

2.5 Concluding remarks and perspectives

In this chapter we have used the classical residual based stabilization method given by Franca and Hughes [68] to develop the stabilized reduced basis method for parametrized steady and unsteady Stokes problem. We have compared the *offline-online stabilization* approach with the supremizer enrichment approach through a series of test cases for the steady and unsteady problems. To support our idea of *offline-online stabilization* we have also experimented other stabilization techniques like Brezzi-Pitkãranta [27] and also using lowest order stabilized element $\mathbb{P}_1/\mathbb{P}_0$. We summarized the main outcome as:

- all the numerical results carried out in this chapter shows that the *offline-online stabilization* is the most appropriate way to perform reduced basis stabilization of parametrized Stokes problem rather than *offline-only stabilization*;
- in case of velocity for both steady and unsteady problems, *offline-online stabilization* method allows us to avoid the addition of supremizer enrichment to fulfill the reduced order inf-sup condition which reduces the dimension and therefore we can reduce the computational cost;
- in case of pressure for unsteady problem, *offline-online stabilization* with supremizer has better performance in terms of error as compare to *offline-online stabilization* without supremizer, however supremizer enrichment does not improve so much the pressure in steady problem;
- the lack of consistency is causing larger errors if we use stabilized bilinear forms during the *offline* stage and non-stabilized bilinear forms during the online stage, i.e if we use *offline-only stabilized* method;
- construction of stable reduced basis functions in the *offline* stage does not guarantee the stable RB solution in the *online* stage (see Fig. 2.2).

Chapter 3

Reduced Basis Stabilization of Parametrized Navier-Stokes Problem

This chapter extends the stabilized RB method developed in Chapter 2 to parametric Navier-Stokes problem [41, 108]. We study the RB stabilization of steady and unsteady Navier-Stokes problems with physical and geometrical parametrization [8]. As already mentioned, our focus in this thesis is to deal only with inf-sup stability, that is, the instability of RB solution caused by the improper choice of velocity-pressure FE pair and not the instability due to dominating advection field [23, 117]. We compare the numerical results carried out using *offline-online stabilization* approach [2, 3, 64] with the existing supremizer approach [121, 132].

Similar to what we have already done in Chapter 2, we divide this chapter into two main parts, with the focus on steady Navier-Stokes problem in the first part (section: 3.1) and unsteady Navier-Stokes problem in the second part (section: 3.3), respectively, both in parametric settings.

Further detailed organization of this chapter is as follows: in section 3.1 we define the steady Navier-Stokes problem and weak formulation in parametrized domain. We introduce the stabilization terms (see for instance, [29, 44, 69, 148]) into the weak formulation. The construction of RB spaces for velocity and pressure, respectively is defined by using Greedy algorithm (section 1.2.3). In section 3.2 we show some numerical results with two examples.

In section 3.3 we define the unsteady Navier-Stokes problem in parametric setting. We define the semi-discrete FE formulation and stabilized FE formulation, respectively. In this case we use POD-Greedy algorithm [63] to select the snapshots for the construction of RB spaces. We present the RB formulation and stabilized RB formulation, respectively for the unsteady Navier-Stokes problem. In section 3.4 we present some numerical results for both physical and geometrical parametrization. Finally in section 3.5 we summarize and organize the main findings of this chapter.

3.1 Parametrized steady Navier-Stokes problem

Let us consider the steady incompressible Navier-Stokes equations in a parametrized domain $\Omega_o(\boldsymbol{\mu}) \subset \mathbb{R}^2$. The continuous parametrized formulation read as follows:

$$\begin{cases} -\nu\Delta \boldsymbol{u}_o + (\boldsymbol{u}_o \cdot \nabla)\, \boldsymbol{u}_o + \nabla p_o = \boldsymbol{f} & \text{in } \Omega_o(\boldsymbol{\mu}) \subset \mathbb{R}^2, \\ \text{div}(\boldsymbol{u}_o) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}), \\ \boldsymbol{u}_o = \boldsymbol{g}_D(\boldsymbol{\mu}) & \text{on } \Gamma_D, \\ \boldsymbol{u}_o = \boldsymbol{0}, & \text{on } \Gamma_o(\boldsymbol{\mu}), \end{cases} \tag{3.1}$$

where $\boldsymbol{u}_o$ is the unknown velocity, p is the unknown pressure, $\nu = \nu(\boldsymbol{\mu})$ is the kinematic viscosity representing the physical properties of our system, $\boldsymbol{f}$ is the body force, $\boldsymbol{g}_D$ is Dirichlet data, Γ_D is the Dirichlet boundary with non-homogeneous data and Γ_o denotes the Dirichlet boundary with zero data. For the sake of simplicity, we consider the case $\boldsymbol{f} = \boldsymbol{0}$. We denote by $\boldsymbol{\mu} \in \mathbb{P}$ a parameter which may be physical or geometrical parameter. In the numerical test cases that are presented in this chapter, our geometrical parameter is the length of domain and physical parameter is Reynolds number defined as $Re = L|\bar{\boldsymbol{u}}|/\nu$, being L a characteristic length of domain, $|\bar{\boldsymbol{u}}|$ a typical flow velocity and ν the kinematic viscosity.

In order to write the weak formulation of problem (3.1), we introduce a reference domain, i.e. a $\boldsymbol{\mu}$-independent configuration Ω by assuming that each parametrized domain $\Omega_o(\boldsymbol{\mu})$ can be obtained as the image of $\boldsymbol{\mu}$-independent domain Ω through a parametrized map $\boldsymbol{T}(.;\boldsymbol{\mu}) : \mathbb{R}^d \to \mathbb{R}^d$, i.e. $\Omega_o(\boldsymbol{\mu}) = \boldsymbol{T}(\Omega;\boldsymbol{\mu})$. We denote by $\boldsymbol{V}$, Q the solution spaces for velocity and pressure, respectively, defined over Ω such that $\boldsymbol{V} = H_0^1(\Omega)$, $Q = L_0^2(\Omega)$.

Now the weak formulation of (3.1) can be obtained by multiplying with the velocity, pressure test functions $\boldsymbol{v}$ and q and using integration by parts; then by tracing everything back onto the reference domain Ω, we have the following parametrized weak formulation of problem (3.1): find $(\boldsymbol{u}, p) \in \boldsymbol{V} \times Q$ such that

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) + b(\boldsymbol{v}, p; \boldsymbol{\mu}) + c(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) + d(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) = F(\boldsymbol{v}; \boldsymbol{\mu}) & \forall\, v \in \boldsymbol{V}, \\ b(\boldsymbol{u}, q; \boldsymbol{\mu}) = G(q; \boldsymbol{\mu}) & \forall\, q \in Q, \end{cases} \tag{3.2}$$

where F, G are the terms due to non-homogeneous Dirichlet boundary conditions, and

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) = \int_\Omega \frac{\partial \boldsymbol{u}}{\partial x_i} \kappa_{ij}(x; \boldsymbol{\mu}) \frac{\partial \boldsymbol{v}}{\partial x_j} d\boldsymbol{x}, \qquad b(\boldsymbol{v}, q; \boldsymbol{\mu}) = -\int_\Omega q \chi_{ij}(x; \boldsymbol{\mu}) \frac{\partial v_j}{\partial x_i} d\boldsymbol{x}, \tag{3.3}$$

are the bilinear forms related to diffusion and pressure-divergence operators, respectively [8], whereas the trilinear form related to the convective term is defined as:

$$c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\mu}) = \int_\Omega u_i \chi_{ji}(x; \boldsymbol{\mu}) \frac{\partial v_m}{\partial x_j} w_m d\boldsymbol{x}. \tag{3.4}$$

Here, we have adopted the convention of summation over repeated indices. We denote

$$\begin{aligned} \boldsymbol{\kappa}(x; \boldsymbol{\mu}) &= \nu(\boldsymbol{\mu})(J_T(x; \boldsymbol{\mu}))^{-1}(J_T(x; \boldsymbol{\mu}))^{-T}|J_T(X; \boldsymbol{\mu})|, \\ \boldsymbol{\chi}(x; \boldsymbol{\mu}) &= (J_T(x; \boldsymbol{\mu}))^{-1}|J_T(X; \boldsymbol{\mu})|, \end{aligned} \tag{3.5}$$

the tensors encoding the parametrization in Navier-Stokes operators; $|J_T|$ is the determinant of the Jacobian matrix $J_T \in \mathbb{R}^{d\times d}$ of the map $T(.;\boldsymbol{\mu})$.

Other terms appearing due to the lifting of Dirichlet boundary conditions are defined as: we denote by $\boldsymbol{l}(\boldsymbol{\mu})$ a parametrized lifting function such that $\boldsymbol{l}(\boldsymbol{\mu})|_{\Gamma_o} = \mathbf{0}, \boldsymbol{l}(\boldsymbol{\mu})|_{\Gamma_D} = \boldsymbol{g}_D(\boldsymbol{\mu})$, and

$$\begin{aligned} d(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) &= c(\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) + c(\boldsymbol{u},\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{v};\boldsymbol{\mu}),\\ F(\boldsymbol{v};\boldsymbol{\mu}) &= -a(\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{v},\boldsymbol{\mu}) - c(\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{l}(\boldsymbol{\mu}),\boldsymbol{v};\boldsymbol{\mu}),\\ G(q;\boldsymbol{\mu}) &= -b(\boldsymbol{l}(\boldsymbol{\mu}),q;\boldsymbol{\mu}). \end{aligned} \tag{3.6}$$

3.1.1 Finite Element formulation

In order to write the Galerkin-FE formulation for (3.2), we first need to introduce two finite-dimensional subspaces $\boldsymbol{V}_h \subset \boldsymbol{V}$, $Q_h \subset Q$ of dimension $\mathcal{N}_u$ and $\mathcal{N}_p$, respectively, being h related to the computational mesh size. The Galerkin-FE approximation of the parametrized problem (3.2) reads as follows: for a given parameter $\boldsymbol{\mu} \in \mathbb{P}$, we look for the full order solution $(\boldsymbol{u}_h(\boldsymbol{\mu}), p_h(\boldsymbol{\mu})) \in \boldsymbol{V}_h \times Q_h$ such that

$$\begin{cases} a(\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}) + b(\boldsymbol{v}_h,p_h(\boldsymbol{\mu});\boldsymbol{\mu}) + c(\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}) & \\ +d(\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}) = F(\boldsymbol{v}_h;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h(\boldsymbol{\mu}),q_h;\boldsymbol{\mu}) = G(q_h;\boldsymbol{\mu}) & \forall\, q_h \in Q_h, \end{cases} \tag{3.7}$$

where

$$\begin{aligned} d(\boldsymbol{u}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}) &= c(\boldsymbol{l}_h(\boldsymbol{\mu}),\boldsymbol{u}_h,\boldsymbol{v}_h;\boldsymbol{\mu}) + c(\boldsymbol{u}_h,\boldsymbol{l}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}),\\ F(\boldsymbol{v}_h;\boldsymbol{\mu}) &= -a(\boldsymbol{l}_h(\boldsymbol{\mu}),\boldsymbol{v}_h,\boldsymbol{\mu}) - c(\boldsymbol{l}_h(\boldsymbol{\mu}),\boldsymbol{l}_h(\boldsymbol{\mu}),\boldsymbol{v}_h;\boldsymbol{\mu}),\\ G(q_h;\boldsymbol{\mu}) &= -b(\boldsymbol{l}_h(\boldsymbol{\mu}),q_h;\boldsymbol{\mu}). \end{aligned} \tag{3.8}$$

To solve the system of nonlinear equations (3.7) arising from the space discretization of (3.2) we use the quadratically convergent Newton method [116], which involves the linearization of nonlinear convective term.

We denote by $\boldsymbol{\phi}_i^h$ and $\boldsymbol{\psi}_j^h$, the basis functions of $\boldsymbol{V}_h$ and Q_h respectively. We introduce the matrices $A(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}_u\times\mathcal{N}_u}$, $C(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}_u\times\mathcal{N}_u}$, and $B(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}_p\times\mathcal{N}_u}$ whose entries are

$$\begin{aligned} (A(\boldsymbol{\mu}))_{ij} &= a(\boldsymbol{\phi}_j^h,\boldsymbol{\phi}_i^h;\boldsymbol{\mu}) + d(\boldsymbol{\phi}_j^h,\boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \qquad (B(\boldsymbol{\mu}))_{ki} = b(\boldsymbol{\phi}_i^h,\boldsymbol{\psi}_k^h;\boldsymbol{\mu}),\\ (C(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}))_{ij} &= \sum_{m=1}^{\mathcal{N}_u} \boldsymbol{u}_h^m(\boldsymbol{\mu}) c(\boldsymbol{\phi}_m^h,\boldsymbol{\phi}_j^h,\boldsymbol{\phi}_i^h;\boldsymbol{\mu}), \quad \text{for } 1\le i,j\le\mathcal{N}_u, 1\le k\le\mathcal{N}_p, \end{aligned} \tag{3.9}$$

and therefore the nonlinear algebraic system is:

$$\begin{bmatrix} A(\boldsymbol{\mu}) + C(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(\boldsymbol{\mu}) \\ \boldsymbol{P}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}, \tag{3.10}$$

for the vectors of coefficients $\boldsymbol{U} = (u_h^{(1)}, ..., u_h^{(\mathcal{N}_u)})^T, \boldsymbol{P} = (p_h^{(1)}, ..., p_h^{(\mathcal{N}_p)})^T$, where for $1 \leq i \leq \mathcal{N}_u$ and $1 \leq k \leq \mathcal{N}_p$:

$$(\bar{\boldsymbol{f}}(\boldsymbol{\mu}))_i = -a(\boldsymbol{l}_h, \boldsymbol{\phi}_i^h; \boldsymbol{\mu}) - c(\boldsymbol{l}_h, \boldsymbol{l}_h, \boldsymbol{\phi}_i^h; \boldsymbol{\mu}), \quad (\bar{\boldsymbol{g}}(\boldsymbol{\mu}))_k = -b(\boldsymbol{l}_h, \boldsymbol{\psi}_k^h; \boldsymbol{\mu}), \tag{3.11}$$

with $\boldsymbol{l}_h = \boldsymbol{l}_h(\boldsymbol{\mu})$, the FE interpolant of lifting function.

For an efficient RB method, we need to ensure the assumption of affine parametric dependence on operators (3.9) and (3.11), i.e, these operators can be written as:

$$\begin{aligned} A(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu}) A^q, \qquad C(.;\boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_q^c(\boldsymbol{\mu}) C^q(.), \qquad B(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_q^b(\boldsymbol{\mu}) B^q, \\ \bar{\boldsymbol{f}}(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_f} \Theta_q^f(\boldsymbol{\mu}) \bar{\boldsymbol{f}}^q, \qquad \bar{\boldsymbol{g}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_q^g(\boldsymbol{\mu}) \bar{\boldsymbol{g}}^q. \end{aligned} \tag{3.12}$$

3.1.2 Stabilized Finite Element formulation

In this section we introduce the stabilization terms into the FE formulation of (3.7). The stabilized FE formulation of (3.7) read as: find $(\boldsymbol{u}_h(\boldsymbol{\mu}), p_h(\boldsymbol{\mu})) \in \boldsymbol{V}_h \times Q_h$ such that

$$\begin{cases} a(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h(\boldsymbol{\mu}); \boldsymbol{\mu}) + c(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) & \\ +d(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) = \mathcal{F}(\boldsymbol{v}_h; \boldsymbol{\mu}) + \xi_h(\boldsymbol{v}_h; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h(\boldsymbol{\mu}), q_h; \boldsymbol{\mu}) = G(q_h; \boldsymbol{\mu}) + \phi_h(q_h; \boldsymbol{\mu}) & \forall\, q_h \in Q_h, \end{cases} \tag{3.13}$$

where $\xi_h(\boldsymbol{v}_h; \mu)$ and $\phi_h(q_h; \mu)$ are the stabilization terms defined as:

$$\begin{aligned} \xi_h(\boldsymbol{v}_h; \boldsymbol{\mu}) &:= \delta \sum_K h_K^2 \int_K (-\nu \Delta \boldsymbol{u}_h + \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h + \nabla p_h, -\nu\gamma \Delta \boldsymbol{v}_h + \boldsymbol{u}_h \cdot \nabla \boldsymbol{v}_h), \\ \phi_h(q_h; \boldsymbol{\mu}) &:= \delta \sum_K h_K^2 \int_K (-\nu \Delta \boldsymbol{u}_h + \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h + \nabla p_h, \nabla q_h), \end{aligned} \tag{3.14}$$

where δ is the stabilization coefficient needs to be chosen properly [24, 91, 92]. For $\gamma = 0, 1$, the stabilization (3.14) is respectively known as Streamline Upwind Petrov Galerkin (SUPG) [30], Galerkin least-squares (GLS) [69]. The case $\gamma = -1$ was studied by Franca and Frey [48]. Several other works on these kind of stabilization techniques can be found in [90, 93, 142, 148] and references therein. In addition one can also use the additional *grad-div stabilization* term

$$\textstyle\sum_K \rho(\nabla \cdot \boldsymbol{u}, \nabla \cdot \boldsymbol{v}),$$

added to the momentum equation (3.13). Where ρ is a suitable stabilization coefficient, studied in several works, for instance see [106] and references therein. The *grad-div*

stabilization term is based on the residual of continuity equation and its role is to improve the discrete mass conservation.

In this chapter we discuss only the SUPG stabilization. Therefore, after adding the SUPG stabilization terms, the stabilized version of nonlinear system reads as (3.10):

$$\begin{bmatrix} A(\boldsymbol{\mu}) + \tilde{C}(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) & \tilde{B^T}(\boldsymbol{\mu}) \\ \tilde{B}(\boldsymbol{\mu}) & -S_N(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(\boldsymbol{\mu}) \\ \boldsymbol{P}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}, \tag{3.15}$$

where $\tilde{B}, \tilde{B^T}$ and $\tilde{C}$ contains the SUPG stabilization matrices [47]. For the detail about the inf-sup condition for equal order FE interpolation we refer to subsection 2.1.2 of Chapter 2. We follow the same pattern here for Navier-Stokes case.

3.1.3 Reduced Basis formulation

In this subsection we present the RB formulation of parametrized Navier-Stokes problem (3.1) [88, 114] in a similar way as we did for the Stokes case (see Chapter 2, subsection 2.1.3). We define the RB velocity space $\boldsymbol{V}_N \subset V_h$ and pressure space $Q_N \subset Q_h$, respectively as:

$$\boldsymbol{V}_N = \text{span}\left\{\xi_n^v = \boldsymbol{u}_h(\boldsymbol{\mu}^n), 1 \leq n \leq N_u\right\}, \tag{3.16}$$

and

$$Q_N = \text{span}\left\{\xi_n^p = p_h(\boldsymbol{\mu}^n), 1 \leq n \leq N_p\right\}, \tag{3.17}$$

where N_u and N_p are the dimensions of RB velocity space $\boldsymbol{V}_N$ and RB pressure space Q_N, respectively. $\{\xi_n^v\}_{n=1}^{N_u}$ and $\{\xi_n^p\}_{n=1}^{N_p}$ are mutually orthonormal basis functions for RB velocity and pressure, respectively obtained by applying the Gram-Schmidt orthogonalization process [63].

In order to fulfill the reduced order inf-sup condition (2.20), we need to solve the supremizer problem (2.19) to enrich the RB velocity space with the solutions of supremizer problem. In this we denote the RB velocity space by $\tilde{\boldsymbol{V}}_N$ with dimension $N_u + N_s$ and is defined as:

$$\tilde{\boldsymbol{V}}_N = \text{span}\left\{\xi_n^v, 1 \leq n \leq N_u; T^{\boldsymbol{\mu}}\xi_n^s, 1 \leq n \leq N_s\right\}. \tag{3.18}$$

Once, we have built the RB for velocity and pressure fields during the *offline* stage, the RB formulation corresponding to FE formulation (3.7) reads as: for any parameter $\boldsymbol{\mu} \in \mathbb{P}$, we look for $(\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \boldsymbol{V}_h \times Q_h$ such that

$$\begin{cases} a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N(\boldsymbol{\mu}); \boldsymbol{\mu}) + c(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) & \\ +d(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) = F(\boldsymbol{v}_N; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \tilde{\boldsymbol{V}}_N, \\ b(\boldsymbol{u}_N(\boldsymbol{\mu}), q_N; \boldsymbol{\mu}) = G(q_N; \boldsymbol{\mu}) & \forall\, q_N \in Q_N. \end{cases} \tag{3.19}$$

The solution $(\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \tilde{\boldsymbol{V}}_N \times Q_N$ of (3.19) can be expressed as a linear combination of the basis functions:

$$\boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{n=1}^{2N} \boldsymbol{u}_{Nn}(\boldsymbol{\mu})\xi_n^v, \qquad p_N(\boldsymbol{\mu}) = \sum_{n=1}^{N} p_{Nn}(\boldsymbol{\mu})\xi_n^p, \tag{3.20}$$

where $\{\boldsymbol{u}_{Nn}(\boldsymbol{\mu})\}_{n=1}^{N_u}$ and $\{p_{Nn}(\boldsymbol{\mu})\}_{n=1}^{N_p}$ denote the coefficients of the RB approximation for velocity and pressure. Finally, (3.19) can be written in compact form as following nonlinear reduced system

$$\begin{bmatrix} A_N(\boldsymbol{\mu}) + C_N(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ B_N(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U_N}(\boldsymbol{\mu}) \\ \boldsymbol{P_N}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_{\boldsymbol{N}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_{\boldsymbol{N}}(\boldsymbol{\mu}) \end{bmatrix}, \tag{3.21}$$

where the reduced order matrices are defined as:

$$\begin{aligned} &A_N(\boldsymbol{\mu}) = Z_u^T A(\boldsymbol{\mu}) Z_u, \quad B_N(\boldsymbol{\mu}) = Z_p^T B(\boldsymbol{\mu}) Z_u, \quad C_N(.;\boldsymbol{\mu}) = Z_u^T C(.;\boldsymbol{\mu}) Z_u, \\ &\bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) = Z_u^T \bar{\boldsymbol{f}}(\boldsymbol{\mu}), \quad \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) = Z_p^T \bar{\boldsymbol{g}}(\boldsymbol{\mu}), \end{aligned} \tag{3.22}$$

with $Z_{u,s}$, the velocity snapshot matrix and Z_p denotes the pressure snapshot matrix. The *offline-online* decomposition is made possible, thanks to the affine parametric dependence (3.12).

3.1.4 Stabilized Reduced Basis formulation

In this section, we present the stabilized RB method for the model problem (3.1). We also refer to some related works in recent past on the stabilization of reduced-order models, see for instance Baiges et al. [5, 6], Caiazzo et al. [32], Løvgren et al. [42] and Lassila et al. [85], Weller et al. [146] and Bergmann et al. [16].

The stabilized RB formulation corresponding to stabilized FE formulation (3.13) reads as:

$$\begin{cases} a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N;\boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N(\boldsymbol{\mu});\boldsymbol{\mu}) + c(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N;\boldsymbol{\mu}) & \\ +d(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N;\boldsymbol{\mu}) = \mathcal{F}(\boldsymbol{v}_N;\boldsymbol{\mu}) + \xi_N(\boldsymbol{v}_N;\boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \boldsymbol{V}_N, \\ b(\boldsymbol{u}_N(\boldsymbol{\mu}), q_N;\boldsymbol{\mu}) = G(q_N;\boldsymbol{\mu}) + \phi_N(q_N;\boldsymbol{\mu}) & \forall\, q_N \in Q_N, \end{cases} \tag{3.23}$$

where $\xi_N(\boldsymbol{v}_N;\mu)$ and $\phi_N(q_N;\mu)$ are the reduced order stabilization terms corresponding to (3.14).

We skip here the discussion about reduced order generalized inf-sup condition for Navier-Stokes problem with the stabilization options and we refer to subsection 2.1.4 of Chapter 2 for a detail presentation.

Finally, we write the matrix formulation of stabilized RB formulation (3.23) as:

$$\begin{bmatrix} A_N(\boldsymbol{\mu}) + \tilde{C}_N(\boldsymbol{u}(\boldsymbol{\mu});\boldsymbol{\mu}) & \tilde{B^T}_N(\boldsymbol{\mu}) \\ \tilde{B}_N(\boldsymbol{\mu}) & -S_N(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(\boldsymbol{\mu}) \\ \boldsymbol{P}_N(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix}, \tag{3.24}$$

where $\tilde{B}_N, \tilde{B^T}_N$ and $\tilde{C}_N$ contain the SUPG stabilization matrices [47].

3.2 Numerical results and discussion

In this section we present some numerical results for the RB approximation of steady parametrized Navier-Stokes problem developed in section 3.1 and subsections therein.

Numerical simulations are carried out using FreeFem++ [59]. We compare three possible options (i) *offline-online stabilization* with supremizer, (ii) *offline-online stabilization* without supremizer, (iii) *offline-only stabilization* with supremizer, whereas option (iv) *offline-only stabilization* without supremizer is not working.

3.2.1 Results for physical parametrization

In this test case, we apply the stabilized RB model for Navier-Stokes problem to the *lid driven-cavity* problem with only one physical parameter μ denotes the Reynolds number. The computational domain is shown in Fig. 3.1 and the boundary conditions are

$$u_1 = 1, u_2 = 0 \text{ on } \Gamma_{D_g} \text{ and } \boldsymbol{u} = \mathbf{0} \text{ on } \Gamma_{D_0} \tag{3.25}$$

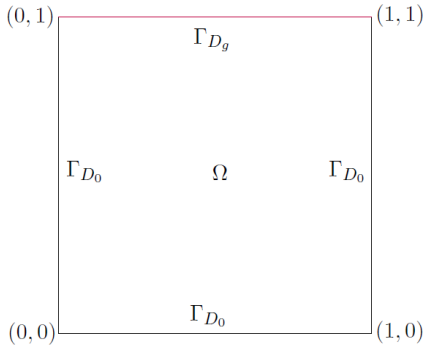
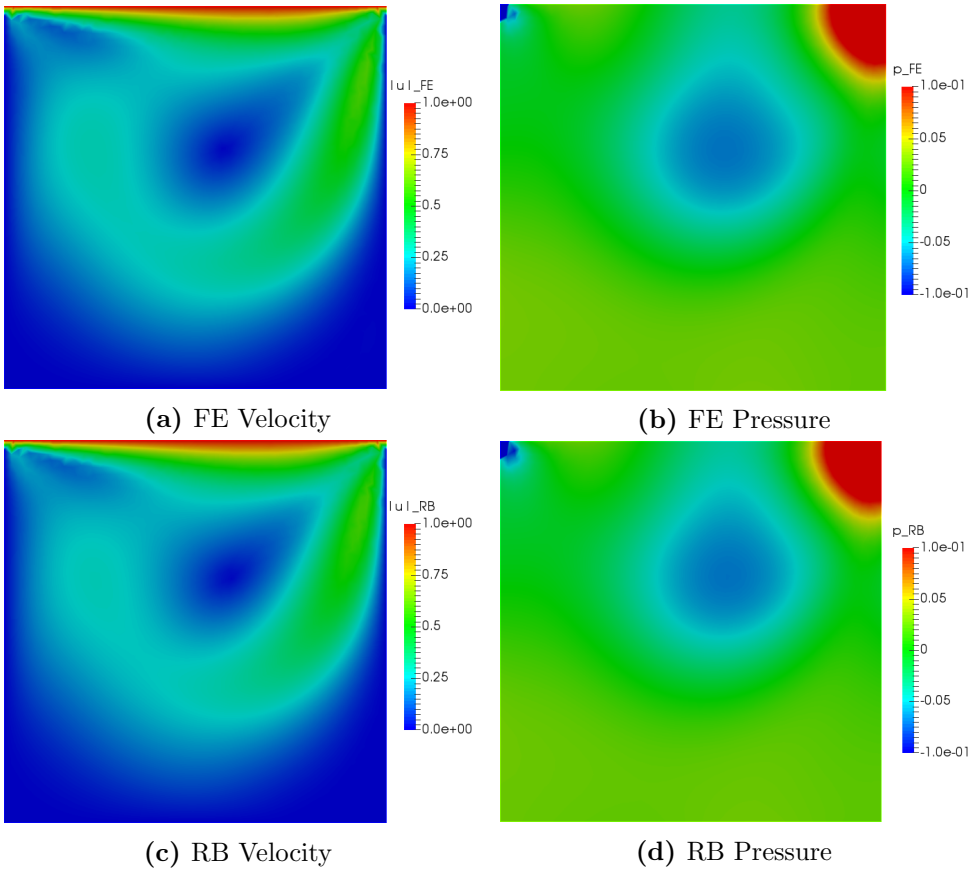


Figure 3.1: Domain Ω with boundaries marked.



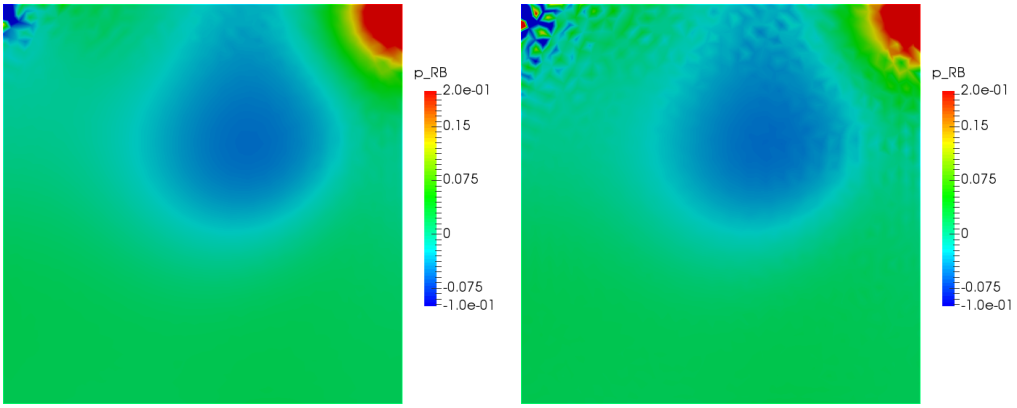
(a) FE Velocity **(b)** FE Pressure

(c) RB Velocity **(d)** RB Pressure

Figure 3.2: FE solution (top) and RB solution (below) at $Re = 200$ with $\mathbb{P}_1/\mathbb{P}_1$ for velocity and pressure, respectively.

The mesh of this problem is non-uniform with 3794 triangles and 1978 nodes, whereas the minimum and maximum size elements are $h_{min} = 0.0193145$ and $h_{max} = 0.0420876$, respectively.

Figure 3.2 plots the comparison by visualization of the fields between FE solutions (top) and RB solutions (below) for parameter value $\mu = 200$. These solutions are obtained by using the *offline-online stabilization* with $\delta = 1.0$ and $\mathbb{P}_1/\mathbb{P}_1$ FE pair. From these plots we see that both the FE and RB solutions for velocity and pressure are same. We have similar results for $\mathbb{P}_2/\mathbb{P}_2$ that do not show here.



(a) RB Pressure at $\delta = 0.1$ **(b)** RB Pressure at $\delta = 0.01$

Figure 3.3: RB pressure for varying δ using $\mathbb{P}_1/\mathbb{P}_1$ FE pair.

Figure 3.3 plots the variation of δ on RB pressure. We found that for $\mathbb{P}_1/\mathbb{P}_1$ FE pair the pressure solution becomes highly oscillatory for small values ($\delta \leq 0.1$) while for the values ($\delta \geq 1$) no oscillations were present as shown in Fig. 3.2. Similarly for $\mathbb{P}_2/\mathbb{P}_2$ FE pair the pressure solution is oscillatory for the values ($\delta \leq 0.01$) and no oscillations appear for the values ($\delta \geq 0.1$). This variation on δ has similar effects on FE pressure [91].

In Figs. 3.4 and 3.5 we plot the comparison between the *offline-online stabilization* with/without supremizer and *offline-only stabilization* for the error between FE and RB velocity using $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$, respectively. From these results we point out that velocity is "polluted" upto one order of magnitude by the enrichment of supremizer solutions.

Similarly in Figs. 3.6 and 3.7 the comparison between the above mentioned three options is made for the error between FE and RB pressure using $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$, respectively. In this case we see that the pressure is improved upto two order of magnitude for $\mathbb{P}_1/\mathbb{P}_1$ and one order of magnitude for $\mathbb{P}_2/\mathbb{P}_2$, by the addition of the supremizers. In both the velocity and pressure results, *offline-only stabilization* method has very poor result as compare to *offline-online stabilization*, i.e, the *offline-only stabilization* is not consistent.

The computation details of the problem considered in this section is given in Table 3.1. From this table we can see that the *offline-online stabilization* without supremizer is less expensive than the *offline-online stabilization* with supremizer. Therefore, from all the results presented in this section, we conclude that the *offline-online stabilization* is good enough to give a stable RB solution and we do not need supremizer enrichment. Because when we deal witht unstable FE pair since the added *offline-online stabilization* is sufficient in operating on the $\beta_{\text{inf}-\text{sup}}$ condition. Consequently, the computation cost can be reduced as shown in Table 3.1. We have not used the Empirical Interpolation Method (EIM) to approximate the nonlinear term. This will be the object of a future extension of the work to improve further computational performance.

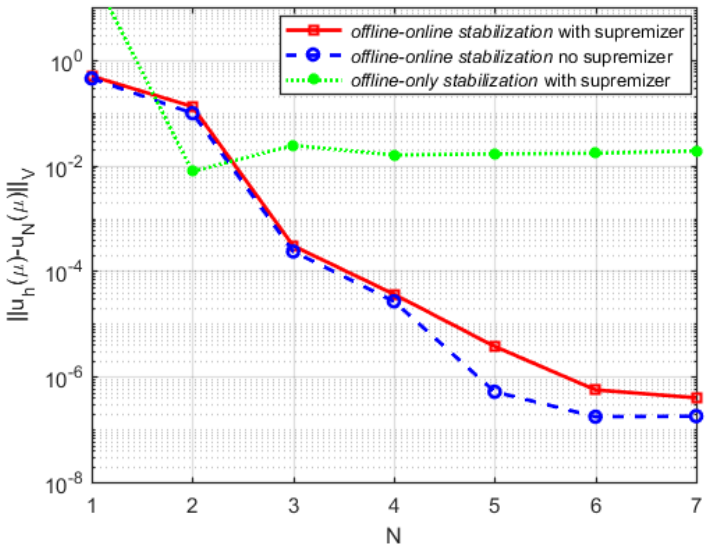


Figure 3.4: SUPG stabilization with physical parameterization on cavity flow: Comparison between FE and RB solution for velocity using $\mathbb{P}_1/\mathbb{P}_1$.

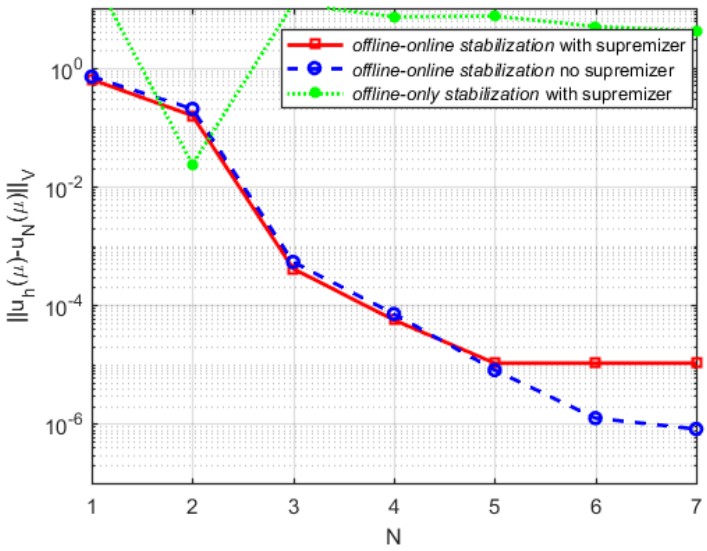


Figure 3.5: SUPG stabilization with physical parameterization on cavity flow: Comparison between FE and RB solution for velocity using $\mathbb{P}_2/\mathbb{P}_2$.

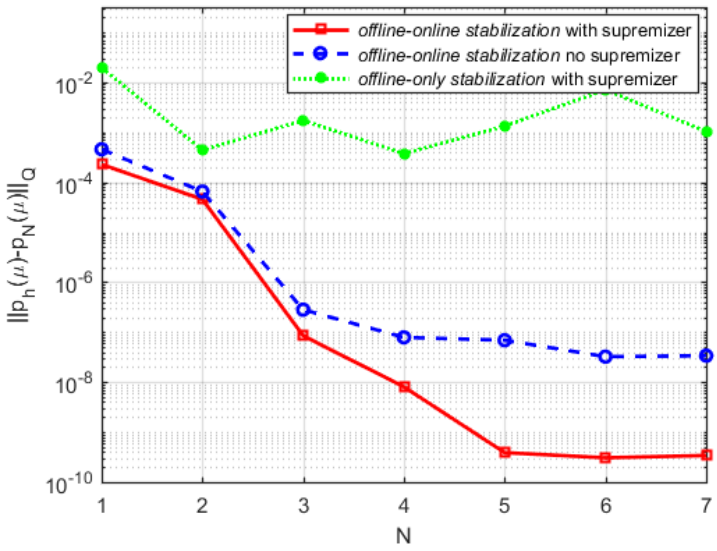


Figure 3.6: SUPG stabilization with physical parameterization on cavity flow: Comparison between FE and RB solution for pressure using $\mathbb{P}_1/\mathbb{P}_1$.

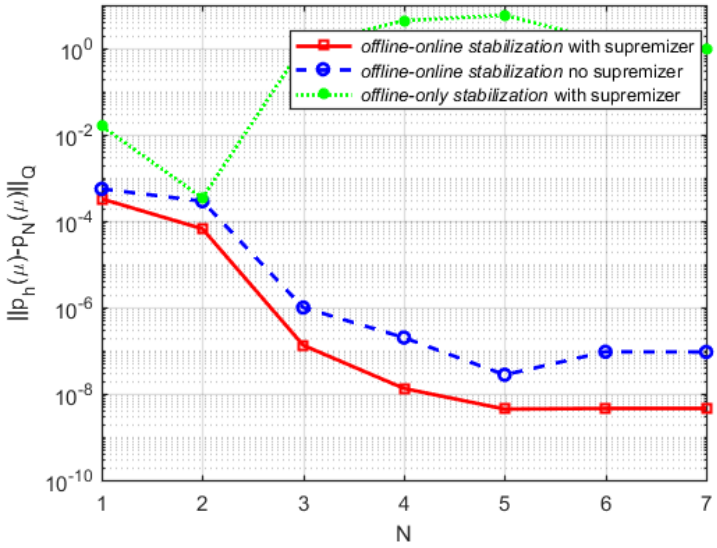


Figure 3.7: SUPG stabilization with physical parameterization on cavity flow: Comparison between FE and RB solution for pressure using $\mathbb{P}_2/\mathbb{P}_2$.

Physical parameter	μ (Reynolds number)
Range of μ	[100,500]
Online μ (example)	200
FE degrees of freedom	13218 ($\mathbb{P}_1/\mathbb{P}_1$) 52143 ($\mathbb{P}_2/\mathbb{P}_2$)
RB dimension	$N_u = N_s = N_p = 7$
Full *offline* time ($\mathbb{P}_1/\mathbb{P}_1$)	$1182s$ (*offline-online stabilization* with supremizer) $842s$ (*offline-online stabilization* without supremizer)
Full *offline* time ($\mathbb{P}_2/\mathbb{P}_2$)	$2387s$ (*offline-online stabilization* with supremizer) $2121s$ (*offline-online stabilization* without supremizer)
Full *Offline* time ($\mathbb{P}_2/\mathbb{P}_1$)	$2103s$ (with supremizer, no stabilization) $1823s$ (without supremizer, no stabilization)
Online time ($\mathbb{P}_1/\mathbb{P}_1$)	$74s$ (with supremizer) $65s$ (without supremizer)
Online time ($\mathbb{P}_2/\mathbb{P}_2$)	$131s$ (with supremizer) $108s$ (without supremizer)
Online time ($\mathbb{P}_2/\mathbb{P}_1$)	$120s$ (with supremizer, no stabilization) $99s$ (without supremizer, no stabilization)
N_{train}	25
N_{max}	25

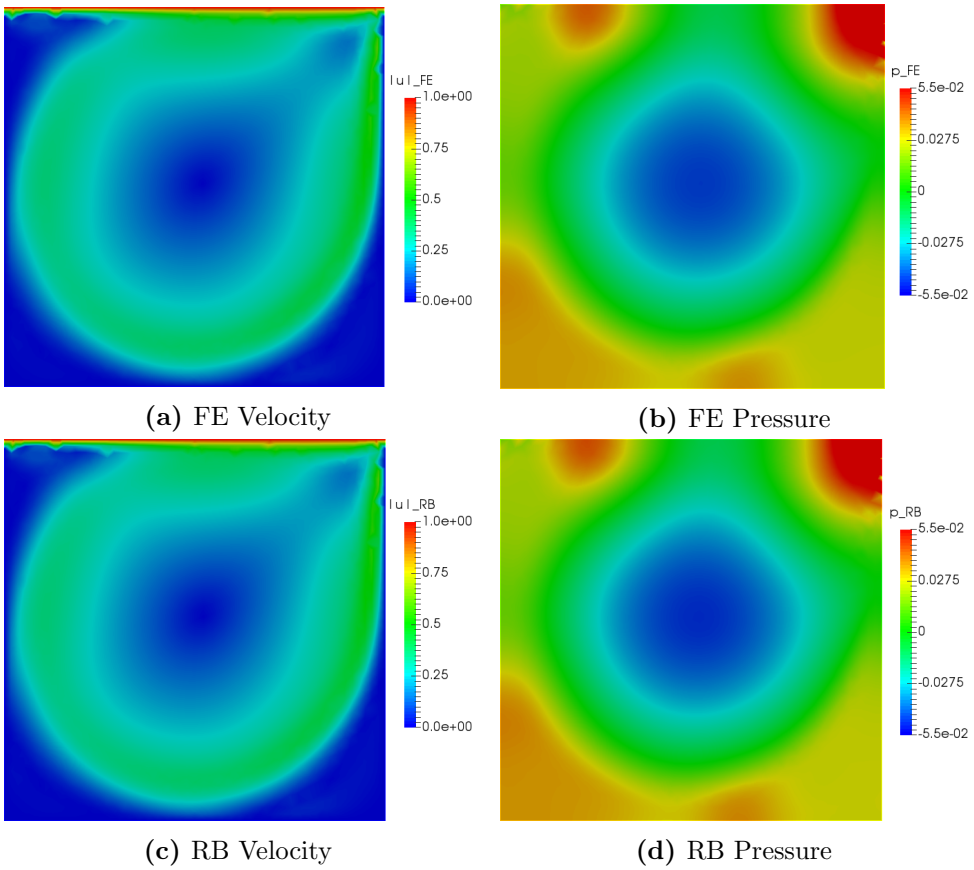
Table 3.1: Physical parameter only: Computational details of steady Navier-Stokes problem without EIM method.

Results for Taylor-Hood $\mathbb{P}_2/\mathbb{P}_1$: no stabilization

In this section we present a cavity flow test using inf-sup stable [132] FE pair $\mathbb{P}_2/\mathbb{P}_1$ for high Reynolds number. The goal of this section is give some numerical results for non-stabilized method with $\mathbb{P}_2/\mathbb{P}_1$ and compare these results, particularly computational cost with stabilized RB.

The range of physical parameter (Reynolds number) during the *offline* stage is [2500, 3500]. Figure 3.8 plots the comparison of FE and RB solutions for velocity and pressure, respectively. In these results the RB solution is obtained by enriching the RB velocity space with supremizer solutions. We see that the FE and RB solutions are similar.

Figures 3.9 and 3.10 plots the error between FE and RB solutions for velocity and pressure, respectively using $\mathbb{P}_2/\mathbb{P}_1$ with/without supremizer. From these Figures we conclude that in order to get a stable RB solution we need to enrich the velocity space with supremizer solutions. Because in this case we do not have any additional stabilization terms which guarantees the inf-sup stability. Therefore, supremizer is necessary in this case.



(a) FE Velocity **(b)** FE Pressure

(c) RB Velocity **(d)** RB Pressure

Figure 3.8: FE solution (top) and RB solution (bottom) for velocity and pressure at Re=2880 using $\mathbb{P}_2/\mathbb{P}_1$ with supremizer, respectively.

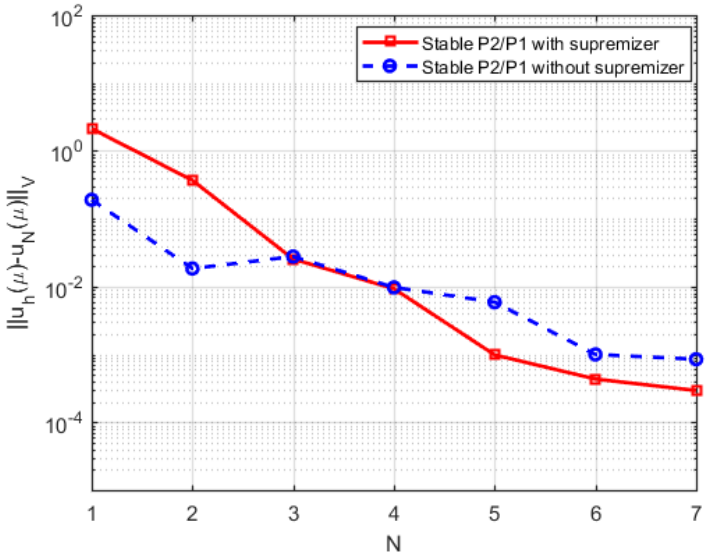


Figure 3.9: Cavity flow problem: Error between FE and RB solutions for velocity with/without supremizer using $\mathbb{P}_2/\mathbb{P}_1$.

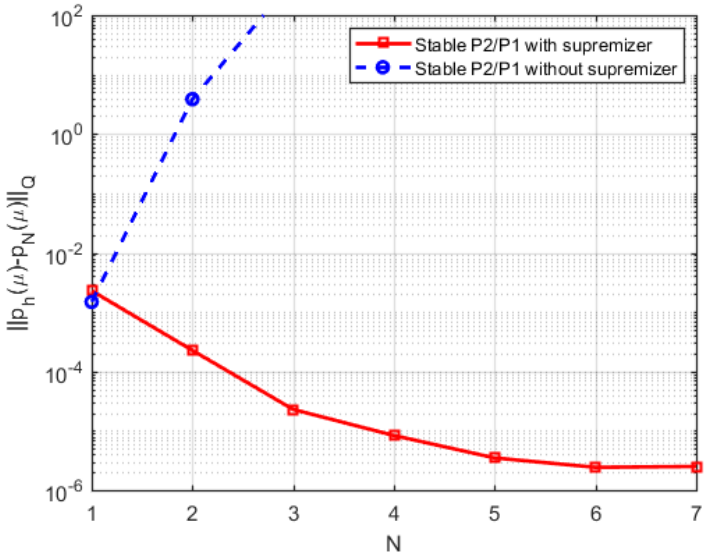


Figure 3.10: Cavity flow problem: Error between FE and RB solutions for pressure with/without supremizer using $\mathbb{P}_2/\mathbb{P}_1$.

3.2.2 Results for physical and geometrical parameterization

In this example, we apply the stabilized RB model for Navier-Stokes to the parametrized *lid driven-cavity* flow problem. In this case we have two parameters; μ_1 denotes the physical parameter (Reynolds number) and μ_2 represents the geometrical parameter (horizontal length of domain) and is shown in Fig. 3.11.

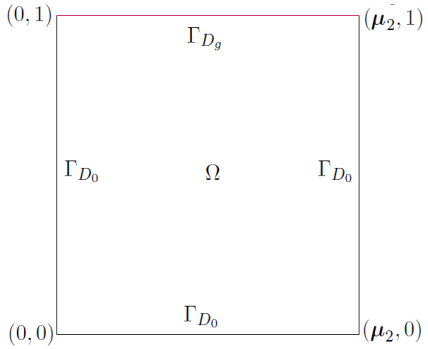


Figure 3.11: Parametrized domain Ω with boundaries marked.

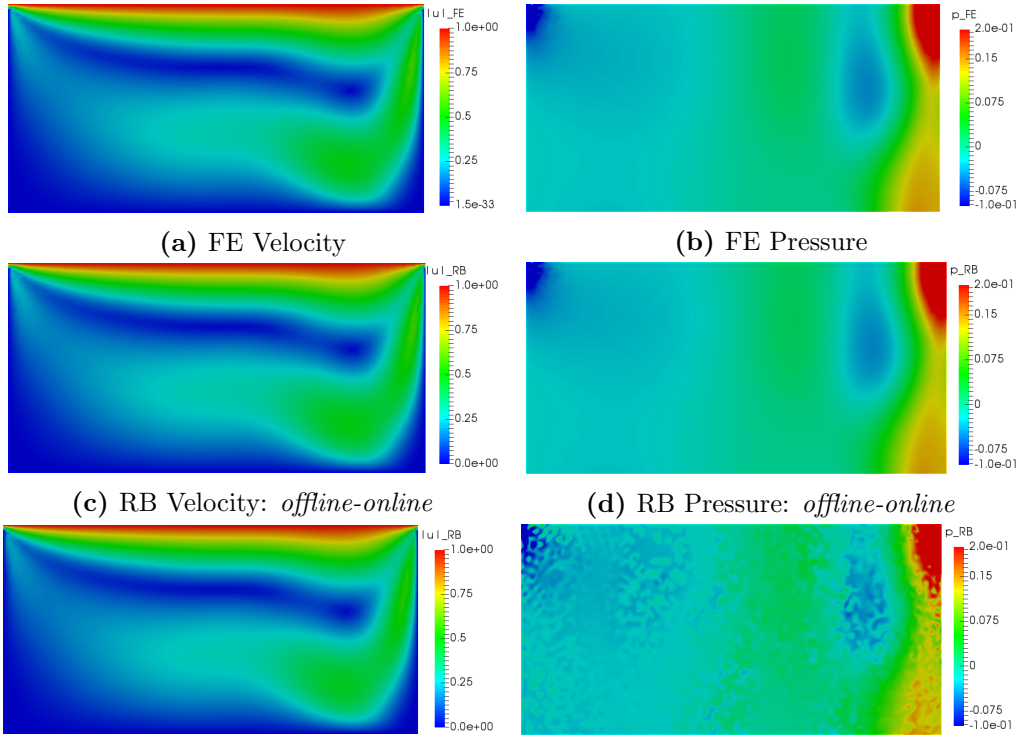
We consider the same boundary conditions defined in (3.25). Also the same mesh size is same as in previous example. Further detail of computations are summarized in Table 3.2.

In Fig. 3.12, we show the FE solution (top) for velocity and pressure; RB solution for velocity and pressure using the *offline-online stabilization* without supremizer (center) and the *offline-only stabilization* with supremizer (bottom). These results are carried out using $\mathbb{P}_1/\mathbb{P}_1$ FE pair (results are similar for $\mathbb{P}_2/\mathbb{P}_2$). From these results, we see that the FE solution and RB solution obtained by *offline-online stabilization* are same. But the RB pressure obtained by *offline-only stabilization* is not stable. We have tested various values of δ but the oscillations in pressure cannot be controlled for *offline-only stabilization*. The reason is that we are stabilizing only *offline* phase and not the online. In these snapshots we have chosen $\delta = 1.0$

Table 3.2 illustrates the computation details and the cost of stabilization options using both $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$ FE pairs. We also compare the computational cost of stable $\mathbb{P}_2/\mathbb{P}_1$ [132]. We see that *offline-only stabilization* method is less costly but on the other hand this option does not give a stable RB pressure (Fig. 3.12 (f)). Therefore, we rely on the other two options from which the *offline-online stabilization* without supremizer has less computation time than the *offline-online stabilization* with supremizer.

Figures 3.13 and 3.14 plots the error between FE and RB velocity obtained by three possible stabilization options using $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$ FE pair, respectively. From these results we conclude that the *offline-online stabilization* without supremizer has better performance as compared to other two options and is the most consistent stabilization option. Similarly we plot the error between FE and RB pressure in Figs. 3.15 and 3.16. We conclude that *offline-online stabilization* is necessary for RB pressure stability. Similar to one parameter case, the error for pressure decreases with the enrichment of

supremizer.



(a) FE Velocity

(b) FE Pressure

(c) RB Velocity: *offline-online*

(d) RB Pressure: *offline-online*

(e) RB Velocity: *offline-only*

(f) RB Pressure: *offline-only*

Figure 3.12: SUPG stabilization: From top to bottom; FE solution (top), RB solution using *offline-online stabilization* (center), RB solution using *offline-only stabilization* (bottom) for $(\mu_1, \mu_2) = (120, 2)$.

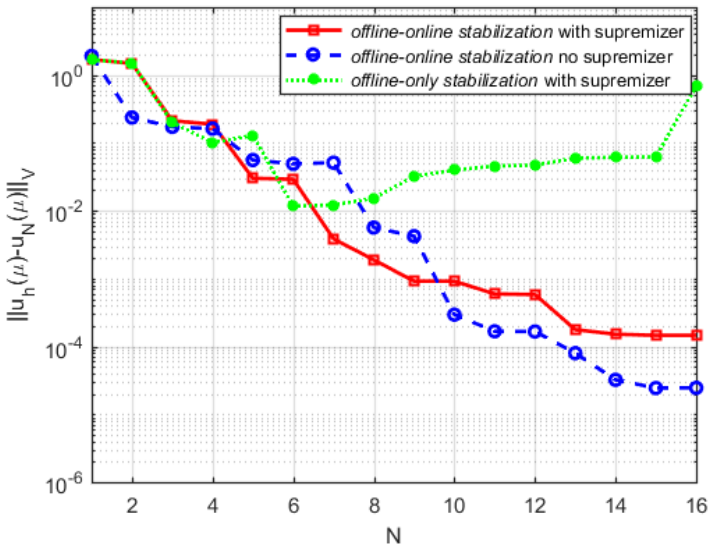


Figure 3.13: SUPG stabilization with geometrical and physical parametrization on cavity flow: Error between FE and RB solution for velocity using $\mathbb{P}_1/\mathbb{P}_1$.

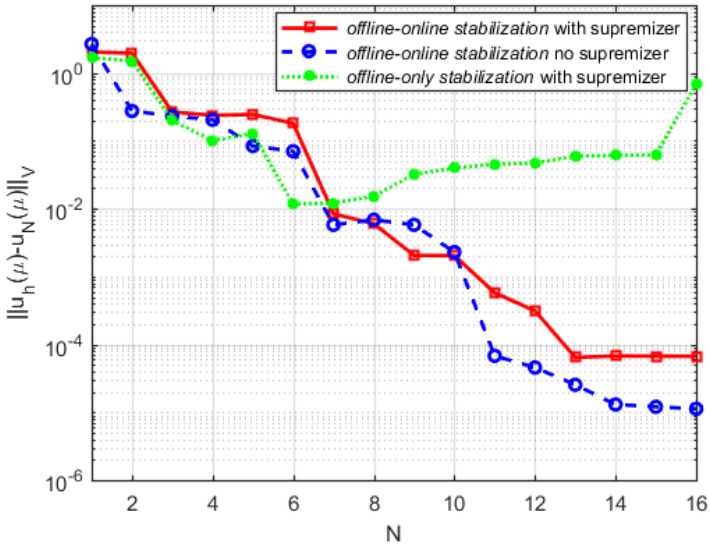


Figure 3.14: SUPG stabilization with geometrical and physical parametrization on cavity flow: Error between FE and RB solution for velocity using $\mathbb{P}_2/\mathbb{P}_2$.

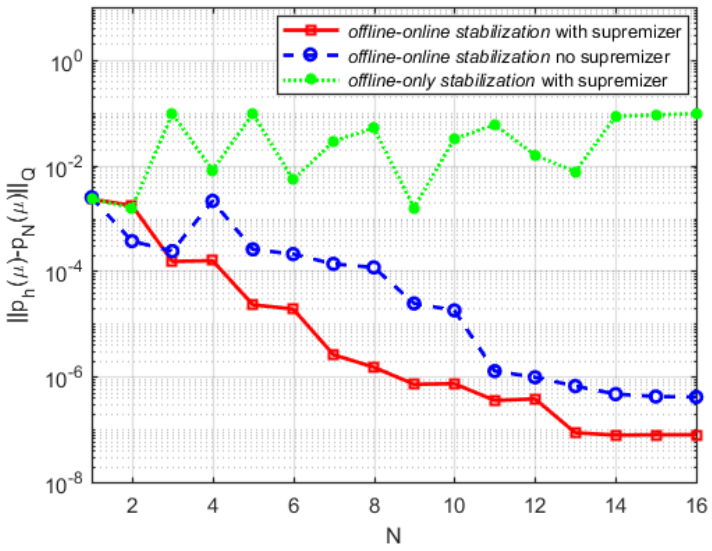


Figure 3.15: SUPG stabilization with geometrical and physical parametrization on cavity flow: Error between FE and RB solution for pressure using $\mathbb{P}_1/\mathbb{P}_1$.

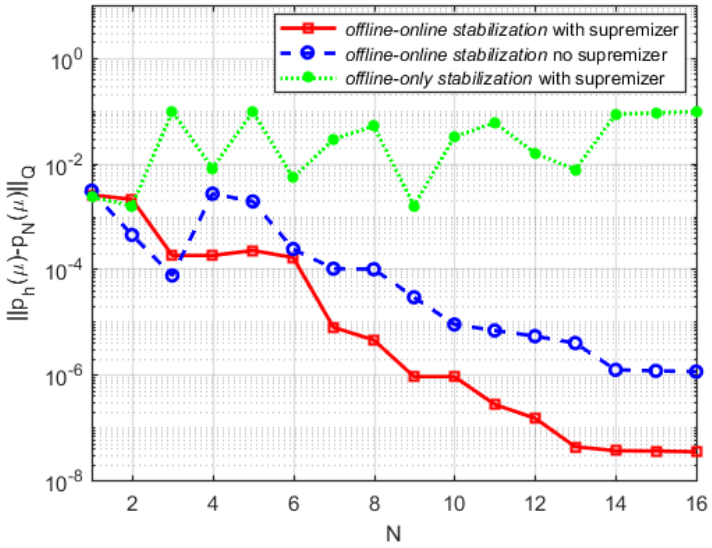


Figure 3.16: SUPG stabilization with geometrical and physical parametrization on cavity flow: Error between FE and RB solution for pressure using $\mathbb{P}_2/\mathbb{P}_2$.

Physical parameter	μ_1 (Reynolds number)
Geometrical parameter	μ_2 (horizontal length of domain)
Range of μ_1	[100,200]
Range of μ_2	[1.5,3]
μ_1 *online*	120
μ_2 *online*	2
FE degrees of freedom	11160 ($\mathbb{P}_1/\mathbb{P}_1$) 44091 ($\mathbb{P}_2/\mathbb{P}_2$)
RB dimension	$N_u = N_s = N_p = 16$
Computation time ($\mathbb{P}_2/\mathbb{P}_1$)	$3909s$ (*offline*), $195s$ (*online*) with supremizer
Offline time ($\mathbb{P}_1/\mathbb{P}_1$)	$2034s$ (*offline-online stabilization* with supremizer) $1920s$ (*offline-online stabilization* without supremizer) 649 (*offline-only stabilization* with supremizer)
Offline time ($\mathbb{P}_2/\mathbb{P}_2$)	$4885s$ (*offline-online stabilization* with supremizer) $4387ss$ (*offline-online stabilization* without supremizer) $1650s$ (*offline-only stabilization* with supremizer)
Online time ($\mathbb{P}_1/\mathbb{P}_1$)	$110s$ (with supremizer) $87s$ (without supremizer) $35s$ (*offline-only stabilization*)
Online time ($\mathbb{P}_2/\mathbb{P}_2$)	$242s$ (with supremizer) $180s$ (without supremizer) $90s$ (*offline-only stabilization*)

Table 3.2: Computational details for physical and geometrical parameters.

3.3 Parametrized unsteady Navier-Stokes problem

In this section, we develop a stabilized RB method using SUPG stabilization method for the approximation of unsteady Navier-Stokes problem in reduced order parametric setting. Let $\Omega \subset \mathbb{R}^2$, be a reference configuration, and we assume that current configuration $\Omega_o(\boldsymbol{\mu})$ can be obtained as the image of map $\boldsymbol{T}(.;\boldsymbol{\mu}) : \mathbb{R}^2 \to \mathbb{R}^2$, i.e. $\Omega_o(\boldsymbol{\mu}) = \boldsymbol{T}(\Omega;\boldsymbol{\mu})$. First we define the unsteady Navier-Stokes problem on a domain $\Omega_o(\boldsymbol{\mu})$ in $\mathbb{R}^2$. We consider the fluid flow in a region $\Omega_o(\boldsymbol{\mu})$, bounded by walls and driven by a body force $\boldsymbol{f}(\boldsymbol{\mu})$. The fluid velocity and pressure are the functions $\boldsymbol{u_o}(t;\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathbb{P}, 0 \leq t \leq T$ and $p_o(t;\boldsymbol{\mu})$ for $0 < t \leq T$, respectively which satisfies

$$\begin{cases} \dfrac{\partial \boldsymbol{u_o}}{\partial t} - \nu \Delta \boldsymbol{u}_o + \boldsymbol{u_o} \cdot \nabla \boldsymbol{u_o} + \nabla p_o = \boldsymbol{f}(\boldsymbol{\mu}) & \text{in } \Omega_o(\boldsymbol{\mu}) \times (0,T)\,, \\ \mathrm{div} \boldsymbol{u}_\mathrm{o} = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \times (0,T)\,, \\ \boldsymbol{u}_o = \boldsymbol{g} & \text{on } \partial\Omega \times (0,T)\,, \\ \boldsymbol{u}_o|_{t=0} = \boldsymbol{u_0} & \text{in } \Omega_o(\boldsymbol{\mu}). \end{cases} \tag{3.26}$$

Now we define the following notions to be used later

$$Q : L_0^2(\Omega) = \{q(x) \in L^2(\Omega) : \int_\Omega q(x)dx = 0\}, \tag{3.27}$$

$$L^2(0,T;H_0^1(\Omega)) = \{\boldsymbol{v}(x,t) : [0,T] \to H_0^1(\Omega) : \int_0^T ||\nabla \boldsymbol{v}||^2 dt \leq \infty, \tag{3.28}$$

$$L^\infty(0,T;L^2(\Omega)) = \{\boldsymbol{v}(x,t) : [0,T] \to L^2(\Omega) : ess \sup_{0<t<T} ||\boldsymbol{v}|| \leq \infty, \tag{3.29}$$

and

$$L^2(0,T;L_0^2(\Omega)) = \{q(x,t) : [0,T] \to L_0^2(\Omega) : \int_0^T ||q(t)||^2 dt \leq \infty. \tag{3.30}$$

By multiplying (3.26) with velocity and pressure test functions $\boldsymbol{v}$ and q, respectively, integrating by parts, and tracing everything back onto the reference domain Ω, we obtain the following parametrized weak formulation of (3.26):

for a given $\boldsymbol{\mu} \in \mathbb{P}$, find $\boldsymbol{u}(t;\boldsymbol{\mu}) \in \boldsymbol{V}$ and $p(t;\boldsymbol{\mu})) \in Q$ such that

$$\begin{cases} m(\dfrac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v};\boldsymbol{\mu}) + a(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) + c(\boldsymbol{u},\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) + b(\boldsymbol{v},p;\boldsymbol{\mu}) = F(\boldsymbol{v};\boldsymbol{\mu}) & \forall\, v \in \boldsymbol{V}, t > 0, \\ b(\boldsymbol{u},q;\boldsymbol{\mu}) = G(q;\boldsymbol{\mu}) & \forall\, q \in Q, t > 0, \\ \boldsymbol{u}|_{t=0} = \boldsymbol{u_0}, \end{cases} \tag{3.31}$$

where the bilinear forms and trilinear form are defined as:

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) &= \int_\Omega \frac{\partial \boldsymbol{u}}{\partial x_i}\kappa_{ij}(x;\boldsymbol{\mu})\frac{\partial \boldsymbol{v}}{\partial x_j}d\boldsymbol{x}, \qquad b(\boldsymbol{v},q;\boldsymbol{\mu}) = -\int_\Omega q\chi_{ij}(x;\boldsymbol{\mu})\frac{\partial v_j}{\partial x_i}d\boldsymbol{x}, \\ m(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\mu}) &= \int_\Omega \pi(\boldsymbol{x};\boldsymbol{\mu})\boldsymbol{u}_i\boldsymbol{v}_i d\boldsymbol{x}, \qquad c(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w};\boldsymbol{\mu}) = \int_\Omega u_i\chi_{ji}(x;\boldsymbol{\mu})\frac{\partial v_m}{\partial x_j}w_m d\boldsymbol{x}. \end{aligned} \tag{3.32}$$

The tensors $\boldsymbol{\kappa}$ and $\boldsymbol{\chi}$ are given by (3.5). The scalar π encoding both physical and geometrical parametrization are defined as:

$$\pi(\boldsymbol{x},\boldsymbol{\mu}) = |J_T(X;\boldsymbol{\mu})|, \tag{3.33}$$

where $J_T \in \mathbb{R}^{2\times 2}$ is the Jacobian matrix of the map $\boldsymbol{T}(.;\boldsymbol{\mu})$, and $|J_T|$ denotes the determinant.

3.3.1 Discrete Finite Element formulation

As in the previous chapter for unsteady Stokes problem, let us now discretize problem (3.31). Consider $\{T_h\}_{h>0}$ be the triangulations and h denotes a discretization parameter. Let $\boldsymbol{V}_h$ and Q_h be two finite dimensional spaces such that $\boldsymbol{V}_h \subset \boldsymbol{H}^1(\Omega)$ and $Q_h \subset L_0^2(\Omega)$. We use implicit Euler scheme for time derivative term. We consider a partition of the

interval $[0, T]$ into K sub-intervals of equal length $\Delta t = T/K$ and $t^k = k\Delta t, 0 \leq k \leq K$. We approximate the time derivative in the $(k) - th$ time layer as

$$\frac{\partial \boldsymbol{u}_h(t^k)}{\partial t} \approx \frac{\boldsymbol{u}_h^k - \boldsymbol{u}_h^{k-1}}{\Delta t}, \tag{3.34}$$

where Δt is a constant time step. We define the semi discrete FE approximation problem of (3.31) while using (3.34) in (3.31) we get as follows:
for a given $\boldsymbol{\mu} \in \mathbb{P}$, and $(\boldsymbol{u}_h^{k-1}(\boldsymbol{\mu}), p_h^{k-1}(\boldsymbol{\mu}))$, find $\boldsymbol{u}_h^k(t; \boldsymbol{\mu}) \in \boldsymbol{V}_h$ and $p_h^k(t; \boldsymbol{\mu}) \in Q_h$ such that

$$\begin{cases} \frac{1}{\Delta t} m(\boldsymbol{u}_h^k, \boldsymbol{v}_h; \boldsymbol{\mu}) + a(\boldsymbol{u}_h^k, \boldsymbol{v}_h; \boldsymbol{\mu}) + c(\boldsymbol{u}_h^k, \boldsymbol{u}_h^k, \boldsymbol{v}_h; \boldsymbol{\mu}) & \\ + b(\boldsymbol{v}_h, p_h^k; \boldsymbol{\mu}) = F(\boldsymbol{v}_h; \boldsymbol{\mu}) + \frac{1}{\Delta t} m(\boldsymbol{u}_h^{k-1}, \boldsymbol{v}_h; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h^k, q_h; \boldsymbol{\mu}) = G(q_h; \boldsymbol{\mu}) & \forall\, q_h \in Q_h, \\ \boldsymbol{u}_h^0 = \boldsymbol{u}_{0,h}. \end{cases} \tag{3.35}$$

Similar to what we did in steady Navier-Stokes in subsection 3.1.1, the algebraic formulation of (3.35) can be written as:

$$\begin{aligned} \begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} + A(\boldsymbol{\mu}) + C(\boldsymbol{u}(t^k; \boldsymbol{\mu}); \boldsymbol{\mu}) & B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^k; \boldsymbol{\mu}) \\ \boldsymbol{P}(t^k; \boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix} \\ + \begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^{k-1}; \boldsymbol{\mu}) \\ \boldsymbol{P}(t^{k-1}; \boldsymbol{\mu}) \end{bmatrix}, \end{aligned} \tag{3.36}$$

where the matrices and vectors are defined as:

$$\begin{aligned} &(M(\boldsymbol{\mu}))_{ij} = m(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_i^h; \boldsymbol{\mu}), \qquad (A(\boldsymbol{\mu}))_{ij} = a(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_i^h; \boldsymbol{\mu}), \\ &(B(\boldsymbol{\mu}))_{ki} = b(\boldsymbol{\phi}_i^h, \boldsymbol{\psi}_k^h; \boldsymbol{\mu}), \qquad (C(\boldsymbol{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu}))_{ij} = \sum_{m=1}^{\mathcal{N}_u} \boldsymbol{u}_h^m(t; \boldsymbol{\mu}) c(\boldsymbol{\phi}_m^h, \boldsymbol{\phi}_j^h, \boldsymbol{\phi}_i^h; \boldsymbol{\mu}), \\ &(\bar{\boldsymbol{f}}(\boldsymbol{\mu}))_i = F(\boldsymbol{\phi}_i^h; \boldsymbol{\mu}), \qquad (\bar{\boldsymbol{g}}(\boldsymbol{\mu}))_k = G(\boldsymbol{\psi}_k^h; \boldsymbol{\mu}), \end{aligned} \tag{3.37}$$

where $\boldsymbol{\phi}_i^h$ and $\boldsymbol{\psi}_j^h$, are the basis functions of $\boldsymbol{V}_h$ and Q_h respectively. As in previous cases, we impose the affine parametric dependence on these matrices and vectors and we skip the detail here.

3.3.2 Stabilized Finite Element formulation

In this section we give the stabilized formulation of time-dependent Navier-Stokes equations defined in previous section. We use the same SUPG stabilization method defined in subsection 3.1.2. We skip the detail and directly write the stabilized algebraic formulation of (3.36).

$$\begin{aligned}\begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} + A(\boldsymbol{\mu}) + \tilde{C}(\boldsymbol{u}(t^k;\boldsymbol{\mu});\boldsymbol{\mu}) & \tilde{B}^T(\boldsymbol{\mu}) \\ \tilde{B}(\boldsymbol{\mu}) + \frac{\tilde{M}(\boldsymbol{\mu})}{\Delta t} & -S(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \tilde{\bar{\boldsymbol{g}}}(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \frac{\tilde{M}(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}(t^{k-1};\boldsymbol{\mu}) \end{bmatrix},\end{aligned} \tag{3.38}$$

where $\tilde{B}, \tilde{B}^T, \tilde{M}$ and $\tilde{C}$, are the sum of original matrices in formulation (3.36) and the SUPG stabilization matrices. Similarly $\tilde{\bar{\boldsymbol{f}}}$ and $\tilde{\bar{\boldsymbol{g}}}$ are vectors on right hand side which are sum of original vectors in formulation (3.36) and SUPG stabilization terms [47].

3.3.3 Reduced Basis formulation

Similar to subsection 2.3.3 in Chapter 2 for unsteady Stokes problem, a reduced order approximation of velocity and pressure field is obtained by means of Galerkin projection on the RB spaces $\boldsymbol{V}_N, Q_N$ and $\tilde{\boldsymbol{V}}_N$, defined in (2.47), (2.48) and (2.49), respectively.

In the *online* stage, the resulting reduced order approximation of (3.36) for any $\boldsymbol{\mu} \in \mathbb{P}$ is as follows:

$$\begin{aligned}\begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} + A_N(\boldsymbol{\mu}) + C_N(\boldsymbol{u}_N(t^k;\boldsymbol{\mu});\boldsymbol{\mu}) & B_N^T(\boldsymbol{\mu}) \\ B_N(\boldsymbol{\mu}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^k;\boldsymbol{\mu}) \end{bmatrix} &= \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^{k-1};\boldsymbol{\mu}) \end{bmatrix},\end{aligned} \tag{3.39}$$

where, similar to what we did in previous chapter, the reduced order matrices are defined as:

$$\begin{aligned} M_N(\boldsymbol{\mu}) = Z_{u,s}^T M(\boldsymbol{\mu})_{u,s}, \quad A_N(\boldsymbol{\mu}) = Z_{u,s}^T A(\boldsymbol{\mu}) Z_{u,s}, \quad B_N(\boldsymbol{\mu}) = Z_p^T B(\boldsymbol{\mu}) Z_u, \\ C_N(.;\boldsymbol{\mu}) = Z_u^T C(.;\boldsymbol{\mu}) Z_u, \quad \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) = Z_u^T \bar{\boldsymbol{f}}(\boldsymbol{\mu}), \quad \bar{\boldsymbol{g}}_N(\boldsymbol{\mu}) = Z_p^T \bar{\boldsymbol{g}}(\boldsymbol{\mu}). \end{aligned} \tag{3.40}$$

3.3.4 Stabilized Reduced Basis formulation

In this section we define the stabilized formulation of (3.39). We skip the detail and write directly the stabilized algebraic formulation for RB problem derived from the stabilized FE formulation (3.38) for any $\boldsymbol{\mu} \in \mathbb{P}$ as follows:

$$\begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} + A_N(\boldsymbol{\mu}) + \tilde{C}_N(\boldsymbol{u}_N(t^k;\boldsymbol{\mu});\boldsymbol{\mu}) & \tilde{B}_N^T(\boldsymbol{\mu}) \\ \tilde{B}_N(\boldsymbol{\mu}) + \frac{\tilde{M}_N(\boldsymbol{\mu})}{\Delta t} & -S_N(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^k;\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^k;\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_N(\boldsymbol{\mu}) \\ \bar{\tilde{\boldsymbol{g}}}_N(\boldsymbol{\mu}) \end{bmatrix} + \begin{bmatrix} \frac{M_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \\ \frac{\tilde{M}_N(\boldsymbol{\mu})}{\Delta t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(t^{k-1};\boldsymbol{\mu}) \\ \boldsymbol{P}_N(t^{k-1};\boldsymbol{\mu}) \end{bmatrix}, \tag{3.41}$$

where $\tilde{B}_N, \tilde{B}_N^T, \tilde{M}_N$ and $\tilde{C}_N$ are RB stabilized matrices, and can be obtained similarly as (3.40).

We follow the same pattern as we did for Stokes problem in Chapter 2 and steady Navier-Stokes problem in this chapter, i.e, we discuss and compare the different stabilization options using unstable FE pair $\mathbb{P}_k/\mathbb{P}_k$.

3.4 Numerical results and discussion

In this section we apply the stabilized RB model for unsteady Navier-Stokes problem presented in section 3.3 and subsections therein to *lid-driven cavity* flow problem. Similar to what we did in section 3.2, we first show some numerical results for only physical parameterization in subsection 3.4.1 and then we show numerical results for both physical and geometrical parametrization in subsection 3.4.2.In both cases we discuss the three options (*i*) *offline-online stabilization* with supremizer, (*ii*) *offline-online stabilization* without supremizer, (*iii*) *offline-only stabilization* with supremizer.

3.4.1 Results for physical parameter case only

In this example, we show some numerical results of stabilized RB model applied to unsteady Navier-Stokes problem. The parameter in this case is only the physical parameter, i.e, the Reynolds number and is denoted by $\boldsymbol{\mu}$. Computational domain is shown in Fig. 3.1. The details of computation is summarized in Table 3.3.

Physical parameter	μ (Reynolds number)
Range of μ	[100,200]
Online μ (example)	130
FE degrees of freedom	5934 ($\mathbb{P}_1/\mathbb{P}_1$)
RB dimension	$N_u = N_s = N_p = 30$
Offline time ($\mathbb{P}_1/\mathbb{P}_1$)	$40612s$ (*offline-online stabilization* with supremizer) $38781s$ (*offline-online stabilization* without supremizer)
Online time ($\mathbb{P}_1/\mathbb{P}_1$)	$4640s$ (with supremizer) $4040s$ (without supremizer)
Time step	0.02
Final time	0.5

Table 3.3: Physical parameter only: Computational details of unsteady Navier-Stokes problem without Empirical Interpolation.

Figures 3.17 and 3.18 plots the L^2-error in time for velocity using $\mathbb{P}_1/\mathbb{P}_1$ and $\mathbb{P}_2/\mathbb{P}_2$ FE pair, respectively. Similarly in Figs. 3.19 and 3.20 we show the L^2-error in time for pressure. In all numerical results presented in this section, we observe that the *offline-online stabilization* without supremizer has better performance for velocity in terms of error. However, in case of pressure, our results show that supremizer is still improving the error but on the other hand addition of supremizer is computationally expensive. The *offline-only stabilization* is not accurate also in this case.

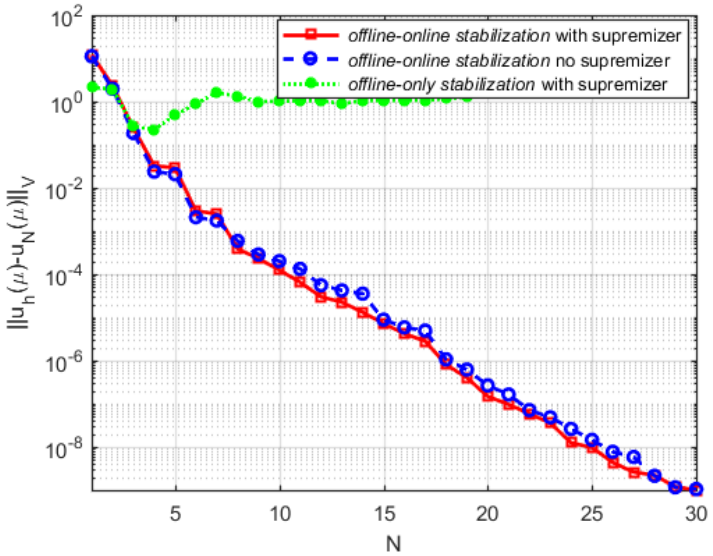


Figure 3.17: SUPG stabilization with physical parametrization on cavity flow: Error between FE and RB solution for velocity using $\mathbb{P}_1/\mathbb{P}_1$.

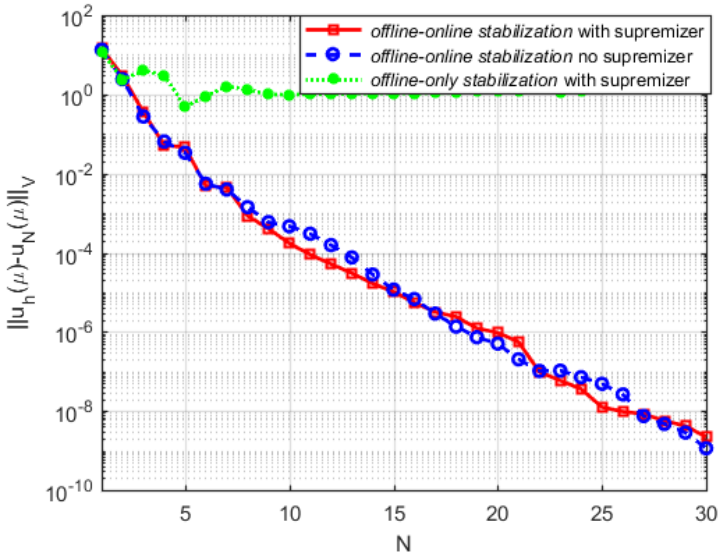


Figure 3.18: SUPG stabilization with physical parametrization on cavity flow: Error between FE and RB solution for velocity using $\mathbb{P}_2/\mathbb{P}_2$.

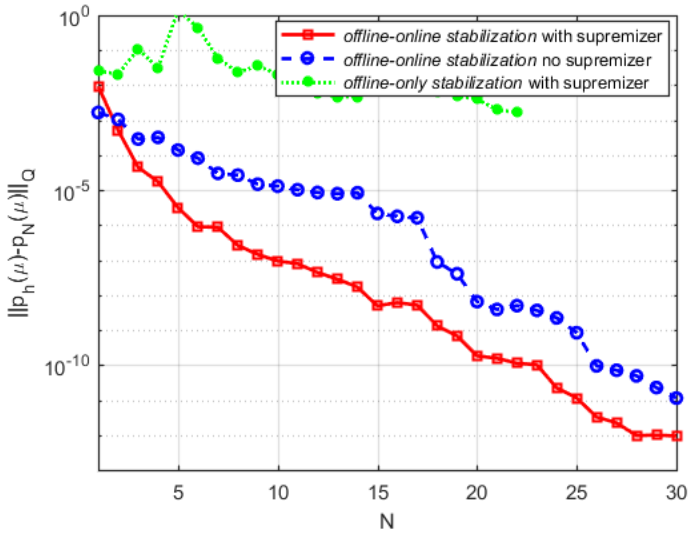


Figure 3.19: SUPG stabilization with physical parametrization on cavity flow: Error between FE and RB solution for pressure using $\mathbb{P}_1/\mathbb{P}_1$.

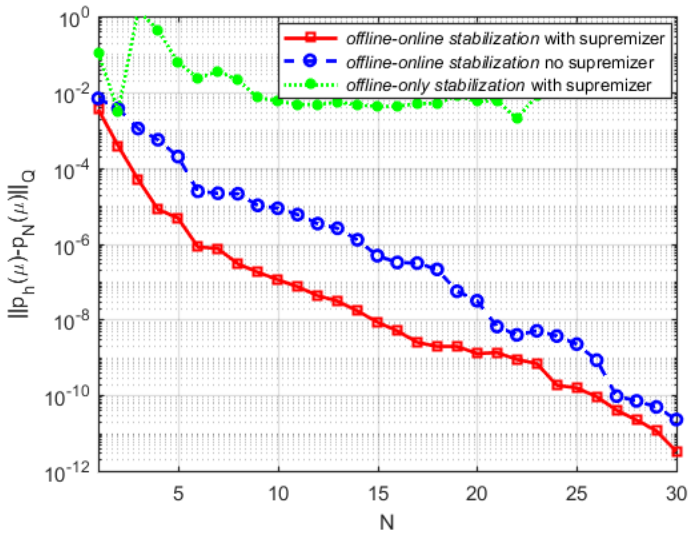


Figure 3.20: SUPG stabilization with physical parametrization on cavity flow: Error between FE and RB solution for pressure using $\mathbb{P}_2/\mathbb{P}_2$.

3.4.2 Results for physical and geometrical parameters

In this section we present some numerical results for unsteady Navier Stokes problem with physical and geometrical parameters. Parametrized domain is shown in Fig. 3.11 and the computation details are presented in Table 3.4. We recall that we are not using any "hyper-reducton" technique to improve online performance at the moment. Our interest at the moment is accuracy and stability.

Figure 3.21 illustrates the error between FE and RB solution for velocity. We observe that the error between two solutions, obtained by using *offline-online stabilization* with/without supremizer is negligible. Similarly in Fig. 3.22 we show the error between FE and RB solutions for pressure.

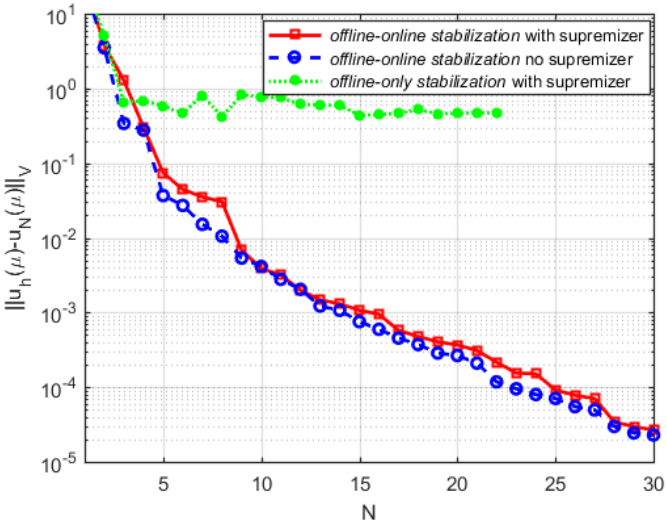


Figure 3.21: SUPG stabilization using $\mathbb{P}_1/\mathbb{P}_1$: Velocity error physical and geometrical parameters on cavity flow.

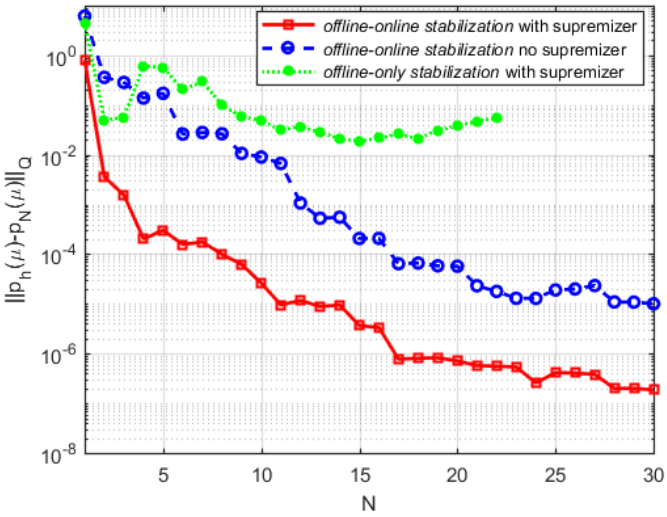


Figure 3.22: SUPG stabilization using $\mathbb{P}_1/\mathbb{P}_1$: Pressure error physical and geometrical parameters on cavity flow.

Physical parameter	μ_1 (Reynolds number)
Geometrical parameter	μ_2 (horizontal length of domain)
Range of μ_1	[100,200]
Range of μ_2	[1.5,3]
μ_1 *online* (example)	130
μ_2 *online* (example)	2
FE degrees of freedom	6222 ($\mathbb{P}_1/\mathbb{P}_1$)
RB dimension	$N_u = N_s = N_p = 30$
Offline time ($\mathbb{P}_1/\mathbb{P}_1$)	$44693s$ (*offline-online stabilization* with supremizer) $40153s$ (*offline-online stabilization* without supremizer)
Online time ($\mathbb{P}_1/\mathbb{P}_1$)	$5169s$ (with supremizer) $4724s$ (without supremizer)
Time step	0.02
Final time	0.5

Table 3.4: Computational details for unsteady Navier-Stokes problem with physical and geometrical parameters: stabilization and computational reduction.

3.5 Concluding remarks

In this chapter we have developed the stabilized RB method for the approximation of parametrized Navier-Stokes problem. We have considered both steady and unsteady cases as we did in previous chapter for the parametrized Stokes problem. The RB formulation is build, using the classical residual based stabilization technique SUPG in full order during the *offline* stage and, then, projecting on the RB space. We have compared our approach [2] with the existing approaches based on supremizers [132] through a series of numerical experiments. For instance, the comparison between *offline-online stabilization* with/without supremizer and *offline-only stabilization* for steady and unsteady Navier-Stokes problems is presented. Our results in this chapter are consistent with those of the Stokes case presented in Chapter 2. On the basis of numerical results the main observations are as follows:

- *offline-online stabilization* is the most appropriate way to perform RB stabilization (if needed) for both steady and unsteady Navier-Stokes problems;
- in case of both steady and unsteady Navier-Stokes problems using SUPG stabilization, velocity is still better using *offline-online stabilization* (without supremizer) even if pressure is improved by the supremizer enrichment;
- *offline-only stabilization* is not consistent, i.e, we loss the consistency of problem if we solve the stabilized system during the *offline* stage and non-stabilized system during the *online* stage;

- we can still improve the computational performances by using the Empirical Interpolation Method (EIM) [13];
- in terms of CPU time, the Taylor-Hood FE pair ($\mathbb{P}_2/\mathbb{P}_1$) is more expensive than ($\mathbb{P}_1/\mathbb{P}_1$) stabilized but less expensive than ($\mathbb{P}_2/\mathbb{P}_2$) stabilized (see Table 3.1);
- all the results presented in this chapter are consistent with results of Chapter 2 for all stabilization options, in terms of stability and also in terms of computational performances using $\mathbb{P}_k/\mathbb{P}_k$.

Chapter 4

An Online Stabilization Method for Parametrized Stokes Problems

In this chapter we propose a new online stabilization strategy for RB approximation of parametrized Stokes problem. In this strategy, online solution is improved by a post processing based on rectification method [37]. Moreover, this stabilization strategy is different from residual based stabilization, discussed in previous chapters. We compare this approach with *offline-online stabilization* approach [2]. We extend the rectification method presented in [37, 62, 94] to the Stokes problem. All the numerical simulations in this chapter are carried out using RBniCS [10, 63], an open-source reduced order modelling library, built on top of FEniCS [87].

This chapter is organized as follows: in section 4.1, first we define advection-diffusion problem and overview of SUPG stabilization method for advection dominated case. Then we review the vanishing viscosity method [94] and we give a brief overview of rectification method applied to advection dominated problem. Finally we present some numerical results and discussions.

In section 4.2 we present the rectification method for Stokes problem, for which we also recall the formulation of Stokes problem from Chapter 2. After the formulation we present some numerical tests on a parametrized cavity to check the validity of rectification method. Finally, in section 4.3 we conclude the main findings of this chapter.

4.1 Rectification method for advection-diffusion problem

In this section we give a brief review of the paper by Maday et al. [94] to recall the concepts of vanishing viscosity and rectification for the case of scalar advection-diffusion problem, before we move to the Stokes problem. We start with the definition of scalar advection-diffusion problem as:

$$\begin{cases} L(\mu)u := -\mu\Delta u + \mathbf{b}\cdot\nabla u = f & \text{in } \Omega = (0,1)^2, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{4.1}$$

where $\mu \in [10^{-6}, 1]$ denotes the diffusion coefficient[1] and $\mathbf{b} = (1,1)^T$, the constant transport field. The SUPG-stabilization method in the offline stage to get the basis matrix Z is implemented. The weak form of problem (4.1) is: for any $\mu \in \mathbb{P}$, find $u(\mu) \in V$ such that

$$a(u(\mu), v; \mu) = f(v) \qquad \forall v \in V, \tag{4.2}$$

where $V = H_0^1(\Omega)$ and

$$a(u(\mu), v; \mu) = \mu \int_\Omega \nabla u \cdot \nabla v d\boldsymbol{x} + \int_\Omega \boldsymbol{b} \cdot \nabla u v d\boldsymbol{x}, \qquad f(v) = \int_\Omega f v d\boldsymbol{x} \tag{4.3}$$

a continuous and coercive bilinear form, and a linear and continuous functional, respectively. Introducing a *high-fidelity* space $V_h \subset V$ of dimension $\mathcal{N}$. The *high-fidelity* solution to problem (4.3) obtained by Galerkin-FE method reads: for any $\mu \in \mathbb{P}$, find $u_h(\mu) \in V_h$ such that

$$a(u_h(\mu), v_h; \mu) = f(v_h) \qquad \forall v_h \in V_h. \tag{4.4}$$

When dealing with advection-dominated, i.e, for $\frac{|\boldsymbol{b}|}{\mu} \ll 1$, solution to (4.4) yields numerical oscillations unless a suitable stabilization technique is introduced. Therefore, in this case SUPG [29] is applied. The stabilized formulation of (4.4) reads:

$$a_{stab}(u_h(\mu), v_h; \mu) = f_{stab}(v_h) \qquad \forall v_h \in V_h, \tag{4.5}$$

where $a_{stab}(.,.;\mu)$ and $f_{stab}(.)$ are bilinear and linear forms including the stabilization terms defined as:

$$\begin{aligned} a_{stab}(u_h(\mu), v_h; \mu) &= a(u_h(\mu), v_h; \mu) + s(u_h(\mu), v_h; \mu), \\ f_{stab}(.) &= f(v_h) + f_s(v_h), \end{aligned} \tag{4.6}$$

being

$$\begin{aligned} s(u_h(\mu), v_h; \mu) &= \sum_{K \in \tau_h} (L(\mu)u_h, \delta_K L_{SS} v_h)_{L^2(K)}, \\ f_s(v_h) &= \sum_{K \in \tau_h} (f, \delta_K L_{SS} v_h)_{L^2(K)}, \end{aligned} \tag{4.7}$$

where $L_{SS}u = \boldsymbol{b} \cdot \nabla u$ is skew symmetric part of operator L and $\delta_K > 0$ a suitable stabilization coefficient. Algebraic formulation of (4.6) can be written as:

$$A_{stab}(\mu)\boldsymbol{u}_h(\mu) = F_{stab}, \tag{4.8}$$

where

$$A_{stab}(\mu) = A(\mu) + S(\mu), \qquad F_{stab} = F + F_s, \tag{4.9}$$

[1] μ is the only parameter we have, therefore we drop the bold symbol

being $\boldsymbol{u}_h(\mu) \in \mathbb{R}^{\mathcal{N}}$ the vectors whose components are the degrees of freedom of $u_h(\mu)$ and for $i, j = 1, ..., \mathcal{N}$

$$\begin{aligned} (A(\mu))_{ij} &= a(\phi_j, \phi_i; \mu), \qquad (S(\mu))_{ij} = s(\phi_j, \phi_i; \mu), \\ (F)_i &= f(\phi_i), \qquad (F_s)_i = f_s(\phi_i), \end{aligned} \tag{4.10}$$

where $\{\phi\}_{i=1}^{\mathcal{N}}$ denote the set of (Lagrange) basis functions on V_h.

Now introducing a low dimensional subspace V_N of dimension N, where $N \ll \mathcal{N}$ and V_N is built from a set of *high-fidelity* solutions (snapshots) computed for properly selected parameter values [63, 113], i.e,

$$V_N = \text{span}\{u_h(\mu^n) | 1 \leq n \leq N\} \subset V_h, \tag{4.11}$$

The RB is obtained by Galerkin-projection onto V_N and reads as follows: for any $\mu \in \mathbb{P}$ find $u_N(\mu) \in V_N$ such that

$$a(u_N(\mu), v_N; \mu) = f(v_N) \qquad \forall v_N \in V_N. \tag{4.12}$$

For advection-dominated case *offline-only stabilization* is not stable and shows spurious oscillations.

On the other hand, performing a Galerkin projection of the stabilized problem (4.5) onto V_N using a stabilized RB formulation yields stable RB approximation and it reads: for any $\mu \in \mathbb{P}$ find $u_N(\mu) \in V_N$ such that

$$a_{stab}(u_N(\mu), v_N; \mu) = f_{stab}(v_N) \qquad \forall v_N \in V_N. \tag{4.13}$$

Algebraically, the RB approximation for SUPG case is the solution of following system:

$$A_N^{stab}(\mu)\boldsymbol{u}_N(\mu) = F_N^{stab}, \tag{4.14}$$

where

$$A_N^{stab}(\mu) = A_N(\mu) + A_N^{SUPG}(\mu), \qquad F_N^{stab} = F_N + F_N^{SUPG}. \tag{4.15}$$

These RB matrices are obtained as:

$$\begin{aligned} A_N(\mu) &= Z^T A_h(\mu) Z, \qquad A_N^{SUPG}(\mu) = Z^T A_h^{SUPG}(\mu) Z, \\ F_N &= Z^T F_h, \qquad F_N^{SUPG} = Z^T F_h^{SUPG}, \end{aligned} \tag{4.16}$$

where $Z \in \mathbb{R}^{\mathcal{N} \times N}$ is the basis matrix, such that $Z = [\xi_1 | ... | \xi_N]$.

4.1.1 Online vanishing viscosity stabilization method

The vanishing viscosity method proposed by Maday et al. [97] is used to stabilize the RB problem independently of the stabilization procedure operated on the FE approximation, provided a set of stable RB functions has been computed offline. This method adds a suitable diffusion term on the RB problem, that depends on N and vanishes on the lower modes; i.e, higher the mode, the stronger is the added stabilization. Also the RB space

is now constructed by rotating the orthonormal reduced basis Z by the matrix W of the eigenvectors of the reduced diffusion operator: Let $K_N, M_N \in \mathbb{R}^{N\times N}$ denotes the reduced stiffness and mass operators, respectively, i.e.,

$$(K_N)_{mn} = \int_\Omega \nabla\xi_n \cdot \nabla\xi_m d\boldsymbol{x}, \quad (M_N)_{mn} = \int_\Omega \xi_n\xi_m d\boldsymbol{x}, \quad m,n = 1,...,N, \tag{4.17}$$

obtained from full order matrices as $K_N = Z^T K_h Z$, $M_N = Z^T M_h Z$, where

$$(K_h)_{ij} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i d\boldsymbol{x}, \quad (M_h)_{ij} = \int_\Omega \phi_i\phi_j d\boldsymbol{x}, \quad i,j = 1,...,\mathcal{N}. \tag{4.18}$$

Solving the following generalized eigenvalue problem:

$$K_N\mathbf{w}_j = \lambda_j M_N\mathbf{w}_j,\ j = 1,..,N, \tag{4.19}$$

to get the matrix of eigenvectors $W = [\mathbf{w}_1|...|\mathbf{w}_N] \in \mathbb{R}^{N\times N}$. Then we rotate the columns of basis matrix Z by W to get the transformed basis $\bar{Z} = ZW$. Finally the new RB space denoted by $\bar{V}_N$ is obtained as the span of new basis functions, i.e., $\bar{V}_N$=span$\{\bar{\xi}_1,...,\bar{\xi}_N\}$

Consider now the following RB vanishing viscosity (RB-VV) approximation: find $\bar{u}_N(\mu) \in \bar{V}_N$ such that

$$a(\bar{u}_N(\mu), v_N; \mu) + d_N(\bar{u}_N(\mu), v_N) = F(v_N) \quad \forall v_N \in \bar{V}_N, \tag{4.20}$$

where $d_N(.,.)$ is an additional viscosity term, whose action on each pair of basis functions $(\bar{\xi}_m, \bar{\xi}_n)$, $m,n = 1,...,N$ is such that

$$d_N(\bar{\xi}_m, \bar{\xi}_n) = f(\lambda_n)\int_\Omega \nabla\bar{\xi}_n \cdot \nabla\bar{\xi}_m d\boldsymbol{x}, \tag{4.21}$$

with $f(\lambda_n)$ to be properly defined [94]. In the simpler case $f(\lambda_n) = c \in \mathbb{R}^+$, where $c = \frac{\nu}{\lambda_N}$ (ratio between desired added viscosity ν and the largest eigenvalue λ_N) i.e., a viscosity contribution is added at each mode. Algebraic formulation of (4.20) can be written as:

$$(\bar{A}_N(\mu) + S_N)\boldsymbol{u}_N(\mu) = \bar{F}_N, \tag{4.22}$$

where $\bar{A}_N = \bar{Z}^T A_h \bar{Z}$, $\bar{F}_N = \bar{Z}^T F_h$, and $S_N \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose generic element is given by $(S_N)_{mn} = d_N(\bar{\xi}_m, \bar{\xi}_n)$.

4.1.2 Rectification method

After solving the problem (4.20), a further post-processing based on a rectification method [36–38, 62] is applied to improve the accuracy of solution. In other words, this rectification method is used to correct the consistency error of RB-VV approximation

$$\bar{u}_N(\boldsymbol{\mu}) = \textstyle\sum_{k=1}^N \alpha_k(\boldsymbol{\mu})\bar{\xi}_k,$$

i.e., the fact that

$$\bar{u}_N(\boldsymbol{\mu}^i) \neq u_h(\boldsymbol{\mu}^i) \quad \forall \boldsymbol{\mu}^i \in S_N = \{\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^N\}.$$

In order to cure this issue, an alternative linear combination of the reduced basis functions has been chosen.

We start by computing the RB Galerkin approximations for all values $\boldsymbol{\mu} = \boldsymbol{\mu}^i$; $i = 1, ..., N$ which gives the coefficients $\bar{u}_N(\boldsymbol{\mu}^i) = \sum_{k=1}^N \alpha_k(\boldsymbol{\mu}^i)\bar{\xi}_k$. We define the matrix B_R with coefficients α_k^i, i.e.,

$$R_N = \begin{pmatrix} \alpha_1(\boldsymbol{\mu}^1) & . & . & . & \alpha_1(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \alpha_N(\boldsymbol{\mu}^1) & . & . & . & \alpha_N(\boldsymbol{\mu}^N) \end{pmatrix}. \tag{4.23}$$

We also express the N snapshots over the reduced basis which gives the coefficients $u_h(\boldsymbol{\mu}^i) = \sum_{j=1}^N \beta_j(\boldsymbol{\mu}^i)\bar{\xi}_j$, from which we define the matrix B of coefficients β_j^i, i.e.,

$$R = \begin{pmatrix} \beta_1(\boldsymbol{\mu}^1) & . & . & . & \beta_1(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \beta_N(\boldsymbol{\mu}^1) & . & . & . & \beta_N(\boldsymbol{\mu}^N) \end{pmatrix}. \tag{4.24}$$

We set $S = RR_N^{-1}$ done in the offline stage and the matrix is stored.

Finally, the rectified solution $\bar{u}_N^r(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathbb{P}$ is computed online by using the new coefficients $\boldsymbol{\alpha}_{new} = S\boldsymbol{\alpha}$, i.e.,

$$\bar{u}_N^r(\boldsymbol{\mu}) = \sum_{j=1}^N \alpha_{new,j}(\boldsymbol{\mu})\bar{\xi}_j. \tag{4.25}$$

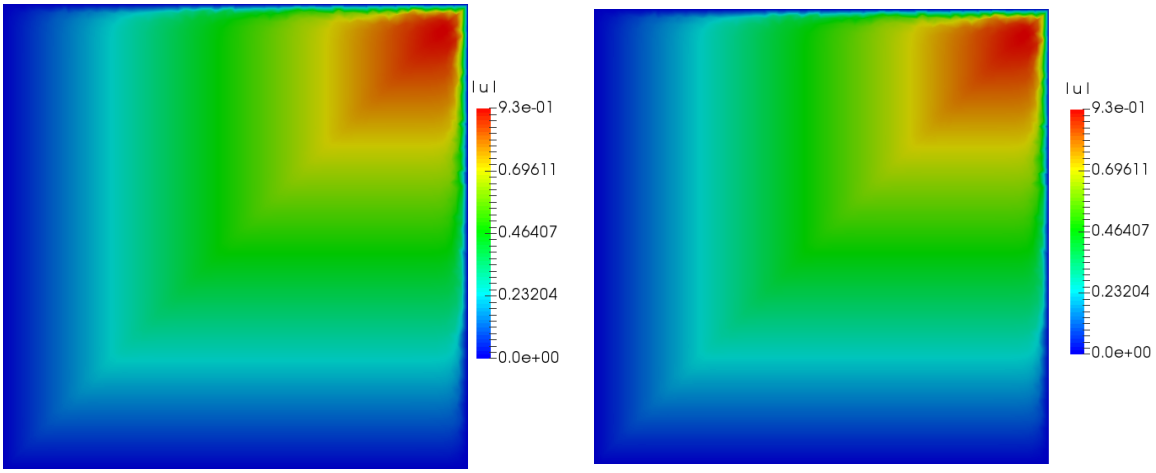
4.1.3 Numerical results and discussion

Combining the SUPG method with vanishing viscosity and rectification method, one can discuss the following options to do the numerical tests:

- *offline-online stabilization* with/without vanishing viscosity with/without rectification;
- *offline-only stabilization* with/without vanishing viscosity with/without rectification.

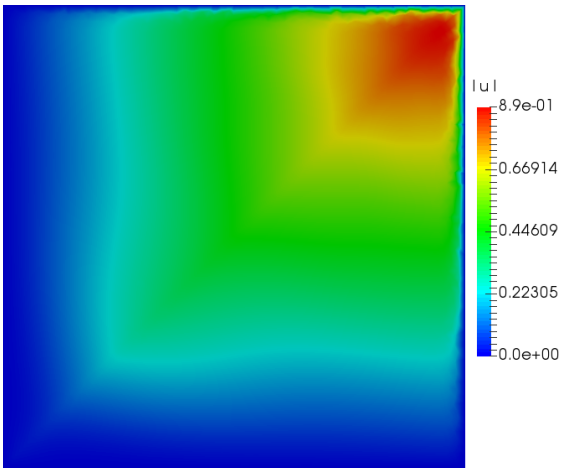
First option above is consistent for any case (with/without vanishing viscosity and rectification). We focus here on second option, because from previous chapters we know that *offline-only stabilization* is not consistent and we are interested here to correct the consistency by using vanishing viscosity and rectification method.

Figure 4.1 plots the RB solutions obtained by *offline-online stabilization* (top left); online vanishing viscosity with post-processing based on rectification (top right); online rectification only without online vanishing viscosity (below left) and online vanishing viscosity without rectification (bottom right).

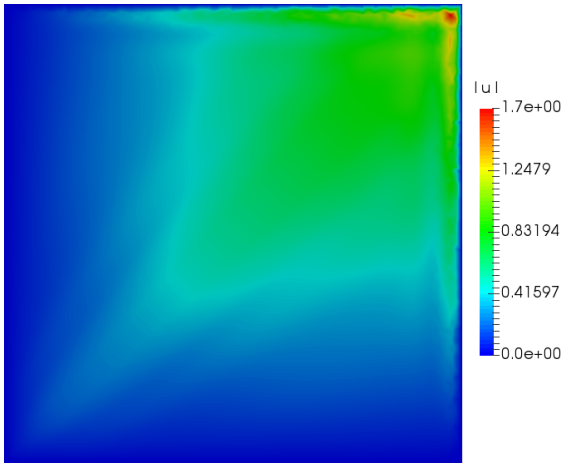


(a) Online stabilization

(b) Vanishing viscosity with rectification



(c) Rectification only



(d) Vanishing viscosity only

Figure 4.1: RB solutions at $\boldsymbol{\mu} = 10^{-6}$ obtained by different options

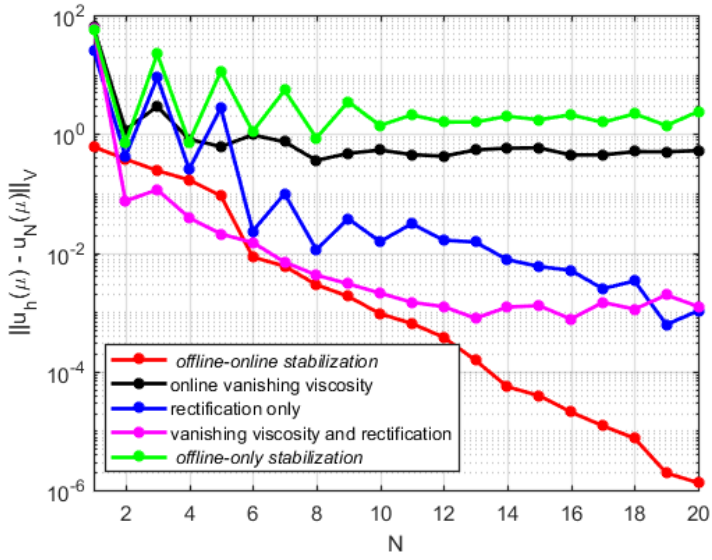


Figure 4.2: Error comparison between different stabilization options for $N = 20$ and $\boldsymbol{\mu} = 10^{-6}$.

Figure 4.2 plots the error between FE and RB solutions obtained for various stabilization options. In all cases the offline stage is stabilized with SUPG-stabilization method but online stage is obtained for different options.

We point out that the online rectification (without vanishing viscosity) option was not reported by Maday et al. [94]. From these results we see that if we perform a post-processing (online rectification) on *offline-only stabilization* (without vanishing viscosity), we are able to improve the error upto 3 order of magnitude when compared to *offline-only stabilization*.

4.2 Rectification method for Stokes problem

In this section we propose the online rectification method for Stokes problem introduced in section 4.1.2. In order to proceed with the rectification process, first we recall the steady Stokes problem in parametrized domain (defined in Chapter 2). The stabilized FE formulation (2.11) is:

$$\begin{cases} \text{Find } \boldsymbol{u}_h(\boldsymbol{\mu}) \in \boldsymbol{V}_h,\ p_h(\boldsymbol{\mu}) \in Q_h : \\ a(\boldsymbol{u}_h(\boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) + b(\boldsymbol{v}_h, p_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = F(\boldsymbol{v}_h; \boldsymbol{\mu}) + s_h^v(\boldsymbol{v}_h; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ b(\boldsymbol{u}_h(\boldsymbol{\mu}), q_h; \boldsymbol{\mu}) = G(q_h; \boldsymbol{\mu}) + s_h^q(q_h; \boldsymbol{\mu}) & \forall\, q_h \in Q_h, \end{cases} \tag{4.26}$$

where $s_h^v(\boldsymbol{v}_h; \boldsymbol{\mu})$ and $s_h^q(q_h; \boldsymbol{\mu})$ are the stabilization terms (residual based) with possible choices given by (2.12) and (2.13). We stabilize the offline stage by one of the above possible stabilization option to get stable basis functions and then, we project on RB

without taking into consideration the stabilization terms for online solve. Therefore, the RB formulation is given by

$$\begin{cases} \text{Find } (\boldsymbol{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \boldsymbol{V}_N \times Q_N : \\ a(\boldsymbol{u}_N(\boldsymbol{\mu}), \boldsymbol{v}_N; \boldsymbol{\mu}) + b(\boldsymbol{v}_N, p_N(\boldsymbol{\mu})) = F(\boldsymbol{v}_N; \boldsymbol{\mu}) & \forall\, \boldsymbol{v}_N \in \boldsymbol{V}_N, \\ b(\boldsymbol{u}_N(\boldsymbol{\mu}), q_N; \boldsymbol{\mu}) = G(q_N; \boldsymbol{\mu}) & \forall\, q_N \in Q_N. \end{cases} \tag{4.27}$$

where the RB spaces $\boldsymbol{V}_N$ and Q_N for velocity and pressure, respectively are defined in subsection 2.1.3 of Chapter 2. For comparison purpose, we also consider the supremizer enrichment, and in this case the velocity space will be $\tilde{\boldsymbol{V}}_N$ defined in subsection 2.1.3 of Chapter 2.

We recall that solving the stabilized FE formulation (4.26) in the offline stage and non-stabilized formulation (4.27) in the online stage is called *offline-only stabilization*. This option has been discussed in previous two chapters but we saw that in all cases this choice is not consistent and we were not able to get a stable RB solution (see, for instance Chapter 2, section 2.1).

In this section we try to recover the consistency of RB solution obtained by *offline-only stabilization* using the the idea of post-processing based on rectification method [37, 62, 94]. We know that in case of *offline-only stabilization*, the solutions from which RB is constructed are actually not the solutions of the problem (4.27) for $\boldsymbol{\mu} = \boldsymbol{\mu}^i$, i.e,

$$\boldsymbol{u}_N(\boldsymbol{\mu}^i) \neq \boldsymbol{u}_h(\boldsymbol{\mu}^i), \qquad p_N(\boldsymbol{\mu}^i) \neq p_h(\boldsymbol{\mu}^i), \quad \forall \boldsymbol{\mu}^i \in S_N = \{\boldsymbol{\mu}^1, ..., \boldsymbol{\mu}^N\}.$$

In other words, we are interested in correcting the consistency error of the RB approximation for velocity and pressure, respectively:

$$\boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{k=1}^{2N} \alpha_k^u(\boldsymbol{\mu})\xi_k^u \quad \text{and} \quad p_N(\boldsymbol{\mu}) = \sum_{k=1}^{N} \alpha_k^p(\boldsymbol{\mu})\xi_k^p, \tag{4.28}$$

where $\{\xi_k^u\}_{k=1}^N$ and $\{\xi_k^p\}_{k=1}^N$ are mutually orthonormal basis functions for RB velocity and pressure, respectively while $\alpha_k^u(\boldsymbol{\mu})$ and $\alpha_k^p(\boldsymbol{\mu})$ denotes the coefficients of the reduced basis approximation for velocity and pressure, respectively. The method of rectification basically replaces these reduced basis coefficients with alternate ones.

In order to calculate the alternate coefficients, first we express the N snapshots for velocity and pressure, respectively over the RB as:

$$\boldsymbol{u}_h(\boldsymbol{\mu}^i) = \sum_{k=1}^{N} \alpha_k^u(\boldsymbol{\mu}^i)\xi_k^u \qquad \text{and} \quad p_h(\boldsymbol{\mu}^i) = \sum_{k=1}^{N} \alpha_k^p(\boldsymbol{\mu}^i)\xi_k^p \tag{4.29}$$

from which we obtain the matrices R^u (for velocity) and R^p (for pressure) with columns equal to the coordinates of $\boldsymbol{u}_h(\boldsymbol{\mu}^i)$ and $p_h(\boldsymbol{\mu}^i)$ in the reduced basis ξ_k^u and ξ_k^p , respectively,

i.e. the coefficient matrices

$$R^u = \begin{pmatrix} \alpha_1^u(\boldsymbol{\mu}^1) & . & . & . & \alpha_1^u(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \alpha_N^u(\boldsymbol{\mu}^1) & . & . & . & \alpha_N^u(\boldsymbol{\mu}^N) \end{pmatrix}, \qquad R^p = \begin{pmatrix} \alpha_1^p(\boldsymbol{\mu}^1) & . & . & . & \alpha_1^p(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \alpha_N^p(\boldsymbol{\mu}^1) & . & . & . & \alpha_N^p(\boldsymbol{\mu}^N) \end{pmatrix}. \tag{4.30}$$

We compute the *offline-only* approximation of (4.27) for $\boldsymbol{\mu} = \boldsymbol{\mu}^i; i = 1, ..., N.$, i.e,

$$\boldsymbol{u}_N(\boldsymbol{\mu}^i) = \sum_{k=1}^{N} \beta_k^u(\boldsymbol{\mu}^i)\xi_k^u \quad \text{and} \quad p_N(\boldsymbol{\mu}^i) = \sum_{k=1}^{N} \beta_k^p(\boldsymbol{\mu}^i)\xi_k^p, \tag{4.31}$$

which gives us the coefficient matrices R_N^u (for velocity) and R_N^p (for pressure) with entries β_k^u and β_k^p, respectively, i.e.,

$$R_N^u = \begin{pmatrix} \beta_1^u(\boldsymbol{\mu}^1) & . & . & . & \beta_1^u(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \beta_N^u(\boldsymbol{\mu}^1) & . & . & . & \beta_N^u(\boldsymbol{\mu}^N) \end{pmatrix}, \qquad R_N^p = \begin{pmatrix} \beta_1^p(\boldsymbol{\mu}^1) & . & . & . & \beta_1^p(\boldsymbol{\mu}^N) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \beta_N^p(\boldsymbol{\mu}^1) & . & . & . & \beta_N^p(\boldsymbol{\mu}^N) \end{pmatrix}. \tag{4.32}$$

Finally, we set $J^u = R^u(R_N^u)^{-1}$ and $J^p = R^p(R_N^p)^{-1}$. The computation of J^u and J^p is done once in the offline stage and matrices are stored.

In the online stage, we compute the rectified solutions $\boldsymbol{u}_N^r(\boldsymbol{\mu})$ and $p_N^r(\boldsymbol{\mu})$ to problem (4.27) for any $\boldsymbol{\mu} \in \mathbb{P}$ as

$$\boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{k=1}^{N} \bar{\beta}_k^u(\boldsymbol{\mu})\xi_k^u \quad \text{and} \quad p_N(\boldsymbol{\mu}) = \sum_{k=1}^{N} \bar{\beta}_k^p(\boldsymbol{\mu})\xi_k^p, \tag{4.33}$$

where $\bar{\boldsymbol{\beta}}^u = J^u\boldsymbol{\beta}^{\boldsymbol{u}}$ and $\bar{\boldsymbol{\beta}}^p = J^p\boldsymbol{\beta}^{\boldsymbol{p}}$ are the coordinates for velocity and pressure, respectively. Now, combining three approaches; the supremizer enrichment [132], the *offline-online stabilization* [2] and the rectification approach, one can have the following possible options in the online stage:

- *offline-online stabilization* with/without supremizer with/without rectification
- *offline-only stabilization* with/without supremizer with/without rectification

In this chapter we are only interested in the following options:

- *offline-only stabilization* with supremizer with rectification
- *offline-only stabilization* without supremizer with rectification

4.2.1 Numerical results and discussion

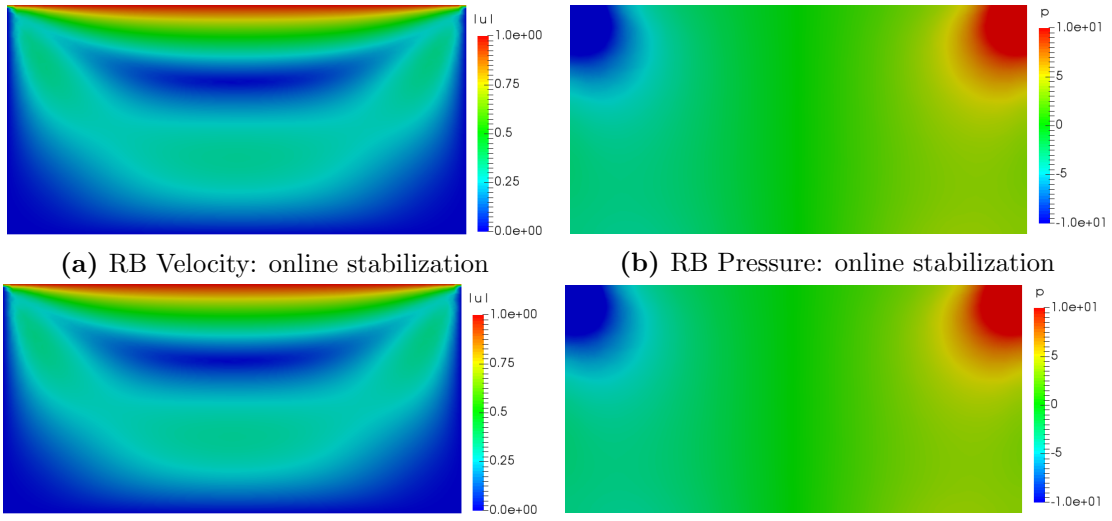
In this section we present some numerical solutions for the new stabilization strategy presented in section 4.2. We consider the following two test cases with increasing complexity as we move from test case one to test case two.

Cavity test case

As a first example we consider the same parametrized cavity domain (Fig. 2.1) as in previous chapters. Figure 4.3 shows the RB solutions for velocity (left) and pressure (right), respectively. These solutions are obtained for online stabilization (top) and online rectification (below). Recall that in both cases the offline stage is stabilized. From these plots, we see that the solutions obtained by two different stabilization approaches are same.

Figure 4.4 illustrates the absolute error between FE and RB solutions for velocity, using different stabilization options. We see that the error for rectification method is 10^{-6} which is almost zero. However *offline-online stabilization* method is still better (as was in previous chapters).

Similarly in Fig. 4.5 we plot the absolute error between FE and RB solution for pressure. In this case the rectification method is able to reduce the error down to 10^{-5}, apart from some peaks at different values of N. These peaks are due to the poor conditioning of the matrix R_N^p, which, in this case is controlled by the enrichment of RB velocity space with supremizer solutions and the error is decreased to 10^{-7}.



(a) RB Velocity: online stabilization

(b) RB Pressure: online stabilization

(c) RB Velocity: online rectification

(d) RB Pressure: online rectification

Figure 4.3: Comparison between the RB solutions for velocity (left) and pressure (right), respectively obtained by online stabilization (top) and online rectification (bottom).

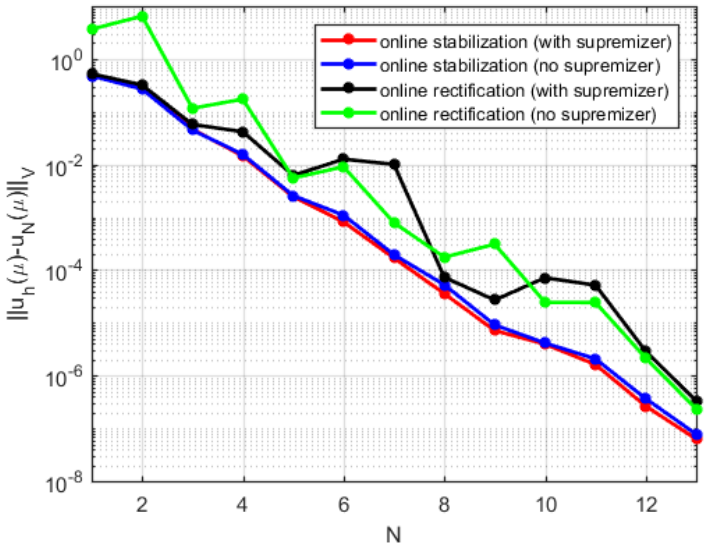


Figure 4.4: Cavity problem: Velocity error between FE solution and RB solution for different possible options with $N_u = N_p = 13$.

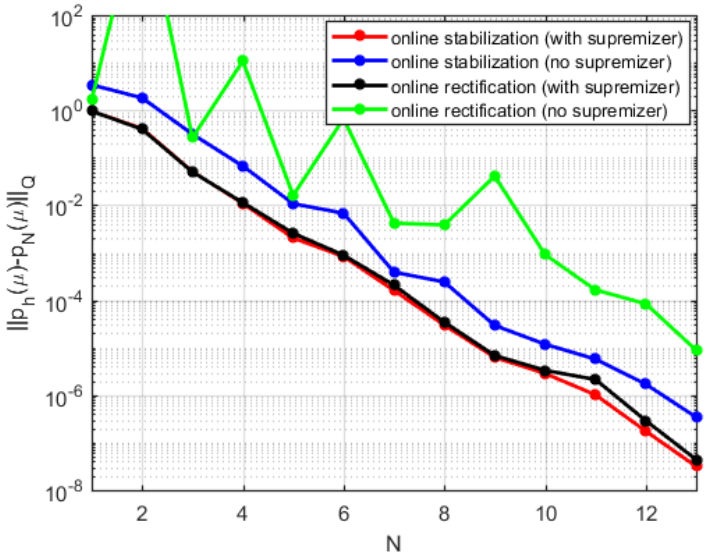


Figure 4.5: Cavity problem: Pressure error between FE solution and RB solution for different possible options with $N_u = N_p = 13$.

T-bypass test

In order to see the validity of rectification method in more challenging problems, for instance, in this example we consider the problem with many parameters. We take the example of "T-bypass" configuration from Rozza and Veroy [132]. Parametrized domain is shown in Fig. 4.6 with vector of parameters $\boldsymbol{\mu} = [t, D, L, S, H, \theta]$ labeled. The parameter ranges in the offline stage are $t = D = L = S = H \in [0.5, 1.5]$ and $\theta \in [0, \pi/6]$. The online parameter values are $t = D = L = S = H = 1.0$ and $\theta = \pi/6$.

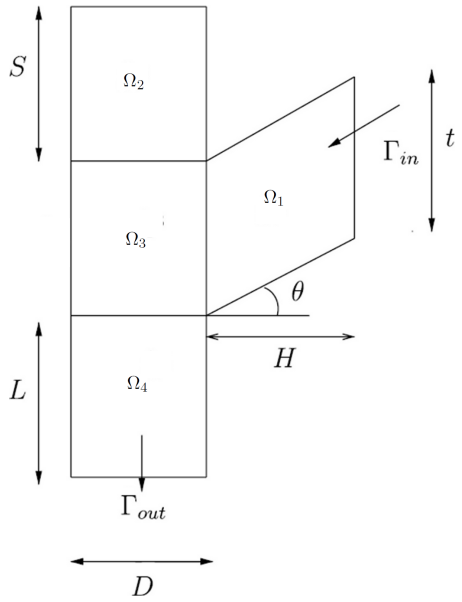


Figure 4.6: Parametrized domain for T-bypass example.

In Figures 4.7 and 4.8 we show the absolute error between FE and RB solutions for velocity and pressure, respectively for different stabilization options. From these results we see that in the online rectification there are some peaks at different values of N. Some of these peaks are controlled by the enrichment of supremizer in RB velocity spaces. But in case of pressure these peaks are not completely controlled by supremizer enrichment. In such cases one can use the POD orthonormalization [36] which can help to reduce the condition number of rectification matrix R_N^p.

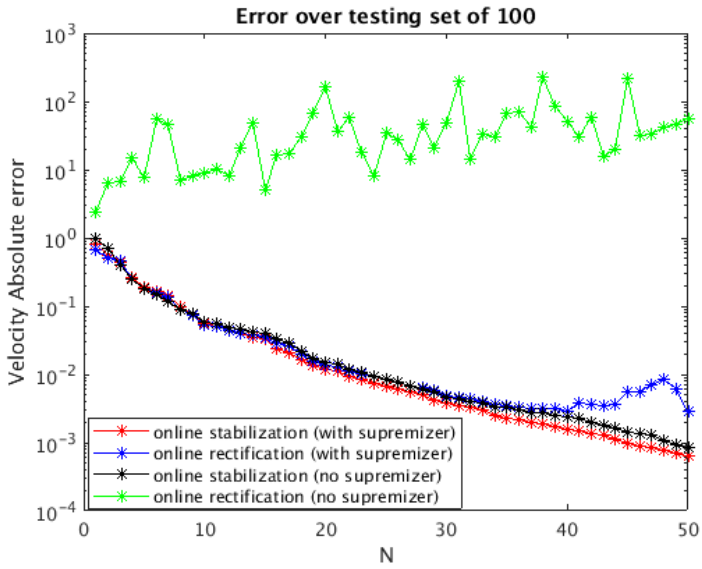


Figure 4.7: T-bypass example: Velocity error between FE solution and RB solution for different possible options with $N_u = N_p = 50$.

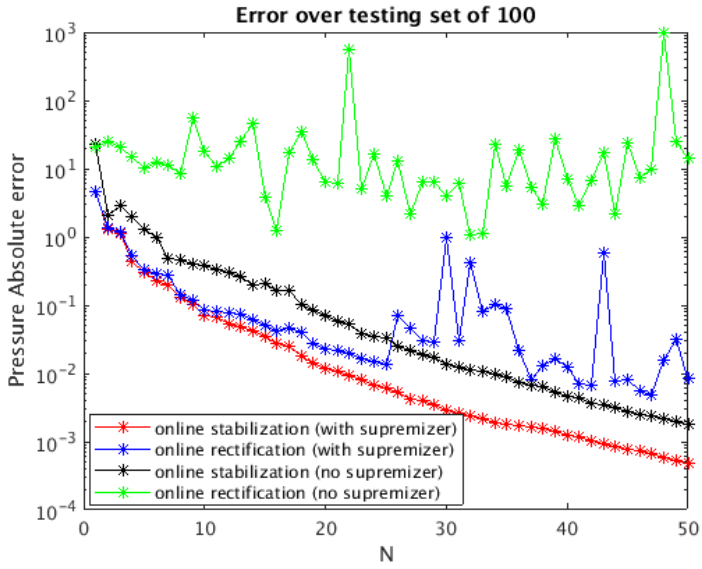


Figure 4.8: T-bypass example: Pressure error between FE solution and RB solution for different possible options with $N_u = N_p = 50$.

4.3 Concluding remarks

In this chapter we have introduced the rectification method for parametrized Stokes problem. We have reviewed the paper by Maday et al. [94] for the advection-diffusion problem from which we extended the idea of post processing to get the rectified solution of reduced parametric viscous problem. More specifically the rectification method is used to improve the *offline-only stabilization* option. The main outcomes of this chapter based on numerical experiments are as follows:

- we point out that in case of advection-dominated problem, even if we do not consider the vanishing viscosity, we are able to get a stable RB solution with the post processing (rectification) only, see for instance Fig 4.2 (blue line). This case was not reported previously in [94];
- In case of Stokes problem we are able to get a stable RB solution for velocity and pressure while doing the rectification on *offline-only stabilized* RB solution;
- We have also compared rectification method with *offline-online stabilization* approach and conclude that *offline-online stabilization* is best way to stabilize;
- supremizers play the same role in cavity flow case as in previous chapters, it improves the pressure approximation and do not effect the velocity. However in more complex problem (T-shape), the role of supremizer for both velocity and pressure is more important.

Chapter 5

Conclusion and Perspectives

5.1 Summary

The focus of this thesis has been the development of a RB method which gives a inf-sup stable solution for parametrized Stokes and Navier-Stokes problem. We have developed the stabilized RB method using the classical stabilization techniques [27, 30, 68, 69] in the *high-fidelity* and, then, projecting on RB. This approach [2] has been combined with the supremizer enrichment approach [132] to get few options; *(i) offline-online stabilization with supremizer, (ii) offline-online stabilization without supremizer, (iii) offline-only stabilization with supremizer and (iv) offline-only stabilization without supremizer*. The first three options have been studied in this thesis and results are compared through numerical tests and using equal order velocity-pressure interpolation ($\mathbb{P}_k/\mathbb{P}_k; k = 1, 2$). The last option is not sufficient to guarantee stability. We have considered both the steady and unsteady problems in parametric settings. The numerical simulations have been carried out in FreeFem++ [59] and then in RBniCS [10].

In Chapter 2, the classical residual based stabilization method given by Franca and Hughes [68] has been used to develop the stabilized RB method for parametrized steady and unsteady Stokes problem. We have compared the *offline-online stabilization* approach [2, 3] with the supremizer enrichment approach [132], through numerical results. In order to support the *offline-online stabilization* we have also experimented the stabilization techniques by Brezzi and Pitkãranta [27] and also lowest order stabilized element $\mathbb{P}_1/\mathbb{P}_0$. The effect of varying stabilization coefficient δ on RB velocity and pressure has been presented. We have also incorporated the sensitivity analysis on time step Δt in order to verify the stability condition on Δt in reduced order. The computational costs of stabilized method using ($\mathbb{P}_k/\mathbb{P}_k; k = 1, 2$) has been compared with the stable Taylor-Hood FE ($\mathbb{P}_2/\mathbb{P}_1$). All the numerical tests have been performed on a parametrized cavity flow, where the parameters are physical (viscosity) and geometrical, acting on the length of the flow cavity.

In Chapter 3, the stabilized RB method for the approximation of parametrized steady and unsteady Navier-Stokes problem have been developed. The RB formulation is build, using the classical residual based stabilization technique SUPG [30] in the full order

method during the *offline* stage and, then, projecting on the RB space. We have only addressed the RB inf-sup stability and not the instability caused by dominating advection field, since we are dealing with low to moderate Reynolds number. We have compared our approach [2] with the existing approaches [132] through a series of numerical experiments. For instance, the comparison between *offline-online stabilization* with/without supremizer and *offline-only stabilization* for steady and unsteady Navier-Stokes problems is presented. Similar to Chapter 2, the proposed method has been tested on a parametrized cavity problem.

In Chapter 4, a new online stabilization strategy has been presented for the approximation of parametrized Stokes problem. We have performed a post processing based on rectification method to correct the consistency of the *offline-only stabilization* approach. This approach has been compared with residual based stabilization approach presented in Chapter 2. We summarize the main findings of this thesis as follows:

- from all the numerical results carried out in this thesis, we conclude that the *offline-online stabilization* is the most appropriate way to perform reduced basis stabilization of parametrized Stokes and Navier-Stokes problem;
- in case of velocity, *offline-online stabilization* method allows us to avoid the addition of supremizer enrichment to fulfill the reduced order parametric inf-sup condition by reducing the dimension of reduced velocity space and therefore it can reduce the computational cost of *online* stage;
- in case of pressure, *offline-online stabilization* with supremizer has better performance in terms of error as compare to *offline-online stabilization* without supremizer;
- the lack of consistency is causing larger errors if we use stabilized bilinear forms during the *offline* stage and non-stabilized bilinear forms during the online stage, i.e if we use *offline-only stabilized* method;
- construction of stable reduced basis functions in the *offline* stage does not guarantee the stable RB solution in the *online* stage, therefore, we also need stabilization in *online* stage;
- a post processing based on rectification is helpful to improve the consistency error in the *offline-only stabilization* even if this approach is still less accurate than the consistent *offline-online stabilization* but it can be useful and less expensive;
- in terms of CPU time, the Taylor-Hood FE pair $(\mathbb{P}_2/\mathbb{P}_1)$ is more expensive than $(\mathbb{P}_1/\mathbb{P}_1)$ stabilized but less expensive than $(\mathbb{P}_2/\mathbb{P}_2)$ stabilized.

5.2 Perspectives for future work

After summarizing the work carried out in this thesis, we still have some open questions and suggestions to improve this work in future, in order to make this approach applicable

to more and more complex problems. We conclude this work by giving some plans, connections and suggestions for future work:

- in order to develop a certified stabilized RB method, an a *posteriori* error analysis [63, 144] is needed for residual based stabilization in a reduced order setting, for which we suggest to have a look into the error analysis of stabilized FE methods [82, 136];
- the computational cost of stabilized RB method in case of nonlinear problems can be decreased by using the Empirical Interpolation method (EIM) [13];
- one can extend this work to develop a Variational MultiScale (VMS) method for turbulent flows with moderate-higher Reynolds number [134];
- this work is applicable to optimal control problems, see for instance environmental applications in marine sciences [137];
- this work can be extended to the fluid structure interaction (FSI) problems in convection dominated regime where the solutions (in particular the pressure of fluid) are traveling waves [9]:
- further application of this work could be the development of efficient stabilized RB methods to simulate conjugate heat transfer for multi-components systems, which appear in several engineering applications [57].

Bibliography

[1] A. Abdulle and O. Budáč. A Petrov–Galerkin reduced basis approximation of the Stokes equation in parameterized geometries. *Comptes Rendus Mathematique*, 353(7):641 – 645, 2015.

[2] S. Ali, F. Ballarin, and G. Rozza. Stabilized reduced basis methods for parametrized steady Stokes and Navier-Stokes equations. *Submitted*, 2018.

[3] S. Ali, F. Ballarin, and G. Rozza. Unsteady stabilized reduced basis methods for parametrized Stokes and Navier-Stokes equations. *Submitted*, 2018.

[4] J. A. Atwell and B. B. King. Reduced order controllers for spatially distributed systems via proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 26(1):128–151, 2004.

[5] J. Baiges, R. Codina, and S. Idelsohn. Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 72:1219–1243, 2013.

[6] J. Baiges, R. Codina, and S. R. Idelsohn. *Reduced-Order Modelling Strategies for the Finite Element Approximation of the Incompressible Navier-Stokes Equations*, pages 189–216. Springer International Publishing, 2014.

[7] M. Balajewicz and E. H. Dowell. Stabilization of projection-based reduced order models of the Navier-Stokes. *Nonlinear Dynamics*, 70(2):1619–1632, 2012.

[8] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza. Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5):1136–1161, 2015.

[9] F. Ballarin, G. Rozza, and Y. Maday. *Reduced-order semi-implicit schemes for fluid-structure interaction problems*, pages 149–167. Springer International Publishing, 2017.

[10] F. Ballarin, A. Sartori, and G. Rozza. RBniCS - reduced order modelling in FEniCS. *http://mathlab.sissa.it/rbnics*, 2016.

[11] E. Balms. Parametric families of reduced finite element models. theory and applications. *Mechanical Systems and Signal Processing*, 10(4):381 – 394, 1996.

[12] H. T. Banks, R. del Rosario, and R. C. Smith. Reduced order model feedback control design: Computational studies for thin cylindrical shells. Technical report, IEEE Trans. Auto. Contr, 1998.

[13] 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. *Comptes Rendus Mathematique*, 339(9):667 – 672, 2004.

[14] M. A. Behr, L. P. Franca, and T. E. Tezduyar. Stabilized finite element methods for the velocity-pressure-stress formulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 104(1):31 – 48, 1993.

[15] P. Benner, M. Ohlberger, A. T. Patera, G. Rozza, D. C. Sorensen, and K. Urban. Model order reduction of parameterized systems (MoRePaS). *Advances in Computational Mathematics*, 41(5):955–960, 2015.

[16] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):516 – 538, 2009.

[17] G. Berkooz, P. Holmes, and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. 25:539–575, 2003.

[18] K. Bicol and A. Quaini. On the sensitivity to model parameters in a filter stabilization technique for advection dominated advection-diffusion-reaction problems. *https://arxiv.org/abs/1805.01376v1*, 2018.

[19] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM Journal on Mathematical Analysis*, 43(3):1457–1472, 2011.

[20] P. B. Bochev, M. D. Gunzburger, , and R. Lehoucq. On stabilized finite element methods for transient problems with varying time scales. *Proceedings of ECOMAS 2004*.

[21] P. B. Bochev, M. D. Gunzburger, and R. B. Lehoucq. On stabilized finite element methods for the Stokes problem in the small time step limit. *International Journal for Numerical Methods in Fluids*, 53(4):573–597, 2007.

[22] P. B. Bochev, M. D. Gunzburger, and J. N. Shadid. On inf-sup stabilized finite element methods for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 193(15):1471 – 1489, 2004.

[23] D. Boffi, F. Brezzi, L. F. Demkowicz, R. G. Durn, R. S. Falk, and M. Fortin. *Mixed Finite Elements, Compatibility Conditions, and Applications*. Springer International Publishing, 2006.

[24] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen problem. *Computer Methods in Applied Mechanics and Engineering*, 196(46):853 – 866, 2007.

[25] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 8:129–151, 1974.

[26] F. Brezzi and J. Douglas. Stabilized mixed methods for the Stokes problem. *Numerische Mathematik*, 53(1-2):225–235, 1988.

[27] F. Brezzi and J. Pitkranta. On the stabilization of finite element approximations of the Stokes equations. In *Efficient Solutions of Elliptic Systems*, volume 10 of *Notes on Numerical Fluid Mechanics*, pages 11–19. 1984.

[28] F. Brezzi, J. Rappaz, and P. A. Raviart. Finite dimensional approximation of nonlinear problems. *Numerische Mathematik*, 36(1):1–25, 1980.

[29] A. Brooks and T. Hughes. Streamline Upwind/Petrov-Galerkin methods for advection dominated flows. *Third International Conference on Finite Element Methods in Fluid Flow*, 2 , Calgary, Canada, Calgary Univ., 1980.

[30] A. N. Brooks and T. J. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(13):199 – 259, 1982.

[31] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: M2AN*, 46(3):595–603, 2012.

[32] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *Journal of Computational Physics*, 259:598 – 616, 2014.

[33] W. Cazemier, R. W. C. P. Verstappen, and A. E. P. Veldman. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. *Physics of Fluids*, 10(7):1685–1699, 1998.

[34] T. Chacón Rebollo. A term by term stabilization algorithm for finite element solution of incompressible flow problems. *Numerische Mathematik*, 79(2):283–319, 1998.

[35] T. Chacón Rebollo, E. Delgado Ávila, M. Mármol Gómez, F. Ballarin, and G. Rozza. On a certified Smagorinsky reduced basis turbulence model. *SIAM Journal on Numerical Analysis*, 55(6):3047–3067, 2017.

[36] R. Chakir and J. Hammond. A non-intrusive reduced basis method for elastoplasticity problems in geotechnics. *Journal of Computational and Applied Mathematics*, 337:1 – 17, 2018.

[37] R. Chakir and Y. Maday. A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE. In *9e Colloque national en calcul des structures*, Giens, France, 2009. CSMA.

[38] R. Chakir, Y. Maday, and P. Parnaudeau. A non-intrusive reduced basis approach for parametrized heat transfer problems. *https://hal.archives-ouvertes.fr/hal-01691647*, 2018.

[39] I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *International Journal for Numerical Methods in Engineering*, 10(6):1389–1396, 1976.

[40] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190(2021):2681 – 2706, 2001.

[41] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. *SIAM Journal on Numerical Analysis*, 46(4):2039–2067, 2008.

[42] S. Deparis and A. E. Løvgren. Stabilized reduced basis approximation of incompressible three-dimensional Navier-Stokes equations in parametrized deformed domains. *Journal of Scientific Computing*, 50(1):198–212, 2012.

[43] 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.

[44] J. J. Douglas and J. Wang. An absolutely stabilized finite element formulation for the Stokes problem. *Mathematics of Computations*, 52(186):495 – 508, 1989.

[45] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[46] J. L. Eftang, D. J. Knezevic, and A. T. Patera. An hp certified reduced basis method for parametrized parabolic partial differential equations. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):395–422, 2011.

[47] L. Formaggia, F. Saleri, and A. Veneziani. *Solving Numerical PDEs: Problems, Applications, Exercises*. Springer-Verlag Mailand, 2012.

[48] L. P. Franca and S. L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99(2):209 – 233, 1992.

[49] L. P. Franca, S. L. Frey, and T. J. Hughes. Stabilized finite element methods: I. application to the advective-diffusive model. *Computer Methods in Applied Mechanics and Engineering*, 95(2):253 – 276, 1992.

[50] F. Gelsomino and G. Rozza. Comparison and combination of reduced-order modelling techniques in 3D parametrized heat transfer problems. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):371–394, 2011.

[51] A.-L. Gerner and K. Veroy. Certified reduced basis methods for parametrized saddle point problems. *SIAM Journal on Scientific Computing*, 34(5):A2812–A2836, 2012.

[52] S. Giere, T. Iliescu, V. John, and D. Wells. SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 289:454 – 474, 2015.

[53] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations*, volume 5. Springer, 1986.

[54] M. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows*, volume 5. Academicr, 1989.

[55] B. Haasdonk. Convergence rates of the POD-greedy method. *ESAIM: M2AN*, 47(3):859–873, 2013.

[56] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.

[57] E. Hachem. *Stabilized finite element method for heat transfer and turbulent flows inside industrial furnaces*. Phd theses, École Nationale Supérieure des Mines de Paris, 2009.

[58] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 84(2):175 – 192, 1990.

[59] F. Hecht. New development in FreeFem++. *Journal of Numerical Mathematics*, 20(3-4):251–266, 2013.

[60] J. C. Heinrich, P. S. Huyakorn, O. C. Zienkiewicz, and A. R. Mitchell. An upwind finite element scheme for two-dimensional convective transport equation. *International Journal for Numerical Methods in Engineering*, 11(1):131–143, 1977.

[61] J. C. Heinrich and O. C. Zienkiewicz. Quadratic finite element schemes for two-dimensional convective-transport problems. *International Journal for Numerical Methods in Engineering*, 11(12):1831–1844, 1977.

[62] H. Herrero, Y. Maday, and F. Pla. RB (Reduced basis) for RB (Rayleigh-Bénard). *Computer Methods in Applied Mechanics and Engineering*, 261-262:132 – 141, 2013.

[63] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics. 2015.

[64] S. Hijazi, S. Ali, G. Stabile, F. Ballarin, and G. Rozza. The Effort of Increasing Reynolds Number in Projection-Based Reduced Order Methods: from Laminar to Turbulent Flows. *Submitted*, 2018.

[65] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 1996.

[66] T. Hughes and A. Brooks. A multi-dimensioal upwind scheme with no crosswind diffusion. *Finite Element Methods for Convection Dominated Flows, New York, U.S.A.,*, 34:19–35, 1979.

[67] T. J. Hughes and L. P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Computer Methods in Applied Mechanics and Engineering*, 65(1):85 – 96, 1987.

[68] T. J. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuka-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1):85 – 99, 1986.

[69] T. J. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173 – 189, 1989.

[70] T. J. Hughes, W. K. Liu, and A. Brooks. Finite element analysis of incompressible viscous flows by the penalty function formulation. *Journal of Computational Physics*, 30(1):1 – 60, 1979.

[71] T. J. R. Hughes. A simple scheme for developing upwind finite elements. *International Journal for Numerical Methods in Engineering*, 12(9):1359–1365, 1978.

[72] T. J. R. Hughes and A. N. Brooks. *A theoretical framework for Petrov-Galerkin methods, with discontinuous weighting functions: application to the streamline upwind procedure*, volume IV, pages 47–65. John Wiley and Sons Ltd, 1982.

[73] D. B. P. Huynh, D. Knezevic, Y. Chen, J. Hesthaven, and A. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Computer Methods in Applied Mechanics and Engineering*, 199(29):1963 – 1975, 2010.

[74] D. B. P. Huynh and A. T. Patera. Reduced basis approximation and a posteriori error estimation for stress intensity factors. *International Journal for Numerical Methods in Engineering*, 72(10):1219–1259, 2007.

[75] D. B. P. Huynh, F. Pichi, and G. Rozza. Reduced basis approximation and a posteriori error estimation: applications to elasticity problems in several parametric settings. *SEMA-SIMAI Series, In Press*, 2018.

[76] D. B. P. Huynh, G. Rozza, S. Sen, and A. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathematique*, 345(8):473 – 478, 2007.

[77] 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.

[78] L. Iapichino, A. Quarteroni, G. Rozza, and S. Volkwein. Reduced basis method for the Stokes equations in decomposable parametrized domains using greedy optimization. In *Progress in Industrial Mathematics at ECMI 2014*, pages 647–654. Springer International Publishing, 2016.

[79] K. Ito and S. Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403 – 425, 1998.

[80] E. Jenkins, V. John, A. Linke, and L. Rebholz. On the parameter choice in grad-div stabilization for the Stokes equations. *Advances in Computational Mathematics*, pages 1–26, 2013.

[81] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Mathematics of Computation*, 47(175):1–18, 1986.

[82] D. Kay and D. Silvester. A Posteriori Error Estimation for Stabilized Mixed Approximations of the Stokes Equations. *SIAM Journal on Scientific Computing*, 21(4):1321–1336, 1999.

[83] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102(2):345–371, 1999.

[84] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002.

[85] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, volume 9, pages 235–274. Springer MS&A Series, 2014.

[86] T. Lassila, A. Manzoni, and G. Rozza. Reduction strategies for shape dependent inverse problems in haemodynamics. *IFIP Advances in Information and Communication Technology*, 391 AICT:397–406, 2013.

[87] A. Logg, K. A. Mardal, and G. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer-Verlag, Berlin, 2012.

[88] A. E. Løvgren. *A reduced basis method for the steady Navier-Stokes problem, in "Reduced basis modelling of hierarchical flow system"*. PhD thesis, Norwegian University of Science and Technology, 2006.

[89] A. E. Lovgren, Y. Maday, and E. M. Ronquist. A reduced basis element method for the steady Stokes problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 40:529–552, 2006.

[90] G. Lube and M. A. Olshanskii. Stable finite element calculation of incompressible flows using the rotation form of convection. *IMA Journal of Numerical Analysis*, 22(3):437–461, 2002.

[91] G. Lube and G. Rapin. Residual-based stabilized higher-order FEM for a generalized oseen problem. *Mathematical Models and Methods in Applied Sciences*, 16(07):949–966, 2006.

[92] G. Lube and G. Rapin. Residual-based stabilized higher-order FEM for advection-dominated problems. *Computer Methods in Applied Mechanics and Engineering*, 195(3336):4124 – 4138, 2006.

[93] G. Lube and L. Tobiska. A nonconforming finite element method of streamline diffusion type for the incompressible Navier-Stokes equations. *Journal of Computational Mathematics*, 8(2):147–158, 1990.

[94] Y. Maday, A. Manzoni, and A. Quarteroni. An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems. *Comptes Rendus Mathematique*, 354(12):1188 – 1194, 2016.

[95] Y. Maday, A. T. Patera, and G. Turinici. Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations. *Comptes Rendus Mathematique*, 335(3):289 – 294, 2002.

[96] Y. Maday, A. T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17(1):437–446, 2002.

[97] Y. Maday and E. Tadmor. Analysis of the Spectral Vanishing Viscosity Method for Periodic Conservation Laws. *SIAM Journal on Numerical Analysis*, 26(4):854–870, 1989.

[98] A. Manzoni. An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier-Stokes flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(4):1199–1226, 2014.

[99] A. Manzoni, T. Lassila, A. Quarteroni, and G. Rozza. *A Reduced-Order Strategy for Solving Inverse Bayesian Shape Identification Problems in Physiological Flows*, pages 145–155. Springer International Publishing, 2014.

[100] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):604–625, 2012.

[101] R. Milani, A. Quarteroni, and G. Rozza. Reduced basis method for linear elasticity problems with many parameters. *Computer Methods in Applied Mechanics and Engineering*, 197(51-52):4812–4829, 2008.

[102] F. Negri, A. Manzoni, and G. Rozza. Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations. *Computers and Mathematics with Applications*, 69(4):319–336, 2015.

[103] N. C. Nguyen, G. Rozza, D. B. P. Huynh, and A. T. Patera. *Reduced Basis approximation and a posteriori error estimation for parametrized parabolic PDEs: Application to real-time Bayesian parameter estimation*, pages 151–177. John Wiley and Sons, Ltd, 2010.

[104] 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.

[105] N. C. Nguyen, K. Veroy, and A. T. Patera. *Certified Real-Time Solution of Parametrized Partial Differential Equations*, pages 1529–1564. Springer Netherlands, 2005.

[106] M. Olshanskii, G. Lube, T. Heister, and J. Lwe. Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 198(4952):3975 – 3988, 2009.

[107] P. Pacciarini and G. Rozza. Stabilized reduced basis method for parametrized advection-diffusion PDEs. *Computer Methods in Applied Mechanics and Engineering*, 274:1–18, 2014.

[108] J. S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777–786, 1989.

[109] G. Pitton, A. Quaini, and G. Rozza. Computational reduction strategies for the detection of steady bifurcations in incompressible fluid-dynamics: Applications to coanda effect in cardiology. *Journal of Computational Physics*, 344:534 – 557, 2017.

[110] G. Pitton and G. Rozza. On the application of reduced basis methods to bifurcation problems in incompressible fluid dynamics. *Journal of Scientific Computing*, 73(1):157–177, 2017.

[111] C. Prud'homme and A. Patera. Reduced-basis output bounds for approximately parametrized elliptic coercive partial differential equations. *Computing and Visualization in Science*, 6(2):147–162, 2004.

[112] A. Quarteroni. *Numerical Models For Differential Problems*, volume 8 of MS&A. Springer-Verlag Italia, Milano, 2013.

[113] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. Springer International Publishing, 2015.

[114] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numerical Methods for Partial Differential Equations*, 23(4):923–948, 2007.

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

[116] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, New York, 2000.

[117] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.

[118] S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, 2000.

[119] 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.

[120] P. J. Roache. *Computational Fluid Dynamics*. Hermosa Publishers, 1976.

[121] D. Rovas. *Reduced-basis output bound methods for parametrized partial differential equations*. PhD thesis, Massachusetts Institute of Technology, 2003.

[122] G. Rozza. Reduced-basis methods for elliptic equations in sub-domains with a posteriori error bounds and adaptivity. *Applied Numerical Mathematics*, 55(4):403 – 424, 2005.

[123] 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, N. 3400, 2005.

[124] G. Rozza. Real-time reduced basis solutions for Navier-Stokes equations: optimization of parametrized bypass configurations. In *ECCOMAS CFD 2006 Proceedings on CD*, number 676, pages 1–16, 2006.

[125] G. Rozza. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Computing and Visualization in Science*, 12(1):23–35, 2009.

[126] G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Communications in Computational Physics*, 9(1):1–48, 2011.

[127] 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. *Numerische Mathematik*, 125(1):115–152, 2013.

[128] G. Rozza, D. B. P. Huynh, N. Nguyen, and A. Patera. Real-time reliable simulation of heat transfer phenomena. In *Proceedings of the ASME Summer Heat Transfer Conference 2009, HT2009*, volume 3, pages 851–860, 2009.

[129] G. Rozza, D. B. P. Huynh, and A. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[130] G. Rozza, T. Lassila, and A. Manzoni. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. *Lecture Notes in Computational Science and Engineering*, 76:307–315, 2011.

[131] G. Rozza, C. Nguyen, A. Patera, and S. Deparis. Reduced basis methods and a posteriori error estimators for heat transfer problems. In *Proceedings of the ASME Summer Heat Transfer Conference 2009, HT2009*, volume 2, pages 753–762, 2009.

[132] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196:1244–1260, 2007.

[133] L. Sirovich. Turbulence and the dynamics of choerent structures part i: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[134] G. Stabile, F. Ballarin, G. Zuccarino, and G. Rozza. Reduced order variational multiscale approach for turbulent flows. *In preparation*, 2018.

[135] G. Stabile and G. Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations. *Computers & Fluids in press*, 2018.

[136] R. Stenberg and J. Videman. On the Error Analysis of Stabilized Finite Element Methods for the Stokes Problem. *SIAM Journal on Numerical Analysis*, 53(6):2626–2633, 2015.

[137] M. Strazzullo, F. Ballarin, R. Mosetti, and G. Rozza. Model reduction for parametrized optimal control problems in environmental marine sciences and engineering. *SIAM Journal on Scientific Computing*, 40(4):B1055–B1079, 2018.

[138] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1(1):73 – 100, 1973.

[139] T. Tezduyar, S. Mittal, S. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95(2):221 – 242, 1992.

[140] T. E. Tezduyar, R. Glowinski, and J. Liou. Petrov-Galerkin methods on multiply connected domains for the vorticity-stream function formulation of the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 8(10):1269–1290, 1988.

[141] L. Tobiska and G. Lube. A modified streamline diffusion method for solving the stationary Navier-Stokes equation. *Numerische Mathematik*, 59(1):13–30, 1991.

[142] L. Tobiska and R. Verfurth. Analysis of a streamline diffusion finite element method for the Stokes and Navier-stokes equations. *SIAM Journal on Numerical Analysis*, 33(1):107–127, 1996.

[143] D. Torlo, F. Ballarin, and G. Rozza. Stabilized weighted reduced basis methods for parametrized advection dominated problems with random inputs. *Submitted*, 2017.

[144] 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.

[145] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237-240:10 – 26, 2012.

[146] J. Weller, E. Lombardi, M. Bergmann, and A. Iollo. Numerical methods for low-order modeling of fluid flows based on POD. *International Journal for Numerical Methods in Fluids*, 63(2):249–268.

[147] D. Wolfgang, P. Christian, and W. Gerrit. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: M2AN*, 48(3):623–663, 2014.

[148] T. X. Zhou and M. F. Feng. A least squares Petrov-Galerkin finite element method for the stationary Navier-Stokes equations. *Mathematics of Computation*, 60(202):531–543, 1993.