POLITECNICO DI TORINO

CORSO DI LAUREA MAGISTRALE
IN INGEGNERIA MATEMATICA

Master thesis

REDUCED ORDER METHODS
FOR UNCERTAINTY QUANTIFICATION
IN COMPUTATIONAL FLUID DYNAMICS

Advisors:
Prof. Claudio Canuto
Prof. Gianluigi Rozza

Co-Advisors:
Dr. Francesco Ballarin

Author:
Julien Genovese

ACCADEMIC YEAR 2018/2019

To my friends and in particular to my best friends close to me, like Laurent, Fedra, Antonella and Thierry, one of the reasons why I am living a very good life.
To my family from father, mother, grandmother, to uncles.
To two fantastic persons, Andrea and Nadia, that I met in a difficoult situation and helped me a lot for finishing this work calmly.
To my professors that brought me here.
To all the people that believed in me and helped me from the beginning.
And finally to the mathematics that gives a sense and a direction to my life.

Contents

1 General notions about Stokes equations **7**
- 1.1 The strong and the weak formulation: Brezzi's theory . . . 7
- 1.2 Babuška's theory . . . 12
- 1.3 Finite element discretization with Brezzi's formulation . . . 13
- 1.4 Conclusions . . . 15

2 Reduced ordel modelling for Stokes equations **17**
- 2.1 Introduction to the reduced problem . . . 17
- 2.2 General theory behind reduced order methods . . . 18
- 2.3 Pull back to the reference domain . . . 19
- 2.4 Inf-sup condition problem, supremizer operator and reduced basis . . . 21
- 2.5 Greedy and POD algorithms . . . 25
  - 2.5.1 Proper Orthogonal Decomposition (POD) . . . 26
  - 2.5.2 Greedy algorithm . . . 28
- 2.6 Conclusions . . . 34

3 Reduced order modelling for Steady Navier-Stokes equations **35**
- 3.1 The steady Navier-Stokes problem . . . 35
- 3.2 Finite element discretization and algebraic formulation . . . 37
- 3.3 Reduced model for Navier-Stokes equations . . . 41
- 3.4 Conclusions . . . 42

4 Weighted reduced order methods **43**
- 4.1 Problem setting . . . 43
- 4.2 Weighted algorithms . . . 48
  - 4.2.1 Weighted Proper Orthogonal Decomposition . . . 49
  - 4.2.2 Weighted greedy algorithm . . . 51
- 4.3 Conclusions . . . 52

5 Random parameter space sampling: tensor products, sparse grids, and random grids **55**
5.1 Tensor Product quadrature rule . . . 56
5.2 Smolyak quadrature rule . . . 58
5.3 Monte-Carlo Methods . . . 66
5.4 Conclusions . . . 67

6 Numerical results **69**
6.1 Stokes problem . . . 72
6.1.1 Stokes: Beta 0.03 0.03 . . . 72
6.1.2 Stokes: Beta 10 10 . . . 77
6.1.3 Stokes: Beta 20 1 . . . 80
6.1.4 Stokes: Beta 75 75 . . . 83
6.1.5 Discussion for the Stokes problem . . . 86
6.2 Navier-Stokes problem . . . 86
6.2.1 Navier-Stokes: Beta 0.03 0.03 . . . 86
6.2.2 Navier-Stokes: Beta 10 10 . . . 91
6.2.3 Navier-Stokes: Beta 20 1 . . . 94
6.2.4 Navier-Stokes: Beta 75 75 . . . 97
6.2.5 Discussion for the Navier-Stokes problem . . . 98
6.3 Conclusions . . . 99

7 Conclusions and future perspectives **101**

A Mathematical preliminaries **103**

Introduction

This master thesis deals with the Stokes and the Navier-Stokes problem in a reduced order framework and its extension to uncertainty quantification.
These equations are used for modeling fluid systems [41]. In the Stokes case we drop out the convective term obtaining a set of linear equations which can only describe phenomena in which inertial forces are negligible compared to viscous forces. In the Navier-Stokes case we put this term in again, obtaining non-linear equations, more difficult to treat but more realistic for describying physical behaviours. In both cases we will not consider the dependency on time.
After their success in computational fluid dynamics, one of the problems that emerged is that often the numerical simulations take too much computational time. Usually these problems are relevant when the equations depend on some physical/geometrical parameters and we are interested in the solution for several such parameters, as in the many-query problems or real-time simulation problems. In these cases the finite element method or finite volume method, called *full order method*, are too slow and we need something faster. One of the solutions for these problems is to use the *reduced order method* in which the idea is to reconstruct fastly the solution for a certain parameter by a linear combination of precomputed solutions obtained with other parameters, knocking down the computation cost, introducing nevertheless an additional error to the approximation.
Two algorithms are usually used for searching these solutions: the *proper orthogonal decomposition*(POD) and the *greedy* [8].
The former uses an eigenvalue problem and with a compression tecnique that retains only the most important informations.
The latter uses an iterative approach based on searching the solutions with an *error estimator*.
In these methods we always have an *offline* phase where we solve the problem several times, exploring the space of the parameters and storing several useful quantities, such as some solutions, matrices and vectors, making this process very expensive.
After if follow the *online* phase, when we have to solve the problem for a other parameter values we use the quantities stored in the previous phase and we solve the problem fastly.
Subsequently we will introduce stocasticity and therefore uncertainty into the problem.

Uncertainty is everywhere: for example it is intrinsic in our measurements due to the noise that can interfere with and so it has to be taken into account in our mathematical models.
Usual problems including uncertainty are the prediction of a uncertain quantity, parameter estima-

tion, inverse problem [19].
In general we know that uncertainty is strictly related to the randomness and this last one can philosophically be divied in two types: the one due to the noise is what we call *aleatoric uncertainty* [23], the case in which the randomness is intrinsic into the problem, and the second one called *epistemic* where the uncertainty arises from a lack of knowledge.
In any case we will treat mathematically both in the same way.
The uncertainty quantification is introduced in the framework of the Stokes and Navier-Stokes problem because we have said that they can depend on several geometrical or physical parameters that in general could be aftected by some uncertainty and so randomness. In this thesis we want to extend the work done in the elliptic case as for example in [24], [25], to the more complicated fluid dynamics case.
In our work, following these articles, we will write a stochastic formulation of the Stokes and Navier-Stokes equations, treating the randomness as parameters.
When we are working in a probabilistic framework we can use the information related to the different distributions associated to the parameters introducing some weights in the two algorithms mentioned before, to catch the most likely parameters in several way, needed for calculating the solutions used in the linear combination.
We note that when we introduce uncertainties we can use some methods for chosing the parameters, such as the tensor product rule, the Monte-Carlo method, both usually expensive from a computational cost point of view, and the Smolyak rule, cheaper but less precise that the previous ones. We will follow [36], [38], [39].
Let us conclude with the oganization of this master thesis:

- In the first chapter we will recall some notations and prerequisites useful for the covered topics.
- In the second chapter we will formulate the Stokes problem, showing the strong formulation both with the Brezzi formulation and the Babuša one, and introducing a finite element discretization afterwards.
- The third chapter is the most important one because we will introduce the reduced order model for the Stokes problem, we will treat the inf-sup condition problem, solved with the supremizer operator, we will explain the greedy and POD algorithm and the reduced algebraic formulation of the Stokes equations.
- In the fourth chapter we will pass to the Navier-Stokes problem introducing the strong and the weak formulation of it, followed by a finite element discretization. We will explain two type of linearization tecniques necessary for treating the non-linear problem that will appear.
- In the fifth chapter we will introduce the uncertainty, formulating the problem with an intrinsic randomness and with a weighted approach, applied both to the greedy and the POD algorithm.

- In the sixth chapter we will explain three methods for sampling the parameters from a probabilistic distribution: tensor product rule, Smolyak rule and Monte-Carlo method.
- The seventh chapter will be focused on some numerical experiments using RBniCS [3] library, developed at SISSA mathLab. We will study some problems associated to several different probabilistic distributions, observing the strong points and the weaknesses associated to the methods that we have treated in the previous chapters.
- In the last chapters we will come up with a conclusion and we will propose some idea for future works in this field.

We finally thank SISSA in Trieste where we have developed this thesis, in particular the *MathLab* team that have helped us to understand the topics involved and finally the European project *"H2020 ERC CoG AROMA-CFD"* that made this work possible.

September 2019, Trieste and Torino

Chapter 1

General notions about Stokes equations

In this chapter we will recall some notions on the Stokes equations and their weak formulation with both the Brezzi and Babuška approach, stressing important concepts such as the *inf-sup condition*, really important for the reduced problem that we will see in the following chapters, while at the end we will expose the finite element approximation with the relative algebraic problem as in [7].

1.1 The strong and the weak formulation: Brezzi's theory

We introduce the parametric steady Stokes problem [9]. We take a $2D$ domain $\Omega \subset \mathbb{R}^2$ with boundary Γ. The Stokes problem is:

$$\begin{cases} -\nu\Delta\mathbf{u}(\mathbf{x};\boldsymbol{\mu}) + \nabla p(\mathbf{x};\boldsymbol{\mu}) = \mathbf{f}(\mathbf{x};\boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = 0 & \text{in } \Omega(\boldsymbol{\mu}), \\ \mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = \mathbf{0} & \text{on } \Gamma_w(\boldsymbol{\mu}), \\ \mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = \mathbf{g}_{in}(\mathbf{x};\boldsymbol{\mu}) & \text{on } \Gamma_{in}(\boldsymbol{\mu}), \\ \nu\dfrac{\partial \mathbf{u}}{\partial \mathbf{n}}(\mathbf{x};\boldsymbol{\mu}) - p(\mathbf{x};\boldsymbol{\mu})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{out}(\boldsymbol{\mu}), \end{cases} \tag{1.1}$$

where $\mathbf{u}$ is the velocity of the fluid, p the pressure, $\mathbf{f}$ a volume force field, ν a kinematic viscosity, $\mathbf{n}$ is the normal unit vector of the boundary, $\boldsymbol{\mu} = (\mu_1, \mu_2, ...) \in \mathbb{P}$ where each μ_i is a physical or geometrical parameter (we will discuss later in more details what kind of parameters we can have) contained in a finite range and $\mathbb{P}$ is the set of all the parameters, $\mathbf{x} = (x, y)$ is the vector of the spacial coordinates.
We have split the boundary in $\Gamma = \Gamma_w \cup \Gamma_{in} \cup \Gamma_{out}$, depending on the boundary conditions imposed. Hereafter the dependence on $\mathbf{x}$ will be ommitted as well as the one on $\boldsymbol{\mu}$ until the next chapter.

Explaining the two equations, the first vectorial one are the *Stokes momentum equations*, obtained

as the limit of the steady Navier-Stokes momentum equations when $Re \to 0$. They are a set of linear equations and so are easier to treat respect to the original Navier-Stokes ones.
The second one, scalar, is related to the hypothesis of incompressibility.
In fact if we remember the continuity equation for the density ϱ:

$$\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \mathbf{u}) = 0,$$

supposing $\varrho \approx constant$ in space and time, we obtain $\nabla \cdot \mathbf{u} = 0$.
The fourth and the fifth equations are boundary conditions that will be introduced in the weak formulation of this problem.

Now let us pass to consider the weak formulation of the system and so let us see what are the appropriate spaces for the test functions.
For the pressure we take a $L^2(\Omega)$ space [7] but we will see why this is the right space to obtain a meaningful weak formulation of our problem. We will define

$$Q := L^2(\Omega). \tag{1.2}$$

The test space for the velocities has to be chosen according to the Dirichlet boundary conditions. So we have to take $H^1_{\Gamma_D}(\Omega) \times H^1_{\Gamma_D}(\Omega)$ as test space for velocities, where $\Gamma_D := \Gamma_w \cup \Gamma_{in}$. We will call it:

$$V := H^1_{\mathbf{0},\Gamma_D}(\Omega) \times H^1_{\mathbf{0},\Gamma_D}(\Omega). \tag{1.3}$$

We anticipate that in a variational approach an homogeneous Neumann condition in Γ_{out} comes naturally if we do not put any other condition.
So we are searching a solution $\mathbf{u} \in V$.

Before continuing, we need to remember three results for a generic tensor S, a generic vector function $\mathbf{v}$ and a generic scalar function q.
It holds that[1] [4]

$$\nabla \cdot (S^T \mathbf{v}) = S : \nabla \mathbf{v} + (\nabla \cdot S) \cdot \mathbf{v}, \tag{1.4}$$

and (the Gauss theorem):

$$\int_\Omega \nabla \cdot (S^T \mathbf{v})\, d\Omega = \int_{\partial\Omega} S^T \mathbf{v} \cdot \mathbf{n}\, d\Gamma, \tag{1.5}$$

and finally:

$$\nabla \cdot (q\mathbf{v}) = \nabla q \cdot \mathbf{v} + q \nabla \cdot \mathbf{v}. \tag{1.6}$$

Now we want to obtain the weak formulation, so let us begin from the first vectorial equation multiplying for a velocity test function and integrating over the domain:

$$\int_\Omega -\nu \Delta \mathbf{u} \cdot \mathbf{v}\, d\Omega + \int_\Omega \nabla p \cdot \mathbf{v}\, d\Omega = \int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\Omega, \quad \forall \mathbf{v} \in V,$$

[1] with two general tensors it is defined $A : B := \sum_{i,j} A_{ij} \cdot B_{ij}$.

and using (1.4), (1.5) and (1.6) using as tensor $S = \nabla \mathbf{u}$, as vectorial function $\mathbf{v}$ and as scalar p, remembering that $\nabla \cdot \nabla \mathbf{u} = \Delta \mathbf{u}$ we obtain:

$$\nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} \, d\Omega - \int_{\partial\Omega} \Big((\nabla \mathbf{u})^T \mathbf{v} - p\mathbf{v} \Big) \cdot \mathbf{n} \, d\Gamma - \int_\Omega p \nabla \cdot \mathbf{v} \, d\Omega = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, d\Omega, \quad \forall \mathbf{v} \in V. \tag{1.7}$$

We also know that

$$(\nabla \mathbf{u})^T \mathbf{v} \cdot \mathbf{n} = \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{v},$$

and considering the boundary conditions and the fact that we are taking $\mathbf{v} \in V$ we have:

$$\int_{\partial\Omega} \Big((\nabla \mathbf{u})^T \mathbf{v} - p\mathbf{v} \Big) \cdot \mathbf{n} \, d\Gamma = 0.$$

Before moving further we need this theorem [7]:

Theorem 1. *Let $\boldsymbol{g}_{in} \in (H^{1/2}(\partial\Omega))^2$ such that $\int_{\partial\Omega} \boldsymbol{g}_{in} \cdot \boldsymbol{n} = 0$ there exists $\boldsymbol{u}_g \in (H^1(\Omega))^2$ such that $\boldsymbol{g}_{in} = \boldsymbol{u}_g|_{\partial\Omega}$ and $\nabla \cdot \boldsymbol{u}_g = 0$ in Ω. $\boldsymbol{u}_g$ is called lifting function.*

According to this theorem we can split the solution in following way:

$$\mathbf{u} = \mathbf{u}_g + \mathbf{u}^{(0)}, \tag{1.8}$$

with $\mathbf{u}^{(0)} \in V$.
Theoretically we know that $\mathbf{u}_g$ exists, but practically we do not know its expression and so within our codes we have to use some tricks related to the boundary to recover it. On the contrary $\mathbf{u}^{(0)}$ is related to differential equations so, it will became our unknown and from now on we will denote it without any index (we will only write $\mathbf{u}$).

Now the problem is to find a couple $(\mathbf{u}, p)$ such that:

$$\nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} \, d\Omega - \int_\Omega p \nabla \cdot \mathbf{v} \, d\Omega = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, d\Omega - \langle F^g, \mathbf{v} \rangle, \quad \forall \mathbf{v} \in V,$$

where

$$\langle F^g, \mathbf{v} \rangle = \nu \int_\Omega \nabla \mathbf{u}_g : \nabla \mathbf{v} \, d\Omega.$$

Now we have to deal with the scalar equation of incompressibility $\nabla \cdot \mathbf{u} = 0$. We multiply it by a pressure test function $q \in Q$ and integrate:

$$\int_\Omega q \nabla \cdot u = 0, \quad \forall q \in Q.$$

So now it is clear why we have taken $L^2(\Omega)$ for the pressure test function so that this integral makes sense.
Using (1.8), we arrive at:

$$\int_\Omega q \nabla \cdot \mathbf{u}^{(0)} \, d\Omega = \langle G, q \rangle, \quad \forall q \in Q,$$

where

$$\langle G, q\rangle = -\int_\Omega q\nabla\cdot\mathbf{u}_g\,d\Omega,$$

and at the end the weak problem is searching a couple $(\mathbf{u}, p)$ such that:

$$\begin{cases}\nu\displaystyle\int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\,d\Omega - \int_\Omega p\nabla\cdot\mathbf{v}\,d\Omega = \int_\Omega \mathbf{f}\cdot\mathbf{v}\,d\Omega - \langle F^g, \mathbf{v}\rangle, \quad \forall\mathbf{v}\in V,\\[2ex] \displaystyle\int_\Omega q\nabla\cdot\mathbf{u}\,d\Omega = \langle G^g, q\rangle, \quad \forall q\in Q.\end{cases} \tag{1.9}$$

Now we want to rewrite this problem introducing some operators.

We define

$$\langle F, \mathbf{v}\rangle := \int_\Omega \mathbf{f}\cdot\mathbf{v}\,d\Omega - \langle F^g, \mathbf{v}\rangle.$$

We introduce two bilinear forms a and b defined as:

$$a : V\times V\to\mathbb{R},$$

such that

$$a(\mathbf{u},\mathbf{v}) := \nu\int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\,d\Omega,$$

and

$$b : V\times Q\to\mathbb{R},$$

such that

$$b(\mathbf{v}, q) := \int_\Omega q\nabla\cdot\mathbf{v}\,d\Omega.$$

Before the final formulation of the Stokes problem, in which we will use these operators, we have to modify the spaces of the pressure test functions. In fact we note that in general there is a problem with the pressures: if we take a constant c, then

$$\nabla(p + c) = \nabla p,$$

and so the pressure solution is known up to a constant if we search the pressure in $L^2(\Omega)$. We note that this problem is present when we have only homogeneous Dirichlet boundary conditions.
To solve the problem, we require that p belongs to $L^2_0(\Omega)$, a subset of $L^2(\Omega)$, where all the functions have the property that

$$\int_\Omega p\,d\Omega = 0.$$

This is important because

$$\int_\Omega c\, d\Omega = 0 \Leftrightarrow c = 0,$$

and so we do not have the problem of the constants anymore if we take the pressure in this space. So the weak Stokes equations are defined in such a way:

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle F, \mathbf{v} \rangle, \quad \forall \mathbf{v} \in V, \\ b(\mathbf{u}, q) = \langle G, q \rangle, \quad \forall q \in Q. \end{cases} \tag{1.10}$$

To introduce a theorem of existence and uniqueness we have to remind three other operators. The first one is $B : V \to Q'$ such that

$$\langle B\mathbf{v}, q \rangle_Q := b(\mathbf{v}, q), \quad \forall q \in Q.$$

So B is the divergence operator $(\nabla \cdot)$.
The second one is $B^T : Q \to V'$ such that

$$\langle B^T q, \mathbf{v} \rangle_V = b(\mathbf{v}, q), \quad \forall \mathbf{v} \in V.$$

So B^T is the gradient operator (∇).
The last one is $A : V \to V'$ such that

$$\langle A\mathbf{v}, \mathbf{w} \rangle_V = a(\mathbf{v}, \mathbf{w}), \quad \forall \mathbf{w} \in V.$$

With these operators we can observe that [2]

$$L^2_0(\Omega) = {}^{L^2(\Omega)}\!/_{ker B^T},$$

where / is the quotient operation and we will use this information in finite element discretization. Now we can introduce the Brezzi theorem for the existence and uniqueness of the solution [5]:

Theorem 2. *Supponing that a and b are both continuous bilinear forms and*

- *a is coercive on $ker B$, i.e. $\exists \alpha > 0$ such that $a(\boldsymbol{v}, \boldsymbol{v}) \geq \alpha ||\boldsymbol{v}||^2_V, \quad \forall \boldsymbol{v} \in ker B$.*
- *b satisfies the inf-sup condition, i.e. $\exists \beta > 0$ such that*

$$\inf_{q \in {}^{Q}\!/_{ker B^T}} \sup_{\boldsymbol{v} \in V} \frac{b(\boldsymbol{v}, q)}{||\boldsymbol{v}||_V ||p||_Q} \geq \beta, \tag{1.11}$$

or equivalently

$$\forall q \in {}^{Q}\!/_{ker B^T} \ \exists \boldsymbol{v} \in V \text{ such that } b(\boldsymbol{v}, q) \geq \beta ||\boldsymbol{v}||_V ||p||_{\boldsymbol{Q}}, \tag{1.12}$$

[2] the kernel of B is defined in such a way $ker B = \{\mathbf{v} \in V | b(\mathbf{v}, q) = 0, \quad q \in Q\}$.

so $\forall F \in V'$, $\forall G \in \mathcal{I}mB$ *we have that* $\exists!(\boldsymbol{u}, p) \in (V, {}^{Q}\!/_{ker B^T})$ *solution of* (1.10) *and we have a continuous dependence on the data:*

$$\begin{cases} ||u||_V \leq \frac{1}{\alpha}||F||_{V'} + \frac{1}{\beta}(1 + \frac{||a||}{\alpha})||G||_{Q'}, \\ ||p||_{{}^{Q}\!/_{ker B^T}} \leq \frac{1}{\beta}(1 + \frac{||a||}{\alpha})||F||_{V'} + \frac{||a||}{\beta^2}(1 + \frac{||a||}{\alpha})||G||'_Q. \end{cases} \tag{1.13}$$

1.2 Babuška's theory

In this section we will see the Babuška formulation and the equivalence with the Brezzi one according to [12].
Let us consider a general continuous bilinear form

$$\mathcal{B}(\cdot, \cdot) : U \times W \to \mathbb{R},$$

and the problem of finding $u \in U$ such that:

$$\mathcal{B}(u, v) = \langle f, v \rangle, \quad \forall v \in W. \tag{1.14}$$

This problem is well posed if and only if the Babuška condition holds:

$$\inf_{u \in U} \sup_{v \in W} \frac{\mathcal{B}(u, v)}{||u||_U ||v||_W} = \inf_{v \in W} \sup_{u \in U} \frac{\mathcal{B}(u, v)}{||u||_U ||v||_W} = \beta_{BA} > 0, \tag{1.15}$$

and in this case the solution is unique and it satisfies

$$||u||_U \leq \frac{||f||_{W'}}{\beta_{BA}}.$$

For seeing the equivalence with Brezzi's theory we take as space

$$U = V \times Q,$$

and as bilinear form

$$\mathcal{B}((\mathbf{u}, p), (\mathbf{v}, q)) = a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + b(\mathbf{u}, q),$$

and

$$\langle f, (\mathbf{v}, q) \rangle = \langle F, \mathbf{v} \rangle + \langle g, q \rangle.$$

So the Stokes problem with the Babuška formulation is to find $(\mathbf{u}, p)$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + b(\mathbf{u}, q) = \langle F, \mathbf{v} \rangle + \langle g, q \rangle. \tag{1.16}$$

The two approaches are equivalent because the functional forms with q as test function are independent from those ones with $\mathbf{v}$. So if we solve the problem splitting the equations or summing them is equivalent. So from now on we can work interchangeably with both formulations.
This one will be very important in the reduced framework for the greedy algorithm in next chapter but for the moment we come back to the Brezzi approach to introduce the finite element discretization.

1.3 Finite element discretization with Brezzi's formulation

Now we take two finite dimensional spaces $V_{N_\delta} \subset V$ and $Q_{N_\delta} \subset Q$ for using a Galerkin approximation of the problem (1.10), with dimensions N_V and N_p, both multiples of N_δ [7].
In this case we search a couple $(\mathbf{u}_{N_\delta}, p_{N_\delta}) \in V_{N_\delta} \times Q_{N_\delta}$, solution of:

$$\begin{cases} a(\mathbf{u}_{N_\delta}, \mathbf{v}_{N_\delta}) + b(\mathbf{v}_{N_\delta}, p_{N_\delta}) = \langle F, \mathbf{v}_{N_\delta} \rangle, \quad \forall \mathbf{v}_{N_\delta} \in V_{N_\delta}, \\ b(\mathbf{u}_{N_\delta}, q_{N_\delta}) = \langle G, q_{N_\delta} \rangle, \quad \forall q_{N_\delta} \in Q_{N_\delta}. \end{cases} \tag{1.17}$$

Now we would like a similar theorem to that 2 in the discrete case [5]:

Theorem 3. *Assuming that a and b are both continuous bilinear forms on the discretized spaces and*

- *a is coercive on $kerB_{N_\delta}$, i.e. $\exists \alpha > 0$ such that $a(\boldsymbol{v}_{N_\delta}, \boldsymbol{v}_{N_\delta}) \geq \alpha ||\boldsymbol{v}_{N_\delta}||_V^2, \quad \forall \boldsymbol{v}_{N_\delta} \in kerB_{N_\delta}$.*
- *b satisfies the inf-sup condition, i.e. $\exists \beta_h > 0$ such that*

$$\beta_{N_\delta} := \inf_{q_{N_\delta} \in {}^{Q_{N_\delta}}\!/_{kerB_{N_\delta}^T}} \sup_{\boldsymbol{v}_{N_\delta} \in V_{N_\delta}} \frac{b(\boldsymbol{v}_{N_\delta}, q_{N_\delta})}{||v_{N_\delta}||_{V_{N_\delta}} ||p_{N_\delta}||_{Q_{N_\delta}}} \geq \beta_h, \tag{1.18}$$

or equivalently

$$\forall q_{N_\delta} \in {}^{Q_{N_\delta}}\!/_{kerB_{N_\delta}^T} \ \exists \boldsymbol{v}_{N_\delta} \in V_{N_\delta} \text{ such that } b(\boldsymbol{v}_{N_\delta}, q_{N_\delta}) \geq \beta_h ||\boldsymbol{v}_{N_\delta}||_{V_{N_\delta}} ||p_{N_\delta}||_{Q_{N_\delta}}, \tag{1.19}$$

so $\forall F \in V'_{N_\delta}$, $\forall G \in \mathcal{I}mB'_{N_\delta}$ we have that $\exists! (\boldsymbol{u}_{N_\delta}, p_{N_\delta}) \in (V_{N_\delta}, {}^{Q_{N_\delta}}\!/_{kerB_{N_\delta}^T})$ solution of (1.17) and we have a continuous dependence on the data:

$$\begin{cases} ||u_{N_\delta}||_{V_{N_\delta}} \leq \dfrac{1}{\alpha} ||F||_{V'_{N_\delta}} + \dfrac{1}{\beta_{N_\delta}} \big(1 + \dfrac{||a||}{\alpha}\big) ||G||_{Q'_{N_\delta}}, \\ ||p_{N_\delta}||_{{}^{Q_{N_\delta}}\!/_{kerB_{N_\delta}^T}} \leq \dfrac{1}{\beta} \big(1 + \dfrac{||a||}{\alpha}\big) ||F||_{V'_{N_\delta}} + \dfrac{||a||}{\beta_{N_\delta}^2} (1 + \dfrac{||a||}{\alpha}) ||G||'_{Q_{N_\delta}}. \end{cases} \tag{1.20}$$

We introduce A_{N_δ}, B_{N_δ}, $B_{N_\delta}^T$ as the restrinction of A, B, B^T to the finite dimensional spaces.
The question is if, knowing that the hypotesis of theorem 2 holds, it also holds those ones of the theorem 3 since $Q_{N_\delta} \subset Q$ and $V_{N_\delta} \subset V$.
In other words we would like that the hypothesis that hold in the infinite cases were inherited in the discrete case so that we could use the theorem again.
But this is not possible.
In fact $kerB_{N_\delta} \not\subset kerB$. To show it we remember the two definitions:

$$kerB = \{\mathbf{v} \in V | b(\mathbf{v}, q) = 0, \quad \forall q \in Q\}, \tag{1.21}$$

and

$$kerB_{N_\delta} = \{\mathbf{v}_{N_\delta} \in V_{N_\delta} | b(\mathbf{v}_{N_\delta}, q_{N_\delta}) = 0, \quad \forall q_{N_\delta} \in Q_{N_\delta}\}. \tag{1.22}$$

It is true that $V_{N_\delta} \subset V$ and so we have less possible candidates to stay in the kernel, but in discrete case we have less constraints because $Q_{N_\delta} \subset Q$. So in general there are no relations between them and so the bilinear form a is no coercive anymore.
It is the same for $kerB^T_{N_\delta}$ and so in general we also loose the inf-sup condition.
For the first case we have no other choice that suppose again that the coercivity hold also for the finite dimensional case.
An alternative can be to suppose that the coercivity holds on the entire space V and so it would also hold for V_{N_δ}, but this hypothesis it is not true in general.
For the second problem we can construct a Fortin operator [10] to stabilize, i.e. to recover the stability, using for example a Taylor-Hood finite element [7] or other type of stabilisation, for example *Streamline Upwind Petrov-Galerkin*(SUPG)[11].
We are talking about the inf-sup condition (1.18) because if it does not hold we have that:

$$\exists q_{N_\delta} \in Q_{N_\delta} \quad \text{s.t. } b(\mathbf{v}_{N_\delta}, q_{N_\delta}) = 0, \quad \forall \mathbf{v}_{N_\delta} \in V_{N_\delta},$$

and so we lose the uniqueness of the pressure solution even though we have it in the infinite dimensional problem. These other solutions are in general called *spurious pressure modes*.
We will have the same problem in the reduced methods.
We conclude this section introducing the algebraic problem.[7]

For doing this we take a basis of the two spaces, for examples

$$\{\boldsymbol{\Phi_i}\}_{i=1,\dots,N_V},$$

for the velocities, where N_V is the dimension of the velocity space and

$$\{\psi_i\}_{i=1,\dots,N_p},$$

for the pressure where N_p is the dimension of the pressure space.
After we decompose our solutions onto this basis in such a way:

$$u_{N_\delta}(\mathbf{x}) = \sum_{i=1}^{N_V} u^i_{N_\delta} \boldsymbol{\Phi_i}(\mathbf{x}),$$

and

$$p_{N_\delta}(\mathbf{x}) = \sum_{i=1}^{N_V} p^i_{N_\delta} \psi_i,$$

and introducing the matrices A, B defined in such a way

$$\begin{aligned} [A]_{ij} &= a(\boldsymbol{\Phi_j}, \boldsymbol{\Phi_i}), \\ [B]_{ij} &= b(\boldsymbol{\Phi_j}, \psi_i), \end{aligned} \tag{1.23}$$

and the vectors $\mathbf{F}_{N_\delta}$, $\mathbf{G}_{N_\delta}$, $\mathbf{u}_{N_\delta}$, p_{N_δ} defined like

$$\begin{aligned}(\mathbf{F}_{N_\delta})_i &= \langle F, \mathbf{\Phi_i} \rangle, \\ (\mathbf{G}_{N_\delta})_i &= \langle G, \psi_i \rangle, \\ (\mathbf{u}_{N_\delta})_i &= \mathbf{u}^i_{N_\delta}, \\ (p_{N_\delta})_i &= p^i_{N_\delta},\end{aligned} \tag{1.24}$$

we arrive at the classical finite element algebraic formulation:

$$\begin{cases} A\mathbf{u}_{N_\delta} + B^T p_{N_\delta} = \mathbf{F}_{N_\delta}, \\ B\mathbf{u}_{N_\delta} = \mathbf{G}_{N_\delta}. \end{cases} \tag{1.25}$$

We can rewrite this equation introducing:

$$S = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix},$$
$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_{N_\delta} \\ p_{N_\delta}, \end{bmatrix},$$
$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{N_\delta} \\ \mathbf{G}_{N_\delta} \end{bmatrix}.$$

This matrix is block symmetric due to the symmetry of A, so it has real eigenvalues. If we want that the system $S\mathbf{U} = \mathbf{F}$ has a solution, the matrix A must have all the eigenvalues different from 0.
Reading (1.25) from the first equation we have

$$\mathbf{u}_{N_\delta} = A^{-1}(\mathbf{F}_{N_\delta} - B^T p_{N_\delta}),$$

and putting it in the second equation we obtain:

$$BA^{-1}B^T p_{N_\delta} = BA^{-1}\mathbf{F} + \mathbf{G}.$$

In this case we have uniqueness of the solution only when $ker B^T = \{\mathbf{0}\}$ that is an equivalent formulation of the inf-sup condition [7].

To resume, we have seen how much is important the *inf-sup* condition and it will be one of the key problem in reduced methods.

1.4 Conclusions

In this chapter we have presented the parametric Stokes equations in the strong and weak form with an associate theorem of existence and uniqueness of the solution based on the *inf-sup condition*. We

got a formulation both with Brezzi's theory and the equivalent Babuška's one. In the former case we introduce two bilinear forms while in the latter only one, linear combinations of the previous ones.
We have finally introduced a finite element discretization with the associated linear algebraic system and we have stressed the importance of the *inf-sup condition* to avoid spurious modes of pressure.

Chapter 2

Reduced ordel modelling for Stokes equations

In this chapter we will talk about reduced order method for Stokes equations in case of domains with parametric geometrical dependences and parametric physical dependences such as diffusivity, length of the domain or the angle of incidence of a flux, etc. References on this topic are in [8], [13], [14] and [15].
In the first part we will explain what a reduced approach is and what are the benefits. Next, we will see why we need to trace back our domain to a reference one by an affine mapping. Then we will go into details into the creation of the reduced basis introducing the *supremizer operator* that gives us velocities useful for satisfying the *inf-sup* condition in the reduced space. At the end we will talk about the two main algorithms for generating the basis for the reduced spaces, the *POD* and the *greedy*, explaining the different computational costs and scopes.

2.1 Introduction to the reduced problem

To start we want to remember the equations (1.10):

$$\begin{cases} a(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) + b(\mathbf{v}, p; \boldsymbol{\mu}) = F(\mathbf{v}; \boldsymbol{\mu}), \quad \forall \mathbf{v} \in V, \\ b(\mathbf{u}, q; \boldsymbol{\mu}) = G(q; \boldsymbol{\mu}), \quad \forall q \in Q. \end{cases} \tag{2.1}$$

In this case we have two main issues:

- Ω is dependent on a vector parameter $\boldsymbol{\mu} \in \mathbb{P}$. This is a problem because we need a different mesh for all the possible parameters, that is prohibitive, and then also the quantities in the problem depend on the geometry.
  So we want to have some reference domain where we will do our simulations independently

from $\boldsymbol{\mu}$ in such a way to transpose the geometric dependence into an algebraic one and have a general treatment of this one.

- we have some physical dependencies into the terms a or b. For example the viscosity ν can be one of the uncertainties of the problem. In this case we have an algebraic dependence from the beginning.

So the problem depends on $\boldsymbol{\mu}$ and without any expedient the cost of one simulation is of order of N_δ as we can see in (1.25).
Sometimes this can be too expensive and we want to reduce this cost. This is the case of a real-time simulation and the time for a single one is too long or a many-query problem where to afford too many simulation can be prohibitive.
The goal of the reduced method approach is to lower the cost to $N \ll N_\delta$ and to let these tasks possible.
This gain has a price to pay as we will see presenting some ideas behind the reduced method.
The implementation of the method we say that is divided in two phases [8]:

- *offline phase*: this is the most expensive phase. Here we compute the truth solution for several parameters chosen according to one of two algorithms called *greedy* and *POD*. The cost hence depends on multiple of N_δ.
  Whatever algorithm we chose, at the end we have N solutions whose linear span is called *"reduced space"*. In addition in this phase we memorize informations associated with these solutions and other important ones for the next phase.

- *online phase*: in this phase we are ready to do fast simulations and for doing them we search the solution in the reduced space using the quantities stored before. The cost of this phase must depend only on N and this allows to satisfy real-time simulations and many-query problems.

2.2 General theory behind reduced order methods

In this section we will go into details of the theory of reduced order methods.[8]
First of all it is classically introduced the concept of *solution manifold*, i.e. the set of the solutions of the parametrized problem under variation of the parameters $\boldsymbol{\mu} \in \mathbb{P}$, where the set $\mathbb{P}$ can have a continuum cardinality. With a reduced basis approach we want to approximate this manifold space with a lower dimensional one of N elements, solutions of the problem for certain parameters.
To better understand we go deeper into the problem. In general we have to solve the equations

(2.1) varying the parameters involved. So we have a manifold of solutions:

$$\mathcal{M} = \{\big(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})\big) | \boldsymbol{\mu} \in \mathbb{P}\} \subset V \times Q,$$

where each pair is a solution of a Stokes problem with a different parameter.
Usually we do not know the exact solution so we use a numerical method to approximate it and obtain $\big(\mathbf{u}_{N_\delta}(\boldsymbol{\mu}), p_{N_\delta}(\boldsymbol{\mu})\big)$, called *truth solution*. In the same way we can obtain:

$$\mathcal{M}_{N_\delta} = \{\big(\mathbf{u}_{N_\delta}(\boldsymbol{\mu}), p_{N_\delta}(\boldsymbol{\mu})\big) | \boldsymbol{\mu} \in \mathbb{P}\}.$$

The hypothesis is that $\mathcal{M}_{N_\delta}$ approximates well $\mathcal{M}$.
In our mind we want to create a N-dimensional space $\mathbb{V}_{rb}$ (that sometimes we will also denote with V_N) that well approximates $\mathcal{M}_{N_\delta}$. This space is composed by N truth solutions ξ_i and so we can indicate $\mathbb{V}_{rb}$ as:

$$\mathbb{V}_{rb} = \text{span}\{\xi_1, ..., \xi_N\} \subset V_{N_\delta}.$$

Now we want to know how good is the approximation from this new space and for this we can see that:

$$||\mathbf{u}(\boldsymbol{\mu}) - \mathbf{u}_{rb}(\boldsymbol{\mu})||_V \leq ||\mathbf{u}(\boldsymbol{\mu}) - \mathbf{u}_{N_\delta}(\boldsymbol{\mu})||_V + ||\mathbf{u}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{u}_{rb}(\boldsymbol{\mu})||_V,$$

using the triangular inequality.
We point out that we have the same results for the pressure.
Let us see the different terms involved in the addition. The first one depends on the numerical method used for searching the truth solution and so we expect that it is small enough. This cames from the assumption that $\mathcal{M}_{N_\delta}$ approximates well $\mathcal{M}$. Usually we suppose that the second one has an exponential decay with N, i.e. augmenting the number of basis we obtain a better approximation.
Before introducing the two algoritms for the generation of the basis we need to consider the geometric dependency to obtain a general formulation to work with.
In this one the problem will be refered to a reference domain. This is important for what we have said before but also because the reduced space is generated from the *span* of several solutions and so we need a common domain to compare and combine them.

2.3 Pull back to the reference domain

The first step for going in this matter is that we can have geometrical and/or physical dependencies. We will reformulate the problem such that we will only have an algebraic dependency that we will treat in a unique way.
Let us begin with the **geometrical dependency**. The physical dependency will be implied until otherwise specified.
To start we have to split the domain Ω in several subdomains such that $\Omega = \bigcup_{r=1}^{R} \Omega^r$ and we rewrite the several functional forms according to the separation.
We can first note that:

$$\nabla \mathbf{u} : \nabla \mathbf{v} = \sum_{k,j=0} \frac{\partial u_k}{\partial x_j} \frac{\partial v_k}{\partial x_j} = \sum_{k,j=0} \sum_{i=0} \delta_{ij} \frac{\partial u_k}{\partial x_i} \frac{\partial v_k}{\partial x_j},$$

so we have, introducing $v_{ij} = \nu\delta_{ij}$, using the Einstein convention for the indices:

$$a(\mathbf{u},\mathbf{v}) = \sum_{r=1}^{R}\int_{\Omega^r} v_{ij}\frac{\partial \mathbf{u}}{\partial x_i}\cdot\frac{\partial \mathbf{w}}{\partial x_j}\,d\Omega,$$

and for the other forms:

$$b(\mathbf{v},p) = -\sum_{r=1}^{R}\int_{\Omega^r} p\nabla\cdot\mathbf{w}\,d\Omega,$$

$$F(\mathbf{v}) = \sum_{r=1}^{R}\int_{\Omega^r} \mathbf{f}\cdot\mathbf{v}\,d\Omega.$$

Now we want to trace back each domain $\Omega^r(\mu)$ to a reference one $\hat{\Omega}^r$ with an affine trasformation of the form:

$$\hat{\mathbf{x}} = T(\mathbf{x}) = G^r(\boldsymbol{\mu})\mathbf{x} + \mathbf{g}^r, \quad 1 \le r \le R\ ,$$

Now using the chain rule:

$$\frac{\partial}{\partial x_i} = \frac{\partial \hat{x}_j}{\partial x_i}\frac{\partial}{\partial \hat{x}_j} = G^r_{ij}(\boldsymbol{\mu})\frac{\partial}{\partial \hat{x}_j},$$

and introducing

$$\hat{\mathbf{u}}(\hat{x}) := \mathbf{u}(T^{-1}(\hat{x})), \quad \hat{p}(\hat{x}) := p(T^{-1}(\hat{x})), \tag{2.2}$$

the bilinear forms become:

$$\hat{a}(\hat{\mathbf{u}},\hat{\mathbf{v}}) = \sum_{r=1}^{R}\int_{\hat{\Omega}^r}\frac{\partial \hat{\mathbf{u}}}{\partial \hat{x}_i}\big(G^r_{ii'}\nu_{i'j'}G^r_{jj'}\det(G^r(\boldsymbol{\mu})^{-1})\big)\frac{\partial \hat{\mathbf{w}}}{\partial \hat{x_j}}\,d\hat{\Omega},\ \forall\hat{\mathbf{w}}\in\hat{V},$$

and

$$\hat{b}(\hat{\mathbf{v}},\hat{p}) = -\sum_{r=1}^{R}\int_{\hat{\Omega}^r}\hat{p}\big(G^r_{ij}(\boldsymbol{\mu})det(G^r(\boldsymbol{\mu}))^{-1}\big)\frac{\partial \hat{w}_j}{\partial \hat{x}_i}\,d\hat{\Omega}, \quad \forall\hat{\mathbf{w}}\in\hat{Q},$$

and the force term that we remember is composed by $\langle F,\hat{\mathbf{w}}\rangle = \langle F_s,\hat{\mathbf{w}}\rangle + \langle F^0,\hat{\mathbf{w}}\rangle$ becames

$$\langle \hat{F}_s,\hat{\mathbf{w}}\rangle = \sum_{r=1}^{R}\int_{\hat{\Omega}^r}\big(\hat{f}^r det(G^r(\boldsymbol{\mu}))^{-1}\big)\hat{\mathbf{w}}\,d\hat{\Omega},$$
$$\langle \hat{F}^0,\hat{\mathbf{w}}\rangle = -\hat{a}(\hat{L}_{\mathbf{g}_{in}},\hat{\mathbf{w}}).$$

Now we redefine:

$$\hat{\nu}^r_{ij}(\boldsymbol{\mu}) := G^r_{ii'}(\boldsymbol{\mu})\nu_{i'j'}G^r_{jj'}det(G^r(\boldsymbol{\mu}))^{-1}, \quad \text{for } 1\le i,i',j,j'\le 2,\ r=1,...,R,$$
$$\chi^r_{ij}(\mu) := G^r_{ij}det(G^r(\boldsymbol{\mu}))^{-1}.$$

Let us introduce now the terms:

$$\Theta^{q(i,j,r)}(\boldsymbol{\mu}) := \hat{\nu}_{ij}^r(\boldsymbol{\mu}), \quad a^{q(i,j,r)}(\hat{\mathbf{u}}, \hat{\mathbf{w}}) := \int_{\hat{\Omega}^r} \frac{\partial \hat{\mathbf{u}}}{\partial \hat{x}_i} \cdot \frac{\partial \hat{\mathbf{w}}}{\partial \hat{x}_j} \, d\hat{\Omega},$$
$$\Phi^{s(i,j,r)}(\boldsymbol{\mu}) := \chi_{ij}^r(\boldsymbol{\mu}), \quad b^{s(i,j,r)}(\hat{p}, \hat{\mathbf{w}}) := -\int_{\hat{\Omega}^r} \hat{p} \frac{\partial \hat{w}_i}{\partial \hat{x}_j} \, d\hat{\Omega},$$

where $q(i,j,k)$ is a function to enumerate the different terns (i,j,k).
We have the *affine decomposition hypothesis* holds, i.e. that we can split in such a way:

$$\hat{a}(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \boldsymbol{\mu}) = \sum_{q=1}^{Q^a} \Theta^q(\boldsymbol{\mu}) a^q(\hat{\mathbf{u}}, \hat{\mathbf{v}}), \tag{2.3a}$$

$$\hat{b}(\hat{p}, \hat{\mathbf{w}}; \boldsymbol{\mu}) = \sum_{s=1}^{Q^b} \Phi^s(\boldsymbol{\mu}) b^s(\hat{p}, \hat{\mathbf{w}}), \tag{2.3b}$$

with a^q and b^q two bilinear forms independent from $\boldsymbol{\mu}$ such that:

$$a^q : \hat{V} \times \hat{V} \to \mathbb{R},$$
$$b^q : \hat{Q} \times \hat{V} \to \mathbb{R},$$

where $\hat{V} := T(V)$, $\hat{Q} := T(Q)$ and finally:

$$\Theta^q : \mathbb{P} \to \mathbb{R},$$
$$\Phi^s : \mathbb{P} \to \mathbb{R},$$

independent from pressure and velocity.
As we will understand in the next chapter this is a very important hypothesis to reduce the cost of the reduced approach in the online phase.
So the problem becames to find a solution $(\hat{\mathbf{u}}(\boldsymbol{\mu}), \hat{p}(\boldsymbol{\mu})) \in \hat{V} \times \hat{Q}$ such that:

$$\begin{cases} \hat{a}(\hat{\mathbf{u}}(\boldsymbol{\mu}), \hat{\boldsymbol{w}}; \boldsymbol{\mu}) + \hat{b}(\hat{p}(\boldsymbol{\mu}), \hat{\mathbf{w}}; \boldsymbol{\mu}) = \langle \hat{F}, \hat{\boldsymbol{w}} \rangle(\boldsymbol{\mu}), \quad \forall \hat{w} \in \hat{V}, \\ \hat{b}(\hat{q}, \hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \langle \hat{G}, \hat{q} \rangle(\boldsymbol{\mu}), \quad \forall \hat{q} \in \hat{Q}. \end{cases}$$

From now on we will work into the reference domain so we will not keep the hat and it will be implied.

2.4 Inf-sup condition problem, supremizer operator and reduced basis

In this section we will talk about the philosophy behind the reduced method approach. We will introduce two spaces $V_N \subset V_{N_\delta}$ and $Q_N \subset Q_{N_\delta}$. The first space has dimension $\mathcal{N}_V \ll N_V$ and the

second one $\mathcal{N}_p \ll N_p$ and they are a multiple of a number N. The idea is to project onto these finite reduced spaces to obtain an algebraic problem with much less degrees of freedom, i.e. that $N \ll N_\delta$.
The first task is how to chose the two spaces, that is what basis is the optimal one to mantain the algebraic stability and a low approximation error.
The problem of the algebraic stability is related to the *inf-sup condition*. As we have said before, one of the issues that arrives when we introduce a Galerkin approximation with the finite dimensional spaces $V_{N_\delta} \subset V$ and $Q_{N_\delta} \subset Q$ is the fact that is not true in general that the *inf-sup condition* holds in the finite dimensional spaces, that is it is not true in general that:

$$\exists \beta_0 > 0 : \quad \beta(\boldsymbol{\mu}) = \inf_{q \in Q_{N_\delta}} \sup_{\mathbf{w} \in V_{N_\delta}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_M} \geq \beta_0, \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{2.4}$$

In a reduced approach we have the same problem because we are still doing a projection.
For this purpose we introduce the *supremizer operator* following [13]:

$$T^{\boldsymbol{\mu}} : Q_{N_\delta} \to V_{N_\delta},$$

that associates a pressure to a velocity and defined such that:

$$(T^{\boldsymbol{\mu}} q, \mathbf{w})_V = b(q, \mathbf{w}; \boldsymbol{\mu}), \quad \forall \mathbf{w} \in V_{N_\delta}, \tag{2.5}$$

that is equivalent to:

$$T^{\boldsymbol{\mu}} q = \arg \sup_{\mathbf{w} \in V_{N_\delta}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V}, \quad \forall \mathbf{w} \in V_{N_\delta}. \tag{2.6}$$

Proof. To understand this last equivalence we can think the supremizer as the operator that associates for each fixed pressure $\overline{q}$ the relative Riesz rappresentation [6] of the operator

$$f(\mathbf{w}; \boldsymbol{\mu}) := b(\overline{q}, \mathbf{w}; \boldsymbol{\mu}).$$

This cames directly from the definition (2.5).
So for the *Riesz theorem* [6] we know that

$$||f||_{V'} = ||T^{\boldsymbol{\mu}} q||_V,$$

but

$$\sup_{\mathbf{w} \in V_{N_\delta}} \frac{f(\mathbf{w})}{||\mathbf{w}||} = ||f||_{V'} = ||T^{\boldsymbol{\mu}} q||_V = \frac{f(T^{\boldsymbol{\mu}} q)}{||T^{\boldsymbol{\mu}} q||_V},$$

and so

$$T^{\boldsymbol{\mu}} q = \arg \sup_{\mathbf{w} \in V_{N_\delta}} \frac{f(\mathbf{w})}{||\mathbf{w}||_V}, \quad \forall \mathbf{w} \in V_{N_\delta}.$$

□

Now we want to create two reduced spaces for velocity and pressure such that the *inf-sup condition*

holds, using the supremizer operator. In this case we take a discretization of the parametric space $\mathbb{P}$ that we call $\mathbb{P}_N^{\boldsymbol{\mu}} = \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, ..., \boldsymbol{\mu}^N\}$. We suppose to have the solution of the Stokes problem with each of these parameters, so we know $\mathbf{u}(\boldsymbol{\mu})$ and $p(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathbb{P}_N^{\boldsymbol{\mu}}$. We will call for simplicity $\mathbf{u}(\boldsymbol{\mu}_i) := \zeta_i$ and $p(\boldsymbol{\mu}_i) := \xi_i$.
So the reduced space for the pressure is:

$$Q_N := span\{\xi_i : i = 1, ..., N\}, \tag{2.7}$$

while for the velocity:

$$V_N^{\boldsymbol{\mu}} := span\{\zeta_i,\ i = 1, ..., N; T^{\boldsymbol{\mu}}\xi_i,\ i = 1, ..., N\}. \tag{2.8}$$

We can see now that this space depens on the parameter $\boldsymbol{\mu}$ due to the dependence on N in the supremizer operator. We will understand that the velocities $T^{\boldsymbol{\mu}}\xi_i$ that we have added are such that the *inf-sup condition* holds.
So the reduced problem consists in finding the solution $(\mathbf{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in V_N^{\boldsymbol{\mu}} \times Q_N$ such that:

$$\begin{cases} a(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) + b(p_N(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) = \langle F, \mathbf{w} \rangle, \quad \forall \mathbf{w} \in V_N^{\boldsymbol{\mu}}, \\ b(q, \mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \langle G, q \rangle, \quad \forall q \in Q_N. \end{cases} \tag{2.9}$$

Now we define the *inf-sup condition* for the reduced basis as:

$$\beta_N(\boldsymbol{\mu}) = \inf_{q \in Q_N} \sup_{\mathbf{w} \in V_N^{\boldsymbol{\mu}}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_Q}. \tag{2.10}$$

Theorem 4. *Defined the inf-sup condition for the reduced problem as in* (2.10) *and that one for the truth problem as in* (1.18) *we have that:*

$$\beta_N(\boldsymbol{\mu}) \geq \beta_{N_\delta}(\boldsymbol{\mu}) \geq \beta_0 > 0, \quad \forall \boldsymbol{\mu}. \tag{2.11}$$

Proof.

$$\beta_{N_\delta}(\boldsymbol{\mu}) = \inf_{q \in Q_{N_\delta}} \sup_{\mathbf{w} \in V_{N_\delta}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_Q} \leq \inf_{q \in Q_N} \sup_{\mathbf{w} \in V_{N_\delta}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_Q}, \tag{2.12}$$

because $Q_N \subset Q_{N_\delta}$. Now we want to change V_{N_δ} into V_N and for this purpose we need the supremizer and in particular the equation (2.6):

$$\inf_{q \in Q_N} \sup_{\mathbf{w} \in V_{N_\delta}} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_Q} = \inf_{q \in Q_N} \frac{b(q, T^{\boldsymbol{\mu}} q; \boldsymbol{\mu})}{||T^{\boldsymbol{\mu}} q||_V ||q||_Q} \leq \inf_{q \in Q_N} \sup_{\mathbf{w} \in V_N} \frac{b(q, \mathbf{w}; \boldsymbol{\mu})}{||\mathbf{w}||_V ||q||_Q} = \beta_N(\boldsymbol{\mu}),$$

and the last inequality is due to the fact that we have put the supremizer in the reduced space. □
Now we want to rewrite the space $V_N^{\boldsymbol{\mu}}$ to obtain an algebraic formulation. So we remind the affine

decomposition of the bilinear form b:

$$b(p, \mathbf{w}; \boldsymbol{\mu}) = \sum_{s=1}^{Q^b} \Phi^s(\boldsymbol{\mu}) b^s(p, \mathbf{w}), \tag{2.13}$$

and so we expect the same decomposition of:

$$T^{\boldsymbol{\mu}}\xi = \sum_{q=1}^{Q^b} \Phi^q(\boldsymbol{\mu}) T^q \xi.$$

With this decomposition we can rewrite in a more compact way:

$$V_N^{\boldsymbol{\mu}} = span\left\{ \boldsymbol{\sigma_i} := \sum_{k=1}^{\overline{Q}^b} \Phi^k(\boldsymbol{\mu}) \boldsymbol{\sigma}_{ki}, \quad \text{for} \quad i = 1, ..., 2N \right\},$$

where $\overline{Q}^b = Q^b + 1, \Phi^{\overline{Q}^b} = 1$.
For $i = 1, ..., N$:

$$\begin{aligned} &\boldsymbol{\sigma}_{ki} = 0, \text{ for } k = 1, ..., Q^b, \\ &\boldsymbol{\sigma}_{\overline{Q}^b i} = \zeta_i = \mathbf{u}_N(\boldsymbol{\mu}^i), \end{aligned}$$

while for $i = N + 1, ..., 2N$:

$$\begin{aligned} &(\boldsymbol{\sigma}_{ki}, \mathbf{w})_V = b^k(\xi_{i-N}, \mathbf{w}), \quad \forall \mathbf{w} \in V_{N_\delta}, \quad \text{for } k = 1, ..., Q^b, \\ &\boldsymbol{\sigma}_{\overline{Q}^b i} = 0. \end{aligned}$$

With this basis we can search the solutions in the reduced space with this form:

$$\mathbf{u}_N(\boldsymbol{\mu}) = \sum_{j=1}^{2N} u_{N_j}(\boldsymbol{\mu}) \boldsymbol{\sigma_j}, \tag{2.14a}$$

$$p_N(\boldsymbol{\mu}) = \sum_{j=1}^{N} p_{N_j}(\boldsymbol{\mu}) \xi_j. \tag{2.14b}$$

For finding the coefficients u_{N_j} and p_{N_j} we introduce this decomposition in the reduced problem (2.9) obtaining:

$$\begin{cases} \sum_{j=1}^{2N} A_{ij}^{\boldsymbol{\mu}} u_{N_j}(\boldsymbol{\mu}) + \sum_{k=1}^{N} B_{ik}^{\boldsymbol{\mu}} p_{Nk}(\boldsymbol{\mu}) = F_i^{\boldsymbol{\mu}}, \quad 1 \le i \le 2N, \\ \sum_{j=1}^{2N} B_{jm} u_{N_j}(\boldsymbol{\mu}) = G_l^{\boldsymbol{\mu}}, \quad 1 \le m \le N, \end{cases} \tag{2.15}$$

where we have introduced the matrices $A^{\boldsymbol{\mu}}$, $B^{\boldsymbol{\mu}}$ and vectors $F^{\boldsymbol{\mu}}$, $G^{\boldsymbol{\mu}}$ such as:

$$
\begin{aligned}
A^{\boldsymbol{\mu}}_{ij} &= \sum_{k=1}^{Q^a} \sum_{k'=1}^{\overline{Q}^b} \sum_{k''=1}^{\overline{Q}^b} \Theta^k(\boldsymbol{\mu}) \Phi^{k'}(\boldsymbol{\mu}) \Phi^{k''}(\boldsymbol{\mu}) a(\boldsymbol{\sigma}_{k'i}, \boldsymbol{\sigma}_{k''j})^k, \quad 1 \leq i, j \leq 2N, \\
B^{\boldsymbol{\mu}}_{il} &= \sum_{k=1}^{Q^b} \sum_{k'=1}^{\overline{Q}^b} \Phi^k(\boldsymbol{\mu}) \Phi^{k'}(\boldsymbol{\mu}) b(\boldsymbol{\sigma}_{k'i}, \xi_l)^k, \quad 1 \leq i \leq 2N, \quad 1 \leq l \leq N, \\
F^{\boldsymbol{\mu}}_{i} &= \sum_{j=1}^{\overline{Q}^b} \Phi^j(\boldsymbol{\mu}) \langle F, \boldsymbol{\sigma}_{k'i} \rangle, \quad 1 \leq i \leq 2N, \\
G^{\boldsymbol{\mu}}_{l} &= \langle G^0, \xi_l \rangle, \quad 1 \leq l \leq N.
\end{aligned}
$$

So we can reconduce our problem to the algebraic one:

$$
\begin{pmatrix} A^{\boldsymbol{\mu}} & B^{\boldsymbol{\mu}} \\ B^{\boldsymbol{\mu}T} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_N(\boldsymbol{\mu}) \\ \mathbf{p}_N(\boldsymbol{\mu}) \end{pmatrix} = \begin{pmatrix} \mathbf{F}^{\boldsymbol{\mu}} \\ \mathbf{G}^{\boldsymbol{\mu}} \end{pmatrix}. \tag{2.16}
$$

So in this case we have a vector of dimensions of order of N with respect to before that was of order of N_δ.
Now we want to discuss a little more the computational cost of finding this solution and understand the importance of the affine decomposition hypothesis.
This hypothesis is significant in the online phase when we are solving (2.16). If we see the decomposition of the several terms involved in the equations we can compute in the offline phase the terms that are independent from $\boldsymbol{\mu}$ such as $a(\boldsymbol{\sigma}_{k'i}, \boldsymbol{\sigma}_{k''j})^k, \langle F, \boldsymbol{\sigma}_{k'i} \rangle, b(\boldsymbol{\sigma}_{k'i}, \xi_l)^k$ and store them. In the online phase we reconstruct the different terms involved in the system with a cost that scales with $\mathcal{O}((Q_a + \overline{Q}_b))$ and we solve the system with a cost of $\mathcal{O}(N)$, independent from N_δ.
When this assumption does not hold there are other tecniques such as *EIM* and *DEIM* [20], [21], two linearization tecniques for obtaining the affine decomposition again.

2.5 Greedy and POD algorithms

Now we want a method to effectively generate the reduced spaces for velocity and pressure. To do this there are two classical algorithms: the *greedy* algorithm and the proper orthogonal decomposition (*POD*) one.
To explain them we have to discretize the space of the parameters $\mathbb{P}$ with a discrete space $\mathbb{P}_h \subset \mathbb{P}$ obtained taking a finite number of parameters according to some rule. Several methods are available to do that and in particular in the weighted approach this is one of the problems that we treat. In any case we expect that the space $\mathcal{M}_\delta(\mathbb{P}_h) = \{\big(\mathbf{u}_{N_\delta}(\boldsymbol{\mu}), p_\delta(\boldsymbol{\mu})\big) | \boldsymbol{\mu} \in \mathbb{P}_h\}$ well approximates $\mathcal{M}_\delta$.

2.5.1 Proper Orthogonal Decomposition (POD)

In this case the goal is to project the velocity solution that, as we have said, belongs to V_{N_δ}, into a space V_N such that this one minimizes the quantity:

$$\sqrt{\frac{1}{M}\sum_{\boldsymbol{\mu}\in\mathbb{P}_h}\inf_{\mathbf{v}_{rb}\in V_N}||\mathbf{u}_{N_\delta}(\boldsymbol{\mu})-\mathbf{v}_{rb}||_V^2}, \tag{2.17}$$

where M is the cardinality of $\mathbb{P}_h$. This error is called *projection error*. So we are searching the solution that minimizes the ℓ^2 norm distance between the truth solution and the reduced solution. For doing this we will follow the work of [16].
In the case of Stokes equations we have the same goal for the pressure but the treatment is similar to the velocity one and so we will treat only this last one. We note that we do not write them together, as a pair, because we are searching a basis for the velocity and one for the pressure, separately.
We first chose the parameters $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_M$ composing $\mathbb{P}_h$ and obtain the corresponding solutions $\mathbf{u}_{N_\delta}(\boldsymbol{\mu}_1), \mathbf{u}_{N_\delta}(\boldsymbol{\mu}_2), ..., \mathbf{u}_{N_\delta}(\boldsymbol{\mu}_M)$. We will denote

$$\boldsymbol{\psi}_m := \mathbf{u}_{N_\delta}(\boldsymbol{\mu}_m).$$

For searching the optimal space we have to introduce the linear and symmetric operator:

$$C(\mathbf{v}_\delta) := \frac{1}{M}\sum_{m=1}^{M}(\mathbf{v}_\delta, \boldsymbol{\psi}_m)_V\boldsymbol{\psi}_m,$$

with $\mathbf{v}_\delta \in V_M := \text{span}\{\mathbf{u}_{N_\delta}(\boldsymbol{\mu})|\boldsymbol{\mu}\in\mathbb{P}_h\}$.
Due to the symmetry we know that we have a sequence of eigenvalues and normalized eigenfunctions $(\lambda_n, \boldsymbol{\xi}_n) \in \mathbb{R}\times V_M$ of C (so with $||\boldsymbol{\xi}_i||_V = 1$) that form a basis, satisfying

$$(C(\boldsymbol{\xi}_n), \boldsymbol{\psi}_i)_V = \lambda_n(\xi_n, \boldsymbol{\psi}_i)_V, \quad 1\le i\le M. \tag{2.18}$$

After we have solved this eigenvalue problem, we order the eigenfunctions according to the decreasing value of the related eigenvalue $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_M \ge 0$. After this sorting we take the first N eigenfunctions and their spanned space will be the reduced space V_N.
To explain this truncation we first introduce the projection operator $P_N : V \to V_N$ defined as:

$$P_N(\boldsymbol{f}) = \sum_{i=1}^{N}(\boldsymbol{f}, \boldsymbol{\xi}_i)_V\boldsymbol{\xi}_i.$$

We want to prove that:

$$\sqrt{\frac{1}{M}\sum_{i=1}^{M}||\boldsymbol{\psi}_m - P_N(\boldsymbol{\psi}_m)||_V^2} = \sqrt{\sum_{i=N+1}^{M}\lambda_i}. \tag{2.19}$$

Proof.

$$\frac{1}{M}\sum_{m=1}^{M}||\boldsymbol{\psi}_m - P_N(\boldsymbol{\psi}_m)||_V^2 = \frac{1}{M}\sum_{m=1}^{M}||\sum_{n=1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V\boldsymbol{\xi}_n - \sum_{n=1}^{N}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V\boldsymbol{\xi}_n||_V^2 =$$
$$\frac{1}{M}\sum_{m=1}^{M}||\sum_{n=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V\boldsymbol{\xi}_n||_V^2 = \frac{1}{M}\sum_{m=1}^{M}(\sum_{n=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V\boldsymbol{\xi}_n, \sum_{j=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_j)_V\boldsymbol{\xi}_j)_V =$$
$$\frac{1}{M}\sum_{m=1}^{M}\sum_{n=N+1}^{M}\sum_{j=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V(\boldsymbol{\psi}_m,\boldsymbol{\xi}_j)_V(\boldsymbol{\xi}_n,\boldsymbol{\xi}_j)_V,$$

and for the ortonormality we have that $(\boldsymbol{\xi}_n,\boldsymbol{\xi}_j)_V = \delta_{nj}$ and so:

$$\sum_{m=1}^{M}\sum_{n=N+1}^{M}\sum_{j=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V(\boldsymbol{\psi}_m,\boldsymbol{\xi}_j)_V\delta_{nj} = \sum_{m=1}^{M}\sum_{n=N+1}^{M}(\boldsymbol{\psi}_m,\boldsymbol{\xi}_n)_V^2,$$

but we know that $\sum_{n=1}^{M}(\boldsymbol{\xi}_n,\boldsymbol{\psi}_m)_V\boldsymbol{\psi}_m = \lambda_n\boldsymbol{\xi}_n$, so doing the scalar product with $\boldsymbol{\xi}_n$ we obtain:

$$\sum_{m=1}^{M}(\boldsymbol{\xi}_n,\boldsymbol{\psi}_m)_V^2 = \lambda_n||\boldsymbol{\xi}_n||_V^2 = \lambda_n,$$

for the normality of $\boldsymbol{\xi}_n$. Introducing this equivalence in the previous one we obtain:

$$\frac{1}{M}\sum_{m=1}^{M}||\boldsymbol{\psi}_m - P_N(\boldsymbol{\psi}_m)||_V^2 = \sum_{n=N+1}^{M}\lambda_n.$$

□

So it is clear that if we recall how the eigenfunctions have been ordered, the choice of the basis has been done to minimize the error of projection along all the possible spaces using the eigenvalues as indicators.

However we note that at the beginning we do not know the proper choice of M and N. For the first value we can say that we take the value such that $\mathbb{P}_h$ well approximates $\mathbb{P}$ in a sense that we have to choose. On the other hand for searching N we can use an energy function $E(N) = \sum_{i=1}^{N}\lambda_i$ and we take the $\overline{N}$ such that

$$E(\overline{N}) \geq \text{fixed tolerance}.$$

We can now pass to the algebraic formulation.

For introducing it we need the *snapshot* matrices, those containing on each column the vector of the different degrees of freedom corresponding to the solutions varying the parameters:

$$S_{\mathbf{u}} := \left[\mathbf{u}(\boldsymbol{\mu}^1)|\mathbf{u}(\boldsymbol{\mu}^2)|...|\mathbf{u}(\boldsymbol{\mu}^M)\right] \in \mathbb{R}^{N_{\mathbf{u}}\times M}, \quad S_p := \left[\mathbf{p}(\boldsymbol{\mu}^1)|\mathbf{p}(\boldsymbol{\mu}^2)|...|\mathbf{p}(\boldsymbol{\mu}^M)\right] \in \mathbb{R}^{N_p\times M},$$

and also the mass matrices:

$$(X_{\mathbf{u}})_{ij} := \left(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i\right)_V, \quad (X_p)_{ij} := \left(\xi_j, \xi_i\right)_Q,$$

where $\{\boldsymbol{\phi}_i\}_{i=1}^{N_\delta}, \{\xi_i\}_{i=1}^{N_\delta}$ are the Lagrangian basis of the truth problem respectively for velocity and pressure.
Next, we introduce the matrices $C_{\mathbf{u}}, C_p$ associated to the relative operator:

$$C_{\mathbf{u}} := S_{\mathbf{u}}^T X_{\mathbf{u}} S_{\mathbf{u}} \in \mathbb{R}^{M\times M}, \quad C_p := S_p^T X_p S_p \in \mathbb{R}^{M\times M}. \tag{2.20}$$

So the eigenvalue problem can be solved computing:

$$C_{\mathbf{u}}\underline{\boldsymbol{\psi}}_{\mathbf{u}}^n = \lambda_{\mathbf{u}}^n \underline{\boldsymbol{\psi}}_{\mathbf{u}}^n, \quad C_p \underline{\xi}_p^n = \lambda_p^n \underline{\xi}_p^n, \tag{2.21}$$

and the vectors $\underline{\boldsymbol{\psi}}_{\mathbf{u}}^n$ and $\underline{\xi}_p^n$ contain the values with respect to the degrees of freedom.
Finally we reconstruct the reduced basis functions $\{\underline{\boldsymbol{\chi}}_i\}_{i=1,\dots,N_{\mathbf{u}}}$ for the velocity and $\{\underline{\zeta}_i\}_{i=1,\dots,N_p}$ for the pressure in the following way:

$$\underline{\boldsymbol{\chi}}_i = \frac{1}{\sqrt{\lambda_i^{\mathbf{u}}}} S_{\mathbf{u}} \underline{\boldsymbol{\psi}}_{\mathbf{u}}^i, \quad \underline{\zeta}_i = \frac{1}{\sqrt{\lambda_i^p}} S_p \underline{\xi}_p^i. \tag{2.22}$$

2.5.2 Greedy algorithm

As we have just seen, in the *POD* algorithm we obtain all the basis at the same time and this requires M evaluations of the truth solution. Sometimes this can be too expensive and so we need other approaches. In the *greedy* one we do several iterations and in each one we add one new basis, each time solving a truth problem.
We will present the algorithm as presented in [14].
We first introduce the general idea for this type of algorithm and we will go into detail later for the Stokes equations.
To run the algorithm we need an error estimator $\Delta_N^{N_\delta}(\boldsymbol{\mu})$ such that

$$||u_\delta(\boldsymbol{\mu}) - u_{rb}(\boldsymbol{\mu})||_V \leq \Delta_N^{N_\delta}(\boldsymbol{\mu}),$$

where u_δ is the truth solution of a generic problem, u_{rb} is the reduced solution of this one and V is the space where the solution u is defined.
In the case of the Stokes equations we will see that u will be the pair of velocity and pressure.
We require that this error estimator is also sharp, i.e.

$$\exists C > 0 \quad \text{such that} \quad C \cdot \Delta_N^{N_\delta}(\boldsymbol{\mu}) \leq ||u_\delta(\boldsymbol{\mu}) - u_{rb}(\boldsymbol{\mu})||_V,$$

for having the complete control over the error from above and below.
The main characteristic of the error estimator is that it has to be cheap to be computed.

Then passing to the algorithm, it works in this way: we begin searching randomly a parameter $\boldsymbol{\mu}_1$, so a truth solution $u_\delta(\boldsymbol{\mu}_1)$ and creating $V_N^1 := \text{span}\{u_\delta(\boldsymbol{\mu}_1)\}$; after when we are at the generic n-th step, we have a $n-1 < N$ dimensional space $V_N^{n-1} := \text{span}\{u_\delta(\boldsymbol{\mu}_1), u_\delta(\boldsymbol{\mu}_2), ..., u_\delta(\boldsymbol{\mu}_{n-1})\}$ and we add to this space a function according to the following rule: we select the parameter $\boldsymbol{\mu}_n$ such that:

$$\boldsymbol{\mu}_n = \arg\max_{\boldsymbol{\mu}\in\mathbb{P}_h} \Delta_{n-1}^{N_\delta}(\boldsymbol{\mu}),$$

so we search the solution $u_\delta(\boldsymbol{\mu}_n)$ and finally we add it to V_N^{n-1} to create the space:

$$V_N^n := \text{span}\{u_\delta(\boldsymbol{\mu}_1), u_\delta(\boldsymbol{\mu}_2), ..., u_\delta(\boldsymbol{\mu}_{n-1}), u_\delta(\boldsymbol{\mu}_n)\}.$$

We repeat this operation until the estimator is under a certain tolerance for all the parameters in the sample space.
We note in fact that $\Delta_N^{N_\delta}(\boldsymbol{\mu})$ changes according also to the reduced solution created with the V_N^{n-1}, as well as the change of the parameter, and this is the reason why we do successive iterations.
Now we note the two main differences between a *POD* algorithm and a *greedy* one.

In the first one we are minimizing the error with a ℓ^2 norm, that we have discretized with a finite sum, while in the case of the greedy we are working with a ℓ^∞ norm. So the reduced basis that we obtain will be usually different. The second difference is that in the *POD* case we have to solve an eigenvalue problem depending on M so we cannot take $\mathbb{P}_h$ too big. Instead in the greedy case we have to evaluate the error estimator several times but this is cheap. The expensive phase is the evaluation of N truth solutions and so we can afford a bigger space $\mathbb{P}_h$.
Another difference is that if we want to add other elements to the reduced space, in the *POD* case we have to compute the eigenvalue problem again and this is very expensive. In the greedy case we have only one additional problem to solve.
So it seems that the greedy algorithm is always better and we could wonder why the *POD* algorithm is used. The fact is that in general we do not know the error estimator that is specific of the equations that we are solving. On the contrary the *POD* approach works in general and so it gives us always a reduced basis space to our problem.

Formalism behind the greedy algorithm

Let us introduce the space $Y := V \times Q$, the pair of pressure and velocity $\mathbf{U} = (\mathbf{u}, p) \in Y$ and the norm:

$$||\mathbf{U}||_Y := (||\mathbf{u}||_V^2 + ||p||_Q^2)^{1/2},$$

induced by the scalar product:

$$(\mathbf{V}, \mathbf{W})_Y := (\mathbf{v}, \mathbf{w})_V + (p, q)_Q,$$

for $\mathbf{V} = (\mathbf{v}, p)$ and $\mathbf{W} = (\mathbf{w}, q)$.
As we can see, in this case the solution u that we have used before is now $u = \mathbf{U} = (\mathbf{u}, p)$.

We can define in the same way $\mathbf{U}_{N_\delta} = (\mathbf{u}_{N_\delta}, p_{N_\delta}) \in Y_{N_\delta} = V_{N_\delta} \times Q_{N_\delta}$ and $\mathbf{U}_N = (\mathbf{u}_N, p_N) \in Y_N = V_N \times Q_N$.
As we have said before we have to take M parameters for chosing the future elements of the basis and for doing this we will use the error estimator, that in this case is such that:

$$||\mathbf{U}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{U}_N(\boldsymbol{\mu})||_Y = (||\mathbf{u}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{u}_N(\boldsymbol{\mu})||_V^2 + ||p_{N_\delta}(\boldsymbol{\mu}) - p_N(\boldsymbol{\mu})||_Q^2)^{1/2} \leq \Delta_N^{N_\delta}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathbb{P}, \forall N, N_\delta.$$

We will see later the expression of this estimator.
Now let us take the maximum number of basis N_{max} that we desire, a tolerance for the error ϵ_{tol}^{rb} and a training sample $\mathbb{P}_h \subset \mathbb{P}$, of cardinality M.
This is the greedy algorithm:

```
choose at random μ_1 ∈ P_h   and create   S_1 = {μ_1}
set  Q_1^N = span{ξ_1 := p_{N_δ}(μ_1)}   V_1^N = span{ζ_1 := u_{N_δ}(μ_1), T_p^{μ_1} ξ_1}
for  i = 2 : N_max
     μ_i = arg max_{μ∈P_h}(μ)
     ε_{i-1} = Δ_{i-1}^N(μ_i)
     if  ε_{i-1} ≤ ε_tol^rb
          N_max = i - 1
     end
     S_i = S_{i-1} ∪ μ_i
     Q_i^N = Q_{i-1}^N + span{ξ_i := p_{N_δ}(μ_i)}
     X_i^{N,μ} = X_{i-1}^{N,μ} + span{ζ_i := u_{N_δ}(μ_i), T_p^{μ_i} ξ_i}
end
```

So at each iteration the algorithm selects a couple (ζ_i, ξ_i) and adds it to the previous space generated by the *span* of the previous solutions founded. The solution is selected in such a way that it maximizes the error between the truth solution and the reduced one generated with the reduced space of the previous iteration. The main fact is that this error is estimated by $\Delta_{i-1}^N(\boldsymbol{\mu}^i)$ that has to be inexpensive. We can also note that we can afford a larger dataset of $\mathbb{P}_h$. In fact in this case we only compute the truth problem or N_{max} times or until we do not reach the convergence but this can be only a number of times $i \leq N_{max}$.
We finally note that the reduced space for the velocity contain also the supremizers.

We pass now to determinate the error estimator using a Babuška approach for the Stokes equations. In particular we will use the two equations (1.14) and (1.16). We note that there are two other ways to work on the problem, explained in [17] and [18].
As we have done previously we introduce a bilinear form $\mathcal{B} : Y \times Y \to \mathbb{R}$ such that:

$$\mathcal{B}(\mathbf{V}, \mathbf{W}; \boldsymbol{\mu}) := a(\mathbf{v}, \mathbf{w}; \boldsymbol{\mu}) + b(p, \mathbf{w}; \boldsymbol{\mu}) + b(q, \mathbf{v}; \boldsymbol{\mu}), \tag{2.23}$$

and a linear form $f : Y \to \mathbb{R}$ such that:

$$f(\mathbf{W}) := F(\mathbf{W}) + g(q), \tag{2.24}$$

where $\mathbf{V} := (\mathbf{v}, p)$ and $\mathbf{W} = (\mathbf{w}, q)$.
We can also introduce the continuity constant:

$$\gamma(\boldsymbol{\mu}) = \sup_{\mathbf{V}\in Y} \sup_{\mathbf{W}\in Y} \frac{\mathcal{B}(\mathbf{V}, \mathbf{W}; \boldsymbol{\mu})}{||\mathbf{W}||_Y ||\mathbf{V}||_Y} < +\infty, \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{2.25}$$

and the Babuška inf-sup stability condition:

$$\exists \beta_0^b > 0 \text{ such that } \beta^b(\boldsymbol{\mu}) := \inf_{\mathbf{W}\in Y} \sup_{\mathbf{V}\in Y} \frac{\mathcal{B}(\mathbf{W}, \mathbf{V}; \boldsymbol{\mu})}{||\mathbf{W}||_Y ||\mathbf{V}||_Y} \geq \beta_0^b, \quad \forall \boldsymbol{\mu} \in \mathbb{P}. \tag{2.26}$$

These two constant are needed for the well posedness of the Stokes problem formulated with the Babuška formulation.
We can define the same constant in the case of the finite element space and the reduced space and we call them:

$$\beta_{N_\delta}^b(\boldsymbol{\mu}) := \inf_{\mathbf{W}\in Y^{N_\delta}} \sup_{\mathbf{V}\in Y^{N_\delta}} \frac{\mathcal{B}(\mathbf{W}, \mathbf{V}; \boldsymbol{\mu})}{||\mathbf{W}||_Y ||\mathbf{V}||_Y}, \quad \beta_N^b(\boldsymbol{\mu}) := \inf_{\mathbf{W}\in Y^N} \sup_{\mathbf{V}\in Y^N} \frac{\mathcal{B}(\mathbf{W}, \mathbf{V}; \boldsymbol{\mu})}{||\mathbf{W}||_Y ||\mathbf{V}||_Y}.$$

To satisfy the stability in the reduced case we need this condition holds:

$$\beta_N^b \geq \beta_{N_\delta}^b \geq \beta_0^b.$$

Now we introduce the residuals $r_{\mathbf{u}}(\cdot; \boldsymbol{\mu})$ and $r_p(\cdot; \boldsymbol{\mu})$ defined as:

$$r_{\mathbf{u}}(\mathbf{w}; \boldsymbol{\mu}) := F(\mathbf{w}) - a(\mathbf{u}_N, \mathbf{w}; \boldsymbol{\mu}) - b(p_N, \mathbf{w}; \boldsymbol{\mu}) = a(\mathbf{e}_{\mathbf{u}}(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) + b(e_p(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}), \tag{2.27a}$$

$$r_p(q; \boldsymbol{\mu}) := g(q) - b(q, \mathbf{u}_N; \boldsymbol{\mu}) = b(q, \mathbf{e}_{\mathbf{u}}; \boldsymbol{\mu}), \tag{2.27b}$$

with

$$\mathbf{e}_{\mathbf{u}}(\boldsymbol{\mu}) := \mathbf{u}_{N_\delta} - \mathbf{u}_N,$$
$$e_p(\boldsymbol{\mu}) := p_{N_\delta} - p_N.$$

Using the definition of β^{N_δ} we have:

$$\beta_{N_\delta}^b(\boldsymbol{\mu}) ||\mathbf{U}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{U}_N||_Y \leq \sup_{\mathbf{W}\in Y^{N_\delta}} \frac{\mathcal{B}(\mathbf{U}_{N_\delta} - \mathbf{U}_N, \mathbf{W}; \boldsymbol{\mu})}{||\mathbf{W}||_Y},$$

and introducing $r(\mathbf{W}; \boldsymbol{\mu}) := r_{\mathbf{u}}(\mathbf{w}; \boldsymbol{\mu}) + r_p(q; \boldsymbol{\mu})$ and its dual norm:

$$||r(\cdot; \boldsymbol{\mu})||_{Y'} := \sup_{\mathbf{V}\in Y_{N_\delta}} \frac{r(\mathbf{V}; \boldsymbol{\mu})}{||\mathbf{V}||_Y},$$

noting that:

$$\mathcal{B}(\mathbf{U}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{U}_N, \mathbf{W}; \boldsymbol{\mu}) = r(\mathbf{W}; \boldsymbol{\mu}) \quad \forall \mathbf{W} \in Y_{N_\delta}$$

we obtain:

$$||\mathbf{U}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{U}_N(\boldsymbol{\mu})||_Y \leq \frac{||r(\cdot;\boldsymbol{\mu})||_{Y'}}{\beta_{LB}(\boldsymbol{\mu})} := \Delta_N^{N_\delta}(\boldsymbol{\mu}),$$

where β_{LB} is a lower bound of the inf-sup constant $\beta^b_{N_\delta}$.
We can rewrite this inequality in a decoupled way:

$$||\mathbf{u}_{N_\delta}(\boldsymbol{\mu}) - \mathbf{u}_N(\boldsymbol{\mu})||^2_V + ||p_{N_\delta} - p_N||^2_Q \leq \frac{1}{\beta^2_{LB}(\boldsymbol{\mu})}\bigg(||r_{\mathbf{u}}(\cdot;\boldsymbol{\mu})||^2_{V'} + ||r_p(\cdot;\boldsymbol{\mu})||^2_{Q'}\bigg),$$

with

$$||r_{\mathbf{u}}(\cdot;\boldsymbol{\mu})||_{V'} := \sup_{\mathbf{w}\in V_{N_\delta}} \frac{r_{\mathbf{u}}(\mathbf{w};\boldsymbol{\mu})}{||\mathbf{v}||_V}, \quad ||r_p(\cdot;\boldsymbol{\mu})||_{Q'} := \sup_{q\in Q_{N_\delta}} \frac{r_p(q;\boldsymbol{\mu})}{||q||_Q},$$

and these two norms are such that:

$$||r(\cdot;\boldsymbol{\mu})||^2_{Y'} = ||r_{\mathbf{u}}(\cdot;\boldsymbol{\mu})||^2_{V'} + ||r_p(\cdot;\boldsymbol{\mu})||^2_{Q'}. \tag{2.28}$$

So now we know that the posterior error bound is:

$$\Delta_N^{N_\delta}(\boldsymbol{\mu}) := \frac{||r(\cdot;\boldsymbol{\mu})||_{Y'}}{\beta_{LB}(\boldsymbol{\mu})}. \tag{2.29}$$

For the β_{LB} we refer to the appendix of [14] where a *SCM* approach is used while now we will explain how to compute the residual norm.
For this last part in which we will find the norm of the residual we will introduce the Riesz representation of $r_{\mathbf{u}}(\cdot;\boldsymbol{\mu})$ and $r_p(\cdot;\boldsymbol{\mu})$ that is the two vectors $\hat{\boldsymbol{e}}_{\mathbf{u}}(\boldsymbol{\mu}) \in V_{N_\delta}$ and $\hat{e}_p(\boldsymbol{\mu}) \in Q_{N_\delta}$ such that:

$$\begin{aligned}(\hat{\boldsymbol{e}}_{\mathbf{u}}(\boldsymbol{\mu}), \mathbf{w})_V &= r_{\mathbf{u}}(\mathbf{w};\boldsymbol{\mu}), \quad \forall \mathbf{w} \in V_{N_\delta},\\ (\hat{e}_p(\boldsymbol{\mu}), q)_Q &= r_p(q;\boldsymbol{\mu}), \quad \forall q \in Q_{N_\delta}.\end{aligned}$$

So according with what we have written in (2.27a) and (2.27b) we obtain:

$$a(\mathbf{e}_{\mathbf{u}}(\boldsymbol{\mu}), \mathbf{w};\boldsymbol{\mu}) + b(\mathbf{e}_p(\boldsymbol{\mu}), \mathbf{w};\boldsymbol{\mu}) = (\hat{\boldsymbol{e}}_p(\boldsymbol{\mu}), \mathbf{w})_V, \quad \forall \mathbf{w} \in V_{N_\delta}, \tag{2.30}$$

$$b(q, \mathbf{e}(\boldsymbol{\mu});\boldsymbol{\mu}) = (\hat{e}_p(\boldsymbol{\mu}), q)_Q, \quad \forall q \in Q_{N_\delta}, \tag{2.31}$$

and in addition from the Riesz theorem we know that:

$$||r_{\mathbf{u}}(\cdot;\boldsymbol{\mu})||_{V'} = ||\hat{\boldsymbol{e}}_{\mathbf{u}}(\boldsymbol{\mu})||_V, \quad ||r_p(\cdot;\boldsymbol{\mu})||_{Q'} = ||\hat{e}_p||_Q.$$

The next step is to suppose the affine decomposition of $\mathcal{B}(\mathbf{V}, \mathbf{W};\boldsymbol{\mu}) = \sum_{q=1}^{Q_a+2Q_b} \hat{\Theta}(\boldsymbol{\mu})\mathcal{B}^q(\mathbf{V}, \mathbf{W})$ where

$$\begin{aligned}&\hat{\Theta}^q(\boldsymbol{\mu}) = \Theta^q(\boldsymbol{\mu}), \quad q = 1, ..., Q_a,\\ &\hat{\Theta}^{q+Q_a}(\boldsymbol{\mu}) = \hat{\Theta}^{q+Q_a+Q_b}(\boldsymbol{\mu}) = \Phi^q(\boldsymbol{\mu}), \quad q = 1, ...Q_b,\end{aligned}$$

and

$$\mathcal{B}^q(\mathbf{V},\mathbf{W}) = a^q(\mathbf{v},\mathbf{w}), \quad q = 1,...,Q_a,$$
$$\mathcal{B}^q(\mathbf{V},\mathbf{W}) = b^{q-Q_a}(p,\mathbf{w}), \quad q = Q_a+1,...,Q_a+Q_b,$$
$$\mathcal{B}^q(\mathbf{V},\mathbf{W}) = b^{q-Q_a-Q_b}(q,\mathbf{v}), \quad q = Q_a+Q_b+1,...,Q_a,+2Q_b,$$

remembering (2.3a) and (2.3b).
Denoting our pair of solutions $\mathbf{U}_N = (\mathbf{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in \mathbb{R}^{3N}$ and using the decomposition that we have introduced in (2.14a) and (2.14b) we obtain:

$$r(\mathbf{W};\boldsymbol{\mu}) = F(\mathbf{W}) - \mathcal{B}(\mathbf{U}_N(\boldsymbol{\mu}),\mathbf{W};\boldsymbol{\mu}) = F(\mathbf{W}) - \sum_{j=1}^{3N} U_{Nj}(\boldsymbol{\mu}) \sum_{q=1}^{\hat{Q}} \hat{\Theta}^q(\boldsymbol{\mu})\mathcal{B}(\boldsymbol{\Phi}_j,\mathbf{W}), \tag{2.32}$$

with $\hat{Q} = Q_a + 2Q_b$ and

$$\boldsymbol{\Phi}_j := (\boldsymbol{\sigma_j}, 0), \quad j = 1,...,2N,$$
$$\boldsymbol{\Phi}_j := (\mathbf{0}, \xi_j), \quad j = 2N+1,...,3N.$$

Introducing now $\hat{\boldsymbol{e}}(\boldsymbol{\mu}) := (\hat{\boldsymbol{e}}_{\mathbf{u}}(\boldsymbol{\mu}), \hat{e}_p(\boldsymbol{\mu}))$ we obtain

$$(\hat{\boldsymbol{e}}(\boldsymbol{\mu}),\mathbf{W})_Y = (\hat{\boldsymbol{e}}_{\mathbf{u}}(\boldsymbol{\mu}),\mathbf{w})_V + (\hat{e}_p(\boldsymbol{\mu}),q)_Q =$$
$$F(\mathbf{W}) - \sum_{j=1}^{3N} U_{Nj}(\boldsymbol{\mu}) \sum_{q=1}^{\hat{Q}} \hat{\Theta}^q(\boldsymbol{\mu})\mathcal{B}(\boldsymbol{\Phi}_j,\mathbf{W}),$$

and so we have that:

$$\hat{\boldsymbol{e}}(\boldsymbol{\mu}) = \mathcal{F} + \sum_{q=1}^{\hat{Q}} \sum_{j=1}^{3N} \hat{\Theta}^q(\boldsymbol{\mu}) U_{Nj}(\boldsymbol{\mu}) \hat{\mathcal{B}}_j^q, \tag{2.33}$$

where $\mathcal{F} \in Y_{N_\delta}, \hat{\mathcal{B}}_j^q \in Y_{N_\delta}$ and they are such that:

$$(\mathcal{F},\mathbf{W})_Y = F(\mathbf{W}), \quad \forall \mathbf{W} \in Y^{N_\delta},$$
$$(\mathcal{B}_j^q,\mathbf{W})_Y = -\mathcal{B}(\boldsymbol{\Phi}_j,\mathbf{W}), \quad \forall \mathbf{W} \in Y^{N_\delta}, \quad 1 \le n \le 3N, \quad 1 \le q \le \hat{Q}.$$

So we have that:

$$||\hat{e}(\boldsymbol{\mu})||_Y^2 = \Big(\mathcal{F} + \sum_{q=1}^{\hat{Q}} \sum_{j=1}^{3N} \hat{\Theta}(\boldsymbol{\mu}) U_{Nj}(\boldsymbol{\mu}) \mathcal{B}_j^q, \mathcal{F} + \sum_{q'=1}^{\hat{Q}} \sum_{j'=1}^{3N} \hat{\Theta}'(\boldsymbol{\mu}) U_{Nj'}(\boldsymbol{\mu}) \mathcal{B}_{j'}^{q'}\Big)_Y = \tag{2.34}$$

$$(\mathcal{F},\mathcal{F})_Y + \sum_{q=1}^{\hat{Q}} \sum_{j=1}^{3N} \hat{\Theta}^q(\boldsymbol{\mu}) U_{Nj}(\boldsymbol{\mu}) \Big\{ 2(\mathcal{F},\mathcal{B}_j^q)_Y + \sum_{q'}^{Q_a} \sum_{j'}^{N} \hat{\Theta}^{q'}(\boldsymbol{\mu}) U_{Nj'}(\boldsymbol{\mu}) (\mathcal{B}_j^q, \mathcal{B}_{j'}^{q'})_Y \Big\}. \tag{2.35}$$

So in the offline phase we solve for $\mathcal{F}$ and $\mathcal{B}_j^q$ and we can compute the residual norm. But if we store these quantities we can use them in the online phase for an estimation of the error in the case we are not dealing with a benchmark simulation where we know the real solution in the continuum. As we see, to estimate the residual norm in the online phase we need to compute $\hat{\Theta}^q(\boldsymbol{\mu})$ for $1 \leq q \leq \hat{Q}$ and U_{Nj} for $1 \leq j \leq 3N$ and we need to do all the multiplications and sums. So at the end the computation cost is $\mathcal{O}(\hat{Q}^2 9N^2)$ which is independent from N_δ.

2.6 Conclusions

In this chapter we have seen how to treat the parametric Stokes equations introducing the pull back to a reference domain in the case of geometrical dependency of the problem, to convert it into an algebraic one. We have subsequently presented the reduced methods explaining the benefits of splitting the simulation process in an offline and an online phase and how a reduced basis, the main element in these methods, has to be in general. In particular we have definied the *supremizer operator* and we have seen that it is important to use it for creating a reduced space so that the *inf-sup* condition holds.
In the final sections we have explained two algorithms for the generation of the reduced basis: the greedy algorithm and the POD one.
They lead to different results and have different properties: the former uses a L^∞ norm with an iterative approach, usually cheap, to select the basis and requires an error estimator dependent from the problem we deal with, while the POD works using a projection with a L^2 norm and it does not need anything but can be very expensive.

Chapter 3

Reduced order modelling for Steady Navier-Stokes equations

In this chapter we will talk about the steady Navier-Stokes equations with some geometrical and/or physical parameters. The main difference with the Stokes problem is the presence of the convective term that introduces some non-linearity but makes the equations more realistic to describe some phenomena.
The chapter is organized as follows: in the first part we will introduce the strong formulation of the equations following [37], the weak one following [16] and after we will study the Galerkin approximation with numerical approaches to non linear problems, according to [37]. At the end we will work in a reduced framework using a *POD* approach, coming back to [16].

3.1 The steady Navier-Stokes problem

The equations that we will treat are a semplification of the general Navier-Stokes one in the case we suppose that the solution does not depend on time [41].
We suppose to work with a domain $\Omega \subset \mathbb{R}^d$ where $d = 2, 3$ and Γ is the boundary. The Navier-Stokes equations read as follows:

$$\begin{cases} -\nu\Delta\mathbf{u}(\mathbf{x};\boldsymbol{\mu}) + (\mathbf{u}(\boldsymbol{\mu})\cdot\nabla)\mathbf{u}(\boldsymbol{\mu}) + \nabla p(\mathbf{x};\boldsymbol{\mu}) = \mathbf{f}(\mathbf{x};\boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}), \\ \nabla\cdot\mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = 0 & \text{in } \Omega(\boldsymbol{\mu}), \\ \mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = \mathbf{0} & \text{on } \Gamma_w(\boldsymbol{\mu}), \\ \mathbf{u}(\mathbf{x};\boldsymbol{\mu}) = \mathbf{g}_{in}(\mathbf{x};\boldsymbol{\mu}) & \text{on } \Gamma_{in}(\boldsymbol{\mu}), \\ \nu\dfrac{\partial\mathbf{u}}{\partial\mathbf{n}}(\mathbf{x};\boldsymbol{\mu}) - p(\mathbf{x};\boldsymbol{\mu})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{out}(\boldsymbol{\mu}), \end{cases} \tag{3.1}$$

where $\mathbf{u}$ is the velocity of the fluid, p the pressure, $\mathbf{f}$ a volume force field, ν a kinematic viscosity, $\mathbf{n}$ is the normal unit vector of the boundary, $\boldsymbol{\mu} = (\mu_1, \mu_2, ...) \in \mathbb{P}$ where each μ_i is a physical or

geometrical parameter and $\mathbb{P}$ is the set of all the parameters, $\mathbf{x} = (x, y)$ is the vector of the spacial coordinates.
We have split the boundary in $\Gamma = \Gamma_w \cup \Gamma_{in} \cup \Gamma_{out}$, depending on the boundary conditions imposed. For our experiments we will take $\mathbf{f} = \mathbf{0}$ for an easier treatment.
Hereafter the dependence on $\mathbf{x}$ will be omitted.

We note firstly that by comparing the Navier-Stokes equations with respect to the Stokes ones they also present a convective term $\mathbf{u} \cdot \nabla \mathbf{u}$. This allows to model other physical phenomena that with the Stokes equations are not possible such as the recirculation zone, separation of the flow, as main examples. Mathematically this term introduces a non-linearity in the problem that needs some treatments in the algebraic problem, for example a linearization, as we will see in the following sections.
Now we introduce the *Reynolds number* as

$$Re = L|\overline{\boldsymbol{u}}|/\nu,$$

where L is a characteristic length of the domain while $\overline{\boldsymbol{u}}$ is a characteristic velocity. In our cases we will take $Re \in [0, 100]$ because we will use a standard finite element method for finding a numerical solution and if we increase too much this number we have problem of instability of the numerical solution.
This term appears clearly when we are working with the adimenzionalized Navier-Stokes equation and enshrines the regime of the flow and so its behavior [41].
To begin with the weak formulation of the problem, first of all we introduce a parametric map

$$T : \Omega(\boldsymbol{\mu}) \to \hat{\Omega}$$

with $\hat{\Omega}$ a reference domain independent from $\boldsymbol{\mu}$ and this transformation is a bijection such that $T(\Omega; \boldsymbol{\mu}) = \hat{\Omega}$. As we have explained before we can split our domain Ω in several subdomains Ω^r, create a map T^r for each of these ones and define $T|_{\Omega^r} = T^r$.
After we denote with V and Q respectively the velocity and pressure test space defined over $\hat{\Omega}$ such that:

$$V := \boldsymbol{H}^1_{0,\Gamma_D}(\hat{\Omega}), \quad Q := L^2(\hat{\Omega}),$$

where $\Gamma_D := \Gamma_{in} \cup \Gamma_w$. We can equip V and Q with respectively the H^1-seminorm equivalent to the H^1-norm if $\Gamma_D \neq \emptyset$. For the weak formulation we work in the same way that in the Stokes case and we end up with the following problem: to find the pair $(\mathbf{u}, p) \in V \times Q$ such that:

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

where, using the Einstein convention of the repeated indices,:

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

the classical bilinear forms associated one with the diffusion and the other with pressure term.
We indicate with k a χ two terms associated with the transformation T defined as:

$$k(\mathbf{x};\boldsymbol{\mu}) := \nu(J_T(\mathbf{x};\boldsymbol{\mu}))^{-1}(J_T(\mathbf{x};\boldsymbol{\mu}))^{-T}|J_T(\mathbf{x};\boldsymbol{\mu})|,$$
$$\chi(\mathbf{x};\boldsymbol{\mu}) := (J_T(\mathbf{x};\boldsymbol{\mu}))^{-1}|J_T(\mathbf{x};\boldsymbol{\mu})|,$$

where $J_T(\mathbf{x};\boldsymbol{\mu}) \in \mathbb{R}^{d\times d}$ is the Jacobian matrix associated to the transformation T, and $|J_T(\mathbf{x};\boldsymbol{\mu})|$ is its determinant.
In addition we have a trilinear form:

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

As we can see c is associated with the convective term. We have written it in general but in our problem we will use $c(\mathbf{u},\mathbf{u},\mathbf{v};\boldsymbol{\mu})$.
If we introduce the lift function $\mathbf{u}_g(\boldsymbol{\mu})$ such that $\mathbf{u}_g|_{\Gamma_{in}} = \mathbf{g}_{in}$, $\mathbf{u}_g|_{\Gamma\setminus\Gamma_{in}} = \mathbf{0}$, we have then the decomposition

$$\mathbf{u}_\Gamma(\boldsymbol{\mu}) = \mathbf{u}(\boldsymbol{\mu}) + \mathbf{u}_g(\boldsymbol{\mu}),$$

where $\mathbf{u}_\Gamma$ is the velocity satisfying the boundary conditions while $\mathbf{u}$ is the velocity such that $\mathbf{u}|_{\Gamma_D} = \mathbf{0}$. With this decomposition we have additional terms related to $\mathbf{u}_g(\boldsymbol{\mu})$:

$$d(\mathbf{u},\mathbf{v};\boldsymbol{\mu}) := c(\mathbf{u}_g,\mathbf{u},\mathbf{v};\boldsymbol{\mu}) + c(\mathbf{u},\mathbf{u}_g,\mathbf{v};\boldsymbol{\mu}),$$
$$F(\mathbf{v};\boldsymbol{\mu}) := -a(\mathbf{u}_g,\mathbf{v};\boldsymbol{\mu}) - c(\mathbf{u}_g,\mathbf{u}_g,\mathbf{v};\boldsymbol{\mu}),$$
$$G(q;\boldsymbol{\mu}) := -b(\mathbf{u}_g,q;\boldsymbol{\mu}).$$

We can do some additional hypothesis related to the dependency on $\boldsymbol{\mu}$ of $\mathbf{g}_{in}$ and $\mathbf{u}_g$:

- $\mathbf{g}_{in}(\boldsymbol{\mu}) = \Theta_{in}(\mu)\widetilde{\mathbf{g}}_{in}$.
- $\mathbf{u}_g(\boldsymbol{\mu}) = \Theta_{in}(\mu)\widetilde{\boldsymbol{u}}_g$.

where $\widetilde{\mathbf{g}}_{in}$ and $\widetilde{\boldsymbol{u}}_g$ are independent from $\boldsymbol{\mu}$. These decompositions can be useful in the reduced basis approach for the online phase.

3.2 Finite element discretization and algebraic formulation

We want to introduce a finite element discretization of the problem (3.2) with the classical Galerkin approach according to [16]. As usual we take two finite dimensional spaces V_{N_δ} and Q_{N_δ} of dimensions $N_{\mathbf{u}}$ and N_p respectively, both multiple of a number N_δ. We denote with $\{\boldsymbol{\phi}_i\}_{j=1,\dots,N_{\mathbf{u}}}$ and $\{\xi_j\}_{j=1,\dots,N_p}$ the lagrangian basis associated with them.
The problem becames, given $\boldsymbol{\mu} \in \mathbb{P}$, to find a solution $(\mathbf{u}_{N_\delta}, p_{N_\delta}) \in V_{N_\delta} \times Q_{N_\delta}$ such that:

$$\begin{cases} a(\mathbf{u}_{N_\delta},\mathbf{v}_{N_\delta};\boldsymbol{\mu}) + d(\mathbf{u}_{N_\delta},\mathbf{v}_{N_\delta};\boldsymbol{\mu}) + b(\mathbf{v}_{N_\delta},p_{N_\delta}(\boldsymbol{\mu});\boldsymbol{\mu}) \\ +c(\mathbf{u}_{N_\delta}(\boldsymbol{\mu}),\mathbf{u}_{N_\delta}(\boldsymbol{\mu}),\mathbf{v}_{N_\delta};\boldsymbol{\mu}) = F(\mathbf{v}_{N_\delta};\boldsymbol{\mu}), \quad \forall \mathbf{v}_{N_\delta} \in V_{N_\delta}, \\ b(\mathbf{u}_{N_\delta}(\boldsymbol{\mu}),q_{N_\delta};\boldsymbol{\mu}) = G(q_{N_\delta}), \quad \forall q_{N_\delta} \in Q_{N_\delta}. \end{cases} \tag{3.5}$$

For the well posedness of the problem we need as in the Stokes case that the *inf-sup* condition and the continuity of the bilinear forms must hold.
Now let us pass to the algebraic formulation of the problem.
We decompose our functions with respect to the lagrangian basis in the following way:

$$\mathbf{v}_{N_\delta} = \sum_{i=1}^{N_\mathbf{u}} v_{N_\delta}^i \boldsymbol{\phi}^i \in V_{N_\delta}, \quad q_{N_\delta} = \sum_{i=1}^{N_p} q_{N_\delta}^i \xi_i, \tag{3.6}$$

and we can associate to this function the corresponding vector:

$$\mathbf{v}_{N_\delta} \leftrightarrow \underline{\boldsymbol{v}} = \left(v_{N_\delta}^{(1)}, v_{N_\delta}^{(2)}, ..., v_{N_\delta}^{(N_\mathbf{u})} \right) \in \mathbb{R}^{N_\mathbf{u}}, \tag{3.7}$$

$$q_{N_\delta} \leftrightarrow \underline{\boldsymbol{q}} = \left(q_{N_\delta}^{(1)}, q_{N_\delta}^{(2)}, ..., q_{N_\delta}^{(N_p)} \right) \in \mathbb{R}^{N_p}. \tag{3.8}$$

So we can reformulate (3.5) in the algebraic way:

$$\begin{bmatrix} A(\boldsymbol{\mu}) & C(\underline{\boldsymbol{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{p}}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}, \tag{3.9}$$

where

$$\begin{aligned}
&\underline{\boldsymbol{u}} := \left(u_{N_\delta}^{(1)}, u_{N_\delta}^{(2)}, ..., u_{N_\delta}^{(N_\mathbf{u})} \right), \\
&\underline{\boldsymbol{p}} := \left(p_{N_\delta}^{(1)}, p_{N_\delta}^{(2)}, ..., p_{N_\delta}^{(N_p)} \right), \\
&(A(\boldsymbol{\mu}))_{ij} := a(\phi_j, \phi_i; \boldsymbol{\mu}) + d(\phi_j, \phi_i; \boldsymbol{\mu}), \\
&(B(\boldsymbol{\mu}))_{ki} := b(\phi_i, \xi_k; \boldsymbol{\mu}), \\
&(C(\underline{\boldsymbol{u}}; \boldsymbol{\mu}))_{ij} := \sum_{n=1}^{N_\mathbf{u}} u_{N_\delta}^{(n)} c(\phi_n, \phi_j, \phi_i; \boldsymbol{\mu}), \\
&\left(\underline{\boldsymbol{g}}(\boldsymbol{\mu})\right)_k := -b(\mathbf{u}_{g,N_\delta}, \xi_k; \boldsymbol{\mu}), \\
&(\underline{\boldsymbol{f}}(\boldsymbol{\mu}))_k := -a(\mathbf{u}_{g,N_\delta}, \phi_k; \boldsymbol{\mu}) - c(\mathbf{u}_{g,N_\delta}, \mathbf{u}_{g,N_\delta}, \phi_k; \boldsymbol{\mu}),
\end{aligned}$$

where $\mathbf{u}_{g,N_\delta}$ is a lifting function approximation of the real one, using a finite element discretization. For solving (3.21) that is a non-linear system in the velocity we need to linearize it and for doing it we will use the Picard/Oseen or the Newton iteration [40].

- *Picard/Oseen iteration*: in this type of approximation the velocity in the previous step is substituted into the convective term. So in the strong form we have that:

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \mathbf{u}^k \cdot \nabla \mathbf{u}^{k+1}.$$

We do not do the approximation

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \mathbf{u}^{k+1} \nabla \mathbf{u}^{k},$$

because in this way we would solve a Stokes problem that does not take into account the non-linearity and so the gain of using the Navier-Stokes equations that involve non-linear terms. With this iteration we need to start with an initial guess on the velocity $u^{(0)}$ but none on the pressure.
The Picard iteration costructs a sequence of solutions $(\mathbf{u}^{k+1}, p^{k+1})$ that solves:

$$\begin{cases} -\nu \Delta \mathbf{u}^{k+1} + (\mathbf{u}^{k} \cdot \nabla)\mathbf{u}^{k+1} + \nabla p^{k+1} = \mathbf{f}, \\ \nabla \cdot \mathbf{u}^{k+1} = 0. \end{cases} \tag{3.10}$$

In the weak formulation, we are solving:

$$\begin{cases} a(\mathbf{u}^{k+1}, \mathbf{v}; \boldsymbol{\mu}) + d(\mathbf{u}^{k+1}, \mathbf{v}; \boldsymbol{\mu}) + b(\mathbf{u}^{k+1}, p^{k+1}) + c(\mathbf{u}^{k}, \mathbf{u}^{k+1}, \mathbf{v}) = F(\mathbf{v}), \quad \forall \mathbf{v} \in V, \\ b(\mathbf{u}^{k+1}, q) = 0, \quad \forall q \in Q. \end{cases} \tag{3.11}$$

We continue the iteration until we do not reach the convergence that we have when:

$$||\nabla(\mathbf{u}^{k+1} - \mathbf{u}^{k})||_V \leq tol. \tag{3.12}$$

In the algebraic sense this approximation is translated into solving iteratively this system:

$$\begin{bmatrix} A(\boldsymbol{\mu}) & C(\underline{\boldsymbol{u}}^{\boldsymbol{k}}(\boldsymbol{\mu}); \boldsymbol{\mu}) B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}^{\boldsymbol{k+1}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{p}}^{\boldsymbol{k+1}}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}. \tag{3.13}$$

We note that this iteration has a linear convergence [40], i.e.:

$$||\nabla(\mathbf{u} - \mathbf{u}^{k+1})||_V \leq \varrho ||\nabla(\mathbf{u} - \mathbf{u}^{k})||_V \leq \varrho^{k} ||\nabla(\mathbf{u} - \mathbf{u}^{0})||_V, \tag{3.14}$$

where ϱ is a positive constant.
The stationary solution is approximated by the one we obtain at convergence of the iterations.

- *Newton iteration*: this method is usually faster than the previous one when we are near the solution. The idea behind is that the solution at the $k+1$-step does not differ too much from the one at the k one, i.e.:

$$\mathbf{u}^{k+1} = \mathbf{u}^{k} + \delta \mathbf{u}^{k},$$

with $\delta \mathbf{u}^{k}$ small.
So the convective term becames:

$$\begin{aligned} &\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} = (\mathbf{u}^{k} + \delta \mathbf{u}^{k}) \cdot \nabla (\mathbf{u}^{k} + \delta \mathbf{u}^{k}) = \\ &\mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k+1} + (\mathbf{u}^{k+1} - \mathbf{u}^{k}) \cdot \nabla (\mathbf{u}^{k} + \delta \mathbf{u}^{k}) = \\ &\mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k+1} + \mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k} - \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k} + \delta \mathbf{u}^{k} \cdot \nabla \delta \mathbf{u}^{k}. \end{aligned}$$

We neglet the quadratic term in $\delta\mathbf{u}$ because is supposed to be too small with respect to the other terms.
So the strong equation becames:

$$\begin{cases} -\nu\Delta\mathbf{u}^{k+1} + \mathbf{u}^{k+1}\cdot\nabla\mathbf{u}^{k} + \mathbf{u}^{k}\cdot\nabla\mathbf{u}^{k+1} + \nabla p^{k+1} = \mathbf{f} + \mathbf{u}^{k}\cdot\nabla\mathbf{u}^{k}, \\ \nabla\cdot\mathbf{u}^{k+1} = 0. \end{cases} \tag{3.15}$$

In this case we can use a solution from the Stokes problem as initial guess but the method does not converge if we are working with high Reynolds numbers and the initial guess is not good enough. In this case we use the Stokes problem as initial guess for a problem with low Reynolds number and after we pass this last one as initial guess for the high Reynolds number problem.
Passing to the weak equation:

$$\begin{cases} a(\mathbf{u}^{k+1},\mathbf{v};\boldsymbol{\mu}) + c(\mathbf{u}^{k},\mathbf{u}^{k+1},\mathbf{v};\boldsymbol{\mu}) + c(\mathbf{u}^{k+1},\mathbf{u}^{k},\mathbf{v};\boldsymbol{\mu}) + \\ b(\mathbf{v},p^{k+1};\boldsymbol{\mu}) + d(\mathbf{u}^{k+1},\mathbf{v};\boldsymbol{\mu}) = F(\mathbf{v}) + c(\mathbf{u}^{k},\mathbf{u}^{k},\mathbf{v};\boldsymbol{\mu}), \quad \forall\mathbf{v}\in V_{N_\delta}, \\ b(\mathbf{u}^{k+1},q;\boldsymbol{\mu}) = 0, \quad \forall q\in Q. \end{cases} \tag{3.16}$$

For what it concernes the algebraic problem we have to introduce this other form:

$$N(\mathbf{u};\boldsymbol{\mu})_{nm} := \sum_{j=1}^{N_{\mathbf{u}}} u_{N_\delta}^{(j)} c(\phi_n,\phi_j,\phi_m;\boldsymbol{\mu}), \tag{3.17}$$

and we obtain the system:

$$\begin{bmatrix} A(\boldsymbol{\mu}) & (C(\underline{\boldsymbol{u}}^{\boldsymbol{k}}(\boldsymbol{\mu});\boldsymbol{\mu}) + N(\underline{\boldsymbol{u}}^{\boldsymbol{k}}(\boldsymbol{\mu});\boldsymbol{\mu}))B^T(\boldsymbol{\mu}) \\ B(\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}^{\boldsymbol{k+1}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{p}}^{\boldsymbol{k+1}}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{\mu}) \\ \underline{\boldsymbol{g}}(\boldsymbol{\mu}) \end{bmatrix}. \tag{3.18}$$

For this iteration, from [40], we have a quadratic convergence:

$$||\nabla(\mathbf{u}-\mathbf{u}^{k})||_V \leq ||\nabla(\mathbf{u}^{k-1}-\mathbf{u})||_V^2. \tag{3.19}$$

In our results we will use only the *Newton* approach.

3.3 Reduced model for Navier-Stokes equations

In this section we discuss the reduced formulation of the Navier-Stokes problem.
Firstly we need an affine decomposition hypothesis on the severals terms involved:

$$
\begin{aligned}
a(\mathbf{u},\mathbf{v};\boldsymbol{\mu}) &= \sum_{q=1}^{Q^a} \Theta^q(\boldsymbol{\mu}) a^q(\mathbf{u},\mathbf{v}),\\
b(p,\mathbf{w};\boldsymbol{\mu}) &= \sum_{s=1}^{Q^b} \Phi^s(\boldsymbol{\mu}) b^s(p,\mathbf{w}),\\
C(\mathbf{u},\mathbf{v},\mathbf{z};\boldsymbol{\mu}) &= \sum_{q=1}^{Q_c} \Theta_q^C(\boldsymbol{\mu}) C^q(\mathbf{u},\mathbf{v},\mathbf{z}),\\
F(\mathbf{v})(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_f} \Theta_q(\boldsymbol{\mu}) F^q(\mathbf{v}).
\end{aligned}
$$

In general if this hypothesis does not hold we can use linearization tecniques such as *EIM* or *DEIM* [22].
As method for reducing the problem we use a *POD* approach as in the Stokes case of (2.21). There are no differences in the Navier-Stokes case so we will not spend time on it but let us see the reduced formulation of the problem following [19]. Using the *POD* approach we obtain two reduced spaces:

$$
\begin{aligned}
&Q_N := \text{span}\{\psi_n := p_{N_\delta}(\boldsymbol{\mu}^n), n = 1, ..., N\},\\
&V_N := \text{span}\{\zeta_n := \mathbf{u}_{N_\delta}(\boldsymbol{\mu}^n), T^{\boldsymbol{\mu}}\psi_n, n = 1, ..., N\} =\\
&\text{span}\{\boldsymbol{\sigma}_n, \quad n = 1, ..., 2N| \quad \boldsymbol{\sigma}_i = \mathbf{u}_{N_\delta}(\boldsymbol{\mu}^i), \text{ for } i = 1, ..., N, \quad \boldsymbol{\sigma}_i = T^{\boldsymbol{\mu}}\psi_i, \text{ for } i = N+1, ...2N\},
\end{aligned}
$$

where $T^{\boldsymbol{\mu}} : Q_{N_\delta} \to V_{N_\delta}$ is the *supremizer operator* defined as in (2.5). As we have said, the enrichment of the reduced velocity space with the supremizer is necessary for satisfying the *inf-sup condition*. So the reduced problem is to find a pair $(\mathbf{u}_N, p_N) \in V_N \times Q_N$:

$$
\begin{cases}
a(\mathbf{u}_N,\mathbf{v}_N;\boldsymbol{\mu}) + d(\mathbf{u}_N,\mathbf{v}_N;\boldsymbol{\mu}) + b(\mathbf{v}_N, p_N(\boldsymbol{\mu});\boldsymbol{\mu})\\
+c(\mathbf{u}_N(\boldsymbol{\mu}),\mathbf{u}_N(\boldsymbol{\mu}),\mathbf{v}_N;\boldsymbol{\mu}) = F(\mathbf{v}_N;\boldsymbol{\mu}), \quad \forall \mathbf{v}_N \in V_N,\\
b(\mathbf{u}_N(\boldsymbol{\mu}), q_N;\boldsymbol{\mu}) = G(q_N), \quad \forall q_N \in Q_N,
\end{cases}
\tag{3.20}
$$

that we can write in an matrix form as:

$$
\begin{bmatrix} A_N(\boldsymbol{\mu}) & C_N(\underline{\boldsymbol{u}}_N(\boldsymbol{\mu});\boldsymbol{\mu})B_N^T(\boldsymbol{\mu}) \\ B_N(\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}(\boldsymbol{\mu})_N \\ \underline{\boldsymbol{p}}(\boldsymbol{\mu})_N \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}(\boldsymbol{\mu})_N \\ \underline{\boldsymbol{g}}(\boldsymbol{\mu})_N \end{bmatrix}. \tag{3.21}
$$

where

$$
\begin{aligned}
&(A(\boldsymbol{\mu}))_{ij} := a(\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i; \boldsymbol{\mu}) + d(\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i; \boldsymbol{\mu}),\\
&(B(\boldsymbol{\mu}))_{ki} := b(\boldsymbol{\sigma}_i, \psi_k; \boldsymbol{\mu}),\\
&(C(\underline{\boldsymbol{u}}_N; \boldsymbol{\mu}))_{ij} := \sum_{n=1}^{2N} u_N^{(n)} c(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i; \boldsymbol{\mu}),\\
&\left(\underline{\boldsymbol{g}}_N(\boldsymbol{\mu})\right)_k := -b(\mathbf{u}_{g,N_\delta}, \psi_k; \boldsymbol{\mu}),\\
&(\underline{\boldsymbol{f}}(\boldsymbol{\mu})_N)_k := -a(\mathbf{u}_{g,N_\delta}, \boldsymbol{\sigma}_k; \boldsymbol{\mu}) - c(\mathbf{u}_{g,N_\delta}, \mathbf{u}_{g,N_\delta}, \boldsymbol{\sigma}_k; \boldsymbol{\mu}).
\end{aligned}
$$

As in the finite element approximation this is a non-linear problem that we can solve using an iterative method such as Picard/Oseen or Netwon.

3.4 Conclusions

In this chapter we have presented the parametric steady Navier-Stokes equations in the strong and weak form, stressing the importance of the non-linear convective term. We have subsequently introduced a finite element discretization with the associated non-linear algebraic system. To deal with this non-linearity we have explained two tipes of approximation, the Picard/Osseen iteration and the Newton one, seeing the different rates of convergence for each one.
In the last part we have exposed the reduced method for the Navier-Stokes problem, similar to the Stokes one.

Chapter 4

Weighted reduced order methods

When we are working with physical and in particular fluid dynamical systems, uncertainties come naturally due to the lack of information about some quantities or due to measure errors that introduce some stochasticity in the model. In these cases sometimes we can only have a probabilistic information such as a probability density function associated to the occurence of a certain random event. So it can be interesting to obtain some statistics and for doing this we need to do a lot of simulations. Since this can be very expensive, the idea is to use the reduced method approach. In this case we will see that stochasticity is treated as a set of parameters with a probability distribution. The distribution can be used for assigning an "importance" to the parameters and can result in a reduction of the computational cost as we will see in the numerical experiments. This idea will be realized with the *weighted approach*.
For what concerns the organization of the chapter, we will first introduce the stochastic formulation of the Stokes problem and after we will introduce the *weighted greedy* and *weighted POD* algorithms, following the works of [25], [26], [27], [28].
We cite some articles that exposes a comparison of the weighted approach with the stochastic collocation method in [29] and [30].
We finally note that we will treat only the Stokes problem because the Navier-Stokes is analogous.

4.1 Problem setting

We first begin with the formulation of the stochastic Stokes problem [26]. First of all let us introduce a triple $(\Omega, \mathcal{F}, P)$ that denotes a complete probability space where Ω is the set of the outcomes $\omega \in \Omega$ [1], $\mathcal{F}$ is a σ-algebra of events and $P : \mathcal{F} \to [0, 1]$ is a probability measure, i.e. $P(\Omega) = 1$. In such a framework we introduce a real-valued continuous *random variable* $Y : (\Omega, \mathcal{F}) \to (\mathbb{R}, \mathcal{B})$ being $\mathcal{B}$ the Borel σ-algebra on $\mathbb{R}$. We denote with Γ the image of Y and with $\rho : \Gamma \to \mathbb{R}$ the probability density

[1] We note that in the previous chapters Ω was the physical domain but now it is the probability space.

function. We can also define the k-th moment of Y as:

$$\mathbb{E}[Y^k] := \int_\Omega Y^k dP = \int_\Gamma y^k \rho(y) dy.$$

Now let us take a open bounded domain $D \subset \mathbb{R}^d$ $(d = 2, 3)$ with Lipschitz boundary. We can define a *random field* $v : D \times \Omega \to \mathbb{R}$ as a random variable fixed $x \in D$. We can also define the Hilbert space

$$\mathcal{H}^s(D) := L^2(\Omega) \otimes H^s(D), \tag{4.1}$$

with $s \in \mathbb{R}$, equipped with the norm:

$$||v||_{\mathcal{H}^s(D)} := \left(\int_\Omega ||v(\cdot, \omega)||^2_{H^s(D)} dP \right)^{1/2}. \tag{4.2}$$

Thus we can understand that $\mathcal{H}^s(D)$ is the space of the functions such that they belong to $H^s(D)$ when we fix ω and such that the $\mathcal{H}^s-$norm, with respect to the space variable, of the vector stays in the $L^2(\Omega)$ space. We observe that $\mathcal{H}^0(D) = L^2(D) \otimes L^2(\Omega) := \mathcal{L}^2(D)$. We can introduce the inner product that induces the previous norm as:

$$(w, v)_{\mathcal{L}^2} := \int_\Omega \int_D wv \; dx dP, \quad \forall w, v \in \mathcal{L}^2(D). \tag{4.3}$$

We define a *random vector* $\mathbf{v} = (v_1, v_2, ..., v_d) : D \times \Omega \to \mathbb{R}$ as the vector whose components are random fields. This one belongs to the space

$$\mathcal{H}^{s,d}(D) := (L^2(\Omega) \otimes H^s(D))^d,$$

and we have that $\mathcal{H}^{0,d}(D) = \mathcal{L}^{2,d}(D)$. This space has the norm:

$$||\mathbf{v}||_{\mathcal{H}^{s,d}(D)} := \sum_{i=1}^d ||v_i||_{\mathcal{H}^s(D)}, \tag{4.4}$$

and the inner product:

$$(\mathbf{v}, \mathbf{w})_{\mathcal{H}^{s,d}(D)} := \sum_{i=1}^d (v_i, w_i)_{\mathcal{H}^s}. \tag{4.5}$$

Now we are ready to introduce the stochastic Stokes problem in the strong formulation as in [26]. We take a random variable $\nu : \Omega \to \mathbb{R}_+$, a random field $\mathbf{f} : D \times \Omega \to \mathbb{R}^d$ and $\mathbf{h} : \partial D_N \times \Omega \to \mathbb{R}$. As usual we search a solution $(\mathbf{u}, p) : D \times \Omega \to \mathbb{R}^d \times \mathbb{R}$ such that:

$$\begin{cases} -\nu(\omega)\Delta \mathbf{u}(\mathbf{x}, \omega) + \nabla p(\mathbf{x}, \omega) = \mathbf{f}(\mathbf{x}, \omega) & \text{in } D(\omega), \\ \nabla \cdot \mathbf{u}(\mathbf{x}, \omega) = 0 & \text{in } D(\omega), \\ \mathbf{u}(\mathbf{x}, \omega) = \mathbf{0} & \text{on } \partial D_{D,\mathbf{0}}(\omega), \\ \mathbf{u}(\mathbf{x}, \omega) = \mathbf{g}_{in}(\mathbf{x}, \omega) & \text{on } \partial D_{in}(\omega), \\ \nu \dfrac{\partial \mathbf{u}}{\partial \mathbf{n}}(\mathbf{x}, \omega) - p(\mathbf{x}, \omega)\mathbf{n} = \mathbf{h}(\mathbf{x}, \omega) & \text{on } \partial D_N(\omega), \end{cases} \tag{4.6}$$

with $\partial D_{D,\mathbf{0}} \cup \partial D_{in} \cup \partial D_N = \partial D$. We denote with $\partial D_D := D_{D,\mathbf{0}} \cup \partial D_{in}$.
Now we want to pass to the weak formulation, so we introduce the test space for the velocity:

$$\mathcal{V} := \{\mathbf{v} \in \mathcal{H}^{1,d}(D) : \mathbf{v} = \mathbf{0} \text{ on } \partial D_D\}, \tag{4.7}$$

and the test space for the pressure:

$$\mathcal{Q} := L^2(\Omega) \otimes L^2_0(D). \tag{4.8}$$

With these spaces we can introduce the weak formulation. We are searching the pair $(\mathbf{u}, p) \in \mathcal{V} \times \mathcal{Q}$ such that:

$$\begin{cases} \mathbb{A}(\mathbf{u}, \mathbf{v}) + \mathbb{B}(\mathbf{v}, p) = \mathbb{F}(\mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{V}, \\ \mathbb{B}(\mathbf{u}, q) = \mathbb{G}(q), \quad \forall q \in \mathcal{Q}, \end{cases} \tag{4.9}$$

with the bilinear form $\mathbb{A} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ defined as:

$$\mathbb{A}(\mathbf{u}, \mathbf{v}) := \int_\Omega \int_D \nu(\omega) \nabla \mathbf{u}(\omega) : \nabla \mathbf{v}(\omega) \; d\mathbf{x} dP = \sum_{i,j=1}^{d} \int_\Omega \int_D \nu(\omega) \frac{\partial u_i(\omega)}{\partial x_j} \frac{\partial v_i(\omega)}{\partial x_j} \; d\mathbf{x} dP, \tag{4.10}$$

the bilinear form $\mathbb{B} : \mathcal{V} \times \mathcal{Q} \to \mathbb{R}$ defined as:

$$\mathbb{B}(\mathbf{v}, q) := -\int_\Omega \int_D \nabla \cdot \mathbf{v}(\omega) q(\omega) \; d\mathbf{x} dP = -\sum_{i=1}^{d} \int_\Omega \int_D \frac{\partial v_i(\omega)}{\partial x_i} q(\omega) \; d\mathbf{x} dP, \tag{4.11}$$

the linear form $\mathbb{F} : \mathcal{V} \to \mathbb{R}$ defined as:

$$\mathbb{F}(\mathbf{v}) := (\mathbf{f}(\omega), \mathbf{v})_{\mathcal{H}^{s,d}(D)} + (\mathbf{h}(\omega), \mathbf{v})_{\partial D_N} - a(\mathbf{u}_{g_i n}(\omega), \mathbf{v}), \tag{4.12}$$

and finally the linear form $\mathbb{G} : \mathcal{Q} \to \mathbb{R}$ defined as:

$$\mathbb{G}(q) := -\mathbb{B}(\mathbf{u}_{g_i n}(\omega), q(\omega)), \tag{4.13}$$

where $\mathbf{u}_{g_i n}$ is the lifting function.
Now to have the well-posedness of the Stokes problem we need some hypothesis on ν, $\mathbf{f}$, $\mathbf{h}$, $\mathbb{A}$ and $\mathbb{B}$.

1. The random viscosity ν is uniformly bounded from below and from above almost everywhere, i.e. there exist two constants $0 < \nu_{min} \leq \nu_{max} < \infty$ such that

$$P(\omega : \nu_{min} \leq \nu(\omega) \leq \nu_{max}) = 1. \tag{4.14}$$

2. The random force $\mathbf{f}$ and $\mathbf{h}$ satisfy:

$$||\mathbf{f}||_{\mathcal{L}^{2,d}(\Omega)} < \infty, \tag{4.15}$$

$$||\mathbf{h}||_{\mathcal{H}} < \infty. \tag{4.16}$$

3. $\mathbb{A}$ is continuous and coercive, i.e. it holds:

$$\exists \gamma_a > 0 \text{ such that } \mathbb{A}(\mathbf{u}, \mathbf{v}) \leq \gamma_a ||\mathbf{u}||_{\mathcal{V}} ||\mathbf{v}||_{\mathcal{V}}, \quad \forall \mathbf{u}, \mathbf{v} \in \mathcal{V}, \tag{4.17}$$

for the continuity and:

$$\exists \alpha_a > 0 \text{ such that } \mathbb{A}(\mathbf{u}, \mathbf{u}) \geq \alpha_a ||\mathbf{u}||_{\mathcal{V}}^2, \quad \forall \mathbf{u} \in \mathcal{V}_0, \tag{4.18}$$

for the coercivity, where $\mathcal{V}_0 := \{\mathbf{v} \in \mathcal{V} : \mathbb{B}(\mathbf{v}, q) = 0, \forall q \in \mathcal{Q}\}$ is the kernel of $\mathbb{B}$.

4. $\mathbb{B}$ is continuous and the *inf-sup condition* holds:

$$\exists \gamma_b > 0 \text{ such that } \mathbb{B}(\mathbf{v}, q) \leq \gamma_b ||\mathbf{v}||_{\mathcal{V}} ||q||_{\mathcal{Q}}, \quad \forall \mathbf{v} \in \mathcal{V}, \forall q \in \mathcal{Q}, \tag{4.19}$$

for the continuity, while

$$\exists \beta_0 > 0 \text{ such that } \inf_{q \in \mathcal{Q}} \sup_{\mathbf{v} \in \mathcal{V}} \frac{\mathbb{B}(\mathbf{v}, q)}{||\mathbf{v}||_{\mathcal{V}} ||q||_{\mathcal{Q}}} \geq \beta_0. \tag{4.20}$$

Now with these hypothesis we have the following theorem that we will not prove because the proof is similar to the deterministic case that can be seen in [7]:

Theorem 5. *Under the previous hypothesis we have done on ν, $\boldsymbol{f}$, $\boldsymbol{h}$, $\mathbb{A}$ and $\mathbb{B}$ there exists a unique solution to the stochastic Stokes problem in* (4.9) *and moreover we have these inequalities on the solution $(\boldsymbol{u}, p)$:*

$$\begin{aligned}
||\boldsymbol{u}||_{\mathcal{V}} &\leq \frac{1}{\alpha_a}(C_p ||\boldsymbol{f}||_{\mathcal{L}^{2,d}} + \frac{\alpha_a + \gamma_a}{\beta_b} ||\boldsymbol{h}||_{\mathcal{H}}), \\
||p||_{\mathcal{Q}} &\leq \frac{1}{\beta_b}\left(\left(1 + \frac{\gamma_a}{\alpha_a}\right) C_p ||\boldsymbol{f}||_{\mathcal{L}^{2,d}} + \frac{\gamma_a(\alpha_a + \gamma_a)}{\alpha_a \beta_b} C_T ||\boldsymbol{h}||_{\mathcal{H}}\right),
\end{aligned}$$

where C_T and C_P are the constant from the trace theorem and the Poincaré constant [7].

Let us do an important assumption on the stochastic dependence of the quantities involved in the equations that will be useful for applying the reduced order models.
We suppose that $\nu, \mathbf{f}$ and $\mathbf{h}$ depend on a finite number N of random variables that we collect in a random vector $Y(\omega) = (Y_1(\omega), Y_2(\omega), ..., Y_N(\omega)) : \Omega \to \Gamma = \Gamma_1 \times \Gamma_2 \cdots \times \Gamma_N \subset \mathbb{R}^N$ with a probability density function $\rho = (\rho_1, ..., \rho_N) : \Gamma \to \mathbb{R}^N$. In other words we are doing an hypothesys on how the stochastic dependency is expressed:

$$\begin{aligned}
\nu(\omega) &= \nu(Y(\omega)), \\
\mathbf{f}(\cdot, \omega) &= \mathbf{f}(\cdot, Y(\omega)), \\
\mathbf{h}(\cdot, \omega) &= \mathbf{h}(\cdot, Y(\omega)).
\end{aligned}$$

In general each quantity will depend on different random vectors Y_ν, Y_f, Y_h but for ease of notation we will compact them in a single vector $Y = (Y_\nu, Y_f, Y_h)$ with dimension N.
Thanks to the Doob-Dynkin lemma [33] we also have that:

$$\mathbf{u}(\cdot, \omega) = \mathbf{u}(\cdot, Y(\omega)),$$
$$p(\cdot, \omega) = p(\cdot, Y(\omega)).$$

Now we want to lead us back to the parametric partial differential equations and to see the random vector Y as a parameter, following the same idea of [27] for the elliptic case.
For doing this we need an other important assumption: we suppose that Γ_k is a compact set, for each k. If this last hypothesis is not true we can obtain it truncating the probability distribution on a compact set where we have the higher probability.
The next step is to define again the linear and bilinear forms, changing the test spaces from $\mathcal{V}$ and $\mathcal{Q}$ to V and Q defined as in the deterministic case (1.2) and (1.3):

$a : V \times V \to \mathbb{R}$ such that:

$$a(\mathbf{u}, \mathbf{v}; y) := \int_D \nu(y) \nabla \mathbf{u}(y) : \nabla \mathbf{v} \; d\mathbf{x} = \sum_{i,j=1}^{d} \int_D \nu(y) \frac{\partial u_i(y)}{\partial x_j} \frac{\partial v_i}{\partial x_j} \; d\mathbf{x},$$

$b : V \times Q \to \mathbb{R}$ such that:

$$b(\mathbf{v}, p; y) := -\int_D (\nabla \cdot \mathbf{v}) p(y) \; d\mathbf{x} dP = -\sum_{i=1}^{d} \int_D \frac{\partial v_i}{\partial x_i} p(y) \; d\mathbf{x},$$

$F : V \to \mathbb{R}$ such that:
$$F(\mathbf{v}; y) := (\mathbf{f}(y), \mathbf{v})_{\mathcal{H}^{s,d}(D)} + (\mathbf{h}(y), \mathbf{v})_{\partial D_N} - a(\mathbf{u}_{g_i n}(y), \mathbf{v}),$$

$G : Q \to \mathbb{R}$ such that:
$$G(q; y) := -b(\mathbf{u}_{g_i n}(y), q).$$

So we can reformulate the (4.9). We want to find a solution $(\mathbf{u}, p) : \Gamma \to V \times Q$ such that:

$$\begin{cases} a(\mathbf{u}, \mathbf{v}; y) + b(\mathbf{v}, p; y) = \langle F, \mathbf{v} \rangle(y), & \forall \mathbf{v} \in V, \\ b(\mathbf{u}, q; y) = \langle G, q \rangle(y), & \forall q \in Q. \end{cases} \tag{4.21}$$

for a.e. $y \in \Gamma$ distributed according to $\rho(y)$. In this way we have recast our stochastic problem to a parametric one. But at this level if we do not do something else we do not use the probability density function. For this reason we will introduce the weighted approach in the next section.

Before going on we want to discretize our problem (4.21) with a Galerkin approximation. As usual we introduce two finite dimensional spaces V_{N_δ} and Q_{N_δ} with a dimensional proportional to N_δ. So we search a pair $(\mathbf{u}_{N_\delta}, p_{N_\delta}) \in V_{N_\delta} \times Q_{N_\delta}$ such that:

$$\begin{cases} a(\mathbf{u}_{N_\delta}, \mathbf{v}_{N_\delta}; y) + b(\mathbf{v}_{N_\delta}, p_{N_\delta}; y) = \langle F, \mathbf{v}_{N_\delta} \rangle(y), \quad \forall \mathbf{v}_{N_\delta} \in V_{N_\delta}, \\ b(\mathbf{u}_{N_\delta}, q_{N_\delta}; y) = \langle G, q_{N_\delta} \rangle(y), \quad \forall q_{N_\delta} \in Q_{N_\delta}, \end{cases} \tag{4.22}$$

for a.e. $y \in \Gamma$. As usual we will refer to this problem as the truth one.
We note that in the following sections we will use the probabilistic information to search the best snapshots in order to reduce the computation cost.

4.2 Weighted algorithms

Several weighted reduced order methods have been invented, see for example [32] where the value at risk is used, but in this thesis we will follow [26], [27] and [28].
Since we want to work in a reduced framework, as usual we have to find two reduced spaces V_N and Q_N with dimensions that are multiples of $N \ll N_\delta$ and generated by a finite linear combination of solutions of the truth problem (4.22). For doing that we have to chose a discrete parameters space $\mathbb{P}_h$ and searching in it some parameters y for computing the truth solutions.
The main novelty in the weighted approach is that we assign different weights to the parameters according to a weight function $w(y)$, chosen following some rules that we will see. This function will have the role of chosing the parameters space and the basis.
What we will find in the numerical experiments is that when we are working with parameters derived from a distribution far from the uniform one (which we were implicitely supposing in the deterministic case), the weighted approach can be useful for lowering the computational cost. For example this method can take the most likely parameters and so with few basis we can obtain good results for the more likely parameters.
We note that the reduced formulation is the same as the one in (2.9) so we will not repeat it. What changes it is the method with which we select the parameters for the solutions and the solutions chosen.
As usual we have to do an affine decomposition assumption of the different terms in the equations. So we suppose they are a linear combination of the components of the random vector $Y(\omega) =$

$(Y_1(\omega), Y_2(\omega), ..., Y_N(\omega)) : \Omega \to \mathbb{R}^k$, i.e.:

$$a(\mathbf{u}, \mathbf{v}; Y(\omega)) = a_0(\mathbf{u}, \mathbf{v}) + \sum_{k=1}^{K} a_k(\mathbf{u}, \mathbf{v}) Y_k(\omega),$$
$$b(\mathbf{v}, q; Y(\omega)) = b_0(\mathbf{v}, q) + \sum_{k=1}^{K} b_k(\mathbf{v}, q) Y_k(\omega),$$
$$F(\mathbf{v}; Y(\omega)) = F_0(\mathbf{v}) + \sum_{k=1}^{K} F_k(\mathbf{v}) Y_k(\omega),$$
$$G(q; Y(\omega)) = G_0(q) + \sum_{k=1}^{K} G_k(q) Y_k(\omega),$$

with the bilinear forms a_k, b_k and the linear forms F_k, G_k defined in the different spaces:

$$a_k : V \times V \to \mathbb{R} \quad \forall k = 0, ..., K,$$
$$b_k : V \times Q \to \mathbb{R} \quad \forall k = 0, ..., K,$$
$$F_k : V \to \mathbb{R} \quad \forall k = 0, ..., K,$$
$$G_k : Q \to \mathbb{R} \quad \forall k = 0, ..., K.$$

4.2.1 Weighted Proper Orthogonal Decomposition

In the *weighted POD approach* we would like to find the N-dimensional subspace V_N such that it minimizes the error:

$$\int_{\Gamma} ||\mathbf{u}_{N_\delta}(y) - \mathbf{u}_N(y)||_V^2 \rho(y)\, dy. \tag{4.23}$$

We have to discretize this integral with a finite sum, so the goal is to minimize:

$$\sum_{y \in \mathbb{P}_h} w(y) ||\mathbf{u}_{N_\delta}(y) - \mathbf{u}_N(y)||_V^2 = \sum_{i=1}^{M} w_i ||\mathbf{u}_{N_\delta}(y_i) - P_N(\mathbf{u}_{N_\delta}(y_i))||_V^2, \tag{4.24}$$

where $\mathbb{P}_h$ is a finite discretization of Γ (that in the deterministic case was $\mathbb{P}$) of cardinality equal to M.
The weights w_i have to be chosen according to some rule such as the Monte-Carlo method, the tensor product rule or the Smolyak rule that we will see in next chapter.
For minimizing the quantity (4.23) we can follow a similar strategy to that in the deterministic case. We introduce the operator $\hat{C}$ defined as:

$$\hat{C}(\mathbf{v}_{N_\delta}) := \sum_{m=1}^{M} w_m (\mathbf{v}_{N_\delta}, \psi_m)_V \psi_m,$$

where $\psi_m := \mathbf{u}_{N_\delta}(y_m)$. So we search the eigenfunctions and eigenvalues of $\hat{C}$. If we do the same math as in the deterministic case we arrive to the algebraic formulation in which we have to find the eigenvector of $\hat{C} := P \cdot C$ where C is the same matrix defined in (2.20) while $P := \text{diag}(w_1, ..., w_M)$. In this case P is a preconditioner matrix.
We note that $\hat{C}$ is not symmetric in the usual sense but it is with with respect to the scalar product induced by the matrix C. Infact if we have the scalar product induced defined as:

$$\langle x, y \rangle_C := x^T C y,$$

$\hat{C}$ is symmetric with respect to this scalar product if and only if:

$$\langle \hat{C}x, y \rangle_C = \langle x, \hat{C}y \rangle_C, \tag{4.25}$$

that is equivalent to, using the definition:

$$x^T \hat{C}^T C y = x^T C \hat{C} y,$$

and so $\hat{C}$ is symmetric if:

$$\hat{C}^T C = C\hat{C}.$$

So if we do the simple computations:

$$\hat{C}^T C = (PC)^T C = C^T P^T C = CPC = C\hat{C},$$

for the symmetry of C and P.
So with this scalar product we can use the spectral theorem and we have an orthogonal basis of eigenvectors.
Now the problem is how to chose the set $\mathbb{P}_h$.
The easiest technique is the Monte-Carlo one that we will explain in more details in the next section. We only say that is this case we take M realizations of the random vector Y and put the weight $w(y) = \dfrac{1}{M}$ in the (4.24).
Another approach is to select $\mathbb{P}_h$ and w according to some quadrature rule for approximating (4.23). So if we take a quadrature rule $\mathcal{Q}_\rho$, considering the function ρ in the integral and an integrable function $f : \Gamma \to \mathbb{R}$ we have:

$$\mathcal{Q}_\rho(f) := \sum_{i=1}^{M} w_i f(x_i), \tag{4.26}$$

that approximates the general integral:

$$\int_\Omega f(y)\rho(y)\ dy. \tag{4.27}$$

If we change $\mathcal{Q}_\rho$ obviously we change the nodes, the weights and so the preconditioner P.
Finally we say that in the deterministic case we could not take M too big. However in the stochastic case we have the alternative of using a sparse grid quadrature rule (chapter 6).
Let us see now the *weighted POD algorithm*:

1) choose the training set $\mathbb{P}_h \subset \Gamma$ and the weights w_i according to some quadrature rule.
2) solve the truth problem for each of the parameters in $\mathbb{P}_h$ and find the solutions $\{\psi_i\}_{i=1,...,M}$
3) assemble the matrix $\hat{C}_{ij} = w_i C_{ij}$ with $C_{ij} = (S_{\mathrm{u}}^T X_{\mathrm{u}} S_{\mathrm{u}})_{ij}$ and search the N biggest engenvectors $\xi_1, ..., \xi_N$ and related eigenvalues
4) construct $V_N = \mathrm{span}\{\xi_1, ..., \xi_N\}$

For the pressure we have the similar objective of minimizing the quantity:

$$\int_\Gamma ||p_{N_\delta}(y) - p_N(y)||_Q^2 \rho(y)\ dy, \tag{4.28}$$

and since the idea is the same that for the velocity, we will not repeat it.

4.2.2 Weighted greedy algorithm

Let us pass to the weighted greedy algorithm, similar to the deterministic one, following the works in [25], [27] and [28].
The algorithm works in the following way: at the beginning we chose a parameter y_1 at random using the probability distribution, we solve the truth problem (4.22) and we obtain the solution $\mathbf{U}_{N_\delta}(y_1) = (u_{N_\delta}(y_1), p_{N_\delta}(y_1))$. After this inizialitation the algorithm proceeds iteratively: at the $n-th$ iteration we search the parameter y_n such that $\mathbf{U}_{N_\delta}(y_n) := (\mathbf{u}_{N_\delta}(y_n), p_{N_\delta}(y_n))$ is the worst approximated solution by $\mathbf{U}_N(y_n) := (\mathbf{u}_N(y_n), p_N(y_n))$ on the whole discrete space $\mathbb{P}_h$, so we compute the truth solution with this parameter and we add it to the reduced space.
For this choice we want also use the probabilistic information of the parameters, so we choose $y \in \mathbb{P}_h$ such that:

$$\arg\max_{y \in \mathbb{P}_h} w(y) ||\mathbf{U}_{N_\delta}(y) - \mathbf{U}_N(y)||_Y, \tag{4.29}$$

where $Y := V \times Q$.
As we have said in the third chapter, explaining the greedy algorithm, we need an error estimator $\hat{\Delta}_N^{N_\delta}(y)$. In this case we take

$$\hat{\Delta}_N^{N_\delta}(y) := \Delta_N^{N_\delta}(y) \cdot w(y), \tag{4.30}$$

with $w : \Gamma \to \mathbb{R}$ a weight function and $\Delta_N^{N_\delta}(y)$ the error estimator (2.29) in the deterministic case. If we introduce the weighted norm:

$$||\mathbf{U}(y)||_w := w(y)||\mathbf{U}(y)||_Y, \tag{4.31}$$

we can give a different meaning on the maximization of $||\mathbf{U}_{N_\delta}(y) - \mathbf{U}_N(y)||_w$ according to the choice of w.

In fact if we take:

$$\mathbb{E}\left[||\mathbf{U}_{N_\delta} - \mathbf{U}_N||_Y^2\right] = \int_\Omega ||\mathbf{U}_{N_\delta} - \mathbf{U}_N||_Y^2 dP =$$
$$\int_\Gamma ||\mathbf{U}_{N_\delta}(y) - \mathbf{U}_N(y)||_Y^2 \rho(y)\ dy \leq \int_\Gamma \Delta_N^{N_\delta}(y)^2 \rho(y)\ dy,$$

and so if we take $w(y) := \sqrt{\rho(y)}$ the greedy algorithm controls the average of $||\mathbf{U}_{N_\delta} - \mathbf{U}_N||_Y^2$. In fact we have from above that:

$$\mathbb{E}[||\mathbf{U}_{N_\delta} - \mathbf{U}_N||_Y^2] \leq \int_\Gamma \hat{\Delta}_N^{N_\delta}(y)^2\ dy \leq |\Gamma| \sup_{y\in\Gamma} \hat{\Delta}_N^{N_\delta}(y)^2, \tag{4.32}$$

with $|\Gamma| < +\infty$ measure of the set Γ, finite because we have supposed the compactness.
On the contrary if we chose $w(y) := \rho(y)$ we have this control:

$$||\mathbb{E}[\mathbf{u}_{N_\delta}] - \mathbb{E}[\mathbf{u}_N]||_Y \leq \int_\Gamma ||\mathbf{U}_{N_\delta}(y) - \mathbf{U}_N(y)||_Y \rho(y)\ dy \leq |\Gamma| \sup_{y\in\Gamma} \hat{\Delta}_N^{N_\delta}(y). \tag{4.33}$$

So changing the weight w we are minimizing different quantities.
We finally note that in the algorithm we work with a finite sum, approximation of the integral, and so the parameter space over we will search the maxima of the error estimator will be $\mathbb{P}_h$ and not Γ.
We finish this chapter presenting the weighted greedy algorithm:

```
Initialization:
  take a discrete space P_h according to the probability density function
  take a tolerance ε_tol as stopping criteria for the algorithm
  choose a maximum number of reduced bases N_max
  choose a first parameter y_1 and create the sample space S_1 := {y_1}
  solve the truth problem for y_1 and create Y_N^1 := span{U_N(y_1)}

Iteration:
  for i=2,...,N_max
     choose y_i ∈ P_h such that it maximizes Δ̂_N^{N_δ}(y)
     if  Δ̂_N^{N_δ}(y_i) ≤ ε_tol
       N_max = i
     end
     solve the truth problem for y_i to obtain U_N(y_i)
     add y_i to the space S_{i-1}, creating S_i = S_{i-1} ∪ {y_i}
     add  U_N(y_i) to the reduced basis space Y_N^{i-1}, creating  Y_N^i ⊕ span{U_N(y_i)}
```

4.3 Conclusions

In this section we have firstly introduced the stochastic Stokes problem, similar to the stochastic Navier-Stokes one, in the strong and weak formulation. This one was too general and to reconduct

our problem into a reduced method framework we have done a hypothesis on the stochastic dependence on the parameters and thanks to the Doob-Lemma we have obtained the classical reduced formulation but with also a probability distribution associated to the parameters. Finally we have explained the weighted algorithms for the generation of the reduced basis: the weighted greedy and the weighted POD. They are similar to the determinist version but they use the probabilistic distribution to weight the parameters. This has led to a better choice of the basis and a lower computational cost.
We finally cite the article [31] that treats a stochastic problem in which the advection is a dominating phenomena and we need a stabilization method.

Chapter 5

Random parameter space sampling: tensor products, sparse grids, and random grids

In this chapter we will discuss different types of grid that we can use for the discrete sampling space introduced in the previous chapters. We will use the theory developed in [34]. These different types of grid will be used in the numerical experiments.
They are usually seen in a more general context, the one of the approximation of a general multi-dimensional integral of a function $f : \mathbb{R}^d \to \mathbb{R}$ with a weight function $g : \mathbb{R}^d \to \mathbb{R}$ as follows:

$$\mathcal{I}_g^d f = \int_{I_1 \times I_2 \times ... \times I_d} g(x_1, x_2, ..., x_d) f(x_1, x_2, ..., x_d)\ dx_d ... dx_1,$$

where $I_i \subset \mathbb{R}$ is an interval and so $I := I_1 \times I_2 \times ... \times I_d$ is an hyperrectangle in $\mathbb{R}^d$, $g(x_1, x_2, ..., x_d) = g_1(x_1) \cdots g_d(x_d)$ is the weight function where $g_i : I_i \to \mathbb{R}$, $g_i(x_i) \geq 0\ \forall i$: in particular in our case it is the probability density function of a random vector with probabilistic independent components. There are several methods for approximate this integral:

- with a tensor product rule [36].
- with a sparse Smolyak rule [38].
- with a Monte-Carlo method writing the integral as $\mathbb{E}[f]$ and sampling according to g seen like a probability density function of a random variable. [39]
- with a Monte-Carlo method writing the integral as $\mathbb{E}[f \cdot g]$ and sampling according to a uniform random variable [39].

The Monte-Carlo is the easiest way to do this integration but it has a slow convergence rate to the truth integral; the tensor product rule suffers of the problem of the curse of dimensionality while

the sparse Smolyak rule tries to solve this problem.
We will describe the four types of approximation in the next sections.

5.1 Tensor Product quadrature rule

In this section we will talk about tensor product rule and an algorithm to implement it.
In this case we take first a set of univariate quadrature rules $(U_k^{(j)})_{j=1}^d$ where k is the number of nodes used for the approximation.
In our case the rule is chosen depending on g_j according to the numerical integration [35]. For example if $g_j(x_j) = 1$ we use a $Gauss - Legendre$ quadrature; for $g_j(x_j) = (1 - x_j)^\alpha(1 + x_j)^\beta$ we use a $Gauss - Jacobi$ quadrature.
So for each j we have a set of k nodes $(x_i^{(j)})_{i=1}^k$ and weights $(w_i^{(j)})_{i=1}^k$ associated with the rule $U_k^{(j)}$.
With this in mind we can approximate in such a way:

$$\mathcal{I}_g^d f \approx \sum_{i_1=1}^{k} \cdots \sum_{i_d=1}^{k} w_{i_1}^{(1)} \cdots w_{i_d}^{(d)} f(x_{i_1}^{(1)}, ..., x_{i_d}^{(d)}). \tag{5.1}$$

In practice in this case we take a set of the nodes for each dimension and we do a cartesian product obtaining nodes in $\mathbb{R}^d$. After we evaluate f on them multiplying by a number that is the product of the weights associated to each components of the $d-$ dimensional node.
We can see an example of tensor product grid using a Gauss-Jacobi univariate rule quadrature in figure 5.1.
With this grid we have a problem with *the curse of dimensionality*. In fact if we use n nodes for

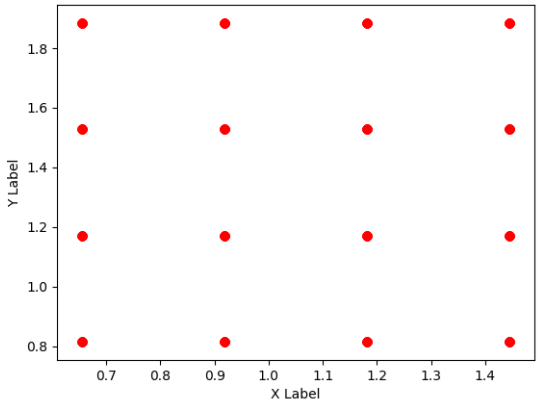

Figure 5.1: Tensor product grid with 4 nodes for a *Gauss-Jacobi* univariate rule with $d = 2$

each dimension, at the end we have n^d nodes. So if we increase the dimension of the space of the

parameters, the number of nodes involved will grow exponentially.
We will try to solve this problem with the next grid introduced in section 6.2.
Now let us pass to analyze the algorithm in Python to generate it.

```
TensorProductRule(d,n,univariate_rule,bounds,param=[])

    tmpnodes, tmpweights = univariate_rule(n,param[0],bounds[0])
  for j in range(d):
        tmp1, tmp2 = univariate_rule(n,alpha=param[j],bounds[j])

        tmpnodes = combvec(tmpnodes,tmp1)
        tmpweights = combvec(tmpweights,tmp2)

    return nodes, weights
```

with

```
combvec(Mat,vec):

    # .shape returns the dimensions of the object: 0 for rows, 1 for columns
    num_raw_mat = Mat.shape[0]
    num_col_mat = Mat.shape[1]
    num_col_vec = vec.shape[1]
    # zeros generates a matrix of zeros
    a = zeros((num_raw_mat+1,num_col_mat*num_col_vec))

    for j in range(num_raw_mat):
      for i in range(num_col_mat):
            for k in range(num_col_vec):
                a[j,k*num_col_mat+i] = Mat[j,i]

    for i in range(num_col_mat):
        for j in range(num_col_vec):
            a[num_raw_mat,j*num_col_mat+i] = vec[0,j]

    return a
```

The function *combvec* only do a cartesian product of its arguments in all the possible ways. For example if we have $\begin{pmatrix} 3 & 5 \\ 1 & 7 \end{pmatrix}$ and $\begin{pmatrix} 9 \end{pmatrix}$ the result will be

$$\begin{pmatrix} 3 & 5 \\ 1 & 7 \\ 9 & 9 \end{pmatrix}.$$

So in this code we take an univariate rule for each dimension with certain parameters (for example treating the *Beta* function we need α and β) and each time we do the cartesian product to obtain the grid.

Finally, talking about the rate of convergence we know that is $\mathcal{O}(N^{-\frac{r}{d}})$ [34] where r is the highest regularity in $H^r(I)$ of the function f.

5.2 Smolyak quadrature rule

In this section we will see what a *Smolyak quadrature rule* is using the theory developed in [34]. This rule is created from a sequence of tensor product grids of low order of approximation, but to define it we have to introduce the tensor product more formally than before.
Let S and T two univariate rule operators such that

$$Sf = \sum_{i=1}^{m} a_i f(x_i),$$

and

$$Tf = \sum_{i=1}^{n} b_i f(y_i),$$

we define the tensor product operator $T \otimes S$ as follows

$$(T \otimes S)f = \sum_{i=1}^{m}\sum_{j=1}^{n} a_i b_j f(x_i, y_j),$$

In general if we have a sequence of univariate rule operators $\{T_i\}_{i=1,...,n}$ we can define recursively

$$\bigotimes_{i=1}^{1} T_i = T_1, \qquad \bigotimes_{i=1}^{n} T_i = \bigotimes_{i=1}^{n-1} T_i \otimes T_n, \text{ for } n = 2, 3, 4, ...,$$

and the quadrature operator $\bigotimes_{i=1}^{n} T_i$ is such that:

$$\bigotimes_{i=1}^{n} T_i f = \sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} w_{i_1}^{(1)} w_{i_2}^{(2)} \cdots w_{i_n}^{(n)} f(x_{i_1}^{(1)}, ..., x_{i_n}^{(n)}), \text{ for } n = 1, 2, 3, ...$$

So it is nothing but the tensor product rule of the previous section. We are now ready to introduce the new rule.

Definition 1. *(Smolyak quadrature rule): Let $(U_i^{(j)})_{i=1}^{\infty}$ be a sequence of univariate quadrature rules in the interval $\emptyset \neq I_j \subset \mathbb{R}, j = 1, ..., d$.*
We introduce the difference operators in I_j by setting

$$\Delta_0^{(j)} = 0, \qquad \Delta_1^{(j)} = U_1^j, \qquad \Delta_{i+1}^{(j)} = U_{i+1}^{(j)} - U_i^{(j)}, \text{ for } i = 1, 2, 3, ...$$

The Smolyak quadrature rule of order k in the hyperrectangle $I_1 \times I_2 \times \cdots \times I_d$ is the operator

$$\mathcal{Q}_k^d = \sum_{|\alpha|_1 \leq k, \alpha \in \mathbb{N}^d} \bigotimes_{i=1}^{d} \Delta_{\alpha_i}^{(i)}. \tag{5.2}$$

We note that the tensor product $\Delta_{\alpha_1}^{(1)} \otimes \cdots \otimes \Delta_{\alpha_d}^{(d)}$ vanishes whenever $\alpha_i = 0$ for some index i so we will assume that $\alpha_i \geq 1 \ \forall i$ and so we need $k \geq d$.
Let us try to understand a simple case, $d = 1$:

$$\mathcal{Q}_k^1 = \sum_{i=1}^{k} \Delta_i^{(1)} = U_1^{(1)} + (U_2^{(1)} - U_1^{(1)}) + \cdots + (U_k^{(1)} - U_{k-1}^{(1)}) = U_k^{(1)}, \ \forall k \geq 1. \tag{5.3}$$

We can write with the tensor product quadrature written with the difference operators:

$$\begin{aligned} &\bigotimes_{i=1}^{d} U_k^{(i)} = \Big(\sum_{\alpha_1=0}^{k} \Delta_{\alpha_1}^{(1)} \Big) \otimes \cdots \otimes \Big(\sum_{\alpha_d}^{d} \Delta_{\alpha_d}^{(d)} \Big) = \\ &\sum_{\alpha_1=0}^{k} \cdots \sum_{\alpha_d=0}^{k} \bigotimes_{i=1}^{d} \Delta_{\alpha_i}^{(i)} = \sum_{|\alpha|_\infty \leq k, \alpha \in \mathbb{N}^d} \bigotimes_{i=1}^{d} \Delta_{\alpha_i}^{(i)}, \end{aligned} \tag{5.4}$$

in which we have used the distributive property.
As we can see in the tensor product grid we are changing the norm of α from a L^1 norm of the sparse grid to a L^∞ one.
Now we need another form for the Smolyak quadrature rule, easier to implement and we want something related to the tensor product because we already know how to implement it.
We will do this in two steps with two theorems.

Theorem 6. *Let $\alpha \in \mathbb{N}^d$ and $\alpha \geq \mathbb{1}$, i.e. $\alpha_i \geq 1, \forall i$. Then*

$$\bigotimes_{i=1}^{d} \Delta_{\alpha_i}^{(i)} = \sum_{\substack{\gamma \in \{0,1\}^d \\ \alpha - \gamma \geq \mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U_{\alpha_i - \gamma_i}^{(i)}. \tag{5.5}$$

Proof. Let us use the induction on d.
In the case of $d = 1$ we can have two cases: for $i = 1$:

$$\Delta_1^{(1)} = U_1^{(1)} = (-1)^0 U_{1-0}^{(0)},$$

and for $i > 1$:

$$\Delta_i^{(1)} = U_i^{(1)} - U_{i-1}^{(1)} = (-1)^0 U_{i-0}^{(1)} + (-1)^1 U_{i-1}^{(1)}, \quad \forall i \geq 2.$$

Now we suppose that the claim holds for all values less or equal than d. We want to prove for $d+1$. Let $\alpha \in \mathbb{N}^{d+1}$ and $\alpha \geq \mathbb{1}$.

$$\begin{aligned}\sum_{\substack{\gamma\in\{0,1\}^{d+1}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d+1} U^{(i)}_{\alpha_i-\gamma_i} &= \sum_{\substack{\gamma\in\{0,1\}^{d}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1+0} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i} \otimes U_{\alpha_{d+1}-0} \\ &\quad + \sum_{\substack{\gamma\in\{0,1\}^{d}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1+1} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i} \otimes U_{\alpha_{d+1}-1} \\ &= \sum_{\substack{\gamma\in\{0,1\}^{d}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i} \otimes \Delta^{(d+1)}_{\alpha_{d+1}},\end{aligned}$$

since $\Delta^{(d+1)}_{\alpha_{d+1}} = U^{(d+1)}_{\alpha_{d+1}} - U_{\alpha_{d+1}-1}$. The first equivalence is related to the fact that we are summing on γ that can be any element of $\{0,1\}^d$ and so γ_{d+1} can be or 0 or 1.
From the induction hypothesis we know that:

$$\sum_{\substack{\gamma\in\{0,1\}^{d}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i} = \bigotimes_{i=1}^{d} \Delta^{(i)}_{\alpha_i},$$

that substituted above gives the thesis. □
So now we are arrived to:

$$\mathcal{Q}^d_k = \sum_{\substack{|\alpha|_1\leq k\\ \alpha\in\mathbb{N}^d}} \sum_{\substack{\gamma\in\{0,1\}^{d}\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i} = \sum_{\gamma\in\{0,1\}^d} \sum_{\substack{|\alpha|_1\leq k\\ \alpha\in\mathbb{N}^d\\ \alpha-\gamma\geq\mathbb{1}}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U^{(i)}_{\alpha_i-\gamma_i}.$$

Introducing $\beta = \alpha - \gamma$ with condition $\beta \geq \mathbb{1}$, $|\beta|_1 \leq k - |\gamma|_1$ and $|\beta|_1 \leq |\beta|_1 + |\gamma|_1 \leq k$, we can change the order of summation:

$$\mathcal{Q}^d_k = \sum_{\substack{|\beta|_1\leq k\\ \beta\in\mathbb{N}^d,\beta\geq\mathbb{1}}} \sum_{\substack{\gamma\in\{0,1\}^{d}\\ |\gamma|_1\leq k-|\beta|_1}} (-1)^{|\gamma|_1} \bigotimes_{i=1}^{d} U^{(i)}_{\beta_i}.$$

Moreover it holds:

$$\begin{aligned}\sum_{\substack{\gamma\in\{0,1\}^d\\ |\gamma|_1\leq k-|\beta|_1}}(-1)^{|\gamma|_1} &= \sum_{i=0}^{\min\{d,k-|\beta|_1\}}(-1)^i\sum_{\substack{\gamma\in\{0,1\}^d\\ |\gamma|_1=i}}1\\ &= \sum_{i=0}^{\min\{d,k-|\beta|_1\}}(-1)^i\#\{\gamma\in\{0,1\}^d;|\gamma|_1=i\}\\ &= \sum_{i=0}^{\min\{d,k-|\beta|_1\}}(-1)^i\binom{d}{i},\end{aligned}$$

where the first equivalence is true because $|\gamma|_1\leq d$ but it holds $d\leq k-|\beta|_1$ or $d\geq k-|\beta|_1$ but in this last case we have to take into account the restrinction that $|\gamma|_1\leq k-|\beta|_1$.
From this observation we conclude that the term above vanishes whenever $d\leq k-|\beta|_1$ and so we can discard these multi-indices. If we remember that $\beta\geq\mathbb{1}$, adding this last condition, we obtain:

$$|\beta|_1\geq max\{d,k-d+1\}.$$

Now we can use a result from [34]

$$\sum_{\substack{\gamma\in\{0,1\}^d\\ |\gamma|_1\leq k-|\beta|_1}}(-1)^{|\gamma|_1}=(-1)^{k-|\beta|_1}\binom{d-1}{k-|\beta|_1}.$$

Putting all togheter we obtain that if $k\geq d$, we have the characterization:

$$\mathcal{Q}_k^d=\sum_{\substack{max\{d,k-d+1\}\leq|\beta|_1\leq k\\ \beta\in\mathbb{N}^d,\beta\geq\mathbb{1}}}(-1)^{k-|\alpha|_1}\binom{d-1}{k-|\alpha|_1}\bigotimes_{i=1}^{d}U_{\beta_i}^{(i)}. \tag{5.6}$$

□

Let us see an example to understand better the Smolyak formula:

$$\mathcal{Q}_5^3=\sum_{\substack{3\leq|\alpha|_1\leq5\\ \alpha\in\mathbb{N}^3,\alpha\geq\mathbb{1}}}(-1)^{5-|\alpha|_1}\binom{2}{5-|\alpha|_1}U_{\alpha_1}\otimes U_{\alpha_2}\otimes U_{\alpha_3}=$$

$$(-1)^2\binom{2}{2}U_1\otimes U_1\otimes U_1+(-1)^1\binom{2}{1}(U_2\otimes U_1\otimes U_1+U_1\otimes U_2\otimes U_1+U_1\otimes U_1\otimes U_2)+$$

$$(-1)^0\binom{2}{0}(U_2\otimes U_2\otimes U_1+U_2\otimes U_1\otimes U_2+U_1\otimes U_2\otimes U_2+U_3\otimes U_1\otimes U_1+U_1\otimes U_3\otimes U_1+$$

$$U_1\otimes U_1\otimes U_3)=U_1\otimes U_1\otimes U_1-2U_2\otimes U_1\otimes U_1-2U_1\otimes U_2\otimes U_1-2U_1\otimes U_1\otimes U_2+$$

$$U_2\otimes U_2\otimes U_1+U_2\otimes U_1\otimes U_2+U_1\otimes U_2\otimes U_2+U_3\otimes U_1\otimes U_1+U_1\otimes U_3\otimes U_1+$$

$$U_1\otimes U_1\otimes U_3.$$

As we can see $\mathcal{Q}_5^3$ is a combination of terms of tensor grid rules but with a lower order with respect to the associated tensor product rule $\bigotimes_{i=1}^{3} U_5^{(i)}$.

We can better understand visually with an example of the nodes generated by this rule using $\mathcal{Q}_5^2$ in the case in which the univariate rule is the *Gauss Jacobi*, in the figure 5.2.

In figure 5.3 we have a $\mathcal{Q}_6^2$ with the same univariate rule.

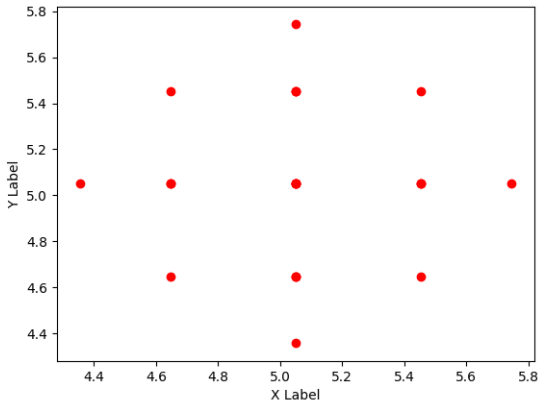

Figure 5.2: Sparse tensor grid $\mathcal{Q}_5^2$ with a *Gauss Jacobi* rule

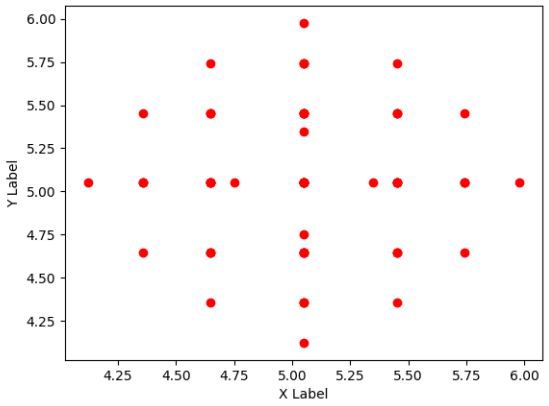

Figure 5.3: Sparse tensor grid $\mathcal{Q}_6^2$ with a *Gauss Jacobi* rule

As we can see the points are not distributed on the boundary but in the inner part of the domain. If we want to know how many nodes we will have with a certain rule we can use this theorem:

Theorem 7. *Let $U_i^{(j)}$ be univariate quadrature rules with $n_i^{(j)} = 2^{i-1}$ nodes and $k \geq d \geq 1$. The number of evaluation nodes of $\mathcal{Q}_k^d$ is*

$$\sum_{i=max\{d,k-d+1\}}^{k} 2^{i-d} \binom{i-1}{d-1}.$$

Let us pass to the algorithms for creating a Smolyak grid:

```
def SmolyakRule(d,q,rule,bounds,param=[]):


    # d is the dimension of the integral and the quadrature nodes too.
    # q is the order of the quadrature rule
    # param is the set of parameters
    # bounds is the vector that contains the bounds of the interval

    bounds = np.array(bounds)
    nodes = np.zeros((d,1))
    weights = np.zeros((1,1))
    for l in range(max(d,q-d+1),q+1):
        # l==d means that I've only alpha = (1,1,1,...,1)
        # so I cannot generate with d_tuple algorithm

        if l == d:

            tmpnodes = np.zeros((d,1))
            tmpweights = np.ones((1,1))
            for i in range(d):

                tmp1, tmp2 = univariate_rule(1,rule,alpha=param[i][1],beta=param[i
    ][0])
                tmpnodes[i,0] = (bounds[i,1]-bounds[i,0])*tmp1[0,0] + bounds[i,0]
                # i do not want a vector of all weights but only the product
                tmpweights[0,0] = (bounds[i,1]-bounds[i,0])*tmp2[0,0]*tmpweights
    [0,0]

            tmpweights[0,0] = (-1)**(q-l)*au.binom_coeff(d-1,q-l)*tmpweights[0,0]
            nodes = np.concatenate((nodes,tmpnodes),axis=1) # axis = 1 adds as a
    column
            weights = np.concatenate((weights,tmpweights),axis=1)

        else:

            ind, m = d_tuples(d,l)

            for i in range(m):

```

```
                gamma = ind[i,:] # we take one alpha at a time
                tmpnodes, tmpweights = univariate_rule(int(gamma[0]),rule,alpha=
    param[0][1],beta=param[0][0])
                #tmpnodes, tmpweights = univariate_rule(1,rule,alpha=param[0][0],
    beta=param[0][1])
                for j in range(1,d):
                    tmp1, tmp2 = univariate_rule(int(gamma[j]),rule,alpha=param[j
    ][1],beta=param[j][0])
                    #tmp1, tmp2 = univariate_rule(1,rule,alpha=param[j][0],beta=
    param[j][1])
                    tmpnodes = combvec(tmpnodes,tmp1)
                    tmpweights = combvec(tmpweights,tmp2)

                for j in range(d):
                    for k in range(tmpnodes.shape[1]):
                        tmpnodes[j,k] = (bounds[j,1]-bounds[j,0])*tmpnodes[j,k] +
    bounds[j,0]
                        tmpweights[j,k] = (bounds[j,1]-bounds[j,0])*tmpweights[j,k]

                # product of different component of the weight associated to the
    same node
                tmpweights = (-1)**(q-l)*au.binom_coeff(d-1,q-l)*np.array([np.prod(
    tmpweights, axis=0)])

                nodes = np.concatenate((nodes,tmpnodes),axis=1)
                weights = np.concatenate((weights,tmpweights),axis=1)



    return nodes, weights, count
```

with

```
def d_tuples(d, q):
    # this combinatorial algorithm generates the different tuples alpha for the
    Smolyak rule
    k = np.ones((1,d))
    khat = (q-d+1)*k
    ind = np.zeros((1,d))

    p = 0
    while k[0,d-1] <= q:
        k[0,p] = k[0,p]+1
        if k[0,p] > khat[0,p]:
            if p != d-1:
                k[0,p] = 1
                p = p+1
        else:
            for j in range(p):
                khat[0,j] = khat[0,p]-k[0,p]+1
            k[0,0] = khat[0,0]
```

```
            p = 0
            ind = np.concatenate((ind,k))

    n = ind.shape[0]
    ind = ind[1:n,:]
    n = ind.shape[0]

    return ind, n
```

For the algorithm we need these ingredients:

1. An univariate rule for generating nodes and weights.
2. A generator of vector $\alpha \in \mathbb{N}^d$ such that $\alpha \geq \mathbb{1}$ and $|\alpha| = l$ with $l \geq d$ that is implemented in *d_tuples*.
3. The *combvec* algorithm for the cartesian product.

We have split the cases of $l == d$ and $l > d$ because the implementation of the algorithm *d_tuples*. In both cases the structure is this one:

1. We generate the nodes and the weights of an univariate rule.
2. We do a cartesian product along all the directions.
3. We multiply for the coefficient $(-1)^{k-|\alpha|_1}\binom{d-1}{k-|\alpha|_1}$.
4. We concatenate (append) with the existing nodes and weights.

Finally concerning the accuracy we present a result from [34].

Theorem 8. *(Polinomial precision): Let U_i be univariate quadrature rules that correspond to the weight w_j and have polynomial exactness $m_i \leq m_{i+1}$. Let $w(x_1, ..., x_d) = w_1(x_1) \cdots w_d(x_d)$. Then:*

$$\mathcal{I}_w^d f = \mathcal{Q}_q^d(f), \quad \forall f \in \sum_{\substack{|\alpha_1|=q \\ \alpha \in \mathbb{N}^d}} \bigotimes_{i=1}^d \mathbb{P}^1_{m_{\alpha_i}}, \quad q \geq d,$$

where

$$\bigotimes_{i=1}^d \mathbb{P}^1_{m_{\alpha_i}} = \{f : (x_1, x_2, ..., x_d) \in \mathbb{R}^d \to \prod_{i=1}^d p_i(x_i) \in \mathbb{R}; p_i \in \mathbb{P}^1_{m_i}, for\ i = 1, ..., d\}.$$

The following theorem gives us the convergence rate of the Smolyak quadrature rule [34].

Theorem 9. *(Fundamental theorem of Smolyak quadrature). Let $n_i = 2^{i-1}$ denote the number of evaluation points of interpolatory quadrature rules U_i with positive weights in $[-1, 1]$. If we denote $N(k, d) = \#$ number of nodes, then the corresponding Smolyak quadrature rule of degree k has the asymptotic convergence rate of*

$$\left| \int_{[-1,1]^d} f(x_1, ..., x_d) dx_d \cdots dx_1 - \sum_{|\alpha|_1 \leq k, \alpha \in \mathbb{N}^d} \bigotimes_{i=1}^{d} \Delta_{\alpha_i} f \right| = \mathcal{O}\Big(\frac{log(N(k,d))^{(r+1)(d-1)}}{N(k,d)^r}\Big),$$

for $f \in H^r([-1, 1]^d)$.

5.3 Monte-Carlo Methods

In this case we have a completely different approach which relies on seeing the integral in the following way:

$$\mathbb{E}\big[f(X)\big],$$

with X random variable with density distribution g with compact support.
So we have to compute:

$$\mathbb{E}\big[f(X)\big] = \int_{I_1 \times I_2 \times ... \times I_d} g(x_1, x_2, ..., x_d) f(x_1, x_2, ..., x_d)\ dx_d ... dx_1.$$

We can approximate this integral with a *Monte-Carlo method* in two ways:

- we take several realizations of the random variable X and after we approximate as:

$$\mathbb{E}\big[f(X)\big] \approx \sum_{i=1}^{N} \frac{1}{N} f(X_i),$$

  with N large enough.

- we take several realizations of a uniform random variable U and after we approximate as:

$$\mathbb{E}\big[f(X)\big] = \mathbb{E}\big[(f \cdot g)(U)\big] \approx \sum_{i=1}^{N} \frac{1}{N} f(U_i) g(U_i),$$

  with N large enough.

We expect that the two methods for $n \to \infty$ converge to the same value. In this case the convergence rate is $\mathcal{O}(\frac{1}{\sqrt{N}})$ [36].

5.4 Conclusions

In this chapter we have introduced some methods to approximate a multidimensional weighted integral of a general function: the tensor product rule, the sparse Smolyak rule and the Monte-Carlo rule with two variants.
In the first rule we numerically integrate along all the dimensions and we subsequently do a cartesian product of the nodes to obtain a grid of nodes. We subsequently multiply the weights along all the dimensions associated with the components of a multidimensional node to obtain a weight related to it.
The sparse rule uses instead a linear combination of tensor product rules of low computational cost.
The Monte-Carlo rule approximates the average integral with a finite sum: in the first variant of this rule we sample according to the weight distribution and in the second one according to a uniform distribution.
We have seen, studying the convergence rate, that except for the sparse rule, the computational cost is too high due to the curse of dimensionality and then the Smolyak rule is sometimes the only one that we can use for a numerical mutidimensional integration.

Chapter 6

Numerical results

In this final chapter we will propose some experiments we have done for testing the effectiveness of the algorithms that we have introduced before.
We will work with the same problem in all the cases but changing the distributions from the parameters came. In particular we have used four types of $Beta(\alpha, \beta)$, normally concentrated in $[0, 1]$ and after translated into the desired range, which depend on α and β: by varying them the shape of the distribution can completely change. This is one of the two reasons why we have chosen the *Beta* distribution. The other one is because the *Beta* has a compact support, necessary for the hypothesis of the stochastic reduced approach, and we do not need to truncate it.
Its density distribution is:

$$f(x; \alpha, \beta) := \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{6.1}$$

with $\dfrac{1}{B(\alpha, \beta)} = \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$ where Γ is the Gamma function.
The set of parameters we have chosen for our experiments are $(\alpha, \beta) = (75, 75), (10, 10)$, $(20, 1)$, $(0.03, 0.03)$ and we can see the plot of the associated distributions in the figure 6.1.
The choice of $(\alpha, \beta) = (75, 75)$ or $(10, 10)$ was done for simulating something similar to a Gaussian distribution but with different heights. In these cases the distribution is symmetric and concentrated in a certain range. In the case of $(\alpha, \beta) = (20, 1)$ we have a distribution not symmetric and concentrated around $x = 1$. Finally with $(\alpha, \beta) = (0.03, 0.03)$ we have again a symmetric distribution but concentrated in two zones instead of one as in the other cases.

The problem we have chosen in the Stokes case is the following:

$$\begin{cases} -\Delta \mathbf{u}(x, y; \boldsymbol{\mu}) + \nabla p(x, y; \boldsymbol{\mu}) = \mathbf{f}(x, y; \boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u}(x, y; \boldsymbol{\mu}) = 0 & \text{in } \Omega(\boldsymbol{\mu}), \\ \mathbf{u}(x, y; \boldsymbol{\mu}) = \mathbf{0} & \text{on } \Gamma_w(\boldsymbol{\mu}), \\ \mathbf{u}(x, y; \boldsymbol{\mu}) = \mu_4 \cdot (-y \cdot (y-3), 0) & \text{on } \Gamma_{in}, \\ \nu \dfrac{\partial \mathbf{u}}{\partial \mathbf{n}}(x, y; \boldsymbol{\mu}) - p(x, y; \boldsymbol{\mu})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{out}. \end{cases} \tag{6.2}$$

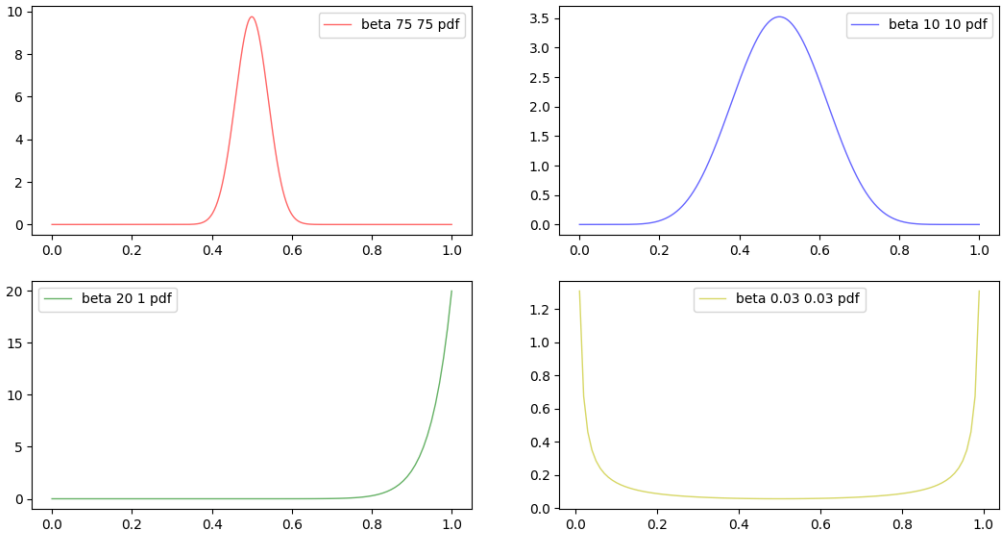

Figure 6.1: Beta distributions used in the experiments

For the Navier-Stokes case we have taken the same problem but the only difference is the addition of the convective term $\mathbf{u} \cdot \nabla \mathbf{u}$ as usual.
The geometry is that one in figure 6.2, where the triangles denotes the decomposition of our domain ($\Omega = \cup_{r=1}^{R} \Omega^r$) as explained in the third chapter.

For what concerns the boundaries, Γ_{in} is the left side, Γ_{out} is the right side while Γ_w is equal to $\partial\Omega \setminus (\Gamma_{out} \cup \Gamma_{in})$.
The parameters involved are:

$$\boldsymbol{\mu} = (\mu_0, \mu_1, \mu_2, \mu_3, \mu_4) = (L_1, h_1, L_2, h_2, y_{max}). \tag{6.3}$$

Let us explain them. With y_{max} we denote the maximum value of the parabola in the boundary condition. The other parameters are geometrical and referred to the two rectangles. L is the lenght of the base of the rectangle while h is the height of this one. So in this case we are parameterizing the dimensions of the two rectangles.

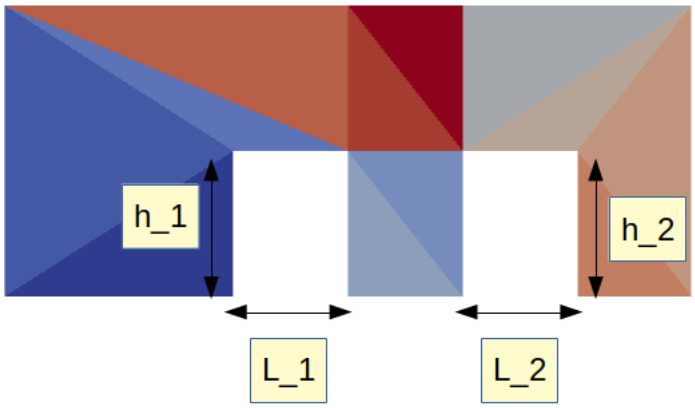

Figure 6.2: Geometry of the problem with four geometrical dependencies: L_1, h_1, L_2, h_2.

The ranges considered for the five parameters are:

$$\mu_0 \in (0.2, 1.9),$$
$$\mu_1 \in (0.2, 2.0),$$
$$\mu_2 \in (0.2, 1.9),$$
$$\mu_3 \in (0.2, 2.0),$$
$$\mu_4 \in (0.2, 20.0).$$

We have taken around $M = 240$ parameters for the learning set $\mathbb{P}_h$ in all the experiments. Sometimes, with the tensor product rule and the Smolyak rule, we have obtained more or less parameters depending on the case because it is more difficult to have the exact number that we want.
The experiments have been done changing each time the probability distribution and the algorithm used.
For the Stokes problem we have organized the experiment for the velocity comparing the methods as written below:

- First experiment: Standard greedy, weighted greedy, standard *POD* and weighted *POD*.
- Second experiment: Monte-Carlo *POD*, Tensor product *POD* with a Gauss-Jacobi quadrature rule, Sparse rule *POD*.
- Third experiment: Monte-Carlo *POD*, Uniform Monte-Carlo *POD*

The pressure has been investigated in the Navier-Stokes section. In this case we have organized the results of the velocity in the following way:

- First experiment: standard *POD* and weighted Monte-Carlo *POD*.
- Second experiment: Monte-Carlo *POD* and tensor product rule *POD* with a Gauss-Jacobi quadrature rule, sparse rule *POD*.

We have compared also the pressures:

- Third experiment: standard *POD* and weighted *POD* with tensor product rule.

In the following plots we will see the absolute error and the relative error with a H^1-semi-norm for the velocity and a L^2-norm for the pressure on the y-axis (only for the Navier-Stokes problem) while on the $x-$axis we have the number N of basis used for the reduced solution. They are plotted with a logarithm scale. In addition we will show the maximum error and the relative maximum error with the L^∞-norm.
These errors have been computed taking 100 parameters obtained randomly according to the chosen distribution. For each one we have found the truth solution and the reduced one changing the number of basis.
Mathematically, for each N we have computed :

$$\text{absolute error: } \int_\Omega |\mathbf{u}_{N_\delta}(Y(\omega)) - \mathbf{u}_N(Y(\omega))|_{H^1} dP \approx \frac{1}{100}\sum_{i=1}^{100} |\mathbf{u}_{N_\delta}(y_i) - \mathbf{u}_N(y_i)|_{H^1},$$

$$\text{absolute maximum error: } \max_{i=1,\dots,100} |\mathbf{u}_{N_\delta}(y_i) - \mathbf{u}_N(y_i)|_{H^1},$$

$$\text{relative error: } \frac{1}{100}\frac{\sum_{i=1}^{100} |\mathbf{u}_{N_\delta}(y_i) - \mathbf{u}_N(y_i)|_{H^1}}{|\mathbf{u}_{N_\delta}(y_i)|_{H^1}},$$

$$\text{relative maximum error: } \max_{i=1,\dots,100} \frac{|\mathbf{u}_{N_\delta}(y_i) - \mathbf{u}_N(y_i)|_{H^1}}{|\mathbf{u}_{N_\delta}(y_i)|_{H^1}}.$$

All the following computations have been done using RBniCS library [3] which is based on FEniCS.

6.1 Stokes problem

In this section we will present the numerical experiments for the Stokes problem as we have explained above for the case of parameters coming from a *Beta* distribution with (α, β) equal to: $(0.03, 0.03), (10, 10), (20, 1), (75, 75)$.
We can see in the figure 6.3 an example of simulation with a reduced basis where we have plotted the velocity magnitude. In this case we have no separation zone as expected because it is a typical phenomena related to the convective term not present in the Stokes equations.

6.1.1 Stokes: Beta 0.03 0.03

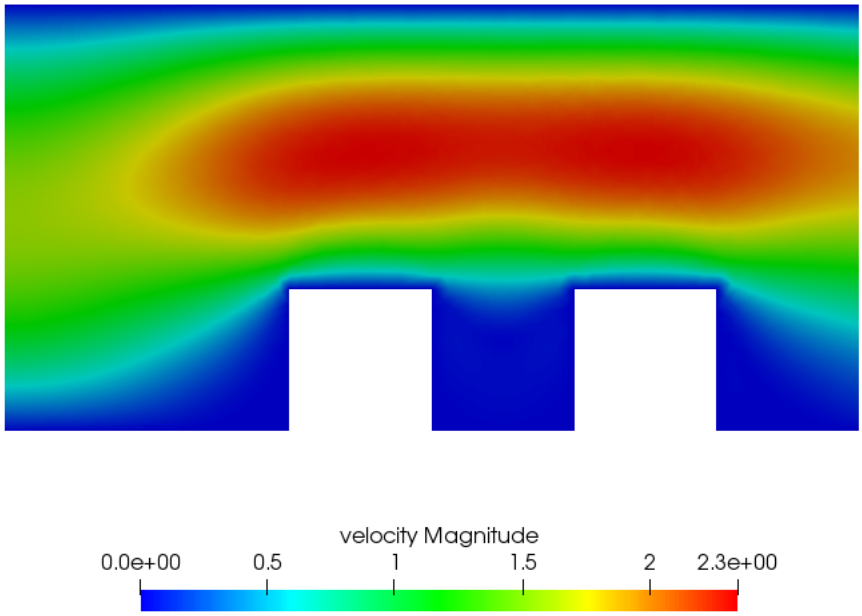

Figure 6.3: Stokes simulation with parameter $(1.0, 1.0, 1.0, 1.0, 0.7)$: velocity magnitude profile.

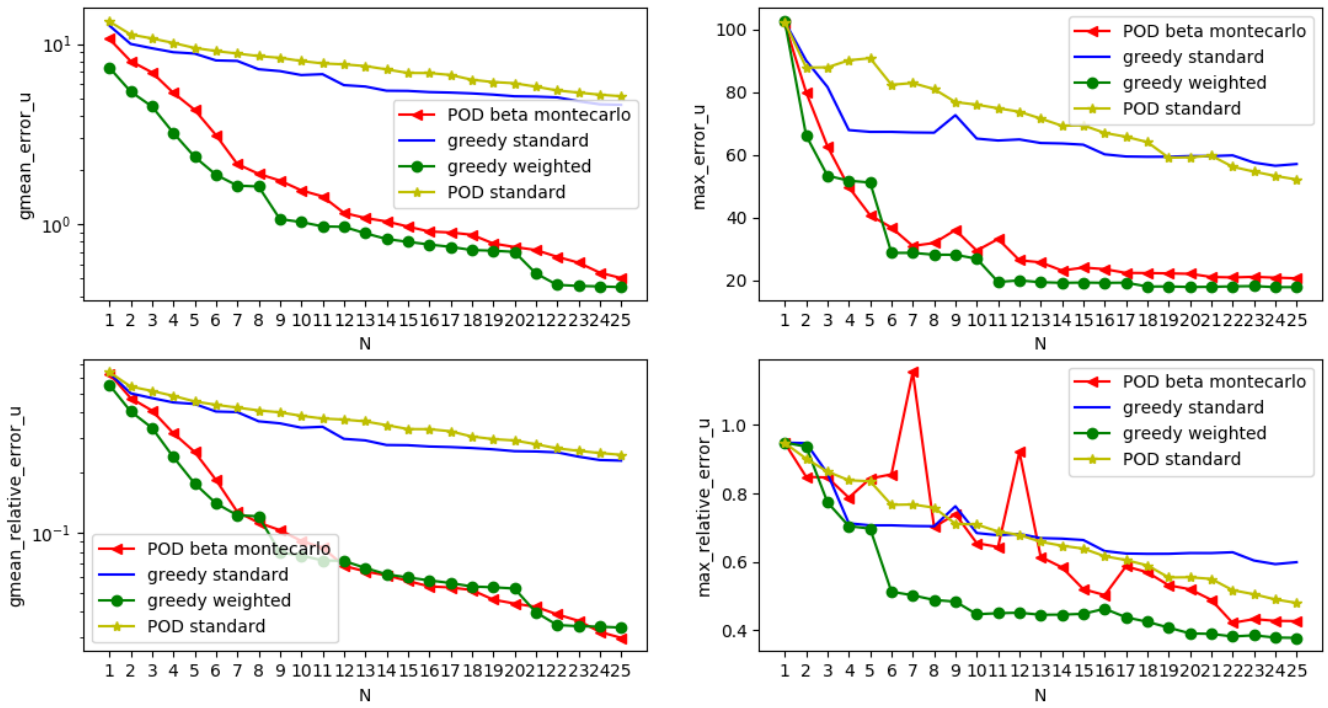

Figure 6.4: Stokes, first experiment: standard greedy, weighted greedy, standard *POD* and weighted *POD*, with a $Beta(0.03, 0.03)$

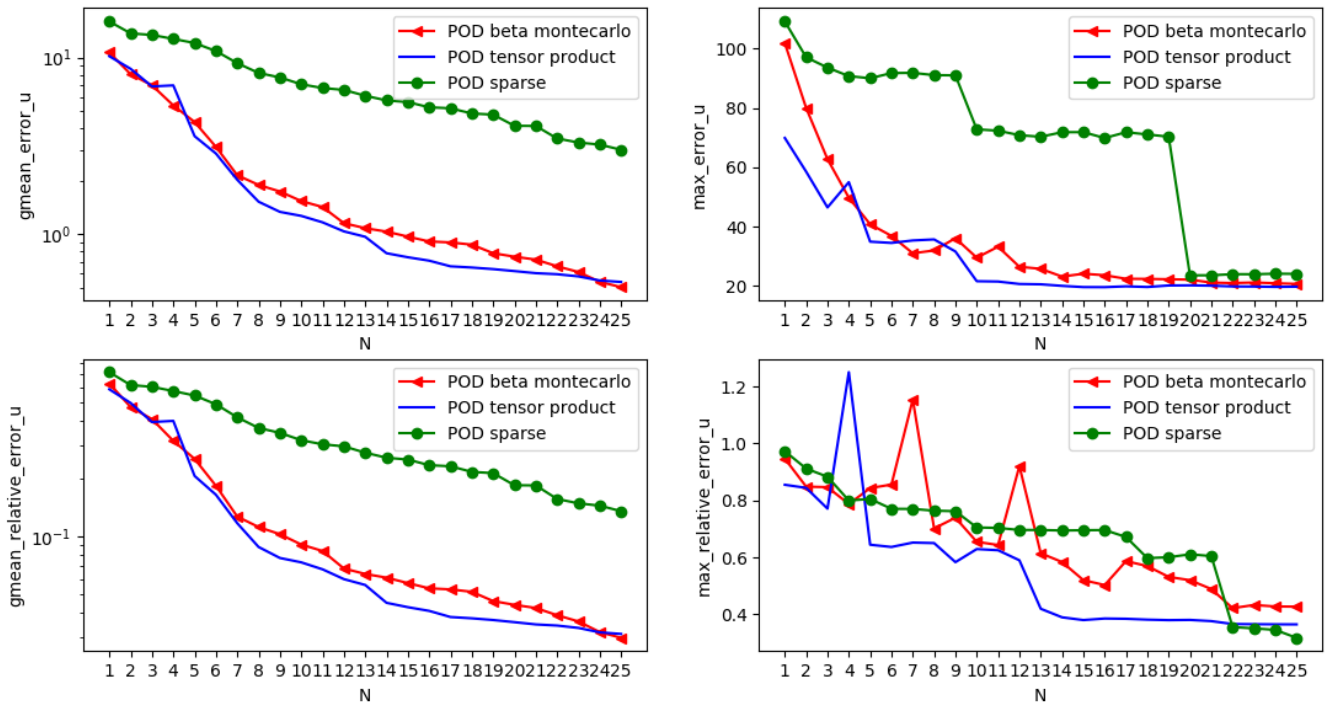

Figure 6.5: Stokes, second experiment: Monte-Carlo *POD*, Tensor product *POD* with a Gauss-Jacobi quadrature rule, Sparse rule *POD*, with a $Beta(0.03, 0.03)$.

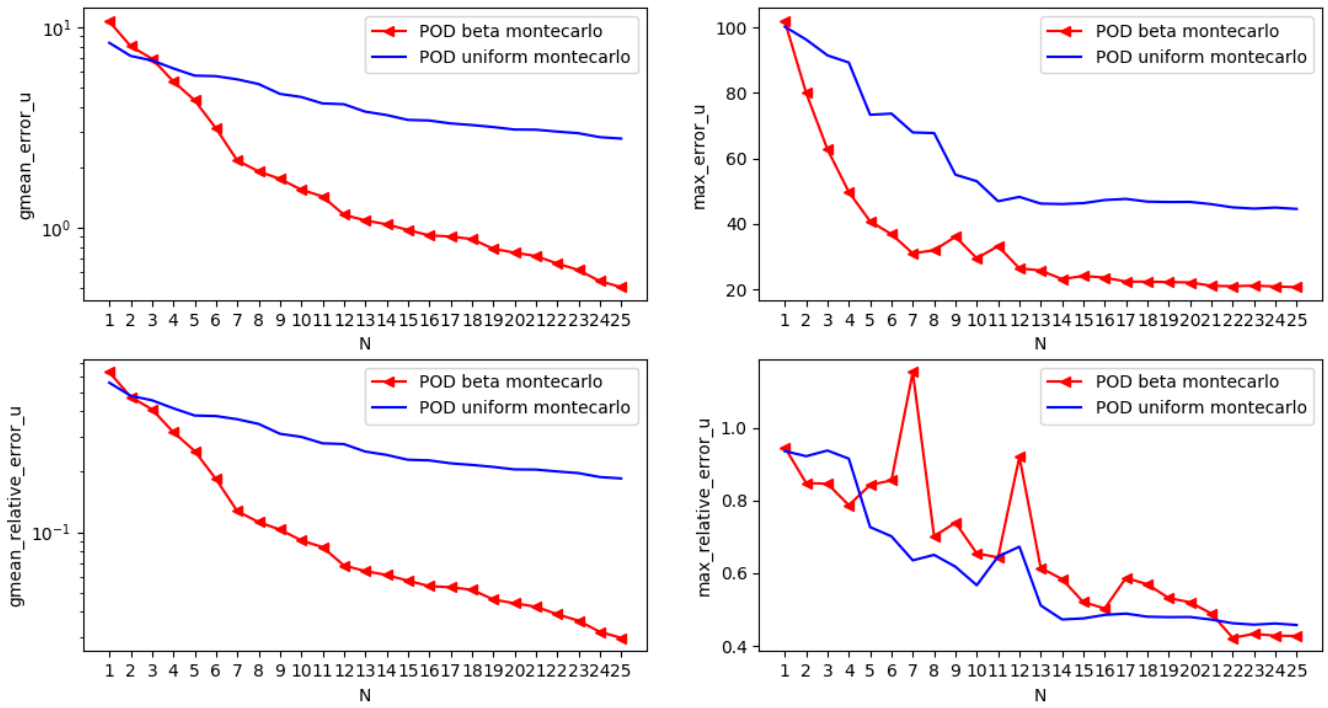

Figure 6.6: Stokes, third experiment: Monte-Carlo *POD* and Uniform Monte-Carlo *POD*, with a $Beta(0.03, 0.03)$.

6.1.2 Stokes: Beta 10 10

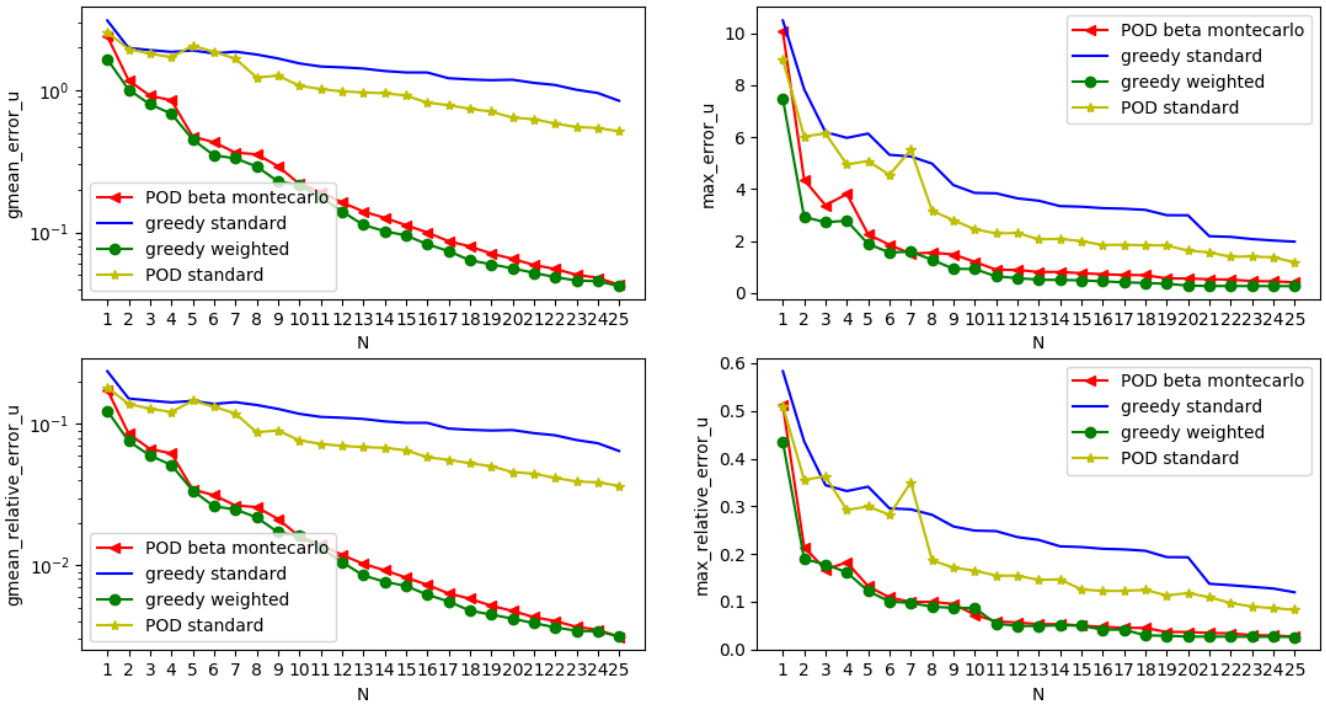

Figure 6.7: Stokes, first experiment: standard greedy, weighted greedy, standard *POD* and weighted *POD*, with a $Beta(10, 10)$.

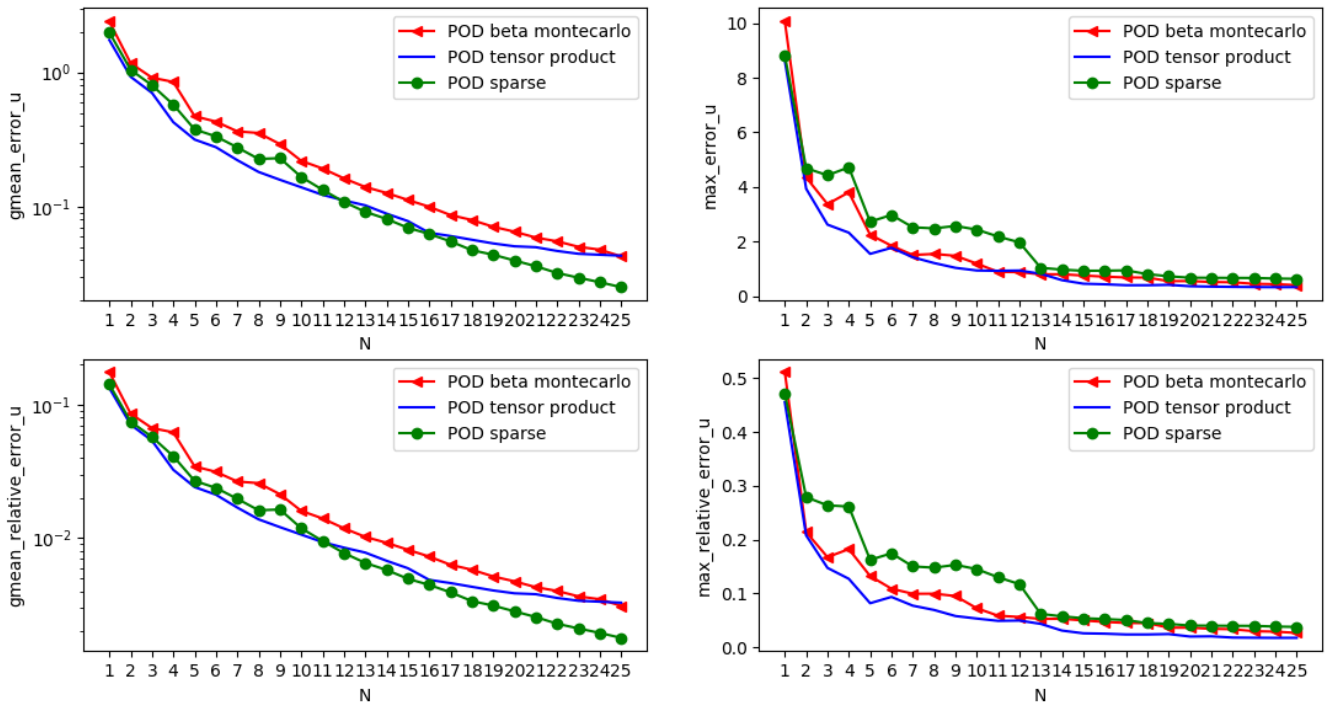

Figure 6.8: Stokes, second experiment: Monte-Carlo *POD*, Tensor product *POD* with a Gauss-Jacobi quadrature rule, Sparse rule *POD*, with a $Beta(10, 10)$.

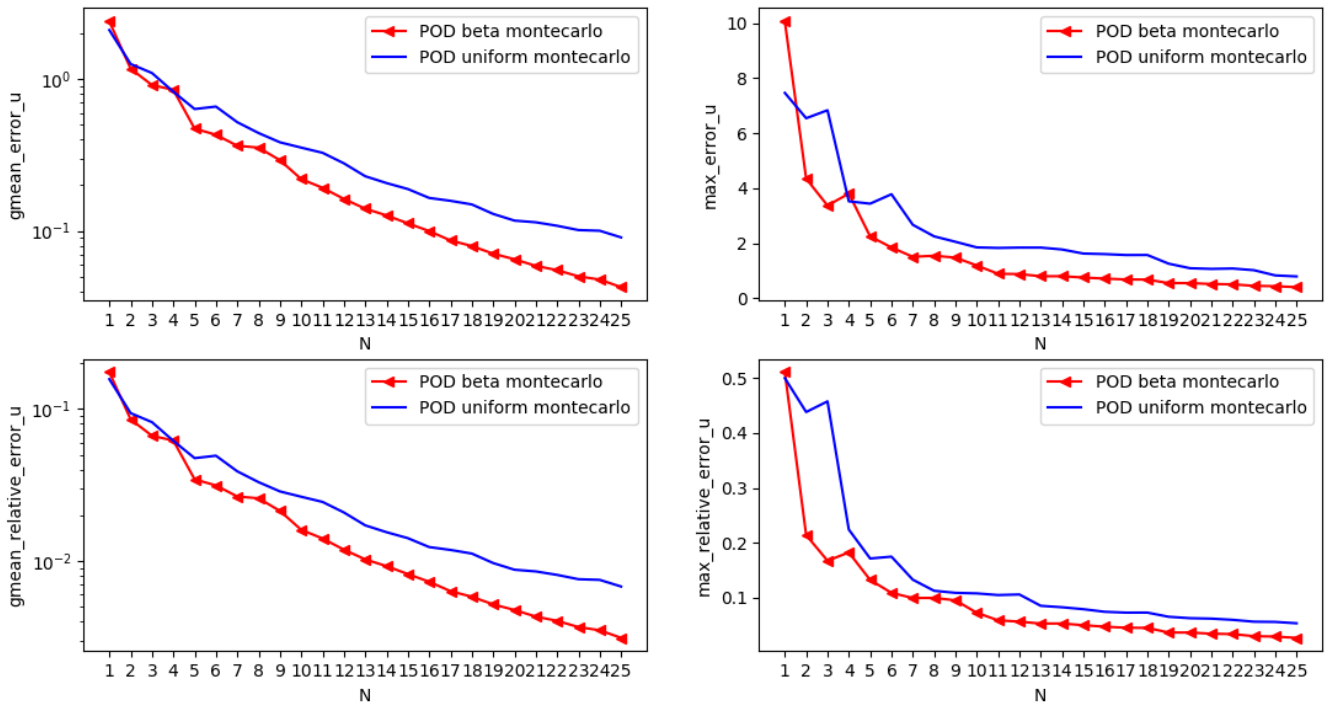

Figure 6.9: Stokes, third experiment: Monte-Carlo *POD* and Uniform Monte-Carlo *POD*, with a $Beta(10, 10)$.

6.1.3 Stokes: Beta 20 1

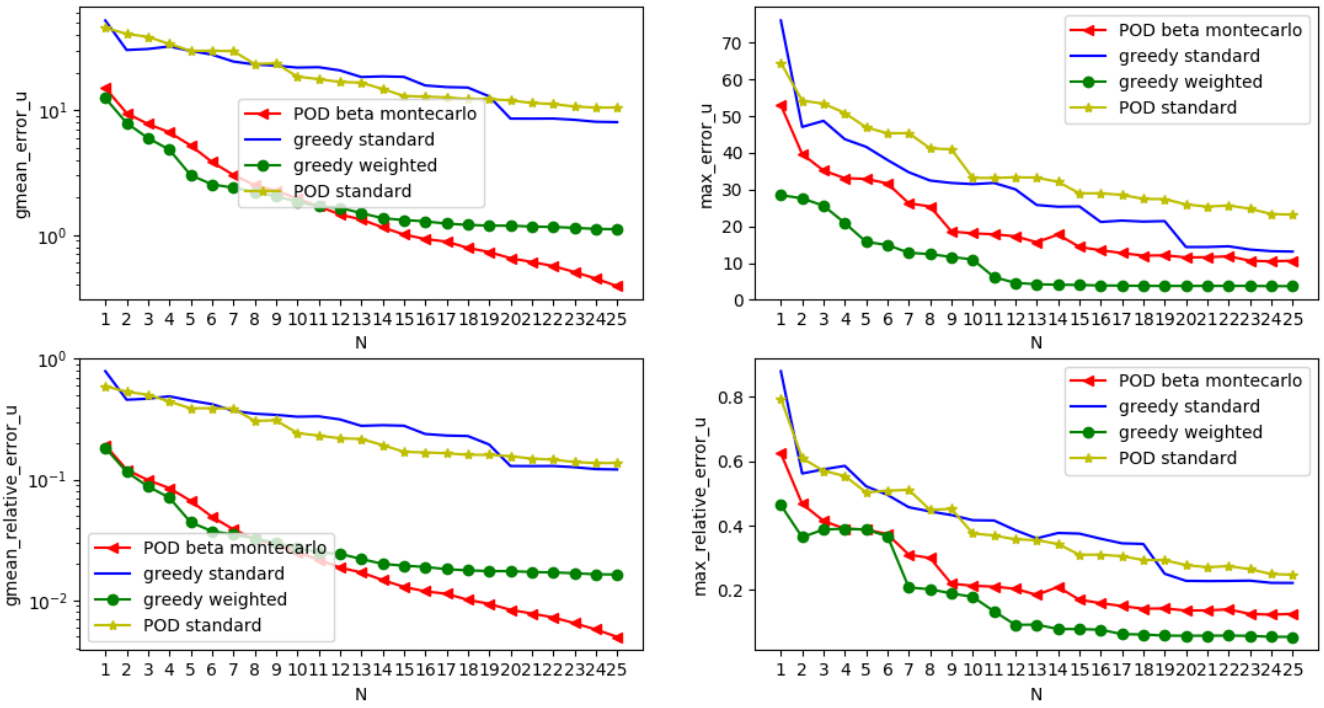

Figure 6.10: Stokes, first experiment: standard greedy, weighted greedy, standard *POD* and weighted *POD*, with a $Beta(20, 1)$.

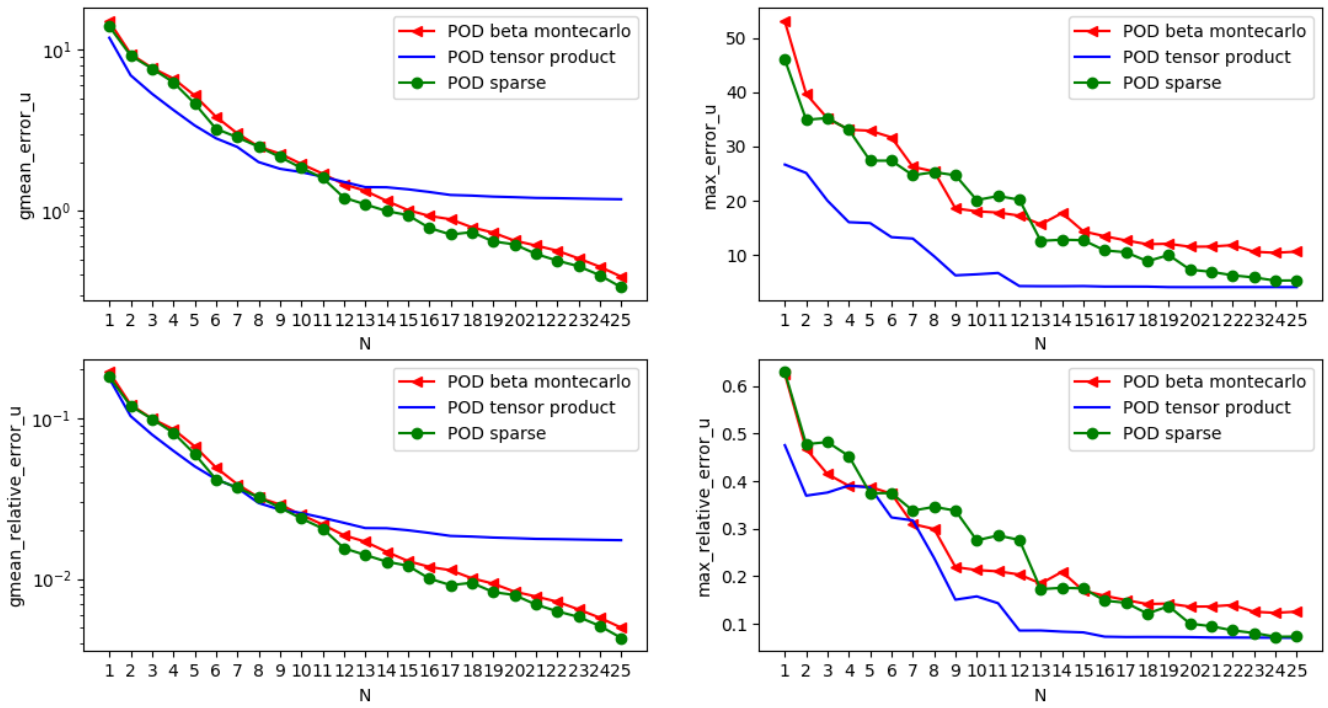

Figure 6.11: Stokes, second experiment: Monte-Carlo *POD*, Tensor product *POD* with a Gauss-Jacobi quadrature rule, Sparse rule *POD*, with a $Beta(20, 1)$.

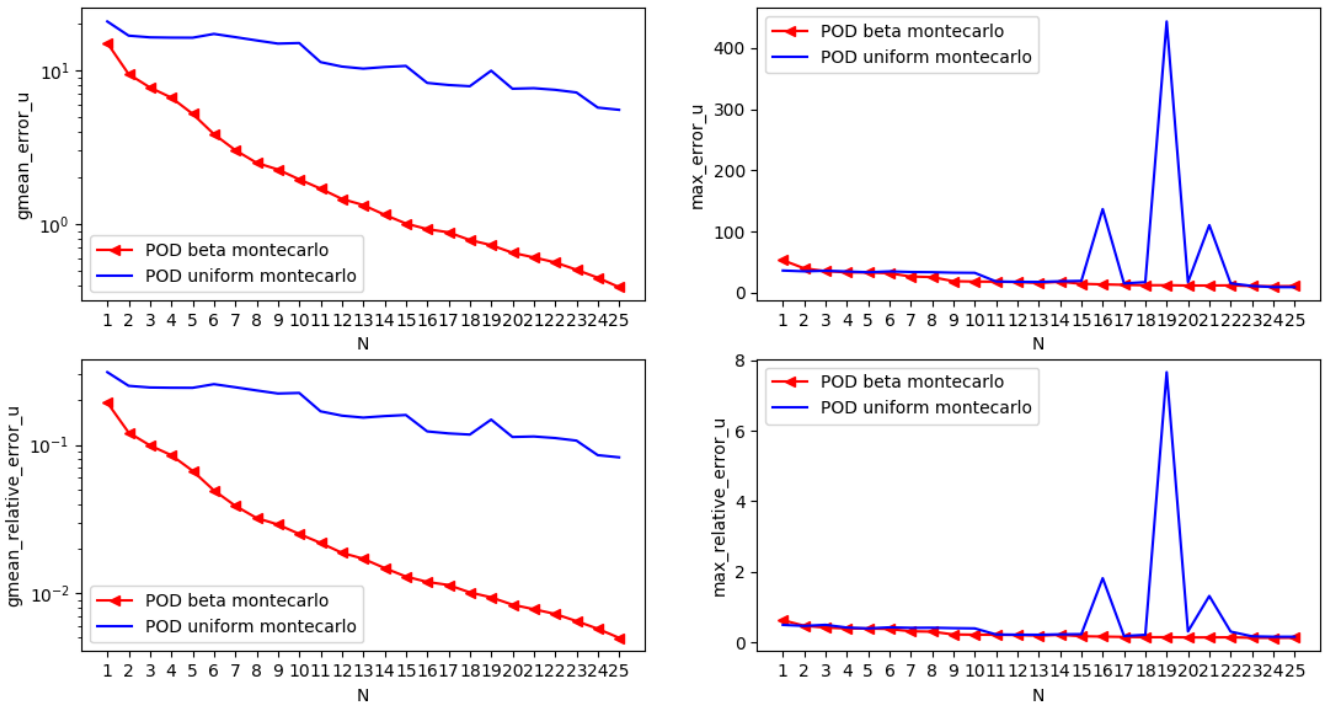

Figure 6.12: Stokes, third experiment: Monte-Carlo *POD* and Uniform Monte-Carlo *POD*, with a $Beta(10, 10)$.

6.1.4 Stokes: Beta 75 75

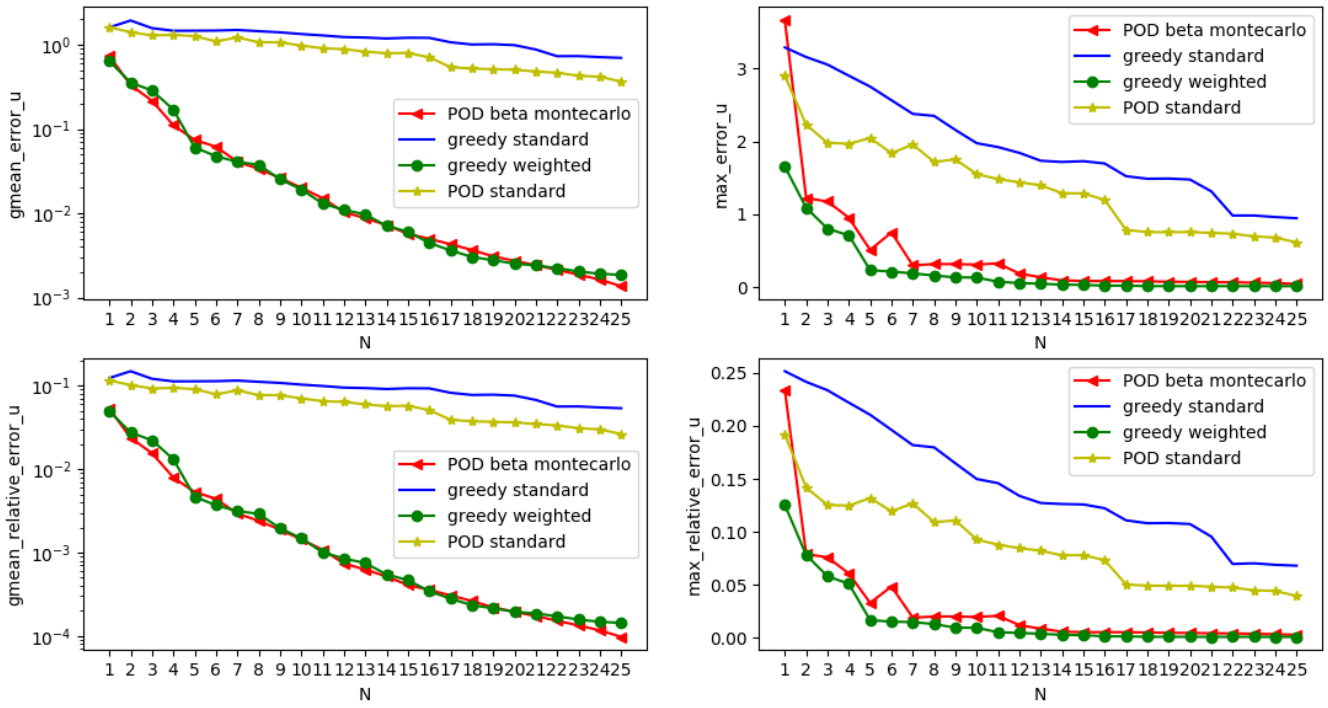

Figure 6.13: Stokes, first experiment: standard greedy, weighted greedy, standard *POD* and weighted *POD*, with a $Beta(75, 75)$.

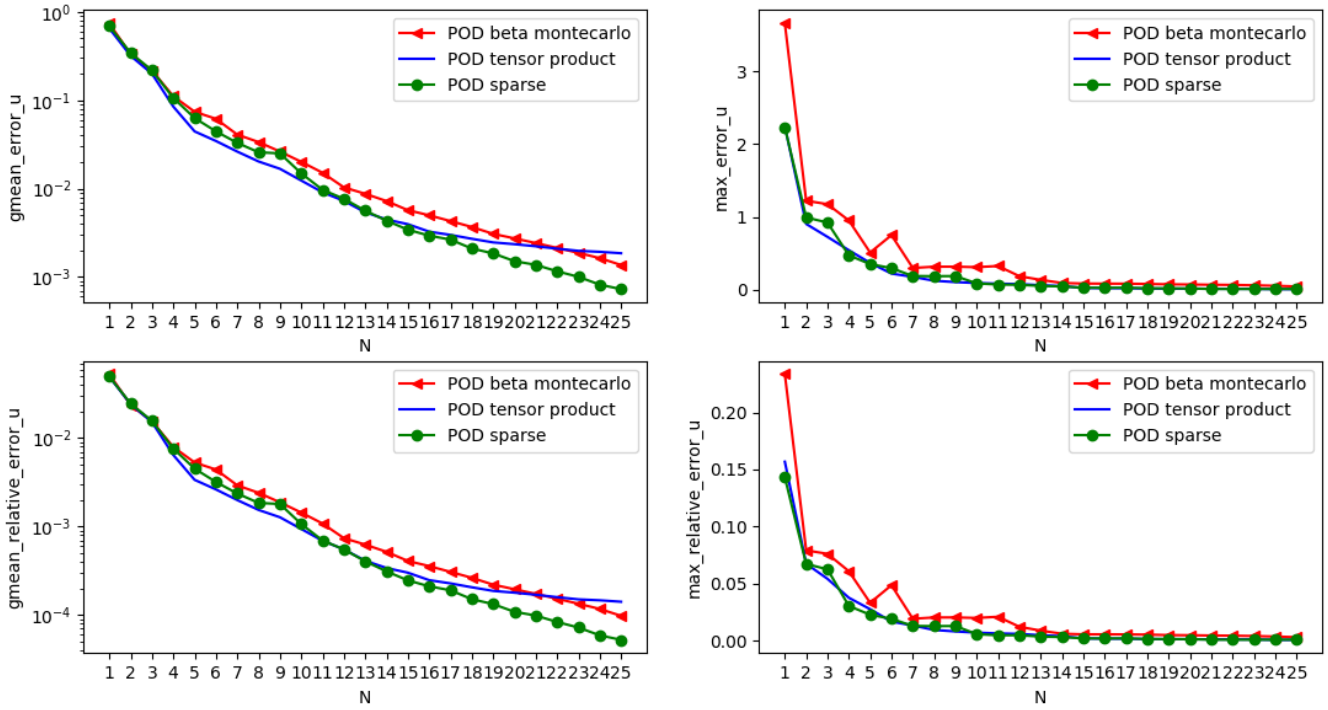

Figure 6.14: Stokes, second experiment: Monte-Carlo *POD*, Tensor product *POD* with a Gauss-Jacobi quadrature rule, Sparse rule *POD*, with a $Beta(75, 75)$.

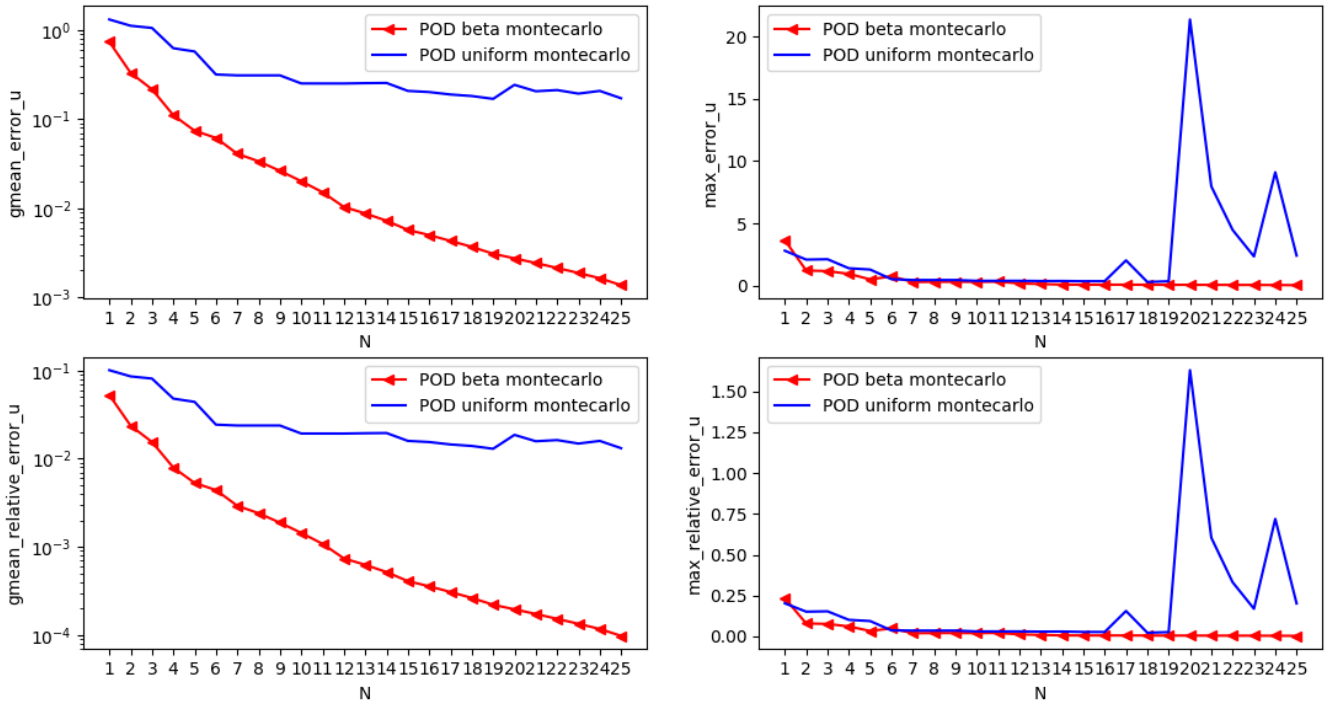

Figure 6.15: Stokes, third experiment: Monte-Carlo *POD* and Uniform Monte-Carlo *POD*, with a $Beta(75, 75)$.

6.1.5 Discussion for the Stokes problem

As we can see in the first experiment, the two weighted algorithms work better that the standard ones and these ones have no big differences except for the case of $Beta(20, 1)$, figure 6.10, where the weighted greedy has some problems but without any apparent reason.
For what concernes the second experiment we can see that the Monte-Carlo works very well. The other two approaches do a good job except for the case of $Beta(20, 1)$, figure 6.11, for the tensor product and the $Beta(0.03, 0.03)$, figure 6.5, for the sparse rule where we probably need more parameters because the distribution is concentrated in two zones.
Finally, even though it might seem strange, the Monte-Carlo method, taking parameters from the $Beta$ distribution but without weights, works better with respect to the Uniform Monte-Carlo method that takes parameters from a uniform distribution and after put some weights as we can see in 6.6, 6.9, 6.12, 6.15.
The fact can be clearer thinking that the two approaches converge to the same result only when we take an infinite number of parameters. In the experiments this is not possible and the fact that we take few parameters from the distribution and we do not weight them is more clever that taking the parameters randomly, with some of which that could be with a low probability, and subsequently weight them.

6.2 Navier-Stokes problem

In this section we will present the numerical experiments for the Navier-Stokes problem as we have explained above for the case of parameters coming from a $Beta$ distribution with (α, β) equal to: $(0.03, 0.03), (10, 10), (20, 1), (75, 75)$.
We can see in the figure 6.16 an example of simulation with a reduced basis.
If we compare with the Stokes case in figure 6.3 we can see two completely different physical behaviours. In this case we have a separation zone related to the convective term.

6.2.1 Navier-Stokes: Beta 0.03 0.03

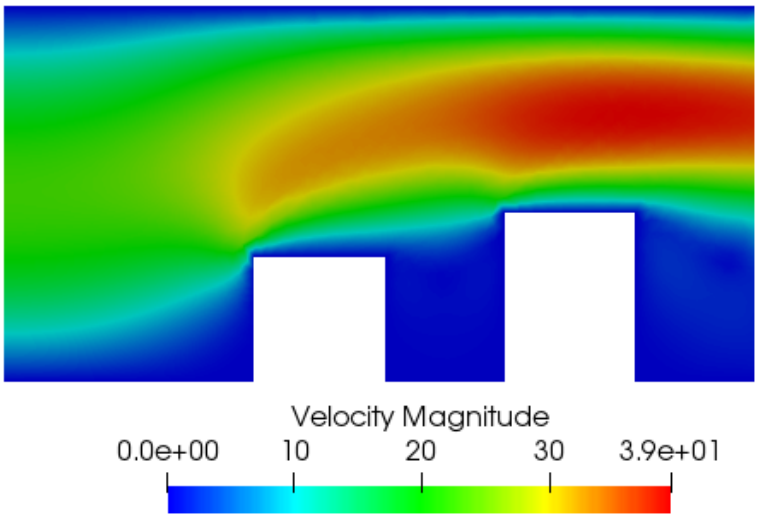

Figure 6.16: Navier-Stokes simulation with parameter $(1.05, 1.0, 1.05, 1.35, 10.0)$: velocity magnitude profile.

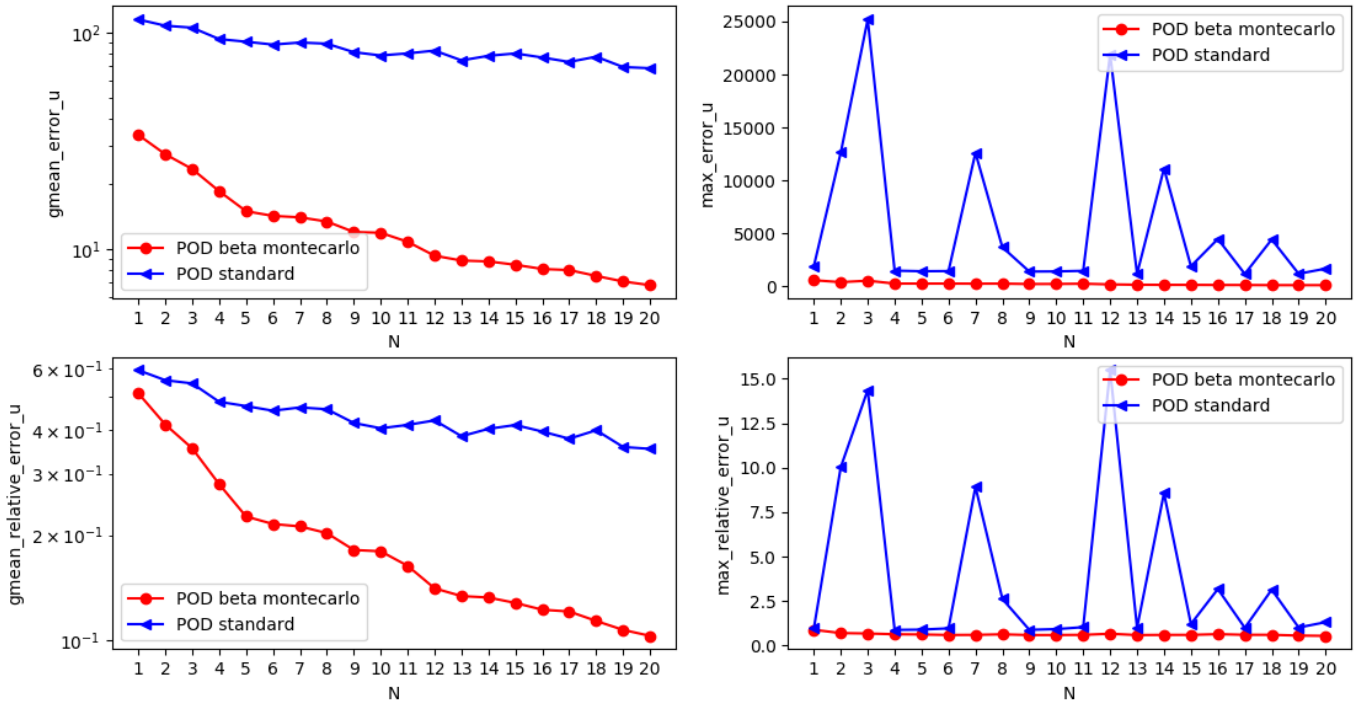

Figure 6.17: Navier-Stokes, first experiment: standard *POD* and weighted Monte-Carlo *POD*, with a $Beta(0.03, 0.03)$.

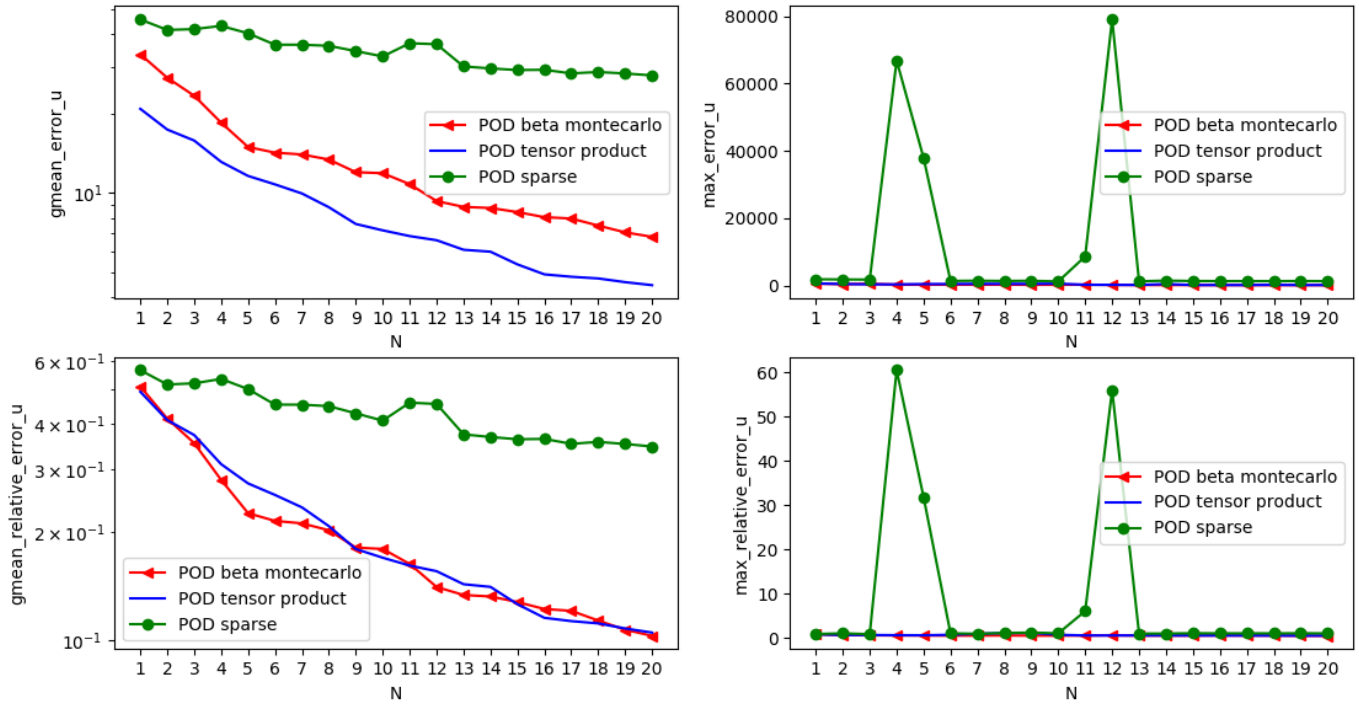

Figure 6.18: Navier-Stokes, second experiment: Monte-Carlo *POD* and tensor product rule *POD* with a Gauss-Jacobi quadrature rule, sparse rule *POD*,with a $Beta(0.03, 0.03)$.

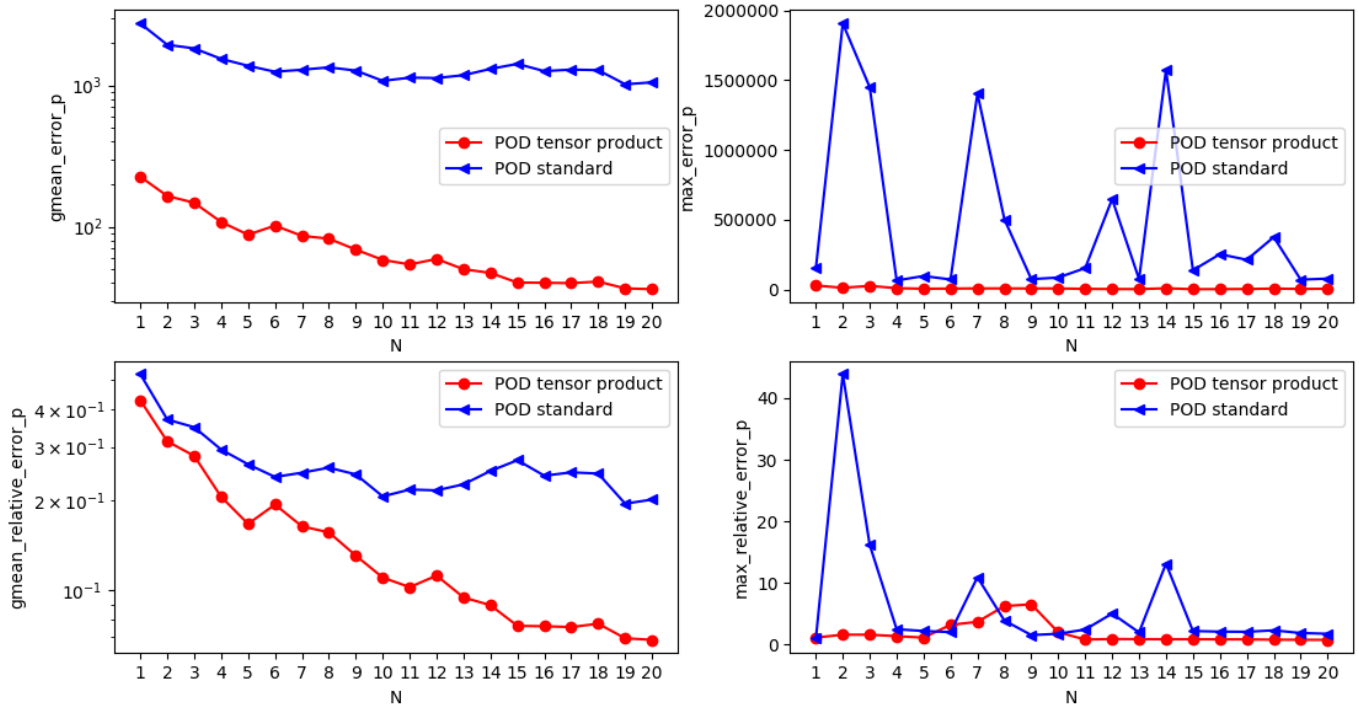

Figure 6.19: Navier-Stokes, third experiment: standard *POD* and weighted *POD* with tensor product rule with a $Beta(0.03, 0.03)$.

6.2.2 Navier-Stokes: Beta 10 10

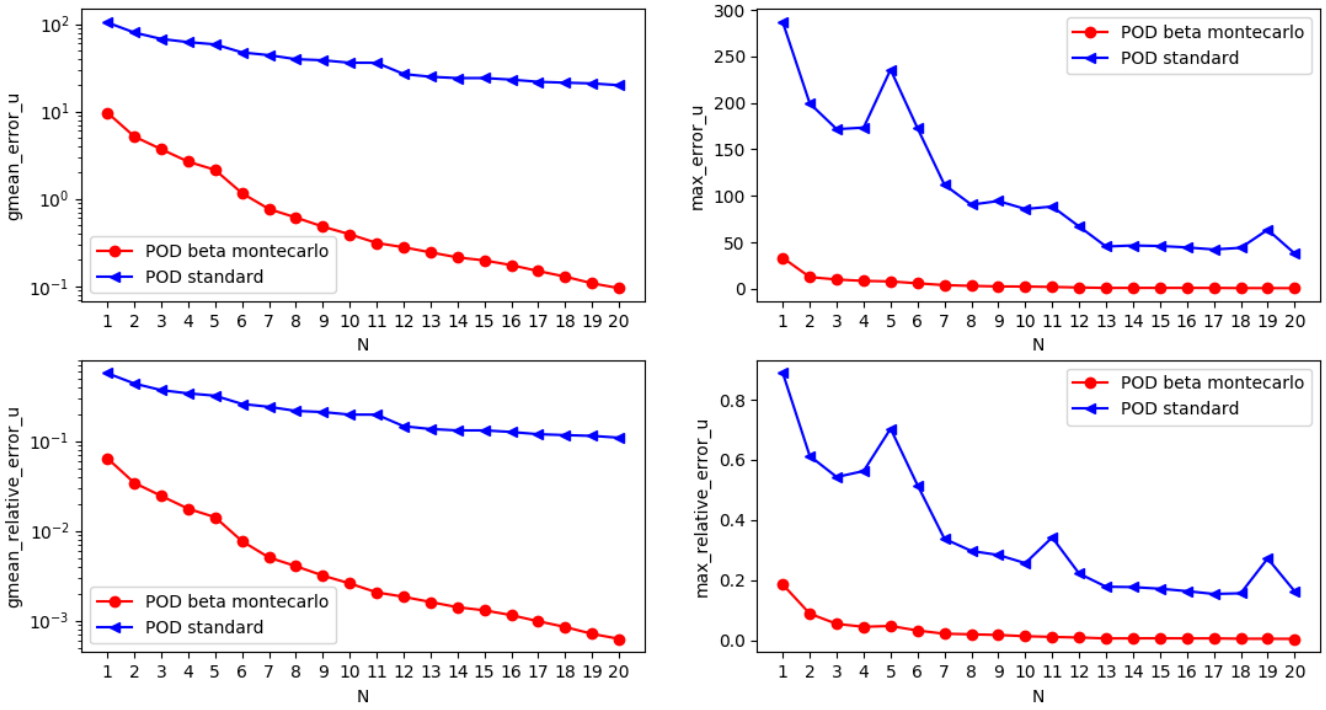

Figure 6.20: Navier-Stokes, first experiment: standard *POD* and weighted Monte-Carlo *POD*, with a $Beta(10, 10)$.

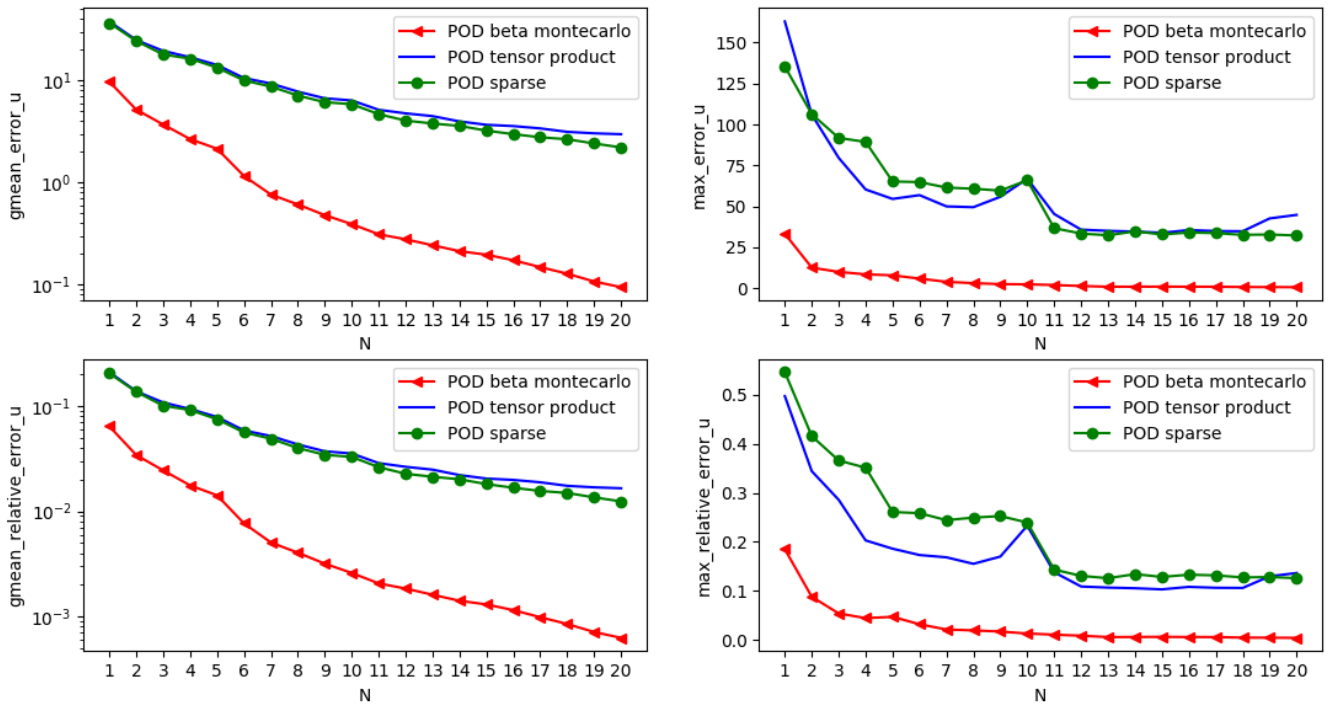

Figure 6.21: Navier-Stokes, second experiment: Monte-Carlo *POD* and tensor product rule *POD* with a Gauss-Jacobi quadrature rule, sparse rule *POD*,with a $Beta(10, 10)$.

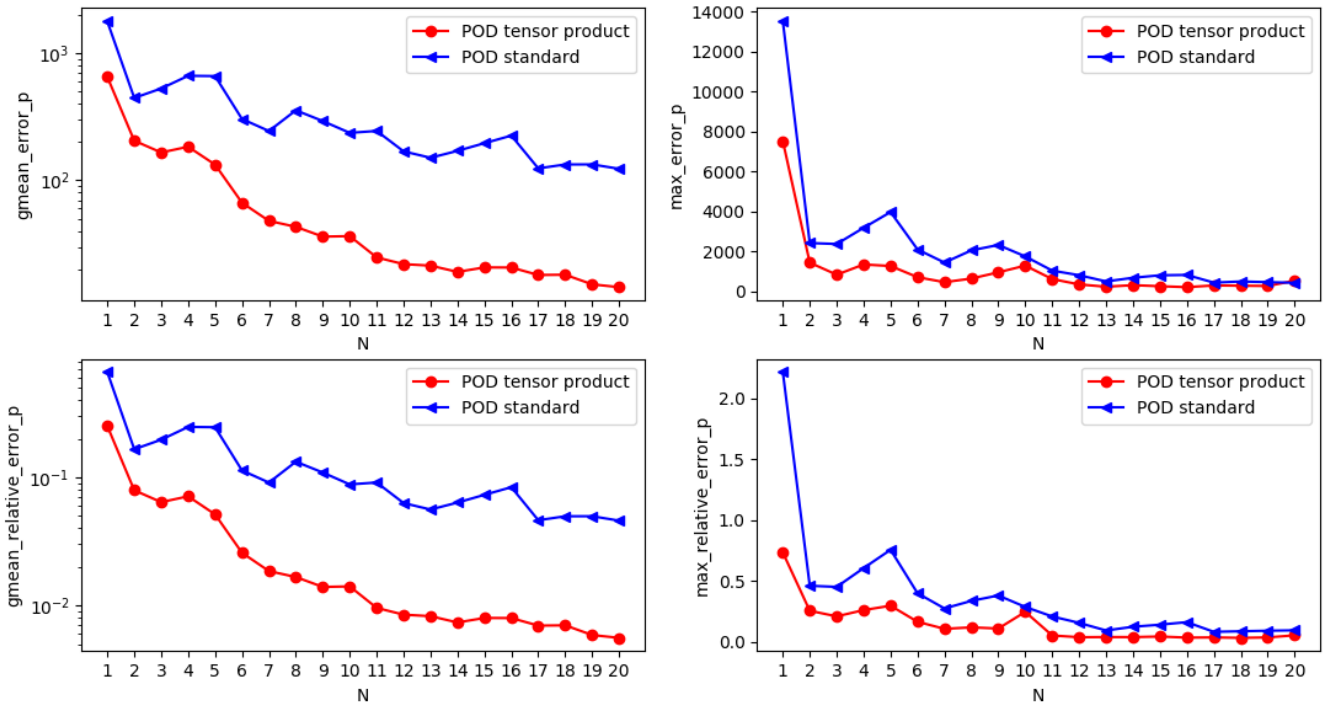

Figure 6.22: Navier-Stokes, third experiment: standard *POD* and weighted *POD* with tensor product rule with a $Beta(10, 10)$.

6.2.3 Navier-Stokes: Beta 20 1

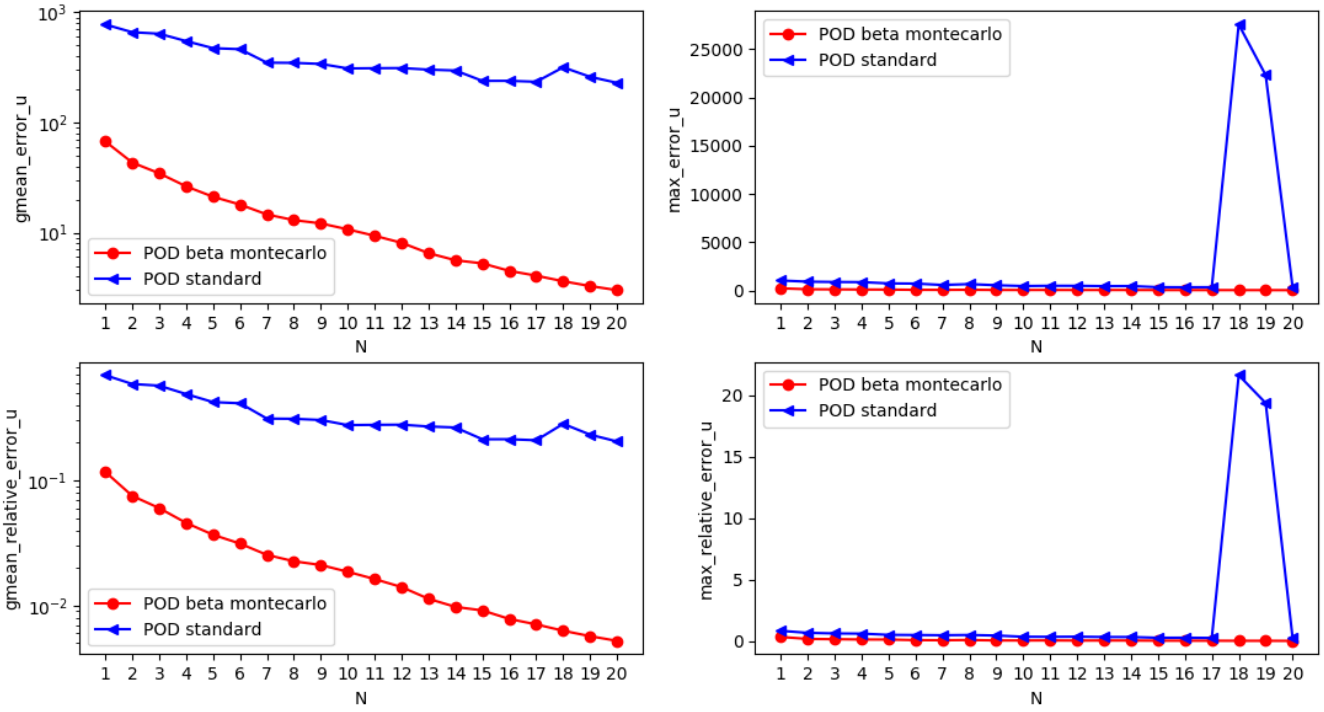

Figure 6.23: Navier-Stokes, first experiment: standard *POD* and weighted Monte-Carlo *POD*, with a $Beta(20, 1)$.

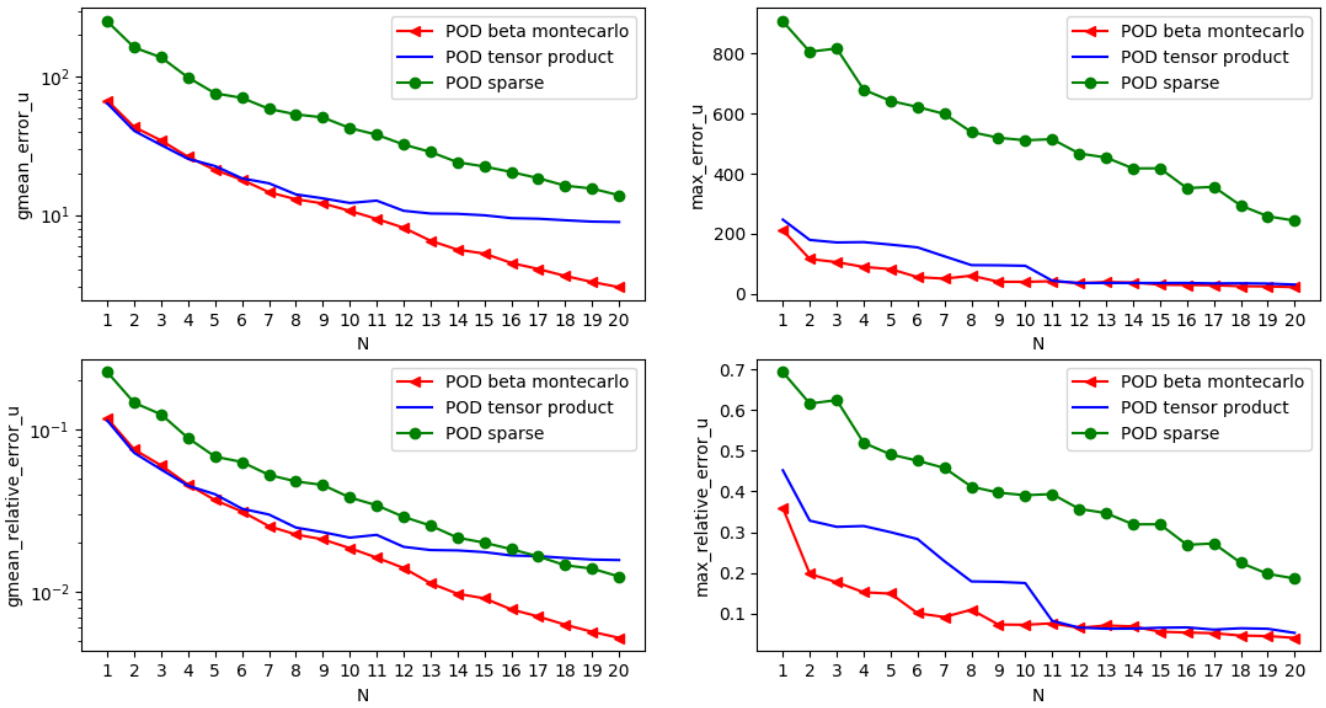

Figure 6.24: Navier-Stokes, second experiment: Monte-Carlo *POD* and tensor product rule *POD* with a Gauss-Jacobi quadrature rule, sparse rule *POD*,with a $Beta(20, 1)$.

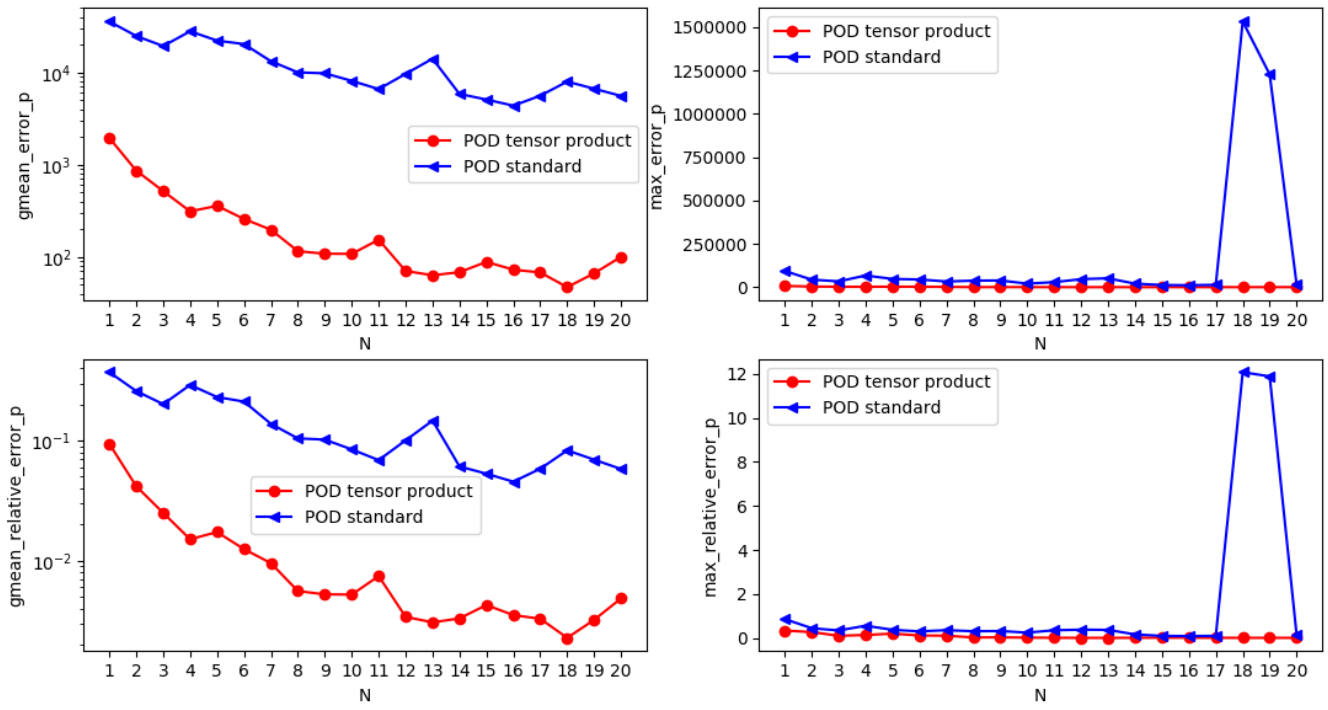

Figure 6.25: Navier-Stokes, third experiment: standard *POD* and weighted *POD* with tensor product rule with a $Beta(20, 1)$.

6.2.4 Navier-Stokes: Beta 75 75

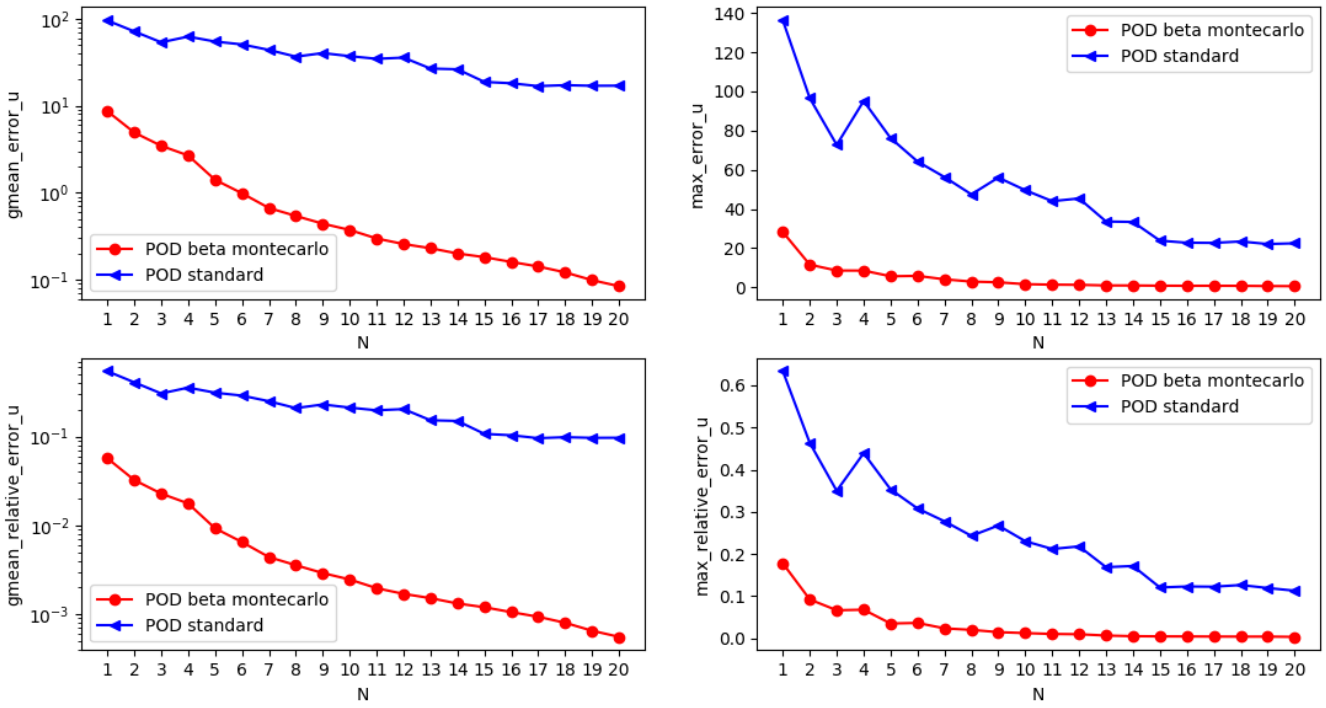

Figure 6.26: Navier-Stokes, first experiment: standard *POD* and weighted Monte-Carlo *POD*, with a $Beta(75, 75)$.

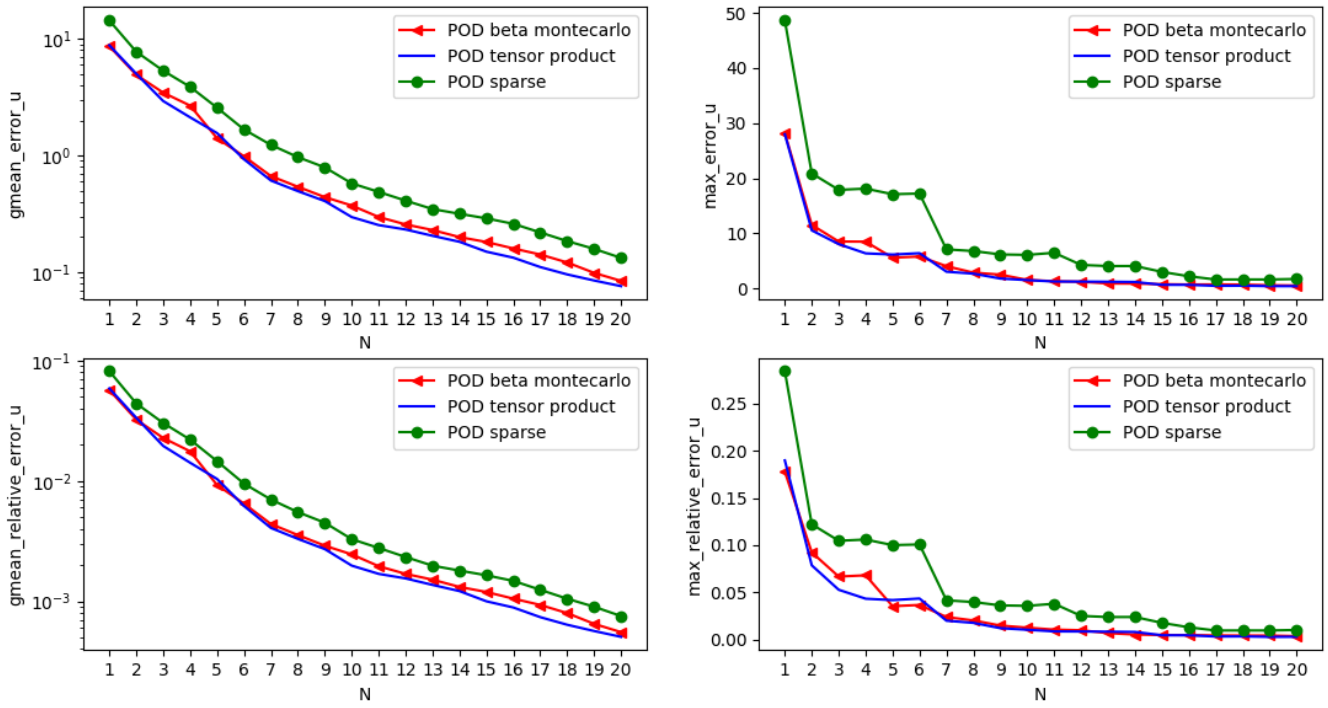

Figure 6.27: Navier-Stokes, second experiment: Monte-Carlo *POD* and tensor product rule *POD* with a Gauss-Jacobi quadrature rule, sparse rule *POD*,with a $Beta(75, 75)$.

6.2.5 Discussion for the Navier-Stokes problem

Also in this case the weighted approach works better than the standard one as we can see in the first experiment for all the distributions. The standard one sometimes seems to have big troubles very clear in the maximum error plot for $Beta(20, 1)$ when we see the first and the third experiment, figures 6.23 and 6.25.
For what concernes the second experiment we have similar results to the Stokes case. In fact the sparse rule has some problems for $Beta(0.03, 0.03)$, figure 6.18, due probably to the fact we have few parameters. In the other case the Monte-Carlo is the best approach followed by the sparse one, sometimes similar to the tensor product. But in all the cases we have a good H^1 error and maximum error.

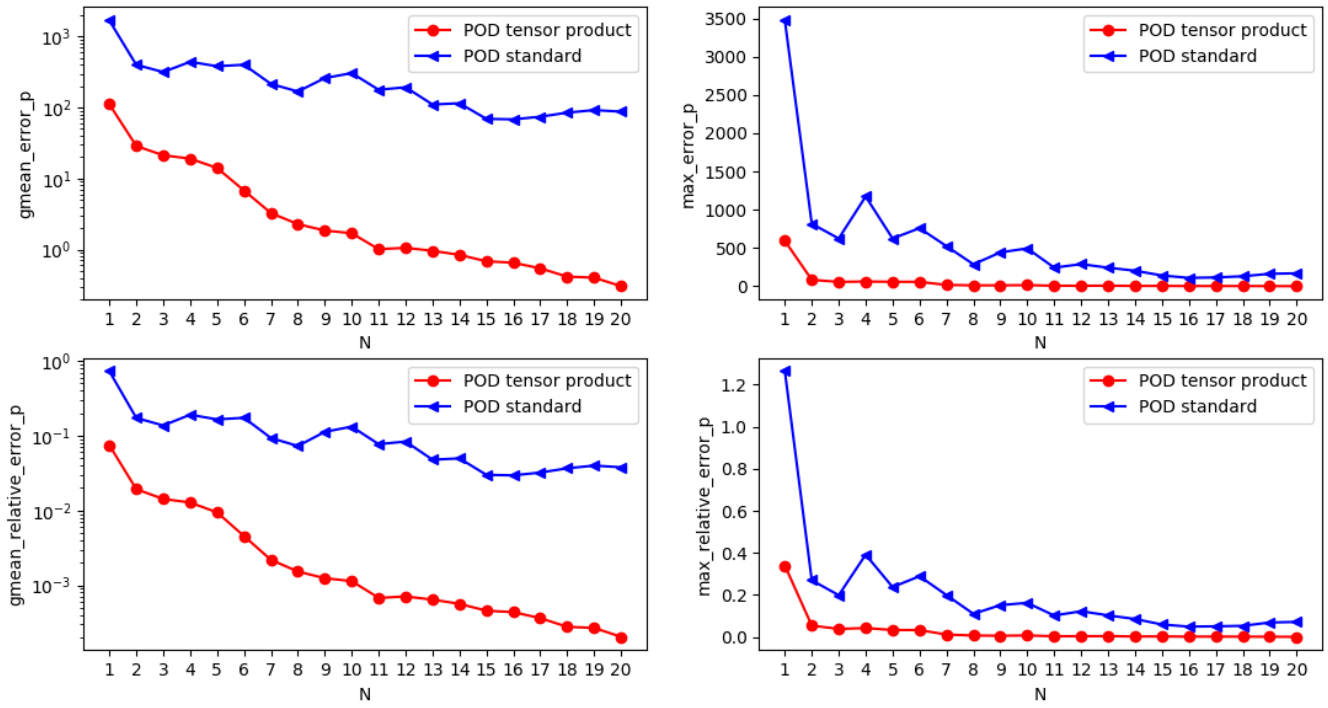

Figure 6.28: Navier-Stokes, third experiment: standard *POD* and weighted *POD* with tensor product rule with a $Beta(75, 75)$.

6.3 Conclusions

In these numerical experiments we have obtained similar results for the Stokes case and the Navier-Stokes one. As expected, we have seen that the weighted algorithms work better than the standard ones, in particular for the case of concentrated probability distributions, e.g. the $Beta(75, 75)$.
We also have verified that the sparse Smolyak rule is reliable to avoid the curse of dimensionality and to obtain good results for the numerical integration, but in the cases where the distribution is concentrated in more than one zone we need more parameters.
The tensor product rule does not give particular better result than the Monte-Carlo one but it is more complicated to implement, so we think it is not a good solution for our problem.
Finally studying the results of the two versions of Monte-Carlo approximation, we have seen that the one that samples from the *Beta* distribution is better.
In general this version is the best solution with respect to all the other possibilities when we do not

have too many parameters.

Chapter 7

Conclusions and future perspectives

In this master thesis we have presented the Stokes and the Navier-Stokes problem in the steady case, formulating the problem both for the deterministic and the stochastic case, introducing first the strong formulation of the equations and later the weak one. Subsequently we have introduced a Galerkin approximation with the associated algebraic system.
We have explained the two algorithms used in the reduced framework: the greedy and the POD, both in the standard and the weighted approach.
We have finally done some simulations of the same problem taking parameters coming from a *Beta* distribution of different values α and β.
In our experiments we have used several methods to search the reduced solution such as the standard and weighted greedy, the standard and the weighted *POD* using in this last case different variants: Monte-Carlo method, sampling from a distribution different from the uniform one; Uniform Monte-Carlo, taking the parameters from a uniform distribution and after weighting them according to the probability distribution; Tensor product and Smolyak sparse rule using a Gauss-Jacobi rule: in this case we follow an approach related to the numerical integration.
We firstly note that when we are working with a parameter that comes from a probability distribution a standard approach gives us a poor approximation if we use not too many basis like in our cases. The weighted one, on the contrary, with few basis can result in a good approximation and so it is really better for obtaining a reduced solution with a lower computational cost.
In all the simulations we have seen that the best choice is the Monte-Carlo method sampling from the distribution of interest.
On the contrary the Uniform Monte-Carlo method is not really good even though it is a similar tecnique to the previous one. So we understand that the sample where we take the parameters is very important, probably more important of using weights on the parameters.
All other cases can give us a good approximation of the solutions. In the Stokes case, in which we know the posterior error bound, the weighted greedy is a good method for almost all the distributions except one. The tensor product is quite good but in our opinion, not giving better results with respect to the Monte-Carlo tecnique and being more complicated to implement, it is not interesting

to use it in other applications. On the contrary the Smolyak grid gives us good results. Infact considering that sometimes is the only possible method when we have a huge space of parameters, these graphs that we have seen can reassure us that this rule can be one of the possible solutions when we have the problem of the curse of dimensionality. However we have to pay attention in the case of a distribution that is not concentrated in only one zone: in such a situation we need more parameters for obtaining a good approximation.
For possible future investigations we think that the study of other non-linear stochastic problems can be interesting, as for the case of a non-linear elastic beam [42] or the nonlinear Schrödinger equation [43], perhaps inserting a lot of parameters for better noting the issue of the curse of dimensionality.
We also hope to see the application of the Smolyak rule to industrial problems where there are usually a lot of uncertain parameters and so where this method could deal with.
It would be also interesting to find a weighted posterior estimator for the Navier-Stokes case in alternative to the *POD* approach.
Some topics are not be treated such as a the non-affine case where we need an empirical interpolation method and for doing this in the Stokes case it could be followed this article: [44].
We finally think that other types of sparse grids can be implemented, as described in [23] or with a more general approach in [34].

Appendix A

Mathematical preliminaries

In this appendix we want to remember some concepts and notations that we will use in the next chapters. We refer to [1] and [2] for further details.
Let us take a normed space V. We define a *linear form* as an application $F : V \to \mathbb{R}$ such that:

$$F(u+v) = F(u) + F(v), \quad \forall u, v \in V,$$
$$F(\alpha v) = \alpha F(v), \quad \forall \alpha \in \mathbb{R}, \forall v \in V.$$

We can write $F(v)$ also with the notation $\langle F, v \rangle$.
Usually we will take a *bounded* linear form, i.e. such that:

$$|F(v)| \leq C||v||_V, \quad \forall v \in V. \tag{A.1}$$

We define *the dual of* V and we denote it with V' the set:

$$V' := \{F : V \to \mathbb{R} \text{ such that } F \text{ is linear and bounded}\}, \tag{A.2}$$

equipped with the norm:

$$||F||_{V'} = \sup_{v \in V \setminus \{0\}} \frac{|F(v)|}{||v||_V}. \tag{A.3}$$

We have a central theorem that we will use in the reduced method:

Theorem 10. *(Riesz rappresentation theorem). Let H be a Hilbert space with a scalar product $(\cdot, \cdot)_H$. For each linear and bounded form $F \in H'$ it exists a single element $x_F \in H$ such that:*

$$F(y) = (y, x_F)_H \quad \forall y \in H \text{ and } ||f||_{H'} = ||x_F||_H. \tag{A.4}$$

If we take two normed spaces V and Q we can define a *bilinear form* as an application $a : V \times Q \to \mathbb{R}$ such that:

$$a(\alpha u + \beta v, w) = \alpha a(u, w) + \beta a(v, w), \quad \forall \alpha, \beta \in \mathbb{R}, \forall u, v, w \in V,$$
$$a(u, \alpha v + \beta w) = \alpha a(u, v) + \beta a(u, w), \quad \forall \alpha, \beta \in \mathbb{R}, \forall u, v, w \in V.$$

Now let us pass to the Sobolev spaces.
We take an open domain $\Omega \subset \mathbb{R}^d$ and a positive integer k. We denote with $L^2(\Omega)$ the space:

$$L^2(\Omega) := \{f : \Omega \to \mathbb{R} \text{ such that } \int_\Omega f^2 \, d\Omega < \infty\}. \tag{A.5}$$

The *Sobolev space of order k* is defined as:

$$H^k(\Omega) = \{f \in L^2(\Omega) \text{ such that } D^\alpha f \in L^2(\Omega), |\alpha| \leq k\}, \tag{A.6}$$

where:

$$D^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}, \tag{A.7}$$

with the partial derivate intended in the sense of the distributions and where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_d) \in \mathbb{N}^d$ with $|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_d$.
By definition we have $H^{k+1}(\Omega) \subset H^k(\Omega)$ and $H^0(\Omega) = L^2(\Omega)$.
It holds that $H^k(\Omega)$ is a Hilbert space with the scalar product:

$$(f, g)_{H^k(\Omega)} = \sum_{\alpha \in \mathbb{N}^k, |\alpha| \leq k} \int_\Omega (D^\alpha f)(D^\alpha g) \, d\Omega, \tag{A.8}$$

that induces the norm:

$$||f||_{H^k(\Omega)} = \sqrt{(f, f)_{H^k(\Omega)}} = \sqrt{\sum_{\alpha \in \mathbb{N}^d, |\alpha| \leq k} \int_\Omega |D^\alpha f|^2 \, d\Omega}. \tag{A.9}$$

We introduce also the seminorm because we will use it in the numerical experiments:

$$|f|_{H^k(\Omega)} = \sqrt{\sum_{|\alpha| = k} \int_\Omega (D^\alpha f)^2 \, d\Omega}. \tag{A.10}$$

Bibliography

[1] H. Brezis: *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer, 2010.

[2] B. Rynne, M.A. Youngson: *Linear Functional Analysis*. Springer, 2000.

[3] *RBniCS-reduced order modelling in FEniCS*. http://mathlab.sissa.it/rbnics, 2015.

[4] M. R. Spiegel, S. Lipschutz, D. Spellman: *Vector Analysis. Schaum's Outlines*. USA: McGraw Hill (2nd ed.), 2009.

[5] F. Brezzi: *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers*. R.A.I.R.O. Anal. Numér. 8: 129-151, 1974.

[6] W. Rudin: *Real and Complex Analysis*. McGraw-Hill, 1966.

[7] A. Quarteroni, A. Valli: *Numerical approximation of partial differential equations*. Springer-Verlag Italia, Milano, 2013.

[8] Jan S.Hesthaven, G. Rozza, B. Stamm: *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[9] G.P. Galdi: *An introduction to the mathematical theory of the Navier-Stokes equations, Linearized Steady Problem*. Springer, 2011.

[10] M. Fortin: *An analysis of the convergence of mixedfinite element methods*. RAIRO. Analyse numérique, tome 11, n. 4, p. 341-354, 1977.

[11] A. N. Brooks, T. J. R. Hughes: *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*. Comput. Methods Appl. Mech. Engrg., 32, 199–259, 1982

[12] J. Xu, L. Zikatanov: *Some observations on Babuška and Brezzi theories*. Numer. Math. 94: 195-202, 2003.

[13] G. Rozza, K. Veroy: *On the stability of the reduced basis method for Stokes equations in parametrized domains*. ScienceDirect, 196, 1244-1260, 2007.

[14] G. Rozza, D.B.P. Huynh, A. Manzoni: *Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants.* A. Numer. Math. 125: 115. https://doi.org/10.1007/s00211-013-0534-8, 2013.

[15] A. Quarteroni, G. Rozza, A. Manzoni: *Certified reduced basis approximation for parametrized PDE and applications.* J. Math Ind. **3**, 2011.

[16] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza: *Supremizer stabilitation of POD-Galerkin approximation of parametrized steady Navier-Stokes equations.* Int. J. Numer. Meth. Engng; **102**:1136:1161, 2015.

[17] A. Gerner, K. Veroy: *Reduced basis a posteriori error bounds for the stokes equations in parametrized domains: a penalty approach.* Math. Mod. and Meth. in Appl. Sc., 2010.

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

[19] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza: *A reduced computational and geometrical framework for inverse problems in haemodynamics.* International journal for numerical methods in biomedical engineering, vol. 29, 2013.

[20] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera: *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations.* C.R. Math. **339**, 667-672, 2004.

[21] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera: *Efficient reduced-basis treatment of non-affine and nonlinear partial differential equations.* ESAIM Math. Model. Numer. Anal. **41**, 575-605, 2007.

[22] A. Quarteroni, G. Rozza: *Numerical solution of parametrized Navier-Stokes equations by reduced basis methods.* Numerical methods for partial differential equations, **23**(4):923-948, 2007.

[23] T.J. Sullivan: *Introduction to Uncertainty Quantification.* Springer International Publishing Switzerland, 2015.

[24] L. Venturi: *Weighted reduced order methods for parametrized PDEs in uncertainty quantification problem.* Master thesis at SISSA, 2015/2016.

[25] P. Chen, A. Quarteroni, G. Rozza: *A weighted reduced basis method for elliptic partial differential equations with random input data.* SIAM Journal on Numerical Analysis, Vol. 51, No. 6 : pp. 3163-3185, 2013.

[26] P. Chen, A. Quarteroni, G. Rozza: *Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations.* Numerische Mathematik, vol. 133, pg. 67–102, 2016.

[27] L. Venturi, F. Ballarin, G. Rozza: *A weighted POD method for elliptic PDEs with random inputs*. Journal of scientific computing, 81, 2019.

[28] L. Venturi, D. Torlo, F. Ballarin, G. Rozza: *Weighted reduced order methods for parametrized partial differential equations with random inputs*. Uncertainty Modeling for Engineering Applications, Springer International Publishing, pg. 27-40, 2019.

[29] P. Chen, A. Quarteroni, G. Rozza: *Comparison between reduced basis and stochastic collocation methods for elliptic problems*. Journal of Scientific Computing, 59(1), p. pp. 187–216, 2014.

[30] P. Chen, A. Quarteroni, G. Rozza: *Reduced basis methods for uncertainty quantification*. SIAM/ASA Journal on Uncertainty Quantification, 5, p. pp. 813–869, 2017.

[31] D. Torlo, F. Ballarin, G. Rozza, *Stabilized Weighted Reduced Basis Methods for Parametrized Advection Dominated Problems with Random Inputs*. SIAM/ASA Journal on Uncertainty Quantification, 6(4), pp. 1475-1502, 2018.

[32] M. Heikenschloss, B. Kramer, T. Takhtaganov: *Adaptive reduced-order model construction for conditional value-at-risk estimation*. ACDL Technical Report, 2019.

[33] B. Oksendal: *Stochastic Differential Equations. An introduction with applications*. Springer-Verlag, 1998.

[34] V. Kaarnioja: *Smolyak quadrature*. Master's thesis, University of Helsinki, 2013.

[35] A. Quarteroni, R. Sacco, F. Saleri: *Numerical mathematics*. Spriger, 2007.

[36] P. Davis, P. Rabinowitz: *Methods of Numerical Integration*. Academic Press, New York, 1975.

[37] V. Girault, P. Raviart: *Finite element methods for Navier-Stokes equations: theory and algorithms*. Springer Science & Business Media, 2012.

[38] S. Smolyak: *Quadrature and interpolation formulas for tensor products of certain classes of functions*. Soviet Mathematics, Vol. 4, pp. 240-243, Translation of Doklady Akademii Nauk SSSR, 1963.

[39] J. Von Neumann : *Various techniques used in connection with randomdigits.Proceedings of Symposium on "Monte Carlo Method"*. Los Angeles, chapter 13, 1951.

[40] MS. Engelman, G. Strang, KJ. Bathe: *The application of quasi-Newton methods in fluid mechanics*. International Journal for Numerical Methods in Engineering, **17**(5):707-718, 1981.

[41] D.J. Tritton: *Physical Fluid Dynamics*. Oxford Science Pubblication, 1988.

[42] A. Pielorz: *On nonlinear equations of a beam*. In: Refined Dynamical Theories of Beams, Plates and Shells and Their Applications. Lecture Notes in Engineering, vol 28. Springer, Berlin, Heidelberg, 1987.

[43] G. Fibich: *The nonlinear Schrödinger equation*. Spriger, 2015.

[44] P. Chen, A. Quarteroni, G. Rozza: *A weighted empirical interpolation method: A priori convergence analysis and applications*. ESAIM: Mathematical Modelling and Numerical Analysis, 48(4), p. pp. 943–953, 2014.

Be the change you wish to see in the world