# Variations and further developments on the Non-Intrusive Reduced Basis two-grid method

Elise Grosjean

Elise Grosjean. Variations and further developments on the Non-Intrusive Reduced Basis two-grid method. Numerical Analysis [cs.NA]. Sorbonne Université, 2022. English. NNT : 2022SORUS019 . tel-03715386

SORBONNE
UNIVERSITÉ

# THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité

**Mathématiques appliquées**

École doctorale de Sciences Mathématiques (Paris-Centre ED386)
Laboratoire Jacques-Louis Lions (UMR 7598)

Présentée par

**Elise GROSJEAN**

Pour obtenir le grade de

**DOCTEURE de SORBONNE UNIVERSITÉ**

Sujet de la thèse :

# Variations and further developments on the Non-Intrusive Reduced Basis two-grid method

soutenue le mardi 8 Mars 2022

devant le jury composé de :

| | |
|---|---|
| Mme Nora Aïssiouene | Examinatrice |
| Mme Rachida Chakir | Examinatrice |
| M. Albert Cohen | Examinateur |
| M. Yvon Maday | Directeur de thèse |
| M. Andrea Manzoni | Rapporteur |
| Mme Olga Mula | Examinatrice |
| M. Christophe Prud'homme | Rapporteur |

# Contents

## Remerciements

Ces années de recherche m'ont beaucoup apporté, et j'aimerais témoigner de ma reconnaissances aux personnes ayant contribué à cette belle aventure. Tout d'abord, je tiens à remercier mon directeur de thèse Yvon Maday, du Laboratoire Jacques-Louis Lions (LJLL). Grâce à ta confiance et à ton foisonnement d'idées (à l'instar de ton bureau), j'ai bénéficié d'une collaboration avec plusieurs profils professionnels, des personnes que tu as rassemblées autour de ce projet, et c'était une chance de pouvoir travailler dans un contexte à la fois académique et industriel. J'ai beaucoup appris de ta capacité à cerner les enjeux industriels et à y répondre par des méthodes mathématiques adaptées. Tu m'as à la fois soutenue lorsque cela était nécessaire tout en me laissant une grande liberté, et je t'en suis très reconnaissante.

Je souhaite ensuite remercier chaleureusement Nora Aïssiouene, membre de mon jury, qui a toujours été très disponible, et qui a su m'encourager au début de la thèse. Tous les conseils que tu m'as donnés ainsi que ton temps m'ont été très précieux.

Andrea Manzoni et Christophe Prud'homme ont bien sûr toute ma gratitude pour avoir accepté de rapporter ma thèse, pour leur relecture attentive et l'intérêt porté à mes travaux de recherche. Vos remarques pertinentes et vos conseils m'ont été très utiles.

Un grand merci également à Rachida Chakir d'avoir pris part à ma soutenance en tant que membre de mon jury ainsi que pour sa participation sur le projet des erreurs a-posteriori. J'espère que nous allons continuer à travailler ensemble. Je suis également ravie qu'Olga Mula et Albert Cohen aient accepté de faire partie du jury. C'est un plaisir et un grand honneur de soutenir devant vous.

Je tiens également à citer ici toute l'équipe d'EDF avec qui j'ai travaillé, sans qui rien de tout cela n'aurait été possible. Merci d'avoir apporté la problématique des éoliennes, et le projet de la bibliothèque open-source MORDICUS. Grâce à cette collaboration, j'ai entre autre pu participer à une variété de discussions sur l'élaboration de ce dernier projet, celles-ci concernant tout aussi bien la mise en place d'un dictionnaire sur les méthodes de réduction de calculs que leurs implémentations. Ainsi, je souhaite en particulier nommer les personnes suivantes: Guilhem Ferte pour son enthousiasme et son implication dans tous les projets; Fabien Casenave pour son aide sur la bibliothèque MORDICUS et l'utilisation de Basictools; Denis Wendum et Guillaume Angot pour leurs temps concernant la mise en place des jeux de données sur les champs d'éoliennes; Alexei Mikchevitch, ainsi qu'Alexandra Martin Sanchez; Yvan Fournier pour l'aide précieuse sur code_saturne; et Jean-François Wald pour avoir pris le temps de reprendre mes scripts et mon travail. Cela m'a notamment permis d'avoir un retour sur mon travail à chaque étape de ma thèse.

En ce qui concerne le laboratoire Jacques-Louis Lions, je suis très reconnaissante à Frederic Hecht d'avoir toujours pris le temps de répondre à toutes mes questions sur FreeFem++. Un grand merci également à Khashayar pour toute la logistique informatique, ainsi qu'à Pierre-Henri pour l'installation de code_saturne sur le serveur de calculs Leto. En continuant dans le contexte de la logistique, je tiens à citer Salima, Malika et Catherine quant à l'organisation administrative.

Je remercie également Martin Vohralík de nous avoir expliqué les différentes approches possibles sur les erreurs a-posteriori, et je recommande également le polycopié de son cours.

Je suis très reconnaissante aux professeurs qui m'ont fait confiance et grâce à qui j'ai pu donner des cours de TD et de TP (Au LJLL, Martin Campos Pinto, Marie Postel, et Xavier Claeys; et à l'ENSAE, Sidi-Mahmoud Kaber). C'était une belle expérience enrichissante.

Je n'oublie pas au passage Bertrand Thierry pour sa gentillesse et les cours passionnants d'éléments finis qui m'ont confortée dans mon choix de sujet de thèse.

Ici suit un clin d'œil aux personnes qui ont partagé mon quotidien ces trois dernières années et je peux donc commencer par mon co-bureau Alex qui m'a recommandé d'écrire mes remerciements en avance, donc "mer-" (la suite vient plus tard).

Je rends évidemment hommage à mes deux chats: Lucky pour toute l'attention qu'elle m'a portée; Youpi, par contre, n'a eu de cesse d'essayer de me détouner de mon travail.
Au LJLL, mes premières louanges vont au bureau des légendes, un bureau comme nous n'avons pas tous les jours l'occasion d'en voir un (cf covid), et en particulier Gabriela, je suis très heureuse que cette expérience m'ait permis de te connaître et que nous soyons amies, tu as été un véritable soutien. Lydie, bien que n'étant pas au LJLL, tu as presque fait partie de ce bureau, et je projette de t'envoyer des oranges par milliers (Merci pour toutes tes cartes!). Alex, je t'approuve beaucoup, j'espère que tu continueras à annoncer de grandes et belles nouvelles toujours aussi bien... Christophe, c'est très aimable de partager ton bureau pour les temps de sieste. Je rends grâce à Amaury pour le record mondial en quantité de (succulent!) Tiramisu. Il m'est impossible d'oublier de citer Rémi, toujours paré pour des sorties sportives. Yipang, je te sais gré pour les séances de thé versus eau chaude. Je fais cas de Shengquan, Noémi (j'attends la revanche au fameux jeu "mito"), Willy et Léon.

Viens la suite pour mon co-bureau "-ci Alex". En dehors du bureau, beaucoup de personnes ont participé de près ou de loin à rendre cette expérience aussi plaisante, donc je m'excuse pour cette liste non-exhaustive.

Ainsi, j'affectionne beaucoup Lise, qu'elle le lise ou non. J'estime beaucoup Jules, Nany (merci pour l'escape game!), Giorgia, Luidi, Anouk et Olivier (peut-être nous recroiserons-nous par hasard sur un GR?), Maria, Marc (ici je ne fais pas allusion au café du labo que j'apprécie également), Katia, Valentin, Gontran, David, Fatima, Mohammad, Ludo, Alexandre P., Nicolas (le vampire), Nicolas A., Allen (alias Michael), Jesus, Emma, Gong, Poyi, ..., et toutes celles et ceux avec qui j'ai partagé des déjeuners ou des thés du labo, merci pour les moments conviviaux et chaleureux! La PPG avec Serge a été un moment de sport intense et gratifiant et il a été rendu possible grâce à Anne-Françoise, et le groupe de JF, Pierre et Jean (Jean, tu es cité avec le groupe PPG, bien que tu n'y soit allé qu'une fois je crois). Les jeux de société avec vous ont aussi la cote.
Je garderai aussi un très bon souvenir des soirées avec Driss, Lucas, Louis, Fadwa, Adrian, Felix, Agnès et David ...
Je tiens à remercier Florian et Eva pour avoir participé au clip sur les éoliennes.
Pour finir, la pandémie actuelle a favorisé le travail à distance, et aurait pu induire plus d'isolement si je n'avais croisé la route de mes voisins (et là je ne parle ni de Pierre ni de Margot), par hasard chez moi. Cyril et Naïla, merci pour votre gentillesse, je vais tenter de trouver un appartement pour vous dans mon prochain immeuble.
J'ai également une pensée pour mes amis du lycée, et en particulier Ines et Ying, deux exceptionnelles amitiés. Cela fait maintenant plus de 10 ans que vous me supportez et je vous félicite.

Je voudrais dire un immense merci à mes parents qui m'ont toujours soutenue et encouragée. Merci à vous, et à toute la famille.

Et bien sûr, pour conclure, ma gratitude incommensurable va à Camille. Merci pour toutes les belles absurdités passées, présentes, et à venir.

# Nomenclature

CFD: Computational Fluid Dynamics
EIM: Empirical Interpolation Method
FE: Finite Element
FEM: Finite Element Method
FV: Finite Volume
FVM: Finite Volume Method
HF: High-Fidelity
MOR: Model Order Reduction
NIRB: Non-Intrusive Reduced Basis
PCA: Principal Component Analysis
POD: Proper Orthogonal Decomposition
PODI: POD-Interpolation
RANS: Reynolds Averaged Navier-Stokes
RB: Reduced Basis
SVD: Singular Value Decomposition

# Mathematical notations

$(\cdot,\cdot)$ $L^2$-inner product

# Introduction

## 0.1 Context and Motivation

This thesis is devoted to the numerical approximation of parameter-dependent functions of the generic form

$$u : \Omega \times \mathcal{G} \to \mathbb{R}, \tag{1}$$

where $\Omega \in \mathbb{R}^d$ is the spatial domain ($d$ is the spatial dimension), and $\mathcal{G} \subset \mathbb{R}^{N_p}$ is the parameter domain, with $N_p$ the number of parameters. The function $u$ will always be the solution to a parameterized Partial Differential Equation (PDE) modeling a physical phenomenon. Let us denote $\mu = (\mu_1, \ldots, \mu_{N_p}) \in \mathcal{G}$ the varying parameter. For each parameter value $\mu$, we have a new function $u(\mu) \in V$ where $V$ is a suitable Banach space and $V'$ its topological dual solving a PDE of the form

$$\begin{aligned} \mathcal{L}(\mu)(u(\mu)) = F(\mu), \text{ in } \Omega, \\ + \text{ boundary conditions on } \partial\Omega, \end{aligned} \tag{2}$$

where $\mathcal{L}(\mu) : V \to V'$ is the PDE operator that depends on $\mu \in \mathcal{G}$ and $F(\mu) \in V'$ is the PDE right-hand side which does not depend on $u$. For instance, we will often consider elliptic equations of the form

$$\begin{cases} -\operatorname{div}(A(\mu)\nabla u) = f(\mu) \text{ in } \Omega, \\ u = 0 \text{ on } \partial\Omega. \end{cases}$$

This thesis focuses on Reduced Basis Methods (RBM) which are part of the Model Order Reduction (MOR) family. The purpose of RBM is to very quickly find a good finite dimensional approximation of any solution to the problem (2). Usually, classical methods such as Finite Volume schemes (FV) or the Finite Element Method (FEM) are used to provide an accurate approximation. This consists in solving the problem (2) in a subspace $V_h \subset V$, where $h$ is the mesh size. The obtained discrete approximations are denoted $u_h \in V_h$. RBM are not meant to replace such methods but are employed in addition to such solvers, in order to reduce the computational time.

To summarize our contributions:

- We present an overview of the Non-Intrusive Reduced Basis (NIRB) methods studied in this thesis in the first chapter. We explain what their non-intrusiveness means. Our focus is on the two-grid method, which has first been developed in the FEM framework.
- Our main contribution is its generalization to FV solvers and its numerical analysis in this context.

- We generalize and analyze this method to time-dependent equations.
- We study the NIRB capability to recover accurate approximations with singular domains.
- Another important aim of this thesis is concerned with the development of new NIRB methods. We describe a new promising non-intrusive tool based on domain truncations. This NIRB method can be applied to a wide variety of problems and allows us to further reduce their complexity.
- Finally, in the last part, we apply NIRB methods to a highly complex case: the simulation of offshore wind farms. This study is a collaboration with EDF. We detail how NIRB approaches may substantially reduce the computational costs.

Let us now introduce the context of the industrial application.

### 0.1.1 Industrial context

The main context is concerned with developing a new open-source library on non-intrusive MOR techniques, called MOR_DICUS, with several academic and industrial partners. Another industrial goal relies on the application of the NIRB methods to wind farm simulations using this library.

The offshore wind market is growing exponentially all over the world. To achieve carbon neutrality, the European Union is committed to supporting the development of this sector. The aim is to increase European offshore wind production capacity from currently 12 GW to at least 60 GW by 2030 and to 300 GW by 2050. In France, seven offshore wind farm projects are being developed for a total capacity of around 3 900 MW, and calls for tenders on floating systems are increasing.

Building a wind farm requires preliminary studies of the wind power yield. Their aim is to estimate which amount of energy a wind farm can produce taking into account the wind potential and the wake effects.
Offshore wind turbines work in the same way as onshore wind turbines, with the main difference that they are located at sea in order to enhance the use of the wind energy moving its blades. The winds are stronger and more regular than on land. They allow the production of up to 60% more energy than with onshore wind turbines, in particular thanks to the size of turbines. Therefore, they are generally preferred to onshore wind turbines since their efficiency is higher [104].
One drawback is that the installation and the maintenance costs of offshore wind turbines are higher than those of conventional onshore wind turbines. Thus, finely evaluating the local climatology before installing a wind farm is a key issue when it comes to estimating its energy production. The choice of turbines and their precise location requires a more precise analysis of the wind and turbulence conditions taking into account local climatological factors through several simulations. Therefore, the position of the wind turbines is based on a parameterized problem, which makes it a very challenging and relevant application of this thesis. This context is illustrated by Figure 1 and will be detailed later on in subsection 0.3.4.

### 0.1.2 Motivation

Modeling wind farms can be very expensive in terms of computational costs. The energy production is calculated as a function of the wind, taking into account the wake effects. One

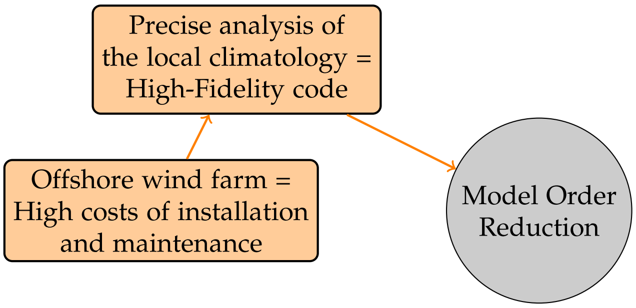


Figure 1: An offshore wind farm application

essential step before calculating the energy production corresponds to the calculation of the wind potential. It is performed as to optimize the position of the turbines in one given farm. This often involves high-fidelity Computational Fluid Dynamics (CFD) codes launched for several wind parameters [88]. It enables the analysis of a range of wind turbine configurations.

## 0.2 Mathematical tools

In general, solving a physical problem with a numerical method is costly in time. To simulate offshore wind farms, an EDF solver, namely Code_Saturne, can be used with the Reynolds Averaged Navier-Stokes method (RANS) which is a method often used to model turbulence. The RANS equations are recalled thereafter (41). This kind of simulation (for example, a wind farm possessing 9 turbines), for one wind setting parameter, can take up to several days.
Thus, reduced order modeling is a major tool to lower the computational time of such problems while keeping approximations as good as with CFD solvers. It is a very quickly emerging field in applied mathematics and computational science. MOR techniques are answering a large demand of efficient computational tools for optimization problems of parameter fitting in the industry. Such problems are modeled by parameter-dependent PDEs, which must be solved for several parameter values. The simulation of wind farms, for instance, relies on parameter-dependent PDEs.

### 0.2.1 Which tools for the simulation of parameter-dependent PDEs?

To solve a parameterized problem, a natural choice consists in seeking a solution in a Banach space with FEM for instance [19, 129]. This solution, which is based on the resolution of a High-Fidelity (HF) code, is costly in time. Thus, for complex applications, it is often more logical to employ MOR [126, 67, 10, 73]. RBM do not replace HF codes, but can be used in addition to such codes to reduce runtimes.
With the RBM, we look for a solution on a manifold which implies a reduction of complexity. This complexity reduction relies on the notion of the Kolmogorov n-width [85]. It is linked to the concept of solution manifold, which is the set of all solutions, computed with a HF code, to the parameterized problem (2) under a parameter variation. This manifold is denoted $\mathcal{S}_h$.

$$\mathcal{S}_h = \{u_h(\mu) \in V_h |\ \mu \in \mathcal{G}\}. \tag{4}$$

RBM can be successful if the Kolmogorov n-width is small, which means that the solution manifold $\mathcal{S}_h$ (4) may be approximated by a finite set of well-chosen solutions (it is illustrated

by Figure 2).
We define the Kolmogorov n-width [85] of $\mathcal{S}_h$ as follows:

**Definition 1.** *If $\mathcal{S}_h$ is a subset of a Banach space $V$, and $\mathbf{Y}_n$ a generic n-dimensional subspace of $V$, then the deviation between $\mathcal{S}_h$ and $\mathbf{Y}_n$ is*

$$E(\mathcal{S}_h; \mathbf{Y}_n) = \sup_{x \in \mathcal{S}_h} \big( \inf_{y \in \mathbf{Y}_n} \|x - y\|_V \big). \tag{5}$$

*Then the Kolmogorov n-width of $\mathcal{S}_h$ in $V$ is*

$$d_n(\mathcal{S}_h, V) = \inf_{\mathbf{Y}_n} \{E(\mathcal{S}_h; \mathbf{Y}_n); \mathbf{Y}_\text{n} \text{ is a } \textit{n-dimensional} \text{ subspace of } V\}. \tag{6}$$

To approximate any solution in $\mathcal{S}_h$, we create a $N$-dimensional subspace of $V_h \subset V$ denoted $X_h^N$ and a Reduced Basis (RB) of this space, where the basis functions are denoted $(\Phi_i^h)_{i=1,\ldots,N}$. RBM aim at approximating any solution belonging to $\mathcal{S}_h$ with a small number of basis functions $N$. This set of basis functions is derived from HF solutions for several well chosen parameter values, $\{u_h(\mu_1), \ldots, u_h(\mu_N)\}$, called *snapshots*.
The small Kolmogorov n-width (6) implies that the manifold $\mathcal{S}_h$ can be approximated with very few RB functions, provided that the parameters are properly chosen for the RB construction [34]. Thanks to that, RBM enable HF real-time simulations and widely reduce the computational costs, with speedups that can reach several orders of magnitude.

**An illustration of the solution manifold.** Figure 2 shows an example of a solution manifold (here denoted $\mathcal{S}_h$ (4), the black curve). All the snapshots $\{u_h(\mu_1), u_h(\mu_2), \cdots, u_h(\mu_N)\}$, belong to a Banach space $V_h$. All the solutions also belong to that same manifold $\mathcal{S}_h$. The parameters of the parameterized problem are denoted $\mu_1, \mu_2, \cdots, \mu_N$ and belong to the set $\mathcal{G}$. The reduced basis space corresponds to the yellow hyperplane, and the green curve is the projected snapshots onto this space. The Kolmogorov N-width represents "the deviation" between that hyperplane and the manifold.
Imagine we search for an approximation $u_h(\mu)$, for $\mu$, a new parameter belonging to the range of the parameters $\mathcal{G}$, then our RB approximation will belong to the space $X_h^N$, generated by the snapshots, and close to the manifold $\mathcal{S}_h$ if the Kolmogorov N-width is small enough.

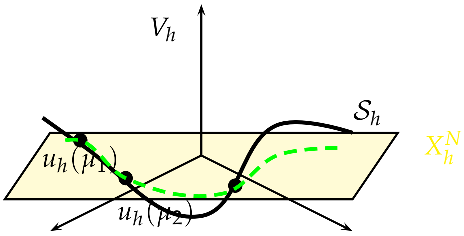


Figure 2: Manifold of the snapshots

One way to confirm the small size of the Kolmogorov n-width is to analyze the complexity of the manifold with a Singular Value Decomposition method (SVD) applied to the correlation matrix of the snapshots. To do so, one should verify the rapid decay of the SVD eigenvalues. The reason is detailed in the reminders on the Proper Orthogonal Decomposition (POD) 1.2 and on the SVD 1.2.1.

**An innovative tool.** Generally, RB methods are intrusive. It means that the user needs to enter the HF solver code and modify it. Conversely, methods termed non-intrusive are those which do not need to modify the solver code. For instance, the Galerkin-Proper Orthogonal Decomposition (Galerkin-POD) [5, 14, 86, 93] is intrusive and requires to modify the assembly of the FEM matrices. This can be difficult or even impossible when the code has been bought, as it is often the case in an industrial context. To alleviate this issue, we use Non-Intrusive Reduced Basis methods (NIRB). They utilize the HF codes as a black-box solver. Several approaches already exist in this framework [117, 118, 119, 69, 95, 64, 28]. We will present some of them, and focus on the two grids method, developed in [96]. This innovative approach has already been employed with complex problems [30], however, it is only beginning to be applied to problems with industrial dimensions. We will analyze it in several contexts, and apply it to the case of an offshore wind farm to highlight the efficiency of this method on highly complex simulation models. We will also propose other non-intrusive tools, adapted from the two-grid method, for different situations.

### 0.2.2 The two-grid method

Let $\Omega$ be a bounded domain in $\mathbb{R}^d$, with $d \leq 3$. The two-grid method, in the context of a HF solver of finite element or finite volume type, involves two partitioned meshes, one fine mesh $\mathcal{M}_h$ and one coarse $\mathcal{M}_H$, where the respective sizes $h$ and $H$ of the meshes are such that $h << H$. The size $h$ (respectively $H$) is defined as

$$h = \max_{K \in \mathcal{M}_h} h_K \text{ (respectively } H = \max_{K \in \mathcal{M}_H} H_K), \tag{7}$$

where the diameter $h_K$ (or $H_K$) of any element $K$ in a mesh is equal to $\sup\limits_{x,y \in K} |x - y|, K \in M_h$ (or $\in M_H$).

One grid is needed for the reduced basis generation, and another one to roughly approximate the solution. This will be the key ingredient to the reduction of computational costs, as we will highlight in chapters 1 and 2. The implementation has two main steps:

- First, the RB functions are prepared in an "offline" stage with a fine mesh. It involves a greedy algorithm (algorithm 2) or a POD procedure (algorithm 1). This part is costly in time, but only executed once, as for other RBM. At the end of this stage, we obtain $N$ $L^2$-orthonormalized basis functions $(\Phi_i^h)_{i=1,\dots,N}$.
- Then, a coarse approximation of the solution, for the new parameter $\mu$ we are interested in, is computed "online". We denote this coarse solution $u_H(\mu)$. This rough approximation is not of sufficient precision but is calculated with a smaller number of degrees of freedom compared to the fine mesh ones. It is used as a cheap surrogate of the optimal coefficients
$$\alpha_i^h(\mu) = \int_\Omega u_h(\mu) \cdot \Phi_i^h \, d\mathbf{x}.$$
  Reduced basis post-processing then makes it possible to notably improve the precision by projection and rectification on the reduced basis, within a very short runtime. The classical NIRB approximation is given by
$$u_{Hh}^N(\mu) := \sum_{i=1}^N (u_H(\mu), \Phi_i^h) \, \Phi_i^h \in X_h^N. \tag{8}$$

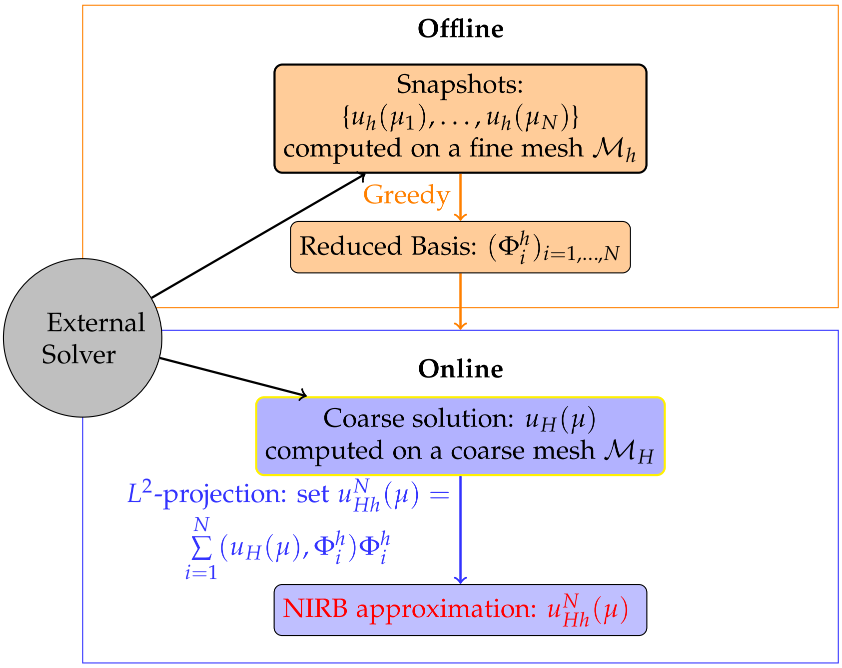


Figure 3: The two-grid method

Figure 3 illustrates the two-grid method.

This method, which can be used for several types of PDEs, is easy to implement. Moreover, its non-intrusive nature makes it applicable to a wide range of problems. A post-treatment, called the rectification post-processing, which has been introduced in [30] and employed in [72, 99], may be used to further improve the two-grid approximation accuracy. We now recall the main steps of this post-processing stage.

### The rectification post-treatment

The main idea of its strategy consists in recovering the accuracy of the approximation given by the optimal coefficients without sacrificing on the computational complexity. As explained in [30], it may be employed in addition to the NIRB classical algorithm. During the offline stage, after the fine snapshots generation, for the same parameter values, the corresponding coarse snapshots are computed in $\mathcal{M}_H$. Thus, we introduce the fine and coarse coefficients

$$\alpha_i^h(\mu) = \int_\Omega u_h(\mu)\cdot\Phi_i^h\, d\mathbf{x} \text{ and } \alpha_i^H(\mu) = \int_\Omega u_H(\mu)\cdot\Phi_i^h\, d\mathbf{x}. \tag{9}$$

The purpose is to create a rectification matrix that allows us to pass from the coarse coefficients to the fine ones. This implies that if the true solution is in the reduced space, then the NIRB method will give this true solution. Let us discretize $\mathcal{G}$ and obtain $Ntrain$ parameters in $\mathcal{G}$, considered as a training set. Let $N$ be the number of basis functions ($N \leq Ntrain$). We define $\mathbf{A} \in \mathbb{R}^{Ntrain\times N}$ the matrix of the coarse coefficients and $\mathbf{B} \in \mathbb{R}^{Ntrain\times N}$ the one constructed from the fine coefficients such that

$$\forall i = 1,\cdots,N, \text{ and } \forall \mu_k \in \mathcal{G}, \quad A_{k,i} = \alpha_i^H(\mu_k), \quad \text{and } B_{k,i} = \alpha_i^h(\mu_k). \tag{10}$$

The approach is based on a regularized least-square method. Without regularization, its purpose is to minimize the error between the projection of the rectified approximation onto the basis and the optimal approximation as a function of the rectification matrix. Thus, let us introduce the rectification matrix $\mathbf{R} = (R_{i,j})_{1 \leq i,j \leq N} \in \mathbb{R}^{N \times N} : X_h^N \to X_h^N$. The rectification step aims to find $\mathbf{R}$ minimizing

$$\left\| \sum_{i=1}^{N} \alpha_i^h(\mu_k) \Phi_i^h - \sum_{i=1}^{N} \sum_{j=1}^{N} R_{i,j} \alpha_j^H(\mu_k) \Phi_i^h \right\|^2, \quad \forall k = 1, \ldots, Ntrain.$$

With the $L^2$ orthonormalization of the RB functions, it is equivalent to minimize

$$|\alpha_i^h(\mu_k) - \sum_{j=1}^{N} R_{i,j}\, \alpha_j^H(\mu_k)|^2, \quad \forall k = 1, \ldots, Ntrain.$$

as a function of $\mathbf{R}$. Thus, it consists in looking for $\mathbf{R}$ minimizing the cost functions

$$\|\mathbf{A}\mathbf{R}_i - \mathbf{B}_i\|_2^2, \quad i = 1, \cdots, N, \tag{11}$$

where $\|\cdot\|_2$ stands for the Euclidian $l^2$-norm. Here, Tikhonov regularization consists in promoting solutions of such problems with small norms. Thus equation (11) becomes

$$\|\mathbf{A}\mathbf{R}_i - \mathbf{B}_i\|_2^2 + \lambda \|\mathbf{R}_i\|_2^2, \quad i = 1, \cdots, N, \tag{12}$$

where $\lambda$ is a regularization term. The solution to this problem (12) is the rectification matrix:

$$\mathbf{R}_i = (\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}_N)^{-1} \mathbf{A}^T \mathbf{B}_i,\ i = 1, \cdots, N, \tag{13}$$

Then, the NIRB approximation given by (8) becomes

$$Ru_{Hh}^N(\mu) = \sum_{i,j=1}^{N} R_{ij}\, \alpha_j^H(\mu)\, \Phi_i^h. \tag{14}$$

## 0.3 Layout of the thesis

This thesis, focusing on the two-grid method, is decomposed in four connected parts, combining theoretical and applied parts.

### 0.3.1 Chapter 1: State of the art

The first chapter 1 is dedicated to the state of the art of the RBM. The main idea is to present a non-exhaustive list of RBM methods, to exhibit the advantages and disadvantages of each of them. In this chapter, we will consider as a model problem the 2D lid driven cavity with steady Navier-Stokes 1.1. Let $\Omega$ be the unit square with its upper bound denoted $\Gamma_3$. The 2D steady Navier-Stokes equation writes:

$$\begin{cases} (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = 0, \text{ on } \Omega, \\ \nabla \cdot \mathbf{u} = 0, \text{ on } \Omega, \\ \mathbf{u}_{|\Gamma_3} = (1, 0), \\ \mathbf{u}_{|\partial\Omega \backslash \Gamma_3} = (0, 0), \end{cases} \tag{15}$$

where $\mathbf{u} = (u, v)$ is the velocity, $p$ is the pressure, $\nu = 1/Re$, and $Re$ is the Reynolds number.

$$\boxed{\text{In this chapter, } \nu = \frac{1}{Re} \text{ will represent our varying parameter.}}$$

Many existing RBM have emerged in recent years and have reached an industrial degree of maturity:

1. We will first introduce an intrusive method, which is the standard projection-based model reduction method, known as the Galerkin Proper Orthogonal Decomposition (Galerkin-POD) [86, 4, 110]. The offline part employs the Snapshot POD. We analyze the link between the latter and the Singular Value Decomposition (SVD) and the Principal Component Analysis (PCA). We will explain what makes the Galerkin-POD intrusive. The Snapshots POD is non-intrusive and it may be used for the offline part of the two-grid NIRB method, even if in general the greedy algorithm is preferred. In the Galerkin-POD, when dealing with nonlinearities, it may be necessary to employ in addition the Empirical Interpolation Method (EIM) [64] which is recalled in appendix A.0.1. This method will not be employed in the other chapters. Indeed, one of the advantage of the two-grid method is that its algorithm covers a large scope of problems and EIM method is not required for nonlinearities.

2. In the second section, we will review some Non-Intrusive Reduced Basis methods that we studied during this thesis. In appendix A, a short presentation on other RB methods that further reduce the cost of the POD is given.

    - One often used non-intrusive approach is the POD-Interpolation (PODI) [15, 57]. It combines the offline part of the Galerkin-POD and a regression with several possibilities during the online stage.
    - We will end this chapter with a detailed presentation of the two-grid method.

    For each approach, its advantages and disadvantages will be highlighted. We will also compare the two-grid numerical results on the model problem (15) to other RB methods.

All RBM are based on a two stage procedure, including an offline and an online stage. The offline part, executed only once, which consists in building the reduced basis, can be very costly, even if some methods are emerging to reduce these costs [82]. The online stage, in the standard RBM, consists of a Galerkin projection with a varying parameter value $\mu \in \mathcal{G}$, onto the space spanned by the RB.

The second part of this chapter is directly applied to the project of developing an open-source shared software library in Python, called MOR_DICUS, on Non-Intrusive Reduced Basis methods. In this project still in progress, there will be several NIRB methods available, and we implemented the two-grid method and the rectification post-treatment with Python. All "objects" such as the mesh or the RB are defined with Python and C++ classes. Python classes provide all the standard features of Object Oriented Programming (OOP). it is a classical approach for software design. It uses a particular process of defining objects and their interactions when planning code to develop a computer program. The classes, which can inherit from other classes, contain attributes that define the data. The goal is to be able to highlight the non-intrusiveness of all the NIRB methods. The objects are defined very generally with attributes, and then, for each algorithm, we may define new routines with these objects. The two-grid

method and several routines (such as the greedy algorithm 2, or mesh reading subroutines, . . . ) have been implemented in order to be able to use FreeFem++ [71], GMSH [24] or VTK [122] meshes and 2-dimensional or 3-dimensional solutions. These formats are three different ways of defining the mesh attributes. We also implemented a routine that saves the NIRB approximation in the VTK format.
The code is still being improved to deal with xml format. Currently, the NIRB codes with VTK inputs/outputs has been used with the wind farm tests (see chapter 4). Indeed, the black-box solver used in these tests is code_saturne and it may save solutions with VTK or with Ensight Gold. Currently, the algorithm for the two-grid method takes in input

- the directories path where the snapshots and the solutions files are located,
- the problem dimension,
- the components number of the solution and its fieldname,
- the chosen method (POD 1, Greedy 2, with or without rectification post-treatment 0.2.2),
- a boolean indicating whether the solution must be saved or not,
- and one boolean for the error computation if one fine solution is provided.

It clearly shows the non-intrusiveness of these algorithms. For one solver solution, its values are read at the degrees of freedom for FEM (with the function “LoadVtuWithVTK” of the Python library “Basictools”) and they may also be retrieved from the cells values for FV (function “vtk.vtkCellDataToPointData()” from VTK module).

The FEM matrices required for the errors and for the RB generation are created with the library Basictools. The reduced basis is saved with Pickle format. At the end of the online stage, the NIRB approximation may be saved on the fine mesh in VTK format.

We implemented the Greedy algorithm and subroutines to read the meshes and solutions and to interpolate them. The code uses several containers such as the mesh definitions or a collection of problem data. The functions and attributes inheritate from parent classes that we previously defined, which makes the code more efficient and easier to update. The inheritance makes it possible to give a class all the characteristics of one or more other classes (for example we use the class “Solution” to declare the snapshots which saves the degrees of freedom, the number of components, and its spatial dimension, . . . ).

### 0.3.2 Chapter 2: Numerical analysis of the two-grid method

This chapter 2 is concerned with the two-grid method “in all its greatness”. To analyze the method, we will consider the following elliptic model parameterized problem, on a spatial domain $\Omega$:

$$\begin{cases} -\operatorname{div}(A(\mu)\nabla u) = f(\mu) \text{ in } \Omega, & \text{(16a)} \\ u = 0 \text{ on } \partial\Omega, & \text{(16b)} \end{cases}$$

where $f(\mu) \in L^2(\Omega)$, $\mu \in \mathcal{G}$ is a parameter value, and for any $\mu \in \mathcal{G}$, $A(\cdot;\mu) : \Omega \to \mathbb{R}^{d\times d}$ is measurable, bounded, uniformly elliptic, and $A(\mathbf{x};\mu)$ is symmetric for a.e. $\mathbf{x} \in \Omega$.

In this chapter, $\mu \in \mathcal{G}$ will represent our varying parameter.

Under the previous hypotheses, it is well known that this problem (16a)-(16b) has a unique solution in $H_0^1$. Let $k \in \mathbb{N}$ , and we consider the $H^k$ Sobolev space and its associated norm $\|\cdot\|_{H^k(\Omega)}$.
The usual weak formulation for the problem (16a)-(16b) reads:
Find $u \in H_0^1(\Omega)$ such that,

$$\forall v \in H_0^1(\Omega), \quad a(u, v; \mu) = (f, v), \tag{17}$$

where

$$a(w, v; \mu) = \int_\Omega A(\mathbf{x}; \mu) \nabla w(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x}, \quad \forall w, v \in H_0^1(\Omega). \tag{18}$$

1. In the first section, we will explain the different stages of the NIRB approach. We will underline its efficiency with a precise numerical analysis in the FEM context. Its numerical analysis relies on two theorems of the FEM theory, which are Cea's Lemma and Aubin-Nitsche's Lemma, recalled below.

   **Lemma 1.** *Céa's Lemma.*

   *Let $V = H_0^1$ be a real Hilbert space with the associated norm $\|\cdot\|$. Let $a : V \times V \to \mathbb{R}$ be a bilinear form with the properties*

   - *$|a(v, w)| \leq \gamma \|v\| \, \|w\|$ for some constant $\gamma > 0$ and all $v, w$ in $V$ (continuity).*
   - *$a(v, v) \geq \alpha \|v\|^2$ for some constant $\alpha > 0$ and all $v \in V$ (coercivity or $V$-ellipticity).*

   *Let $L : V \to \mathbb{R}$ be a bounded linear operator. Consider the problem of finding an element $u \in V$ such that*

   $$a(u, v) = L(v), \text{ for all } v \in V. \tag{19}$$

   *Consider the same problem on a finite-dimensional subspace $V_h$ of $V$, such that, $u_h \in V_h$ satisfies*

   $$a(u_h, v) = L(v) \text{ for all } v \in V_h. \tag{20}$$

   *By Lax–Milgram theorem, each of these problems has exactly one solution. Céa's lemma states that*

   $$\|u - u_h\| \leq \frac{\gamma}{\alpha} \|u - v\| \text{ for all } v \in V_h. \tag{21}$$

   **Lemma 2.** *Aubin-Nitsche's Lemma.*

   *Under the previous hypothesis, if $\Omega$ is a convex polyhedron, the approximation $u_h$ of the problem 19, given by the Galerkin method in the space $V_h$ of the FEM space $\mathbb{P}_1$, verifies the estimate*

   $$\|u - u_h\|_{L^2} \leq Ch \|u - u_h\|_{H^1}. \tag{22}$$

   Let denote $I_h$ the nodal inteprolation operator from $C(\overline{\Omega})$ into the $C^0$ Lagrange FE space $V_h$, we have the following classical theorem (Theorem 4.4.20 [19])

   **Theorem 3.** *If $u \in H^2(\Omega)$,*

   $$\left\|u - I_h(u)\right\|_{H^1(\Omega)} \leq Ch \|u\|_{H^2(\Omega)}. \tag{23}$$

   Thus, Céa's lemma (equation (1)) and theorem 3 entails:

   $$\left\|u(\mu) - u_h(\mu)\right\|_{H^1} \leq C \, \left\|u(\mu) - I_h(u(\mu))\right\|_{H^1} \leq Ch \, \|u\|_{H^2(\Omega)} \tag{24}$$

The two-grid method has been developed and analyzed (with Céa's and Aubin-Nitsche's lemmas) in the context of $\mathbb{P}_1$ FEM in [96]. The energy-error estimate for an elliptic problem is given by

$$\left\| u(\mu) - u_{Hh}^{N}(\mu) \right\|_{H^1(\Omega)} \leq \varepsilon(N) + C_1 h + C_2(N) H^2, \tag{25}$$

with $u(\mu)$ the exact solution, for a parameter $\mu \in \mathcal{G}$, $u_{Hh}^{N}(\mu)$ the corresponding NIRB approximation (8) (with $N$ RB functions), $\varepsilon(N)$ depends on the Kolmogorov $N$-width, $h$ is the size of the fine mesh, and $H$ the size of the coarse mesh (7). The constants $C_1$ and $C_2$ do neither depend on $h$ nor $H$. The constant $C_2$ depends on $N$ (see remark 4).
In this estimate (25),

- the first term, $\varepsilon(N)$, negligible, is linked to the Kolmogorov $N$-width,
- the second term, $C_1 h$, comes from the Céa's Lemma (1),
- and the last one, $C_2(N)H^2$, follows from the Aubin-Nitsche's Lemma (2).

If the coarse mesh is such that $H^2 \simeq h$, then the error estimate in the $H^1$-norm is in $\mathcal{O}(h)$, which is the classical convergence order with $\mathbb{P}_1$ FEM space. This estimate may be generalized with $\mathbb{P}_k$ FEM spaces, $k > 1$ [96]. Thus, we recover the classical FEM energy error estimate on the fine mesh while reducing the degrees of freefom, since we are only using the coarse mesh approximation in the online stage.

**Main contributions of this section**
Usually the RB is $L^2$-orthonormalized with a Gram-Schmidt procedure. We will demonstrate that the constants $C_1$ and $C_2$ of the inequality (25) change as a function of the RB functions chosen, whether or not it is orthogonalized only in $L^2$ or also orthogonalized in $H^1$. We will also illustrate that fact with numerical results. We also detail the NIRB numerical analysis with the $L^2$ norm.

**Remark 4.** *The constant $C_2$ depends on the number of reduced basis $N$, and thus, a trade-off needs to be done between increasing $N$ to obtain a more accurate manifold, and keeping a constant $C_2$ as low as possible. We will illustrate this $N$-dependance with several examples (see for instance Figure 2.2 in chapter 2 or Figure 4.7 in chapter 4).*

2. This section is concerned with two further analyses on the NIRB two-grid method.
   (a) The first one 2.2.1 is a study on the NIRB two-grid FEM $\mathbb{P}_1$ estimate when several spatial grids are used, with various connectivities but with the same nodes. It is of special interest when the solver meshes are not easily recovered and when only the nodes coordinates can be retrieved.
      - In order to create the RB, we use an external solver to obtain our snapshots.
      - Now, one may not be able to retrieve all the information on the mesh. Suppose that this solver only gives the mesh coordinates without the connectivity.
      - With such a knowledge, it is possible to generate a Delaunay triangulation (for instance with FreFem++ or with Paraview). It yields a new approximation $\widetilde{u_{Hh}^{N}(\mu)} = \widetilde{I}_h(u_{Hh}^{N}(\mu))$. Does one recover the classical error estimate in $\mathcal{O}(h)$ for this new NIRB approximation?

Figure 4 displays an exemple of two simple meshes defined with the same nodes, but using two different connectivities.

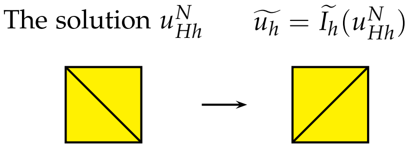


Figure 4: Two meshes with different connectivities

The result of this study comes from [13]. The fractional Sobolev seminorm [2], which is denoted $|\cdot|_{H^{k+\lambda}(\Omega)}$, is recalled in section 2.2.1. We are going to use two estimates from [13]. The first one is linked to a fractional sobolev interpolation. We consider the nodal interpolation operator $\Pi^h_1$ from $C(\overline{\Omega})$ to $C^0$ Lagrange finite element $V_h$ space associated to the mesh $\mathcal{M}_h$.

**Theorem 5** (Theorem 2.6. [13])**.** *Let* $\lambda \in (0,1)$*. Then we have, for* $d = 1,2$*,*

$$|v - \Pi^h_1(v)|_{H^{1+\lambda}(\Omega)} \leq Ch^{\mu-\lambda}|v|_{H^{1+\mu}(\Omega)}, \ \forall v \in H^{1+\mu}(\Omega),\ \lambda \leq \mu \leq 1. \tag{26}$$

The second inequality is concerned with inverse estimates.

**Theorem 6** (Theorem 2.9. [13])**.** *Let* $\lambda \in (0, \frac{1}{2})$ *and* $\theta \in [0,\lambda]$*. Then the following estimate holds*

$$|v_h|_{H^{1+\lambda}(\Omega)} \leq Ch^{\theta-\lambda}|v_h|_{H^{1+\theta}(\Omega)}, \ \forall v_h \in V_h. \tag{27}$$

Our main result is the following theorem.

**Theorem 7.** *We denote* $u$ *the exact solution to* (16a)-(16b)*. Let* $V_h$ *be the FE space associated with the* $\mathbb{P}_1$ *elements, and* $u_h \in V_h$ *solution to* (20) *on the mesh* $\mathcal{M}_h$*. Suppose that another mesh* $\widetilde{M}_h$ *with the same nodes is used to obtain another approximation* $\widetilde{u_h}$*. Then, the following estimate holds*

$$\|u - \widetilde{u_h}\|_{H^1} \leq Ch\|u\|_{H^2}\,. \tag{28}$$

Thus, this theorem 7 allows us to retrieve the optimal error estimate with the NIRB approximation.

(b) The next study 2.2.2 aims to extend the NIRB method to parabolic equations. This is the object of an upcoming article. To the best of our knowledge, the two-grid method has not already been studied in the context of time-dependent problems. We will first demonstrate that the two-grid method can be used on such problems with a model problem. For the model problem, we consider the following heat equation with a source term:

$$\begin{aligned} u_t - \mu\Delta u &= f, \ \text{ in } \Omega\times]0,T[,\\ u(\mathbf{x},0) &= u_0(\mathbf{x}), \ \text{ in } \Omega,\\ u(\mathbf{x},t) &= 0, \ \text{ on } \partial\Omega, \end{aligned} \tag{29}$$

where $\mu$ is the parameter. For the spatial discretization, a $\mathbb{P}_1$ FEM is employed, and for time discretization, we will employ two different schemes. Indeed, with the time

discretization, there is no equivalent to the Aubin-Nitsche's trick.
With the same scheme, we recover the same order in time for the $L^2$ norm as well as for the $H^1$ norm. Thus, we will employ two schemes. Their choice is motivated by the fact that we are looking for a rough solution coarser in space as well as in time, compared to the fine solution. Therefore, during the NIRB offline stage, on the fine mesh, we will employ a Backward Euler scheme, and for the rough solution, we will use a Crank-Nicolson scheme. Numerical results will be presented on the model problem.
The main result is the following theorem.

**Theorem 8.** *NIRB error estimate for parabolic equations. Let us consider the problem 29 and its solution, which belongs to $H^2(\Omega) \cap H_0^1(\Omega)$. We define two time grids.*

- *One time grid, denoted $F$, is employed for the fine solution (and for the snapshots). To avoid making notations more cumbersome, we will consider a uniform time grid with $\Delta t_F$ the interval between two time values. The time levels can be written $t = t_n = n\Delta t_F$, where $n \in \mathbb{N}^*$, and we denote $u_h^n \in V_h$ the approximation of $u_h(t^n)$ to be determined.*
- *Another time grid, denoted $G$, is used for the coarse solution. By analogy with the fine grid, we consider a uniform grid with time step $\Delta t_G$. This time, the time levels are written $t = t_m = m\Delta_G$, where $m$ is a nonnegative integer, and we denote $u_H^m \in V_h$ the approximation of $u_H(t^m)$ to be determined.*

*The NIRB approximation writes*

$$u_{Hh}^{N,n}(\boldsymbol{x};\mu) = \sum_{i=1}^{N} \alpha_i^H(\mu, t^n)\ \Phi_i^h(\boldsymbol{x}),\ n \geq 0, \tag{30}$$

*where the basis functions $(\Phi_i^h)_{i=1,\ldots,N}$ are not depending on time.*
*Under some further assumptions that will be detailled in section 2.2.2, the following estimate holds*

$$\forall n,\ \left\| u(t^n)(\mu) - u_{Hh}^{N,n}(\mu) \right\|_{H^1(\Omega)} \leq \varepsilon + C_1 h + C_2(N) H^2 + C_3 \Delta t_F + C_4(N)\Delta t_G^2, \tag{31}$$

*where $C_1, C_2, C_3$ and $C_4$ are constants independent of $h$ and $H$, $\Delta t_F$ and $\Delta t_G$. The term $\varepsilon$ depends on the Kolmogorov $N$-width. If $H$ is such as $H^2 \sim h$, $\Delta t_G^2 \sim \Delta t_F$, and $\varepsilon(N)$ is small enough, it results in an error estimate in $\mathcal{O}(h) + \Delta t_F$.*

3. This section revolves around domain singularities. This is the object of an upcoming article. The two-grid method in the FEM context is applied with a new stategy in order to counterbalance the effects of domain singularities. There are many works in the litterature on re-entrant corners and domain singularities [25, 120, 134, 133, 84, 79]. Most of the methods are based on adaptive refinements in the vicinity of the singularities to achieve a better accuracy. While MOR methods have been proposed for various fields in science and engineering, only few approaches have been developed to treat domain singularities [60, 31, 1]. In this section, the NIRB algorithm is used to numerically solve such problems. We take advantage of the fact that the NIRB methods are decomposed in two stages. All the techniques to retrieve an accurate approximation are employed in the offline step. Here, we consider refinement methods. A refinement is only operated during the offline stage. Hence, the computation times linked to such refinement can be considerably reduced in the online phase. With re-entrant corners, we recover optimal

errors in the energy norm while using a uniform coarse mesh for the NIRB approximation during the online stage. The main advantages of the NIRB in this context are concerned with the size of the rough mesh employed during the online stage, as well as its uniformity. We summarize the offline/online strategy:

(a) During the **"offline"** stage, the fine mesh is refined around the re-entrant corners, in order to obtain optimal results. The fine snapshots are generated on it. Then the RB is created with the greedy algorithm or the snapshots POD.

(b) Then, in the **"online"** part, the coarse approximation is computed on a uniform coarse mesh and it is projected onto the basis space denoted $X_h^N$.

This process allows us to retrieve a very accurate approximation with less degrees of freedom. We present several numerical results on the model problem (16a)-(16b) on the classical L-shape domain with one re-entrant corner and on the backward-facing step 3.1.

4. In the next section, the method will be generalized to Finite Volume (FV) solvers and analysed in this context. This work has been published in the Journal ESAIM: Mathematical Modeling and Numerical Analysis (ESAIM: M2AN) [66]. The main motivation of this study is that, in many industrial applications, and for several reasons, FV Methods (FVM) are preferred to FEM. Firstly, it may be easier to implement and the mesh can be adapted more easily to the domain. Secondly, the problem formulation is direct, unlike the FEM which consists in writing the problem with a weak formulation. Thus, it implies a better preservation of certain physical properties (the flux conservation and the flux balance).

   Several papers have underlined the efficiency of the NIRB method in the finite element context, illustrated both with numerical results presenting error plots and the online part computational time [96, 28, 30, 29]. However, to the best of our knowledge, solvers relying on FV schemes have not yet been studied with a non-intrusive approach [80, 126, 67, 121, 27]. Thanks to recent works on super-convergence [47], and with some technical subtleties, we generalized the two-grids method to FVM and developed its numerical analysis. We will demonstrate that optimal approximations are recovered in this context [66]. Optimal error estimates, similar to those of classical methods, computed with a HF code are retrieved, and these results are illustrated with numerical simulations.

   The main difficulty of this study is that the principal theorem used in the FEM analysis, namely the Aubin-Nitsche's lemma 2, does not hold as it is anymore, in the FV framework. To overcome this issue, we will consider a Finite Volume family, the class of Hybrid Mimetic Mixed methods (HMM) schemes for elliptic equations. We will utilize recent works on a super-convergence property of these schemes ( [41, 47, 42]), which allow us, with several interpolation tricks, to recover the same estimate in the $L_2$-norm as with the Aubin-Nitsche's lemma in the FEM context. It results in a better approximation rate in the $L_2$-norm than in the energy norm. The HMM family [45, 38, 44] is quite general and involves for instance the Two-Points Flux Approximation (TPFA) which is widely used and very easy to implement. What follows is the mathematical background used to obtain the error estimate.

### The Hybrid Mixed Mimetic methods (HMM) family

Describing the HMM family requires to introduce the Gradient Discretization (GD) method [45], which is a general framework for the definition and the convergence analysis of many numerical methods (finite element, finite volume, mimetic finite difference methods, etc). Indeed, in some cases, HMM boil down to GD, and GD are quite pratical for numerical analysis.
The GD schemes involve a discete space, a reconstruction operator and a gradient operator, which taken together are called a Gradient Discretization. Selecting the gradient discretization mostly depends on the boundary conditions (BCs). We now introduce the definition of GD for Dirichlet BCs as in [45] and the GD scheme associated to our model problem.

**Definition 2.** *(Gradient Discretization). For homogeneous Dirichlet BCs, a gradient discretization $\mathcal{D}$ is a triplet $(X_{\mathcal{D},0}, \Pi_{\mathcal{D}}, \nabla_{\mathcal{D}})$, where the space of degrees of freedom $X_{\mathcal{D},0}$ is a discrete version of the continuous space $H^1_0(\Omega)$.*

- $\Pi_{\mathcal{D}} : X_{\mathcal{D},0} \to L^2(\Omega)$ *is a function reconstruction operator that relates an element of* $X_{\mathcal{D},0}$ *to a function in* $L^2(\Omega)$.
- $\nabla_{\mathcal{D}} : X_{\mathcal{D},0} \to L^2(\Omega)^d$ *is a gradient reconstruction in* $L^2(\Omega)$ *from the degrees of freedom. It must be chosen such that* $\|\cdot\|_{\mathcal{D}} = \|\nabla_{\mathcal{D}}\cdot\|_{L^2(\Omega)^d}$ *is a norm on* $X_{\mathcal{D},0}$.

The definition of these operators will be precisely given in the case of HMM-GD in section 2.4.

In what follows, we will refer to $\Pi^H_{\mathcal{D}}$ or $\Pi^h_{\mathcal{D}}$ depending on the mesh considered and for the gradient reconstruction too (respectively $\nabla^H_{\mathcal{D}}$ or $\nabla^h_{\mathcal{D}}$).

**Definition 3.** *(Gradient discretization scheme). For the variational form* (17)*, the related gradient discretization scheme with the new operators is defined by:*
*Find* $u_{\mathcal{D}} \in X_{\mathcal{D},0}$ *such that,* $\forall v_{\mathcal{D}} \in X_{\mathcal{D},0}$,

$$\int_{\Omega} A(\mu)\nabla_{\mathcal{D}} u_{\mathcal{D}} \cdot \nabla_{\mathcal{D}} v_{\mathcal{D}} \, d\boldsymbol{x} = \int_{\Omega} f \, \Pi_{\mathcal{D}} v_{\mathcal{D}} \, d\boldsymbol{x}. \tag{32}$$

We will use two general polytopal meshes (Definition 7.2 [45]) which are admissible meshes for the HMM scheme. We recall the definition of a polytopal mesh in 2.4. To sum up, the mesh must be a partition of connected disjoint polytopes (the cells) where each cell is star-shaped with respect to one point denoted $\mathbf{x}_K$. Such meshes are defined by quadruplets $\mathcal{T} = (\mathcal{M}, \mathcal{F}, \mathcal{P}, \mathcal{V})$.

Figure 5 illustrates a cell of a 2D polytopal mesh.

In what follows, we will consider two polytopal meshes. The fine mesh will be denoted $\mathcal{T}^h = (\mathcal{M}^h, \mathcal{F}^h, \mathcal{P}^h, \mathcal{V}^h)$ and $\mathcal{T}^H = (\mathcal{M}^H, \mathcal{F}^H, \mathcal{P}^H, \mathcal{V}^H)$ will be referred to as the coarse mesh.

All HMM schemes require to choose one point inside each mesh cell $\mathbf{x}_K$, and in the case where the center of mass $\overline{\mathbf{x}}_K$ is chosen, then the scheme corresponds to hMFD and superconvergence is well known [41, 47, 38, 46]. The definition of the hMFD scheme will be recalled in section 2.4.

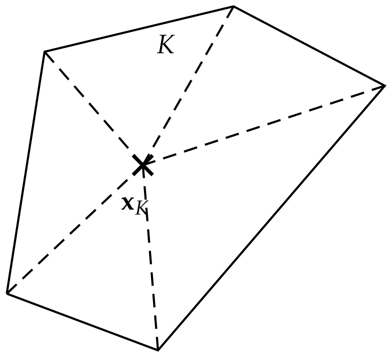


Figure 5: A cell $K$ of a polytopal 2D mesh

### The Hybrid Mimetic Finite Difference (hMFD) method

We now introduce the super-convergence property on hMFD [47], which will be used in the proof of the theorem 10. But first we need the following $H^2$-regularity assumption (which holds if $A$ is Lipschitz continuous and $\Omega$ is convex):

**Assumption 1.** *Let $f \in L^2(\Omega)$, the solution $u(\mu)$ to* (17) *belongs to $H^2(\Omega)$, and*

$$\|u(\mu)\|_{H^2(\Omega)} + \|A(\mu)\nabla u(\mu)\|_{H^1(\Omega)^d} \leq C\|f\|_{L^2(\Omega)}, \tag{33}$$

*with $C$ depending only on $\Omega$ and $A$.*

We define $\pi_{\mathcal{M}^h} : L^2(\Omega) \to L^2(\Omega)$ as the orthogonal projection on the piecewise constant functions over $\mathcal{M}^h$ that is

$$\forall \Psi \in L^2(\Omega), \quad \forall K \in \mathcal{M}^h, \quad \pi_{\mathcal{M}^h}\Psi = \frac{1}{|K|}\int_K \Psi(\mathbf{x})\, d\mathbf{x} \text{ on } K.$$

**Theorem 9** (Super-convergence for hMFD schemes [47]). *Let $d \leq 3$, $f \in H^1(\Omega)$, and $u(\mu)$ be the solution to* (17) *under assumption* (33). *Let $\mathcal{T}_h$ be a polytopal mesh, and $\mathcal{D}$ be an HMM gradient discretization on $\mathcal{T}_h$ with the unknowns defined on $\mathbf{x}_K$, and let $u_h(\mu)$ be the solution to the corresponding GD. Recall that $\overline{\mathbf{x}}_K$ is the center of mass of $K$ and we place ourselves in the case where $\mathbf{x}_K = \overline{\mathbf{x}}_K$. Then, considering $u_{\mathcal{P}}(\mu)$ as the piecewise constant function on $\mathcal{M}_h$ equal to $u(\overline{\mathbf{x}}_K;\mu)$ on $K \in \mathcal{M}$, there exists $C > 0$ not depending on $h$ such that*

$$\left\|\Pi^h_{\mathcal{D}} u_h(\mu) - u_{\mathcal{P}}(\mu)\right\|_{L^2(\Omega)} \leq C\,(\|f\|_{H^1(\Omega)} + \|u\|_{H^2(\Omega)})\, h^2. \tag{34}$$

To recover (34) in the case $\mathbf{x}_K = \overline{\mathbf{x}}_K$, we used the Lemma 7.5 of [47] on the approximation of $H_2$ functions by affine functions to obtain

$$\|\pi_{\mathcal{M}^h} u(\mu) - u_{\mathcal{P}}(\mu)\|_{L^2(\Omega)} \leq Ch^2\|u\|_{H^2(\Omega)}.$$

and Theorem 4.7 [47].

### Main results of this section

First, we will focus on one class of the HMM, which is the hybrid Mixed Finite Difference method (hMFD). It is a finite volume method despite its name. Indeed hMFD scheme relies on both a flux balance equation and on a local conservativity of numerical fluxes. It uses interface values and fluxes as unknowns. The particularity of hMFD is that the cell unknowns must be located at the center of mass of cells. Then the results will be generalized to other FV schemes. The main result is theorem 10 below:

**Theorem 10** (NIRB error estimate for hMFD solvers)**.** *Let* $u^N_{Hh}(\mu)$ *be the NIRB approximation, computed with FV approximations given with an hMFD solver with the unknowns defined on the cell centers of mass, and* $u(\mu)$ *be the exact solution to* (17) *under an* $H^2$ *regularity assumption (33), then the following estimate holds*

$$\left\| u(\mu) - u^N_{Hh}(\mu) \right\|_{\mathcal{D}} \leq \varepsilon(N) + C_1 h + C_2(N) H^2, \tag{35}$$

*where* $C_1$ *and* $C_2$ *are constants independent of* $h$ *and* $H$*,* $C_2$ *depends on* $N$*, the number of functions in the basis, and* $\|\cdot\|_{\mathcal{D}}$ *is the discrete norm which has been introduced in the GD definition 2, and* $\varepsilon$ *depends of the Kolmogorov N-width. If* $H$ *is such as* $H^2 \sim h$*, and* $\varepsilon(N)$ *small enough, it results in an error estimate in* $\mathcal{O}(h)$*.*

This theorem shows that for a certain class of FV schemes, generalized to other FV settings thereafter, we are able to recover optimal estimates, as in the FEM context (25). The degrees of freedom in FVM do not have the same status as in FEM and the transfer of information from one grid to another must be adapted. To recover the estimate (35), we will employ interpolation lemmas such as Bramble-Hilbert's Lemma [19], recalled below.

**Lemma 11.** *Bramble-Hilbert's Lemma*

*Let* $\Omega$ *be a bounded domain and let* $B$ *be a ball in* $\Omega$ *such that* $\Omega$ *is star-shaped with respect to* $B$*. Let* $\rho_{max} = sup\ \{\rho : \Omega \text{ is star-shaped with respect to a ball of radius } \rho\}$*. Let* $B$ *be such that its radius* $\rho > (1/2)\rho_{max}$*. Let* $Q_m u$ *be the Taylor polynomial of order* $m$ *and of total degree* $m-1$ *of* $u$ *averaged over* $B$ *(as defined in [19]) where* $u \in W^m_p(\Omega)$ *and* $p \geq 1$*. Then*

$$|u - Q_m u|_{W^k_p(\Omega)} \leq C h^{m-k} |u|_{W^m_p(\Omega)}, \quad k = 0, 1, \cdots, m, \tag{36}$$

*where* $h$ *=diam(*$\Omega$*).*

**Remark 12.** *We are going to use Bramble-Hilbert's Lemma for* $m = 1$*,* $p = 2$ *and* $k = 0$*. The Taylor polynomial* $Q_1 u$ *can be written*

$$Q_1 u = \int_B u(\mathbf{y}) \zeta(\mathbf{y})\ d(\mathbf{y}),$$

*where* $\zeta$ *is a cut-off function supported in* $\overline{B}$ *such that*

$$\int_B \zeta(\mathbf{y}) d\mathbf{y} = 1, \ \text{ and } \|\zeta\|_{L^\infty(B)} \leq C\rho^{-d}. \tag{37}$$

*Bramble-Hilbert's Lemma yields*

$$\|u - Q_1 u\|_{L^2(\Omega)} \leq C h \|u\|_{H^2(\Omega)}\,. \tag{38}$$

*Since we will consider $\mathbf{x}_K$ as the center of mass of B, lemma 7.5 from [47] on approximation of $H^2$ functions and the equation* (38) *entail*

$$\begin{aligned} \|u - u(\mathbf{x}_K)\|_{L^2(\Omega)} &\leq \|u - Q_1(u)\|_{L^2(\Omega)} + \|Q_1(u) - u(\mathbf{x}_K)\|_{L^2(\Omega)}, \\ &\leq Ch\|u\|_{H^2(\Omega)}. \end{aligned} \quad (39)$$

### 0.3.3 Chapter 3: Development of new non-intrusive tools

In this chapter, two new NIRB tools will be developed. We will apply these methods on the 2D backward-facing step with steady Navier-Stokes. The flow will be laminar and incompressible.

$$\boxed{\text{The varying parameter will be the Reynolds number } Re \in \mathcal{G} = [30, 300].}$$

1. The first tool developed during this thesis is a constrained version of the NIRB method. We consider an $L^2$ orthonormalized RB. The NIRB approximation corresponds to the $L^2$-projection of the coarse solution onto the basis. In mathematical terms, this is equivalent to the following minimization problem

   $$\min_{\alpha=(\alpha_1,\ldots,\alpha_N)} \left\| u_H(Re) - \sum_{i=1}^{N} \alpha_i \, \Phi_i^h \right\|_{L^2}. \quad (40)$$

   as a function of the coefficients. The NIRB coefficients, which correspond to the $L^2$-inner product between the coarse solution and the fine basis functions, are decreasing in absolute value. Indeed, they are linked to the POD eigenvalues, which are decreasing for each basis function (see the section on the POD 1.2.1 for further details). Thus, we can use a new algorithm to minimize the error between the rough solution and the NIRB approximation (40) as a function of the coefficients, while decreasing them. Two sorts of constraints can be applied:

   (a) The first idea is to directly impose the coefficients decrease. However, their decline is not sufficient, thus we will consider other constraints as introduced below.

   (b) The second possibility is to use the RB coefficients (10) from the snapshots ones. We then impose the new coefficients decrease by seeking them in the range of the snapshots coefficients.

   - First we will observe the coefficients decrease on the model problem.
   - Then we will recover the classical NIRB error with this new algorithm. The runtimes are compared to the classical NIRB. This allows us to stabilize the NIRB algorithm. However, this new algorithm is not accurate enough to recover the same numerical results as with the rectification post-processing.

2. Another new tool is an adaptation of the NIRB algorithm to improve the accuracy and to further reduce the computational time of the online part, in particular by using a domain truncation. To do so, we exploit the fact that the solutions to parameterized problems behave physically similarly for a suitable range of parameters. The coarse approximation is reduced twice, using two RB. This new algorithm is applied to a model flow problem which is the classical descending stair in a 2D channel.

**Main idea.** Several methods combine RBM and the domain decomposition framework [77, 78, 58]. In [58], the autors use a domain decomposition and machine learning techniques in the offline and in the online stages of the RBM algorithm. In this section, we combine a domain truncation with the NIRB two-grid method.
In order to choose the truncation, a prior knowledge of the whole domain and of the physical phenomena is required. To the best of our knowledge, this idea has not already been developed with NIRB methods. We adapt the NIRB two-grid algorithm with two reduced bases and a deterministic procedure that goes from one to the other. The main interest of this new online stage is that it greatly reduces its runtime. Indeed, during the online stage, this process allows us to consider a domain truncation to reduce the degrees of freedom of the coarse solution, and this subsequently increases the speed-up of the NIRB method. This new approach can be used with many parameter-dependent problems, even when the variable parameter depends on the geometry.

**Application to the 2D backward facing step.** In section 3.3, we will apply this new NIRB tool to the 2D backward-facing step with steady Navier-Stokes. The channel domain is supposed to be infinite, with upper and lower walls. To cope with this infinite domain in simulations, we use Neumann boundary conditions on the exit border of the channel, and Dirichlet conditions at the entrance. The length of the channel to simulate the flow is chosen such that

- we observe a Poiseuille flow and a maximum velocity at the upstream of the channel,
- a vortex at the concave corner behind the step,
- and a Poiseuille flow at the backstream of the channel.

This new approach allows us to further reduce the length of the channel during the online stage, and thus to substantially reduce the degrees of freedom. It can also lead to a reduction of the error between the fine solution and the reduced approximation, compared to the one obtained with the classical NIRB two-grid method, as shown by the results obtained with the model problem in section 3.3.2. One ongoing project is concerned with its numerical analysis.

### Description of the Non-Intrusive Reduced Basis tool with domain truncations

Let us consider an infinite domain $\Omega_\infty$ where we introduce artificial boundaries, in order to simulate the full problem. Thus, the artificial boundaries must not be too close, so as to not interfere with the modeling, and not too far, to avoid too expensive calculations. The conditions on these artificial boundaries result from the modeling phase. Thus, let $\Omega \subset \Omega_\infty$ be a bounded domain in $\mathbb{R}^d$ with $d \leq 3$, adapted to the parameterized problem. We then introduce $\omega \subset \Omega$, a truncation of $\Omega$ such that the domain $\Omega$ is cut much before the fictitious boundaries. The NIRB method with the domain truncation, in the context of a classical HF solver, involves two partitioned meshes:

- one fine mesh $\mathcal{M}_h$ on $\Omega$,
- and one coarse mesh $\mathcal{M}_H$ on $\omega$,

where $h$ and $H$ are the respective sizes of the meshes and $h << H$. The sizes $h$ and $H$ are defined as in (7).

(a) The **"offline"** part of the algorithm is costly in time. Indeed, the snapshots must be generated with the HF code on the fine mesh $\mathcal{M}_h$. Then, for the same parameter values, the corresponding coarse snapshots are computed on $\mathcal{M}_H$.
In the standard NIRB method [66, 28], a RB is created with a greedy procedure or through an SVD approach on the whole domain. Then, the coarse solution in the online part is $L^2$-projected on the RB. The coefficients are the $L^2$-inner product of the interpolated coarse solution with the fine reduced basis on $\Omega$. In [68], two POD RB are used to reduce the condition number of the snapshots rectification matrix. In our new deterministic process, in addition to the fine RB, we also create one coarse RB generated on $\omega$. Thus, there are two sorts of coefficients:
    - The fine ones, which correspond to the $L^2$-inner product of the fine snapshots with the fine reduced basis on the whole domain $\Omega$,
    - and the coarse coefficients, which are the $L^2$-inner product of the coarse snapshots with the coarse reduced basis on the subdomain $\omega$.

    Then, we involve a rectification matrix $\mathbb{T}$, which will be generated during the offline part. This rectification matrix allows us to go from the coarse coefficients to the fine ones. The coarse solutions are represented by the coarse basis well (respectively the fine solutions by the fine basis) and that is what makes this method so efficient. The offline stage is executed only once.

(b) Then, the **"online"** step is executed on the coarse mesh $\mathcal{M}_H$, and is thus much less expensive than a HF computation. Since the coarse mesh is computed on $\omega \subset \Omega$ and since no interpolation is needed, the online part is even less computationally expensive than with the usual NIRB algorithm. The coarse solution coefficient is then rectified thanks to the matrix $\mathbb{T}$, and then projected onto the fine basis. We also tested this algorithm with $\omega = \Omega$.

The entire algorithm is described in section 3.3.1. We carried out tests on the 2D backward facing step model, which is described in section 3.1. The results are presented in section 3.3.2.

### 0.3.4 Chapter 4: An offshore wind farm application

This chapter is related to the wind farm application and is a collaboration with EDF.
To begin with, let us recall technical terminology about wind turbines, illustrated with Figure 6 (picture taken from [36]). A wind turbine contains

- The rotor, which is the rotating part of the wind turbine located at the top in order to capture the winds. It is made of blades (usually three) which can measure 25 to 60 m in length.
- The hub which connects the blades.
- The nacelle which contains the alternator.
- The tower, composed of the mast, the electrical control system and the transformer.

The two-grid method has already been tested on fluid flows [30] in complex configurations but this approach is in its early stages for problems with industrial dimensions [127]. In this chapter, we will illustrate the two-grid method with a direct application on offshore wind farms.

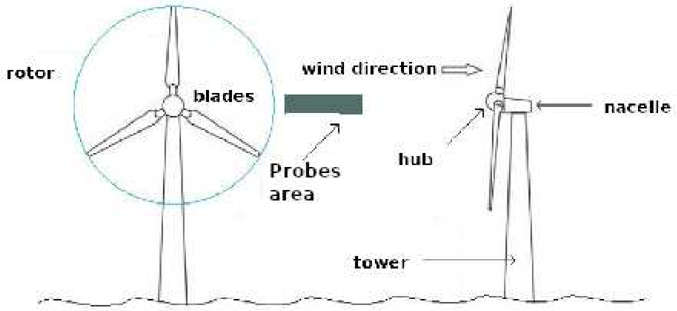


Figure 6: Wind turbine terminology

To simulate the wind farms, EDF uses a solver called Code_Saturne. **Code_Saturne** is a free computer simulation software in fluid mechanics developed by EDF, using FVM. The purpose of this solver is to simulate Computational Fluid Dynamics (CFD) for a large variety of applications [59, 91, 124]. One of them is concerned with the energy production of offshore wind farms. Its purpose is to find the best positions of the wind turbines within a wind farm to optimize its energy production. Indeed, the wind turbines wake is characterized by a velocity reduction and an increase of the turbulence behind the rotor, which will influence the energy production of other wind turbines downstream. In order to optimize the turbines positions, the wind around the turbines is simulated using the Reynolds Averaged Navier-Stokes (RANS) equations for a large range of given input velocities.

Thus, the varying parameter, denoted $u_{ref}$, is the magnitude of the input reference velocity .

Before explaining when the two-grid method operates, let us summarize the main steps of the energy power calculation of an offshore wind farm and the means implemented to maximize it. For each potential input velocity, the goal is to recover the velocity around the wind turbines or around some probes located upstream the turbine, after several time steps, to model their power (see Figure 6). The power is calculated as a function of this velocity for several wind turbines positions, in order to optimize their placement. Before the power simulation, the wind must be simulated for several initial parameters:

- First, the velocity at the entrance is obtained as a function of the altitude and the reference magnitude $u_{ref}$. Then, this input velocity is lifted to create a complete initial profile. This profile gives the velocity at the initial time on the whole domain (the wind farm).
- Then, the RANS code of code_saturne is launched for each initial profil. RANS method separates the velocity in one mean field, denoted $\overline{\mathbf{u}}$, and one fluctuating field $\mathbf{u}'$. Derived

from the continuity and the momentum equations, the RANS states:

$$\begin{cases} \frac{\partial\rho}{\partial t} + \mathrm{div}(\overline{\rho\mathbf{u}}) & = 0, \\ \rho\frac{\partial\overline{\mathbf{u}}}{\partial t} + \nabla\overline{\mathbf{u}}\cdot(\rho\overline{\mathbf{u}}) & = -\nabla\overline{p} + \mathrm{div}[\mu(\nabla\overline{\mathbf{u}} + \nabla\overline{\mathbf{u}}^T) - \frac{2}{3}tr(\nabla\overline{\mathbf{u}})Id] + \rho g - \ \mathrm{div}\ (\rho\mathbf{R}) + \overline{\mathbf{F}}, \end{cases} \tag{41}$$

where $\rho$ is the fluid density, $\overline{p}$ its averaged pressure, $\mu$ the dynamic viscosity, $g$ the gravity, and $\overline{\mathbf{F}}$ are additional momentum source terms ($tr$ is the trace of a tensor, and $Id$ the identity). In these equations, only the mean fields of the velocity and the pressure are computed. An additional term appears, namely $\mathbf{R}$, which is the Reynolds tensor and is equal to $\overline{\mathbf{u}' \otimes \mathbf{u}'}$, which corresponds to the covariance tensor of the fluctuating velocity field. This Reynolds tensor requires a closure model. For the turbines simulations, we will use the closure model $k - \varepsilon$ [8, 97].

- For each input parameter value, the specific velocity positioned upstream of a wind turbine can be retrieved and the power extracted by the turbine is calculated as a function of the thrust force and this velocity. Since we want to maximize the power production, analyzing the wind inside and upstream the rotors for several turbines is necessary. Even if the physics is simplified by the use of an actuator disk model [11, 102, 128], these simulations are still very computationally expensive. To further reduce calculation times, **non-intrusive reduced basis methods** are suitable, such as the **two-grid method**.

**Main idea.** It is well known that a wind turbine, located behind another one, will be influenced by the latter due to the wake effects. The same phenomenon will occur for a wind turbine located just downstream the first two, and, as a result, each turbine of a given line is significantly influenced by the others located upstream. Thus, we can imagine that a turbine will be fairly well approached by those located upstream it. Therefore, the purpose of this chapter is to approximate a wind farm with several wind turbines with only few of them. Several cases will be presented.

1. To get a first idea of which parameters to use for the snapshots, the first study will correspond to a simple case in 2 dimensions (thus, the 3D effects of the turbulence will not be taken into account) and one wind turbine.

2. Then, we will consider several wind turbines in line. To apply the two-grid algorithm, we focus on the rotors and probes upstream them (see Figure 6). All the areas of interest are projected on the same reference mesh.

3. The data will be complexified while reaching 3D and 3 wind turbines in line.

4. Finally, we will consider a squared wind farm with 9 wind turbines, and present several numerical results.

This study is in line with the ThorntonBank offshore wind farm operated by "C-Power". It is located at 30 km from the Belgian coast line. It is a 9 × 6 wind turbines rectangle with a total installed capacity of 325.20 MW. In this work, several simplifications are done. For instance, since we consider offshore wind farms, we did not use towers (the ground is the sea level). In the wind turbines simulations, only the rotors are represented, with actuator disks as in Figure 7.

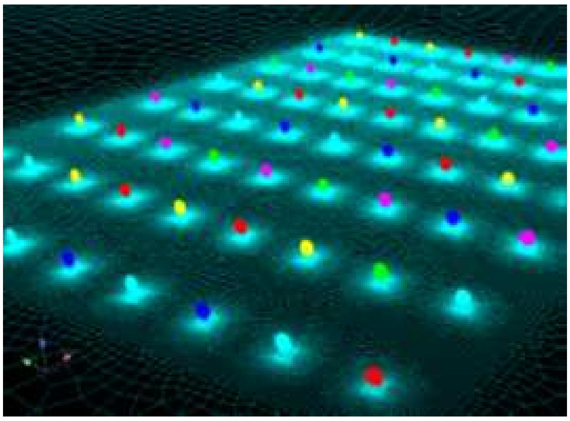


Figure 7: Wind farm with actuator disks, image from EDF.

### 0.3.5 Some conclusions and perspectives

**On the NIRB numerical analysis in the FEM context.** We detailed in this analysis the $N$-dependence of the constants in the NIRB energy-error estimate, where $N$ denotes the number of modes. We illustrated our contribution with numerical results on the classical Poisson's equations. We presented tests on the NIRB two-grid method with and without the rectification postprocessing stage. Witout the latter, the error increases quickly. On the other side, let us suppose that we chose a-priori a small tolerance and that the Relative Information Content (RIC) gave us the number of required modes. It does not mean that if we increase this number, we will get a worse approximation. However, we numerically observed this fact most of the times, due to the $N$-dependence of the constants. Thus, an open question is about a more precise analysis. It is concerned with the trade-off between increasing $N$ to obtain a more accurate manifold, and keeping the $N$-dependent constant as low as possible. The rectification allows us to significantly increase the accuracy of the NIRB approximations. Its analysis has been developed in a general context in [99]. An iteresting perspective would be to analyze more precisely the rectification in the two-grid method framework.

**On the a-posteriori error estimates.** Let us consider the model problem 16a-16b in 2D. We will denote

$$\sigma := A(\mu)\nabla u \tag{42}$$

the exact flux. This study, section B in the appendix, is an ongoing project on "a posteriori" estimates. When the exact solution is not known, there are two main ways to study the convergence of approximations.

- The first one starts by choosing an arbitrary reference mesh size. Then one observes the convergence while using several other coarser meshes tending towards the reference one. And finally, one stops when reaching a sufficient rate.
- The second possibility is to employ a-posteriori estimates.

There are different types of a-posteriori error estimates such as residual-based estimates, or equilibrated fluxes estimates, ... Several papers present results on RBM errors estimates with a residual approach [123, 132, 26]. To the best of our knowledge, the two-grid method has not already been studied with reequilibrated a-posteriori estimators. They give guaranteed upper and lower bounds for very general approximations that do not need to be solution of the discrete problem. We recalled three approaches based on a flux reconstruction [115, 49, 106]. Let us present these three approaches. We consider a flux reconstruction of $\sigma$, defined by (42), denoted $\sigma_h$ in $\mathbf{H}(\mathrm{div}, \Omega)$.

- A general approach comes from [115]. It yields a certified bound but rougher than the other methods.

- The second flux reconstruction estimate is based on an approach derived from mixed FEM theory and a reequilibration on patchwise Neumann problems, as in [18]. This work comes from [49]. To obtain this bound, the NIRB approximation needs to fulfill a Galerkin-orthogonality property, and it is not always the case.

- The last one, which has not yet been tested with the NIRB algorithm comes from a recent work [106]. With the previous flux reconstitution, an algebraic flux part is added in order to satisfy the Galerkin-orthogonality for very general approximations.

Several promising results with the two-grid method are presented with the first two in section B. However, with the second one, the approximations must fulfill a Galerkin orthogonality and it turned out that is was not often the case. Once again, the rectification enhances the results.

One perspective is to test the two-grid a-posteriori estimator with the additional algebraic flux. It comes from recent studies allowing a-posteriori estimates on very general approximations [106]. Thanks to this additional flux, Galerkin orthogonality is always satisfied leading to highly accurate bounds.

The application possibilities are numerous. For instance, to recover the estimate order in $\mathcal{O}(h)$ with the classical $\mathbb{P}_1$ FE, the coarse mesh size must be chosen such that $h = H^2$, or $h = CH^2$, where $C$ is a constant, not depending on the meshes size. Indeed, this constant can be integrated into the constant $C_2$ (25). When the exact solution is not known analytically, a way of optimizing the coarse mesh size is to use a-posteriori estimates to bound the NIRB error. Then, NIRB a-posteriori bounds may be compared to a-posteriori estimates, derived from the fine solution.

**On the parabolic equations.** We extended the two-grid method to time-dependent equations with its numerical analysis and numerical results on a model problem. This is an innovative study since the two-grid method has not already been applied to such problems. The first results are promising and as a perspective, we could apply the method to another more complex problem in order to be able to notice the computational time reduction.

**On the FV analysis.** We generalized the NIRB method to FV solvers, and developed its numerical analysis in this context with Poisson's equations. We employed recent super-convergence results on the family of Hybrid Mimetic Mixed Schemes [47]. This family under some properties contains the Two-Points Flux Approximation Finite Volumes which is often used in industrial contexts and easy to implement. Thus, we tested the two-grid method on our model problem with this scheme and we retrieved accurate resuls.

There are several open perspectives to this study. They are concerned with its generalization to other equations and its extend to other FV schemes. For instance, it could be generalized to Discontinuous Galerkin Methods thanks to recents papers on super-convergence [94]. We also want to test the NIRB method with other FV shemes where the theory does not apply.

**On the application with singular domains.** This is still an ongoing project and the object of an upcoming article. We tested this approach on two problems which are the L-shape domain and the backward-facing step. They both yield good results but not certified. Thus, open questions raise on the limits of this approach for instance with highly singular domains.

**On new NIRB tool with domain truncations.** This innovative tool is detailed in the third chapter and is the object of an upcoming article. Tests have been carried out on the backward-facing step problem. We also tested this method with the industrial application on the wind farms simulations. The results are convincing. One ongoing project is concerned with its theoritical analysis.

**On the wind farm applications.** We applied NIRB methods on wind farm simulations with two and three dimensions. The results in two dimensions were expected. They exhibit the fact that the problem becomes very easy to approach with NIRB methods.
We found a more effective way of applying NIRB methods containing two additional steps. The first one is the decomposition of the domain on several subdomains around the rotors. This preprocessing stage allows us to approximate one turbine with a RB generated with snapshots of other turbines. We tested this approach in three dimensions, first with 3 turbines, then with 9. We observed that on these problems, the NIRB method was very efficient to reduce the computational costs. One ongoing project using this idea is to approximate a $5 \times 5$ wind farm with a smaller farm. The latter will contain $3 \times 3$ turbines. The purpose is to approximate a line of turbines of the $5 \times 5$ farm with lines of turbines of the other farm.

# Chapter 1

# A review on Reduced Basis methods (RB)

This section is dedicated to a state of the art of the RBM studied during this thesis. We focus on the two-grid method, and this chapter introduces RBM and highlights the purpose of each RBM method. We apply these methods to a model problem which is the classical 2D lid driven cavity.

When approximating a solution with a HF code, the number of degrees of freedom can be huge. Thus, executing the code can be very costly in time. The RBM aims to greatly decrease the number of degrees of freedom of the problem. RBM are increasingly developed in a non-intrusive way in order to be usable by any software. We recall that the non-intrusivity means that the methods do not require any modifications in the source code of the external solver. There are several degrees of non-intrusivity. We will mainly present non-intrusive methods, but first, we will introduce RBM with a widely used intrusive method which is the Galerkin-Proper Orthogonal Decomposition. We emphasize that the two-grid RB can be generated with the POD offline stage.

## 1.1 A model problem

We start with a model problem to illustrate projection-based Model Order Reduction (MOR) methods.

Let us introduce the stationary Navier-Stokes equation in the 2D lid driven cavity problem with non-homogeneous Dirichlet boundary conditions on the upper side, homogeneous Dirichlet (no-slip) boundary conditions on the remaining sides, as in the figure 1.1. The 2D steady Navier-stokes equation writes:

$$\begin{cases} (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = 0, \text{ on } \Omega, \\ \nabla\cdot\mathbf{u} = 0, \text{ on } \Omega, \\ \mathbf{u}_{|\Gamma_3} = (1,0), \\ \mathbf{u}_{|\partial\Omega\setminus\Gamma_3} = (0,0), \end{cases} \tag{1.1}$$

where $\mathbf{u} = (u, v)$ is the velocity, $p$ is the pressure, $\Omega = [0,1]\times[0,1]$ is the unit square, and $\Gamma = \Gamma_1\cup\Gamma_2\cup\Gamma_3\cup\Gamma_4$ its boundary, where $\Gamma_3$ is the upper border, $\nu = 1/Re$, and $Re$ is the

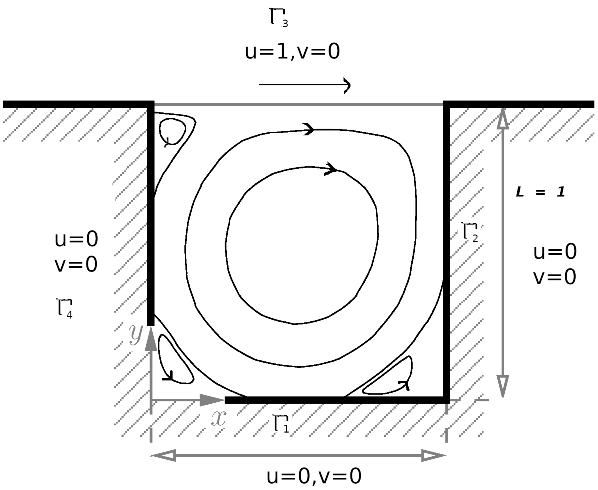


Figure 1.1: The 2D lid driven cavity problem

Reynolds number.

In this chapter, $\nu = \dfrac{1}{Re}$ will represent our varying parameter.

**FE on the model problem.** In our model problem,

$$\mathbf{V} = H^1_{d,0}(\Omega)^2 = \{\mathbf{u} \in H^1(\Omega)^2, \gamma\mathbf{u} = 0,\ \gamma_3\mathbf{u} = 1\},$$

where $\gamma$ stands for the trace operator on $\partial\Omega\backslash\Gamma_3$ and $\gamma_3$ is the trace operator on $\Gamma_3$.

We set

$$M = L^2_0(\Omega) = \{p \in L^2(\Omega), \int_\Omega p = 0\},$$

to ensure the uniqueness of the pressure. The problem 1.1 can be rewritten into its variational form:

$$\text{Find } (\mathbf{u}, p) \in \mathbf{V} \times M, \text{ such that}$$

$$\begin{cases} a(\mathbf{u}, \mathbf{v}; \nu) + b(\mathbf{v}, p) = 0, \forall \mathbf{v} \in H^1_0(\Omega)^2, \\ b(\mathbf{u}, q) = 0, \forall q \in L^2_0(\Omega), \end{cases} \tag{1.2}$$

where

$$a(\mathbf{u}, \mathbf{v}; \nu) = (\mathbf{u} \cdot \nabla)\mathbf{u}, \mathbf{v}) + \nu(\nabla\mathbf{v}, \nabla\mathbf{u}), \text{ and } b(\mathbf{u}, q) = -(\nabla \cdot \mathbf{u}, q).$$

We assume the problem is well posed and that it satisfies the so-called inf sup condition (or LBB) [20]

$$\inf_{q \in M\backslash\{0\}} \sup_{\mathbf{v} \in H^1_0\backslash\{0\}} \frac{\int_\Omega q \nabla \cdot \mathbf{v}}{\|\mathbf{v}\|_{H^1} \|q\|_{L^2}} \geq \beta > 0, \tag{1.3}$$

where $\beta$ is a constant not depending on the mesh. We first create a triangulation $\mathcal{T}_h$ of $\Omega$. With FEM, there exist several types of stable elements. A classical one is the Taylor-Hood element, where basis functions of degree $k$ are used for the pressure and basis functions of degree $k+1$ are employed for the velocities. Thus, we use Taylor-Hood $\mathbb{P}_2 - \mathbb{P}_1$ elements to solve the problem. The velocity is approximated with $\mathbb{P}_2$ FE, whereas the pressure is approximated with $\mathbb{P}_1$ FE, and we set $\mathbf{V}_h \subset \mathbf{V}$ , $\mathbf{V}_{h,0} \subset H_0^1(\Omega)^2$, $M_h \subset M$, where

$$\mathbf{V}_h = \{\mathbf{u} \in \mathbf{V}, \forall K \in \mathcal{T}_h, \mathbf{u}_{|K} \in \mathbb{P}_2\},$$

$$\mathbf{V}_{h,0} = \{\mathbf{u} \in H_0^1(\Omega)^2,\ \forall K \in \mathcal{T}_h, \mathbf{u}_{|K} \in \mathbb{P}_2\} \text{ and}$$

$$M_h = \{p \in M, \forall K \in \mathcal{T}_h, p_{|K} \in \mathbb{P}_1\}.$$

The discrete associated weak form of (1.2) is:

$$\text{Find } (\mathbf{u}_h, p_h) \in \mathbf{V}_h \times M_h, \text{ such that}$$

$$\begin{cases} a(\mathbf{u}_h, \mathbf{v}_h; \nu) + b(\mathbf{v}_h, p_h) = 0, \forall \mathbf{v}_h \in \mathbf{V}_{h,0}, \\ b(\mathbf{u}_h, q_h) = 0, \forall q_h \in M_h. \end{cases} \tag{1.4}$$

To deal with the nonlinearity, a fixed-point iteration scheme is generally used, therefore the variational form (1.4) becomes:
For each step k, find $\mathbf{u}_h^k \in \mathbf{V}_h$ and $p^k \in M_h$ such that:

$$\begin{cases} ((\mathbf{u}_h^k \cdot \nabla)\mathbf{u}_h^{k-1}, \mathbf{v}_h) + \nu(\nabla \mathbf{v}_h, \nabla \mathbf{u}_h^k) - (\nabla \cdot \mathbf{v}_h, p_h^k) = 0, \forall \mathbf{v}_h \in \mathbf{V}_{h,0}, \\ (q_h, \nabla \cdot \mathbf{u}_h^k) = 0, \forall q_h \in M_h, \end{cases} \tag{1.5}$$

until a tolerance is reached, namely until $\left\| \mathbf{u}_h^k - \mathbf{u}_h^{k-1} \right\| < tol$, where $tol$ is a priori chosen.

**Reduced Basis Methods (RBM) on the model problem.** The set of parameters is still denoted $\mathcal{G}$. To illustrate the RBM listed in this chapter, we consider the problem 1.1 as a parameterized problem where the viscosity denoted $\nu$ plays the role of the varying parameter and ranges over $\mathcal{G} \subset \mathbb{R}$. We will only display results on the velocity, but all the methods we will present also work on the pressure.
Projection-based methods all proceed in one costly offline stage and one cheap online stage. The offline stage consists in building the RB. Several types of data can be obtained and analysed (velocity fields, pressure fields, ...), yet we emphasize that we will focus on the velocity fields. We solve the problem (1.5) for several parameters and employ these snapshots for the RB construction. The Singular Value Decomposition (SVD) is explained in the next section, and we underline its link with the Proper Orthogonal Decomposition (POD). We detail this approach since it is widely used and it can also be employed during the offline part of the two-grid method. In order to use the Galerkin POD method, in a general context, we need an affine decomposition with respects to the parameter $\nu$. From the algorithm point of view, the parameter-independent terms are computed offline, making the online computation faster. If this assumption is not fulfilled, we resort to EIM presented in section A.0.1. In the model problem, the quadratic nonlinearity of the convective term does not need EIM to assemble the corresponding arrays, provided $N^2$ terms are constructed. As we will explain later on, in the NIRB two-grid method these assumptions do not need to be satisfied (see section 1.4).

## 1.2 Proper Orthogonal Decomposition (POD) Galerkin

We begin the review on RBM with the Proper Orthogonal Decomposition (POD) Galerkin method [37, 126, 14, 4, 93, 5]. The POD has been applied to a wide range of applications (turbulence, image processing applications, analysis of signal, in data compression, optimal control, ...).
We will detail its offline part, its link with the SVD and the online projection stage. We will see that the NIRB two-grid RB can be generated with the POD offline part and still be non-intrusive. There are several forms of POD (classical POD, spectral POD, ...), but here we will mainly consider the Snapshots POD algorithm. To sum up, the algorithm is as follows:

- In the offline part, the RB is built with several approximations (the snapshots) of the problem 1.1, for several well chosen parameter values. This step consists, first, in forming the snapshots correlation matrix and in retrieving its eigenvalues and its eigenvectors. Then, the RB functions are constructed by linear combinations of the first $N$ eigenvectors with the snapshots, after having sorted the eigenvalues in descending order.

- The online part consists in solving the reduced problem which uses the RB for a new parameter $\nu \in \mathcal{G}$. At the end of the algorithm, a reduced solution for $\nu$ is created.

The full algorithm is detailed in subsection 1.2.1, following its analysis.

### 1.2.1 Proper Orthogonal Decomposition: Offline stage

For one parameterized problem, the POD approximation consists in determining a basis of proper orthogonal modes which represents the best the solutions. The modes are obtained by solving an eigenvalue problem. To put it in a nutshell, the POD consists in extracting dominants modes from random data in order to be able to approximate a solution of the problem for a new parameter very quickly. Thus, we seek a function (a mode) the best correlated in average with the realizations $\{\mathbf{u}(\mathbf{X})\}$, where $\mathbf{X} = (\mathbf{x}, \nu_i)_{i=1,\dots,Ntrain} \in \Omega \times \mathcal{G}$, $Ntrain$ is the number of snapshots, $\mathbf{x} = (x_1, x_2) \in \Omega$, and $\nu$ is the varying parameter in $\mathbb{R}$ (which can also be in $\mathbb{R}^n$). In other words, we should choose a function $\mathbf{\Phi}$ which maximizes the averaged projection on the observations (the average is represented by $\bar{\cdot}$), suitably normalized in the sense of least squares, i.e. which maximizes the quantity $\overline{|(\mathbf{u}, \boldsymbol{\psi})|^2}$ (as shown after with its discrete counterpart in the paragraph on PCA (1.27)). So we end up with the following constrained optimization problem:
Find $\mathbf{\Phi}$ such that

$$\max_{\boldsymbol{\psi} \in L^2(\Omega \times \mathcal{G})} \frac{\overline{|(\mathbf{u}, \boldsymbol{\psi})|^2}}{\|\boldsymbol{\psi}\|^2} = \frac{\overline{|(\mathbf{u}, \mathbf{\Phi})|^2}}{\|\mathbf{\Phi}\|^2}, \tag{1.6}$$

with $\|\mathbf{\Phi}\|^2 = 1$ (To simplify the notation, $\|\cdot\|_{L^2(\Omega\times\mathcal{G})}$ is written $\|\cdot\|$). We can show that equation (1.6) is equivalent to an eigenvalue problem [74, 101].
Indeed, to find the maximum, we use the Lagrangian $J[\boldsymbol{\phi}] = \overline{|(\mathbf{u}, \boldsymbol{\phi})|^2} - \lambda(\|\boldsymbol{\phi}\|^2 - 1)$ and a variation $\boldsymbol{\psi}$, such that [74]:

$$\frac{d}{d\delta} J[\boldsymbol{\phi} + \delta\boldsymbol{\psi}]_{|\delta=0} = 0. \tag{1.7}$$

$$\begin{aligned}
\frac{d}{d\delta}J[\boldsymbol{\phi}+\delta\boldsymbol{\psi}]_{|\delta=0} &= \frac{d}{d\delta}[\overline{|(\mathbf{u},\boldsymbol{\phi}+\delta\boldsymbol{\psi})|^2}-\lambda(\|\boldsymbol{\phi}+\delta\boldsymbol{\psi}\|^2-1]_{|\delta=0},\\
&= \frac{d}{d\delta}[\overline{(\mathbf{u},\boldsymbol{\phi}+\delta\boldsymbol{\psi})(\boldsymbol{\phi}+\delta\boldsymbol{\psi},\mathbf{u})}-\lambda(\boldsymbol{\phi}+\delta\boldsymbol{\psi},\boldsymbol{\phi}+\delta\boldsymbol{\psi})]_{|\delta=0},\\
&= \overline{(\mathbf{u},\boldsymbol{\phi})(\boldsymbol{\psi},\mathbf{u})+(\mathbf{u},\boldsymbol{\psi})(\boldsymbol{\phi},\mathbf{u})}-\lambda((\boldsymbol{\phi},\boldsymbol{\psi})+(\boldsymbol{\phi},\boldsymbol{\psi})),\\
&= 2Re[\overline{(\mathbf{u},\boldsymbol{\psi})(\boldsymbol{\phi},\mathbf{u})}-\lambda(\boldsymbol{\phi},\boldsymbol{\psi})]. \qquad (1.8)
\end{aligned}$$

Thus, permuting mean operations and integrations in (1.8) entails:

$$\begin{aligned}
\frac{d}{d\delta}J[\boldsymbol{\phi}+\delta\boldsymbol{\psi}]_{|\delta=0} &= 2\overline{(\int_X \mathbf{u}(\mathbf{X})\cdot\boldsymbol{\psi}^*(\mathbf{X})\ d\mathbf{X})(\int_X \boldsymbol{\phi}(\mathbf{X}')\cdot\mathbf{u}^*(\mathbf{X}')\ d\mathbf{X}')}-2\lambda\int_X \boldsymbol{\phi}(\mathbf{X})\cdot\boldsymbol{\psi}^*(\mathbf{X})d\mathbf{X},\\
&= 2\int_X[\int_X \overline{\mathbf{u}(\mathbf{X})\cdot\mathbf{u}^*(\mathbf{X}')}\boldsymbol{\phi}(\mathbf{X}')d\mathbf{X}'-\lambda\boldsymbol{\phi}(\mathbf{X})]\cdot\boldsymbol{\psi}^*(\mathbf{X})d\mathbf{X}. \qquad (1.9)
\end{aligned}$$

$\boldsymbol{\psi}$ being an arbitrary variation, and because $\mathbf{u}$ are not functions here but vectors, the autocorrelation function is replaced by a tensor product matrix, and we obtain from (1.7) and (1.9)

$$\int_X \overline{\mathbf{u}(\mathbf{X})\otimes\mathbf{u}^*(\mathbf{X}')}\boldsymbol{\phi}(\mathbf{X}')d\mathbf{X}' = \lambda\boldsymbol{\phi}(\mathbf{X}). \qquad (1.10)$$

Let $\mathcal{R} : L^2(\Omega\times\mathcal{G}) \to L^2(\Omega\times\mathcal{G})$ be the operator defined by $\mathcal{R}\boldsymbol{\Phi}(X) = \int_X R(\mathbf{X},\mathbf{X}')\Phi(\mathbf{X}')d\mathbf{X}'$, where

$$R(\mathbf{X},\mathbf{X}') = \overline{\mathbf{u}(\mathbf{X})\otimes\mathbf{u}^*(\mathbf{X}')} \qquad (1.11)$$

is the correlation tensor in two points. We obtain the following eigenvalue problem:

$$\mathcal{R}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}. \qquad (1.12)$$

$\mathcal{R}$ is a positive linear compact self-adjoint operator on $L^2(\Omega\times\mathcal{G})$. Indeed,

$$\begin{aligned}
(\mathcal{R}\boldsymbol{\Phi},\boldsymbol{\Phi}) &= \int_X\int_X R(\mathbf{X},\mathbf{X}')\boldsymbol{\Phi}(\mathbf{X}')d\mathbf{X}'\cdot\boldsymbol{\Phi}^*(\mathbf{X})d\mathbf{X},\\
&= \int_X\int_X \overline{\mathbf{u}(\mathbf{X})\otimes\mathbf{u}^*(\mathbf{X}')}\boldsymbol{\Phi}(\mathbf{X}')d\mathbf{X}'\cdot\boldsymbol{\Phi}^*(\mathbf{X})d\mathbf{X},\\
&= \overline{\int_X \mathbf{u}(\mathbf{X})\cdot\boldsymbol{\Phi}^*(\mathbf{X})d\mathbf{X}\int_X \mathbf{u}^*(\mathbf{X}')\cdot\boldsymbol{\Phi}(\mathbf{X}')d\mathbf{X}'},\\
&= \overline{\|\mathbf{u},\boldsymbol{\Phi})\|^2} \geq 0.
\end{aligned}$$

In the same manner, we can show that $(\mathcal{R}\boldsymbol{\Phi},\boldsymbol{\Psi}) = (\boldsymbol{\Phi},\mathcal{R}\boldsymbol{\Psi})$ for all $(\boldsymbol{\Psi},\boldsymbol{\Phi}) \in [L^2(\Omega\times\mathcal{G})]^2$.

Therefore, the spectral theory can be applied [116] and guarantees that the maximization problem (1.6) has one unique solution equal to the largest eigenvalue of the problem (1.12) which can be reformulate as a Fredholm integral equation

$$\int_X R(\mathbf{X},\mathbf{X}')\Phi_n(\mathbf{X}')d\mathbf{X}' = \lambda^n\Phi_n(\mathbf{X}), \qquad (1.13)$$

As a consequence, there exists a countable family of solutions $\{\lambda^n,\Phi_n\}$ to equation (1.13) which represent the eigenvalues and the POD eigenvectors of order $n = 1,\ldots,+\infty$.

The Hilbert-Schmidt theory allows us to rewrite $R$ as a convergent series:

$$R(\mathbf{X},\mathbf{X}') = \sum_{n=1}^{+\infty} \lambda_n \mathbf{\Phi}_n(\mathbf{X})\mathbf{\Phi}_n^*(\mathbf{X}'). \tag{1.14}$$

The $(\mathbf{\Phi}_n)_{n=1,\ldots,+\infty}$ are orthogonal, and the eigenvalues are all positives.

The POD eigenfunctions define a orthonormal basis of the reduced space. Thus, any realization of $\mathbf{u}$ can be expressed as a linear combination of these functions:

$$\mathbf{u}(\mathbf{X}) = \sum_{n=1}^{+\infty} a_n \mathbf{\Phi}_n(\mathbf{X}), \tag{1.15}$$

where the coefficients $a_n$ are defined as

$$a_n = \overline{(\mathbf{u},\mathbf{\Phi})} = \int_X \mathbf{u}(\mathbf{X})\mathbf{\Phi}(\mathbf{X})d\mathbf{X}. \tag{1.16}$$

They are such that

$$\overline{a_m a_n^*} = \delta_{mn}\lambda_n, \tag{1.17}$$

where the $\lambda_n$ represent the turbulent kinetic energy in average on each mode $\mathbf{\Phi}_n$. Indeed, from (1.15),

$$\begin{aligned} R(\mathbf{X},\mathbf{X}') &= \overline{(\sum_{n=1}^{+\infty} a_n \mathbf{\Phi}_n(\mathbf{X}))(\sum_{m=1}^{+\infty} a_m^* \mathbf{\Phi}_m^*(\mathbf{X}'))}, \\ &= \sum_{n,m=1}^{+\infty} \overline{a_n a_m^*} \mathbf{\Phi}_n(\mathbf{X})\mathbf{\Phi}_n^*(\mathbf{X}'). \end{aligned} \tag{1.18}$$

and from (1.14) and the orthonormality of the eigenvalues, it entails (1.17).

Therefore, the maximization problem permits to keep the most turbulent kinetic energy on average represented by the eigenvalues $\lambda_n$. The number of modes required for the RB can be given by the Relative Information Content (RIC) [3], denoted $I(N)$. The RIC is defined as the ratio between the $N$ first eigenvalues and their total sum and must be close to one.

$$I(N) = \frac{\sum_{k=1}^{N} \lambda_k}{\sum_{i=1}^{N_{train}} \lambda_i}. \tag{1.19}$$

We recall in the next section basic notions on the Singular Value Decomposition (SVD) of a matrix and its link with the Proper Orthogonal Decomposition (POD) and the Principal Component Analysis (PCA) [109].

### Singular Value Decomposition (SVD)

The singular value decomposition (SVD) is widely used in low-rank approximation.
The SVD consists in factorizing a matrix $\mathbb{A}$ of size $m \times n$ under the form $\mathbb{A} = U\Sigma V^T$,where $\mathbb{A} \in \mathbb{R}^{m\times n}$ is a real matrix, $\mathbb{U} = [\mathbf{\Phi}_1|\ldots|\mathbf{\Phi}_m] \in \mathbb{R}^{m\times m}$ and $\mathbb{V} = [\mathbf{\Psi}_1|\ldots|\mathbf{\Psi}_m] \in \mathbb{R}^{n\times n}$ are unitary matrices with $\Sigma = \mathrm{diag}(\sigma_1,\ldots,\sigma_p) \in \mathbb{R}^{m\times n}$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, with

$p = \min(m,n)$. The $(\sigma_i)_{i=1,\ldots,p}$ are called singular values of $\mathbb{A}$, $(\boldsymbol{\Phi}_i)_{i=1,\ldots,m}$ are left singular vectors of $\mathbb{A}$, whereas $(\boldsymbol{\Psi}_i)_{i=1,\ldots,n}$ are its right singular vectors, such that:

$$\mathbb{A}\boldsymbol{\Psi}_i = \sigma_i \boldsymbol{\Phi}_i, \text{ and } \mathbb{A}^T\boldsymbol{\Phi}_j = \sigma_j \boldsymbol{\Psi}_j, \;\; i,j = 1,\ldots,n. \tag{1.20}$$

We also have the following spectral decomposition:

$$\mathbb{A}\mathbb{A}^T = \mathbb{U}\Sigma\Sigma^T\mathbb{U}^T, \text{ and } \mathbb{A}^T\mathbb{A} = \mathbb{V}\Sigma^T\Sigma\mathbb{V}^T, \tag{1.21}$$

with

$$\Sigma\Sigma^T = \mathrm{diag}(\sigma_1^2,\ldots,\sigma_p^2, \underbrace{0,\ldots,0}_{m-p \text{ times}}),$$

and

$$\Sigma^T\Sigma = \mathrm{diag}(\sigma_1^2,\ldots,\sigma_p^2, \underbrace{0,\ldots,0}_{n-p \text{ times}}).$$

The rank of a diagonal matrix is equal to the number of non zero diagonal terms, and rank($\mathbb{A}$)=rank($\Sigma$), such that if $\mathbb{A} \in \mathbb{R}^{m\times n}$ has r positive singular values, then rank$(\mathbb{A}) = r$. We deduce that $\mathbb{A}$ can be written as the following sum:

$$\mathbb{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{\Phi}_i \boldsymbol{\Psi}_i^T. \tag{1.22}$$

If the singular values decrease fast enough, we can expect a good approximation of $\mathbb{A}$ with a smaller rank (see the truncated SVD [135]). The Eckart–Young–Mirsky theorem states that for a smaller rank $N < r$, the error between $\mathbb{A}$ and $\mathbb{A}_N = \sum_{i=1}^{N} \sigma_i \boldsymbol{\Phi}_i \boldsymbol{\Psi}_i^T$ is given by [110]

$$\|\mathbb{A} - \mathbb{A}_N\|_F^2 = \sum_{k=N+1}^{r} \sigma_k^2,$$

with $\|\cdot\|_F$ the Frobenius norm. We denote by $\lambda_i = \sigma_i^2$ the POD eigenvalues. Thus, if the POD eigenvalues are decreasing fast enough, the RB construction is efficient through a SVD process to recover with few modes any solution of the manifold.

### Link with the Proper Orthogonal Decomposition (POD)

In the offline stage, the POD algorithm requires the generation of the eigenvalues and eigenvectors of the snapshots correlation matrix. This step corresponds to a SVD (or generally to a truncated SVD). Indeed, before digging into details of the POD algorithm, let us consider $Ntrain$ snapshots $(\mathbf{u}_1,\ldots,\mathbf{u}_{Ntrain})$, with $\mathcal{N}$ degrees of freedom, for $Ntrain$ different parameters. The dataset representation is given by $\mathbb{A} = [\mathbf{u}_1|\ldots|\mathbf{u}_{Ntrain}] \in \mathbb{R}^{\mathcal{N}\times Ntrain}$.
According to equation (1.21),

$$\mathbb{A}^T\mathbb{A}\boldsymbol{\Psi}_i = \sigma_i^2\boldsymbol{\Psi}_i, \; i = 1,\ldots,r, \tag{1.23}$$

where $\sigma_i^2, i = 1,\ldots,r$ are the nonzero eigenvalues of the correlation matrix $\mathbb{C} = \mathbb{A}^T\mathbb{A}$ ($\mathbb{C}_{i,j} = \mathbf{u}_i^T\mathbf{u}_j$, $1 \le i,j \le Ntrain$). For any $N \le Ntrain$, the $N$ POD eigenfunctions are defined as the set of the first $N$ left singular vectors $(\boldsymbol{\Phi}_1,\ldots,\boldsymbol{\Phi}_N)$ of $\mathbb{U}$, and from equation (1.20),

$$\boldsymbol{\Phi}_i = \frac{1}{\sigma_i}\mathbb{A}\boldsymbol{\Psi}_i, \; i \le N, \tag{1.24}$$

obtained from the first $N$ eigenvectors $(\boldsymbol{\Psi}_1,\ldots,\boldsymbol{\Psi}_N)$ of the correlation matrix $\mathbb{C}$.

**Link with Principal Component Analysis (PCA)**

The Principal Component analysis (PCA) is employed for multivariate data in statistical analysis, while the POD has first been introduced for the simulation of turbulent flows [83, 61]. The PCA, also known as the Karhunen–Loève Transformation (KLT), has first been introduced in [107], and is now widely used in image processing. Its purpose is to reduce the dimensionality of multivariate data. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, called principal components. It can be proven that PCA follows directly from the SVD. Indeed, the principal components of $\mathbb{A}^T\mathbb{A}$ are the columns of matrix $\mathbb{V}$ 1.21.
We recall the main principle:
Let $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ represents $m$ independent observations of variables $X_1, \dots, X_n, m > n$. We set

$$\mathbb{A} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_m^T \end{pmatrix} \in \mathbb{R}^{m \times n}$$

the representation of this dataset, and

$$\mathbb{C} = \mathbb{A}^T\mathbb{A} \in \mathbb{R}^{n \times n}$$

the correlation matrix. To approximate these data using a unit-length vector $\mathbf{w_1} \in \mathbb{R}^n$, a natural choice is to minimize the squared approximation error. In mathematical terms, it is equivalent to the minimization problem:
Find $\mathbf{w}_1 \in \mathbb{R}^n$ such that

$$\begin{aligned}\mathbf{w}_1 &= \arg\min_{\mathbf{w}\in\mathbb{R}^n} \sum_{i=1}^{m} \left\| \mathbf{x}_i - \mathbf{w}\mathbf{w}^T\mathbf{x}_i \right\|_2^2, \\ &= \arg\min_{\mathbf{w}\in\mathbb{R}^n} \sum_{i=1}^{m} \left\| \mathbf{x}_i^T - \mathbf{x}_i^T\mathbf{w}\mathbf{w}^T \right\|_2^2, \text{ using the transpose,} \end{aligned} \tag{1.25}$$

subject to $\|\mathbf{w}\|_2 = 1$.
Using the Frobenius norm, this is equivalent to
Find $\mathbf{w}_1 \in \mathbb{R}^n$ such that

$$\mathbf{w}_1 = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \left\| \mathbb{A} - \mathbb{A}\mathbf{w}\mathbf{w}^T \right\|_F^2,$$

with $\mathbf{w}^T\mathbf{w} = 1$. Using the trace operator and removing the term that does not depend on $\mathbf{w}$, it yields

$$\mathbf{w}_1 = \arg\min_{\mathbf{w}\in\mathbb{R}^n} - 2\mathrm{Tr}(\mathbb{A}^T\mathbb{A}\mathbf{w}\mathbf{w}^T) + \mathrm{Tr}(\mathbf{w}\mathbf{w}^T\mathbb{A}^T\mathbb{A}\mathbf{w}\mathbf{w}^T), \tag{1.26}$$

subject to $\mathbf{w}^T\mathbf{w} = 1$.
Using the constraint, the equation 1.26 becomes:

$$\begin{aligned}\mathbf{w}_1 &= \arg\min_{\mathbf{w}\in\mathbb{R}^n} - \mathrm{Tr}(\mathbb{A}^T\mathbb{A}\mathbf{w}\mathbf{w}^T), \text{ subject to } \mathbf{w}^T\mathbf{w} = 1, \\ &= \arg\max_{\mathbf{w}\in\mathbb{R}^n} \mathrm{Tr}(\mathbf{w}^T\mathbb{A}^T\mathbb{A}\mathbf{w}), \text{ subject to } \mathbf{w}^T\mathbf{w} = 1, \\ &= \arg\max_{\mathbf{w}\in\mathbb{R}^n} \mathrm{Tr}(\mathbf{w}^T\mathbb{C}\mathbf{w}), \text{ subject to } \mathbf{w}^T\mathbf{w} = 1, \end{aligned} \tag{1.27}$$

We consider centered data around 0 so as to not make the notations cumbersome. For any $\mathbf{w} \in \mathbb{R}^n$, $\mathbb{A}\mathbf{w}$ has $\mathbf{w}^T\bar{\mathbf{x}}$ as mean (where $\bar{\mathbf{x}}$ is the mean of $\mathbf{x}$), and $\mathbf{w}^T\mathbb{C}\mathbf{w}$ as variance.
So minimizing the error between the reconstructed and the actual data and maximizing the variance of the components are equivalent. In order to decorrelate the dimensions, the ideal covariance matrix for the variance maximization and the covariance minimization is a diagonal matrix. Thus, the diagonalization of the covariance matrix gives us the optimal solution (as with the SVD).
The maximum variance corresponds to the largest eigenvalue $\lambda_1$ of $\mathbb{C}$, and is reached for the choice $\mathbf{w}_1 = \mathbf{\Psi}_1$, where $\mathbf{\Psi}_1$ is the eigenvector of $\mathbb{C}$ associated with $\lambda_1$. So the first principal component is given by $\mathbf{\Phi}_1 = \mathbf{\Psi}_1^T\mathbf{x}$. The others components are constructed in the same manner, but such that they are orthogonal to the preceding components. Finally, PCA diagonalizes the correlation matrix $\mathbb{C}$, and the principal components $(\mathbf{\Phi}_1, \dots, \mathbf{\Phi}_n)$ are such that

$$\mathrm{Var}(\mathbf{\Phi}_k) = \lambda_k,\ k = 1, \dots, n, \ \text{ and } \mathrm{Cov}(\mathbf{\Phi}_k, \mathbf{\Phi}_l) = 0,\ 1 \leq k, l \leq n \text{ with } k \neq l.$$

Then, a new data can be expressed in the new coordinate system, whose center is the sample mean $\bar{\mathbf{x}}$, and whose axes are given by the eigenvectors of $\mathbb{C}$. From the equation 1.6, it is obvious that the POD and the PCA are two "twin" methods.

**POD reduced basis algorithm: Snapshots method (Offline stage)**

At the beginning of section 1.2.1, a general analysis of the offline POD is recalled, although there exist two main kinds of POD: one is classical, and one is called the snapshots POD ([125]). The classical method consists in replacing the mean $\overline{(\cdot)}$ by an average over the parameter of interest (usually over the time variable, but in our case, we consider a stationary model problem). On the contrary, the snapshots POD replaces the mean as a space mean over the spatial domain $\Omega$. The snapshots POD, widely used, is chosen when the number of required snapshots ($N$) to construct the basis is much smaller than the degrees of freedom on $\Omega$. In what follows, we will only employ the Snapshots POD. Let us first detail its implementation.

The snapshots POD is obtained from the equation (1.10) where the average is evaluated as a spatial average on the domain $\Omega$ and the variable $\mathbf{X}$ is related to the parameters $\nu$. We denote $Ntrain$ the number of training snapshots ($N <= Ntrain$). The reduced space is denoted $X_N$ and can be represented either by the POD eigenvectors or by the snapshots which define two basis of $X_N$. Thus, the POD eigenfunctions $\mathbf{\Phi}$ can be written as a linear combinaison of the dataset:

$$\mathbf{\Phi}_i = \sum_{k=1}^{Ntrain} a_k^i \mathbf{u}_k, i = 1, \dots, N$$

where $(a_k)_{k=1,\dots,Ntrain}$ remains to be determined. With equation (1.10), we have

$$\frac{1}{Ntrain} \sum_{k=1}^{Ntrain} (\mathbf{u}_i, \mathbf{u}_k)\ \mathbf{\Phi}_k = \lambda \mathbf{\Phi}_i, i = 1, \dots, Ntrain. \tag{1.28}$$

Therefore, replacing $\mathbf{\Phi}$ by its new expression in equation (1.28) yields

$$\frac{1}{Ntrain} \sum_{k=1}^{Ntrain} (\mathbf{u}_i, \mathbf{u}_k) a_i^k = \lambda a_i^i, \quad \forall i = 1, \dots, Ntrain. \tag{1.29}$$

Let us now describe the algorithm in detail:

1. The first step consists in solving numerically the equations of the problem (1.5) for several training parameters. At the end of this step, we obtain a vector of snapshots $(u_h^1, \ldots, u_h^{Ntrain})$, where $h$ corresponds to the mesh size.

   **Remark 13.** *In order to reduce the snapshots covariance, we may decompose the snapshots by one average over the parameters, denoted* $\mathbf{u}_{h,m}$*, and by one fluctuation part written* $\mathbf{u}_{h,f}$*. Then, the POD modes are estimated with the fluctuations.*

$$\mathbf{u}_h(\mathbf{x}, \nu) = \mathbf{u}_{h,m}(\mathbf{x}) + \sum_{i=1}^{N} a_h^i(\nu)\mathbf{\Phi}_h^i(\mathbf{x}), \tag{1.30}$$

   which is equivalent to:

$$\sum_{i=1}^{N} a_h^i(\nu)\mathbf{\Phi}_h^i(\mathbf{x}) = \mathbf{u}_{h,f}(\mathbf{x}, \nu), \tag{1.31}$$

   where $\mathbf{u}_{h,f}(\mathbf{x}, \nu) = \mathbf{u}_h(\mathbf{x}, \nu) - \mathbf{u}_{h,m}(\mathbf{x})$ is the fluctuation.

2. Then, we calculate the correlation matrix $\mathbb{C}_{i,j} = \int_\Omega \mathbf{u}_{h,f}^i \cdot \mathbf{u}_{h,f}^j$, and we solve the $Ntrain \times Ntrain$ eigenvalue problem: $\mathbb{C}\mathbf{v}_h^n = \lambda_n \mathbf{I}_d \mathbf{v}_h^n$, for $n = 1 ... Ntrain$, where $\mathbf{v}_h^n = (a_{h,1}^n, \ldots, a_{h,Ntrain}^n)$.

3. Suppose the eigenvalues are well ordered ($\lambda_1 > ... > \lambda_{Ntrain} > 0$), we calculate the $N$ RB functions normalized, with $N \leq Ntrain$, the number of required modes:

$$\mathbf{\Phi}_h^i = \sum_{j=1}^{Ntrain} a_{h,i}^j \mathbf{u}_{h,f}^j, \forall i = 1, ..., N, \tag{1.32}$$

$$\mathbf{\Phi}_h^i = \frac{\mathbf{\Phi}_h^i}{\left\|\mathbf{\Phi}_h^i\right\|}, \text{(which is equivalent to dividing by } \sqrt{\lambda_i}). \tag{1.33}$$

   For the eigenvalues that are too small, we can use a further step which consists in a Gram-Schmidt procedure to ensure the basis orthonormality.

$$\mathbf{\Phi}_h^i = \mathbf{\Phi}_h^i - \sum_{j=1}^{i-1} (\mathbf{\Phi}_h^i, \mathbf{\Phi}_h^j)\mathbf{\Phi}_h^j. \tag{1.34}$$

   The number $N$ of functions in the POD basis is chosen, such that $N$ is small enough and $I(N) = \frac{\sum_{k=1}^{N} \lambda_k}{\sum_{k=1}^{Ntrain} \lambda_k}$ is close to 1. The following algorithm 1 summarizes the offline stage of the POD algorithm to create the RB.

```
input : $\mathcal{G} = (\nu_1, \ldots, \nu_{Ntrain}) \in \mathbb{R}^{Ntrain}$
output: $\mathbf{\Phi}_h^1, \ldots, \mathbf{\Phi}_h^N$, where $N \leq Ntrain$
Initialization: Construct the snapshots $(\boldsymbol{u}_h^1(\boldsymbol{x}), \cdots, \boldsymbol{u}_h^{Ntrain}(\boldsymbol{x}))$
if POD on the fluctuations then
    Set $\mathbf{u}_{h,m}(\mathbf{x}) = \frac{1}{Ntrain} \sum_{i=1}^{Ntrain} \mathbf{u}_h^i(\mathbf{x})$;
    for i=1:Nt do
        $\mathbf{u}_{h,f}^i = \mathbf{u}_h^i - \mathbf{u}_{h,m}$;
    end
end
Calculate the correlation matrix $\mathbb{C}$:
for i=1:Ntrain do
    for j=1:Ntrain do
        $C_{i,j} = \int_\Omega \mathbf{u}_{h,f}^i \cdot \mathbf{u}_{h,f}^j$;
    end
end
Calculate the eigenvalues $\lambda_1, ..., \lambda_{Ntrain}$ and eigenvectors $\boldsymbol{v}_h^1, ..., \boldsymbol{v}_h^{Ntrain}$:
for i=1:Ntrain do
    $C\mathbf{v}_h^i = \lambda_i \mathbf{I}_d \mathbf{v}_h^i$
end
Obtain the POD functions by linear combination of the snapshots and the eigenvectors:
for i=1:N do
    $\mathbf{\Phi}_h^i = \sum_{j=1}^{Ntrain} a_i^j \mathbf{u}_{h,f}^j$
    Normalize:
    $\mathbf{\Phi}_h^i = \frac{\mathbf{\Phi}_h^i}{\left\|\mathbf{\Phi}_h^i\right\|}$
    Use a Gram-Schmidt procedure for the small eigenvalues:
    $\mathbf{\Phi}_h^i = \mathbf{\Phi}_h^i - \sum_{j=1}^{i-1} (\mathbf{\Phi}_h^i, \mathbf{\Phi}_h^j) \mathbf{\Phi}_h^j$
    Stop if $I(N) \approx 1$ or use of an other error estimate
end
```

**Algorithm 1:** OFFLINE Snapshots POD algorithm

### 1.2.2 POD-Galerkin Projection on the reduced model (Online stage)

During the last step of the offline stage, the RB $(\mathbf{\Phi}_h^i)_{i=1,\ldots,N} \in \mathbf{V}_h^N \subset \mathbf{V}_{div} = \{v \in \mathbf{V}_h : \nabla.v = 0\}$ is generated. To predict the velocity $\mathbf{u}$ for a new parameter $\nu$, a standard method consists in using a Galerkin projection onto this RB.
This stage, which is intrusive, is much faster than HF codes. The assembling reduced order matrices can be computed offline, and therefore only the new problem with these matrices needs to be solved in the online phase. In what follows, the Galerkin-projection for the velocity field does not contain the pressure field. Indeed, in our model problem, we only use the ROM to derived an approximation on the velocity in the reduced space $\mathbf{V}_h^N$, since here, with the Dirichlet type boundary conditions, the basis functions satisfy both the boundary conditions and the divergence-free constrain of the continuity equation. Let us consider the equation (1.5). Using the equation (1.30)

$$\mathbf{u}_{h,f}^k(\mathbf{x}) = \sum_{j=1}^N a_{h,k}^j \mathbf{\Phi}_h^j(\mathbf{x}),$$

and

$$\mathbf{u}_{h,f}^{k-1}(\mathbf{x}) = \sum_{j=1}^N a_{h,k-1}^j \mathbf{\Phi}_h^j(\mathbf{x}),$$

and $\mathbf{\Phi}_h^i$ as the test function, we get the following equation on the coefficients:

$$\mathcal{A}_i + \sum_{j=1}^N \mathcal{B}_{ij} a_{h,k}^j + \sum_{j=1}^N \sum_{l=1}^N \mathcal{C}_{ijl} a_{h,k}^j a_{h,k-1}^l = 0, \tag{1.35}$$

with

$$\mathcal{A}_i = (\mathbf{\Phi}_h^i, \mathbf{u}_{h,m} \cdot \nabla \mathbf{u}_{h,m}) + \nu(\nabla \mathbf{\Phi}_h^i, \nabla \mathbf{u}_{h,m}),$$
$$\mathcal{B}_{ij} = (\mathbf{\Phi}_h^i, \mathbf{u}_{h,m} \cdot \nabla \mathbf{\Phi}_h^j) + (\mathbf{\Phi}_h^i, \mathbf{\Phi}_h^j \cdot \nabla \mathbf{u}_{h,m}) + \nu(\nabla \mathbf{\Phi}_h^i, \nabla \mathbf{\Phi}_h^j),$$
$$\mathcal{C}_{ijl} = (\mathbf{\Phi}_h^i, \mathbf{\Phi}_h^j \cdot \nabla \mathbf{\Phi}_h^l).$$

For the online stage, we set the new parameter of interest $\nu$, and with the notations

$$M_{ij} = \mathcal{B}_{ij} + \sum_{k=1}^N \mathcal{C}_{ijl} a_{h,k-1}^l$$

and

$$b_i = \mathcal{A}_i,$$

the coefficients $(a_{h,k}^j)_{j=1,\ldots,N}$ at iteration $k$ are obtained by solving the equation:

$$\mathbf{a_{h,k}} = \mathbf{M}^{-1}\mathbf{b}, \tag{1.36}$$

and we iterate on the residual $\|\mathbf{a_{h,k}} - \mathbf{a_{h,k-1}}\|_2$ until reaching a small treshold (where $\|\cdot\|_2$ stands for the Euclidian $l^2$-norm) in order to obtain $\mathbf{a_h}$. Finally, the approximation is given by

$$\mathbf{u}_h(\nu) \simeq \mathbf{u}_{h,m} + \sum_{j=1}^N a_h^j \mathbf{\Phi}_h^j. \tag{1.37}$$

We will present numerical results of this method in Figure 1.2.

## 1.3 POD Interpolation (PODI)

This method is a non-intrusive version of the Snapshot POD [113, 57]. The offline part to create the basis functions remains the same. Then, a further step is added. It consists in computing the coefficients for all the original snapshots with a projection (defined by (9)). We denote by $\alpha_i^h(\mu_k), i = 1, \dots, N,\ k = 1, \dots, Ntrain$ these coefficients. We obtain $N$ pairs $(\mu_k, \alpha_i^h(\mu_k))$. Thanks to a Gaussian process regression, the function that maps the input parameters $\mu_k$ to the coefficients can be reconstructed. This function is then used during the online stage to find the interpolated new coefficients for a new given parameter $\nu \in \mathcal{G}$. Finally, the high-dimensional solution is computed by projecting the new coefficients to the original space with the equation (1.15). We present numerical results on the classical method (Figure 1.2) although several enhancements may be added. For instance, a prior sensitivity analysis of the function of interest with respect to the parameters can be done. This preprocessing phase corresponds to the active subspaces property [40, 35].

## 1.4 The two-grid method

This method will be also detailed in the next chapter along with its numerical analysis. It has been developed and analyzed (with Céa's and Aubin-Nitsche's lemmas) in the context of FEM in [96]. As explained in the introduction, its name comes from the fact that it uses two meshes. One fine mesh is employed for the construction of the RB, and an coarse mesh is used to approximate the solution with a classical solver, for instance with FEM or with FV schemes. As other RBM, the two-grid method consists in two stages:

- In the first place, the RB functions are prepared in an "offline" stage with the fine mesh, involving a greedy algorithm 2 or a POD procedure 1.
- Then a coarse approximation of the solution for a new parameter value that is of interest to us is computed "online". This rough approximation is not of sufficient precision but can be calculated with a smaller number of degrees of freedom compared to the fine mesh. This approximation is then $L^2$-projected onto the RB, and other post-processing steps can be added, such as the rectification method 0.2.2 [30], making it possible to notably improve precision.

### 1.4.1 NIRB two-grid algorithm

This section recalls the main steps of the two-grid method algorithm [96, 30].
We emphasize on the fact that only the velocity fields are approximated in what follows even if the method also works with the pressure. In the Galerkin POD 1.2, the ROM could require a stabilization term [6, 7]. With the two-grid method, if the FEM solutions satisfy the inf-sup condition (1.3) which ensures stability, since it consists in projecting such solutions and not solving a reduced model, there are no need of additional stabilization terms. The variable parameter is still denoted $\nu$. Let $\mathbf{u}_h(\nu)$ be the solution approximation computed on a fine mesh $\mathcal{T}_h$, with a classical method, and respectively $\mathbf{u}_H(\nu)$ be the solution approximation computed on the coarse mesh $\mathcal{T}_H$.

We briefly recall the NIRB algorithm. Points 1 and 2 are performed in the offline part, and the others are done online.

1. Several snapshots $\{\mathbf{u}_h(\nu_i)\}_{i\in\{1,\ldots Ntrain\}}$ are computed with a classical solver for the problem (1.5), where $\nu_i \in \mathcal{G} \quad \forall i = 1, \cdots, Ntrain$.
2. We usually generate the basis functions $(\Phi_i^h)_{i=1,\cdots,N}$ and the reduced space

   $$X_h^N = Span\{\Phi_i^h(\nu_1), \ldots, \Phi_i^h(\nu_N)\}$$

   with the following steps, with $N \leq Ntrain$ (Greedy algorithm 2):

   - A Gram-Schmidt procedure is used, which involves $L^2$-orthonormalization of the basis functions.
   - This procedure is completed by the following eigenvalue problem:

     $$\begin{cases} \text{Find } \Phi^h \in X_h^N, \text{ and } \lambda \in \mathbb{R} \text{ such that:} \\ \forall \mathbf{v} \in X_h^N, \displaystyle\int_\Omega \nabla\Phi^h \cdot \nabla\mathbf{v}\, d\mathbf{x} = \lambda \int_\Omega \Phi^h \cdot \mathbf{v}\, d\mathbf{x}. \end{cases} \tag{1.38}$$

     We get an increasing sequence of eigenvalues $\lambda_i$, and orthogonal eigenfunctions $(\Phi_i^h)_{i=1,\cdots,N}$, orthonormalized in $L^2(\Omega)$ and potentially orthogonalized in $H^1(\Omega)$, such that $(\Phi_i^h)_{i=1,\cdots,N}$ defines a new basis of the space $X_h^N$.
3. We compute the approximation $\mathbf{u}_H$ of the problem (1.5) on the coarse mesh $\mathcal{T}_H$, for a new parameter $\nu \in \mathcal{G}$, with the classical solver.
4. We consider the coarse coefficients $\alpha_i^H(\nu) = \int_\Omega \mathbf{u}_H(\nu) \cdot \Phi_i^h\, d\mathbf{x}$. The approximation used in the two-grid method for the problem (1.1) is

   $$\mathbf{u}_{Hh}^N(\nu) = \sum_{i=1}^{N} \alpha_i^H(\nu)\, \Phi_i^h. \tag{1.39}$$

For the RB generation, a Greedy algorithm is usually employed in order to correctly choose the RB parameters 2. By RB parameters, we mean the parameters required for the RB generations. In general terms, a greedy algorithm is a procedure which aims at approximating each element of a compact set (here the manifold $\mathcal{S}_h$) in a Hilbert space $V_h$ by a subspace of properly selected elements of this compact set. The greedy procedure is a fast way to compute the modes by choosing some suitable parameters with respect to a criterion. An other way to generate the RB consists in using the Snapshots POD offline (algorithm 1).

At each Greedy iteration, the new RB parameter is chosen such that the corresponding snapshot is the worse approached by the previous basis space. In general, this strategy does not produce an optimal solution.

The rectification post-treatment, which is detailed in the introduction 0.2.2, is more efficient combined to a greedy algorithm since the basis functions are directly generated with the snapshots [96, 99]. A Gram-Schmidt procedure may be involved to obtain an orthonormal basis [22] and improve the RB space from the one composed of the snapshots $(\mathbf{u}_h(\nu_i))_{i=1,\ldots,N}$. Moreover, the eigenproblem (1.38) can be solved in order to orthogonalize the basis functions in $H^1$. Therefore, in order to obtain an optimal set of parameters $\{\nu_1, \ldots, \nu_N\}$ from a given sample $\mathcal{G}$, we resort to the following greedy algorithm 2

```
Data: $\{\mathbf{u}_h(\nu_1), \cdots, \mathbf{u}_h(\nu_{N_{train}})\}$
Result: Reduced basis $\{\Phi_1^h, \cdots, \Phi_N^h\}$
Choose $\nu_1 = \arg\max_{\nu \in \mathcal{G}} \|\mathbf{u}_h(\nu)\|_{L^2}$ , Set $\mathcal{G}_1 = \nu_1$ and $X_h^1 = \text{span}\{\mathbf{u}_h(\nu_1)\}$ ;
Set $\Phi_1 = \frac{\mathbf{u}_h(\nu_1)}{\|\mathbf{u}_h(\nu_1)\|_{L^2}}$ ;
for $n = 2$ to $N$ do
    $\nu_n = \arg \max_{\nu \in \mathcal{G} \setminus \mathcal{G}_{n-1}} \frac{\|\mathbf{u}_h(\nu) - P^{n-1}(\mathbf{u}_h(\nu))\|_{L^2}}{\|\mathbf{u}_h(\nu)\|_{L^2}}$ ;
    Set $\mathcal{G}_n = \mathcal{G}_{n-1} \cup \nu_n$ and $X_h^n = X_h^{n-1} + span\{\mathbf{u}_h(\nu_n)\}$ ;
    Compute $\widetilde{\Phi_n} = \mathbf{u}_h(\nu_n) - \sum_{i=1}^{n-1} (\mathbf{u}_h(\nu_n), \Phi_i)_{L^2} \Phi_i$ and set $\Phi_n = \frac{\widetilde{\Phi_n}}{\|\widetilde{\Phi_n}\|_{L^2}}$ ;
    Stop when $\frac{\|\mathbf{u}_h(\nu) - P^{n-1}(\mathbf{u}_h(\nu))\|_{L^2}}{\|\mathbf{u}_h(\nu)\|_{L^2}} \leq tol$ ;
end
```

**Algorithm 2:** Greedy algorithm used to choose $\{\nu_1, \cdots, \nu_N\}$

In the algorithm 2, the NIRB fine projection is defined by

$$P^N(\mathbf{u}_h(\nu)) = \sum_{i=1}^{N} (\mathbf{u}_h(\nu), \Phi_i^h)_{L^2} \, \Phi_i^h.$$

and the term

$$\left\| \mathbf{u}_h(\nu) - P^{n-1}(\mathbf{u}_h(\nu)) \right\|_{L^2} \tag{1.40}$$

can be calculated either with a set of training snapshots or with an a-posteriori estimate. With this estimate, $N = Ntrain$, and the term (1.40) has not be calculated for each parameter. The number of modes $N$ is determined by the tolerance threshold *tol* or can be a priori given. The following codes summarize the offline 3 and online 4 parts of the NIRB method with the Greedy procedure and an orthogonalization in $L^2$ and $H^1$.

```
Data: $(\nu_1, \ldots, \nu_{Ntrain}) \in \mathcal{G}$
Result: $\Phi_1, \ldots, \Phi_N$
initialization: Construct the snapshots $(u_1(\mathbf{x}), \ldots, u_{Ntrain}(\mathbf{x}))$; Use the greedy algorithm 2 and
    set $X_h^N = Span\{\Psi_1(\mathbf{x}), \ldots, \Psi_N(\mathbf{x})\}$
Solve equation (1.38) in order to obtain $(\Phi_1, \ldots, \Phi_N)$.
Optional: Add the rectification posttreatment 0.2.2.
```

**Algorithm 3:** NIRB Offline algorithm

The next section presents numerical results with the NIRB and other RB methods.

**Remark 14.** *In our tests, we observe that the classical NIRB approximation is enhanced when using the fluctuations as with the Galerkin POD in remark 13. This approximation is thus based on a new greedy algorithm similar to algorithm 2 although only feasible when the a training set is already computed (see equation 1.40). The NIRB approximation with fluctuations is given by*

$$\boldsymbol{u}_{Hh}^N(\boldsymbol{x};\nu) = \overline{\boldsymbol{u}}_h(\boldsymbol{x}) + \sum_{i=1}^{N} \alpha_i^H(\nu) \Phi_i^h(\boldsymbol{x}). \tag{1.41}$$

```
Data: $\{\Phi_1, \dots, \Phi_N\}, \nu$ (a new parameter in $\mathcal{G}$)
Result: $u_h^N(\nu)$
initialization: Solve $u_H(\nu)$ on the coarse mesh
$u_h^N(\nu) = \sum_{i=1}^N (u_H(\nu), \Phi_i)\ \Phi_i$
or with rectification $Ru_h^N(\nu) = \sum_{i,j=1}^N R_{ij}(u_H(\nu), \Phi_j)\ \Phi_i$.
```

**Algorithm 4:** NIRB Online algorithm

## 1.5 Numerical results

We solved the problem (1.5) for $Ntrain = 25$ snapshots with $\nu_i = \frac{1}{Re_i}$ with $Re_i = 1 + 25i,\ i = 0, \dots, 24$ and we tested the RB algorithms on the new parameter $nu = \frac{1}{110}$. The next figure illustrates the $H_0^1$ and $L^2$ relative errors as a function of the number of modes $N$ for several RB methods. The errors are given by

$$\frac{\left\| u_h(\nu) - u^N(\nu) \right\|_{H^1(\Omega)}}{\left\| u_h(\nu) \right\|_{H^1(\Omega)}} \text{ and } \frac{\left\| u_h(\nu) - u^N(\nu) \right\|_{L^2(\Omega)}}{\left\| u_h(\nu) \right\|_{L^2(\Omega)}}.$$

Figure 1.2 displays the relative $H_0^1$ and $L^2$ errors for several RB methods:

- the classical NIRB,
- the NIRB with the velocity fluctuations as in (1.41),
- the NIRB with the rectification post-treatment,
- the Galerkin-POD with the velocity fluctuations,
- the PODI.

Time execution (min,sec)

| | RB Offline | RB Online | FEM solver |
|---|---|---|---|
| NIRB | 01:05 | 00:02 | 00:02 |
| NIRB + Rectification | 01:31 | 00:02 | 00:02 |
| Galerkin POD | 01:04 | 00:02 | 00:02 |
| PODI | 00:55 | 00:02 | 00:02 |

As it was expected, the Galerkin POD yields to optimal results. The NIRB with the rectification post-treatment tends to mimic these results. The NIRB without the rectification is more efficient with the velocity fluctuations than with the global variables. On the contrary, the NIRB with the rectification post-treatment is more accurate here with the global variables than with the fluctuations. We will explain in the following chapter why the NIRB error stops decreasing after a certain threshold. The PODI gives good but less stable results. With this example, we cannot observe a reduction of runtime since there are too few degrees of freedom in FEM. The results on the wind turbines in the last chapter will exhibit the computational time reduction. To summarize, with the two-grid method, we gain in non-intrusivity in comparison with the Galerkin POD but the speedup is lower. Indeed, with the Galerkin POD, the error only depends on the Kolmogorov $N$-width whereas with the two-grid method, a term depending on the coarse size mesh appears, as we will see in the subsequent chapter.

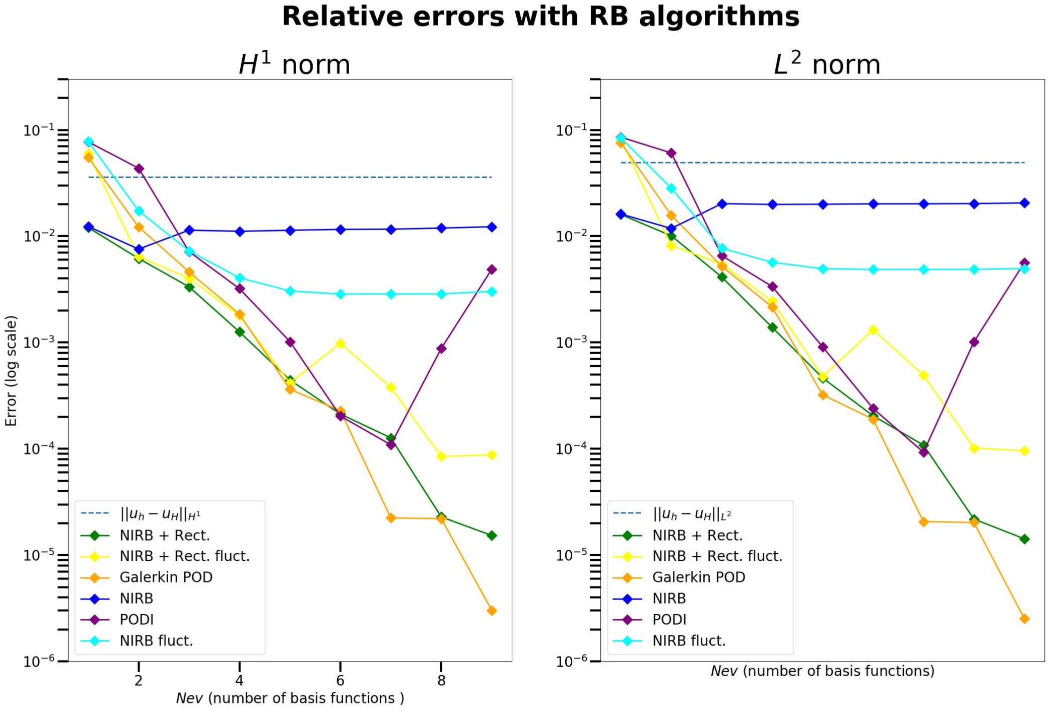


Figure 1.2: RB errors as a function of the number of modes

**Remark 15.** *For the two-grid method implementation, the coarse solution needs to be linearly interpolated into the fine mesh to compute the* $L^2$ *scalar product between this latter and the basis functions. This can be done with the "interpolate" module in Python or with "Basictools". The main idea here is to avoid losing any information about the basis. This method only requires the mesh nodes coordinates. When dealing with uniform meshes, the mass matrix required for the* $L^2$ *scalar product may be replaced by the Identity matrix.*

# Chapter 2

# A Non-Intrusive Reduced Basis method: the two-grid method

This chapter is devoted to the numerical analysis of the two grid method. As a model problem, we will use a classical elliptic equation, namely the Poisson's equation (16a)-(16b).

In this chapter, $\mu \in \mathcal{G}$ represents our varying parameter.

First, we consider a FEM solver and we will recall its FEM analysis. Usually the RB is $L^2$-orthonormalized with a Gram-Schmidt procedure. We will demonstrate that the constants $C_1$ and $C_2$ of the inequality (25) change as a function of the RB, whether or not it is orthogonalized only in $L^2$ or also orthogonalized in $H^1$. We will numerically illustrate this result.
Then, we will propose two further analyses about the NIRB method with FEM solvers. The first one is concerned with its analysis in the case where different meshes are employed. The last one is about its analysis with parabolic problems. The subsequent study is about domain singularities and it is the object of an upcoming article. Its purpose is to use the two grids method with different mesh properties to counterbalance the effects of the singularities. The last section has been published in an article [66] and it is concerned with the NIRB analysis in the context of FV schemes. All these analyses highlight the great number of application possibilities of the two-grid method.

## 2.1 NIRB two-grid error estimate with FEM solver

In this section, we develop a precise analysis of the two-grid method with a FEM solver. We consider as a model problem the Poisson's equation (16a)-(16b), presented in the introduction. The bilinear form $a(u, v; \mu)$ is symmetric, continuous and coercive, such that Lax-Milgram' theorem ensures the well-posedness of the solution. The two grids method is recalled in the last chapter, section 1.4.

Here, we use Delaunay triangulation for the meshes, and thus we have a fine triangular mesh $\mathcal{T}_h$ and a coarse triangular mesh $\mathcal{T}_H$, where $h$ and $H$ are respectively the size of the meshes (see (7)). We use a FEM solver to obtain the fine snapshots and the coarse solutions on these meshes.

The NIRB approximation is denoted $u_{Hh}^N$. We recall that the NIRB approximation is the projection of the coarse solution onto the basis space denoted $X_h^N$ and that it writes

$$u_{Hh}^N(\mu) = \sum_{i=1}^{N} \alpha_i^H(\mu)\, \Phi_i^h = \sum_{i=1}^{N} (u_H(\mu), \Phi_i^h)\, \Phi_i^h, \tag{2.1}$$

where $(\Phi_i^h)_{i=1,\cdots,N}$ is the RB, generated for instance with the Snapshot POD or with a greedy algorithm (see algorithms 1 and 2), and $u_H(\mu)$ is the coarse FEM approximation with a new parameter $\mu$ in $\mathcal{G}$. The RB functions must be orthonormalized in $L^2$ and a further step which implies an $H^1$ orthogonality may be added (see equation (1.38)).

In [96], the NIRB two-grid method has been analyzed with FEM and the following estimate is obtained

$$\left\| u(\mu) - u_{Hh}^N(\mu) \right\|_{H^1} \leq \varepsilon + C_1 h + C_2(N) H^2, \tag{2.2}$$

where $C_1$ and $C_2$ are constants independent of $h$ and $H$ (the mesh sizes), the constant $C_2$ depends on $N$, the usual norm $\|\cdot\|_{H^1} = \|\cdot\|_{H^1(\Omega)}$ is the norm associated to the variational problem, and $\varepsilon$ depends on the Kolmogorov N-width. In this section, we develop a further analysis on the constant $C_2$. We recall that $H^1$ and $H_0^1$ norms are equivalent thanks to Friedrichs' lemma [19]. In this FEM context, our main contribution is the following theorem:

**Theorem 16.** *We consider the model problem* (16a)-(16b) *on a regular spatial domain $\Omega$ (typically $\mathcal{C}^1$). Let $f \in L^2(\Omega)$ be the right-hand side function. Let $\mu$ be a parameter in $\mathcal{G}$ and $u_{Hh}^N(\mu)$ be the NIRB approximation for this parameter defined by* (2.1). *Suppose that the RB $(\Phi_i)_{i=1,\ldots,N}$ is $L^2$ orthonormalized.*

- *The following estimate holds*

$$\left\| u(\mu) - u_{Hh}^N(\mu) \right\|_{H^1} \leq \varepsilon + C_1 h + C_2 \sqrt{\lambda_N} \sqrt{N} H^2, \tag{2.3}$$

  *where $C_1$ and $C_2$ are constants independent of $h$, $H$, and $N$, and $\varepsilon$ depends on the Kolmogorov N-width.*

- *Moreover, if the RB is also orthogonalized in $H^1$, then with the same notations*

$$\left\| u(\mu) - u_{Hh}^N(\mu) \right\|_{H^1} \leq \varepsilon + C_1 h + C_2 \sqrt{\lambda_N} H^2. \tag{2.4}$$

With the $L^2$ norm, we obtain the following theorem.

**Theorem 17.** *With the same assumptions than theorem 16, with the $L^2$ orthonormalized RB, the following estimate holds*

$$\left\| u(\mu) - u_{Hh}^N(\mu) \right\|_{H^1} \leq \varepsilon' + C_1' H^2, \tag{2.5}$$

*where $C_1'$ is a constant independent of $h$, $H$ and $N$, and $\varepsilon'$ depends on the Kolmogorov N-width.*

These estimates show that with elliptic equations, the NIRB method (with a FEM solver) gives the same order estimate in the energy-norm as with classical HF FEM, provided that $H^2 \simeq h$. However, with the $L^2$ norm, we cannot reach the super convergence of HF FEM code. Moreover, with only the $L^2$-orthogonalization of the RB, the constant $C_2$ has a larger dependence on $N$ which may induce an increase of the error after a threshold. We demonstrate that the RB orthogonalization in $H^1$ counterbalances this drawback. From a practical point of view, orthogonalizing in $H^1$ and $L^2$ will stabilize the error. Finally, in the $L^2$ estimate, the constants do not depend on $N$ anymore.

We go on with the proof of theorem (16). To lessen the notations, we will denote by $C$ the constants not dependent of $H$ or $h$. We consider $\mathbb{P}_1$ FE space. We emphasize that this estimate implies an $h$ convergence for the NIRB approximation if $\mathcal{T}_H$ has been chosen such that $H^2 \simeq h$.

*Proof.* The fine and coarse coefficients, recalled in the introduction (9), are denoted $\alpha_i^h(\mu)$ and $\alpha_i^H(\mu)$. With the triangle inequality on $\left\|u(\mu) - u_{Hh}^N(\mu)\right\|_{H^1}$

$$\begin{aligned}\left\|u(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} &\leq \left\|u(\mu) - u_h(\mu)\right\|_{H^1} + \left\|u_h(\mu) - u_{hh}^N(\mu)\right\|_{H^1} + \left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} \\ &=: T_1 + T_2 + T_3, \end{aligned} \tag{2.6}$$

where

$$u_{hh}^N(\mu) = \sum_{i=1}^N \alpha_i^h(\mu)\ \Phi_i^h = \sum_{i=1}^N (u_h(\mu), \Phi_i^h)\ \Phi_i^h. \tag{2.7}$$

- The first term $T_1$ can be estimated using a classical FEM result, following from Céa's lemma 1. Indeed, since $\Omega$ is $\mathcal{C}^1$ and since $f \in L^2(\Omega)$, then the elliptic regularity estimate [19] gives that $u \in H^2(\Omega)$ and

  $$\|u\|_{H^2(\Omega)} \leq C\ \|f\|_{L^2(\Omega)}, \tag{2.8}$$

  where C is a constant independent of $h$. We denote $I_h$ the nodal interpolation operator from $C(\overline{\Omega})$ into the $C^0$ Lagrange finite element $V_h$ space associated to the mesh $\mathcal{T}_h$. From the FEM interpolation theory (inequality (24)), we have

  $$\left\|u(\mu) - u_h(\mu)\right\|_{H^1} \leq Ch\ \|u\|_{H^2(\Omega)} \leq Ch\|f\|_{L^2(\Omega)}. \tag{2.9}$$

- The best achievable error in the uniform sense regarding the parameters of a fine solution projected into $X_h^N$ relies on the notion of Kolmogorov n-width ([85], Theorem 20.1 [111]). The manifold $\mathcal{S}_h$ is a compact set in a Banach space $V_h$, and we refer to the Kolmogorov n-width of $\mathcal{S}_h$ given in definition 1. Here we suppose that the set of all the solutions $\mathcal{S}_h = \{u_h(\mu), \mu \in \mathcal{G}\}$ has a low complexity. It means that for an accuracy $\varepsilon = \varepsilon(N)$ related to the Kolmogorov n-width of the manifold $\mathcal{S}_h$, for any $\mu \in \mathcal{G}$ [34, 96, 28, 22]

  $$T_2 = \left\|u_h(\mu) - \sum_{i=1}^N \alpha_i^h(\mu)\ \Phi_i^h\right\|_{H^1} = \left\|u_h(\mu) - \sum_{i=1}^N (u_h(\mu), \Phi_i^h)\ \Phi_i^h\right\|_{H^1} \leq \varepsilon(N). \tag{2.10}$$

- The third term $T_3$ depends on the method used to create the RB. The following developments correspond to our contribution on the NIRB analysis in the FEM context. Let us recall that $u_{hh}^N$ is defined by equation (2.7).

  – Suppose we use a Greedy algorithm 2 with a Gram-Schmidt procedure in order to orthonormalize the RB in the $L_2$ norm, and in addition, we solve the eigenvalue problem (1.38) to orthogonalize it in $H^1$. By $H^1$ orthogonality,

$$\begin{aligned}\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1}^2 &= \sum_{i=1}^{N} |\alpha_i^h(\mu) - \alpha_i^H(\mu)|^2 \left\|\Phi_i^h\right\|_{H^1}^2, \\ &= \sum_{i=1}^{N} |(u_h(\mu) - u_H(\mu), \Phi_i^h)|^2 \left\|\Phi_i^h\right\|_{H^1}^2. \end{aligned} \tag{2.11}$$

From the RB orthonormalization in $L_2$, the equation (1.38) yields

$$\left\|\Phi_i^h\right\|_{H^1}^2 := \left\|\nabla\Phi_i^h\right\|_{L^2(\Omega)}^2 = \lambda_i \left\|\Phi_i^h\right\|_{L^2(\Omega)}^2 = \lambda_i \leq \max_{i=1,\cdots,N} \lambda_i = \lambda_N, \tag{2.12}$$

such that, from (2.12), the inequality (2.11) becomes

$$\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1}^2 \leq \lambda_N \sum_{i=1}^{N} |(u_h(\mu) - u_H(\mu), \Phi_i^h)|^2 \leq C\lambda_N \left\|u_h(\mu) - u_H(\mu)\right\|_{L^2(\Omega)}^2, \tag{2.13}$$

By Aubin-Nitsche's lemma 2 and the inequality (2.8), we have:

$$\begin{aligned} \left\|u(\mu) - u_H(\mu)\right\|_{L^2} &\leq C\, H^2 \|f\|_{L^2(\Omega)}, \\ \left\|u(\mu) - u_h(\mu)\right\|_{L^2} &\leq C\, h^2 \|f\|_{L^2(\Omega)}, \end{aligned}$$

such that, combining these two inequalities, we obtain

$$\left\|u_h(\mu) - u_H(\mu)\right\|_{L^2(\Omega)} \leq C(h^2 + H^2), \tag{2.14}$$

and from (2.13) (majorizing the $h^2$ term), we conclude that

$$\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1}^2 \leq C\sqrt{\lambda_N} H^2, \tag{2.15}$$

where $C$ does not depend on $N$, which yields inequality (2.4), using the equations (2.6) with (2.9), (2.10) and (2.15).

- Consider now an $L^2$-orthonormalized RB $(\Psi_i^h)_{i=1,\dots,N}$ (with a Gram-Schmidt algorithm or with the Snapshots POD). The functions $(\Psi_i^h)_{i=1,\dots,N}$ and $(\Phi_i^h)_{i=1,\dots,N}$ are both generators of $X_h^N$. Thus, there exists $(\gamma^i)_{i=1,\dots,N} \in \mathbb{R}^N$ such that $\Psi_i^h = \sum_{j=1}^{N} \gamma_j^i \Phi_j^h$. By the $H^1$-orthogonality of the $(\Phi_j^h)_{j=1,\dots,N}$, it follows

$$\begin{aligned} \left\|\Psi_i^h\right\|_{H^1}^2 &= \sum_{j=1}^{N} |\gamma_j^i|^2 \left\|\Phi_j^h\right\|_{H^1}^2, \\ &\leq \lambda_N \sum_{j=1}^{N} |\gamma_j^i|^2 \left\|\Phi_j^h\right\|_{L^2(\Omega)}^2 \quad \text{by equation (1.38)}, \\ &= \lambda_N \left\|\Psi_i^h\right\|_{L^2(\Omega)}^2 \quad \text{by the } L^2\text{-orthogonality of the } (\Psi_i^h)_{i=1,\dots,N}. \end{aligned} \tag{2.16}$$

From the estimate (2.16) and the $L_2$-orthonormalization of the RB,

$$\begin{aligned}
\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} &= \left\|\sum_{i=1}^N \alpha_i^h(\mu)\ \Psi_i^h - \sum_{i=1}^N \alpha_i^H(\mu)\ \Psi_i^h\right\|_{H^1}, \\
&\leq \sum_{i=1}^N |\alpha_i^h(\mu) - \alpha_i^H(\mu)| \left\|\Psi_i^h\right\|_{H^1}, \\
&= \sum_{i=1}^N |(u_h(\mu) - u_H(\mu), \Psi_i^h)| \left\|\Psi_i^h\right\|_{H^1}, \\
&\leq C\sqrt{\lambda_N} \sum_{i=1}^N |(u_h(\mu) - u_H(\mu), \Psi_i^h)|. && (2.17)
\end{aligned}$$

From Cauchy-Schwarz inequality, inequality (2.17) leads to

$$\begin{aligned}
\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} &\leq C\sqrt{\lambda_N}\sqrt{N} \sqrt{\sum_{i=1}^N |(u_h(\mu) - u_H(\mu), \Psi_i^h)|^2}, \\
&\leq C\sqrt{\lambda_N}\sqrt{N} \|u_h(\mu) - u_H(\mu)\|_{L^2(\Omega)}. && (2.18)
\end{aligned}$$

From the estimate 2.18, we end up with

$$\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} \leq C\sqrt{N}\sqrt{\lambda_N} H^2, \tag{2.19}$$

which leads to estimate (2.3) using the equations (2.6) with (2.9), (2.10) and (2.19).

□

Therefore, with an $L^2$ and $H^1$ orthogonalized RB, the constant $C_2$ in theorem 16 has a smaller dependence regarding the number of modes $N$. Thus, the NIRB approximation is stabilized with the $H^1$ orthogonality, compared with a RB only orthogonalized in $L^2$.

**Remark 18.** *These results are highlighted by Figure 2.2 where both methods are compared.*

$L^2$ **estimate.** We proceed with the proof of theorem 17.

*Proof.* In analogy with the $H^1$ estimate, we have

$$\begin{aligned}
\left\|u(\mu) - u_{Hh}^N(\mu)\right\|_{L^2} &\leq \|u(\mu) - u_h(\mu)\|_{L^2} + \left\|u_h(\mu) - u_{hh}^N(\mu)\right\|_{L^2} + \left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{L^2} \\
&=: T_1 + T_2 + T_3. && (2.20)
\end{aligned}$$

- For the first term $T_1$, it follows from Aubin Nitsche's lemma 2 that

$$T_1 \leq Ch^2 \|u\|_H^2. \tag{2.21}$$

- As with the $H^1$ estimate, $T_2$ can be estimated with the Kolmogorov N-width, and thus, for an accuracy $\varepsilon' = \varepsilon'(N) \leq \varepsilon(N)$,

$$T_2 = \left\|u_h(\mu) - u_{hh}^N(\mu)\right\|_{L^2} \leq \varepsilon'. \tag{2.22}$$

- For the last term $T_3$, by $L^2$-orthonormality,

$$\begin{aligned}\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{L^2(\Omega)}^2 &= \sum_{i=1}^{N} |(u_h(\mu) - u_H(\mu), \Psi_i^h)|^2 \left\|\Psi_i^h\right\|_{L^2(\Omega)}^2, \\ &\leq C\|u_h(\mu) - u_H(\mu)\|_{L^2(\Omega)^2}. \end{aligned} \tag{2.23}$$

And equation (2.14) yields

$$\left\|u_{hh}^N(\mu) - u_{Hh}^N(\mu)\right\|_{L^2(\Omega)}^2 \leq C\, H^2, \tag{2.24}$$

where $C$ does not depend on $N$ neither on $H$. Combining (2.6) with (2.24), and majorizing the $h^2$ term, we end up with

$$\begin{aligned}\left\|u(\mu) - u_{Hh}^N(\mu)\right\|_{H^1} &\leq \varepsilon'(N) + C\, H^2 \\ &= \varepsilon'(N) + C\, h\, if\; H^2 \simeq h. \end{aligned} \tag{2.25}$$

□

### 2.1.1 NIRB results with FEM on the model problem

We compare now the NIRB approximations on the Poisson's problem (16a)-(16b) with different RB orthogonalizations. In our example, we consider the diffusion coefficient as the function

$$A(\mu) = \begin{bmatrix} a(\mu, \mathbf{x}) & 0 \\ 0 & a(\mu, \mathbf{x}) \end{bmatrix} \text{ where } \mu = (\mu_1, \mu_2) \text{ and } a(\mu, \mathbf{x}) = 2\mu_1 + \mu_2 sin(x + y)cos(xy),$$

$(\mu_1, \mu_2) \in (0, 1)^2$. The function f is equal to $(1 - y)y + (1 - x)x$.

The solution parameters are $\mu_1 = 0.1$ and $\mu_2 = 0.05$, and we choose $Ntrain = 30$ random parameters for the snapshots. The reference mesh is of size $h_{ref} = \frac{\sqrt{2}}{100}$ ($100 \times 100$ grid in the unit square $[0, 1]^2$), the fine mesh size is $h = \frac{2\sqrt{2}}{100}$ ($50 \times 50$), and we tested the NIRB algorithm with $H = \frac{\sqrt{2}}{10}$ ($10 \times 10$).

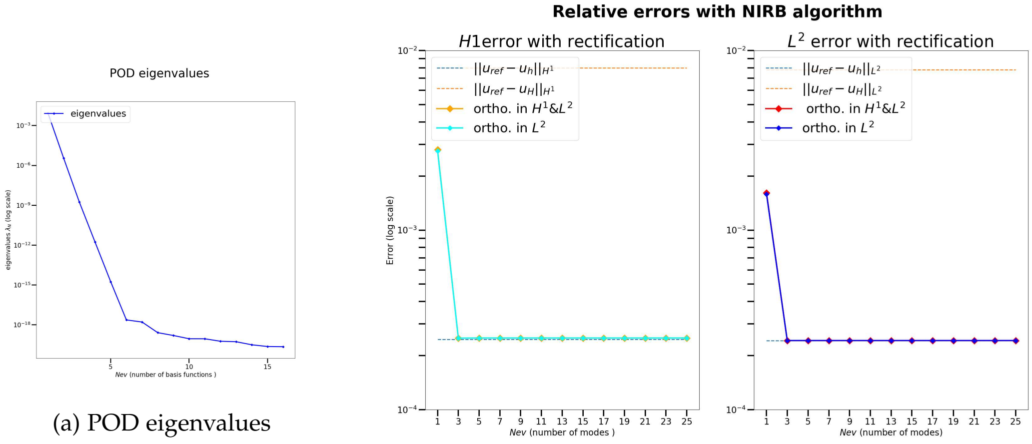
We present the NIRB relative errors in $H^1$ and $L^2$ norm with a basis orthogonalized in $H^1$ and $L^2$ or only in $L^2$. We used the greedy algorithm 2 with and without the rectification post-treatmento.2.2. The POD eigenvalues are decreasing fast enough, as we can see in Figure 2.1a. Here, the RIC (1.19) gives with one mode I(1)=0.9996 and with two I(2)=0.9999998. We can see in Figure 2.1b that two modes are sufficient to retrieve the FEM errors with the rectification post-process.

With the rectification, the NIRB solution corresponds to equation (14), and the relative errors are

$$\frac{\left\|u_{ref}(\mu) - Ru_{Hh}^N(\mu)\right\|_{H^1(\Omega)}}{\left\|u_{ref}\right\|_{H^1(\Omega)}} \text{ and } \frac{\left\|u_{ref}(\mu) - Ru_{Hh}^N(\mu)\right\|_{L^2(\Omega)}}{\left\|u_{ref}\right\|_{L^2(\Omega)}}.$$

In Figure 2.2, we compare the errors without rectification given by

$$\frac{\left\|u_h(\mu) - u_{Hh}^N(\mu)\right\|_{H^1(\Omega)}}{\|u_h\|_{H^1(\Omega)}} \text{ and } \frac{\left\|u_h(\mu) - u_{Hh}^N(\mu)\right\|_{L^2(\Omega)}}{\|u_h\|_{L^2(\Omega)}}.$$



(a) POD eigenvalues

(b) $H^1$ (left) and $L^2$ (right) relative errors with NIRB and rectification

Figure 2.1: NIRB results

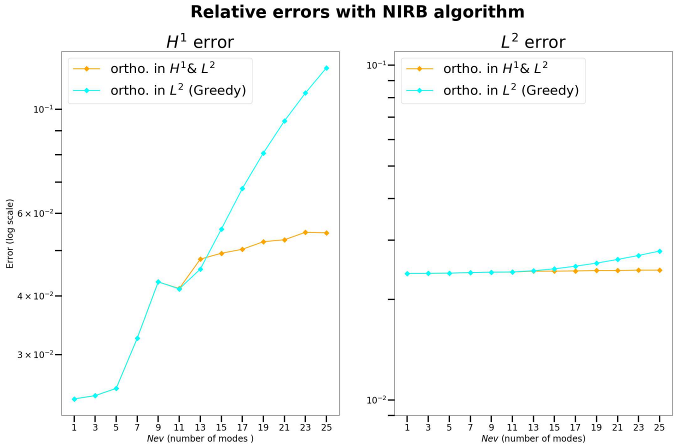


Figure 2.2: $H^1$ (left) and $L^2$ (right) relative errors with NIRB

We can see in Figure 2.2 that the errors with the basis orthogonalized in $L^2$ and in $H^1$ remain more stable than with the $L^2$ orthogonalized basis.

**Remark 19.** *Here we presented results with $H^2 \simeq h$. In practice, the size of the coarse mesh is not necessarily chosen such that $H^2 = h$ since there is one constant upstream this term in* (2.25)*. We will present later an ongoing project on a posteriori NIRB error in order to choose the correct size of the meshes B.*

Time execution (min,sec)

| NIRB Offline | NIRB Online | FEM solver |
|---|---|---|
| 01:50 | 00:01 | 00:03 |

**Remark 20.** *Here, the online time is smaller than the FEM HF solver.*

## 2.2 Some complements on the analysis

This section is concerned with two further analyses of the two-grid method in the FEM context. The first study gives an analysis in the case when the meshes used by the solver and by the black-box code are different. The second one is a NIRB analysis on parabolic equations. Let us first recall the definition of fractional semi-norms.

**Definition 4.** *Let $k$ be a nonnegative integer and $0 < \lambda < 1$ ($d$ is the dimension). The seminorm $|\cdot|_{H^{k+\lambda}(\Omega)}$ is defined by [2]*

$$|v|_{H^{k+\lambda}(\Omega)} = \sum_{|\alpha|=k} \int_\Omega \int_\Omega \frac{[\partial^\alpha v(\boldsymbol{x}) - \partial^\alpha v(\mathbf{y})]^2}{|\boldsymbol{x} - \mathbf{y}|^{d+2\lambda}} \, dxd\mathbf{y}, \tag{2.26}$$

*where $v \in H^{k+\lambda}(\Omega)$.*

Then the norm is defined by

$$\|v\|^2_{H^{k+\lambda}(\Omega)} = \|v\|^2_{H^k(\Omega)} + |v|^2_{H^{k+\lambda}(\Omega)}. \tag{2.27}$$

### 2.2.1 Estimate on two different fine meshes

The previous analysis on an elliptic problem gave us an estimate in $\mathcal{O}(h)$ for the NIRB approximation. Usually the same mesh is used for the black-box solver and the NIRB algorithm, for the fine and the coarse meshes. But this may not be the case, and we may have only access to the mesh coordinates used in the black-box solver. In fact, it is feasible to recover this optimal estimate while only knowing the nodes coordinates, and not the mesh connectivity. This can be useful, for instance, if the black-box solver mesh uses cells non recognized by the non-intrusive code. Thus, a Delaunay triangulation can be generated with the nodes coordinates, and this will leads to the same estimate.

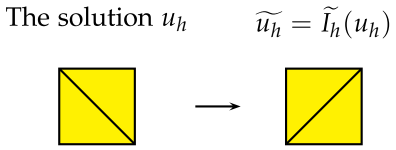


Figure 2.3: Two meshes with different connectivities

Let $\mathcal{M}_h$ be the fine mesh, defined on $\Omega$. Let $u_h$ be the $\mathbb{P}_1$ FE approximation, i.e. the solution on the finite space $V_h$ of (17)

$$a(u_h, v_h) = (f, v_h). \tag{2.28}$$

Then, the following estimate follows directly from Céa's lemma 1 and Aubin-Nitsche's lemma 2

$$\|u - u_h\|_{L^2} + h\|u - u_h\|_{H^1} \leq Ch^2\|u\|_{H^2}. \tag{2.29}$$

From the nodes of $\mathcal{M}_h$, we define another mesh $\widetilde{M}_h$, with another cell connectivity, and we create the interpolate $\widetilde{u_h}$ of the FEM solution on this new mesh. In other words $u_h$ is defined as

$$\widetilde{u_h} = \widetilde{I}_h u_h, \tag{2.30}$$

where $\widetilde{I}_h$ is the Lagrangian interpolated operator on $\widetilde{\mathcal{M}_h}$. We go on with the proof of the theorem 7 presented in the introduction.

*Proof.* With the triangle inequality,

$$\begin{aligned} \|u - \widetilde{u_h}\|_{H^1} &= \left\|u - \widetilde{I}_h u_h\right\|_{H^1}, \\ &\leq \left\|u - \widetilde{I}_h u\right\|_{H^1} + \left\|\widetilde{I}_h u - \widetilde{I}_h u_h\right\|_{H^1}. \end{aligned} \tag{2.31}$$

With theorem 3 from the introduction (theorem 4.4.4 [19]), the inequality (2.31) implies

$$\|u - \widetilde{u_h}\|_{H^1} \leq Ch\|u\|_{H^2} + \underbrace{\left\|\widetilde{I}_h(u - u_h)\right\|_{H^1}}_{t_1}. \tag{2.32}$$

With the inequality (2.29) the second term of the right term of (2.32) denoted $t_1$ yields

$$\begin{aligned} t_1 = \left\|\widetilde{I}_h(u - u_h)\right\|_{H^1} &\leq \|u - u_h\|_{H^1} + \left\|(u - u_h) - \widetilde{I}_h(u - u_h)\right\|_{H^1}, \\ &\leq Ch\|u\|_{H^2} + \left\|(u - u_h) - \widetilde{I}_h(u - u_h)\right\|_{H^1}. \end{aligned} \tag{2.33}$$

The second term of the right-hand side of (2.33) can be estimated with the finite elements theory of fractional order Sobolev norm (see theorem 4.4.20 [19]).We obtain from (2.33)

$$t_1 = \left\|\widetilde{I}_h(u - u_h)\right\|_{H^1(\Omega)} \leq Ch\|u\|_{H^2(\Omega)} + Ch^{\varepsilon}\underbrace{\|u - u_h\|_{H^{1+\varepsilon}(\Omega)}}_{t_2}, \tag{2.34}$$

with $\varepsilon < \frac{1}{2}$. With theorem 5 presented in the introduction, with $\mu = 1$ and $\lambda = \varepsilon$,

$$\|u - I_h u\|_{H^{1+\varepsilon}(\Omega)} \leq Ch^{1-\varepsilon}\|u\|_{H^2(\Omega)}.$$

Thus, the second term of the right-hand side of the inequality (2.34), denoted $t_2$, becomes

$$\begin{aligned} t_2 = \|u - u_h\|_{H^{1+\varepsilon}(\Omega)} &\leq \|u - I_h u\|_{H^{1+\varepsilon}(\Omega)} + \|I_h u - u_h\|_{H^{1+\varepsilon}(\Omega)}, \\ &\leq Ch^{1-\varepsilon}\|u\|_{H^2(\Omega)} + \underbrace{\|I_h u - u_h\|_{H^{1+\varepsilon}(\Omega)}}_{t_3}. \end{aligned} \tag{2.35}$$

Now, since $I_h u - u_h \in V_h$, with theorem 6 (presented in the introduction) on the inverse estimate [13] (with $\lambda = \varepsilon$ and $\theta = 0$), we get

$$t_3 = \|I_h u - u_h\|_{H^{1+\varepsilon}(\Omega)} \leq Ch^{-\varepsilon}\|I_h u - u_h\|_{H^1(\Omega)}. \tag{2.36}$$

and using the inequality (2.29) and theorem 3 from the introduction and the triangle inequality,

$$\begin{aligned} \|I_h u - u_h\|_{H^1(\Omega)} &\leq \|I_h u - u\|_{H^1(\Omega)} + \|u - u_h\|_{H^1(\Omega)}, \\ &\leq Ch\|u\|_{H^2(\Omega)}, \end{aligned} \tag{2.37}$$

such that

$$t_3 = \|I_h u - u_h\|_{H^{1+\varepsilon}(\Omega)} \leq Ch^{1-\varepsilon}\|u\|_{H^2(\Omega)}. \tag{2.38}$$

Therefore, with (2.38), the inequality (2.35) boils down to

$$t_2 \leq Ch^{1-\varepsilon}\|u\|_{H^2(\Omega)}, \tag{2.39}$$

and with the inequality (2.34), we end up with

$$t_1 \leq Ch\|u\|_{H^2(\Omega)}. \tag{2.40}$$

Finally, from the inequality (2.32),

$$\|u - \widetilde{u_h}\|_{H^1(\Omega)} \leq Ch\|u\|_{H^2(\Omega)}, \tag{2.41}$$

which concludes the proof. □

Now, to recover the NIRB estimate, we refer to the following analysis.

- From this estimate (2.41), in the term $T_1$ in the FEM analysis, inequality (2.9) still holds.
- The term $T_2$ remains unchanged.
- The super-convergence in $L^2$, required for the coarse solution in inequality (2.14), follows from the same developments since, with Aubin-Nitsche's lemma, the equation (2.34) becomes

$$\begin{aligned} t_1 = \left\|\widetilde{I_h}(u - u_h)\right\|_{L^2(\Omega)} &\leq \|u - u_h\|_{L^2(\Omega)} + \left\|(u - u_h) - \widetilde{I_h}(u - u_h)\right\|_{L^2(\Omega)}, \\ &\leq Ch^2\|u\|_{H^2(\Omega)} + Ch^{1+\varepsilon}\|u - u_h\|_{H^{1+\varepsilon}(\Omega)}. \end{aligned} \tag{2.42}$$

and thus, the equation (2.32) becomes

$$\|u - \widetilde{u_h}\|_{L^2(\Omega)} \leq Ch^2\|u\|_{H^2(\Omega)} + Ch^{1+\varepsilon}\underbrace{\|u - u_h\|_{H^{1+\varepsilon}(\Omega)}}_{t_2}. \tag{2.43}$$

The rest of the proof of the NIRB estimate remains similar to the previous analysis (see theorems 16 and 17).

All the previous analyses were concerned with elliptic equations. Now, the subsequent section is about extending NIRB to time-dependent problems, in the setting of parabolic equations.

### 2.2.2 Parabolic equations

This part is the object of an upcoming article. The aim of this section is to generalyze to parabolic equations and to analyze the NIRB method in the context of time-dependent problems. The NIRB method has not yet been tested with such problems.
For this purpose, we will consider a parabolic equation on a domain $\Omega$ in $\mathbb{R}^d$ ($d \leq 3$), with smooth boundary $\partial\Omega$ and with homogeneous Dirichlet conditions, which takes the form

$$\begin{aligned} u_t - \mu\Delta u &= f, \ \text{ in } \Omega\times]0,T[, \\ u(\mathbf{x},0) &= u_0(\mathbf{x}), \ \text{ in } \Omega, \\ u(\mathbf{x},t) &= 0, \ \text{ on } \partial\Omega, \end{aligned} \tag{2.44}$$

where $u_t$ denotes $\frac{\partial u}{\partial t}$ and $\mu \in \mathcal{G}$ is the parameter. In analogy with the previous work on the NIRB FEM estimate applied to elliptic equations, we consider one fine spatial grid for the snapshots and one rough grid for the coarse solution, such that the sizes are respectively $h$ for the fine mesh and $H$ for the coarse mesh (with $h << H$) (7). Usually, we proceed in two steps to approximate the problem (2.44) with a HF code. As in the elliptic case just considered, the spatially discrete problem is based on a weak formulation of (2.44). For convenience, we will consider a $\mathbb{P}_1$ FEM space for the spatial discretization, but the following analysis can be generalized to $\mathbb{P}_k$ FEM space, with $k > 1$. We then discretize this system in the time variable to get a fully discrete approximation of the solution of (2.44) by a time stepping method. For this discretization, we will consider finite difference approximation of the time derivative. Thus we deal with three kind of notations:

- $u(t, \mathbf{x})$ denotes the true solution at time $t \geq 0$ for $\mathbf{x} \in \Omega$.
- $u_h(t)$ is the fine solution of the spatially semidiscrete solution, at time $t \geq 0$.
- $u_h^n$ is the full-discretized solution at time $t^n = n \times \Delta t$ where $\Delta t$ is the time step.

To obtain the NIRB estimate (31), we will use two time grids, one rough grid for the coarse solution and one thiner for the fine solution. The NIRB method allows us to recover the optimal estimate in space, as in the previous analysis with elliptic equations. To recover the optimal estimate in time, we consider a higher order time scheme for the coarse solution.

**Remark 21.** *To simplify the notations, we consider that both time discretizations end at time T.*

Let us first introduce the Sobolev space for the time dependent functions

$$L^q(0,T;W^{k,p}(\Omega)) := \{u(\mathbf{x},t) \mid \|u\|_{L^q(0,T;W^{k,p}(\Omega))} := \Big(\int_0^T \|u(\cdot,t)\|_{k,p}^q \, dt\Big)^{1/q} < \infty\}.$$

The variational form of (2.44) is given by:
Find $u \in L^2(0,T;H_0^1(\Omega))$ with $u_t \in L^2(0,T;H^{-1}(\Omega))$ such that

$$\begin{cases} (u_t, v) + a(u,v) = (f,v), \ \forall v \in H_0^1 \text{ and } t \in (0,T), \\ u(\cdot,0) = u_0, \end{cases} \tag{2.45}$$

where $a$ is given by (18). We assume that (2.45) is well posed (see [51] for the existence and the uniqueness of (2.45)) and that $f$ is sufficiently regular for the norms on the right-hand side of the inequalities to be finite in what follows). Let $V_h$ consists of piecewise linear finite element functions. The full discrete form of the variational problem (2.45) writes:

$$\begin{cases} (\overline{\partial} u_h^n, v_h) + a(u_h^n, v_h) = (f(t^n), v_h), \ \forall v_h \in V_h \text{ and } n \geq 1, \\ u_h(\cdot, 0) = u_h^0, \end{cases} \tag{2.46}$$

where the time derivative in the variational form of the problem (2.44) has been replaced by a backward difference quotient, $\overline{\partial} u_h^n = \frac{u_h^n - u_h^{n-1}}{\Delta t_F}$.

We will consider the solutions of (2.46), and we seek the NIRB approximation with the form

$$u_{Hh}^{N,n}(\mathbf{x}; \mu) = \sum_{i=1}^{N} \alpha_i^H(\mu, t^n) \ \Phi_i^h(\mathbf{x}), \ n \geq 0, \tag{2.47}$$

where the basis functions $(\Phi_{h,i})_{i=1,\dots,N}$ do not depend on time, only the coefficients are time-dependent. This means that during the offline part, the snapshots are stored as a matrix, whose shape is $(\frac{T}{\Delta t_F} \times Ntrain, \mathcal{N})$, where $\mathcal{N}_h$ is the number of degrees of freedom on the fine mesh and $Ntrain$ the number of training parameters in $\mathcal{G}$.

**Main result** Our main result is concerned with the following theorem.

**Theorem 22.** *NIRB error estimate for parabolic equations. Let us consider the problem 29 and its solution, which belongs to $H^2(\Omega) \cap H_0^1(\Omega)$. We consider two time grids, ending with at time T.*

- *One time grid, denoted F, is employed for the fine solution (and for the snapshots). To avoid making notations more cumbersome, we will consider a uniform time grid with $\Delta t_F$ the interval between two time values. The time levels can be written $t = t_n = n\Delta t_F$, where $n \in \mathbb{N}^*$, and we denote $u_h^n \in V_h$ the approximation of $u_h(t^n)$ to be determined.*

- *Another time grid, denoted G, is used for the coarse solution. By analogy with the fine grid, we consider a uniform grid with time step $\Delta t_G$. This time, the time levels are written $t = t_m = m\Delta_G$, where m is a nonnegative integer, and we denote $u_H^m \in V_h$ the approximation of $u_H(t^m)$ to be determined.*

*Let $(\Phi_i^h)_{i=1,\dots,N}$ be the $L^2$-orthonormalized and $H^1$-orthogonalized RB generated with the greedy algorithm 2 from fine solutions of (2.46), on a fine mesh of size h, with a $\mathbb{P}_1$ FEM for the spatial discretization, and a Backward Euler scheme for the time discretization. We consider $V_h$ as the associated finite-dimensional subspace of $H_0^1$.*

*For $m = 0, \dots, \frac{T}{\Delta t_G}$, let $u_H^m(\mu)$ be a coarse approximation of (2.46) for $\mu \in \mathcal{G}$, on a rough mesh of size H, with a $\mathbb{P}_1$ FEM for the spatial discretization, and a Crank-Nicolson scheme for the time discretization. We consider $V_H$ as the associated finite-dimensional subspace of $H_0^1$.*

*Let us consider the NIRB approximation, defined by (2.66) and derived from these solutions. Then, the following estimate holds*

$$\text{for } n = 0, \dots, \frac{T}{\Delta t_F}, \ \left\| u(t^n)(\mu) - u_{Hh}^{N,n}(\mu) \right\|_{H^1(\Omega)} \leq \varepsilon + C_1 h + C_2(N) H^2 + C_3 \Delta t_F + C_4(N) \Delta t_G^2, \tag{2.48}$$

*where $C_1, C_2, C_3$ and $C_4$ are constants independent of h and H, $\Delta t_F$ and $\Delta t_G$. The term $\varepsilon$ depends on the Kolmogorov N-width. If H is such as $H^2 \sim h$, $\Delta t_G^2 \sim \Delta t_F$, and $\varepsilon(N)$ is small enough, it results in an error estimate in $\mathcal{O}(h) + \Delta t_F$.*

Theorem 22 proves that the NIRB two grids method can be applied to such problems. We recover optimal estimates in $L^\infty(0,T;H^1(\Omega))$. We emphasize that the choice of the finite difference scheme is motivated by the fact that we want the rough solution to be coarser in space as well as in time, compared to the fine solution. Before entering in the proof details of Theorem 22, we require a few theorems from [129], on the FEM classical estimates and on both finite difference schemes employed. Before recalling them, we summarize their main properties that will be used:

- On the fine grid, the Backward Euler scheme exhibits an estimate in $h^2$ and $\Delta t_F$ for the $L^2$ norm.
- This scheme ensures an estimate in $h$ and $\Delta t_F$ in the $H^1$ norm.
- On the coarse grid, the Crank-Nicolson scheme yields an estimate in $H^2$ and $\Delta t_G^2$ in the $L^2$ norm.

It is well known that with a FEM semi-discretization in space, the following estimate holds.

**Theorem 23** (Theorem 1.2 [129]). *Let $u$ and $u_h$ be the solutions of* (2.44) *and the semidiscretized variational form of* (2.45)*, respectively. Assume $u_0 = 0$ on $\partial\Omega$. Then,*

$$\|u(t) - u_h(t)\|_{L^2(\Omega)} \leq Ch^2 \left[ \|u_0\|_{H^2(\Omega)} + \int_0^t \|u_t\|_{H^2(\Omega)}\, ds \right], \quad \text{for } t \geq 0. \tag{2.49}$$

Once fully discretized on a fine mesh with the backward Euler Galerkin method, the estimate (2.49) yields the following estimate.

**Theorem 24** (Theorem 1.5 [129]). *With $u_h^n$ and $u$ be the solutions of* (2.46) *and* (2.44)*, respectively. We have, if $\left\| u_0^h - u_0 \right\| \leq Ch^2 \|u_0\|_{H^2(\Omega)}$ and $u_0 = 0$ on $\partial\Omega$,*

$$\|u(t^n) - u_h^n\|_{L^2(\Omega)} \leq Ch^2 \left[ \|u_0\|_{H^2(\Omega)} + \int_0^{t^n} \|u_t\|_{H^2(\Omega)}\, ds \right] + \Delta t_F \int_0^{t^n} \|u_{tt}\|_{L^2(\Omega)}\, ds, \quad \forall n \geq 0. \tag{2.50}$$

With the energy semi-norm, we have the following estimate.

**Theorem 25** (Theorem 1.4 [129]). *Under the assumptions of Theorem 23, we have*

$$\|\sqrt{\mu}(\nabla u(t) - \nabla u_h(t))\|_{L^2(\Omega)} \leq Ch \left[ \|u_0\|_{H^2(\Omega)} + \|u(t)\|_{H^2(\Omega)} + \Big(\int_0^t \|u_t\|_{H^1(\Omega)}^2\, ds\Big)^{1/2} \right], \quad \text{for } t \geq 0. \tag{2.51}$$

The estimate (2.51) with the full discretization leads to the following theorem.

**Theorem 26.** *Under the assumptions of Theorem 24, we have*

$$\begin{aligned} \|\sqrt{\mu}(\nabla u_h^n - \nabla u(t^n))\|_{L^2(\Omega)} \leq & Ch \left[ \|u_0\|_{H^2(\Omega)} + \int_0^{t^n} \|u_t\|_{H^2(\Omega)}\, ds \right] \\ & + \Delta t_F \Big(\int_0^{t^n} \|\nabla u_{tt}\|_{L^2(\Omega)}^2\, ds\Big)^{1/2} + \mathcal{O}(h^2), \quad \forall n \geq 0. \end{aligned} \tag{2.52}$$

*Proof.* This may be proved with the same tricks as with the proof on the $L^2$ estimate (Theorem 1.5 [129]), as it is highlighted in [129]. Let us detail. We consider the projection operator $P1$ on $V_h$ which is given by

$$(\nabla P1u, \nabla v) = (\nabla u, \nabla v), \quad \forall v \in V_h. \tag{2.53}$$

We first decompose the error with two components $\rho$ and $\theta$ such that

$$\begin{aligned} e^n &:= \sqrt{\mu}(\nabla u_h^n - \nabla u(t^n)) = \sqrt{\mu}((\nabla u_h^n - \nabla P1u(t^n)) + (\nabla P1u(t^n) - \nabla u(t^n))) \\ &= \sqrt{\mu}(\nabla\theta^n + \nabla\rho^n). \end{aligned} \tag{2.54}$$

- For the estimate on $\rho$, a classical FEM estimate 2.29 is

  $$\|P1v - v\|_{L^2(\Omega)} + h\big\|\nabla(P1v - v)\big\|_{L^2(\Omega)} \leq Ch^2\|v\|_{H^2(\Omega)}, \quad v \in H^2 \cap H_0^1,$$

  which leads to

  $$\big\|\nabla\rho^n\big\| \leq Ch\big\|u(t^n)\big\|_{H^2(\Omega)},$$

  or equivalently ,

  $$\big\|\nabla\rho^n\big\| \leq Ch\bigg[\|u_0\|_{H^2(\Omega)} + \int_0^{t^n} \|u_t\|_{H^2(\Omega)}\ ds\bigg]. \tag{2.55}$$

- For the estimate on $\theta$, let us consider $v \in V_h$. Since the operators $P1$ and $\overline{\partial}$ commute, we write

  $$(\overline{\partial}\theta^n, v) + \mu(\nabla\theta^n, \nabla v) = (\overline{\partial}u_h^n, v) - (P1\overline{\partial}u(t^n), v) + \mu(\nabla u_h^n, \nabla v) - \mu(\nabla P1u(t^n), \nabla v). \tag{2.56}$$

  From (2.45) and (2.46), the equation (2.56) implies

  $$\begin{aligned} (\overline{\partial}\theta^n, v) + \mu(\nabla\theta^n, \nabla v) &= (f, v) - (P1\overline{\partial}u(t^n), v) - \mu(\nabla P1u(t^n), \nabla v), \\ &= (f, v) - (P1\overline{\partial}u(t^n), v) - \mu(\nabla u(t^n), \nabla v), \text{ by definition of } P1, \\ &= (u_t(t^n), v) - (P1\overline{\partial}u(t^n), v). \end{aligned} \tag{2.57}$$

  Then, with a triangle inequality, the equation (2.57) yields

  $$\begin{aligned} (\overline{\partial}\theta^n, v) + \mu(\nabla\theta^n, \nabla v) &= -((P1 - I)\overline{\partial}u(t^n), v) + ((\overline{\partial}u(t^n) - u_t(t^n)), v) \\ &:= -(w_1^n + w_2^n, v) = -(w^n, v). \end{aligned} \tag{2.58}$$

  Instead of replacing $v$ by $\theta^n$ as in the $L^2$ estimate, here we replace $v$ by $\overline{\partial}\theta^n$, thus the equation (2.58) takes the form

  $$(\overline{\partial}\theta^n, \overline{\partial}\theta^n) + (\nabla\theta^n, \overline{\partial}\nabla\theta^n) = -(w^n, \overline{\partial}\theta^n).$$

  Therefore, by definition of $\overline{\partial}$ for the Backward Euler discretization,

  $$\underbrace{(\overline{\partial}\theta^n, \overline{\partial}\theta^n) + \frac{\|\nabla\theta^n\|^2_{L^2(\Omega)}}{\Delta t_F} - \frac{(\nabla\theta^n, \nabla\theta^{n-1})}{\Delta t_F}}_{T_a} = -(w^n, \overline{\partial}\theta^n).$$

  With Young's inequality,

$$(\nabla\theta^n, \nabla\theta^{n-1}) \leq \frac{\|\nabla\theta^n\|^2_{L^2(\Omega)}}{2} + \frac{\left\|\nabla\theta^{n-1}\right\|^2_{L^2(\Omega)}}{2}.$$

Thus,

$$\left\|\overline{\partial}\theta^n\right\|^2_{L^2(\Omega)} + \frac{\|\nabla\theta^n\|^2_{L^2(\Omega)}}{2\Delta t_F} - \frac{\left\|\nabla\theta^{n-1}\right\|^2_{L^2(\Omega)}}{2\Delta t_F} \leq T_a \leq \frac{1}{2}\|w^n\|^2_{L^2(\Omega)} + \frac{\left\|\overline{\partial}\theta^n\right\|^2_{L^2(\Omega)}}{2}, \tag{2.59}$$

and the equation (2.59) results in

$$\left\|\overline{\partial}\theta^n\right\|^2_{L^2(\Omega)} + \frac{\|\nabla\theta^n\|^2_{L^2(\Omega)}}{\Delta t_F} \leq \frac{\left\|\nabla\theta^{n-1}\right\|^2_{L^2(\Omega)}}{\Delta t_F} + \|w^n\|^2_{L^2(\Omega)}. \tag{2.60}$$

Since $\left\|\overline{\partial}\theta^n\right\|^2_{L^2(\Omega)} \geq 0$, we rewrite (2.60) as

$$\|\nabla\theta^n\|^2_{L^2(\Omega)} \leq \left\|\nabla\theta^{n-1}\right\|^2_{L^2(\Omega)} + \Delta t_F\|w^n\|^2_{L^2(\Omega)}. \tag{2.61}$$

We recursively obtain

$$\|\nabla\theta^n\|^2_{L^2(\Omega)} \leq \left\|\nabla\theta^0\right\|^2_{L^2(\Omega)} + \Delta t_F\sum_{j=1}^{n}\left\|w^j\right\|^2_{L^2(\Omega)}, \tag{2.62}$$

and by definition of $\theta$ (and $P1$),

$$\begin{aligned}\left\|\nabla\theta^0\right\|_{L^2(\Omega)} &= \left\|\nabla u_h^0 - \nabla P1u(t^0)\right\|_{L^2(\Omega)} \leq \left\|\nabla u_h^0 - \nabla u(t^0)\right\|_{L^2(\Omega)} + \left\|\nabla u(t^0) - P1\nabla u(t^0)\right\|_{L^2(\Omega)}\\ &\leq \left\|\nabla u^0 - \nabla u(t^0)\right\|_{L^2(\Omega)} + Ch\left\|u(t^0)\right\|_{H^2(\Omega)}.\end{aligned} \tag{2.63}$$

It remains to estimate the terms in $w^j$.

-
$$\begin{aligned}w_1^j &= (P1 - I)\overline{\partial}u(t^j)\\ &= \frac{1}{\Delta t_F}(P1 - I)\int_{t^{j-1}}^{t^j} u_t\, ds,\\ &= \frac{1}{\Delta t_F}\int_{t^{j-1}}^{t^j}(P1 - I)u_t\, ds, \text{ since } P1 \text{ and the time integral commute.}\end{aligned}$$

  Thus,

$$\begin{aligned}\Delta t_F\sum_{j=1}^{n}\left\|w_1^j\right\|^2_{L^2(\Omega)} &\leq \sum_{j=1}^{n}\int_{t^{j-1}}^{t^j}\|(P1 - I)u_t\|^2_{L^2(\Omega)}\, ds,\\ &\leq Ch^4\sum_{j=1}^{n}\int_{t^{j-1}}^{t^j}\|u_t\|^2_{H^2(\Omega)}, \text{ by the definition of P1,}\\ &\leq Ch^4\int_0^{t^n}\|u_t\|^2_{H^2(\Omega)}\, ds.\end{aligned} \tag{2.64}$$

- To estimate $w_2$, we write

$$\begin{aligned} w_2^j &= \frac{1}{\Delta t_F}(u(t^j) - u(t^{j-1})) - u_t(t^j), \\ &= -\frac{1}{\Delta t_F}\int_{t^{j-1}}^{t^j}(s - t^{j-1})u_{tt}(s)\ ds, \end{aligned}$$

  such that we end up with

$$\Delta t_F \sum_{j=1}^{n}\left\| w_2^j \right\|^2_{L^2(\Omega)} \leq \sum_{j=1}^{n}\left\| \int_{t^{j-1}}^{t^j}(s - t^{j-1})u_{tt}(s)\ ds \right\|^2_{L^2(\Omega)} \leq \Delta t_F^2 \int_0^{t^n} \| u_{tt} \|^2_{L^2(\Omega)}\ ds.$$

  Combining the estimates on $\rho$ and $\theta$ concludes the proof.

□

Finally, in the same manner, we can recover the estimate in $H^2$ and $\Delta t_G^2$ with the Crank-Nicolson scheme in the $L^2$ norm (see theorem 1.6 [129]).

**Theorem 27** (Theorem 1.6 [129]). *Let $u_h^n$ be the solution associated to Crank-Nicholson discretization, and $u$ be the solution of* (2.44). *We have, if* $\left\| u_0^h - u_0 \right\| \leq Ch^2 \| u_0 \|_{H^2(\Omega)}$ *and* $u_0 = 0$ *on* $\partial\Omega$,

$$\left\| u(t^n) - u_h^n \right\|_{L^2(\Omega)} \leq Ch^2 \left[ \| u_0 \|_{H^2(\Omega)} + \int_0^{t^n} \| u_t \|_{H^2(\Omega)}\ ds \right] + C\Delta t_G^2 \int_0^{t^n} (\| u_{ttt} \|_{L^2(\Omega)} + \| \Delta u_{tt} \|_{L^2(\Omega)})\ ds, \quad \forall n \geq 0. \tag{2.65}$$

We now go on with the proof of Theorem 22.

*Proof.* The NIRB approximation is

$$u_{Hh}^{N,n}(\mathbf{x};\mu) = \sum_{i=1}^{N} \alpha_i^H(\mu, t^n)\ \Phi_i^h(\mathbf{x}),\ n \geq 0. \tag{2.66}$$

Thus, the triangle inequality gives

$$\forall t = n \times \Delta t_F, \tag{2.67}$$

$$\begin{aligned} \left\| u(t)(\mu) - u_{Hh}^{N,n}(\mu) \right\|_{H^1(\Omega)} &\leq \left\| u(t)(\mu) - u_h^n(\mu) \right\|_{H^1(\Omega)} + \left\| u_h^n(\mu) - u_{hh}^{N,n}(\mu) \right\|_{H^1(\Omega)} + \left\| u_{hh}^{N,n}(\mu) - u_{Hh}^{N,n}(\mu) \right\|_{H^1(\Omega)} \\ &=: T_1 + T_2 + T_3, \end{aligned} \tag{2.68}$$

where $u_{hh}^{N,n}(\mu) = \sum_{i=1}^{N} (u_h^n(\mu), \Phi_i^h)\ \Phi_i^h$.

- The first term $T_1$ may be estimated using the inequality (2.52), such that

$$\left\| u(t)(\mu) - u_h^n(\mu) \right\|_{H^1(\Omega)} \leq C\ (h + \Delta t_F), \tag{2.69}$$

  where $C$ does not depend on $h$.

- Here we suppose that the set of all the solutions $\mathcal{S}_h = \{u_h(\mu,t), \mu \in \mathcal{G}, t \in [0,T]\}$ has a low complexity. It means that for an accuracy $\varepsilon = \varepsilon(N)$ related to the Kolmogorov n-width of the manifold $\mathcal{S}_h$, for any $\mu \in \mathcal{G}$, and any $n \in 0, \dots, \frac{T}{\Delta t_F}$, $T_2$ is bounded by $\varepsilon$ which depends on the Kolmogorov $N$-width.

$$T_2 = \left\| u_h^n(\mu) - \sum_{i=1}^{N} (u_H^n(\mu), \Phi_i^h)\ \Phi_i^h \right\|_{H^1(\Omega)} \leq \varepsilon(N). \tag{2.70}$$

- The third term $T_3$ still depends on the method used to create the basis. We consider the approach with the Gram-Schmidt procedure and the eigenvalue problem (1.38), which yields to an orthogonalization in $L^2$ and in $H_1$. Therefore,

$$\left\| u_{hh}^N - u_{Hh}^N \right\|^2_{H^1(\Omega)} = \sum_{i=1}^{N} |(u_h^n(\mu) - u_H^n(\mu), \Phi_i^h)|^2 \left\| \Phi_i^h \right\|^2_{H^1(\Omega)}. \tag{2.71}$$

and with the normalization of the reduced basis in $L_2$, and its orthogonalization in $H^1$,

$$\left\| u_{hh}^N - u_{Hh}^N \right\|^2_{H^1(\Omega)} \leq C\lambda_N \| u_h^n(\mu) - u_H^n(\mu) \|^2_{L^2(\Omega)},$$

By the equation (2.65) and the triangle inequality, we end up with

$$\left\| u_{hh}^{N,n} - u_{Hh}^{N,n} \right\|^2_{H^1(\Omega)} \leq C\sqrt{\lambda_N}(H^2 + \Delta t_G^2), \tag{2.72}$$

where $C$ does not depend on $N$. Combining these estimates (2.69), (2.70) and (2.72) concludes the proof.

□

**Remark 28.** *On the stability. Since the NIRB error is estimated for all n, we obtain an estimate in the maximum-norm in time and with the $L^2$ norm and $H^1$ norm in space, in other words in $L^\infty(0,T;L^2(\Omega))$ and $L^\infty(0,T;H^1(\Omega))$. This is stronger than with the usual stability study of the parabolic equation* (2.44)*. For the $H^1$ norm stability, we have an $L^2$ stability in time. Indeed, from* (2.45)*,*

$$(u_t, u) + \|\nabla u\|^2 \leq |(f,v)|. \tag{2.73}$$

*From the Young and Poincaré inequalities, there exists $C > 0$ (that will change in the next inequalities) such that*

$$|(f,v)| \leq \frac{1}{2}(\|f\|^2 + \|u\|^2) \leq \frac{1}{2}C(\|f\|^2 + \|\nabla u\|^2).$$

*and since*

$$(u_t, u) = \frac{1}{2}\frac{d}{dt}\|u\|^2,$$

(2.73) *yields*

$$\frac{d}{dt}\|u\|^2 + \|\nabla u\|^2 \leq C\|f\|^2,$$

*and integrating over $(0,T)$ we end up with*

$$\|u(t)\|^2 + \int_0^T \|\nabla u(s)\|^2\ ds \leq C(\|u_0\|^2 + \int_0^T \|f(s)\|^2\ ds), \tag{2.74}$$

*which gives*

$$\|u\|^2_{L^2(0,T;H^1_0(\Omega))} \leq C(\|u_0\|^2_{L^2(\Omega)} + \|f\|^2_{L^2(0,T;L^2(\Omega))}). \tag{2.75}$$

*Thus, the stability is recovered in $L^2$ in time and in $H^1$ in space.*

**Numerical results.** With the following notation

$$F(t) = \mu\Delta u(t) + f(t),$$

the Backward Euler scheme writes

$$\frac{u^{n+1} - u^n}{\Delta t_F} = F^{n+1}, \tag{2.76}$$

which, with a test function $v \in H^1_0(\Omega)$, gives in its weak formulation

$$(u^{n+1}, v) + \ \Delta t_F \ \mu(\nabla u^{n+1}, \nabla v) \ = \ \Delta t_F(f^{n+1}, v) + (u^n, v), \tag{2.77}$$

whereas the Crank-Nicholson is defined by

$$\frac{u^{n+1} - u^n}{\Delta t_G} = \frac{1}{2}(F^{n+1} + F^n), \tag{2.78}$$

and the corresponding variational formulation reads

$$(u^{n+1}, v) + \frac{1}{2}\Delta t_G \ \mu(\nabla u^{n+1}, \nabla u^n) \ = \ \frac{1}{2}\Delta t_G((f^{n+1} + f^n, v) \ - \ \mu(\nabla u^n, \nabla v)) + (u^n, v). \tag{2.79}$$

We tested the two-grid method with the rectification posst-treatment 0.2.2 on the problem (29) with $\mathcal{G} = (0, 10]$ and the right-hand side function

$$f(t, \mathbf{x}) = 10[x^2(x-1)^2 y^2(y-1)^2 - 2t((6x^2 - 6x + 1)(y^2(y-1)^2) + (6y^2 - 6y + 1)(x^2(x-1)^2))],$$

where $\mathbf{x} = (x, y)$. The rectification post-processing step is done for each time step. Thus the NIRB with rectification is given by

$$Ru^{N,n}_{Hh}(\mu) = \sum_{i,j=1}^{N} R^n_{ij} \ \alpha^H_j(\mu, t^n) \ \Phi^h_i(\mathbf{x}), \ n \geq 0, \tag{2.80}$$

where the rectification matrix $R$ is a 3rd-order tensor time-dependent. We took 13 parameters in $\mathcal{G}$ for the RB construction and thus, we obtain $10 \times 13$ snapshots with respect to the time variable. Here follow the results with the NIRB algorithm and the rectification post-treatment for a new parameter $\mu = 1$. The exact solution is given by

$$u(t, \mathbf{x}; 1) = 10tx^2(1-x)^2 y^2(1-y)^2.$$

We implemented both schemes on FreeFem ++ and computed the NIRB rectified error in the maximum-norm. The $H^1$ NIRB rectified error is defined by

$$\max_{n=1,\ldots,(T-t_0)/\Delta t_F} \frac{\left\| u(t_0 + n\Delta t_F)(\mu) - Ru^{N,n}_{Hh}(\mu) \right\|_{H^1_0(\Omega)}}{\|u(t_0 + n\Delta t_F)(\mu)\|_{H^1_0(\Omega)}}, \tag{2.81}$$

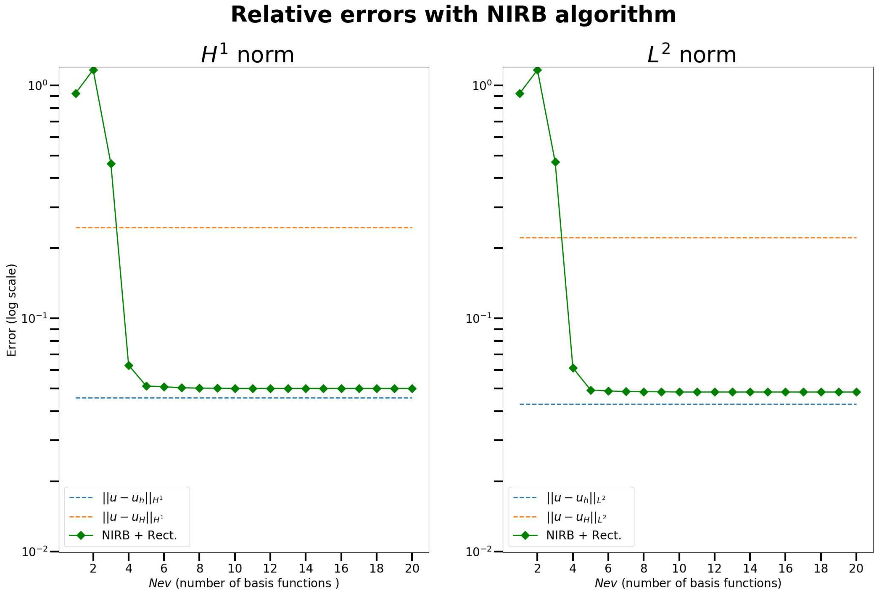


Figure 2.4: NIRB $L^\infty(0,T;H^1(\Omega))$ (left) and $L^\infty(0,T;L^2(\Omega))$ (right) relative errors with parabolic equations with a new parameter $\mu = 1$

and this error is compared to the FEM ones defined as

$$\max_{n=1,\dots,(T-t_0)/\Delta t_F} \frac{\left\| u(t_0+n\Delta t_F)(\mu) - u_h^n(\mu) \right\|_{H_0^1(\Omega)}}{\left\| u(t_0+n\Delta t_F)(\mu) \right\|_{H_0^1(\Omega)}} \text{ and } \max_{n=1,\dots,(T-t_0)/\Delta t_G} \frac{\left\| u(t_0+n\Delta t_G)(\mu) - u_H^n(\mu) \right\|_{H_0^1(\Omega)}}{\left\| u(t_0+n\Delta t_G)(\mu) \right\|_{H_0^1(\Omega)}}. \tag{2.82}$$

Since both schemes are stable, we can take $\Delta t_F \simeq h$ and $\Delta t_G \simeq H$. For the Crank-Nicholson scheme we fixed $\Delta t_G = 0.2$ and $H = 0.235702$, whereas for the Euler scheme we set $\Delta t_F = 0.1$ and $h = 0.11785$. In Figure 2.4, we plot the errors ((2.81) with the $H_0^1$-norm) as a function of the number of modes $N$ between $t_0 = 1$ and $T = 2$ and we compare these results with the FEM errors (2.82). The NIRB errors saturate very short of reaching the fine errors. In Figure 2.6, we display the NIRB approximation at several time steps.

**Remark 29.** *We may also consider NIRB aproximations of* (2.46) *under the form*

$$u_{Hh}^{N,n}(\mathbf{x};\mu) = \sum_{i=1}^{N} \alpha_i^H(\mu, t^n)\, \Phi_{h,i}^n(\mathbf{x}),\ n \geq 0, \tag{2.83}$$

*with $(\Phi_{h,i}^n)_{i=1,\dots,N}$ time-dependent basis functions. This time, the greedy algorithm is executed for each time step.*

*With this decomposition, we obtained the following results (see Figure 2.5).*

**Remark 30.** *We retrieve the fine and coarse solutions in the VTK format and use Python to execute the NIRB algorithm. For the interpolation in time and in space, we employed linear interpolations (see for instance the Python module "interpolate" with the function "griddata" for space and "interp1d" for the time interpolation).*

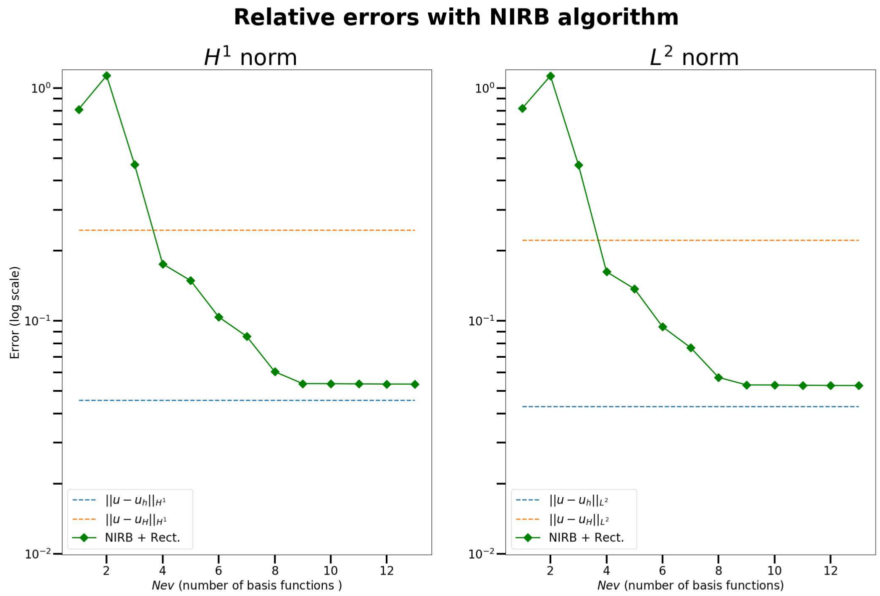
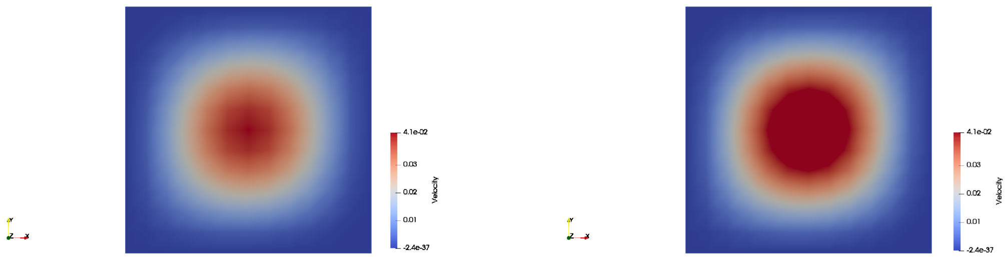
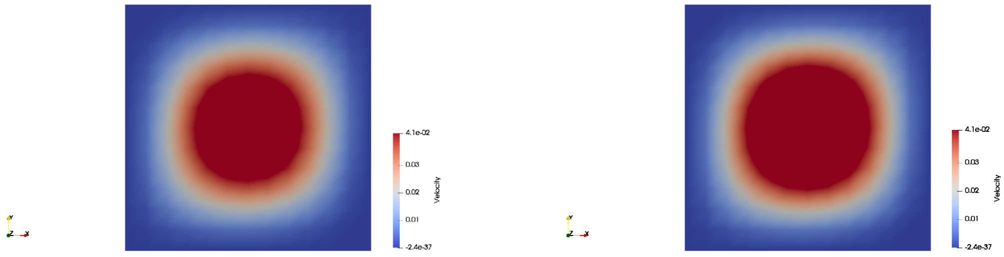


Figure 2.5: NIRB $L^\infty(0, T; H^1(\Omega))$ (left) and $L^\infty(0, T; L^2(\Omega))$ (right) relative errors with parabolic equations with a new parameter $\mu = 1$, another NIRB decomposition



(a) $n = 1$

(b) $n = 4$



(c) $n = 7$

(d) $n = 10$

Figure 2.6: Evolution of the NIRB approximation for $N = 10$, at time $t = t_0 + n\Delta t_F$, $n = 1, 4, 7$ and 10

## 2.3 NIRB with domain singularities

This part is the object of an upcoming article.

**Main idea.** In this section, the NIRB method is exploited on singular domains. The two-grid method with FE solver is applied with a new stategy in order to overcome the effects caused by the singularities. There are many studies in the litterature on re-entrant corners and domain singularities [25, 120, 16, 133, 84, 79]. Most of the developed methods are based on mesh adaptive refinement in the vicinity of the singularities to achieve accuracy [120, 1]. While MOR methods have been proposed for various fields in science and engineering, only few approaches have been developed to treat domain singularities [60, 31, 1].
Therefore, the main idea of this section is to employ the NIRB two-grid algorithm with the rectification post-treatment to approximate such problems. We take advantage of the fact that the RB methods are decomposed in two stages. All the techniques in retrieving an accurate approximation are employed in the offline step. Here, we consider refinement methods. Thus, a refinement is only operated during the offline stage. As a result, the computation times linked to such refinement are considerably reduced in the online phase. We focus on domain with re-entrant corners, and apply the NIRB two-grid method on such problems. It allows us to retrieve optimal errors in the energy norm while using a uniform coarse mesh during the online stage. We emphasize that the NIRB advantages are concerned with the size of the coarse mesh employed during the online stage as well as its uniformity. We tested this new approach on the L-shape domain and on the backward-facing step 3.1, and we present several numerical results. Let us summarize the offline/online strategy:

1. During the **"offline"** stage, the fine mesh is refined in the vicinity of the re-entrant corners, in order to obtain optimal results. This fine mesh is employed to generate the snapshots. Then the RB is created with the greedy algorithm 2 or the Snapshots POD 1.

2. Then, in the **"online"** phase, the coarse approximation is computed on a uniform coarse mesh. Afterwards, the coarse solution is projected onto the basis space $X_h^N$.

This process allows us to retrieve an accurate approximation with less degrees of freedom. For instance, with the backward-facing step 3.1, the recirculation around the step is very well captured.

**Re-entrant corners.** The NIRB two-grid method can be very sucessful to retrieve optimal error estimates, assuming that the snapshots $u_h(\mu_i)(i = 1, \cdots, N)$, given by the HF solver (with FEM or FV solvers [28, 66]) are such that (see section 2.1)

$$\left\| u(\mu_i) - u_h(\mu_i) \right\|_{H^1} \leq Ch, \forall i = 1, \ldots, N,$$

and

$$\left\| u(\mu) - u_H(\mu) \right\|_{L^2} \leq CH^2.$$

However, difficulties arise if the spatial domain $\Omega$ has singularities such as cracks or re-entrant corners. We will focus on re-entrant corners problem. They generate a local singularity on the solution and imply that the solution does not globally lie in $H^2(\Omega)$. To counterbalance these effects, some particular treatments have to be developed. For instance, we may add singular functions (xFEM) to the classical FE basis [55, 112]. Another approach is concerned with the potential function and employs boundary element methods [79]. In this application, we will consider adaptative mesh methods, which are widely used [1, 120]. They may be based on a-posteriori error estimates [23, 114, 131, 39].

Thus, let us consider three meshes:

- a uniform coarse domain, for the coarse FEM approximation,
- a fine mesh, refined around the re-entrant corner, for the snapshots,
- and a highly refined thin mesh for the reference solution.

The fine and reference meshes refinement is done in FreeFem++ with the "adaptmesh" function. It uses a variable metric/Delaunay automatic meshing algorithm [70] (bamg software). Now, the fine mesh size is defined as

$$h = \min_{K \in \mathcal{M}_h} h_K \tag{2.84}$$

where the diameter $h_K$ of any element $K$ in a mesh is equal to $\sup\limits_{x,y \in K} |x-y|, K \in \mathcal{T}_h$.

In the domain without refinement, in the presence of a re-entrant corner, it is well known [134, 16] that the solution belongs to $H^{1+2/3-\varepsilon}$. Thus, the standard FEM yields reduced convergence rates and for the coarse uniform mesh the following estimates hold

$$\begin{cases} \|u(\mu) - u_H(\mu)\|_{H^1(\Omega)} \leq CH^{2/3-\varepsilon}\|f\|_{L^2(\Omega)}, \\ \|u(\mu) - u_H(\mu)\|_{L^2(\Omega)} \leq CH^{4/3-\varepsilon}\|f\|_{L^2(\Omega)}, \end{cases} \tag{2.85}$$

This estimates are recovered with our coarse mesh and displayed in Figure 2.11. Now, if the mesh is sufficiently refined around the corner, we may retrieve a convergence in $h$ with the $H^1$ norm, as shown in Figure 2.12. Thus, it motivates two questions with the NIRB application on such problems:

1. We may wonder if it is possible to eliminate the pollution effects in order to recover classical errors on a subdomain located far away from the re-entrant angle. In other words, is there a minimum distance which allows us to eliminate the singularity effects?

2. The next question is about the direct application of the NIRB method on the whole domain. Does the NIRB method allows us to recover optimal errors on the full domain?

We will present numerical tests highlighting the fact that NIRB applied to such problems allows us to obtain accurate approximations. With the NIRB approximation, we obtain the same convergence rate as with the fine solution on the refined mesh.

**L-shape domain.** We consider $\Omega$, a plane polygonal domain with a re-entrant angle $\alpha_0 = \frac{3\pi}{2} > \pi$ at a vertex $v$. We introduce $\beta = \frac{\pi}{\alpha_0} < 1$. We consider the Poisson's equation (16a)-(16b) on $\Omega$.

For the coarse mesh, we use a uniform triangulation $\mathcal{T}_H$ of $\Omega$ as in Figure 2.15a. Figure 2.7 displays the L-shape domain FEM results.

Figure 2.11 presents the relative $H^1$ error is in $\beta = \frac{2}{3}$ and the error in $L^2(\Omega)$ is in $2\beta = \frac{4}{3}$, as it was expected with our re-entrant corner (2.85). We observe in Figure 2.7 that the error increases when getting closer to the re-entrant corner. Therefore, it brings us to the first raised question. The idea is to employ a subdomain of the uniform mesh to be able to recover the

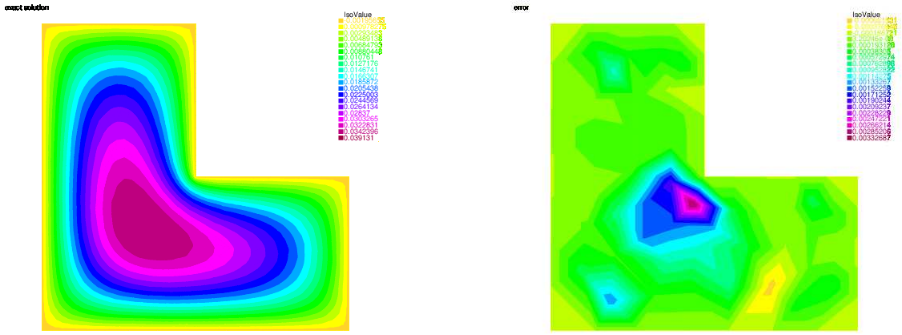


Figure 2.7: Solution on the reference mesh (left) and coarse FEM error (right)

same rate in the $L^2$ norm on this subdomain as with the Aubin-Nitsche's trick. Therefore the NIRB analysis would prove that we may recover an optimal error estimate. Thus, we are interested in retrieving a superconvergence with the $L^2$ norm at a given distance from the singularity. In fact, for any subdomain away from the singular vertex $v$, the error in $H^1$ norm converges in $\mathcal{O}(H)$ whereas for the $L^2$ norm, we cannot expect a better rate than 2 $\beta$, as we underline it in the following remark 31.

**Remark 31.** *Expected rate on a subdomain. We summarize the following result with the $L^2$ and $H^1$ norms:*
*If we consider a subdomain $\Omega_0 \subset \Omega_1 \subset \Omega$, far enough from the boundary, then if $u \in H^2(\Omega_1)$, Theorem 9.1 [134] and Theorem 5.1 [103] give*

$$\|u - u_H\|_{s,\Omega_0} \leq CH^{2-s}\|u\|_{H^2(\Omega_1)} + Ch^{2\beta-2\varepsilon}\|u\|_{H^{1+\beta}(\Omega)}, \; s = 0, 1.$$

*Thus for any subdomain away from the vertex $v$, the error in $H^1$ norm converges in $H$ and for $L^2$ we cannot expect a better rate than $2\beta$.*

Therefore, a subdomain of $\Omega$ will not yield a $L^2$ super-convergence. However, we may still apply the NIRB approach to this problem and observe the numerical results.

That brings us to the second question, and therefore, we present numerical results on this NIRB application. We sought several fine meshes with a correct refinement as in Figure 2.8b such that the error between the reference mesh and the fine one was in $\mathcal{O}(h)$ as we can see in Figure 2.12. We employ a uniform coarse mesh of size $H = 0.16$ as in Figure 2.15a.

We test two diffusion coefficients:

- With $A(\mu) = \mu$, constant in the domain. In our test, we took $\mu = 0.8$. The results are presented in Figure 2.9.
- The second test is with $A(\mu)$ piecewise-constant on $\Omega$. Therefore, the L-shape domain is decomposed in three subdomains for the varying parameter where

$$A(\mu) = \mu(\mathbf{x}) = \begin{cases} \mu_1 \text{ if } x \geq 0.5, \; y \geq 0.5 \\ \mu_2 \text{ if } x < 0.5, \; y < 0.5 \\ \mu_3 \text{ if } x \leq 0.5, \; y \geq 0.5, \end{cases}$$

(a) Coarse mesh

(b) Fine mesh

Figure 2.8: Coarse and fine meshes

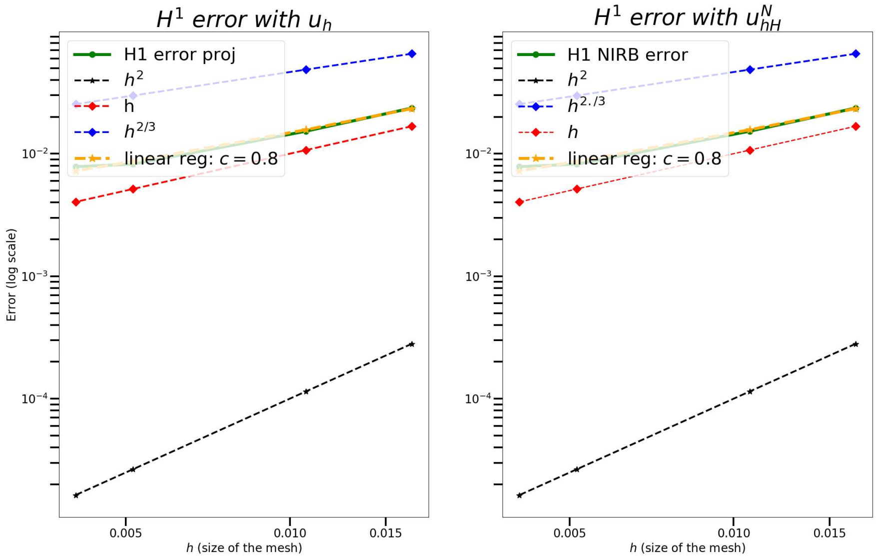


Figure 2.9: $H^1$ FEM relative error (left) and NIRB rectified error (right)

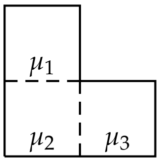


Figure 2.10: L-shape parameters domain

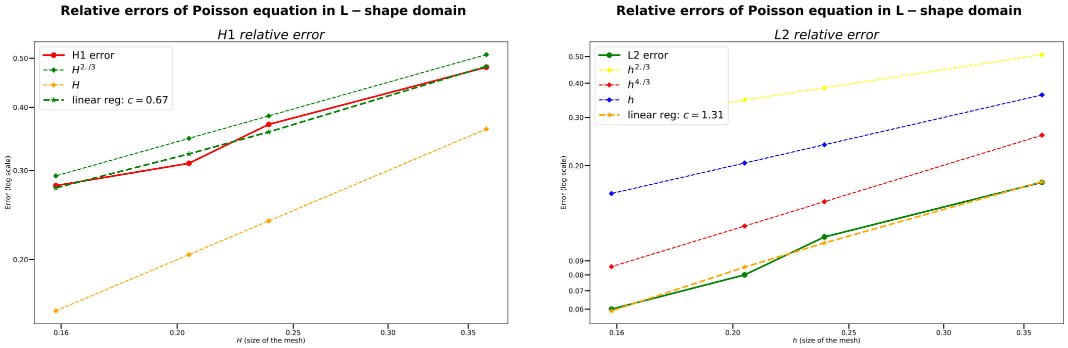


Figure 2.11: Relative errors with the coarse solution $H^1$ (left) and $L^2$ (right) norms, $A(\mu) = \mu$ constant

as in Figure 2.10. We tested the problem with $\mu_1 = 0.05,\ \mu_2 = 0.033,\ \mu_3 = 0.08$.

NIRB results are illustrated in Figure 2.12.

The NIRB method allows to recover the fine solutions and thus, it yields optimal results with an appropriate refinement, as we can see in Figure 2.9 and in Figure 2.12. This optimal bound is not certified since this optimal rate is not retrieved theoretically. We also test this new approach on the 2D backward-facing step problem and it yields a good convergence rate as well.

**2D backward-facing step problem.** We consider the 2D Backward-Facing Step (BFS) problem as described in section 3.1.

The refined reference mesh size is equal to $2.54e^{-5}$.
In the domain without refinement, the estimates (2.85) hold with a coarse and uniform mesh, as observed in Figure 2.13 with the $H^1$ semi-norm and the $L^2$ error between the reference and coarse meshes with $Re = 52$.

Now, as with the L-shape domain, if the mesh is sufficiently refined around the corner, we may retrieve a convergence in $\mathcal{O}(h)$ with the $H^1$ norm, as shown in Figure 2.14. This figure displays the $H^1$ semi-norm between the reference mesh and several fine meshes.

We sought with FreeFem++ several fine meshes with a correct refinement as in Figure 2.16a such that the error between the reference mesh and the fine ones was in $\mathcal{O}(h)$, as in Figure 2.14. We first employed a uniform coarse mesh of size $H = 0.32$ as in Figure 2.15a. In all our cases this NIRB approach preserves an optimal convergence, as shown in Figure 2.17.

The same results have been retrieved with a coarse mesh size $H = 0.7$.

### Time execution (min,sec)

We present the FEM and NIRB runtimes with the meshes as in Figure 2.15 for the fine mesh and Figure 2.16 for the coarse one.

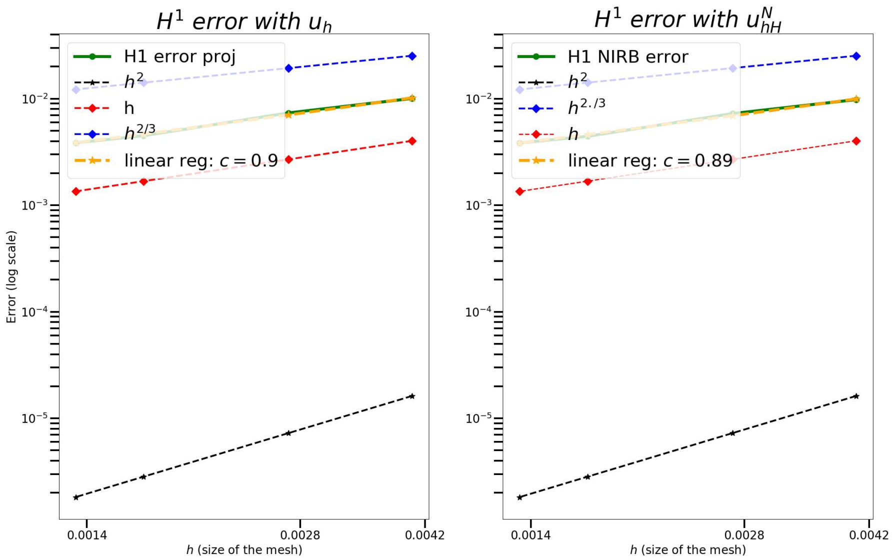


Figure 2.12: $H^1$ FEM relative error (left) and NIRB rectified error (right), $A(\mu)$ piecewise-constant

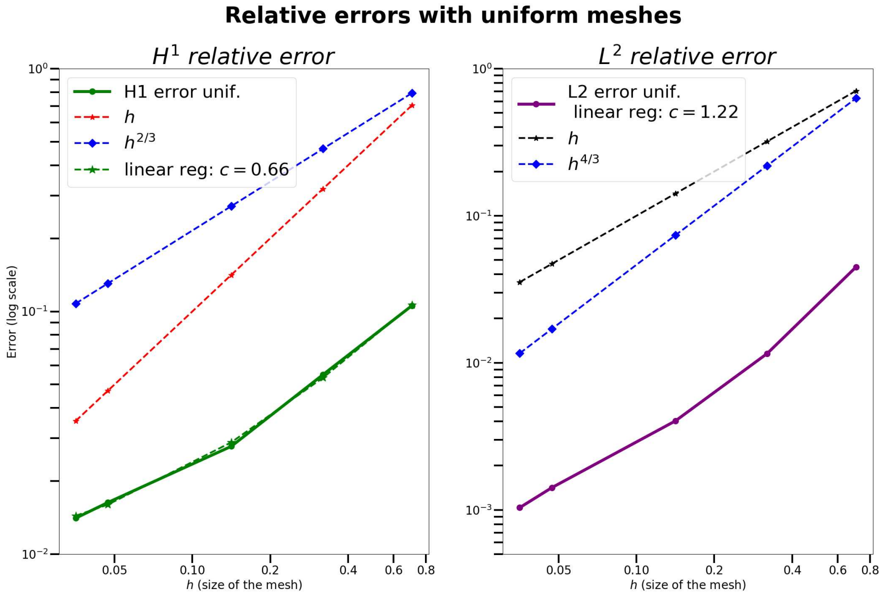


Figure 2.13: BFS: Coarse uniform domain, convergence

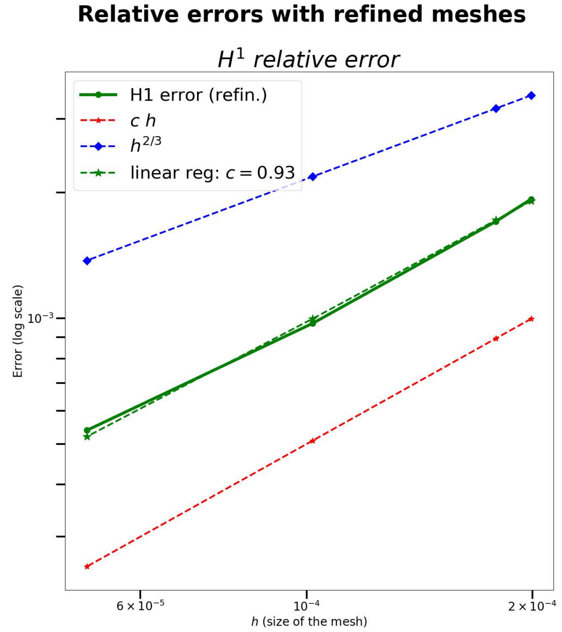
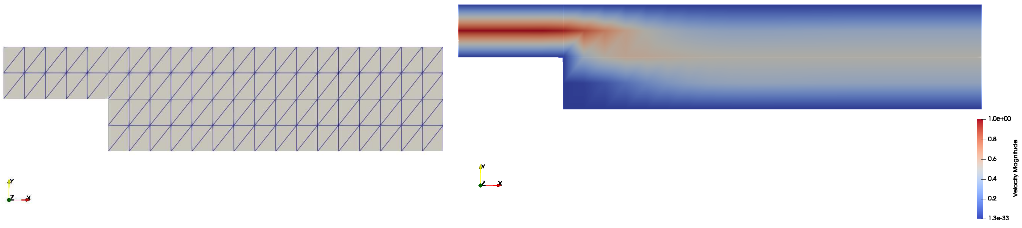


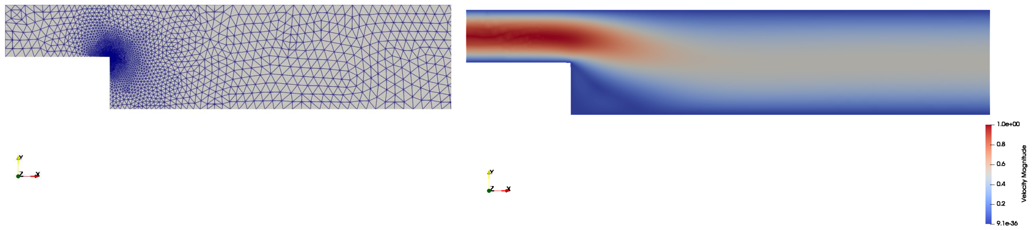
Figure 2.14: BFS: Fine domain, relative error between reference and fine meshes with $H^1$ semi-norm, $Re = 52$



(a) Coarse uniform mesh ($H = 0.1414$)

(b) FEM coarse approximation (Magnitude)

Figure 2.15: BFS: FEM coarse mesh and solution



(a) Example of fine refined mesh ($h = 0.00010187$)

(b) FEM fine approximation (Magnitude)

Figure 2.16: BFS: FEM fine mesh and solution

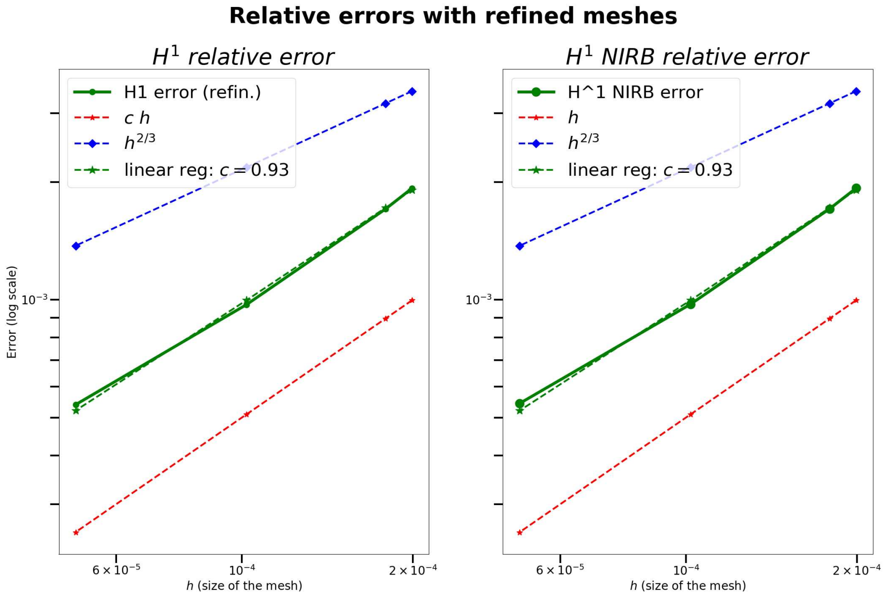


Figure 2.17: BFS: Relative $H^1$ errors between reference and fine meshes (left) vs NIRB (+ rectification) convergence (right) with $Re = 52$

| FEM HF solver | FEM coarse solution |
|---|---|
| 01:24 | 00:01 |

Table 2.1: FEM runtimes

| NIRB Offline | NIRB online |
|---|---|
| 08:34 | 00:16 |

Table 2.2: NIRB runtimes ($N = 6$)

To conclude, this NIRB approach considerably reduced the runtimes since we employ a coarse mesh and we avoid the costs of a refinement during the NIRB online stage, as observed in tables 2.1 and 2.2.

## 2.4 The two-grid method with FV solvers

This work has been published in the Journal ESAIM: Mathematical Modeling and Numerical Analysis (ESAIM: M2AN) [66].

### The FV setting for the Non-Intrusive Reduced Basis Method

Let $\Omega$ be a bounded domain in $\mathbb{R}^d$ with $d \leq 3$. The NIRB method in the context of a high fidelity solver of Finite Volume (FV) types involves two partitioned meshes, one fine mesh $\mathcal{M}_h$ and one coarse mesh $\mathcal{M}_H$, where $h$ and $H$ are the respective sizes of the meshes and $h << H$. The meshes can be more complex than with the FEM solver but we still use the same definition of the mesh size (7). The meshes are polytopal. We now recall the definition of a polytopal mesh.

**Definition 5.** *(Polytopal mesh) Let $\Omega$ be a bounded polytopal open subset of $\mathbb{R}^d (d \geq 1)$. A polytopal mesh of $\Omega$ is a quadruplet $\mathcal{T} = (\mathcal{M}, \mathcal{F}, \mathcal{P}, \mathcal{V})$, where:*

1. *$\mathcal{M}$ is a finite family of non-empty connected polytopal open disjoint subsets $K$ (the cells) such that $\overline{\Omega} = \underset{K \in \mathcal{M}}{\cup} \overline{K}$. For any $K \in \mathcal{M}$, $|K| > 0$ is the measure of $K$ and $h_K$ denotes the diameter of $K$.*

2. *$\mathcal{F} = \mathcal{F}_{int} \cup \mathcal{F}_{ext}$ is a finite family of disjoint subsets of $\overline{\Omega}$ (the edges of the mesh in 2D), such that any $\sigma \in \mathcal{F}_{int}$ is contained in $\Omega$ and any $\sigma \in \mathcal{F}_{ext}$ is contained in $\partial\Omega$. Each $\sigma \in \mathcal{F}$ is assumed to be a nonempty open subset of a hyperplane of $\mathbb{R}^d$, with a positive $(d-1)$-dimensional measure $|\sigma|$. Furthermore, for all $K \in \mathcal{M}$, there exists a subset $\mathcal{F}_K$ of $\mathcal{F}$ such that $\partial K = \underset{\sigma \in \mathcal{F}_K}{\cup} \overline{\sigma}$. We assume that for all $\sigma \in \mathcal{F}, \mathcal{M}_\sigma = \{K \in \mathcal{M} : \sigma \in \mathcal{F}_K\}$ has exactly one element and $\sigma \subset \partial\Omega$ or $\mathcal{M}_\sigma$ has two elements and $\sigma \subset \Omega$. The center of mass is $\overline{\boldsymbol{x}}_\sigma$, and, for $K \in \mathcal{M}$ and $\sigma \in \mathcal{F}_K$, $\boldsymbol{n}_{K,\sigma}$ is the (constant) unit vector normal to $\sigma$ outward to $K$.*

3. *$\mathcal{P}$ is a family of points of $\Omega$ indexed by $\mathcal{M}$ and $\mathcal{F}$, denoted by $\mathcal{P} = ((\boldsymbol{x}_K)_{K \in \mathcal{M}}, (\boldsymbol{x}_\sigma)_{\sigma \in \mathcal{F}})$, such that for all $K \in \mathcal{M}$, $\boldsymbol{x}_K \in K$ and for all $\sigma \in \mathcal{F}$, $\boldsymbol{x}_\sigma \in \sigma$. We then denote by $d_{K,\sigma}$ the signed orthogonal distance between $\boldsymbol{x}_K$ and $\sigma \in \mathcal{F}_K$, that is:*

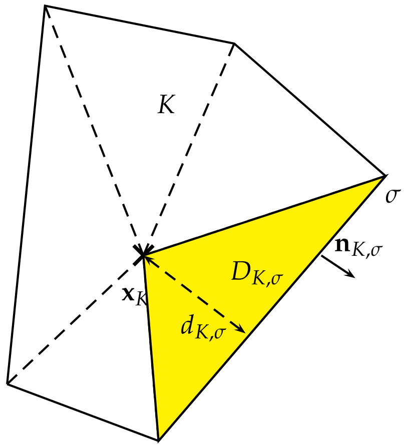


Figure 2.18: Finite Volume: A cell $K$ of a polytopal 2D mesh

$d_{K,\sigma} = (\boldsymbol{x} - \boldsymbol{x}_K) \cdot \boldsymbol{n}_{K,\sigma}$, *for all* $\boldsymbol{x} \in \sigma$. *We then assume that each cell* $K \in \mathcal{M}$ *is strictly star-shaped with respect to* $\boldsymbol{x}_K$, *that is* $d_{K,\sigma} > 0$ *for all* $\sigma \in \mathcal{F}_K$. *This implies that for all* $\boldsymbol{x} \in K$, *the line segment* $[\boldsymbol{x}_K, \boldsymbol{x}]$ *is included in K. We denote* $\overline{\boldsymbol{x}}_K$ *the center of mass of K and by* $\overline{\boldsymbol{x}}_\sigma$ *the one of* $\sigma$. *For all* $K \in \mathcal{M}$ *and* $\sigma \in \mathcal{F}_K$, *we denote by* $D_{K,\sigma}$ *the cone with vertex* $\boldsymbol{x}_K$ *and basis* $\sigma$, *that is* $D_{K,\sigma} = \{t\boldsymbol{x}_K + (1-t)\mathbf{y}, t \in (0,1), \mathbf{y} \in \sigma\}$.

4. $\mathcal{V}$ *is a set of points (the vertices of the mesh). For* $K \in \mathcal{M}$, *the set of vertices of K, i.e. the vertices contained in* $\overline{K}$, *is denoted* $\mathcal{V}_K$. *Similarly, the set of vertices of* $\sigma \in F$ *is* $\mathcal{V}_\sigma$.

The figure 2.18 illustrates a cell of a 2D polytopal mesh.
The regularity factor for polytopal meshes is

$$\theta = \max_{\sigma \in \mathcal{F}_{int}, \mathcal{M}_\sigma = \{K,K'\}} \frac{d_{K,\sigma}}{d_{K',\sigma}} + \max_{K \in \mathcal{M}} \left( \max_{\sigma \in \mathcal{F}_K} \frac{h_K}{d_{K,\sigma}} + \mathrm{Card}(\mathcal{F}_K) \right). \tag{2.86}$$

We introduced the GD setting in the introduction 2 but we now enonce the GD operators employed for HMM scheme.

**Definition 6.** *(Hybrid Mimetic Mixed gradient discretization (HMM-GD))*
*For hMFD scheme, we use the following GD (Definition 13.1.1 [45]):*

1. *Let* $X_{\mathcal{D},0} = \{v = ((v_K)_{K \in \mathcal{M}}, (v_\sigma)_{\sigma \in \mathcal{F}}) : v_K \in \mathbb{R}, v_\sigma \in \mathbb{R}, v_\sigma = 0 \text{ if } \sigma \in \mathcal{F}_{ext}\}$,
2. $\Pi_{\mathcal{D}} : X_{\mathcal{D},0} \to L^2(\Omega)$ *is the following piecewise constant reconstruction on the mesh:* $\forall v \in X_{\mathcal{D},0}, \forall K \in \mathcal{M}$,
$$\Pi_{\mathcal{D}} v(\boldsymbol{x}) = v_K \text{ on } K. \tag{2.87}$$
3. $\nabla_{\mathcal{D}} : X_{\mathcal{D},0} \to L^2(\Omega)^d$ *reconstructs piecewise constant gradients on the cones* $(D_{K,\sigma})_{K \in \mathcal{M}, \sigma \in \mathcal{F}_K}$: $\forall v \in X_{\mathcal{D},0}, \forall K \in \mathcal{M}, \forall \sigma \in \mathcal{F}$,
$$\nabla_{\mathcal{D}} v(\boldsymbol{x}) = \nabla_K v + \frac{\sqrt{d}}{d_{K,\sigma}} [\mathcal{L}_K R_K(v)]_\sigma \, \boldsymbol{n}_{K,\sigma} \text{ on } D_{K,\sigma}, \tag{2.88}$$

*where:*

- $\nabla_K v = \frac{1}{|K|} \sum_{\sigma \in \mathcal{F}_K} |\sigma| v_\sigma \boldsymbol{n}_{K,\sigma}$,
- $R_K : X_{\mathcal{D},0} \to \mathbb{R}^{\mathcal{F}_K}$ *is given by* $R_K(v) = (R_{K,\sigma(v)}))_{\sigma \in \mathcal{F}_K}$ *with* $R_{K,\sigma}(v) = v_\sigma - v_K - \nabla_K v \cdot (\overline{\boldsymbol{x}}_\sigma - \boldsymbol{x}_K)$,
- $\mathcal{L}_K$ *is an isomorphism of the space* $Im(R_K)$.

We emphasize that we will refer to $\Pi_{\mathcal{D}}^H$ on the coarse mesh and to $\Pi_{\mathcal{D}}^h$ on the fine mesh, likewise for the gradient reconstruction (respectively $\nabla_{\mathcal{D}}^H$ or $\nabla_{\mathcal{D}}^h$). The schemes hMFD [21], HFV (a hybrid version of the SUSHI scheme) [52] and MFV [43] are three different presentations of the same method [46]. With the notations above, any HMM method for the weak form (17) can be written (Equation 2.25 [46]):
Find $u_{\mathcal{T}}(\mu) \in X_{\mathcal{D},0}$ such that, for all $v_{\mathcal{T}} \in X_{\mathcal{D},0}$,

$$\sum_{K \in \mathcal{M}} |K| A_K(\mu) \nabla_K u_{\mathcal{T}} \cdot \nabla_K v_{\mathcal{T}} + \sum_{K \in \mathcal{M}} R_K(v_{\mathcal{T}})^T \mathbb{B}_K R_K(u_{\mathcal{T}}) = \sum_{K \in \mathcal{M}} v_K \int_K f(\mathbf{x}) \, d\mathbf{x},$$

where $\mu$ is our variable parameter, $A_K(\mu)$ is the $L_2$ projection of $A(\mu)$ on $K$ and $\mathbb{B}_K = ((\mathbb{B}_K)_{\sigma,\sigma'})_{\sigma,\sigma' \in \mathcal{F}_K}$ is a symmetric positive definite matrix, resulting from the definition of $\nabla_{\mathcal{D}}$.
For a certain choice of isomorphism $\mathcal{L}_K : \Im(R_K) \to \Im(R_K)$, the HMM scheme (2.4) is identical to GDs (32) (see Theorem 13.7 [45]).
We underline the fact that if $\mathbf{x}_K$ is chosen such that it is the cell center of mass, then HMM schemes boil down to hMFD [38]. In the next section, we recall the offline and the online parts of the two-grid algorithm in a FV setting.

### 2.4.1 Main steps

This section details the main steps of the two-grid method algorithm in a FV context [66].

Let $u_h(\mu)$ refer to the hMFD solution on a fine polytopal mesh $\mathcal{T}_h$, with cells $\mathcal{M}_h$ and respectively $u_H(\mu)$ the one on a coarse mesh $\mathcal{T}_H$, with the cells $\mathcal{M}_H$.

We briefly recall the NIRB method. Points 1 and 2 are performed in the offline part, and the others are done online.

1. Several snapshots $\{u_h(\mu_i)\}_{i \in \{1,\ldots N\}}$ are computed with the hMFD scheme (32), where $\mu_i \in \mathcal{G} \quad \forall i = 1, \cdots, N$.
   The space generated by the snapshots is named $X_h^N = Span\{u_h(\mu_1), \ldots, u_h(\mu_N)\}$.
2. We generate the RB functions $(\Phi_i^h)_{i=1,\cdots,N}$ with the following steps:
   - A Gram-Schmidt procedure, which involves $L^2$ orthonormalization of the reconstruction functions.
   - This procedure is also completed by the following eigenvalue problem:

$$\begin{cases} \text{Find } \Phi^h \in X_h^N, \text{ and } \lambda \in \mathbb{R} \text{ such that:} \\ \forall \mathbf{v} \in X_h^N, \int_\Omega \nabla_{\mathcal{D}}^h \Phi^h \cdot \nabla_{\mathcal{D}}^h \mathbf{v} \, d\mathbf{x} = \lambda \int_\Omega \Pi_{\mathcal{D}}^h \Phi^h \cdot \Pi_{\mathcal{D}}^h \mathbf{v} \, d\mathbf{x}, \end{cases} \tag{2.89}$$

where $\nabla_{\mathcal{D}}^h$ and $\Pi_{\mathcal{D}}^h$ are respectively the discrete gradient and the discrete reconstruction operators as in the definition of the HMM GD ((2.87), (2.88)). We get an increasing sequence of eigenvalues $\lambda_i$, and orthogonal eigenfunctions $(\Pi_{\mathcal{D}}^h \Phi_i^h)_{i=1,\cdots,N}$, orthonormalized in $L^2(\Omega)$ and orthogonalized in $H^1(\Omega)$, such that $(\Phi_i^h)_{i=1,\cdots,N}$ defines a new basis of the space $X_h^N$.

3. We solve the hMFD problem (32) on the coarse mesh $\mathcal{T}_H$ for a new parameter $\mu \in \mathcal{G}$. Let us denote by $u_H(\mu)$ the solution.

4. We then introduce $\alpha_i^H(\mu) = \int_\Omega \Pi_{\mathcal{D}}^H u_H(\mu) \cdot \Pi_{\mathcal{D}}^h \Phi_i^h \, d\mathbf{x}$. The approximation used in the two-grid method is $u_{Hh}^N(\mu) = \sum\limits_{i=1}^{N} \alpha_i^H(\mu) \, \Pi_{\mathcal{D}}^h \Phi_i^h$.

**Rectification post-process**

We introduce $\alpha_i^h(\mu) = \int_\Omega \Pi_{\mathcal{D}}^h u_h(\mu) \cdot \Pi_{\mathcal{D}}^h \Phi_i^h \, d\mathbf{x}$. The rectification process, explained in [96, 28, 30], can be employed in addition to the NIRB classical algorithm. This implies that if the true solution is in the reduced space, then the NIRB method will give this true solution. We still consider $Ntrain$ as the number of parameters in $\mathcal{G}$. Let $\mathbf{A}$ be the matrix such that $A_{i,k} = \alpha_i^H(\mu_k), \quad \forall \mu_k \in \mathcal{G}, \quad \forall i = 1, \cdots, N$, and $\mathbf{B}$ be the matrix such that $B_{i,k} = \alpha_i^h(\mu_k), \quad \forall \mu_k \in \mathcal{G}, \quad \forall i = 1, \cdots, N$. The solution of this problem with a regularization parameter is the rectification matrix:

$$\mathbf{R}_i = (\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}_N)^{-1} \mathbf{A}^T \mathbf{B}_i, \; \forall i = 1, \cdots, N, \tag{2.90}$$

where $\lambda$ is the regularization parameter.

The approximation used with this post-process is

$$u_{Hh}^N(\mu) = \sum_{i,j=1}^{N} R_{ij} \, \alpha_j^h(\mu) \, \Pi_{\mathcal{D}}^h \Phi_i^h. \tag{2.91}$$

**Remark 32.** *We consider here* $\|\cdot\|_{\mathcal{D}}$ *as the discrete semi-norm of* $H^1$ *so as not to make notations too cumbersome. The usual discrete semi-norm for* $H^1$ *is defined by*

$$\forall v \in \mathcal{T}, \; |v|_{\mathcal{T},2}^2 = \sum_{K \in \mathcal{M}} \sum_{\sigma \in \mathcal{F}_K} |\sigma| d_{K,\sigma} \left| \frac{v_\sigma - v_K}{d_{K,\sigma}} \right|^2. \tag{2.92}$$

*Under some conditions on the regularity of the mesh, this norm and* $\|\nabla_{\mathcal{D}} \cdot\|_{L^2(\Omega)^d}$ *are equivalent (Lemma 13.11 [45]).*

In the next section, we detail how to obtain the classical finite elements estimate in $\mathcal{O}(h)$ on the NIRB algorithm, when the snapshots are computed with the hMFD GD using a polytopal mesh.

### 2.4.2 NIRB error estimate

In this section, we consider $\mathbf{x}_K = \overline{\mathbf{x}}_K$ which is the case with the hMFD scheme. Some other cases will be detailed in section 2.4.3.
We now continue with the proof of theorem 10.

*Proof.* In this proof, we will denote $A \lesssim B$ for $A \leq CB$ with $C$ not depending on $h$ or $H$.

We use the triangle inequality on $\left\| u(\mu) - u^N_{Hh}(\mu) \right\|_{\mathcal{D}}$ to get

$$\begin{aligned}\left\| u(\mu) - u^N_{Hh}(\mu) \right\|_{\mathcal{D}} \leq \left\| u(\mu) - \Pi^h_{\mathcal{D}} u_h(\mu) \right\|_{\mathcal{D}} + \left\| \Pi^h_{\mathcal{D}} u_h(\mu) - u^N_{hh}(\mu) \right\|_{\mathcal{D}} + \left\| u^N_{hh}(\mu) - u^N_{Hh}(\mu) \right\|_{\mathcal{D}} \\ =: T_1 + T_2 + T_3, \end{aligned} \tag{2.93}$$

where $u^N_{hh}(\mu) = \sum\limits_{i=1}^{N} \alpha^h_i(\mu) \Pi^h_{\mathcal{D}} \Phi^h_i$,.

- The first term $T_1$ can be estimated using a classical result for finite volume schemes (Consequence of Proposition 13.16 [45]) such that:

  $$\left\| u(\mu) - \Pi^h_{\mathcal{D}} u_h(\mu) \right\|_{\mathcal{D}} \lesssim h \|u\|_{H^2(\Omega)}. \tag{2.94}$$

  For more details, see in the appendix C.0.2.

- Here the second term $T_2$ is estimated as in the FEM analysis. It relies on the Kolmogorov $N$-width (6). Here we suppose the set of all the reconstructions of the solutions $\mathcal{S}_h = \{\Pi^h_{\mathcal{D}} u_h(\mu), \mu \in \mathcal{G}\}$ has a low complexity which means for an accuracy $\varepsilon = \varepsilon(N)$ related to the Kolmogorov n-width of the manifold $\mathcal{S}_h$, there exists a set of parameters $\{\mu_1, \ldots, \mu_N\} \in \mathcal{G}$, such that [34, 96, 28, 22]

  $$T_2 = \left\| \Pi^h_{\mathcal{D}} u_h(\mu) - \sum_{i=1}^{N} \alpha^h_i(\mu) \Pi^h_{\mathcal{D}} \Phi^h_i \right\|_{\mathcal{D}} \leq \varepsilon(N). \tag{2.95}$$

- Consider the term $T_3$ now. We will need the following proposition. This proposition comes from the hMFD super-convergence property (34) and several interpolation tricks.

**Proposition 33.** *Let $u_H(\mu)$ be the solution of the hMFD on a polytopal mesh $\mathcal{T}_H$ with the unknowns on $\mathbf{x}_K = \overline{\mathbf{x}}_K$. Denote by $u(\mu)$ the exact solution of equation* (17)*, and let $(\Phi^h_i)_{i=1,\cdots,N}$ be the basis functions of the NIRB algorithm, then there exists a constant $C = C(N) > 0$ not depending on H or h, and depending on N such that*

$$\left| \int_\Omega (u(\mu) - \Pi^H_{\mathcal{D}} u_H(\mu)) \cdot \Pi^h_{\mathcal{D}} \Phi^h_i \, d\boldsymbol{x} \right| \lesssim ((\|\Phi_i\|_{L^\infty(\Omega)} + C(N)) \|u\|_{H^2(\Omega)} + \|f\|_{H^1(\Omega)}) H^2. \tag{2.96}$$

*Proof.* Since $\mathcal{M}_H$ is a partition of $\Omega$,

$$\int_\Omega \Pi^H_{\mathcal{D}} u_H(\mu) \cdot \Pi^h_{\mathcal{D}} \Phi^h_i \, d\mathbf{x} = \sum_{K \in \mathcal{M}_H} \int_K \Pi^H_{\mathcal{D}} u_H(\mu) \cdot \Pi^h_{\mathcal{D}} \Phi^h_i \, d\mathbf{x}. \tag{2.97}$$

To begin with, let $\Pi^H_0 : \mathcal{C}(\Omega) \to L^\infty(\Omega)$ be the piecewise constant projection operator on $\mathcal{M}_H$ such that:

$$\Pi^H_0 \Psi(\mathbf{x}) = \Psi(\mathbf{x}_K), \quad \text{on } K, \quad \forall K \in \mathcal{M}_H, \quad \forall \Psi \in \mathcal{C}(\Omega). \tag{2.98}$$

We use the triangle inequality on the left part of the inequality (2.96) and therefore,

$$\begin{aligned}\left|\int_\Omega (u(\mu) - \Pi_\mathcal{D}^H u_H(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h \, d\mathbf{x}\right| &\le \left|\int_\Omega (u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h \, d\mathbf{x}\right| \\ &+ \left|\int_\Omega (\Pi_0^H u(\mu) - \Pi_\mathcal{D}^H u_H(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h \, d\mathbf{x}\right|, \\ &=: T_{3,1} + T_{3,2}. \qquad (2.99)\end{aligned}$$

- We first consider the term $T_{3,1}$. But first, this requires the use of a further operator which we now introduce. Each cell $K \in \mathcal{M}_H$ is star-shaped with respect to a ball $B_K$ centered in $\mathbf{x}_K$ of radius $\rho = \min_{\sigma \in \mathcal{F}_K} d_{K,\sigma}$ (Lemma B.1 [45]). We then use an averaged Taylor polynomial as in [19] but simplified. Let us consider the following polynomial of $u(\mu)$ averaged over $B_K$:

  $$Q_K u(\mathbf{x};\mu) = \frac{1}{|B_K|}\int_{B_K} [u(\mathbf{y};\mu) + D^1 u(\mathbf{y};\mu)(\mathbf{x}-\mathbf{y})] \, d\mathbf{y}. \qquad (2.100)$$

  This polynomial is of degree less or equal to 1 in $\mathbf{x}$ .
  Let us introduce $\Pi_1^H : H^1(\Omega) \cap \mathcal{C}(\Omega) \to \mathbb{R}$, the piecewise affine projection operator on $\mathcal{M}_H$ such that:

  $$\Pi_1^H \Psi = Q_K \Psi(\mathbf{x}), \quad \text{on } K, \quad \forall K \in \mathcal{M}_H, \quad \forall \Psi \in H^1(\Omega) \cap \mathcal{C}. \qquad (2.101)$$

  With the triangle inequality, we obtain

  $$\begin{aligned}T_{3,1} &\le \left|\int_\Omega (u(\mu) - \Pi_1^H u(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h \, d\mathbf{x}\right| + \left|\int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h \, d\mathbf{x}\right|, \\ &=: T_{3,1,1} + T_{3,1,2}. \qquad (2.102)\end{aligned}$$

  * Using the Cauchy-Schwarz inequality,

    $$\begin{aligned}T_{3,1,1} &\le \int_\Omega \left|(u(\mu) - \Pi_1^H u(\mu)) \cdot \Pi_\mathcal{D}^h \Phi_i^h\right| \, d\mathbf{x}, \\ &\le \left\|u(\mu) - \Pi_1^H u(\mu)\right\|_{L^2(\Omega)} \left\|\Pi_\mathcal{D}^h \Phi_i^h\right\|_{L^2(\Omega)}, \\ &\le \left\|u(\mu) - \Pi_1^H u(\mu)\right\|_{L^2(\Omega)}, \text{ since } \Pi_\mathcal{D}^h \Phi_i^h \quad \forall i = 1, \cdots, N \text{ are normalized in } L^2. \\ &\qquad (2.103)\end{aligned}$$

    Let $K \in \mathcal{M}_H$. As in Proposition 4.3.2 [19] (for more details, see the appendix C.1, equation (C.24)),

    $$\sup_{\mathbf{x} \in \overline{K}} |u(\mathbf{x};\mu) - Q_K u(\mathbf{x};\mu)| \lesssim H_K^{2-\frac{d}{2}} |u(\mu)|_{H^2(K)}. \qquad (2.104)$$

    Since $K \subset B(\mathbf{x}, H)$ for all $\mathbf{x} \in K$,

    $$|K| \le |B(\mathbf{x}_K, H)| = |B(0,1)| H_K^d. \qquad (2.105)$$

    Thus, with the inequalities (2.105) and (2.104), we get

    $$\sup_{\mathbf{x} \in \overline{K}} |u(\mathbf{x};\mu) - Q_K u(\mathbf{x};\mu)| \lesssim H_K^2 |K|^{-\frac{1}{2}} |u(\mu)|_{H^2(K)}, \qquad (2.106)$$

taking the square and integrating over $K$, we obtain

$$\int_K |u(\mu) - \Pi_1^H u(\mu)|^2 \, d\mathbf{x} \lesssim H_K^4 |u(\mu)|_{H^2(K)}^2, \tag{2.107}$$

and summing over $K$ yields

$$\left\| u(\mu) - \Pi_1^H u(\mu) \right\|_{L^2(\Omega)} \lesssim H^2 |u(\mu)|_{H^2(\Omega)}. \tag{2.108}$$

The inequality (2.108), combined with (2.103), entails that

$$T_{3,1,1} \lesssim H^2 |u(\mu)|_{H^2(\Omega)}. \tag{2.109}$$

* The term $T_{3,1,2}$ can be estimated using a continuous reconstruction of $\Phi_i^h$, denoted by $\Phi_i$ .
With the triangle inequality,

$$\begin{aligned} \left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_{\mathcal{D}}^h \Phi_i^h \, d\mathbf{x} \right| \leq & \underbrace{\left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu))(\Pi_{\mathcal{D}}^h \Phi_i^h - \Pi_0^H \Phi_i) \, d\mathbf{x} \right|}_{a} \\ & + \underbrace{\left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i) \, d\mathbf{x} \right|}_{b}. \end{aligned} \tag{2.110}$$

Let us begin with $b$. Since $\mathbf{x}_K$ is the center of mass, $\int_K \mathbf{x} \, d\mathbf{x} = |K| \mathbf{x}_K$. Therefore,

$$\int_K Q_K u(\mathbf{x};\mu) \, d\mathbf{x} = |K| Q_K u(\mathbf{x}_K;\mu). \tag{2.111}$$

From the inequality (2.104) (case $\mathbf{x} = \mathbf{x}_K$ detailed in the appendix C.1),

$$|Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)| \lesssim H_K^{2-\frac{d}{2}} |u(\mu)|_{H^2(K)}. \tag{2.112}$$

Thus, since $\Pi_0^H \Phi_i$ is constant on each cell $K \in \mathcal{M}_H$, and $|K| \lesssim H_K^d$ (2.105),

$$\begin{aligned} \left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x} \right| &= \left| \sum_{K \in \mathcal{M}_H} \int_K (Q_K u(\mathbf{x};\mu) - u(\mathbf{x}_K;\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x} \right|, \\ &\leq \sum_{K \in \mathcal{M}_H} \left| \Phi_i(x_K) \int_K Q_K u(\mathbf{x};\mu) - u(\mathbf{x}_K;\mu) \, d\mathbf{x} \right|, \\ &\leq \sum_{K \in \mathcal{M}_H} |K| \left| \Phi_i(x_K)(Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)) \right|, \text{ from (2.111)}, \\ &\leq \|\Phi_i\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{M}_H} |K| \left| Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu) \right|, \\ &\lesssim \|\Phi_i\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{M}_H} |K| H_K^{2-\frac{d}{2}} |u(\mu)|_{H^2(K)} \text{ from (2.112)}, \\ &\lesssim \|\Phi_i\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{M}_H} H_K^{2+\frac{d}{2}} |u(\mu)|_{H^2(K)}. \end{aligned} \tag{2.113}$$

Since $\mathrm{Card}(\mathcal{M}_H) \simeq H^{-d}$, using the Cauchy-Schwarz inequality, the inequality (2.113) becomes

$$\begin{aligned}\left|\int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x}\right| &\lesssim \|\Phi_i\|_{L^\infty} H^2 \Big(\sum_{K \in \mathcal{M}_H} |u(\mu)|^2_{H^2(K)}\Big)^{\frac{1}{2}}, \\ &= \|\Phi_i\|_{L^\infty} \, |u(\mu)|_{H^2(\Omega)} H^2, \qquad (2.114)\end{aligned}$$

which implies that there exists a constant $\widetilde{C_1} > 0$ not depending on $h$ or $H$ such that (2.110) becomes

$$T_{3,1,2} \leq \underbrace{\int_\Omega \left|(\Pi_1^H u(\mu) - \Pi_0^H u(\mu))(\Pi_{\mathcal{D}}^h \Phi_i^h - \Pi_0^H \Phi_i)\right| d\mathbf{x}}_{a} + \widetilde{C_1} \|\Phi_i\|_{L^\infty} \, |u(\mu)|_{H^2(\Omega)} H^2. \qquad (2.115)$$

Let us continue with the estimate of the term $a$. From the Cauchy-Schwarz inequality and the inequality (2.115),

$$T_{3,1,2} \leq \left\|\Pi_1^H u(\mu) - \Pi_0^H u(\mu)\right\|_{L^2(\Omega)} \left\|\Pi_{\mathcal{D}}^h \Phi_i^h - \Pi_0^H \Phi_i\right\|_{L^2(\Omega)} + \widetilde{C_1} \|\Phi_i\|_{L^\infty} \, |u(\mu)|_{H^2(\Omega)} H^2. \qquad (2.116)$$

From Bramble-Hilbert's Lemma 11 (Remark 12, equation (39)) [19], we deduce that

$$\left\|u(\mu) - \Pi_0^H u(\mu)\right\|_{L^2(\Omega)} \lesssim H \|u(\mu)\|_{H^2(\Omega)}. \qquad (2.117)$$

For the first term in the right-hand side of (2.116), from (2.108)-(2.117) and the triangle inequality,

$$\begin{aligned}\left\|\Pi_1^H u(\mu) - \Pi_0^H u(\mu)\right\|_{L^2(\Omega)} &\leq \left\|\Pi_1^H u(\mu) - u(\mu)\right\|_{L^2(\Omega)} + \left\|u(\mu) - \Pi_0^H u(\mu)\right\|_{L^2(\Omega)}, \\ &\lesssim H \|u\|_{H^2(\Omega)}, \text{ neglecting the estimate in } H^2, \qquad (2.118)\end{aligned}$$

and the inequality (2.117) and the classical finite volume estimate as for (2.94) ($\Pi_{\mathcal{D}}^h \phi_i^h$ being a linear combination of the family $(\Pi_{\mathcal{D}}^h u_j^h)_{j=1}^N$, $\forall i = 1, \cdots, N$) implies that there exists $\widetilde{C_2} = \widetilde{C_2}(N) > 0$ not depending of $H$ or $h$ but depending on $N$ such that

$$\begin{aligned}\left\|\Pi_{\mathcal{D}}^h \Phi_i^h - \Pi_0^H \Phi_i\right\|_{L^2(\Omega)} &\leq \left\|\Pi_{\mathcal{D}}^h \Phi_i^h - \Phi_i\right\|_{L^2(\Omega)} + \left\|\Phi_i - \Pi_0^H \Phi_i\right\|_{L^2(\Omega)}, \\ &\leq \widetilde{C_2}(N) H, \text{ neglecting the estimate in } h. \qquad (2.119)\end{aligned}$$

From (2.118)-(2.119), we deduce that each $L^2$ term is in $\mathcal{O}(H)$ in the product of the right-hand side of (2.116). Hence the equation (2.110) yields

$$T_{3,1,2} = \left|\int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_{\mathcal{D}}^h \Phi_i^h \, d\mathbf{x}\right| \lesssim (\widetilde{C_1} \|\Phi_i\|_{L^\infty(\Omega)} + \widetilde{C_2}(N)) \|u\|_{H^2(\Omega)} H^2. \qquad (2.120)$$

- We now proceed with the estimate on $T_{3,2}$ :
  With the super-convergence property on the hMFD scheme (34), and with the normalization of $\Pi^h_{\mathcal{D}}\Phi^h_i$ in $L^2(\Omega)$

$$\left|\int_\Omega (\Pi^H_{\mathcal{D}} u_H(\mu) - \Pi^H_0 u(\mu)) \cdot \Pi^h_{\mathcal{D}}\Phi^h_i \, d\mathbf{x}\right| \leq \int_\Omega \left|(\Pi^H_{\mathcal{D}} u_H(\mu) - \Pi^H_0 u(\mu)) \cdot \Pi^h_{\mathcal{D}}\Phi^h_i\right| \, d\mathbf{x},$$
$$\leq \left\|\Pi^H_{\mathcal{D}} u_H(\mu) - \Pi^H_0 u(\mu)\right\|_{L^2(\Omega)} \left\|\Pi^h_{\mathcal{D}}\Phi^h_i\right\|_{L^2(\Omega)},$$
$$\lesssim (\|f\|_{H^1(\Omega)} + \|u\|_{H^2(\Omega)})H^2. \tag{2.121}$$

Combining the estimates (2.109)-(2.120)-(2.121) with the inequalities (2.99)-(2.102), this results in the inequality (2.96). □

We are now able to continue with the proof of theorem 10 and we consider the third term $T_3 = \left\|u^N_{hh}(\mu) - u^N_{Hh}(\mu)\right\|_{\mathcal{D}}$.

$$T_3 = \left\|\sum_{i=1}^N \alpha^h_i(\mu)\Pi^h_{\mathcal{D}}\Phi^h_i - \sum_{i=1}^N \alpha^H_i(\mu)\Pi^h_{\mathcal{D}}\Phi^h_i\right\|_{\mathcal{D}},$$
$$\leq \sum_{i=1}^N \left|\alpha^h_i(\mu) - \alpha^H_i(\mu)\right| \left\|\Pi^h_{\mathcal{D}}\Phi^h_i\right\|_{\mathcal{D}},$$
$$= \sum_{i=1}^N \left|(\Pi^h_{\mathcal{D}} u_h(\mu) - \Pi^H_{\mathcal{D}} u_H(\mu), \Pi^h_{\mathcal{D}}\Phi^h_i)_{L^2}\right| \left\|\Pi^h_{\mathcal{D}}\Phi^h_i\right\|_{\mathcal{D}}. \tag{2.122}$$

From (2.89), we get that

$$\left\|\Pi^h_{\mathcal{D}}\Phi^h_i\right\|^2_{\mathcal{D}} = \int_\Omega |\nabla_{\mathcal{D}}\Phi^h_i|^2 \, d\mathbf{x} = \lambda_i \left\|\Pi_{\mathcal{D}}\Phi^h_i\right\|^2_{L^2(\Omega)} \leq \max_{i=1,\cdots,N}(\lambda_i) = \lambda_N. \tag{2.123}$$

Therefore we obtain from (2.122) and (2.123),

$$T_3 \leq \sqrt{\lambda_N}\sum_{i=1}^N \left|(\Pi^h_{\mathcal{D}} u_h(\mu) - \Pi^H_{\mathcal{D}} u_H(\mu), \Pi^h_{\mathcal{D}}\Phi^h_i)_{L^2}\right|. \tag{2.124}$$

Using the triangle inequality in the right-hand side of (2.124),

$$T_3 \leq \sqrt{\lambda_N}\sum_{i=1}^N \left|(\Pi^h_{\mathcal{D}} u_h(\mu) - u(\mu), \Pi^h_{\mathcal{D}}\Phi^h_i)\right| + \left|(u(\mu) - \Pi^H_{\mathcal{D}} u_H(\mu), \Pi^h_{\mathcal{D}}\Phi^h_i)\right|. \tag{2.125}$$

From Proposition 1, with the estimate (2.96) applied to $\mathcal{M}_h$ and $\mathcal{M}_H$, neglecting the estimate in $\mathcal{O}(h^2)$

$$T_3 \lesssim \sqrt{\lambda_N} N((\|\Phi_i\|_{L^\infty(\Omega)} + C(N))\|u\|_{H^2(\Omega)} + \|f\|_{H^1(\Omega)})H^2. \tag{2.126}$$

The conclusion follows combining the estimates on $T_1, T_2$ and $T_3$ (estimates (2.94),(2.95) and (2.126)).

$$\left\|u(\mu) - u^N_{Hh}(\mu)\right\|_{\mathcal{D}} = \left\|u(\mu) - \sum_{i=1}^N \alpha^H_i(\mu)\Pi^h_{\mathcal{D}}\Phi^h_i\right\|_{\mathcal{D}},$$
$$\leq \varepsilon(N) + C_1 h + C_2(N)H^2 \sim \mathcal{O}(h) \text{ if } h \sim H^2. \tag{2.127}$$

□

### 2.4.3 Results on other FV schemes

In this section, we consider the case where $\mathbf{x}_K$ is not the center of mass, as it is the case for some FV schemes. Therefore the left hand side of the inequality (2.113) cannot be estimated using equation (2.111). The unknowns $\mathbf{x}_K$ are not necessarily the centers of mass of the cells neither with HMM methods nor with the Two-Point Flux Approximation (TPFA) scheme [17, 42]. Under the following superadmissibility condition

$$\forall K \in \mathcal{M}_H,\ \sigma \in \mathcal{F}_K:\ \mathbf{n}_{K,\sigma} = \frac{\overline{\mathbf{x}}_\sigma - \mathbf{x}_K}{d_{K,\sigma}}, \tag{2.128}$$

the TPFA scheme is a member of the the HMM family schemes ( [45] section 13.3 , [46] section 5.3) with the choice $\mathcal{L}_K = Id$. This leads to take $\mathbf{x}_K$ as the circumcenters of the cells with 2D triangular meshes. Theorem 1.1 holds in 2D on uniform rectangles with TPFA since the superadmissibility condition is satisfied in this case where $\mathbf{x}_K$ is the centre of mass of the cells. The TPFA scheme is rather simple to implement, and therefore we will present in the last section numerical results with a TPFA solver. We will use the definition of a local grouping of the cells as in [47] (Definition 5.1). We will extend Theorem 1.1 in the case where such groupings of cells exist.

**Definition 7.** *(Local grouping of the cells). Let $\mathcal{T}_H$ be a polytopal mesh of $\Omega$. A local grouping of the cells of $\mathcal{T}_H$ is a partition $\mathfrak{G}$ of $\mathcal{M}_H$, such that for each $G \in \mathfrak{G}$, letting $U_G := \underset{K\in G}{\cup} K$, there exists a ball $B_G \subset U_G$ such that $U_G$ is star-shaped with respect to $B_G$. This implies that for all $\boldsymbol{x} \in U_G$ and all $\mathbf{y} \in B_G$, the line segment $[\boldsymbol{x}, \mathbf{y}]$ is included in $U_G$. We then define the regularity factor of $\mathfrak{G}$*

$$\mu_G := \max_{G\in\mathfrak{G}} \ Card(G) \ + \ \max_{G\in\mathfrak{G}} \ \max_{K\in G} \ \frac{H_K}{diam(B_G)}, \tag{2.129}$$

*and, with $e_K = \overline{\boldsymbol{x}}_K - \boldsymbol{x}_K$, and*

$$e_G := \frac{1}{|U_G|} \sum_{K\in G} |K|\ \mathbf{e}_K, \quad \forall G \in \mathfrak{G}, \tag{2.130}$$

$$e_{\mathfrak{G}} := \max_{G\in\mathfrak{G}} \ |\mathbf{e}_G|\,. \tag{2.131}$$

Note that we are interested in situations where $|\mathbf{e}_G| = \left|\frac{1}{|U_G|}\sum_{K\in G}|K|\ \mathbf{e}_K\right|$ is much smaller than $|\mathbf{e}_K| \quad \forall K \in G$. The aim of this section is to estimate the left hand side of the inequality (2.113) in $\mathcal{O}(H^2)$ using a local grouping of the cells. The rest of the proof remains unchanged.

We will need the following theorem of super-convergence for HMM schemes with local grouping (Theorem 5.4 [47]).

**Theorem 34** (Super-convergence for HMM schemes with local grouping (Theorem 5.4 [47])). *Let $f \in H^1(\Omega)$, and $u(\mu)$ be the solution of (17) under assumption (33). Let $\mathcal{T}_h$ be a polytopal mesh, and $\mathcal{D}$ be an HMM gradient discretization on $\mathcal{T}_h$ and $e_{\mathfrak{G}}$ be a local grouping, and let $u_h(\mu)$ be the solution of the corresponding GD. Then, considering $u_{\mathcal{P}}(\mu)$ as the piecewise constant function on $\mathcal{M}_h$ equal to $u(\boldsymbol{x}_K;\mu)$ on $K \in \mathcal{M}$, there exists C not depending on H or h such that*

$$\left\|\Pi^h_{\mathcal{D}} u_h(\mu) - u_{\mathcal{P}}(\mu)\right\|_{L^2(\Omega)} \leq C\|f\|_{H^1(\Omega)}\,(h^2 + e_{\mathfrak{G}}). \tag{2.132}$$

**Theorem 35** (NIRB error estimate with local grouping.). *Let $u_{Hh}^N(\mu)$ be the reduced solution projected on the fine mesh and generated with the hMFD solver with the unknowns defined on $\mathbf{x}_k$ such that $e_{\mathfrak{G}}$ is in $\mathcal{O}(H^2)$ on the coarse mesh, and $u(\mu)$ be the exact solution of* (17) *under assumption* (33)*, then the following estimate holds*

$$\left\| u(\mu) - u_{Hh}^N(\mu) \right\|_{\mathcal{D}} \leq \varepsilon(N) + C_1 h + C_2(N) H^2, \tag{2.133}$$

*where $C_1$ and $C_2$ are constants independent of $h$ and $H$, $C_2$ depends on $N$, the number of functions in the basis, and $\|\cdot\|_{\mathcal{D}}$ is the discrete norm introduced in the introduction 4, and $\varepsilon$ depends of the Kolmogorov n-width. If $H$ is such as $H^2 \sim h$, and $\varepsilon(N)$ small enough, it results in an error estimate in $\mathcal{O}(h)$.*

*Proof.* In this proof, we will still denote $A \lesssim B$ for $A \leq CB$ with $C$ not depending on $h$ or $H$. The reconstruction $\Phi_i$ of $\Phi_i^h$ must belong to $W^{1,\infty}$. We emphasize on the fact that in what follows $\mathbf{x}_K \neq \overline{\mathbf{x}_K}$. We begin by the estimate $b$ in the term $T_{3,1,2}$ (2.110). The equation (2.111) still holds and with the triangle inequality,

$$\begin{aligned}
\left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x} \right| &= \left| \sum_{K \in \mathcal{M}_H} \int_K (Q_K u(\mathbf{x};\mu) - u(\mathbf{x}_K;\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x} \right|, \\
&= \left| \sum_{K \in \mathcal{M}_H} \Phi_i(\mathbf{x}_K) |K| \Big[ Q_K u(\overline{\mathbf{x}}_K;\mu) - u(\mathbf{x}_K;\mu) \Big] \right|, \\
&\leq \underbrace{\left| \sum_{K \in \mathcal{M}_H} \Phi_i(\mathbf{x}_K) |K| \Big[ Q_K u(\overline{\mathbf{x}}_K;\mu) - Q_K u(\mathbf{x}_K;\mu) \Big] \right|}_{c} \\
&\quad + \underbrace{\|\Phi_i\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{M}_H} |K| \left| Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu) \right|}_{d}.
\end{aligned} \tag{2.134}$$

We begin by the term $d$. As in the previous section (2.114),

$$\|\Phi_i\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{M}_H} |K| \left| Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu) \right| \lesssim \|\Phi_i\|_{L^\infty(\Omega)} \|u\|_{H^2(\Omega)} H^2. \tag{2.135}$$

Thus, the inequality (2.134) yields

$$\begin{aligned}
\left| \int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x} \right| &\lesssim \underbrace{\left| \sum_{K \in \mathcal{M}_H} \Phi_i(\mathbf{x}_K) |K| \Big[ Q_K u(\overline{\mathbf{x}}_K;\mu) - Q_K u(\mathbf{x}_K;\mu) \Big] \right|}_{c} \\
&\quad + \|\Phi_i\|_{L^\infty(\Omega)} \|u\|_{H^2(\Omega)} H^2.
\end{aligned} \tag{2.136}$$

Now, we continue with the first term $c$ in (2.136). With the triangle inequality,

$$\begin{aligned}
\left|\sum_{K\in\mathcal{M}_H}\Phi_i(\mathbf{x}_K)|K|\Big[Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\Big]\right| &\lesssim |\sum_{K\in\mathcal{M}_H}|K| \\
&\times\Big[\Phi_i(\mathbf{x}_G)+(\Phi_i(\mathbf{x}_K)-\Phi_i(\mathbf{x}_G))\Big]\Big[Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\Big]\ |, \\
&\lesssim \underbrace{\left|\sum_{K\in\mathcal{M}_H}\Phi_i(\mathbf{x}_G)|K|\Big[Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\Big]\right|}_{c_1} \\
&\underbrace{+\|\nabla\Phi_i\|_{L^\infty(\Omega)}\sum_{K\in\mathcal{M}_H}H_K|K|\,\big|Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\big|,}_{c_2} \\
&\text{since } \operatorname{diam}(U_G)\le\mu_G H_K\,. \qquad (2.137)
\end{aligned}$$

Using the definition of $Q_K$ (2.100), the second term, noted $c_2$, in (2.137) yields

$$\begin{aligned}
\|\nabla\Phi_i\|_{L^\infty(\Omega)}\sum_{K\in\mathcal{M}_H}H_K|K|\,\big|Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\big| &= \|\nabla\Phi_i\|_{L^\infty(\Omega)}\sum_{K\in\mathcal{M}_H}H_K\frac{|K|}{|B_K|}\left|\int_{B_K}D^1u(\mathbf{y})\cdot\mathbf{e}_K\,d\mathbf{y}\right|, \\
&\lesssim \|\nabla\Phi_i\|_{L^\infty(\Omega)}\sum_{K\in\mathcal{M}_H}H_K^2\|\nabla u\|_{L^1(B_K)}\,, \\
&\text{since } |B_K|\ge\theta_H^{-1}|K| \text{ (2.86)}, \\
&\le H^2\|\nabla\Phi_i\|_{L^\infty(\Omega)}\|\nabla u\|_{L^1(\Omega)}\,. \qquad (2.138)
\end{aligned}$$

Using the decomposition of the mesh in patches $U_G$ and with the definition of $Q_K$, the first term of (2.137), denoted $c_1$, gives

$$\begin{aligned}
\left|\sum_{K\in\mathcal{M}_H}\Phi_i(\mathbf{x}_G)|K|\Big[Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\Big]\right| &\le \left|\sum_{G\in\mathfrak{G}}\sum_{K\in G}\Phi_i(\mathbf{x}_G)\frac{|K|}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\cdot\mathbf{e}_K\,d\mathbf{y}\right|, \\
&\le \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\ \mathbf{e}_K\right|.
\end{aligned} \tag{2.139}$$

Thus (2.137) becomes

$$\begin{aligned}
\left|\sum_{K\in\mathcal{M}_H}\Phi_i(\mathbf{x}_K)|K|\Big[Q_K u(\bar{\mathbf{x}}_K;\mu)-Q_K u(\mathbf{x}_K;\mu)\Big]\right| &\lesssim \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\ \mathbf{e}_K\right| \\
&+H^2\|\nabla\Phi_i\|_{L^\infty(\Omega)}\|\nabla u\|_{L^1(\Omega)}\,.
\end{aligned} \tag{2.140}$$

Now, Lemma 7.6. in [47] is going to be used three times on the first term the right hand side of (2.140). This lemma reads:

**Lemma 36** (lemma 7.6 [47])**.** *Let $U$, $V$ and $O$ be open sets of $\mathbb{R}^d$ such that, for all $(x,\mathbf{y})\in U\times V$, $[x,\mathbf{y}]\subset O$. There exists $C$ only depending on $d$ such that, for all $\Phi\in W^{1,1}(O)$,*

$$\left|\frac{1}{|U|}\int_U\Phi(x)\,dx-\frac{1}{|V|}\int_V\Phi(x)\,dx\right|\le C\frac{diam(O)^{d+1}}{|U||V|}\int_O|\nabla\Phi(x)|\,dx. \tag{2.141}$$

We will use it successively with

$$[U,V,O]=[B_K,K,U_G],\ [U,V,O]=[K,B_G,U_G],\ \text{and } [U,V,O]=[B_G,U_G,U_G].$$

We use the triangle inequality on (2.139),

$$\begin{aligned}\sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\ d\mathbf{y}\right)|K|\ \mathbf{e}_K\right| \leq \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\Bigg|\sum_{K\in G}\Bigg(\ |\ &\frac{1}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\ d\mathbf{y}\\ &-\frac{1}{|K|}\int_K D^1u(\mathbf{y};\mu)\ d\mathbf{y}\ |\\ &+\left|\frac{1}{|K|}\int_K D^1u(\mathbf{y};\mu)\ d\mathbf{y}\ -\ \frac{1}{|B_G|}\int_{B_G}D^1u(\mathbf{y};\mu)\ d\mathbf{y}\right|\\ &+\left|\frac{1}{|B_G|}\int_{B_G}D^1u(\mathbf{y};\mu)\ d\mathbf{y}\ -\ \frac{1}{|U_G|}\int_{U_G}D^1u(\mathbf{y};\mu)\ d\mathbf{y}\right|\\ &+\frac{1}{|U_G|}\int_{U_G}D^1u(\mathbf{y};\mu)\ d\mathbf{y}\Bigg)\ |K|\ \mathbf{e}_K\Bigg|. \quad (2.142)\end{aligned}$$

and we obtain

$$\begin{aligned}\sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K}D^1u(\mathbf{y})\ d\mathbf{y}\right)|K|\ \mathbf{e}_K\right| \lesssim \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\Bigg|\sum_{K\in G}\Bigg(&\|u\|_{W^{2,1}(U_G)}\operatorname{diam}(U_G)^d\\ &\times\left[\frac{\operatorname{diam}(U_G)}{|B_K||K|}+\frac{\operatorname{diam}(U_G)}{|B_G||K|}+\frac{\operatorname{diam}(U_G)}{|U_G||B_G|}\right]+\\ &\frac{1}{|U_G|}\int_{U_G}D^1u(\mathbf{y};\mu)\ d\mathbf{y}\Bigg)\ |K|\ \mathbf{e}_K\Bigg|. \qquad (2.143)\end{aligned}$$

With the regularity factor $\theta_H$ (see the previous definition of a polytopal mesh (2.86)),

$$|K|\leq|B(0,1)|H_K^d\lesssim|B_K|\theta_H^d.$$

Since $\operatorname{Card}(G)$ is bounded by $\mu_G$,

$$\operatorname{diam}(U_G)\leq\mu_G H_K.$$

Thus, the following inequalities follow

$$\begin{gathered}\operatorname{diam}(U_G)^d\leq\mu_G^d H_K^d,\\ \frac{\operatorname{diam}(U_G)}{|B_K|}\leq C,\\ |B_G|\geq\mu_G^{-d}\operatorname{diam}(U_G)^d,\\ |B_G|\gtrsim\mu_G^{-d}H_K^d\gtrsim\mu_G^{-d}|K|,\\ \text{and } |U_G|\geq\operatorname{diam}(U_G)^d.\end{gathered}$$

Therefore (2.143) becomes

$$\begin{aligned}\sum_{G\in\mathfrak{G}} \|\Phi_i\|_{L^\infty(G)} \left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K} D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\mathbf{e}_K\right| \lesssim{}& \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\sum_{K\in G}\left(\|u\|_{W^{2,1}(U_G)}\frac{\operatorname{diam}(U_G)}{|K|}\right.\right.\\ &\left.\left.+\frac{1}{|U_G|}\int_{U_G} D^1u(\mathbf{y};\mu)\,d\mathbf{y}\right)|K|\,\mathbf{e}_K\right|. \end{aligned}\tag{2.144}$$

Since $\operatorname{diam}(U_G) \le \mu_G H_K$ and $|\mathbf{e}_K| \le H_K$,

$$\begin{aligned}\sum_{G\in\mathfrak{G}} \|\Phi_i\|_{L^\infty(G)} \left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K} D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\,\mathbf{e}_K\right| \lesssim{}& \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left[\sum_{K\in G}H_K^2\|u\|_{W^{2,1}(U_G)}\right.\\ &\left.+\left|\frac{1}{|U_G|}\sum_{K\in G}\int_{U_G} D^1u(\mathbf{y};\mu)\,d\mathbf{y}|K|\mathbf{e}_K\right|\right]. \end{aligned}\tag{2.145}$$

Then,

$$\begin{aligned}\sum_{G\in\mathfrak{G}} \|\Phi_i\|_{L^\infty(G)} \left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K} D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\mathbf{e}_K\right| \lesssim{}& \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\sum_{K\in G}H_K^2\|u\|_{W^{2,1}(U_G)}\\ &+\sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\frac{1}{|U_G|}\sum_{K\in G}|K|\,\mathbf{e}_K\right|\left|\int_{U_G} D^1u(\mathbf{y};\mu)\,d\mathbf{y}\right|, \end{aligned}\tag{2.146}$$

which implies, since $\operatorname{Card}(G) \le \mu_G$,

$$\begin{aligned}\sum_{G\in\mathfrak{G}} \|\Phi_i\|_{L^\infty(G)} \left|\sum_{K\in G}\left(\frac{1}{|B_K|}\int_{B_K} D^1u(\mathbf{y})\,d\mathbf{y}\right)|K|\mathbf{e}_K\right| \lesssim{}& \sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\,H^2\|u\|_{W^{2,1}(U_G)}\\ &+\sum_{G\in\mathfrak{G}}\|\Phi_i\|_{L^\infty(G)}\left|\frac{1}{|U_G|}\sum_{K\in G}|K|\,\mathbf{e}_K\right|\|u\|_{W^{1,1}(U_G)}. \end{aligned}\tag{2.147}$$

and finally,

$$\begin{aligned}\sum_{G\in\mathfrak{G}} \|\Phi_i\|_{L^\infty(G)} \left|\sum_{K\in G}\frac{1}{|B_K|}\int_{B_K} D^1u(\mathbf{y})\,d\mathbf{y}|K|\,\mathbf{e}_K\right| \le{}& \|\Phi_i\|_{L^\infty(\Omega)}\|u\|_{W^{2,1}(\Omega)}\,H^2\\ &+\|\Phi_i\|_{L^\infty(\Omega)}\max_{G\in\mathfrak{G}}\|u\|_{W^{1,1}(\Omega)}\left|\frac{1}{|U_G|}\sum_{K\in G}|K|\,\mathbf{e}_K\right|. \end{aligned}\tag{2.148}$$

This results using (2.134), (2.135), (2.137), (2.138), and (2.148) in

$$\begin{aligned}\left|\int_\Omega(\Pi_1^H u(\mu)-\Pi_0^H u(\mu))\cdot\Pi_0^H\Phi_i\,d\mathbf{x}\right| \lesssim{}& (\|\Phi_i\|_{W^{1,\infty}(\Omega)}\|u\|_{W^{2,1}(\Omega)}+\|u\|_{H^2(\Omega)}\|\Phi_i\|_{L^\infty(\Omega)})H^2\\ &+(\|\Phi_i\|_{L^\infty(\Omega)}\|u\|_{W^{1,1}(\Omega)})e_{\mathfrak{G}}. \end{aligned}\tag{2.149}$$

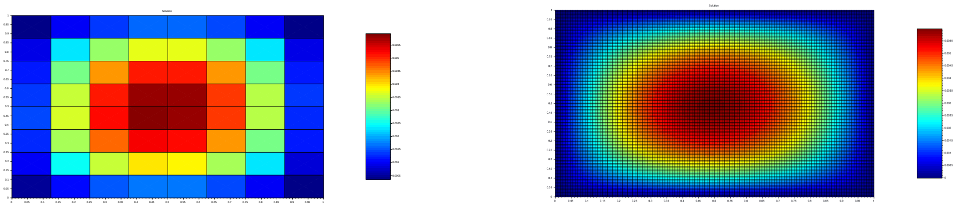


Figure 2.19: coarse and fine solution with the uniform grid

If $e_{\mathfrak{G}} = \max\limits_{G\in\mathfrak{G}} \left|\left(\frac{1}{|U_G|}\sum\limits_{K\in G} |K|\ \mathbf{e}_K\right)\right|$ is in $\mathcal{O}(H^2)$ then the estimate of

$$T_{3,1,2} = \left|\int_\Omega (\Pi_1^H u(\mu) - \Pi_0^H u(\mu)) \cdot \Pi_0^H \Phi_i \, d\mathbf{x}\right|$$

is in $\mathcal{O}(H^2)$. This concludes the proof since the rest is similar to the one of Theorem 10. Note that for the estimate of $T_{3,2}$ (2.121), the equation (2.132) from the theorem of super-convergence with local grouping is used instead of (34). ☐

### 2.4.4 Some details on the implementation and numerical results

We consider two simple cases in 2D for the numerical results with the TPFA scheme. Both results are computed on the unit square. We use an harmonic averaging of the diffusion coefficient( [46] section 5.3). Our variable parameter is $\mu \in \mathbb{R}^4 = (\mu_1, \mu_2, \mu_3, \mu_4)$. For both cases, the size of the meshes is defined as the maximum length of the edges. For the diffusion coefficient, which is isotropic, we consider here is $A(\mu_1, \mu_2) = (2\mu_1 + \mu_2 \sin(x+y)\cos(xy))$ and $f(x, y; \mu_3, \mu_4) = (\mu_3 y(1-y) + \mu_4 x(1-x))$. We choose random coefficients in $[0, 1]$ for the snapshots with $N = 5$ and our solution is defined with $\mu_1 = 0.99$, $\mu_2 = 0.8$, $\mu_3 = 0.2$, $\mu_4 = 0.78$. For the exact solution, we consider the TPFA solution on a finer mesh (Figures 2.19, 2.20). For the computation of the norm, we use the discrete semi-norm as in the remark 32 (2.92). NIRB results (with and without the rectification 2.4.1) are compared to the classical FV errors (Figures 2.21, 2.22). We measure the following relative error

$$\frac{\left\|u(\mu) - u_{Hh}^N(\mu)\right\|_{\mathcal{T},2}}{\left\|u(\mu)\right\|_{\mathcal{T},2}}. \tag{2.150}$$

In practice, one approach, based on the computation times, consists in choosing a precise time $t_1$ and in finding the associated coarse solution computed within this time. Then, the fine grid is chosen such that $H^2 = h$. In our tests, we choose several fine mesh sizes to analyze the rate of the error, and the coarse mesh size $H$ is equal to 0.25. Another approach can be to select a fine mesh size such that the method works for several coarse mesh sizes.

#### Uniform grid

The first case presents results on a rectangular uniform grid where $\mathbf{x}_K$ is the center of mass of the cell.

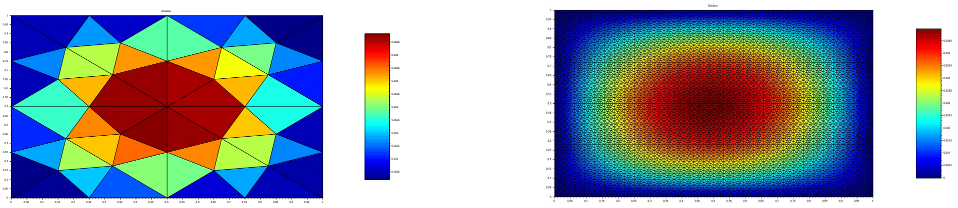


Figure 2.20: coarse and fine solution with the triangular mesh

### Triangular mesh

The second case is defined on a triangular mesh where $\mathbf{x}_K$ are the circumcenter of the cells, such that $e_{\mathfrak{G}}$ is in $\mathcal{O}(H^2)$.

### Discussion on the implementation

We implemented the TPFA scheme on Scilab and retrieved several solutions for the NIRB algorithm on Python to highlight the black box side of the solver. The scilab files consist in three text files with solution values, the cell center coordinates, and one file with information on the edges (distance $d_{KL}$, the area between the cell center and the edge, and the labels).

- Implementation of the TPFA method
  We want to solve the linear system $Au_K = b$. The TPFA on $\mathcal{T}_h$ reads:
  Find $u_h = (u_K)_{K \in \mathcal{M}}$ such that

  $$\forall K \in \mathcal{M}_h, \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}} \tau_\sigma (u_K - u_L) + \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}} \tau_\sigma u_K = \int_K f(\mathbf{x}) d\mathbf{x}, \tag{2.151}$$

  where the harmonic average $\tau_\sigma = |\sigma| \frac{A(\mathbf{x}_L;\mu) A(\mathbf{x}_K;\mu)}{A(\mathbf{x}_L;\mu) \times d_{L,\sigma} + A(\mathbf{x}_K;\mu) \times d_{K,\sigma}}$ on $\mathcal{F}_{int}$, and $\tau_\sigma = |\sigma| \frac{A(\mathbf{x}_K;\mu)}{d_{K,\sigma}}$ on $\mathcal{F}_{ext}$.

  To assemble the matrices $A$ of the TPFA scheme, we iterate on each edge, and we add the harmonic average $\tau_\sigma$ on each cell, and for $b$ we add the term $|D_{K,\sigma}| \times f(x_K)$.

- Time execution (min,sec)

| | NIRB Offline | NIRB Online | FV solver |
|---|---|---|---|
| uniform grid | 07:49 | 00:06 | 01:48 |
| triangular mesh | 06:15 | 00:05 | 01:15 |

**Remark 37.** *In dimension 2, we expect a speedup of $1/h$. Indeed, the degrees of freedom $\mathcal{N}_h$ (for the fine mesh) are of order $(1/h)^2$ (resp. $\mathcal{N}_H = (1/H)^2$ for the coarse mesh), and the costs of an optimal solver are in $\mathcal{O}(\mathcal{N}_h)$ (or $\mathcal{O}(\mathcal{N}_H)$ for the coarse mesh). Thus the speedup with $h = H^2$ is equal to $1/h$ and differs from other classical reduced-basis methods. In our case, this is difficult to observe since our model problem is very simple with few degrees of freedom, and the computational costs take into account other subroutines such as mesh readers which are not proportional.*

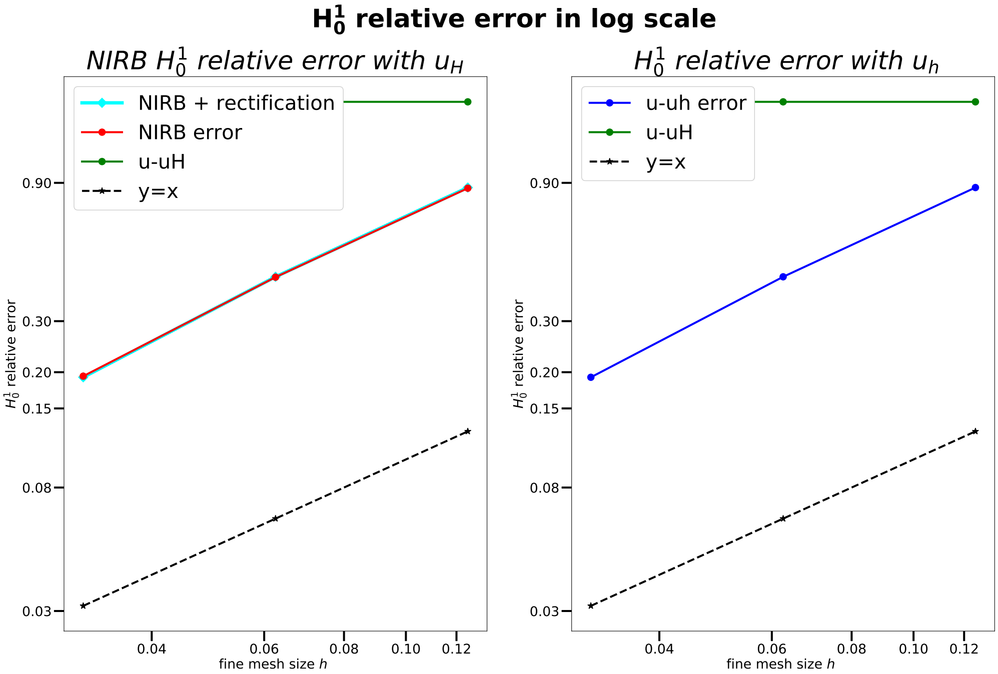


Figure 2.21: Numerical result on the uniform grid

**Remark 38.** *Note that for the discontinuous diffusion coefficient $A$, with the TPFA scheme, we recovered numerically the same estimate as in the Lipschitz continuous case, when we use the harmonic mean even if the proof no longer works.*

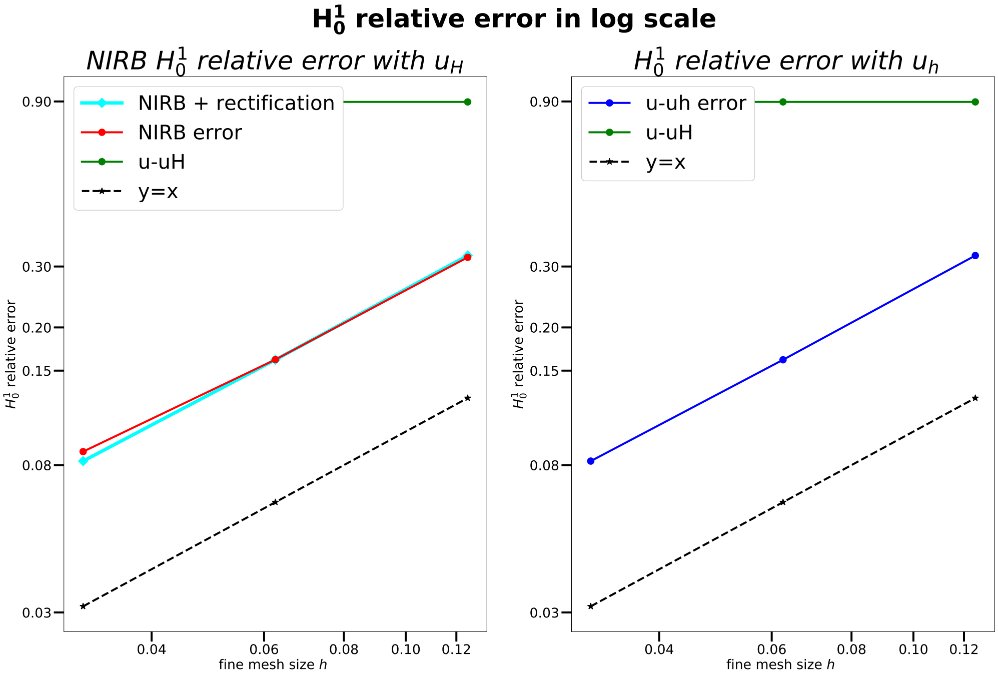


Figure 2.22: Numerical result on the triangular mesh

# Chapter 3

# New NIRB tools

The running example of this section is the 2D backward facing step as the model problem 3.1. We will present two new NIRB tools developed during this thesis, which can be applied to very general problems of the form (2).

- The first tool is a minimization problem under constraints. As said in section 1.4, the NIRB approximation is the $L^2$ projection of the coarse solution onto the basis. Thus, it is equivalent to a minimization problem. Constraints on the coefficients are added in order to rectify the approximation. There are two ways of adding these constraints that we will detail.

- The second one is a new method that uses two RB and which reduces the solution twice. The main purpose of this tool is to enable domain truncations on the spatial domain for the coarse solution. Thus, it produces a significant decrease of computational costs.

We will present numerical results on the model problem for both methods.

$$\boxed{\text{The varying parameter is the Reynolds number } \mu = Re \in \mathcal{G} = [30, 300].}$$

Let us first describe the model problem.

## 3.1 Description of the model problem

The backward-facing step problem constitutes an important branch of fundamental fluid mechanics, and has been widely studied [50, 92, 76, 48, 90, 65]. Let us consider as a model problem a fluid passing through a 2D channel $\Omega$ with a descending stair. The modelized channel, which is already a truncation of an infinite channel, is described by $\Omega = (0, L_1] \times (H_1 - H_0, H_0) \cup (L_1, L) \times (0, H_0)$ as shown in Figure 3.1.

The fluid has a parabolic entering flow profile of maximum unit speed, zero speed on the top and bottom part, and a Neumann type condition at the outlet (i.e. $\nu \partial_n u = pn$). The flow is laminar and incompressible.

Therefore, we consider the following parameter-dependent problem as our model problem: Find the velocity of the incompressible fluid $\mathbf{u} \in H^1(\Omega)^2$ and the pressure $p \in L^2(\Omega)$, solutions

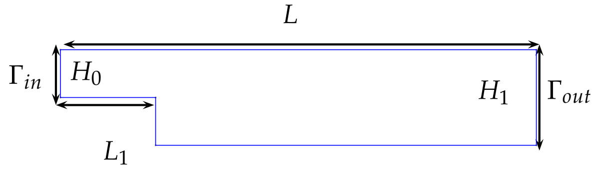


Figure 3.1: The flow's channel

of the steady Navier-Stokes equation

$$\left\{\begin{array}{rcl} \mathbf{u}.\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p &=& 0 \text{ on } \Omega, \\ \nabla.\mathbf{u} &=& 0 \text{ on } \Omega, \\ \mathbf{u}(\mathbf{x}) &=& \begin{bmatrix} \frac{4}{H_0^2}\Big(H_1 - x_2\Big)\Big(x_2 - H_1 + H_0\Big) \\ 0 \end{bmatrix} \text{ on } \Gamma_{int}, \\ \nu\partial_{\mathbf{n}}\mathbf{u}_{|\Gamma} - p\mathbf{n} &=& 0 \text{ on } \Gamma_{out}, \\ \mathbf{u}(\mathbf{x}) &=& 0, \text{ on } \partial\Omega\backslash(\Gamma_{out}\cup\Gamma_{in}), \end{array}\right. \tag{3.1}$$

where $\mathbf{x} = (x_1, x_2) \in \Omega$, $\mathbf{u} = (u_1, u_2)$, and $\nu = \frac{1}{Re}$, $Re$ beeing the Reynolds number. The solution of this problem is uniquely defined by the parameters $[\nu, H_0, L_1, H_1, L]$. We consider $\mu := Re \in \mathcal{G} = [30, 300]$ as the problem variable parameter. Indeed, for $Re > 300$, no stable steady approximation exists with the geometric parameters we are going to consider, since the solutions are strongly affected by the nonlinearity.

## 3.2 NIRB constrained version

The first tool developed during this thesis is a constrained version of the NIRB method. We use a RB , denoted $(\Phi_i^h)_{i=1,\dots,N}$, derived from a SVD or from a Gram-Schmidt procedure to ensure an $L^2$-orthonormalization. The NIRB approximation corresponds to the $L^2$-projection of the coarse solution onto the basis 1.4. In mathematical terms, this is equivalent to the following minimization problem

$$\min_{\alpha=(\alpha_1,\dots,\alpha_N)} \left\|\mathbf{u}_H - \sum_{i=1}^{N} \alpha_i\ \Phi_i^h\right\|_{L^2}. \tag{3.2}$$

as a function of the coefficients. We recall that the fine and coarse coefficients are denoted $\alpha_i^h$ and $\alpha_i^H$ and they reads (9)

$$\alpha_i^h(\mu) = \int_\Omega \mathbf{u}_h(\mu)\cdot\Phi_i^h\ d\mathbf{x} \text{ and } \alpha_i^H(\mu) = \int_\Omega \mathbf{u}_H(\mu)\cdot\Phi_i^h\ d\mathbf{x}. \tag{3.3}$$

Once squared, and since the minimization does not depend on $\mathbf{u}_H$, equation (3.2) yields

$$\min_{\alpha=(\alpha_1,\dots,\alpha_N)} (\mathbf{u}_H - \sum_{i=1}^{N} \alpha_i\ \Phi_i^h,\ \sum_{j=1}^{N} \alpha_j\ \Phi_j^h). \tag{3.4}$$

Thus, the equation (3.4) entails (see the appendix D for more details)

$$(\mathbf{u}_H - \sum_{i=1}^{N} \alpha_i \, \Phi_i^h, \, \Phi_j^h) = 0, \, \forall j = 1, \dots, N. \tag{3.5}$$

From equation (3.5), we end up with the following linear problem

$$\mathbb{A}\alpha + \mathbf{b} = 0, \tag{3.6}$$

where $A_{ij} = (\Phi_i^h, \Phi_j^h) = \delta_{ij}$, and $b_i = -(\mathbf{u}_H, \Phi_j^h)$.

**Remark 39.** *In our tests bellow, we used the python function "minimize" from the module "scipy.optimize" to solve this problem.*

We recall that the RB errors (for a parameter involved in the RB) write

$$\left\|\mathbf{u}_h(\mu_i) - \mathbf{u}_{hh}^N(\mu_i)\right\|_{H^1(\Omega)} = \left\|\mathbf{u}_h(\mu_i) - \sum_{i=1}^{N} \alpha_i^h(\mu_i) \, \Phi_i^h\right\|_{H^1(\Omega)}, \, \forall i = 1, \dots, N. \tag{3.7}$$

The optimal RB approximation is the $L^2$-projection of the fine solution onto the basis (see the section on the POD 1.2.1). With the NIRB method, the coefficients (3.3) are obtained with the coarse approximation in order to reduce the time computation (see section 1.4). Thus, in order to rectify the NIRB approximation we propose to add constraints on the coefficients. There are two ways of computing the coefficients:

- The first test on the problem model has been done with decreasing coefficients. Indeed, it follows from equation (1.17) (see the POD section for further details) that the true coefficients are decreasing on average. Thus, we assume that the coefficients of the optimal NIRB approximation are decreasing in absolute value. Hence, we can use a new algorithm that minimizes the error between the rough solution and the NIRB approximation (3.2) as a function of the coefficients, while decreasing them. Thus, we seek $(\alpha_i)_{i=1,\dots,N}$ minimizing the problem (3.2) such that

  $$-|\alpha_i| \leq \alpha_{i+1} \leq |\alpha_i|, \text{ for } i = 1, \dots, N-1. \tag{3.8}$$

  Formulation (3.2) with the constraints (3.8) reads as a constrained programming problem where the objective function corresponds to the $L^2$-norm of the error between the coarse FEM solution and the optimal projection, while the constraints impose that each coefficient of the N-term expansion are decreasing. This algorithm is not efficient enough to recover the same error as with the classical NIRB approximation as shown in Figure 3.2, but it may stabilize the method for very large $N$ due to the decrease of the coefficients. The results are not accurate because of the regularization parameter and the fact that the decrease of the coefficients is not strong enough. One possible idea which we have not tested yet to overcome this effect would be to put weights on the coefficients. We also found another way to constrain the coefficients.

- The RB coefficients are the ones used in (3.7) for each snapshot. They correspond to the $L^2$-inner product between the fine snapshots and the fine basis functions. We assume that all the snapshots belong to the same manifold (4). The RB coefficients can be interpreted as a deterministic sample. Then, for a new parameter, we can imagine than the fine

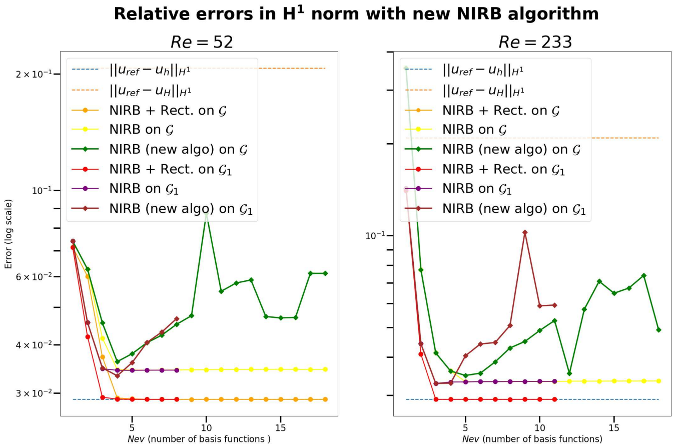


Figure 3.2: Relative NIRB $H^1$ errors for the first method

coefficients will belong to the range of the RB coefficients. We emphasize the fact that the RB coefficients are decreasing (in absolute value) on average. We define two bounds as in [54]:

$$m_i = \min_{j=1,\dots,Ntrain} (\mathbf{u}_h(\mu_j), \Phi_i^h), \tag{3.9}$$

$$M_i = \max_{j=1,\dots,Ntrain} (\mathbf{u}_h(\mu_j), \Phi_i^h). \tag{3.10}$$

Figure 3.3 illustrates the range of the RB coefficients and for a new parameter $Re = 52$ (with $\mathcal{G}_1$) the NIRB coefficients obtained with the minimization problem (3.2) (compared to the rectification posttreatment and the classical NIRB coefficients). We already now from several tests in chapter 2 that the rectification post-treatment yields to accurate results and that the rectified coefficients are very close to the optimal ones. In Figure 3.3, we observe that this NIRB minization tends to mimic the NIRB coefficients generated by the rectification postprocessing stage.

Figure 3.4 displays the relative $H^1$ errors. We recover the classical NIRB error with this new algorithm. Moreover, it stabilizes the NIRB error when $N$ is large but the results are less accurate than with the rectification post-treatment. Thus, it can be a good alternative of the NIRB method when the coarse snapshots are not available. The online runtime is compared to those of the classical NIRB in the table 3.3 (the method employs the same offline procedure).

| FEM high fidelity solver | FEM coarse solution |
|---|---|
| 00:43 | 00:01 |

Table 3.1: FEM runtimes (min-sec)

| NIRB Offline | classical NIRB online |
|---|---|
| 14:30 | 00:26 |

Table 3.2: NIRB runtimes ($N = 18$)

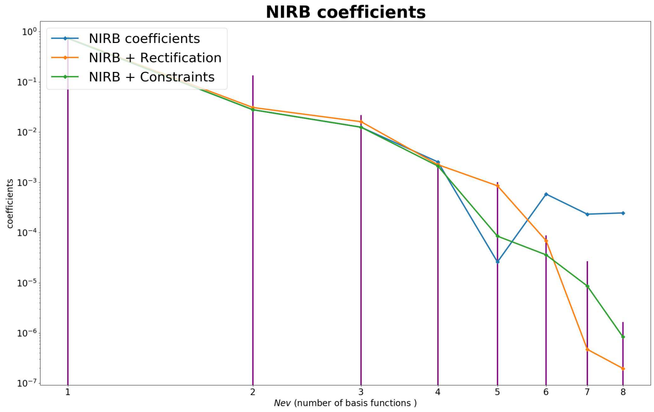


Figure 3.3: NIRB coefficients (absolute value). The vertical lines correspond to the RB bounds. Three kinds of coefficients: classical NIRB coefficients (blue line), Rectification post-treatment coefficients (yellow line), and constrained version (green line). Constrainted coefficients are much more closer to the rectified ones compared to classical NIRB.

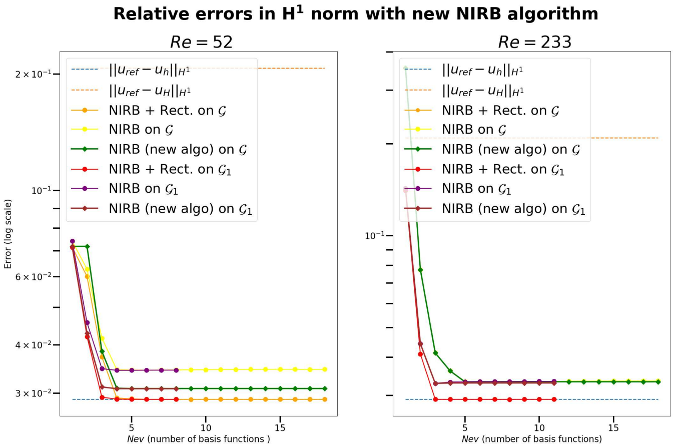


Figure 3.4: NIRB relative $H^1$ errors for $Re = 52$ (left) and $Re = 233$ (right): Classical NIRB, NIRB + rectification post-treatment vs new algorithm with the second method

| new NIRB online |
|---|
| 00:35 |

Table 3.3: online runtime ($N = 18$)

**Remark 40.** *We also tested this method on the 2D lid driven cavity problem (as in the 1) and on the wind farms 2D problems 4 and we observed the same results of stabilization.*

Here follows the algorithm (offline and online parts) for this constrained version.

```
Data: $(\mu_1, \dots, \mu_{Ntrain}) \in \mathcal{G}$
Result: $\Phi_1, \dots, \Phi_N, m_1, \dots, m_N, M_1, \dots, M_N$
initialization: Construct the snapshots $(u_h(\mu_1)(\mathbf{x}), \dots, u_h(\mu_{Ntrain})(\mathbf{x}))$; Use the greedy
    algorithm 2 and set $X_h^N = Span\{\Psi_1^h(\mathbf{x}), \dots, \Psi_N^h(\mathbf{x})\}$
Solve equation (1.38) in order to obtain $(\Phi_1^h, \dots, \Phi_N^h)$.
Compute $m_i = \underset{j \in \mathcal{G}}{min}\ (u_h(\mu_j), \Phi_i^h)$ and $M_i = \underset{j \in \mathcal{G}}{max}\ (u_h(\mu_j), \Phi_i^h)$, $i = 1, \dots, N$.
```

**Algorithm 5:** NIRB Offline algorithm with constraints

```
Data: $\{\Phi_1^h, \dots, \Phi_N^h\}, \mu$ (a new parameter in $\mathcal{G}$)
Result: $u_h^N(\mu)$
initialization: Solve $u_H(\mu)$ on the coarse mesh
Solve (3.6) under the constraints $m_i \leq \alpha_i \leq M_i$
Set $u_{Hh}^N(\mu) = \sum_{i=1}^N \alpha_i\ \Phi_i^h$
```

**Algorithm 6:** Constrained NIRB online algorithm

## 3.3 A double reduced basis method based on a domain truncation

This section is part of an upcoming article.

**Main idea.** As explained in the introduction 2, this method exploits the NIRB benefits and further enhances this approach by using domain truncations. The truncation is done a-priori, and for now without analysis. The a-posteriori study B in the annex (still in progress), could be used for this purpose and should also allows us to enrich the reduced basis, if necessary. Currently, in order to find an appropriate truncation, a prior knowledge of the whole domain and of the physical phenomena is required. To the best of our knowledge, this idea has not already been developed with NIRB methods. We adapt the NIRB two-grid algorithm with two RB and a deterministic procedure that goes from one to the other. The main interest of this new online stage is that it greatly reduces its runtime. Indeed, during the online stage, this process allows us to consider a domain truncation to reduce the degrees of freedom of the coarse solution, and it subsequently increases the speed-up of the NIRB algorithm. This new approach can be employed with many parameter-dependent problems, even when the variable parameter depends on the geometry.

**Application to the model problem.** In what follows, we applied this method to the model problem 3.1. The channel domain is supposed to be infinite, with upper and lower walls. To cope with this infinite domain, we use particular boundary conditions, as described in 3.1. This new approach allows us to further reduce the length of the channel during the online stage, and thus to substantially reduce the degrees of freedom. It may also lead to a reduction of the error between the fine solution and the reduced approximation, compared to the one obtained with the classical NIRB two-grid method with the rectification post-treatment, as shown by the results obtained on the model problem in subsection 3.3.2. As explained in the introduction, the method is decomposed in two stages. In order to find the number of modes required for the fine and the coarse basis, we use the definition of the RIC [3] (1.19), denoted $I(N)$, where $N$ is the number of modes. The number of required basis functions can be a priori given or chosen such that the RIC is close to one. It represents the fraction of variance of the correlation matrix that can be recovered using a specific number $N$ of basis functions. In practice, we choose $\eta$ a priori and we find $N_1$ for the fine snapshots (resp. $N_2$ for the coarse ones), such that

$$I(N_1) < 1 - \eta^2. \tag{3.11}$$

Once these numbers have been fixed, the offline stage is performed only once.

We will truncate the domain $\Omega$ according to the parameter $L \in [1, 5]$. We analyzed the benefits and the limits of this new NIRB algorithm by varying the parameter $L$ for two Reynolds values in $\mathcal{G}$, but first, let us present the full algorithm.

### 3.3.1 Full algorithm

This section detailes the full algorithm of the NIRB method with domain truncations.

Let $\mathbf{u}_h(\mu)$ be the HF solutions on the fine mesh of $\Omega$, $\mathcal{T}_h$ and respectively $\mathbf{u}_H(\mu)$ be the rough solutions on $\mathcal{T}_H$, the coarse mesh of $\omega \subset \Omega$.

The first points belong to the offline part, and the two last ones are done online.

1. Several fine snapshots $\{\mathbf{u}_h(\mu_i)\}_{i\in\{1,\ldots N_{train}\}}$ and the corresponding coarse ones $\{\mathbf{u}_H(\mu_i)\}_{i\in\{1,\ldots N_{train}\}}$ are computed with the solver.

2. We generate two spaces $X_h^{N_1} = \text{span}\{\mathbf{u}_h(\mu_1), \ldots, \mathbf{u}_h(\mu_{N_1})\}$ and $X_H^{N_2} = \text{span}\{\mathbf{u}_H(\mu_1), \ldots, \mathbf{u}_H(\mu_{N_2})\}$ with the fine and coarse snapshots and we create the basis functions $(\Phi_i^h)_{i=1,\cdots,N_1}$ and $(\Phi_i^H)_{i=1,\cdots,N_2}$ through two SVD approaches or with two greedy procedures (see algorithm 2 in the case of the whole domain). $N_1$ and $N_2$ can be a priori chosen or the equation (3.11) can be used.

3. To not make the notations too cumbersome, we introduce :

   - The vectors $\mathbb{A}_i \in \mathbb{R}^{N_{train}}$ for all $i = 1, \cdots, N_2$, such that

   $$(\mathbb{A}_i)_k = (\mathbf{u}_H(\mu_k), \Phi_i^H)_{L^2}, \forall k = 1, \cdots, N_{train}, \tag{3.12}$$

   - The vectors $\mathbb{B}_j \in \mathbb{R}^{N_{train}}$ for all $j = 1, \cdots, N_1$, such that

   $$(\mathbb{B}_j)_k = (\mathbf{u}_h(\mu_k), \Phi_j^h)_{L^2}, \forall k = 1, \cdots, N_{train}, \tag{3.13}$$

- The matrix $\mathcal{D} = \begin{pmatrix} \mathbb{A}_1 \\ \vdots \\ \mathbb{A}_N \end{pmatrix} \in \mathbb{R}^{N_{train} \times N_2}$,

The rectification matrix with a regularization parameter [130] is given by

$$\mathcal{R}_i = (\mathcal{D}^T\mathcal{D} + \lambda \mathbf{I}_{N_2})^{-1}\mathcal{D}^T\mathbb{B}_i, \ \forall i = 1, \cdots, N_1, \tag{3.14}$$

where $\lambda$ is the regularization parameter.

4. We solve the parameterized problem (the equation (3.1) for the model problem) with the solver on the coarse mesh $\mathcal{T}_H$ for a new parameter $\mu \in \mathcal{S}$. Let us denote by $\mathbf{u}_H(\mu)$ the solution.

5. The approximation used in this method is

$$u_{Hh}^N(\mu) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \mathcal{R}_{ij} \ (\mathbf{u}_H(\mu), \Phi_j^H)_{L^2(\omega)} \ \Phi_i^h. \tag{3.15}$$

### 3.3.2 Numerical results on the model problem

We now go back to our model problem (3.1). For the fine mesh, we will consider the reference values $H_0 = 0.5$, $H_1 = 1, L_1 = 1$, and $L = 5$ (In our tests, these values for the fine mesh will not change. In what follows, we will write $L$ for the length of the subdomain $\omega$). We recall that $Re \in \mathcal{G} = [30, 300]$ is the problem variable parameter, and the domain $\Omega$ is truncated according to the parameter $L \in [1, 5]$. The size of the coarse mesh $H < 1$ is such that $H^2 \simeq h$. For instance, the figure 3.5 represents the case where $L$ varies ($L = 3$):

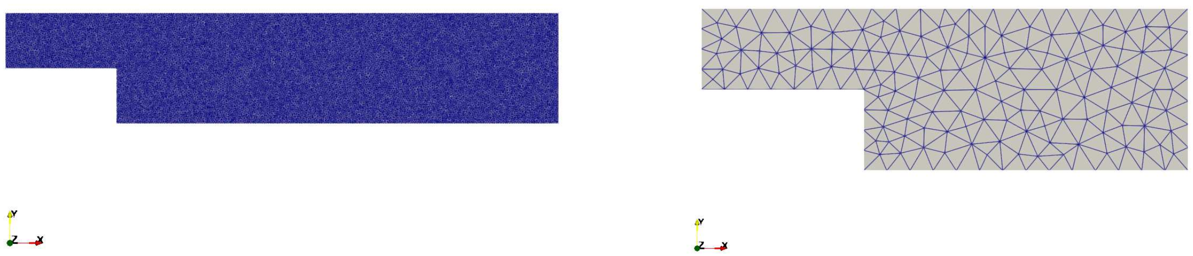


Figure 3.5: Fine mesh on $\Omega$ (right) and coarse truncated mesh on $\omega$ with $L = 3$ (left): subdomain $\omega$ framed in green (left) and the coarse mesh on $\omega$ with $H = 0.2$ (right), not on the same scale

The Navier-Stokes equations involved in this problem lead to a vortex at the bottom to the step for any parameter in $\mathcal{G}$, and the recirculation length increases when the Reynolds number increases. Thus, we already know the physical phenomena involved here for a new parameter in $\mathcal{G}$. The domain $\Omega$ can be decomposed into three subdomains where several phenomena take place:

- the entrance, where the flow has a parabolic profile. Therefore, before the step, the velocity is almost independent of $x_1$ and can be approximated by a Poiseuille's flow.
- the vicinity of the step where we observe a recirculation.
- the outlet, where the flow can roughly be approximated by a Poiseuille's flow after the recirculation.

With a higher Reynolds number, the recirculation after the step is expanded, as we can see in Figure 3.6. Thus, we will also consider two subgroups of $\mathcal{G}$, $\mathcal{G}_1 = [30, 140]$ and $\mathcal{G}_2 = ]140, 300]$. In what follows, we vary the parameter $L$, and we apply the new NIRB algorithm in order to analyze its limits. We will keep the step for each test and observe the effects of the domain truncations on the phenomena. We compare our results with the errors obtained by the classical NIRB algorithm with and without the rectification postprocessing step, as illustrated in Figure 3.7. We consider 19 snapshots with different Reynolds numbers, from 30 to 300 with a step of 15 for $\mathcal{G}$. We take 8 of them for $\mathcal{G}_1$, and 11 for $\mathcal{G}_2$ in function of the Reynold value.

$$\mathcal{G}_1 = [30, 150[ and \mathcal{G}_2 = [150, 300]. \tag{3.16}$$

We consider two new solutions in $\mathcal{G}$: $Re = 52$ and $Re = 233$. Numerical simulations are carried out in FreeFem++, with the Newton algorithm, and classical Taylor-Hood finite elements $\mathbb{P}_2 - \mathbb{P}_1$. The fine mesh size is equal to 0.03, and the coarse one to 0.21. We also consider a finer reference mesh with a size equal to 0.016. The $H^1$ relative errors between the reference solution and the fine solution (respectively the coarse solution) with a piecewise linear interpolator $I$ are given in the following table.

| Reynolds | 52 | 233 |
|---|---|---|
| $\frac{\left\Vert \mathbf{u}_{ref} - I(\mathbf{u}_h) \right\Vert_{H^1(\Omega)}}{\left\Vert \mathbf{u}_{ref} \right\Vert_{H_1(\Omega)}}$ | 0.028 | 0.029 |
| $\frac{\left\Vert \mathbf{u}_{ref} - I(\mathbf{u}_H) \right\Vert_{H^1(\Omega)}}{\left\Vert \mathbf{u}_{ref} \right\Vert_{H_1(\Omega)}}$ | 0.207 | 0.209 |

.

Table 3.4: Projection errors

All the following results are concerned with the $H^1$ norm. We proceed with several numerical results on the new NIRB approach. We detail the presented results:

1. First, we compare the $H^1$ relative error between the reference solution and the fine one (given in table 3.4) to the error between the reference solution and the (classical) NIRB approximation (with and without the rectification postprocessing phase). We recalled

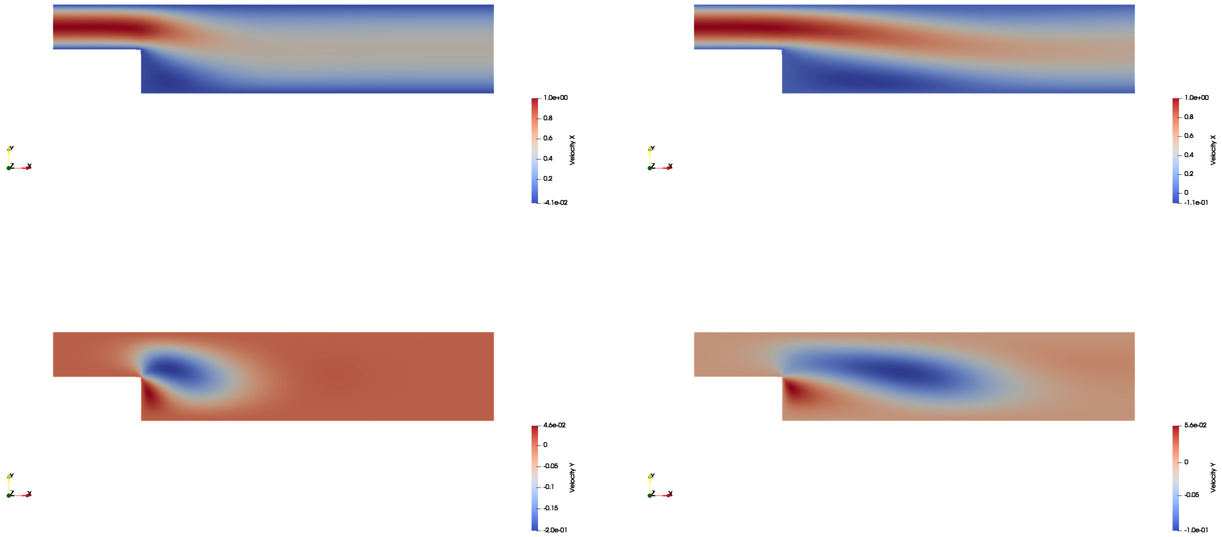


Figure 3.6: Velocities ($u_1$ and $u_2$) for Reynolds=52 (left) and Reynolds=233 (right)

that this error is defined by

$$\frac{\left\| u_{ref}(\mu) - u_{Hh}^{N}(\mu) \right\|_{H^1(\Omega)}}{\left\| u_{ref} \right\|_{H^1(\Omega)}}. \tag{3.17}$$

The purpose of the NIRB method is to retrieve the FEM error, with few modes. This corresponds to Figure 3.7. We observe that the error between $\mathbf{u}_{ref}(\mu)$ and $\mathbf{u}_h(\mu)$ is recovered with the NIRB approximation (with the rectification post-treatment) provided that the new Reynolds parameter belongs to the parameters group used for the RB generation. With the group $\mathcal{G}$, 4 modes are required for both Reynolds. Concerning the parameters subgroups ($\mathcal{G}_1$ and $\mathcal{G}_2$, see (3.16)), when the new parameter is tested with its associated subgroup, 3 modes are sufficient. However, in case of extrapolation, then the NIRB method with post-treatment does not yield optimal results.

2. The results of Figure 3.7 are confronted to those of Figure 3.8 and Figure 3.9. They present relative errors with the new algorithm with different domain truncations. We retrieve a good approximation with this new NIRB algorithm in the sense that with few modes we obtain the same error as the one between $\mathbf{u}_{ref}$ and $\mathbf{u}_h$.
These results are with $N_1 = N_2$ (same number of modes for both bases), using a POD RB (the orthogonalization in $L^2$ is sufficient here (whereas for the classical NIRB, we use $H^1$ and $L^2$ orthogonalization). We observe that without truncation (Figure 3.8), fewer modes are required to recover this error rate compared to the classical NIRB with the rectification post-treatment. Even when the associated subgroup of the new parameter $\mu$ is not used for the RB construction, which corresponds to the extrapolating of the solution out of the parameter space, the test with $\omega = \Omega$ gives more accurate results (with 9 modes,

the error is almost recovered with $Re = 52$ and we retrieve the reference's error with 8 basis functions with $Re = 233$).

We also present two tests with domain truncation in Figure 3.9, with $L = 3$ ($\omega \subset \Omega$), and $L = 1.2$ (and no regularization parameter i.e. $\lambda = 0$). With the truncation corresponding to $L = 3$, only 3 modes with $\mathcal{G}$ ($N_1 = N_2 = 3$) allow us to obtain the same error than with $\mathbf{u}_h(\mu)$ with both Reynolds (whereas 4 are required with the classical NIRB and the rectification post-treatment). Yet, if the new parameter $\mu$ does not belong to the subgroup range ($\mathcal{G}_1$ or $\mathcal{G}_2$) employed during the offline part, on some tests, the error is not recovered and there are some instabilities (case $Re = 52$). The truncation $L = 1.2$ is much more complex. The recirculation area behind the step is cut. As a consequence, the results present more instabilities. However, choosing the associated subgroup for the new parameter to generate the RB helps us retrieve the optimal error. We present also results on the same tests ($L = 1.2$ with POD basis) but with the regularization term $\lambda = 1e^{-10}$ in Figure 3.10 (see equation (3.14)). With this regularization parameter, all the errors remain stable, showing the capability of this new algorithm to retrieve accurate approximations even with highly truncated domains.

3. The new NIRB solution is an approximation of the fine solution $\mathbf{u}_h(\mu)$. Thus, it is also interesting to compare the error between $\mathbf{u}_h(\mu)$ and the new NIRB solution, and to confront these results with the classical NIRB errors. Therefore, we first show the error between the fine solution and the (classical) rectified NIRB approximation in Figure 3.11). Now, the error is given by

$$\frac{\left\| u_h(\mu) - u_{Hh}^N(\mu) \right\|_{H^1(\Omega)}}{\|u_h\|_{H^1(\Omega)}}. \tag{3.18}$$

   In this case, the error measures the RB error and must not reach a plateau, unlike the error (3.17).

4. Finally, we display the errors between the fine solution and the new NIRB approximation with several truncations to analyze the limits of this new algorithm. These results are presented using heatmaps since the error depends on $N_1$ and $N_2$ in Figure 3.12, Figure 3.13, Figure 3.16, Figure 3.17 and Figure 3.18. The runtimes are written in table 3.7.

In what follows, we present results on the error between the fine solution and the new NIRB approximation with different truncations (3.18). We compare the numerical results with the classical NIRB errors in Figure 3.11. Figure 3.11 displays the errors between $\mathbf{u}_h$ and the classical rectified NIRB approximation for both Reynolds 52 and 233.

Figure 3.12 presents results when $\omega$ equals to the whole domain. The relative error in $H^1$ norm is presented through a heatmap regarding the number of modes $N_1$ for the fine RB and $N_2$ for the coarse one. With the new method, the error reaches a lower threshold than with the classical rectified NIRB approximation, although there are some instabilities when the matrix $\mathcal{D}^T\mathcal{D}$ (3.14) is inverted, as we can see with $N_2 = 18$. The condition number of the rectification matrix increases with the number of modes but the regularization parameter allows us to reduce it in some cases 3.14. Figure 3.13 illustrates the relative $H^1$ error with $\omega = \Omega$ and $Re = 52$ as in Figure 3.12 but with $\lambda = 1e^{-10}$.

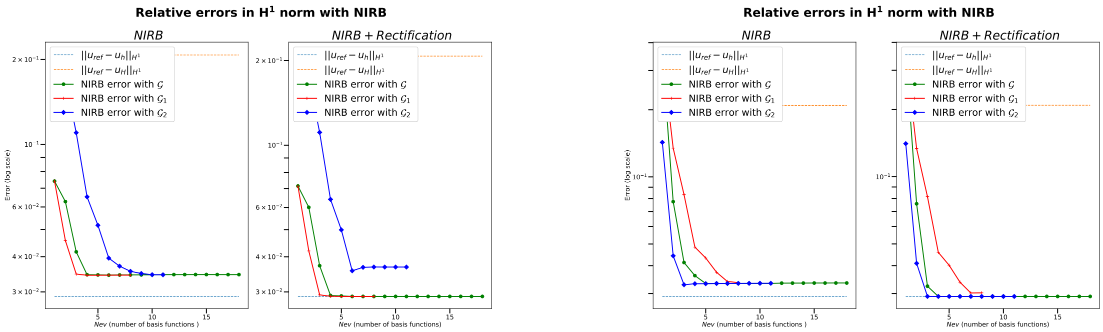


Figure 3.7: classical relative NIRB $H^1$ errors with and without the rectification postprocessing step with Re=52 (left), Re=233 (right)

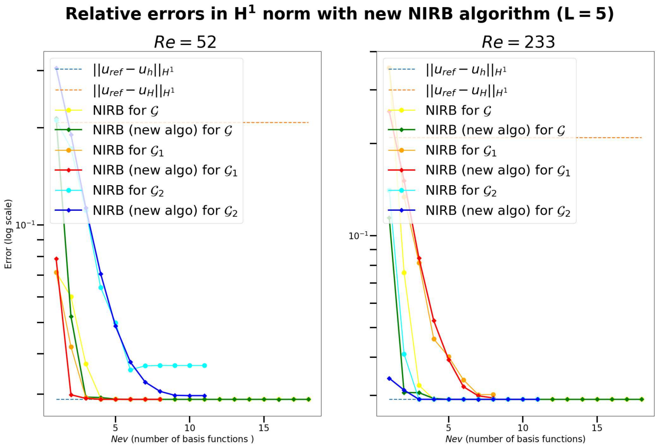


Figure 3.8: Relative $H^1$ FEM errors vs NIRB errors: between reference solution and the NIRB approximations, compared to FEM projection errors (classical algorithm with rectification vs new NIRB algorithm with $\omega = \Omega$), tests with $Re = 52$ (left) and $Re = 233$ (right) ($\lambda = 0$).

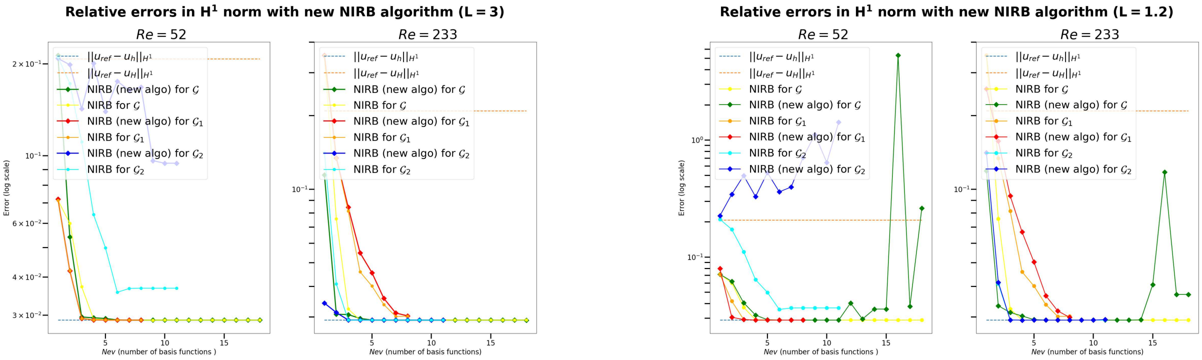


Figure 3.9: Relative $H^1$ FEM errors vs NIRB errors: between reference solution and the NIRB approximations, compared to FEM projection errors (classical algorithm with rectification vs new NIRB process with domain truncations) with $Re = 52$ and $Re = 233$, $\lambda = 0$. Case $L = 3$ (left), case $L = 1.2$ (right)

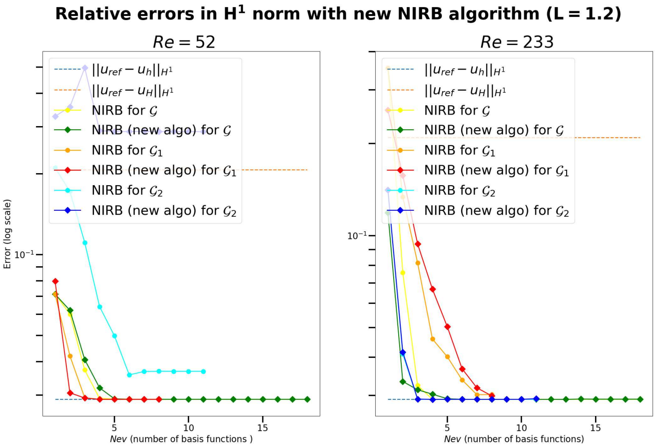


Figure 3.10: Relative $H^1$ FEM errors vs NIRB errors: between reference solution and the NIRB approximations, compared to FEM projection errors (classical algorithm with rectification vs new NIRB process with domain truncations) with $Re = 52$ and $Re = 233$. Case $L = 1.2$, $\lambda = 1e^{-10}$.

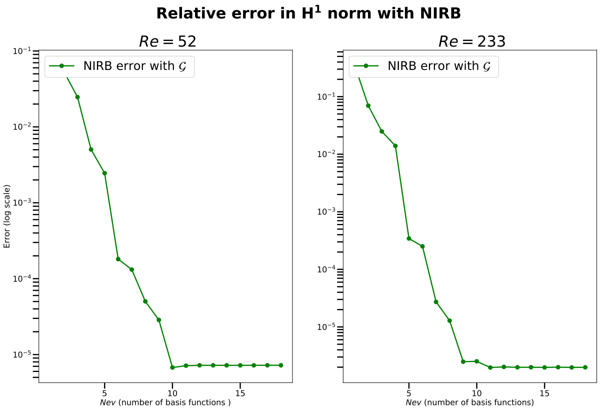


Figure 3.11: $H^1$ relative errors between $u_h$ and classical rectified NIRB solution for Re=52 (left) and Re=233 (right)

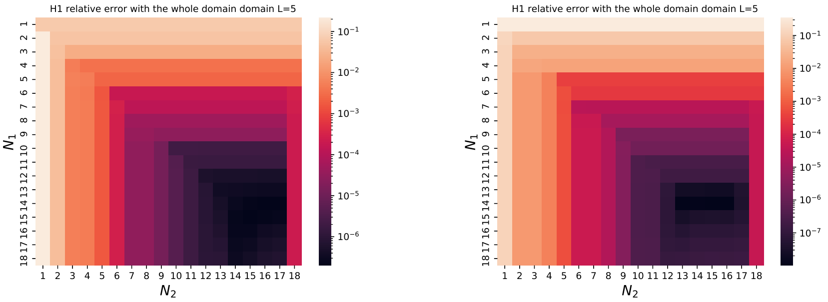


Figure 3.12: relative $H^1$ errors with the new algorithm on $\omega = \Omega$, Re=52 (left), Re=233 (right)

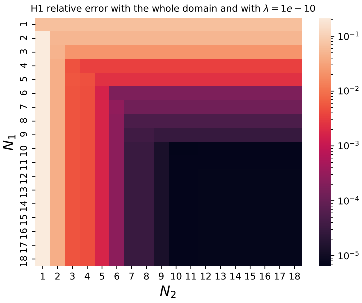


Figure 3.13: NIRB relative $H^1$ errors with the new algorithm on $\omega = \Omega$, Re=52 and $\lambda = 1e^{-10}$

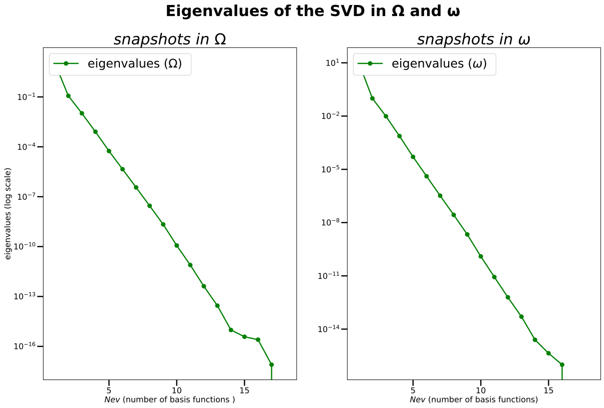


Figure 3.15: Eigenvalues of the correlation matrix

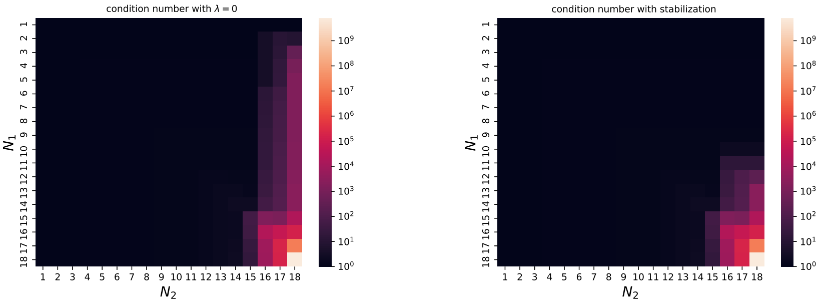


Figure 3.14: Condition number for $Re = 52$ with $\omega = \Omega$ and $\lambda = 0$, $\lambda = 1e^{-10}$

With a SVD approach for the basis generations, the RIC (3.11) gives the smallest numbers of modes $N_1$ and $N_2$ required with $\eta$ a-priori chosen. It is related to the fact that the eigenvalues of the fine and the coarse correlation matrices are decreasing fast enough and converge to zero. Indeed, it indicates that the reduced approximations for both configurations are consistent (see equation (3.15)). As a result, this algorithm is very efficient since it recovers the required number of modes on the coarse mesh as well as on the fine mesh.

**The following figures (3.16,3.17, and 3.18) show the results with a truncated domain $\omega$ with different values of $L$ for both Reynolds 52 and 233.** The coarse solutions for $Re = 52$ on the truncated subdomains with $L = 1.2$ and $L = 3$ are presented in Figure 3.19.

We end this section with the NIRB errors (for $u_1$ and $u_2$) and the new NIRB approximation with $L = 3$ and $Re = 52$, $N_1 = 7$, $N_2 = 8$ (Figure 3.20).

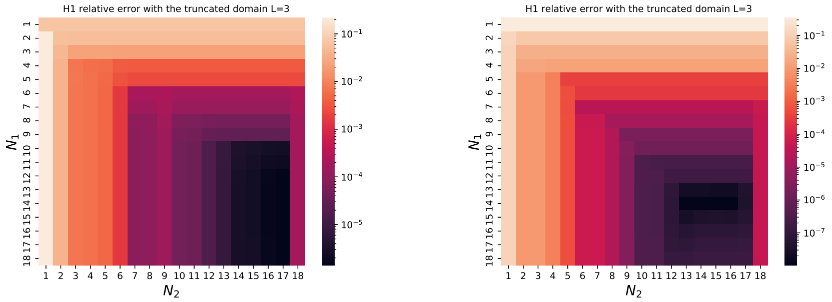


Figure 3.16: $H_1$ relative errors for Re=52 (left) and Re=233 (right) with Greedy algorithm ($\lambda = 0$) on $\mathcal{G}$ with $L = 3$

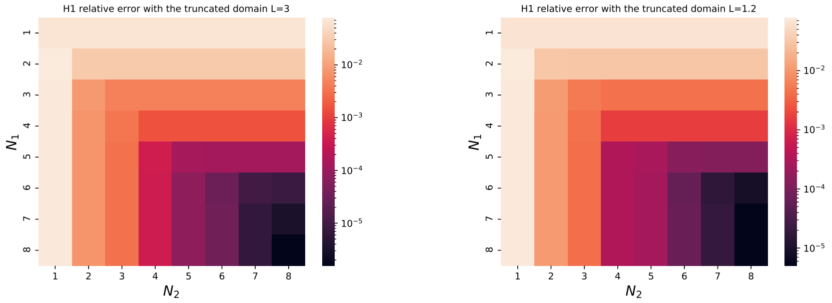


Figure 3.17: $H_1$ relative error for Re=52: Greedy algorithm on $\mathcal{G}_1$ with $L = 3$ (left), and with $L = 1.2$ (right) ($\lambda = 0$)

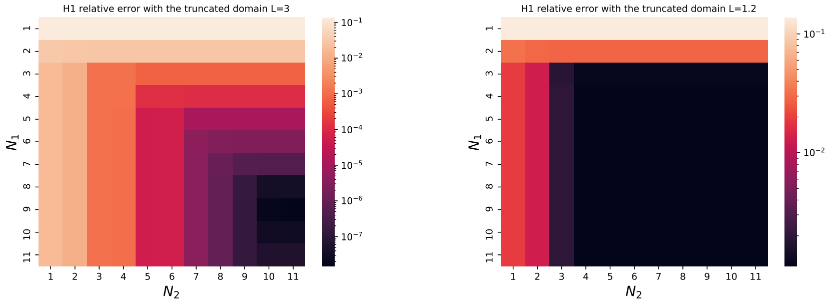


Figure 3.18: $H_1$ relative error for Re=233: Greedy algorithm ($\lambda = 0$) on $\mathcal{G}_2$ with $L = 3$ (left), and with $L = 1.2$ ($\lambda = 1e - 10$) (right)

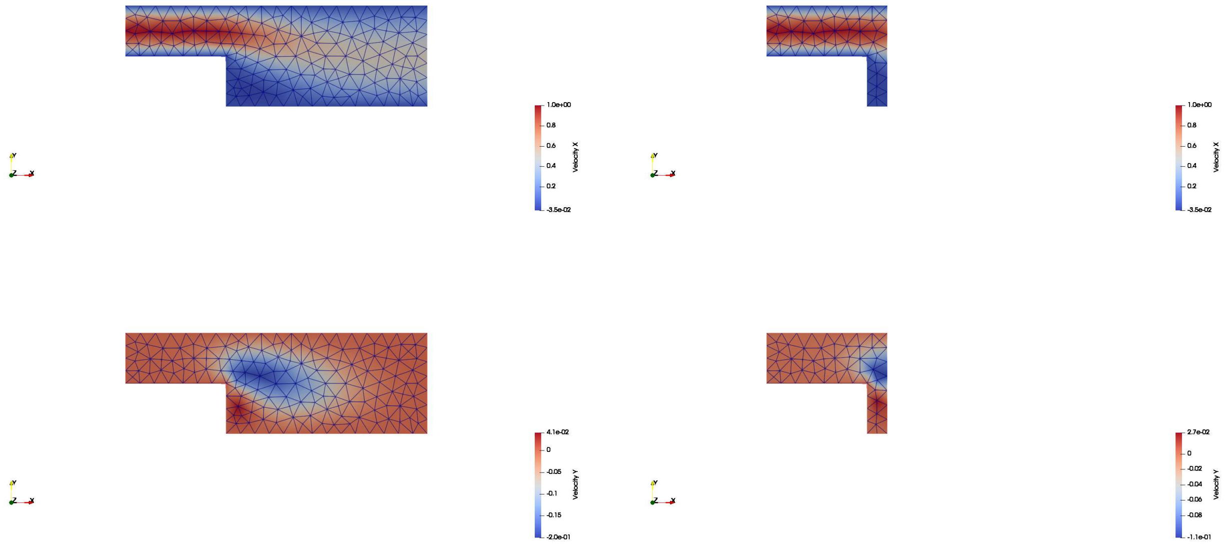


Figure 3.19: Coarse solution Re=52 on the truncated domain with $L = 3$ (left), and with $L = 1.2$ (right)

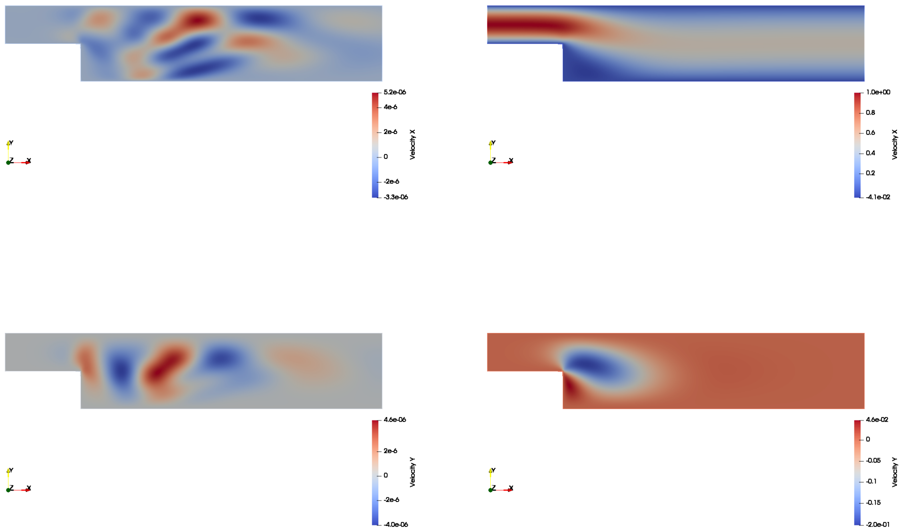


Figure 3.20: $N_1 = 7$ and $N_2 = 8$, $L = 3$, $Re = 52$ error (left) and approximation (right)

**Time execution (min,sec)**

| FEM high fidelity solver | FEM coarse solution |
|---|---|
| 00:43 | 00:01 |

Table 3.5: FEM runtimes

| NIRB Offline | classical NIRB online |
|---|---|
| 14:30 | 00:26 |

Table 3.6: NIRB runtimes ($N = 18$)

| | new NIRB offline (coarse RB generation) | new NIRB online |
|---|---|---|
| $\omega = \Omega$ | 00:12 | 00:10 |
| $L = 3$ | 00:12 | 00:09 |
| $L = 1.2$ | 00:10 | 00:09 |

Table 3.7: Offline (coarse RB) and online runtimes ($N_1 = N_2 = 18$)

### 3.3.3 Conclusion

All these results show that this new algorithm may enhance the NIRB method both in terms of runtime and accuracy (section 3.3.2, Figure 3.8, Figure 3.9). The execution time is reduced thanks to the truncations on the coarse mesh. As shown by our tests, the error between the reference solution and the fine one is recovered with this approximation by means of several truncations, sometimes with fewer modes compared to the classical NIRB. Some cases (when the domain is highly truncated) present instabilities but they are stabilized using the Tikhonov regularization parameter $\lambda = 1e^{-10}$ (see equation (3.14), and Figure 3.10). This regularization parameter is useful when the condition number of the rectification matrix is too high. The new approximation may be more accurate than the one obtained from the classical rectified NIRB two-grid method. Indeed, when considering the whole domain $\omega = \Omega$, with the case $Re = 52$, the error is of order $1e^{-6}$ with $N_1 = N_2 = 12$ (Figure 3.12), compared to $1e^{-2}$ with the NIRB classical method and $1e^{-5}$ with $N = 10$ with the rectification posttreatment [30] (Figure 3.7). For $Re = 233$, we obtain an error of order $1e^{-7}$ with the treshold $N_1 = N_2 = 13$ compared to $1e^{-2}$ with the NIRB classical method and $1e^{-5}$ with $N = 7$ with the rectification postprocessing phase. Furthermore, this new algorithm reduces the online computational costs, as shown in table 3.3.2. We already explained that degrees of freedom were reduced. With the NIRB method, the coarse solution is interpolated onto the fine mesh and $L^2$-projected onto the reduced space $X_h^N$. Note that the interpolating matrix can be computed during the offline stage. Here with this new algorithm, there is no need of interpolation, and in particular, the $L^2$-inner product between the coarse solution and the coarse basis is performed on the coarse mesh.

With a truncated domain $\omega \subset \Omega$, we present results with $L = 1.2$ and $L = 3$. With $L = 3$, the errors (3.18) remain smaller than with the classical NIRB with rectification, with both Reynolds (Figure 3.16) and still quite low when $L = 1.2$ (Figure 3.17 and Figure 3.18). These results highlight that for a new Reynolds in $\mathcal{G}$, any physically acceptable truncation of the domain can be used with this new method.

However, this algorithm is not appropriate if the domain is no longer coherent (for instance, in our model problem, if the step is removed and only a Poiseuille flow remains). Thus, a prior knowledge of the whole domain is necessary, and the cut must be done properly by the expert users. If the truncation is correctly chosen, then we may retrieve the energy-error between the reference solution and the fine one, and the costs are reduced compared to the NIRB method. In table 3.3.2, the time reduction is not obvious since the FEM coarse solution on $\omega = \Omega$ is already computed in less than one second. Yet, we believe that this new method can be succesfully applied to a wide variety of problems, with complex domains having large degrees of freedom, and therefore the computational times will be significantly reduced with a truncated domain.

# Chapter 4

# An industrial application: Offshore wind farm simulations

## 4.1 Context

### 4.1.1 Offshore wind farms

This chapter is a collaboration with EDF (Electricité De France). The purpose of this chapter is to apply the two-grid method to a computational fluid dynamics (CFD) problem, which is the simulation of offshore wind farms. Unlike onshore wind farms, offshore wind farms are located at sea, in order to enhance the wind energy production. As explained in the introduction 0.3.4, they are more efficient but they imply higher costs of installation and maintenance. Thus, a precise analysis of the wind turbines position within the wind farm is necessary before its setup. The position of the wind turbines is determined as a function of the local climatology. A great number of wind simulations within a wind farm are required for several turbines arrangements. In this chapter, we apply the two-grid method on wind farm simulations in order to reduce their computational cost. The goal is not to find the best turbines position but to explain and exhibit by means of numerical results how the two-grid method can efficiently be used on such complex situations.

**Remark 41.** *In this chapter, the figures side by side with similar results will share the same scale, and thus, the ordinate of the right-hand side figure will not be displayed (see for instance Figure 4.14).*

In what follows, the varying parameters are the reference input velocity magnitude, denoted $u_{ref}$, and its incidence angle $\theta$.

In an industrial context, these simulations aim at predicting the power generated by the turbines as a function of the wind. The physics is simplified by the use of an actuator disc model [11, 102, 128], but these simulations are still very costly in time. A squared wind farm with actuator discs is represented in Figure 4.1. Turbines affect each other by the wake effects. Indeed, the wake effects of a turbine have two main effects on a backstream turbine:

- a reduction of the wind velocity, which reduces the energy production.
- and an increase of turbulence behind the rotor, which makes the production more unpredictable.

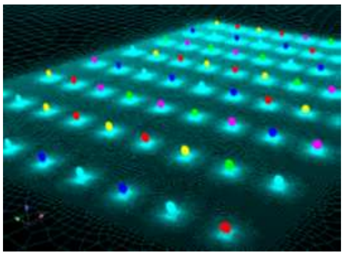


Figure 4.1: Wind farm with actuator discs, image from EDF.

This is why the positions of the wind turbines must be thoroughly studied. The wind turbine power is calculated as a function of a power coefficient $C_P$ and an upstream velocity, denoted $\mathbf{u}_*$ [36, 88], located at some probes (see Figure 6 in the introduction). The turbine power is defined by

$$P = c_P \frac{1}{2} \rho A \mathbf{u}_*^3, \tag{4.1}$$

where $A$ is the area of the turbine actuator disc and $\rho$ the wind density. In what follows, the maximal generated power for one wind turbine corresponds to 6 MW. In what follows, we study the wind behavior around the turbines, and we effectively apply the two-grid method to reproduce it with coarser 2D and 3D simulations.

### 4.1.2 Code_Saturne and RANS equations

The wind simulations of a wind farm employ the Reynolds-Averaged Navier-Stokes (RANS) equations. The RANS method relies on the decomposition of the variables into their statistical average and their fluctuations. The time averaging of the mass and momentum equations leads to a system for which various models provide different levels of closure. We will consider the closure called $k - \varepsilon$, where $k$ represents the turbulent kinetic energy and $\varepsilon$ the dissipation rate of the turbulent energy. We denote by

- $\mathbf{u} = (u_1, u_2, u_3)$ the wind velocity,
- $p$, the wind pressure,
- $\rho$ its density,
- $\mu$ its dynamic viscosity.

The mass and momentum equations write

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0, \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \rho \left( \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + F, \end{cases}$$

where $F$ represents the additional momentum source terms which will contain the forces generated by the wind turbine. Let us consider the following decomposition: $\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}'$, where $\overline{\mathbf{u}} = (u_1, u_2, u_3)$ is the averaged wind velocity, and $\mathbf{u}'$ is its fluctuation. We make the same decomposition for its pressure $p = \overline{p} + p'$. The kinetic viscosity is denoted $\nu = \frac{\mu}{\rho}$. With the

Einstein convention notation, the RANS equations, derived from (4.2) (see Appendix E.2 for details), read:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho \overline{u}_i)}{\partial x_i} & = 0, \\ \rho \frac{\partial \overline{u}_i}{\partial t} + \rho \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} & = -\frac{\partial \overline{p}}{\partial x_i} + \rho \frac{\partial}{\partial x_j}(\mu(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}) - \overline{u'_i u'_j}) + \rho \overline{F}, \end{cases} \tag{4.2}$$

which involves the Reynolds tensor $\overline{u'_i u'_j}$, and $\overline{F}$ represents the additional averaged momentum source terms which contains the forces generated by the wind turbine.

Only the averaged velocity and the averaged pressure are solved with code_saturne [8]. We recall that code_saturne is a free FV software for fluid mechanics developed by EDF. The standard $k - \varepsilon$ from the Eddy Viscosity Models (EVM) is employed to close the system (4.2) where the velocity correlations are modeled in 3D with the Boussinesq formulation:

$$-\rho \overline{u'_i u'_j} = \mu_t (\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}) - \frac{2}{3} \rho k \delta_{ij},$$

where $\mu_t$ is the turbulent viscosity, defined as

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon},$$

and $C_\mu$ is a constant (see table E.1). Finally, the $k - \varepsilon$ closure adds two transport equations to the system 4.2, which $k$ and $\varepsilon$ respectively solve [89, 32]. We recall both equations in appendix E.2 and the constants used in code_saturne. The other parameters used for the climatological setup conditions of this study are given in the following table 4.1,

| | | |
|---|---|---|
| Sea pressure [Pa] | $P_{sea}$ | 101325 |
| Sea altitude [m] | $z_0$ | $2e^{-4}$ |
| Hub height [m] | $z_{ref}$ | 100 |
| Karman constant | $K$ | 0.42 |
| Specific heat at constant pressure [J/K/Kg] | $C_p$ | 1005 |
| Gravity [m/$s^2$] | $g$ | 9.81 |
| Rayleigh number [J/Kg/K] | $Ra$ | 287 |
| Reference pressure [J/Kg/K] | $P_{ref}$ | $1e^5$ |
| Initial temperature (°C) | $T_0$ | 20 |

Table 4.1: Climatological parameters

The fluid parameters (the initial fluid density $\rho_0$ and its viscosity $\mu$) can be changed with code_saturne GUI. In this study, the wind parameters are summarized in table below 4.2.

| $\rho_0 [kg/m^3]$ | $\mu$ |
|---|---|
| 1.17862 | $1.83e^{-5}$ |

Table 4.2: Wind parameters

We summarize the main steps of the wind simulations:

1. The first mesh representing the wind farm is created with Salome. The mesh is unstructured, fixed and adapted to the domain with a refinement around the turbines. The

rotors are not geometrically modeled but are represented by actuator discs as in Figure 4.2 (image from the article [102]) and additional source terms (explicit and implicit formulations) in the RANS equations [98, 128, 102]. There are no fluid structure interaction since the rotors are not represented geometrically. The rotor representes the space where the source terms take effects.

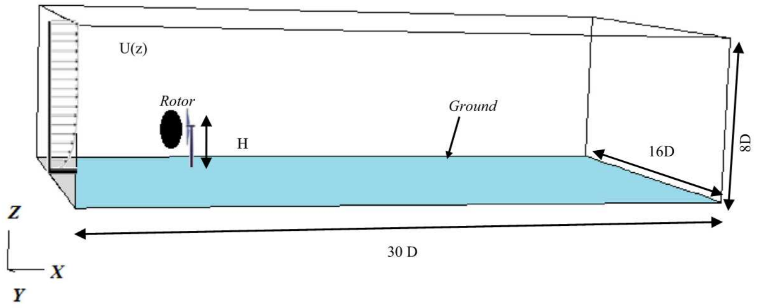


Figure 4.2: Rotor in the spatial domain

2. Then, climatological conditions are configured (with Python) in order to initialize the velocities, the temperature and other variables for code_saturne simulations. We use the parameters summarized in table 4.1. The initial wind profile and the air temperature are generated as a function of the altitude $z$ and $u_{ref}$. In other words, $\mathbf{u}_0 = \mathbf{u}_0(z; u_{ref})$. The initial velocities are first generated at the boundary and lifted in whole domain. The magnitude $u_{ref}$ corresponds to the initial velocity magnitude at the hub height. A reduction of the wind magnitude is observed near the sea surface, as we can see for instance in Figure 4.3.

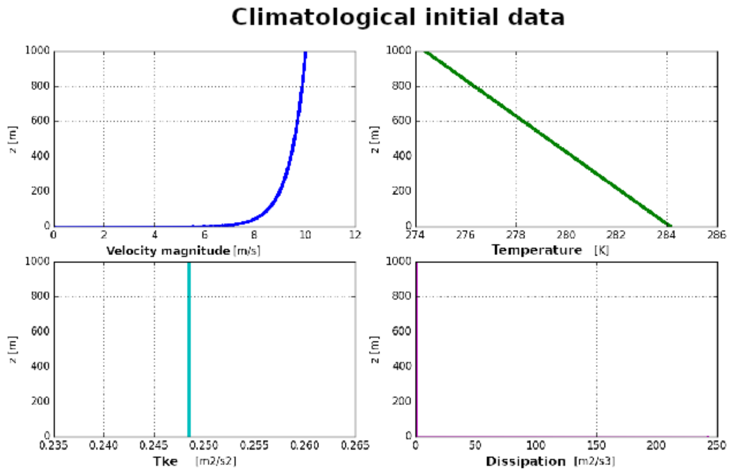


Figure 4.3: magnitude velocity, temperature, initial turbulent kinetic energy $k$ and dissipation $\varepsilon$ as function of the altitude $z$ with $u_{ref} = 8.5$, $\theta = 0$

3. Finally, for each reference velocity magnitude $u_{ref}$ and each direction $\theta$, the RANS simulations are launched with code_saturne, which employs the SIMPLEC algorithm for the velocity-pressure coupling [81]. The time discretization is achieved through a fractional step scheme [20, 33, 8]. The software outputs give information on several variables and their values on the mass center of cells, but we will focus on the velocity values after several time step, mainly until reaching a quasi-stationary state.

## 4.2 The two-grid method applied to the wind farm simulations

As explained in the introduction, the NIRB two-grid method may be very efficient in reducing the complexity simulations which are very costly in time. In what follows, we present numerical results on several wind farms studies. They are very complex because of their large number of degrees of freedom. These industrial problems are also very innovative, since they are no longer based on the theoretical studies of the previous chapters 2 (Theorem 16 does not hold anymore). All the numerical results, with the NIRB two-grid method (see algorithms 3-4) and the new NIRB tools explained in chapter 3 show their great potential.

In all the following studies, the velocity values in an area located upstream the turbines (which correspond to probes) and around the rotors are considered as the quantities of interest and they are analyzed for several input given velocities (reference magnitude $u_{ref}$ and direction $\theta$). For these simulations, the NIRB strategy is applied to approximate a quasi-stationary state. Thus, we retrieve the quasi-stationary solution after several time iterations.

1. We first analyze the wind behavior around a 2D wind turbine. Thus, the 3D turbulence effects are not taken into account. This first step allowed us to analyze $\mathbf{u}^*$ and to choose the general parameters for the two-grid method (such as the meshes size and the snapshots parameters).

2. Then, we consider two and three turbines in line, still in two dimensions. The purpose is to experiment the two-grid method in order to use it in an efficient way. As said in

the introduction, wind turbines affect each other due to wake effects. Thus, we tested a way of optimizing the NIRB algorithm by representing one turbine with the others. By "representing one turbine", we mean approximating the wind around it. We tested this new approach with the last turbine represented by the first two. More precisely, we are looking at zooms around each turbine. Then, we project all of them on a reference mesh which may be the one of the first turbine. Then, we use a coarse solution around the last turbine and fine solutions around the first two turbines for the RB generation (projected on the reference mesh).
With two dimensions, the results show that the turbines are relatively undisturbed by each other. As a result, the two-grid method can be very efficient in these cases. The rectification post-treatment (explained in the introduction 0.2.2) enables us to retrieve an accurate approximation as if we had used a fine FV solution.

3. Since the wake effects of a wind turbine were not sufficient for disturbing other turbines, we complexified the data with a 3-dimensional case and three in line wind turbines. We present several numerical results on the two-grid algorithm with the previous approach.

4. Finally, we consider a squared wind farm with nine wind turbines. We present several numerical results on the area upstream the wind turbines which reprensents the probe in (4.1) and inside the rotors.

5. One ongoing project is concerned with a $5 \times 5$ wind farm approximated with a $3 \times 3$ wind farm. The $5 \times 5$ farm will be simulated only once with a coarse mesh. Then, we will generate snapshots for several parameters and we will focus on the velocity around the rotors. Finally, each line of turbines of the $5 \times 5$ farm will be approximated with one or several lines of the $3 \times 3$ wind farm.

**Main idea.** The main purpose is to study the two-grid method capacity for representing one particular turbine with the help of several others. This is a challenging problem since the degrees of freedom are very large, and the number of time iterations is not always sufficient for obtaining the coarse solutions convergence.

The following sections present all the 2D and 3D numerical results.

### 4.2.1 2D wind turbine results

**Parameters configuration.** We first tested the NIRB two-grid method on a simple problem, consisting in a single wind turbine in two dimensions. This first case was useful for determining the quantities of interest and all the general parameters. Since code_saturne only works with 3D studies, the 2D plane is extrapolated in order to get one cell in the direction of the third axis. Then, a plane of the solution is retrieved to apply the two-grid method. In two and three dimensions, the turbines are represented by actuator discs (replaced by rectangles in 2D), and the wind perturbation around the turbine is modeled by an additional source term in the movement equation (4.2). It is decomposed in implicit and explicit parts representing the thrust

$$T = c_T \frac{1}{2} \rho A \mathbf{u}_*^2,$$

where $C_T$ is a thrust coefficient.

- Concerning the mesh parameters, in two dimensions, we employed the following mesh parameters 4.3

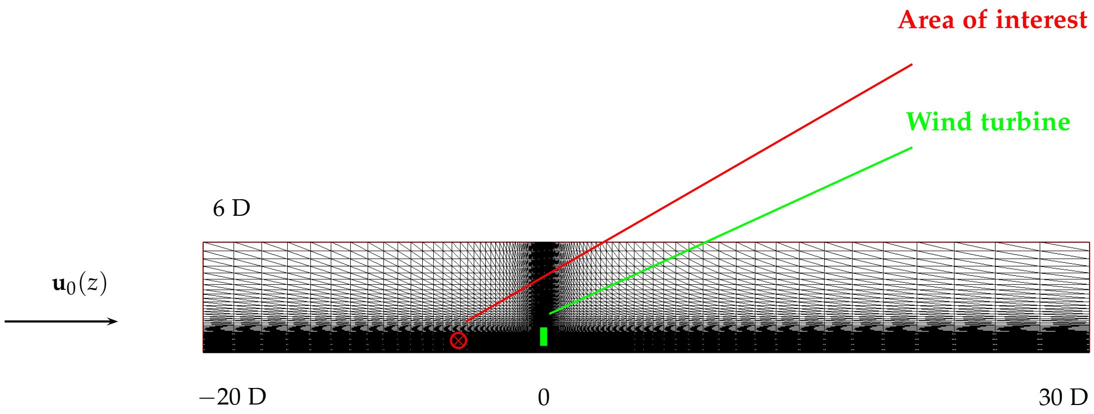


Figure 4.4: Mesh for one wind turbine

| | |
|---|---|
| Rotor diameter (D) | 150m |
| Hub height | 95.6 m |
| Number of cells | 6500 |
| fine mesh size $h$ | 5 m |
| coarse mesh size $H$ | 10 m |
| reference mesh size $h_{ref}$ | 2 m |

Table 4.3: Mesh parameters

The mesh is unstructured, adapted and finer around the turbine and the probe, as in Figure 4.4. The borders are far enough to simulate an infinite domain.

- Concerning the time iterations of the RANS equations, 500 iterations were sufficient to observe that the velocity has reached a quasi-stationary state (This information can be retrieved in the output file "listing" generated by code_saturne in the section "information on convergence"). For instance, the residual norm for the velocity magnitude in our case was about $1e^{-9}$ with this number of time iterations and the mesh parameters summarized in table 4.3. It is important to fully observe the fine solution convergence to apply the NIRB method. In the 3-dimensional case, we tested the capacity of the two-grid method to recover a good approximation even when the coarse solution had not completely converged.
- In this first 2-dimensional case, the wind direction $\theta$ is not changing, and the varying parameter is the reference velocity magnitude $u_{ref}$, used in the climatological profile generation. Our quantities of interest are the wind values after several time iterations at the probe located upstream the turbine at a distance $2.5D$ from the hub and the wind values in the vicinity of the actuator disc. Indeed, to predict the generated power (4.1), the velocity upstream the wind turbine denoted $\mathbf{u}_*$ is simulated. Moreover, it is also interesting to compute the wind velocity inside the actuator disc and to observe its effect

on the wind. For the snapshots, we chose a reference velocity magnitude between $3m.s^{-1}$ and $16m.s^{-1}$. These values are generally employed in the industrial context. Indeed, $3m.s^{-1}$ corresponds to the minimum velocity required by a turbine to produce energy. From equation (4.1), we can see that a turbine with a wind of magnitude $16m.s^{-1}$ in the probe is 64 times more productive than with a wind of magnitude $4m.s^{-1}$.

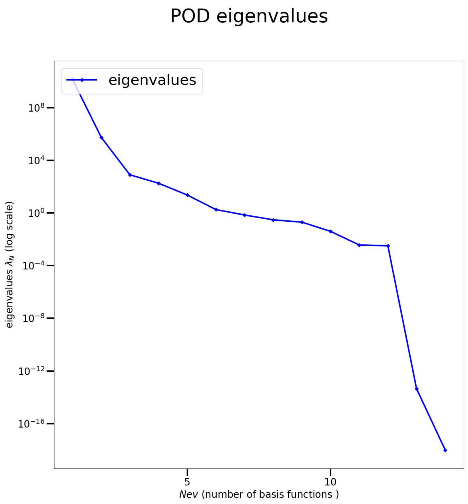
Before presenting the NIRB results, we summarize the main steps involved in the application of the two-grid method:

1. Choose the mesh parameters and the time iterations number that ensure the convergence of the fine solutions.
2. Select the training set for the snapshots parameters.
3. Carry out a SVD on the snapshots to observe the eigenvalues decay (see section 1.2 for explanations).
4. If the eigenvalues correctly decrease, identify the coarse mesh size $H$ for applying the NIRB method. We emphasize the fact that the size $H$ is not necessarily chosen such that $H^2 = h$.
5. Apply the NIRB method with a new parameter inside the range of the snapshots parameters.
6. Save the NIRB approximation on the fine mesh.

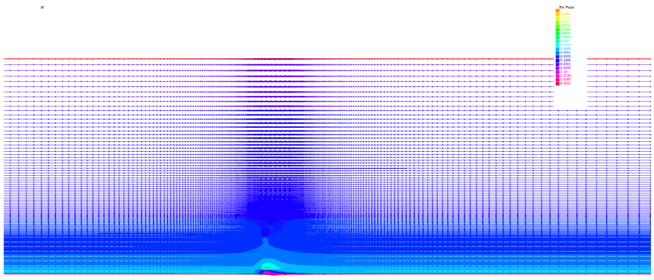
**Results on one turbine.** We first tested the two-grid algorithm with the FreeFem++ solver. We retrieved code_saturne solutions in the VTK format. We recovered the values on the nodes and recreated a Delaunay triangulation with Paraview. Thus, as in section 2.2.1, we compute the NIRB approximation with $\widetilde{I}_h u_{Hh}^N$, where $\widetilde{I}_h$ is the Lagrangian interpolated operator on the new mesh. FreeFem++ can read VTK solutions with the "iovtk" library.

- Our first result is concerned with the single turbine on the whole domain $\Omega$. We used 14 snapshots, and we present the NIRB approximation obtained with one mode in Figure 4.5b. On the whole domain, the POD eigenvalues are fast decreasing and the RIC (1.19) is equal to $I(1) = 0.99995$ with one function in the basis (Figure 4.5a). Thus, we recover 99.995% of the energy information with one mode. This is coherent since the wind perturbation created by the turbine is very small in two dimensions compared to the whole domain. This is another reason why we focus in what follows on subdomains around the turbines and in the probes. The rest of the domain can easily be approximated with a coarse solution. The Figure 4.5b displays the NIRB approximation on the whole domain.
- Figure 4.6a diplays the eigenvalues with 66 snapshots on a zoom in the vicinity of the turbine (Figure 4.6b). We used a reference mesh in addition to the coarse and fine meshes to compute the NIRB errors.

  Concerning the eigenvalues decay, with the zoom on the turbine, $I(1) = 99.86$ %, and $I(2) = 99.9998$ % (Figure 4.6a). It shows that two modes are sufficient to approximate the solutions. Figure 4.7 illustrates the NIRB results with a POD basis ($L^2$ orthogonalization). We compare the error between a solution on the reference mesh denoted $\mathbf{u}_r$ and the NIRB

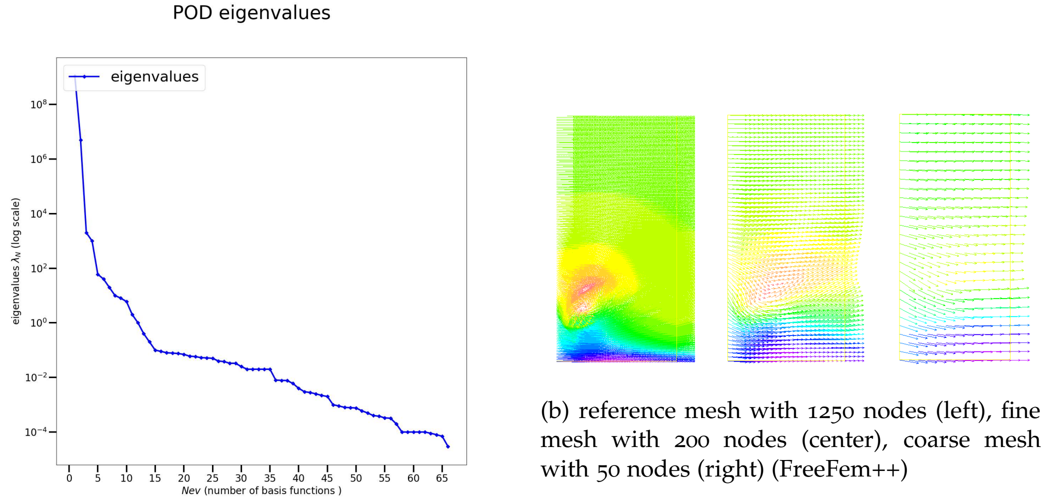


(a) POD Eigenvalues, Ntrain=14, $u_{ref} = 3:1:16$



(b) NIRB approximation with $u_{ref} = 16$ ($N = 1$) on the whole domain

Figure 4.5: Results on the whole domain



(b) reference mesh with 1250 nodes (left), fine mesh with 200 nodes (center), coarse mesh with 50 nodes (right) (FreeFem++)

(a) POD Eigenvalues, Ntrain=14, Ntrain=66, $u_{ref} = 3 : 0.2 : 16$

Figure 4.6: Zoom on the turbine

approximation to the fine and coarse errors (velocities respectively denoted $\mathbf{u}_h$ and $\mathbf{u}_H$). For a new parameter $u_{ref}$, the relative error in the $H^1$ norm is defined as

$$\frac{\left\|\mathbf{u}_r(u_{ref}) - \mathbf{u}_{Hh}^N(u_{ref})\right\|_{H^1}}{\left\|\mathbf{u}_r(u_{ref})\right\|_{H^1}}. \tag{4.3}$$

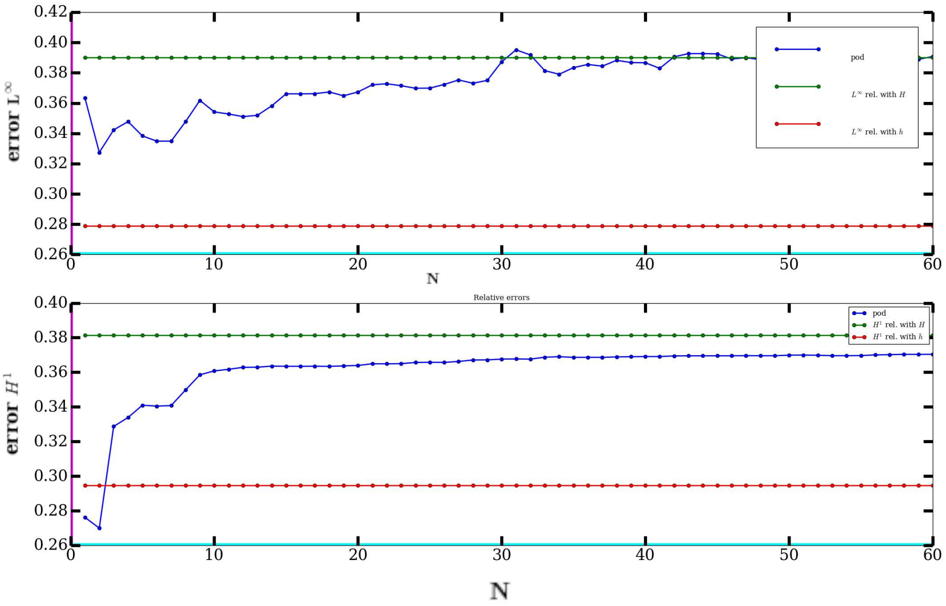
The $N$-dependence of the NIRB approximation is clearly highlighted by these results, as explained in the introduction (remark 4) and in the chapter 2 (theorem (16)). For both tests, two modes are sufficient to recover the fine FV error in $L^\infty$ and $H^1$ norms, and with the $H^1$ norm the error is even smaller.

**Two and three wind turbines in line.**

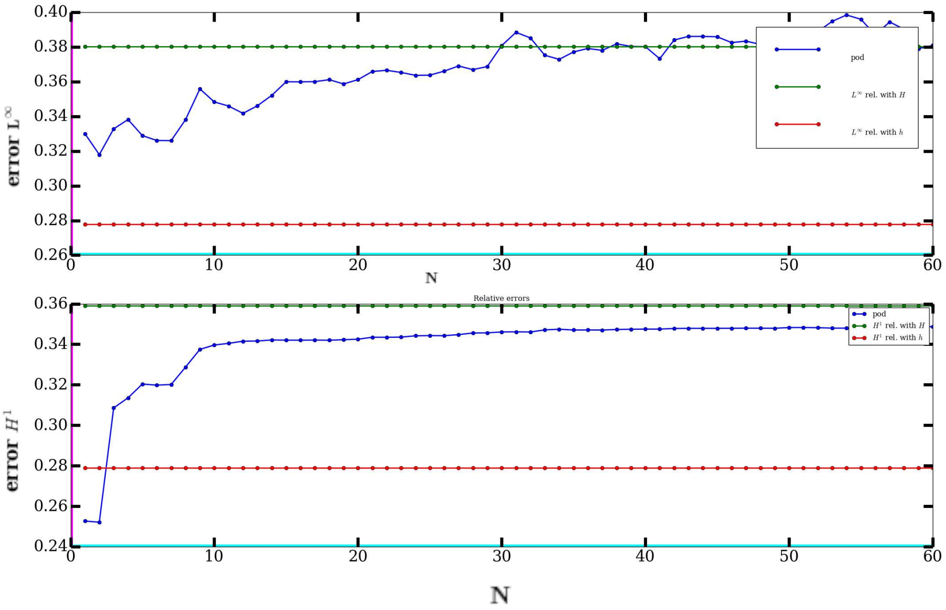
- We first observed how turbines were affected by others on 2-dimensional problems. Figure 4.8 illustrates the tests on two turbines in line. We first separated them with a distance equal to $7D$ (where $D$ is the characteristic length corresponding to the rotor diameter), but they were too distant to affect each other. Therefore, in the case of three turbines in line, we put them closer at a distance equal to $3D$. Still, two modes were still sufficient to recover optimal results on the probe or in the vicinity of the rotors.

  We observed the same results whether for the first turbine or for the second one. In Figure 4.9, we present the $H^1$ and $L^2$ errors between the fine solution $\mathbf{u}_h(u_{ref})$ and the NIRB approximation without rectification $\mathbf{u}_{Hh}^N(u_{ref})$ (left) and the errors of the true projections (3.7) (right) on the first rotor. The best NIRB approximation without the rectification is computed with two modes. Then the error increases.

  On the probe (which corresponds to the area located forward the second turbine), we can see in Figure 4.10 that two modes are also sufficient to correctly represent the velocity in



(a) $u_{ref} = 7$



(b) $u_{ref} = 10.5$

Figure 4.7: Relative $L^\infty$ and $H^1$ errors between the reference solution $\mathbf{u}_r$ and the NIRB approximation (without rectification post-treatment) on the zoom with Ntrain=66 as a function of $N$, compared to the fine and coarse FV errors, showing the $N$-dependance of the error (see remark 4).

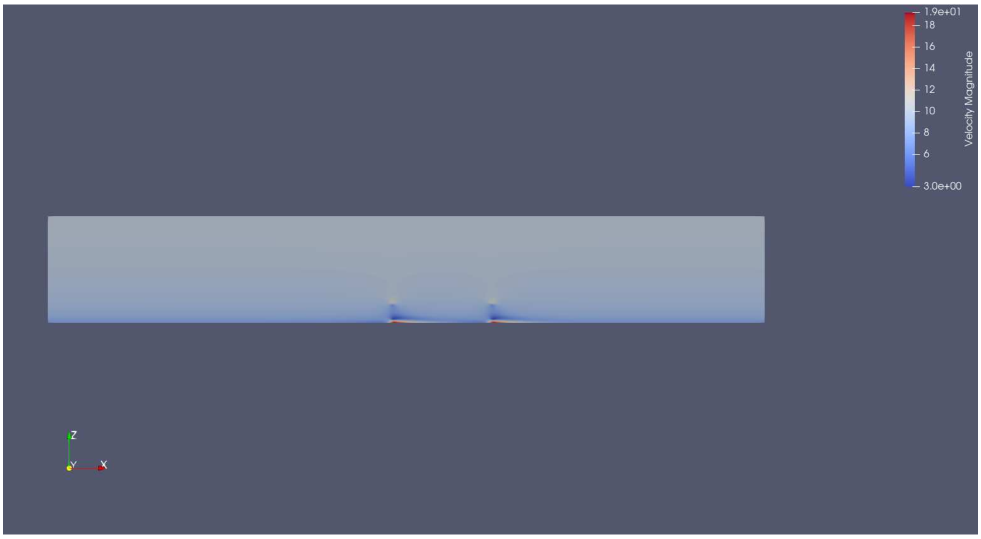


Figure 4.8: Visualization of two turbines in line with $u_{ref} = 10.5$ (fine solution)

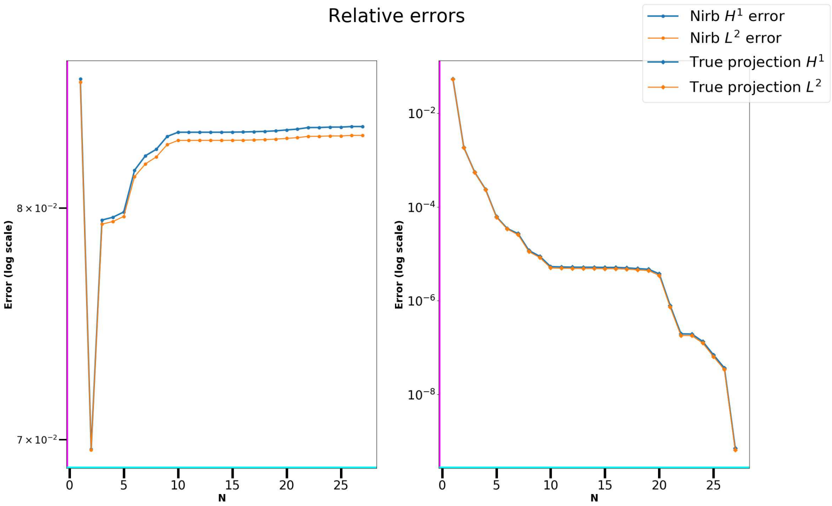


Figure 4.9: NIRB errors between the fine solution $\mathbf{u}_h$ and the NIRB approximation $\mathbf{u}_{Hh}^N$ around the first rotor with $u_{ref} = 10.5$ and the true projection $u_{hh}^N$

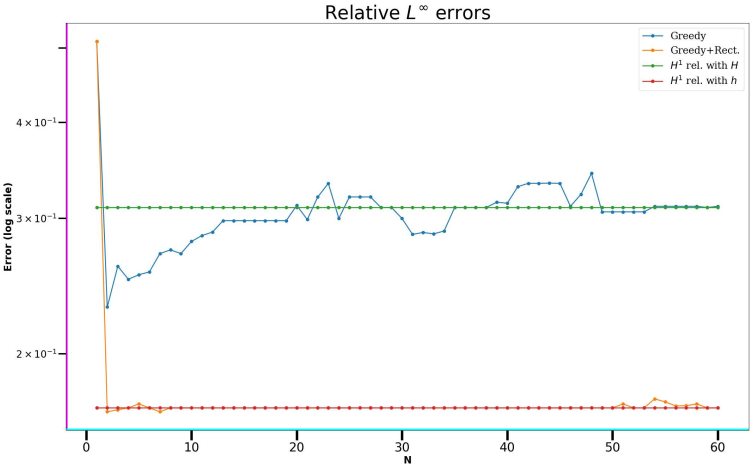


Figure 4.10: $L^\infty$ relative errors between the reference solution $\mathbf{u}_r$ and the NIRB approximation (with and without rectification post-treatment) as a function of the number of modes $N$, Ntrain=66, $u_{ref} = 10.5$, on the probe (which corresponds to the area located forward the second turbine

this area, thanks to the rectification post-treatment 0.2.2. We computed the relative $L^\infty$ error between the reference solution $\mathbf{u}_r$ and the NIRB approximation (with and without rectification) with $u_{ref} = 10.5$, and we compared these errors with the ones given by the fine and the coarse FV solutions.

- The results on three turbines in line are quite similar. We display errors with the rectification post-treatment and with the constrained NIRB, and we present the first results on approximating with NIRB one turbine with a RB generated by other ones. As already said, with a 2-dimensional problem, the wake effects are not sufficient to disturbed other backstream turbines. In other words, the perturbation of the first turbine is similar to the one of the second turbine, .... As a result, with three turbines in line, the wind around a rotor can be very well approached with the NIRB method by the wind around a previous turbine. Here, we present results on the wind around the third turbine approximated with a RB generated by snapshots on the second turbine. We introduce the following notations

  - $\mathbf{u}^2$ represents the wind around the second turbine,
  - and $\mathbf{u}^3$ stands for the wind around the third one.

  Since we represent the third turbine with a RB constructed from the second one, we also need the RB functions generated by the snapshots $(\mathbf{u}_i^2)_{i=1,\dots,Ntrain}$. They are denoted $(\Phi_i^{h,2})_{i=1,\dots,N}$.
  We observe that

  - we easily recover the error of the fine solution around each rotor using the rectification posttreatment (even if there are some instabilities when the regularization parameter $\lambda = 0$ (13)) (Figure 4.11a),

- and we may compute an accurate approximation of the third rotor area with a RB generated by the second rotor part as shown in Figure 4.11b which illustrates the error defined by the NIRB constrained version (see algorithms 5-6)

$$\text{for } u_{ref} \in \mathcal{G}, \left\| \mathbf{u}_{\mathbf{r}}^{3}(u_{ref})(\mathbf{x}) - \sum_{i=1}^{N} \alpha_i(u_{ref}) \Phi_i^{h,2}(\mathbf{x}) \right\|_{H^1(R)}, \tag{4.4}$$

  where $R$ is a region around the rotor, $u_{ref}$ is the parameter, and $\mathbf{u}_{\mathbf{r}}^{3}(u_{ref})$ its exact solution around the third rotor, and $\Phi_i^{h,2}$ the RB generated with snapshots of the second turbine.

In Figure 4.11a, we compare the results of the classical NIRB, with the NIRB constrained version 3.2 and the rectification postprocessing phase (with the regularization parameter $\lambda = 0$ or $\lambda = 1e^{-10}$). As expected, the NIRB constrained version stabilizes the error and the rectification allows us to retrieve the fine solution error. Moreover, with the regularization parameter $\lambda = 1e^{-10}$, the NIRB error remains stable. Figure 4.11b shows that with two modes the constrained NIRB is efficient to retrieve the classical NIRB error on the third turbine with a RB generated by the second one (The error is defined by (4.4)).

All these results are consistent with what was expected of a 2-dimensional problem regarding the studies of the previous chapters. The wake effects are almost non-existent and two modes are always sufficient to accurately approximate the solutions. Therefore, it was a quite natural idea to complexify the data until reaching three dimensions.

### 4.2.2 3D wind turbine results

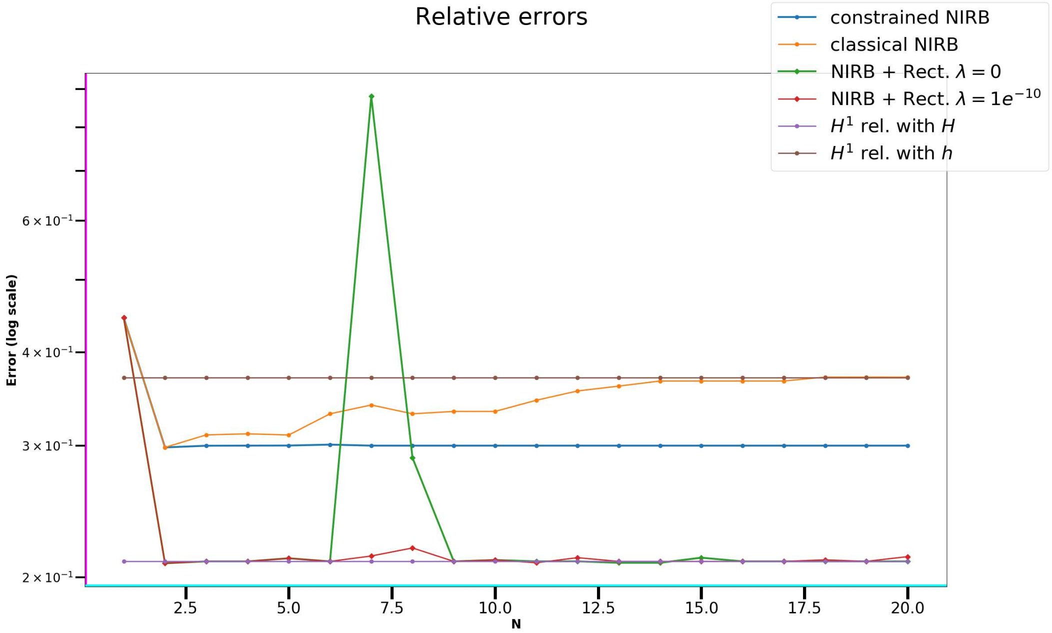
With three dimensions, we present two tests. They are quite similar to industrial cases, and as a consequence, there are so many degrees of freedom that we could not compare the NIRB approximation to a finer reference solution. Therefore, we present several comparisons between the NIRB approximation and the fine solution.

- The first study represents three turbines in line.
- The second one is a squared wind farm with 3 × 3 turbines.
- Another ongoing project is a 5 × 5 wind farm.

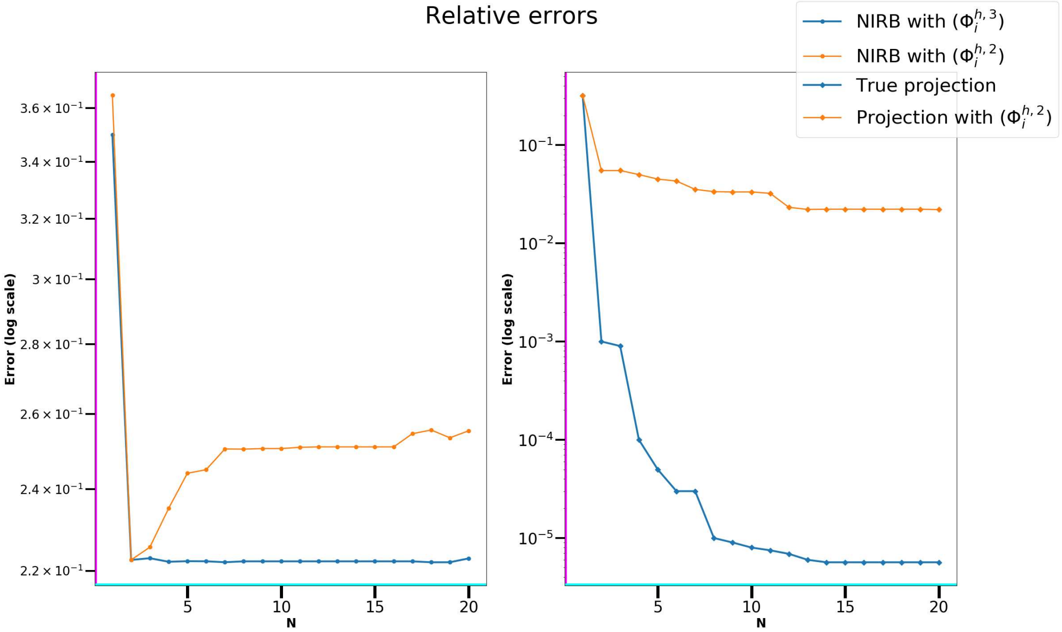
**3 wind turbines in line.** The general parameters are almost identical to the 2-dimensional cases, although now the problem is much more complex in terms of number of cells. Now the turbines are represented by discs. With this first test, we only change the velocity magnitude $u_{ref}$ (and not the incidence angle). We consider 27 snapshots with $u_{ref,i} = 3 + 0.5 \times i$, $i = 0, \ldots, 26$. The following table summarizes the mesh parameters 4.4.

| Rotor diameter (D) | 150m |
|---|---|
| Hub height | 100 m |
| Number of cells | 4 767 552 |
| fine mesh size $h$ | 5 m |
| coarse mesh size $H$ | 10 m |
| reference mesh size $h_{ref}$ | 2 m |

Table 4.4: Mesh parameters



(a) Relative $H^1$ errors between the reference solution $\mathbf{u}_r$ and the NIRB approximation (with and without rectification post-treatment $\lambda = 0$ and $\lambda = 1e^{-10}$ and with the constrained version) on the third rotor area with Ntrain=66 as a function of $N$, compared to the fine and coarse FV errors, $u_{ref} = 5$



(b) Averaged relative $H^1$ errors between the fine solution and the constrained NIRB (left) and between the fine solution and the true projection $\mathbf{u}_{hh}^N$(right) on the wind around the third turbine, RB generated with zooms on the second one

Figure 4.11: Relative $H^1$ errors with NIRB as a function of $N$ on the wind around the third turbine

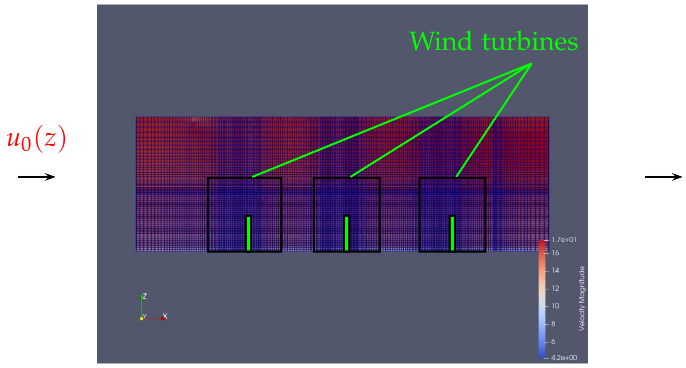


Figure 4.12: 3 wind turbines ($\mathbf{u}_0(z)$ depends on $u_{ref}$)

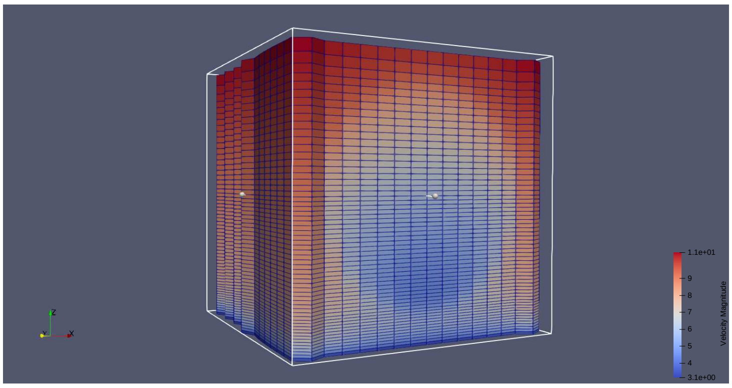


Figure 4.13: Zoom around one turbine

**Remark 42.** *We obtained these results using the library "Mordicus" in Python (see also the introduction 0.3).*

Now that we deal with three dimensions, the domain goes from $-150m$ to $150m$ on the third axis. The whole domain contains a large number nodes (1 924 461 nodes) whereas our subdomains around the turbines have 451 716 nodes for the fine mesh and 57 676 for the coarse mesh. Therefore, applying the two-grid method only on these areas significantly reduces the complexity. Figure 4.12 illustrates a plane of the domain and Figure 4.13 a zoom around the rotor. The actuator disc is clearly visible.

We present the NIRB errors with and without rectification on the third turbine in Figure 4.14. We emphasize that the figures side by side with similar results share the same scale and that the ordinate of the right-hand side figure is not displayed. The same results occur with the first two, we refer to the appendix for the corresponding Figure E.1.
We compare these results with the truncated NIRB method detailed in section 3.3 in Figure 4.15. For the coarse solution, the whole domain is truncated just before the first turbine and after the third one so that the whole domain is not large enough to properly represent the physics. As a consequence, the simulations do not accurately converge.

We observe in Figure 4.16 that the third turbine is well approximated by the second one but not by the first one.

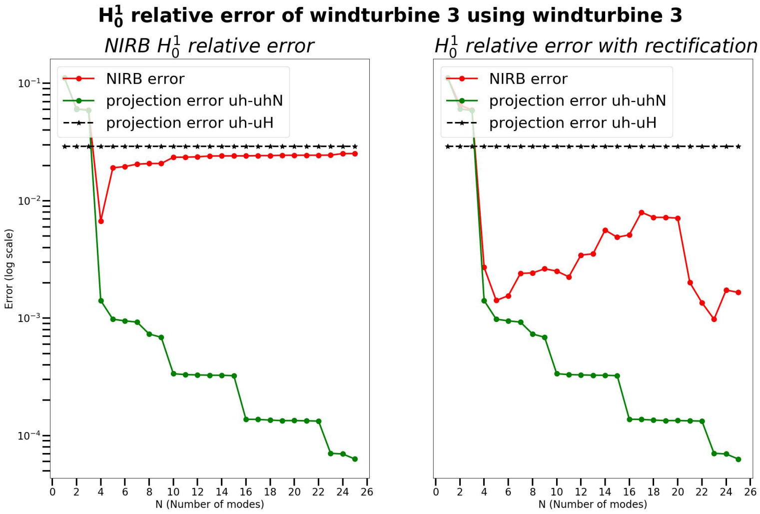


Figure 4.14: Third turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$, compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)

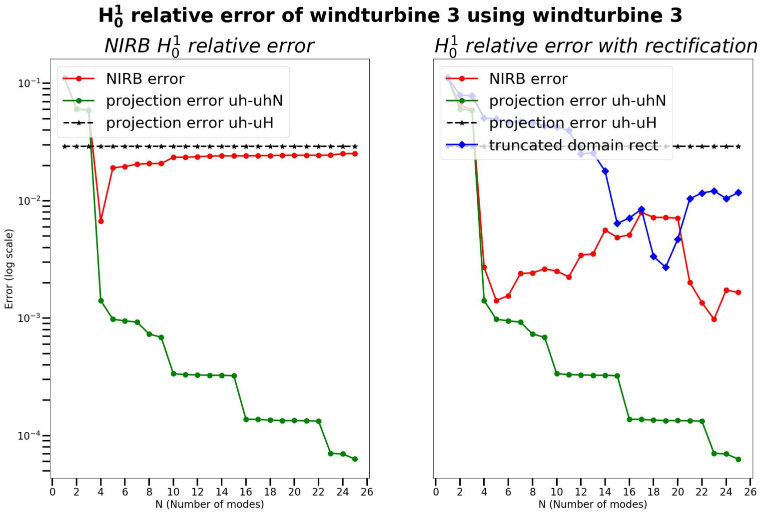
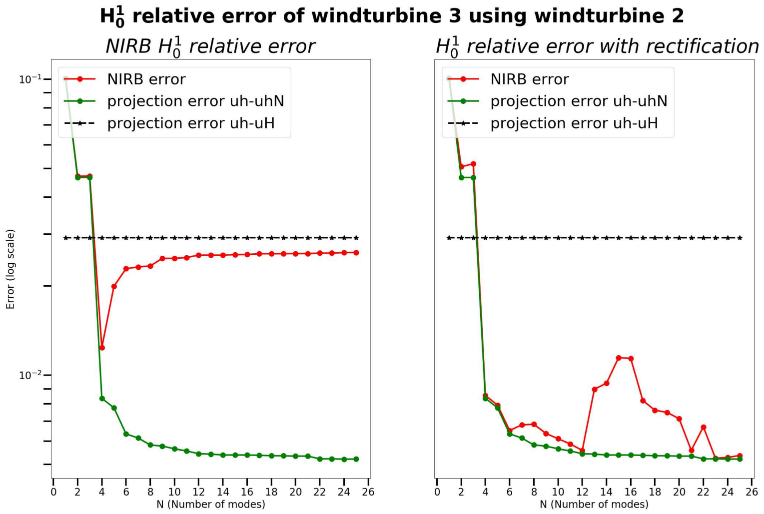
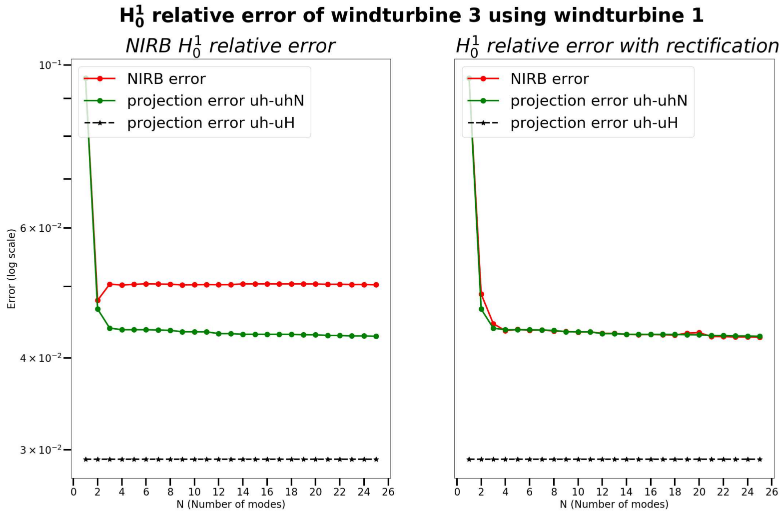


Figure 4.15: Third turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$ compared to the truncated NIRB (blue) (obtained from a smaller domain) and compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)



(a) Third turbine with RB generated by turbine n°2, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation as a function of the number of modes



(b) Third turbine with RB generated by turbine n°1, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation as a function of the number of modes

Figure 4.16: NIRB errors with (right) and without rectification (left)

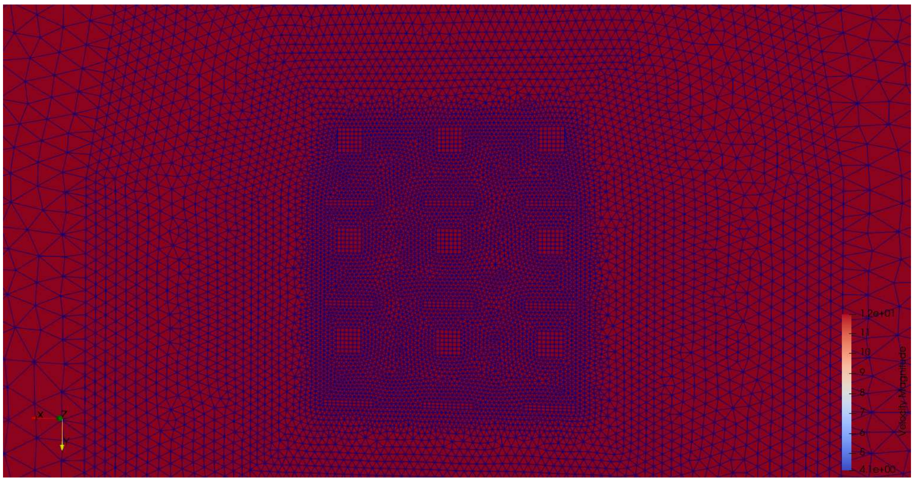


Figure 4.17: Zoom on the wind turbines mesh within the wind farm (wind coming from the bottom and moving to the left). The squares represent the rotors and the rectangles are the upstream probes.

$3 \times 3$ **turbines.** Figure Figure 4.17 present a zoom of the offshore wind farm with the 9 turbines. The turbines are located at the squares whereas the probes are located at the rectangles located upstream the rotors. Figure 4.18 illustrates the wake effects with $u_{ref} = 6$.

Now the magnitude is changing as well as the incidence angle $\theta$ which varies from $0$ to $45$. We have 220 snapshots and since we decompose the wind farm in terms of turbines it results in $9 \times 220 = 1980$ **snapshots**. We number the turbines as in the following table 4.5 where the wind comes from the bottom and shifts to the left (as shown for instance in Figure 4.18).

| | | |
|---|---|---|
| 6 | 7 | 8 |
| 3 | 4 | 5 |
| 0 | 1 | 2 |

Table 4.5: Numbering of the wind turbines with the wind coming from the bottom, and moving on the left

**The group that contains all the $1980$ snapshots is denoted $\mathcal{G}$. Then, we decompose on each turbine the snapshots to obtain subgroups denoted $\mathcal{G}_0, \ldots, \mathcal{G}_8$ according to the turbine which is considered.** The spatial parameters are summarized in table 4.6.

| | |
|---|---|
| Spatial domain | $[-4400, 4400]^2 \times [0, 1000]$ |
| Rotor diameter (D) | 150m |
| Hub height | 100 m |
| Number of nodes (fine mesh) | 3 637 438 nodes |
| Number of nodes (coarse mesh) | 686 504 nodes |
| fine mesh size $h$ | 5 m |
| coarse mesh size $H$ | 10 m |
| reference mesh size $h_{ref}$ | 2 m |

Table 4.6: Mesh parameters

Let us proceed with the numerical results.

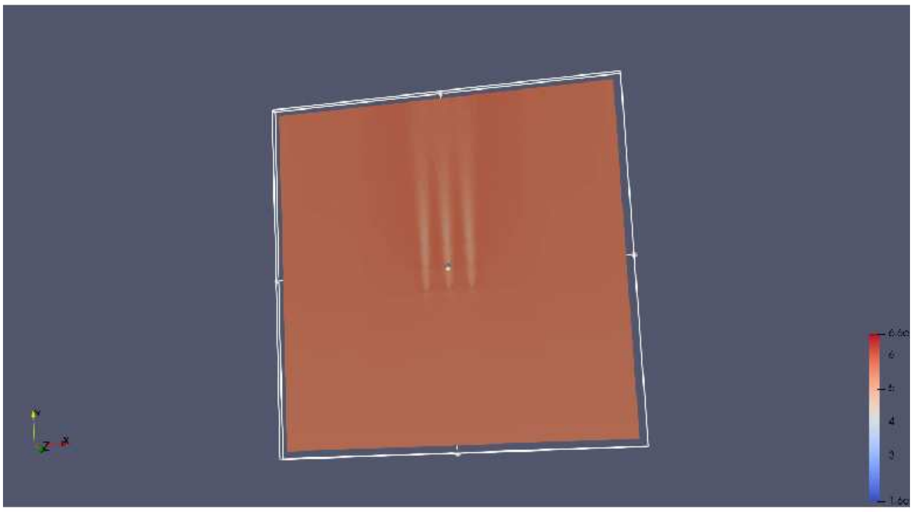


Figure 4.18: Turbines wake effects with $\theta = 0$ (wind coming from the bottom), $u_{ref} = 5.5$

**Results on the rotors.** First, we computed the eigenvalues (Figure 4.19a) and we noticed that I(1)=0.987 and I(100)=0.99999996. Thus, we present several results as a function of the number of modes varying from 1 to 100. We present the averaged NIRB error on all the turbines in Figure 4.19b.

We observed that turbines 2 and 8 were the most present within the RB. Thus, we employed them in the RB for representing the other turbines. The results are presented in Figure 4.20.

A natural idea is to approximate a turbine with the ones located upstream. This approach is illustrated in Figure 4.21. One ongoing project is to compare these results with the new NIRB tools (see chapter 3) and several truncations as in section 3.3.

**Results on the probes.** We obtained similar results on the probes, represented by the rectangles in Figure 4.17. In Figure 4.22, we display the results on the wind turbine 6.

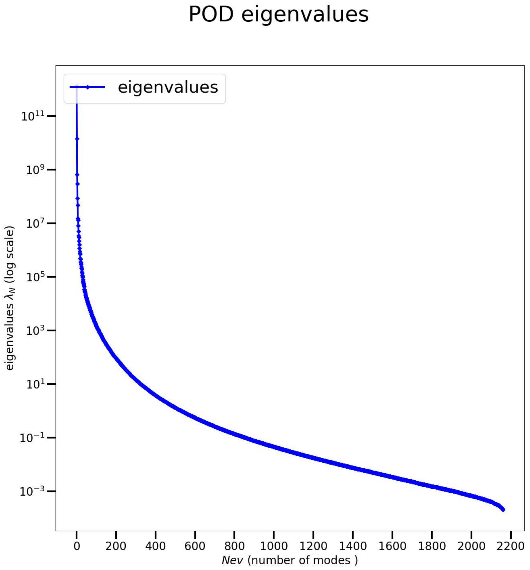
Sometimes the rectification leads to worse errors for instance when the coarse solution has not properly converged. A perspective would be to first classify the snapshots with respects to their coarse discretization error as in [127]. About the runtimes, with 500 iterations, we launched code_saturne in parallel with 100 processors and the computational times for coarse and fine FV solutions are summarized in subsequent table 4.7. The NIRB computational times depend on $N$ but for instance the latter with $N = 20$ are sumed up in table 4.8

| | |
|---|---|
| FV averaged time for one parameter on the fine mesh | 45:00 min |
| FV averages time for one parameter on the coarse mesh | 02:00 min |

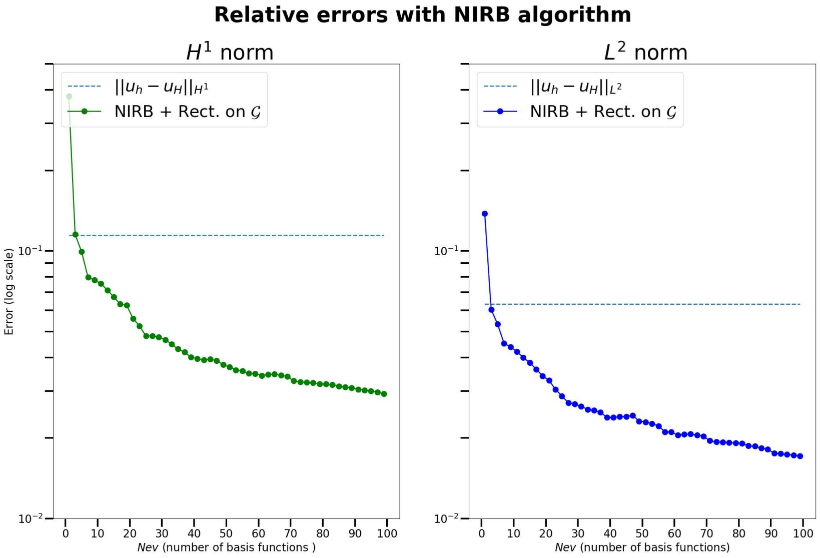
Table 4.7: 3D runtimes

| | |
|---|---|
| NIRB offline | 15:10 hours,min |
| NIRB online | 04:00 min |

Table 4.8: 3D NIRB runtimes



(a) Eigenvalues as a function of the number of modes



(b) Averaged relative $H^1$ (left) error and $L^2$ (right) between the fine solution and the NIRB rectified approximation as a function of the number of modes

Figure 4.19: NIRB results on all the turbines $\mathcal{G}$

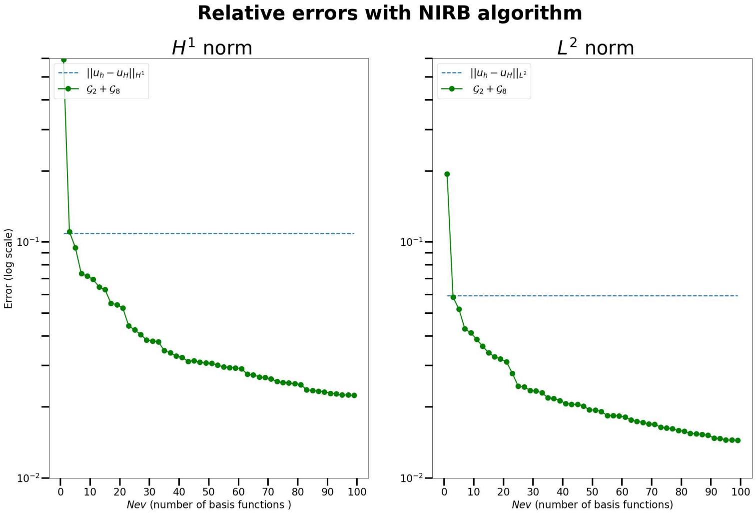


Figure 4.20: Averaged relative $H^1$ (left) and $L^2$ (right) errors between fine and NIRB rectified approximation on $\mathcal{G}\backslash(\mathcal{G}_2 + \mathcal{G}_8)$, RB generated with rotor 2 and 8 (compared to the error between $\mathbf{u}_h$ and $\mathbf{u}_H$).

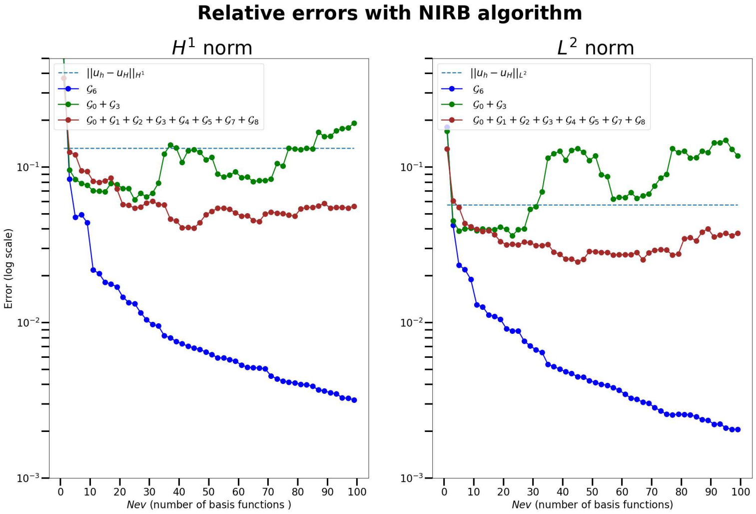


Figure 4.21: Averaged relative $H^1$ (left) and $L^2$ (right) errors between fine and NIRB rectified approximation with turbine 6, RB generated with different subgroups of rotors ($\mathcal{G}_6$ (blue) vs $\mathcal{G}\backslash\mathcal{G}_6$ (red) vs $\mathcal{G}_0 + \mathcal{G}_3$ (green, turbines upstream )).

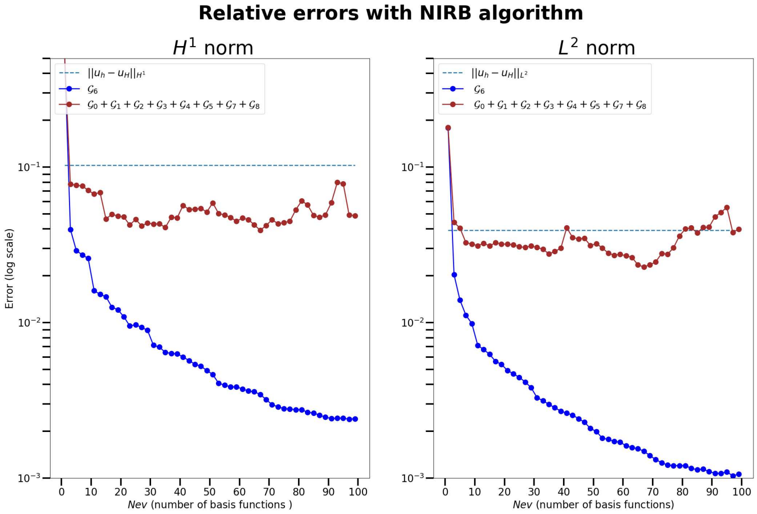


Figure 4.22: Averaged relative $H^1$ (left) and $L^2$ (right) errors between fine and NIRB rectified approximation with probes upstream turbine 6, RB generated with different subgroups of probes of rotors ($\mathcal{G}_6$ (blue) vs $\mathcal{G} \backslash \mathcal{G}_6$ (red)).

# Appendix A

# Short review on other NIRB methods

We present here RB methods that enable further reductions in POD runtimes. These methods have several degrees of non-intrusivity. We emphasize the fact that the non-intrusivity here means that the methods do not required any modifications in the source code of the external solver.

### A.0.1 Empirical Interpolation Method (EIM)

If the assumptions (**??**) and (**??**) do not hold, a key ingredient to treat the nonlinearity with the Galerkin POD is the Empirical Interpolation Method (EIM) [10]. The EIM build a linear combination of fully determined solutions from basis functions $(q_i)_{i=1,\dots,N}$ depending on some interpolating points (magic points) and some values $(\alpha_i)_{i=1,\dots,N}$ relying on certain instances of the parameter $\nu$, selected within the algorithm.
Let us introduce the method with the following example:
Consider a function

$$g(\mathbf{x},\boldsymbol{\nu}) = \frac{1}{\sqrt{(x_1-\nu_1)^2+(y_1-\nu_2)^2}},$$

with $\mathbf{x} = (x_1, x_2)$, and $\boldsymbol{\nu} = (\nu_1, \nu_2)$. Then, the first chosen parameter $\boldsymbol{\nu}^1$ is the one which maximizes $g$ on norm $\mathcal{L}^\infty$, and the associated magic point is the point which gives the most information on $g(.,\boldsymbol{\nu}^1)$, i.e. which maximizes its modulus. Then the first basis function is $q_1(\cdot) = \frac{g(.,\boldsymbol{\nu}^1)}{g(\mathbf{x}^1,\boldsymbol{\nu}^1)}$. To simplify notations, we will denote $g_i = g(\cdot,\boldsymbol{\nu}^i)$. The $N$ basis functions are constructed by recursivity with the following interpolation problem
Find $\{\alpha_j^{M-1}(g)\}_{1\leq M\leq N}$ such that

$$\forall\ 1 \leq i \leq M-1,\ \mathcal{I}_{M-1}(g)(\mathbf{x}^i) = g(\mathbf{x}^i), \tag{A.1}$$

and the EIM operator is defined by

$$\mathcal{I}_{M-1}[g] = \sum_{j=1}^{M-1} q_j \alpha_j^{M-1}. \tag{A.2}$$

We detail the algorithm (in the example, $n = 2$, and we denote $\mathcal{T}_h$ a triangulation of $\Omega$):

```
input : $\Gamma^N = (\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^N)$, where $\boldsymbol{\nu}^i \in \mathbb{R}^n$
output: $q_1, \ldots, q_N$
Initialization: $\boldsymbol{\nu}^1 = \arg\max_{\boldsymbol{\nu}} \| g(.,\boldsymbol{\nu}) \|_{\mathcal{L}^\infty}$
$g_1(.) = g(.,\boldsymbol{\nu}^1)$
$\mathbf{x}^1 = \arg\max_{\mathbf{x} \in \mathcal{T}_h} |g_1(\mathbf{x})|$
$q_1(.) = \frac{g_1(.)}{g_1(\mathbf{x}^1)}$
for $M$=2:$N$ do
    Set $\mathcal{I}_{M-1}[g(\boldsymbol{\nu})] = \sum_{j=1}^{M-1} \alpha_j^{M-1}(\boldsymbol{\nu}) q_j$, and find $\alpha_j^{M-1}(\boldsymbol{\nu})$ such that
        $$\mathcal{I}_{M-1}[g(\boldsymbol{\nu})](\mathbf{x}^i) = g(\mathbf{x}^i, \boldsymbol{\nu}), \forall i \in \{1, \ldots, M-1\},$$
      $\boldsymbol{\nu}_M = \arg\max_{\boldsymbol{\nu} \in \Gamma^N} \| g(.,\boldsymbol{\nu}) - \mathcal{I}_{M-1}[g(\boldsymbol{\nu})](.) \|_{\mathcal{L}^\infty}$,
    $g_M = g(.,\boldsymbol{\nu}_M)$,
    $\mathbf{x}_M = \arg\inf_{\mathbf{x} \in \mathcal{T}_h} |g_M(\mathbf{x}) - \mathcal{I}_{M-1}[g_M](\mathbf{x})|$,
    $q_M(.) = \frac{g_M(.) - \mathcal{I}_{M-1}[g_M](.)}{g_M(\mathbf{x}_M) - \mathcal{I}_{M-1}[g_M](\mathbf{x}_M)}$.
end
Solve the final interpolation problem for a specific value of $\boldsymbol{\nu}$: Find $\alpha_j^N(\boldsymbol{\nu})$ such that
        $$\mathcal{I}_M[g(\boldsymbol{\nu})](\mathbf{x}^i) = g(\mathbf{x}^i, \boldsymbol{\nu}), \forall i \in \{1, \ldots, N\},$$
  For a specific point $\mathbf{x}$, compute $\mathcal{I}_M(g(\boldsymbol{\nu}))(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j^N \cdot q_j$
```

**Algorithm 7:** EIM algorithm

There exists a generalized form of this method (GEIM). The GEIM replaces the M pointwise evaluations used by the EIM by general measures [68]. In the presence of measurement noise, a stabilization of the method can be found in [9]. We implemented the classical EIM algorithm with FreeFem++.

### A.0.2 Proper Generalized Decomposition (PGD)

The POD RB construction requires appropriate snapshots, computed in the offline stage. This step is very greedy in time. The Proper Generalized Decomposition (PGD) method is a good way to overcome this issue, if the RB orthogonalization is not required during the online stage. As a result, the basis is valid for any range parameters. The PGD is used for problems depending on several parameters. It has first been introduced as the " radial time-space approximation" in [87]. It is based on a variable separation to break down the solution into less costly pieces. This decomposition can be done on spatial, temporal or any control parameters (such as the Reynolds number for instance). The RB is generated with an iterative process, and several approaches exist. An application of the PGD method can be found in [63, 62] in the field of railway dynamics. This method is rather intrusive, since a prior knowledge of the model is required. A way to develop the PGD in a non-intrusive way can be found in [53], using the s-PGD method and in [136]. This approach involves looking for a solution to a PDE as a product sum of the functions of each space variable. We consider a problem defined in a space of dimension $d$ with the unknown field $u(x_1, \dots, x_d)$. The PGD yields an approximation in the separated form

$$u(x_1, \dots, x_d) \simeq \sum_{i=1}^{N} F_i^1(x_1) \times \cdots \times F_i^d(x_d), \tag{A.3}$$

where the functions $F_i^j(x_j)$ are unknown a priori and determined through an iterative process. This is achieved by using the weak form of the problem.

### A.0.3 Hyper-reduction

Another way to reduce the space solution is by achieving hyper-reduction techniques [117, 119, 75]. To estimate a new solution for a system with several constitutive equations, the Galerkin POD procedure does not modify the number of equations that must be solved. An approach for reducing it is by the creation of a Reduced Integration Domain "RID". These methods are useful when the shape functions of the FE model have a bounded support that do not cover all the spatial domain. After having determinate the integration domain, the equations are solved for a new parameter on the RID. This is the key ingredient to the reduction of complexity. Then with the Gappy-POD [100], it is possible to reconstruct the solution on the whole spatial domain. The method is not fully non-intrusive since the black-box code need to retrieve the Dirichlet BCs. In [56], this approach is compared to the DEIM method [12].

# Appendix B

# On the "a posteriori" study

This section is an ongoing project with Y. Maday and R. Chakir.
Let us consider the model problem 16a-16b in 2D, simulated for several parameters $\mu_i \in \mathcal{G}$, and its NIRB approximation $u_{Hh}^N(\mu)$ for a new parameter $\mu \in \mathcal{G}$. The corresponding exact solution is denoted $u(\mu)$. In order not to make the notations too cumbersome, we will denote by $u_{Hh}^N$ and $u$ the NIRB approximation and the exact solution for the new parameter $\mu$. The main purpose of this study is to give certified a posteriori error estimates for the NIRB approximation. Such error bounds are useful to find the required coarse mesh size. Usually, errors estimates on RBM relies on a residual approach [123, 132, 26]. In such case, the approximation needs to be the solution of the reduced problem. To the best of our knowledge, the two-grid method has not already been studied with equilibrated fluxes a-posteriori estimators. This very general approach allows us to obtain guaranteed upper and lower bounds for very general approximations that do not need to be solution of the discrete problem. Thus, they are very adequate for two-grid approximations.

We will denote by $\sigma := -A(\mu)\nabla u$ the associated exact flux, such that

$$-\ \text{div}\ (\sigma) = f. \tag{B.1}$$

In this section, we consider certified a posteriori errors, from several a posteriori studies [115, 49, 106] on flux reconstruction. An indicator error can also be employed. Let us first introduce the $\mathbf{H}(\text{div}, \Omega) = \mathbf{H}(div, \Omega)$ space, we recall that the space $\mathcal{D}(\Omega)$ is the space of functions from $C^\infty(\overline{\Omega})$ with a compact support in $\Omega$.

**Definition 8.** *Weak divergence. Let a vector function $\boldsymbol{v} : \Omega \to \mathcal{R}^d$ be given. We say that $\boldsymbol{v}$ admits a weak divergence if*

1. $\boldsymbol{v} \in [L^2(\Omega)]^d$;
2. *there exists a function $w : \Omega \to \mathcal{R}$ such that*
    - $w \in L^2(\Omega)$;
    - $(\boldsymbol{v}, \nabla\Phi) = -(w, \Phi), \forall \Phi \in \mathcal{D}(\Omega)$.

The function $w$ is called the weak divergence of $\mathbf{v}$. We use the notation $\nabla \cdot \mathbf{v} = w$.

**Definition 9.** *The space $\mathbf{H}(div, \Omega)$. The space $\mathbf{H}(div, \Omega)$ is the space of all the functions which admit the weak divergence.*

If the NIRB solution is non-conform, we need the following definition of the broken $H^1$ space.

**Definition 10.** *The space $H^1(\mathcal{T}_h)$. The so-called broken $H^1(\mathcal{T}_h)$ space is given by*

$$H^1(\mathcal{T}_h) = \{v_h \in L^2(\Omega),\ v_{h|K} \in H^1(K),\ \forall K \in \mathcal{T}_h\}. \tag{B.2}$$

We can thus write the broken weak gradient $\nabla_h v \in [L^2(\Omega)]^d$, where

$$(\nabla_h v)_K := \nabla(v_{|K}). \tag{B.3}$$

If $v \in H^1(\Omega)$, then the broken gradient is equal to the usual gradient, so we will consider the same notation $\nabla$ for both. In the same way, the broken divergence can be defined and denoted $\nabla \cdot_h v$ and we have for functions $v \in \mathbf{H}(\mathrm{div}, \Omega)$, $\nabla \cdot_h v = \nabla \cdot v$.
Since the exact solution is not known, for the NIRB a-posteriori estimate, we consider three approaches based on a flux reconstruction $\sigma_h$ in $\mathbf{H}(\mathrm{div}, \Omega)$. Let $u \in H^1_0(\Omega)$ be the solution of 16a-16b, and let $u_h \in H^1_0(\Omega)$ be an approximate solution in $H^1(\mathcal{T}_h)$. Let us consider the associated approximate flux $-\mu \nabla u^N_{Hh} \in \mathbf{H}(\mathrm{div}, \Omega)$.
**In general,** $u^N_{Hh} \notin H^1_0(\Omega)$, $-A(\mu)\nabla u^N_{Hh} \notin \mathbf{H}(\mathbf{div}, \Omega)$, $\nabla \cdot (-A(\mu)\nabla u^N_{Hh}) \neq f$.

1. The first approach comes from an estimator in [115]. It provides a general certified estimator. However, it is less accurate as an indicator. The positive side is that this estimator is very general. The idea is the following:
   If a flux $\sigma_h = f$ can be found, then we may apply the Prager-Synge equality [108] and we obtain
   $$\left\| A(\mu)^{1/2} \nabla (u - u^N_{Hh}) \right\|_{L^2(\Omega)} \leq \left\| A(\mu)^{1/2} \nabla u^N_{Hh} + \frac{\sigma_h}{A(\mu)^{1/2}} \right\|_{L^2(\Omega)},$$
   but in practice, it is rarely the case. If the flux $\sigma$ is not in $\mathbf{H}(\mathrm{div}, \Omega)$ (for instance if $u^N_{Hh}$ is in $H^1(\mathcal{T}_h)$) then for the following estimator, the flux needs to be reconstructed and thus it leads the minimization problem
   $$\min_{\sigma_h \in \mathbf{H}(\mathrm{div}, \Omega)} \left\| \sigma_h - A(\mu) \nabla u^N_{Hh} \right\|_{H^1}. \tag{B.4}$$
   This global reconstruction can be greedy in time. Thus, for the flux reconstruction in $\mathbf{H}(\mathrm{div}, \Omega)$, we can use the flux reequilibration as in the subsequent method. The estimator is defined by
   $$|u - u^N_{Hh}|^2_{H^1_0} \leq \sum_{K \in \mathcal{T}_h} \Bigg( \underbrace{\left\| \sqrt{A(\mu)} \nabla u^N_{Hh} + \frac{\sigma_h}{\sqrt{A(\mu)}} \right\|_{L^2(K)}}_{\text{Flux part}} + \frac{h_\Omega}{\pi} \underbrace{\| f - \nabla \cdot \sigma_h \|_{L^2(K)}}_{\text{oscillations part}} \Bigg)^2 \tag{B.5}$$
   $$+ \sum_{K \in \mathcal{T}_h} |u^N_{Hh} - s_h|^2_{H^1_0(K)},$$
   The main difference between this estimator and the next one, is that $h_K$ is used instead of $h_\Omega$ in the estimator defined by (B.7). As a consequence, this approach yields a worse estimator bound. If the Galerkin orthogonality, introduced in the next approach (see (B.6)) is fulfilled, then we retrieve the previous estimator, since the term linked to the oscillations of the function $f$ in (B.5) is very small, as we will observe in the numerical results.

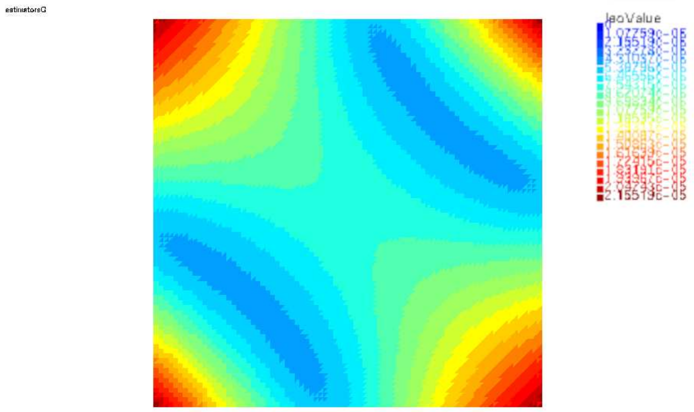


Figure B.1: Error indicator with the rectification, range from 0 to $2.15519e^{-5}$

**Numerical results.** We consider two tests. The first one is very simple, with a constant diffusion coefficient on the whole domain $\Omega$, and the second one is with a piecewise constant diffusion coefficient on $\Omega$.

(a) $A(\mu) = \mu$ constant on $\Omega$.
Figure B.1 displays the error indicator with the rectification post-treatment and may be compared to Figure B.3a (which gives the energy error with the constant diffusion coefficient).

| | |
|---|---|
| Energy error | 0.000962165 |
| Estimator | 0.00101037 |
| effectivity index | 1.0501 |

Table B.1: Energy error and estimator

These results illustrate the fact that this estimator can be efficient if the oscillation term is very small (table B.3).

(b) $A(\mu)$ piecewise-constant on $\Omega$.
Figure B.2 may be compared to Figure B.6a which gives the energy error.

| | |
|---|---|
| Energy error | 0.000104113 |
| Estimator | 0.00267907 |
| effectivity index | 25.732 |

Table B.2: Energy error and estimator

2. The second approach uses a local reconstruction with a reequilibrated flux $\sigma_h$ in $\mathbf{H}(\text{div}, \Omega)$, which must fulfill the Galerkin orthogonality condition (B.6) [49]. This condition implies that the flux gradient reconstruction must be equal to the function $f$ on average on each cell $K$.

$$(\nabla \sigma_h, 1)_K = (f, 1)_K. \tag{B.6}$$

This flux reconstruction is based on an approach derived from mixed FE theory and a reequilibration on patchwise Neumann problems, as in [18]. It yields a certified estimator

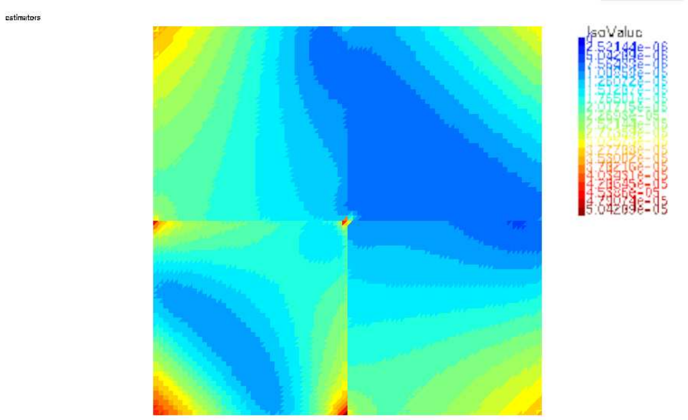


Figure B.2: Error indicator with the rectification, range from 0 to $5.04209e^{-5}$

and a good indicator. The NIRB approximation may satisfy equation (B.6) in particular when using the rectification postprocessing stage [30]. The estimator gives either a lower or an upper bound. It is decomposed in one term linked to the flux and one term linked to the problem residual. The estimator is defined by

$$|u - u_{Hh}^N|_{H_0^1}^2 \leq \sum_{K \in \mathcal{T}_h} \Bigg( \underbrace{\left\| \sqrt{A(\mu)} \nabla u_{Hh}^N + \frac{\sigma_h}{\sqrt{A(\mu)}} \right\|_{L^2(K)}}_{\text{Flux part}} + \frac{h_K}{\pi} \underbrace{\| f - \nabla \cdot \sigma_h \|_{L^2(K)}}_{\text{oscillations part}} \Bigg)^2 \tag{B.7}$$
$$+ \sum_{K \in \mathcal{T}_h} |u_{Hh}^N - s_h|_{H_0^1(K)}^2,$$

where $s_h \in H_0^1(\Omega)$ is a potential reconstruction of the NIRB approximation. This term is added into the estimator if the NIRB approximation is non-conform and does not belong to $H_0^1$ but to $H^1(\mathcal{T}_h)$ (as in the definition B.2). The indicator is the same than B.7 without the sum on each cell. The effectivity index to assess the estimator is given by

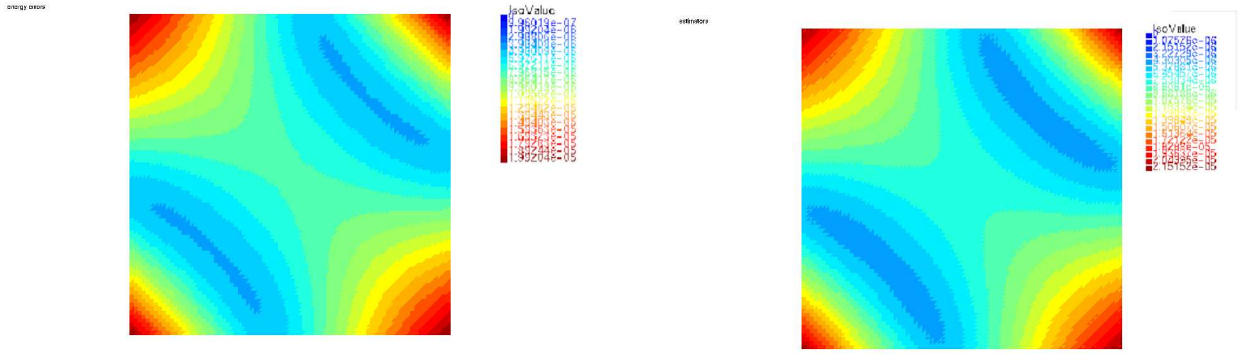
$$\text{Effectivity index } := \frac{\text{Estimator}}{\text{Energy error}}, \tag{B.8}$$

and it must be close to one.

**Numerical results on this first estimator.** We consider the same tests as with the previous approach.

(a) $A(\mu) = \mu$ constant on $\Omega$.
The domaine $\Omega$ is the unit square. The diffusion coefficient is such that

$$A(\mu) = \mu \in \mathcal{G} = [10, 60]. \tag{B.9}$$



(a) Energy error with the rectification, range from 0 to $1.99204e^{-5}$

(b) NIRB Indicator, range from 0 to $2.15519e^{-5}$

Figure B.3: NIRB with rectification post-process

The new parameter is $\mu = 20$ and the function $f$ is defined by

$$f(x,y) = -2(x^2 + y^2) + 2(x + y) \tag{B.10}$$

such that the analytical solution is given by

$$u(x,y;\mu) = \frac{x(x-1)y(y-1)}{A(\mu)}. \tag{B.11}$$

We compare the energy error between the exact solution and the NIRB approximation to the a-posteriori error with and without the rectification post-treatment in Figure B.3.

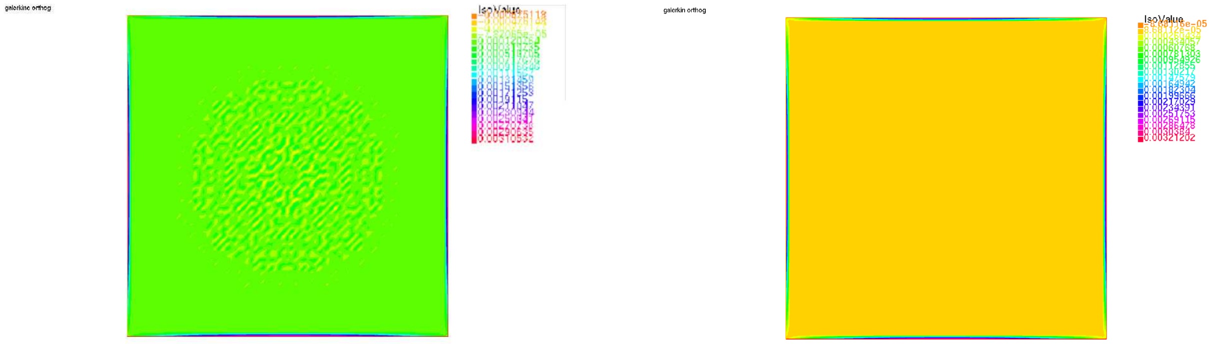
The results on the estimator and on the energy error are summarized in the following table B.3.

| | |
|---|---|
| Galerkin orthogonality (max) | 0.00312521 |
| Energy error | 0.000962165 |
| Estimator | 0.00100668 |
| Flux part | 0.00100653 |
| Oscillation part | 1.65786e-07 |
| effectivity index | 1.0462 |

Table B.3: Energy error and estimator

With the rectification post-treatment, the a-posteriori error is very close to the energy error, as well as for the indicator. Indeed, this postprocess helps us retrieve the fine FEM approximation which satisties the Galerkin orthogonality B.6 (since the parameter is constant on the whole domain). Without the rectification, Figure B.5 display less accurate results than with the rectification. It is not surprising since Galerkin orthogonality is satisfied to a lesser extent without rectification as shown by Figure B.4a.

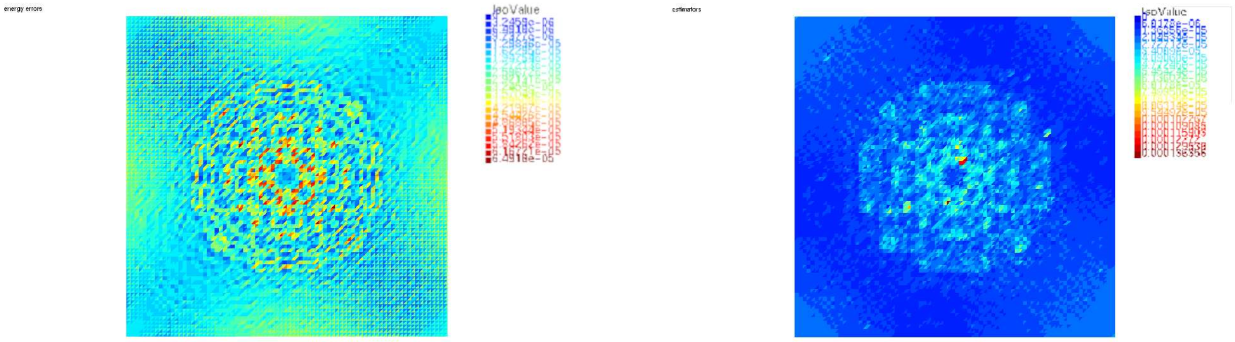
(b) $A(\mu)$ piecewise constant on $\Omega$.
We proceed with a non-constant diffusion coefficient, and in our tests we decomposed the domain $\Omega$ in four subdomains. We chose random coefficients for the



(a) without the rectification, range from $-0.000625118$ to 0.00310532

(b) with the rectification, range from $-8/68116e^{-5}$ to 0.00321202

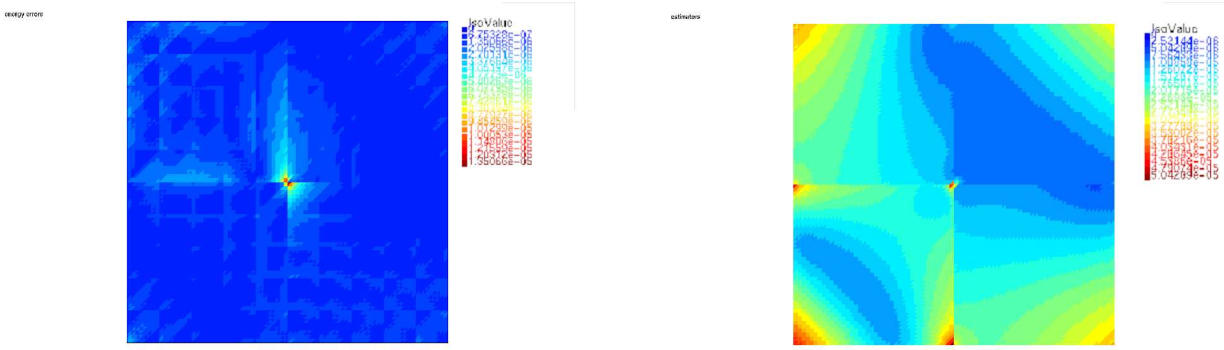
Figure B.4: Galerkin-Orthogonality



(a) Energy error, range from 0 to $6.4918e^{-05}$

(b) Error indicator, range from 0 to 0.000136356

Figure B.5: NIRB without rectification post-process



(a) Energy error, range from 0 to $1.350866e^{-05}$

(b) Error indicator

Figure B.6: NIRB with the rectification post-process, range from 0 to $5.04209^{-05}$

snapshots and the new parameter $\mu$ is such that

$$A(\mu, \mathbf{x}) = \begin{cases} 1.5, & \text{for } x \leq 0.5, y \leq 0.5, \\ 3.8, & \text{for } x \geq 0.5, y > 0.5, \\ 4.7, & \text{for } x > 0.5, y \leq 0.5, \\ 7, & \text{for } x > 0.5, y > 0.5, \end{cases}$$

where $\mathbf{x} = (x, y)$.

The results with the rectification post-treatment on the estimator and on the energy error are summarized in the following table B.3.

| | |
|---|---|
| Energy error | 0.000104113 |
| Estimator | 0.00177461 |
| Flux part | 0.00177446 |
| Oscillation part | 1.65786e-07 |
| effectivity index | 17.0451 |

Table B.4: Energy error and estimator

We observe that the estimator is not very accurate. This is justified by the fact that the rectification post-treatment does not help us in recovering Galerkin orthogonality property as shown in Table B.4.

3. The results on the non-constant diffusion coefficient exhibit the importance of the Galerkin-orthogonality property B.6 to obtain a certified and accurate a-posteriori bound. A third possibility we did not yet test, is to construct an algebraic flux in order to satisfy the Galerkin orthogonality for any approximation in $H_0^1$. Thus, we need to construct in addition to $\sigma_h$ an algebraic flux, denoted by $\sigma_{alg} \in \mathbf{H}(\text{div}, \Omega)$, such that

$$(div(\sigma_h + \sigma_{alg}), 1)_K = (f, 1)_K.$$

The a-posteriori estimate that follows is very general and thus, it yields an accurate indicator and good upper and lower bounds [106].

# Appendix C

# Reminders on TPFA and HMM methods

### C.0.1 Two-Point Flux Approximation Finite Volumes (TPFA)

The TPFA Finite Volume scheme is a non conforming scheme. Let us consider the case of Cartesian grids, namely TPFA-CG, which is an easier case to understand how we find the related gradient discrete scheme.

#### TPFA-CG

Let $\mathcal{M}$ be a rectangular mesh of a rectangle $\Omega \subset \mathbb{R}^2$. Let $\mathcal{E}$ be the finite set of edges of the mesh. For any $\sigma \in \mathcal{E}$, $\overline{x}_\sigma$ is the center of mass of $\sigma$.
For each $K \in \mathcal{M}, \mathbf{x}_K$ is the center of mass of K. $\mathcal{P}$ is the set of vertices of the mesh and for $K \in \mathcal{M}$, $\mathcal{V}_K$ is the set of vertices of K. For each $K \in \mathcal{M}$ and each $s \in \mathcal{V}_K, V_{K,s}$ is the rectangle defined by $\overline{x}_\sigma, s, \overline{x}_{\sigma'}$ and $\mathbf{x}_K$, where $\sigma$ and $\sigma'$ are the edges of K touching $s$.
For a FV scheme, we have to solve:

$$\forall K \in \mathcal{M}, \sum_{\sigma \in \mathcal{E}} F_{K,\sigma} = \int_\Omega f, \tag{C.1}$$

where $F_{K,\sigma}$ is an approximation of the exact fluxes, which in this example is $-\int_\sigma \nabla \overline{\mathbf{u}}.\mathbf{n}_{K,\sigma}$. The TPFA-CG consists in taking

$$F_{K,\sigma} = -|\sigma| \frac{u_\sigma - u_K}{d_{\sigma,K}}, \tag{C.2}$$

where $d_{\sigma,K} = |\overline{x}_\sigma - \mathbf{x}_K|$.
With all these definitions, we get the following scheme, (Part 1, [45]):

$$\sum_{K \in \mathcal{M}} \sum_{\sigma \in \mathcal{E}_K} \frac{|\sigma|}{d_{\sigma,K}} (v_\sigma - v_K)(u_\sigma - u_K) = \sum_{K \in \mathcal{M}} v_K \int_K f, \tag{C.3}$$

and this equation can be expressed as a gradient scheme.

- Let $X_{\mathcal{D},0} = \{v = ((v_K)_{K \in \mathcal{M}}, (v_\sigma)_{\sigma \in \mathcal{E}}) : v_K \in \mathbb{R}, v_\sigma \in \mathbb{R}, v_\sigma = 0 \; if \; \sigma \in \mathcal{E}_{ext}\}$
- $\Pi_\mathcal{D} : X_{\mathcal{D},0} \to L^2(\Omega)$ is the following piecewise constant reconstruction on the mesh: $\forall v \in X_{\mathcal{D},0}, \forall K \in \mathcal{M}, \Pi_\mathcal{D} v = v_K \; on \; K.$

- If $K \in \mathcal{M}$ and $s \in \mathcal{V}_K$ is such that $\sigma$ and $\sigma'$ are the edges of K sharing the vertex $s$, we reconstruct a gradient by setting:

$$\nabla_{K,s} u_{\mathcal{D}} = \frac{u_\sigma - u_K}{d_{\sigma,K}} \mathbf{n}_{K,\sigma} + \frac{u_{\sigma'} - u_K}{d_{\sigma',K}} \mathbf{n}_{K,\sigma'}. \tag{C.4}$$

  Then, $\nabla_{\mathcal{D}} u_{\mathcal{D}}$ is the piecewise constant function equal to $\nabla_{K,s} u_{\mathcal{D}}$ on $V_{K,s}$.

Thus, the equation C.1 can be written as the following variational problem:
Find $u_{\mathcal{D}} \in X_{\mathcal{D},0}$ such that, $\forall v_{\mathcal{D}} \in X_{\mathcal{D},0}$,

$$\int_\Omega \nabla_{\mathcal{D}} u_{\mathcal{D}} . \nabla_{\mathcal{D}} v_{\mathcal{D}} = \int_\Omega f \Pi_{\mathcal{D}} v_{\mathcal{D}}. \tag{C.5}$$

The third item (C.4) is equivalent to the following definition of $\nabla_{\mathcal{D}}$:
$\forall v \in X_{\mathcal{D},0}, \forall K \in \mathcal{M}$,

$$\nabla_{\mathcal{D}} v = \nabla_K v = \frac{1}{|K|} \sum_{\sigma \in \mathcal{E}_K} |\sigma| (v_\sigma - v_K) \mathbf{n}_{K,\sigma}, \tag{C.6}$$

$$= \frac{1}{|K|} \sum_{\sigma \in \mathcal{E}_K} |\sigma| v_\sigma \mathbf{n}_{K,\sigma}. \tag{C.7}$$

Now, we can define the equivalent norm of $H_0^1$ on $X_{\mathcal{D},0}$:

$$\|\nabla_{\mathcal{D}} u_{\mathcal{D}}\|_{L^2}^2 = \|u_{\mathcal{D}}\|_{\mathcal{D}}^2 = \sum_{K \in \mathcal{M}} \sum_{\sigma \in \mathcal{E}_K} |\sigma| d_{K,\sigma} \left(\frac{v_\sigma - v_K}{d_{K,\sigma}}\right)^2. \tag{C.8}$$

We notice that these two norms are not always equals but with a general polytopal mesh, we have (See Lemma B.10 [45] for more precisions):

$$\|\nabla_{\mathcal{D}} u_{\mathcal{D}}\|_{(L^2)^d} \leq \sqrt{d} \|u_{\mathcal{D}}\|_{\mathcal{D}}, \tag{C.9}$$

where d is the dimension of $\Omega$.

Let us now consider the case of triangular meshes.

**Definition 11** (Classical TPFA triangulation. [47]). *Let $\Omega$ be a polygonal open set of $\mathbb{R}^2$. A classical TPFA triangulation of $\Omega$ is a conforming acute triangulation $\mathcal{T}$ of $\Omega$ such that, for all $K \in \mathcal{M}$, $\boldsymbol{x}_K$ is the circumcenter of K, and that is constructed in one of the following ways:*

- *Subdivision: an initial triangulation $\mathcal{T}_0$ of $\Omega$ is chosen, and then subdivided by creating on each edge an identical number of equally spaced points, by joining the corresponding points on different edges, and by adding the interior points resulting from intersections of the lines thus created,*
- *Reproduction by symmetry: an initial triangulation $\mathcal{T}_0$ of the unit square is chosen, this unit square is shrunk by a factor N and reproduced in entire domain by symmetry.*
- *Reproduction by translation: an initial triangulation $\mathcal{T}_0$ of the unit square is chosen, this unit square is shrunk by a factor N and reproduced in the entire domain by translation.*

**Definition 12.** *(TPFA scheme.) Consider a TPFA-admissible mesh $\mathcal{T}$ with $(\boldsymbol{x}_K)_{K \in \mathcal{M}}$ circumcenters in the cells (acute triangles) such that, denoting by $\mathcal{E}_K$ the edges of $K \in \mathcal{M}$,*

1. *for any neighbouring cells K and L in $\mathcal{M}$, if $\sigma \in \mathcal{E}_K \cap \mathcal{E}_L$ then $(\boldsymbol{x}_K \boldsymbol{x}_L) \perp \sigma$.*

2. *for any cell K $\in \mathcal{M}$, if $\sigma \subset \partial\Omega$ then $(\boldsymbol{x}_K + \mathbb{R}^+ \boldsymbol{n}_{K,\sigma}) \cap \sigma \neq \emptyset$, where $\boldsymbol{n}_{K,\sigma}$ is the normal to $\sigma$ pointing outward K.*

*Let $\mathcal{E}_{int}$ be the set of edges interior to $\Omega$ and $\mathcal{E}_{ext}$ be the set of edges lying on $\partial\Omega$. If A is an isotropic tensor, that is, $A(\boldsymbol{x}) = a(\boldsymbol{x})Id$ for some $a(\boldsymbol{x}) \in (0, \infty)$, the TPFA method for problem* (16a)-(16b) *on $\mathcal{T}$ reads:*

$$\begin{cases} \text{Find } \boldsymbol{u}_h = (\boldsymbol{u}_K)_{K\in\mathcal{M}} \text{ such that:} \\ \forall K \in \mathcal{M},\ \mu \sum\limits_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{int}} \tau_\sigma(\boldsymbol{u}_K - \boldsymbol{u}_L) + \sum\limits_{\sigma\in\mathcal{E}_K\cap\mathcal{E}_{ext}} \tau_\sigma \boldsymbol{u}_K = |K| f_K, \end{cases} \tag{C.10a}$$

*where $f_K$ is the value of f on $\boldsymbol{x}_K$, L is the triangle on the other side of $\sigma$ if $\sigma \in \mathcal{E}_K \cap \mathcal{E}_{int}$, and with $a_K$ being the average value of a on K,*
*$\forall \sigma \in \mathcal{E}_{int}$, if $K \neq L$ are the cells on both sides of $\sigma$, $\tau_\sigma = |\sigma| \frac{a_K a_L}{a_K d_{L,\sigma} + a_L d_{K,\sigma}}$,*
*$\forall \sigma \in \mathcal{E}_{ext}$, if K is the cell such that $\sigma \in \mathcal{E}_K$, $\tau_\sigma = |\sigma| \frac{a_K}{d_{K,\sigma}}$.*

We define

$$\|\mathbf{u}_h\|^2_{L^2(\Omega)} = \sum_{K\in\mathcal{M}} |K| \mathbf{u}_K^2, \tag{C.11}$$

and a discrete $H^1_0$ semi-norm

$$\|\mathbf{u}_h\|^2_{1,\mathcal{D}} = \sum_{\sigma\in\mathcal{E}_{int}} \tau_\sigma(\mathbf{u}_K - \mathbf{u}_L)^2 + \sum_{\sigma\in\mathcal{E}_{ext}} \tau_\sigma \mathbf{u}_K^2 = \frac{1}{\mu}\int_\Omega f(\mathbf{x})\mathbf{u}_h(\mathbf{x}). \tag{C.12}$$

This semi-norm satisfies the discrete Poincaré's inequality [42]

$$\|\mathbf{u}_h\|_{L^2(\Omega)} \leq C\|\mathbf{u}_h\|_{1,\mathcal{D}} \tag{C.13}$$

### C.0.2 HMM convergence

The classical convergence in $\mathcal{O}(h)$ with the discrete $H^1$ can be proved with the following functions. See [45] for more details. The quality of the triplet $(\mathcal{M}, \mathcal{E}, \mathcal{P})$ is measure through the constant $C_\mathcal{D}$ , and the functions $S_\mathcal{D} : H^1_0(\Omega) \to [0, +\infty)$ and $W_\mathcal{D} : H_{div}(\Omega) \to [0, +\infty)$ respectively defined by:

$$C_\mathcal{D} = \underset{w\in X_{\mathcal{D},0}, w\neq 0}{sup} \frac{\|\Pi_\mathcal{D} w\|_{L^2}}{\|\nabla_\mathcal{D} w\|_{(L^2)^d}}, \tag{C.14}$$

$$\forall \varphi \in H^1_0,\ S_\mathcal{D}(\varphi) = \underset{w\in X_{\mathcal{D},0}}{min} (\|\Pi_\mathcal{D} w - \varphi\|_{L^2} + \|\nabla_\mathcal{D} w - \nabla\varphi\|_{(L^2)^d}), \tag{C.15}$$

$$\forall \psi \in H_{div}, W_\mathcal{D}(\psi) = \underset{w\in X_{\mathcal{D},0}, w\neq 0}{max} \frac{\int_\Omega (\Pi_\mathcal{D} w\ div(\psi) + \nabla_\mathcal{D} w.\psi)}{\|\nabla_\mathcal{D} w\|_{(L^2)^d}} \tag{C.16}$$

The constant $C_\mathcal{D}$ is used to measure the coercivity (it yields a discrete Poincaré inequality). The function $S_\mathcal{D}$ is the interpolation error and it measures the consistency of the GD. $W_\mathcal{D}$ measures the limit-conformity of the GD. We then get the following estimations (Part 1.2 [45]):

$$\|\Pi_\mathcal{D} u_\mathcal{D} - \overline{\mathbf{u}}\|_{L^2} \leq (C_\mathcal{D} + 1) S_\mathcal{D}(\overline{\mathbf{u}}) + C_\mathcal{D} W_\mathcal{D}(\overline{\mathbf{u}}), \tag{C.17}$$

$$\|\nabla_\mathcal{D} u_\mathcal{D} - \nabla\overline{\mathbf{u}}\|_{L^2} \leq 2 S_\mathcal{D}(\overline{\mathbf{u}}) + W_\mathcal{D}(\nabla\overline{\mathbf{u}}) \tag{C.18}$$

If a triplet $(\mathcal{M}_m, \mathcal{E}_m, \mathcal{P}_m)_{m\in\mathbb{N}}$ is such that:

1. $(C_{\mathcal{D}_m})_m$ is bounded,

2. $\underset{m\to\infty}{lim} S_{\mathcal{D}_m}(\overline{\mathbf{u}}) = 0$,

3. $\underset{m\to\infty}{lim} W_{\mathcal{D}_m}(\nabla\overline{\mathbf{u}}) = 0$,

then, $\left\|\Pi_{\mathcal{D}_m} u_{\mathcal{D}} - \overline{\mathbf{u}}_m\right\|_{L^2} \to 0$ and $\left\|\nabla_{\mathcal{D}_m} u_{\mathcal{D}_m} - \nabla\overline{\mathbf{u}}\right\|_{L^2} \to 0$.

## C.1 Reminder on the Bramble-Hilbert's lemma

The following lines are similar to [19] Proposition (4.3.2) but to lighten the proof with $\mathbf{x}_K$ the center of the $B_K$ (see [19] for the general case). With the change of variable $\mathbf{z} = \mathbf{x}_K + s(\mathbf{y} - \mathbf{x}_K)$,

$$ds\ d\mathbf{z} = s^d ds\ d\mathbf{y}, \tag{C.19}$$

where $d$ is the dimension. Therefore, by definition of $Q_K$ (2.100),

$$|Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)| = |\frac{1}{|B_K|}\int_{B_K}\int_0^1 \mathbb{1}_A(\mathbf{z},s)s^{-d-1}D^2u(\mathbf{z};\mu)(\mathbf{x}_K - \mathbf{z})^2\ ds\ d\mathbf{y}|, \tag{C.20}$$

where $A = \{(\mathbf{z},s) : s \in (0,1],\ |(1/s)(\mathbf{z}-\mathbf{x}_K)| < \rho\}$ and $\mathbb{1}_A$ is the characteristic function of $A$.

$$|\int_0^1 \mathbb{1}_A(\mathbf{z},s)s^{-d-1}\Psi_K(\mathbf{x_K} + \frac{\mathbf{z}-\mathbf{x_K}}{\mathbf{s}})\ ds| \le \frac{1}{|B_K|}\|\Psi_K\|_{L^\infty(B_K)}\int_t^1 s^{-d-1}\ ds,$$

where $t = \frac{|\mathbf{z}-\mathbf{x}_K|}{\rho} < s$. Then,

$$\begin{aligned}|\int_0^1 \mathbb{1}_A(\mathbf{z},s)s^{-d-1}\Psi_K(\mathbf{x_K} + \frac{\mathbf{z}-\mathbf{x_K}}{\mathbf{s}})\ ds| &\le \frac{1}{|B_K|}\frac{1}{d}(1-t^{-d}),\\ &\le \frac{1}{|B_K|}\frac{1}{d}(t^{-d}-1),\\ &\le \frac{1}{|B_K|}\frac{1}{d}t^{-d},\\ &\le \frac{1}{|B_K|}\frac{1}{d}(\rho^d|\mathbf{z}-\mathbf{x}_K|^{-d}),\\ &\le C|\mathbf{z}-\mathbf{x}_K|^{-d} \text{ since } |B_K| \text{ is of order } \rho^d.\end{aligned} \tag{C.21}$$

From that equation (C.21) and (C.20), we deduce

$$|Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)| \le |\int_{B_K} D^2u(\mathbf{z};\mu)|\mathbf{z}-\mathbf{x}_K|^{2-d}\ d\mathbf{z}|. \tag{C.22}$$

Hence, since $B_K \subset K$,

$$|Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)| \le C\rho^{2-\frac{d}{2}}\left\|u(\mu)\right\|_{H^2(K)}. \tag{C.23}$$

Finally, by definition of $H$ and since $\rho \le H \le \theta\rho$ we end up with

$$|Q_K u(\mathbf{x}_K;\mu) - u(\mathbf{x}_K;\mu)| \le CH^{2-\frac{d}{2}}\left\|u(\mu)\right\|_{H^2(K)}. \tag{C.24}$$

# Appendix D

# On the minimization problem

We detail here how we end up with the linear problem (3.5) in chapter 3 from the minimization problem (3.2). This NIRB version (without constraints) is equivalent to the classical NIRB method 2.

We recall that we want to minimize the following problem

$$\min_{\alpha=(\alpha_1,\dots,\alpha_N)} \left\| \mathbf{u}_H - \sum_{i=1}^{N} \alpha_i \ \Phi_i^h \right\|_{L^2}. \tag{D.1}$$

Since the minimization does not depend on $\mathbf{u}_H$, it is equivalent to the problem

$$\min_{\alpha=(\alpha_1,\dots,\alpha_N)} J(\alpha) = -2\sum_{i=1}^{N} \alpha_i \ (\mathbf{u}_H, \Phi_i^h) + \sum_{i,j=1}^{N} \alpha_i \alpha_j \ (\Phi_i^h, \Phi_j^h), \tag{D.2}$$

since the bilinear form $(\cdot, \cdot)$ is symmetric. Then, we use a variation $p$ such that

$$\frac{d}{d\delta} J[\alpha + \delta p]_{|\delta=0} = 0. \tag{D.3}$$

From the definition of J (D.2),

$$\frac{d}{d\delta} J[\alpha + \delta p] = -2\sum_{j=1}^{N} (\alpha_j + \delta p)(\mathbf{u}_H, \Phi_j^h) + \sum_{i,j=1}^{N} (\alpha_i + \delta p)(\alpha_j + \delta p)(\Phi_i^h, \Phi_j^h). \tag{D.4}$$

Thus, from the equation (D.3), the minimization problem (D.2) is equivalent to

$$2p\sum_{j=1}^{N} [-(\mathbf{u}_H, \Phi_j^h) + \sum_{i=1}^{N} \alpha_i (\Phi_i^h, \Phi_j^h)] = 0. \tag{D.5}$$

Since the variation is arbitrary, we end up with the following problem (see (3.5))

$$-(\mathbf{u}_H, \Phi_j^h) + \sum_{i=1}^{N} \alpha_i (\Phi_i^h, \Phi_j^h) = 0, \quad \forall j = 1, \dots, N. \tag{D.6}$$

# Appendix E

# On the wind farm application

## E.1 Mass and momentum equations

### Mass equation

Let $\Omega_F$ be the fluid domain, such that the mass M of the fluid on this region is given by $M_{\Omega_F} = \int_{\Omega_F} \rho dV$ where $\rho$ is its density. The principle of mass conservation relates the fact that the mass of the fluid in a volume V is constant. Therefore, we have:

$$\frac{dM_{\Omega_F}}{dt} = \frac{d}{dt}\int_{\Omega_F} \rho dV = 0. \tag{E.1}$$

And with Leibnitz Theorem yields

$$\int_{\Omega_F} \frac{\partial \rho}{\partial t} dV + \int_{\partial\Omega_F} \mathbf{n}\ \mathbf{u}_S \rho dS = 0. \tag{E.2}$$

Indeed, Leibnitz theorem states

$$\frac{d}{dt}\int_{\Omega_F} f(\mathbf{x}, t) dV = \int_{\Omega_F} \frac{\partial f}{\partial t} dV + \int_{\partial\Omega_F} \mathbf{n}\ \mathbf{u}_S f dS,$$

where $\mathbf{u}_S$ is the velocity at the boundary. Then, the divergence formula gives

$$\int_{\Omega_F} [\frac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{u})] dV = 0.$$

The local form of the mass conservation can be applied on one point $M$ of the volume $\Omega_F$, which implies that:

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{u}) = 0, \tag{E.3}$$

which is the local form of the equation of continuity.
With the notion of particulate derivative, from $\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{u}\nabla.\rho$, we obtain:

$$\frac{D\rho}{Dt} + \rho \nabla.\mathbf{u} = 0.$$

**Momentum equation**

The rate of variation of a dynamic moment of a region is equal to the sum of the forces applied on this region. We distinguish two types of forces: The ones acting on the mass (**F**) and the forces of the surface (**R**). So the sum of these forces on this region is mathematically written: $\int_{\Omega_F} \rho \mathbf{F}\, dV + \int_{\partial\Omega_F} \mathbf{R}\, dS$. The moment by unity of volume is given by $\rho\mathbf{u}$ such that

$$\frac{d}{dt}\int_{\Omega_F} \rho\mathbf{u}\, dV = \int_{\Omega_F} \rho\mathbf{F}\, dV + \int_{\partial\Omega_F} \mathbf{R}\, dS.$$

By using again the Leibnitz theorem (E.1), we obtain:

$$\int_{\Omega_F} \frac{\partial}{\partial t}(\rho\mathbf{u})\, dV + \int_{\partial\Omega_F} \mathbf{nu}\rho\mathbf{u}\, dS = \int_{\Omega_F} \rho\mathbf{F}\, dV + \int_{\partial\Omega_F} \mathbf{R}\, dS.$$

And the divergence formula gives

$$\int_{\Omega_F} \frac{\partial}{\partial t}(\rho\mathbf{u})\, dV + \int_{\Omega_F} \nabla.(\mathbf{u}\rho\mathbf{u})\, dV = \int_{\Omega_F} \rho\mathbf{F}\, dV + \int_{\partial\Omega_F} \mathbf{R}\, dS. \tag{E.4}$$

The term **R** can be written with a tensor of constraints **T**, I. e. $\mathbf{R} = \mathbf{nT}$ (cf chap. 5.3 [105]).

The constraints on the surface **R** correspond to a force of pressure, and a term of viscosity. The tensor of constraints **T** can be written as a sum of the thermodynamic pressure $p$ and the tensor of viscosity $\boldsymbol{\tau}$, i.e.t $\mathbf{T} = -p\mathbf{I} + \boldsymbol{\tau}$.
By the moments formula (equation (E.4)), we have:

$$\int_{\Omega_F} \frac{\partial}{\partial t}(\rho\mathbf{u})\, dV + \int_{\Omega_F} \nabla.(\mathbf{u}\rho\mathbf{u})\, dV = \int_{\Omega_F} \rho\mathbf{F}\, dV + \int_{\partial\Omega_F} \mathbf{nT}\, dS. \tag{E.5}$$

Thus, again with the divergence formula,

$$\int_{\Omega_F} \frac{\partial}{\partial t}(\rho\mathbf{u})\, dV + \int_{\Omega_F} \nabla.(\mathbf{u}\rho\mathbf{u})\, dV = \int_{\Omega_F} \rho\mathbf{F}\, dV + \int_{\Omega_F} \nabla.\mathbf{T}\, dV. \tag{E.6}$$

Because $\Omega_F$ is an arbitrary region, the local formulation can be applied in one point and we get:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) \ + \nabla.(\rho\mathbf{uu}) \ = \rho\mathbf{F} + \nabla.\mathbf{T}.$$

With the decomposition of **T**, we end up with

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) \ + \nabla.(\rho\mathbf{uu}) \ = \rho\mathbf{F} - \nabla p + \nabla.\boldsymbol{\tau}. \tag{E.7}$$

## E.2 RANS equations

Let us enonce some reminds on the statistical Reynolds average for two variables $u$ and $v$:

$$\begin{aligned}
\overline{u+v} &= \overline{u} + \overline{v},\\
\overline{\overline{u}v} &= \overline{u}\overline{v},\\
\overline{\overline{u}} &= \overline{u},\\
\overline{\frac{\partial u}{\partial x_i}} &= \frac{\partial \overline{u}}{\partial x_i},\\
\overline{\frac{\partial u}{\partial t}} &= \frac{\partial \overline{u}}{\partial t}.
\end{aligned}$$

As said in the chapter 4, the RANS equations derived from the instant mass conservation and momentum equations E.1:

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0, \\ \frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} &= -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \end{aligned} \tag{E.8}$$

By the decomposition $u_i = \overline{u}_i + u'_i$, the equation (E.8) becomes:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial u'_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} + \frac{\partial \overline{u}_i u'_j}{\partial x_j} + \frac{\partial u'_i \overline{u}_j}{\partial x_j} + \frac{\partial u'_i u'_j}{\partial x_j} = -\frac{1}{\rho}\left(\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial p'}{\partial x_i}\right) + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u'_i}{\partial x_j} + \frac{\partial u'_j}{\partial x_i}\right), \tag{E.9}$$

and because $\overline{u'_i} = 0$ and $\overline{p'} = 0$, we end up with:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} + \frac{\partial \overline{u'_i u'_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right). \tag{E.10}$$

Here, we considered the averaged movement equation and not the fluctuation movement equation.

**$k - \varepsilon$ closure** The transport equation of the turbulent kinetic energy can be written

$$\frac{\partial}{\partial t}(\rho k) + \rho \overline{u}_i \frac{\partial k}{\partial x_i} + \rho \overline{u}_j \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right] + \frac{\partial}{\partial x_j}\left[\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_j}\right] + \rho[P_r - \varepsilon] + \overline{F}_k, \tag{E.11}$$

where $\overline{F}_k$ are additional forces applying on $k$, $P_r$ account for the energy production and is given by

$$P_r = \frac{\mu_t}{\rho}\left[\left(\frac{\partial \overline{u}_i}{x_j} + \frac{\overline{u}_j}{\partial x_i}\right)^2 + 2\left({\frac{\partial \overline{u}_i}{\partial x_i}}^2 + {\frac{\partial \overline{u}_j}{\partial x_j}}^2\right)\right]. \tag{E.12}$$

The tranport equation for the dissipation rate of the turbulent energy can be writte:

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \rho \overline{u}_i \frac{\partial \varepsilon}{\partial x_i} + \rho \overline{u}_j \frac{\partial \varepsilon}{\partial x_j} - \frac{\partial}{\partial x_i}\left[\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right] - \frac{\partial}{\partial x_j}\left[\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right] = \rho\left(C_{\varepsilon 1} P_r \frac{\varepsilon}{k} - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + C_{\varepsilon 3}\frac{P_r^2}{k}\right) + \overline{F}_\varepsilon, \tag{E.13}$$

where $\overline{F}_\varepsilon$ are additional forces applying on $\varepsilon$ and the constants $C_{\varepsilon 1}, C_{\varepsilon 2}, C_{\varepsilon 3}$ are defined from experimental data. The constants employed in code_saturne are given in the following table E.1.

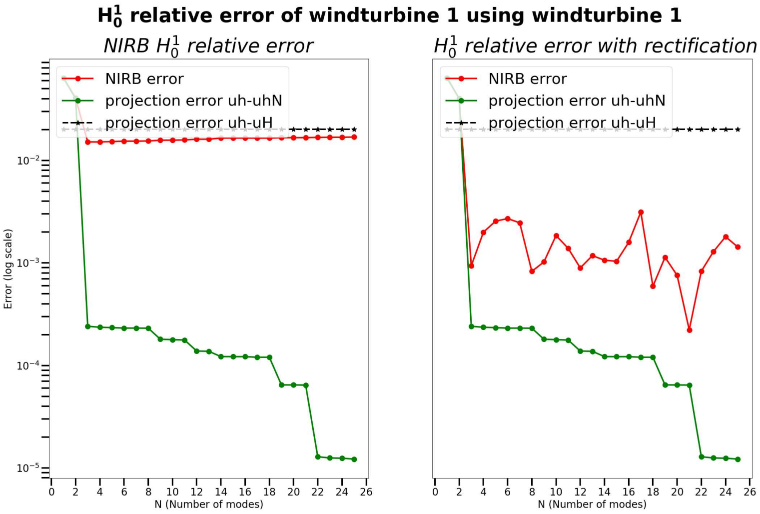
| $C_\mu$ | $C_{\varepsilon 1}$ | $C_{\varepsilon 2}$ | $C_{\varepsilon 3}$ | $\sigma_k$ | $\sigma_\varepsilon$ |
|---|---|---|---|---|---|
| 0.09 | 1.44 | 1.92 | 1 or 0 | 1 | 1.3 |

Table E.1: $k - \varepsilon$ parameters

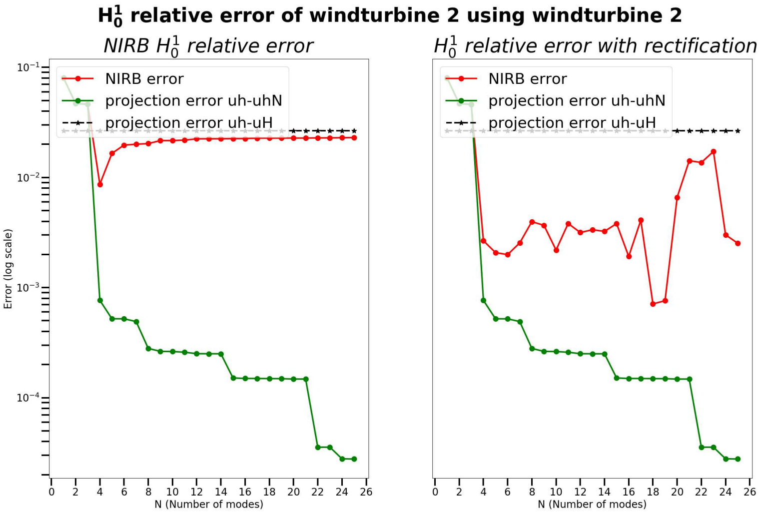
## E.3 More results on the 3-dimensional case

### E.3.1 3 turbines in line

Figure E.1 illustrates the NIRB results with and without rectification on the first and on the second turbine. We compare these results with the ones obtained on a truncated domain and with the method of section 3.3 in Figure E.1.

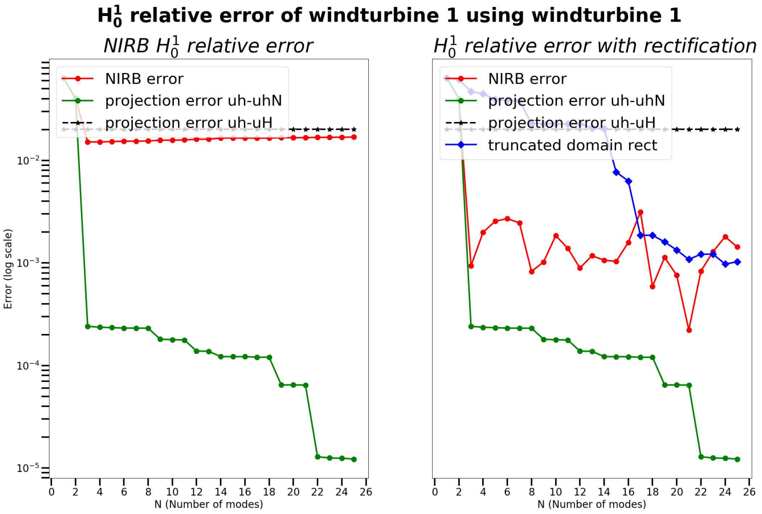


(a) First turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$, compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)

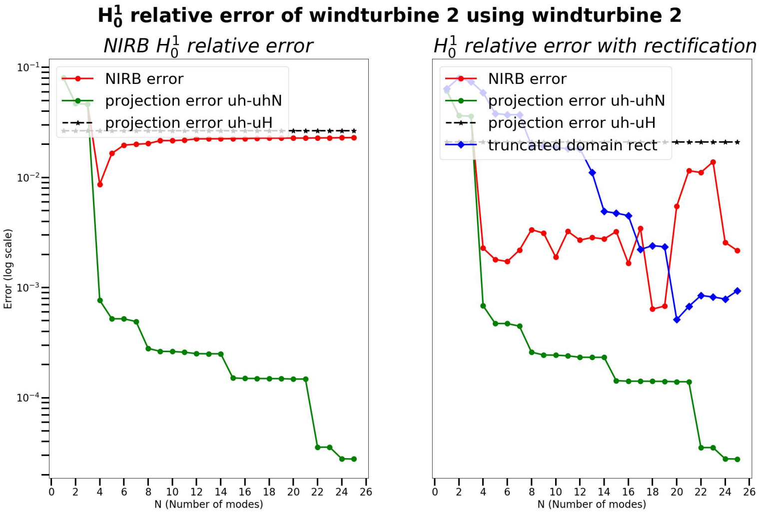


(b) Second turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$, compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)

Figure E.1: NIRB errors with (right) and without rectification (left)



(a) First turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$ compared to the truncated NIRB (blue) (obtained from a smaller domain) and compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)



(b) Second turbine, averaged relative $H_0^1$ error between the fine solution and the NIRB approximation (red curve) with (right) and without (left) rectification as a function of $N$ compared to the truncated NIRB (blue) (obtained from a smaller domain) and compared to the true projection $\mathbf{u}_{hh}^N$ (green) (vs coarse FV error)

Figure E.2: NIRB errors with (right) and without rectification (left) compared to truncated domain results (right)

# Appendix F

# GMSH mesh generator

We realized several meshes with GMSH, since Mordicus may read GMSH files (thanks to “meshio”). CAO OpenCascade allows us to easily mesh on GMSH.
The line DefineConstant is used to modify the graphical interface values (for instance of width and length). To define a point in GMSH, we write the following syntax
Point(indice) = {x,y,z,h}; It works the same for a line, and after all the lines creation, we can create a Line Loop whichs corresponds to the border thanks to the line indices (and then a surface with the Line Loop indices). The physical lines (or surfaces) are the labels that we want to attribute. For instance, here follows an exemple to construct the mesh domain of the model problem 3 (with and without OpenCascade).

```
DefineConstant[
        L={5,Min 3, Max 10, Step 1,Name "00Length",Visible 1},
        l={1,Min 1, Max 2, Step 0.5,Name "00Width",Visible 1}
];
h=0.2;

Point(1) = {0, 1/2, 0,h};
Point(2) = {l, 1/2, 0,h};
Point(3) = {l, 0, 0,h};
Point(4) = {L, 0, 0,h};
Point(5) = {L, 1, 0,h};
Point(6) = {0, 1, 0,h};

Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,5};
Line(5) = {5,6};
Line(6) = {6,1};

Line Loop(1) = {1,2,3,4,5,6};
Plane Surface (1) = {1};
Physical Surface (0)={1};
Physical Line(4)={6};
Physical Line(1)={1,2,3};
Physical Line(2)={4};
Physical Line(3)={5};
```

With OpenCascade (to create a rectangle, we use the syntax Rectangle(indice)={x,y,z,width,height} where (x,y,z) are the left down point coordinates and BooleanUnion allows us to concatenate surfaces)

```
SetFactory("OpenCASCADE"); // OpenCascade

DefineConstant[
        L={5,Min 3, Max 10, Step 1,Name "00Length",Visible 0},
        l={1,Min 1, Max 2, Step 0.5,Name "00Width",Visible 0}
];
h=0.2;
Mesh.CharacteristicLengthMin = h;
Mesh.CharacteristicLengthMax = h;

Rectangle(1)= {0,0.5,0,l,0.5};
Rectangle(2)= {l,0,0,L,1};
BooleanUnion{Surface{1};Delete;}{Surface{2};Delete;}

Physical Surface (0)={1,2};
Physical Line(4)={2};
Physical Line(1)={1,6,5};
Physical Line(2)={7};
Physical Line(3)={8};
```

# Bibliography

[1] A. Abdulle and Y. Bai. Adaptive reduced basis finite element heterogeneous multiscale method. *Computer Methods in Applied Mechanics and Engineering*, 257:203–220, 2013.

[2] R. A. Adams and J. JF. Fournier. *Sobolev spaces*. Elsevier, 2003.

[3] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition1. *Lecture Notes Pure and Applied Mathematics*, 216:317–332, 01 2001.

[4] N. Akkari. *Etude mathématique de la sensibilité POD (Proper orthogonal decomposition)*. PhD thesis, 2012.

[5] N. Akkari, A. Hamdouni, E. Liberge, and M. Jazar. On the sensitivity of the pod technique for a parameterized quasi-nonlinear parabolic equation. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1):1–16, 2014.

[6] S. Ali, F. Ballarin, and G. Rozza. Stabilized reduced basis methods for parametrized steady stokes and navier-stokes equations. *Computers & Mathematics with Applications*, 80(11):2399–2416, 2020.

[7] S. Ali, F. Ballarin, and G. Rozza. A reduced basis stabilization for the unsteady stokes and navier-stokes equations. *arXiv preprint arXiv:2103.03553*, 2021.

[8] F. Archambeau, N. Méchitoua, and M. Sakiz. Code saturne: A finite volume code for the computation of turbulent incompressible flows-industrial applications. *International Journal on Finite Volumes*, 1(1), 2004.

[9] J. P. P Argaud, B. Bouriquet, H. Gong, Y. Maday, and O. Mula. Stabilization of (g)eim in presence of measurement noise: application to nuclear reactor physics. In *International Conference on Spectral and High Order Methods for Partial Differential Equations*, Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016: Selected Papers from the ICOSAHOM conference, pages 133–145, Rio de Janeiro, Brazil, June 2017. Springer.

[10] M. Barrault, N.C. Nguyen, A.T. Patera, and Y. Maday. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes rendus de l'Académie des sciences. Série I, Mathématique*, 339-9:667–672, 2004.

[11] W. M. Batten, M.E. Harrison, and A.S. Bahaj. Accuracy of the actuator disc-rans approach for predicting the performance and wake of tidal turbines. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1985):20120293, 2013.

[12] M. M. Baumann. *Nonlinear Model Order Reduction using POD/DEIM for Optimal Control of Burgers' Equation*. PhD thesis, 07 2013.

[13] F. B. Belgacem and S. C Brenner. Some nonstandard finite element estimates with applications to 3d poisson and signorini problems. *Electronic Transactions on Numerical Analysis*, 12:134–148, 2001.

[14] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[15] A. Bērziņš, J. Helmig, F. Key, and S. Elgeti. Standardized non-intrusive reduced order modeling using different regression models with application to complex flow problems. *arXiv preprint arXiv:2006.13706*, 2020.

[16] H. Blum and M. Dobrowolski. On finite element methods for elliptic equations on domains with corners. *Computing*, 28(1):53–63, 1982.

[17] F. Boyer. An introduction to finite volume methods for diffusion problems. November 25-29 2013.

[18] D. Braess and J. Schöberl. Equilibrated residual error estimator for edge elements. *Mathematics of Computation*, 77(262):651–672, 2008.

[19] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15. Texts in Applied Mathematics. Springer, New York, third edition, 2008.

[20] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15. Springer Science & Business Media, 2012.

[21] F. Brezzi, K. Lipnikov, and M. Shashkov. Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM Journal on Numerical Analysis*, 43(5):1872–1896, 2005.

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

[23] P. Burda, J. Novotnỳ, and B. Sousedík. A posteriori error estimates applied to flow in a channel with corners. *Mathematics and Computers in Simulation*, 61(3-6):375–383, 2003.

[24] M. Carmona Flores, J. M. Gómez Cama, J. Bosch Estrada, and M. López de Miguel. Gmsh-guide to mesh generation. 2018.

[25] A. Carpinteri. Stress-singularity and generalized fracture toughness at the vertex of re-entrant corners. *Engineering Fracture Mechanics*, 26(1):143–155, 1987.

[26] F. Casenave, A. Ern, and T. Lelièvre. Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(1):207–229, 2014. 27pages, 11 figures.

[27] F. Casenave, A. Ern, and T. Lelièvre. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Advances in Computational Mathematics*, 41(5):961–986, Jun 2014.

[28] R. Chakir. *Contribution à l'analyse numérique de quelques problèmes en chimie quantique et mécanique*. PhD thesis, 2009. Doctoral dissertation, Université Pierre et Marie Curie-Paris VI.

[29] R. Chakir, P. Joly, Y. Maday, and P. Parnaudeau. A non intrusive reduced basis method: application to computational fluid dynamics. 2013, September.

[30] R. Chakir, Y. Maday, and P. Parnaudeau. A non-intrusive reduced basis approach for parametrized heat transfer problems. *Journal of Computational Physics*, 376:617–633, January 2019.

[31] J.S. Chen, C. Marodon, and H.Y. Hu. Model order reduction for meshfree solution of poisson singularity problems. *International Journal for Numerical Methods in Engineering*, 102(5):1211–1237, 2015.

[32] Y-S Chen and S-W Kim. Computation of turbulent flows using an extended k-epsilon turbulence closure model. *NASA STI/Recon Tech Rep N*, 88, 11 1987.

[33] A. J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[34] A. Cohen and R. DeVore. Approximation of high-dimensional parametric pdes. 24:1–159, 2015.

[35] P. G. Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. Society for Industrial and Applied Mathematics, 2015.

[36] S. Corciulo. *Dynamic hydro-mechanical analysis of soil-monopile interaction in Offshore Wind Turbines*. PhD thesis, 04 2016. Politecnico Di Milano.

[37] L. Cordier and M. Bergmann. Réduction de dynamique par décomposition orthogonale aux valeurs propres (pod). Ecole de printemps OCET,Optimisation et Contrôle des Ecoulements et des Transferts, 2006.

[38] L. B. da Veiga, Lipnikov K, and Manzini G. *The mimetic finite difference method for elliptic problems*, volume 11. Springer, 2014.

[39] J.P. De S.R. Gago, D. W. Kelly, O. C. Zienkiewicz, and I Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part ii—adaptive mesh refinement. *International journal for numerical methods in engineering*, 19(11):1621–1656, 1983.

[40] N. Demo, M. Tezzele, and G. Rozza. A non-intrusive approach for the reconstruction of pod modal coefficients through active subspaces. *Comptes Rendus Mécanique*, 347(11):873–881, 2019. Data-Based Engineering Science and Technology.

[41] D. A. Di Pietro, A. Ern, and S. Lemaire. An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators. *Computational Methods in Applied Mathematics*, 14(4):461–472, 2014.

[42] J. Droniou. Finite volume schemes for diffusion equations: Introduction to and review of modern methods. 24 (8):1575–1619, 2014.

[43] J. Droniou and R. Eymard. A mixed finite volume scheme for anisotropic diffusion problems on any grid. 105(1):35–71, 2006.

[44] J. Droniou, R. Eymard, T. Gallouet, and R. Herbin. Gradient schemes: a generic framework for the discretisation of linear, nonlinear and nonlocal elliptic and parabolic equations. *Mathematical Models and Methods in Applied Sciences*, 23(13):2395–2432, 2013.

[45] J. Droniou, R. Eymard, T. Gallouët, C. Guichard, and R. Herbin. *The Gradient Discretisation Method*, volume 82. Springer, 2018.

[46] J. Droniou, R. Eymard, T. Gallouët, and R. Herbin. A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods. *Mathematical Models and Methods in Applied Sciences*, 20(02):265–295, 2010.

[47] J. Droniou and N. Nataraj. Improved $l^2$ estimate for gradient schemes and superconvergence of the tpfa finite volume scheme. 3:1254–1293, 2017.

[48] F. Durst and C. Tropea. Flows over two-dimensional backward—facing steps. In *Structure of Complex Turbulent Shear Flow*, pages 41–52. Springer, 1983.

[49] A. Ern and M. Vohralík. Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous galerkin, and mixed discretizations. *Journal on Numerical Analysis*, 53(2):1058–1081, 2015.

[50] E. Erturk. Numerical solutions of 2-d steady incompressible flow over a backward-facing step, part i: High reynolds number solutions. *Computers & Fluids*, 37(6):633–655, 2008.

[51] L. C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010.

[52] R. Eymard, T. Gallouët, and R. Herbin. Discretization schemes for linear diffusion operators on general non-conforming meshes. *In R. Eymard and J.-M. Hérard, editors, Finite Volumes for Complex Applications V*, pages 375–382, 2008.

[53] D. Fan, M. Sebès, A. Qazi, C. Pozzolini, and E. Bourgeois. Study of variability in dynamic simulations of turnouts using the pgd method. *Vehicle System Dynamics*, 0(0):1–18, 2021.

[54] L. Fick, Y. Maday, A. Patera, and T. Taddei. A reduced basis technique for long-time unsteady turbulent flows. *arXiv preprint arXiv:1710.03569*, page 59, 10 2017.

[55] L. C. Foucard and F. J. Vernerey. An x-fem-based numerical–asymptotic expansion for simulating a stokes flow near a sharp corner. *International Journal for Numerical Methods in Engineering*, 102(2):79–98, 2015.

[56] F. Fritzen, B. Haasdonk, D. Ryckelynck, and S. Schöps. An algorithmic comparison of the hyper-reduction and the discrete empirical interpolation method for a nonlinear thermal problem. *Mathematical and Computational Applications*, 23(1):8, 10 2016.

[57] M. Gadalla, M. Cianferra, M. Tezzele, G. Stabile, A. Mola, and G. Rozza. On the comparison of les data-driven reduced order approaches for hydroacoustic analysis. *Computers & Fluids*, 216:21, 2021.

[58] P. Gallinari, Y. Maday, M. Sangnier, O. Schwander, and T. Taddei. Reduced basis' acquisition by a learning process for rapid on-line approximation of solution to pde's: Laminar flow past a backstep. *Archives of Computational Methods in Engineering*, 25(1):131–141, 2018.

[59] L. Gay, B. Sapa, and F. Nmira. Magic and code_saturne developments and simulations for mechanically ventilated compartment fires. *Fire safety journal*, 62:161–173, 2013.

[60] A. Ghisi, S. Mariani, A. Corigliano, and S. Zerbini. Physically-based reduced order modelling of a uni-axial polysilicon mems accelerometer. *Sensors*, 12(10):13985–14003, 2012.

[61] G.H. Golub and C.F. Van Loan. Matrix computations. 1990.

[62] S. Gregori, M. Tur, E. Nadal, J.V. Aguado, F.J. Fuenmayor, and F. Chinesta. Fast simulation of the pantograph–catenary dynamic interaction. *Finite Elements in Analysis and Design*, 129:1–13, 2017.

[63] S. Gregori, M. Tur, E. Nadal, F.J. Fuenmayor, and F. Chinesta. Parametric model for the simulation of the railway catenary system static equilibrium problem. *Finite Elements in Analysis and Design*, 115:21–32, 2016.

[64] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.

[65] P.M. Gresho, D.K. Gartling, J.R. Torczynski, K.A. Cliffe, K.H. Winters, T.J. Garratt, A. Spence, and J. W Goodrich. Is the steady viscous incompressible two-dimensional flow over a backward-facing step at re= 800 stable? *International Journal for Numerical Methods in Fluids*, 17(6):501–541, 1993.

[66] E. Grosjean and Y. Maday. Error estimate of the non-intrusive reduced basis method with finite volume schemes. *ESAIM: M2AN*, 55(5):1941–1961, 2021.

[67] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 42(2):277–302, 2008.

[68] J. K. Hammond. *Méthodes des bases réduites pour la modélisation de la qualité de l'air urbaine*. PhD thesis, Paris Est, 2017. 2017PESC1230.

[69] J. K. Hammond, R. Chakir, F. Bourquin, and Y. Maday. Pbdw: A non-intrusive reduced basis data assimilation method and its application to an urban dispersion modeling framework. *Applied Mathematical Modelling*, 76:1–25, 2019.

[70] F. Hecht. The mesh adapting software: bamg, 1998.

[71] F. Hecht, O. Pironneau, A. Le Hyaric, and K. Ohtsuka. Freefem++ manual, 2005.

[72] H. Herrero, Y. Maday, and F. Pla. Rb (reduced basis) for rb (rayleigh–bénard). *Computer Methods in Applied Mechanics and Engineering*, 261:132–141, 2013.

[73] J. S Hesthaven, G. Rozza, B. Stamm, and al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[74] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge university press, 1996.

[75] M. Horák, D. Ryckelynck, and S. Forest. Hyper-reduction of generalized continua. *Computational Mechanics*, 59(5):753–778, 2017.

[76] M. A. Hossain, Md T. Rahman, and S. Ridwan. Numerical investigation of fluid flow through a 2d backward facing step channel. *International Journal of Engineering Research & Technology*, 2(10):3700–3708, 2013.

[77] D. B.P. Huynh, D. J. Knezevic, and A. T. Patera. A static condensation reduced basis element method : approximation and a posteriori error estimation. *ESAIM: M2AN*, 47(1):213–251, 2013.

[78] D.B.P. Huynh, D.J. Knezevic, and A.T. Patera. A static condensation reduced basis element method: Complex problems. *Computer Methods in Applied Mechanics and Engineering*, 259:197–216, 2013.

[79] H. Igarashi and T. Honma. A boundary element method for potential fields with corner singularities. *Applied Mathematical Modelling*, 20(11):847–852, 1996.

[80] O. Iliev, Y. Maday, and T. Nagapetyan. *A Two-grid Infinite-volume/reduced Basis Scheme for the Approximation of the Solution of Parameter Dependent PDE with Applications the AFFFF Devices*. Citeseer, 2013.

[81] D.S. Jang, R. Jetli, and S. Acharya. Comparison of the piso, simpler, and simplec algorithms for the treatment of the pressure-velocity coupling in steady flow problems. *Numerical Heat Transfer, Part A: Applications*, 10(3):209–228, 1986.

[82] J. Jiang, Y. Chen, and A. Narayan. Offline-enhanced reduced basis method through adaptive construction of the surrogate parameter domain. *Journal of Scientific Computing*, 73:853–875, 12 2017.

[83] I.T. Jolliffe. Principal component analysis. *Technometrics*, 45(3):276, 2003.

[84] S. Kim and H.C. Lee. A finite element method for computing accurate solutions for poisson equations with corner singularities using the stress intensity factor. *Computers and Mathematics with Applications*, 71(11):2330–2337, 2016.

[85] A. Kolmogoroff. Über die beste annäherung von funktionen einer gegebenen funktionen. 37:107–110, 1963.

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

[87] P. Ladevèze. *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*. Springer Science & Business Media, 2012.

[88] L. Laporte. *Application d'un code de CFD atmosphérique à l'estimation du productible éolien en terrain complexe*. Theses, Université Paris-Est, December 2008.

[89] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Pergamon, 1983.

[90] T. Lee and D. Mateescu. Experimental and numerical investigation of 2-d backward-facing step flow. *Journal of Fluids and Structures*, 12(6):703–716, 1998.

[91] B. Lorendeau, Y. Fournier, and A. Ribes. In-situ visualization in fluid mechanics using catalyst: A case study for code saturne. In *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, pages 53–57. IEEE, October 2013.

[92] P. Louda, J. Příhoda, K. Kozel, and P. Sváček. Numerical simulation of flows over 2d and 3d backward-facing inclined steps. *International Journal of Heat and Fluid Flow*, 43:268–276, 2013. 7th International Symposium on Turbulence Heat and Mass Transfer (THMT-7), Palermo Conference on Modelling Fluid Flow (CMFF'12).

[93] Z. Luo, J. Chen, I.M. Navon, and X. Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary navier–stokes equations. *SIAM Journal on Numerical Analysis*, 47(1):1–19, 2009.

[94] L. Ma. Superconvergence of discontinuous galerkin methods for elliptic boundary value problems, 2021.

[95] Y. Maday, T. Anthony, J. D. Penn, and M. Yano. Pbdw state estimation: Noisy observations; configuration-adaptive background spaces; physical interpretations. *ESAIM: Proceedings and Surveys*, 50:144–168, 2015.

[96] Y. Maday and R. Chakir. A two-grid finite-element/reduced basis scheme for the approximation of the solution of parametric dependent p.d.e. *Comptes Rendus Mathematique*, 347:p. 435–440, 2009.

[97] G. Mercier. *Modélisation de parcs d'hydroliennes à flux transverse avec une méthode d'équivalence*. PhD thesis, Université de Grenoble, 2014.

[98] R. Mikkelsen. *Actuator disc methods applied to wind turbines*. PhD thesis, Technical University of Denmark, 2003.

[99] O. Mula Hernandez. *Quelques contributions vers la simulation parallèle de la cinétique neutronique et la prise en compte de données observées en temps réel*. PhD thesis, 2014. Thèse de doctorat dirigée par Maday, Yvon Mathématiques appliquées Paris 6 2014.

[100] N. E Murray and L. S Ukeiley. An application of gappy pod. *Experiments in Fluids*, 42(1):79–91, 2007.

[101] K. K. Nagarajan. *Analysis and control of self-sustained instabilities in a cavity using reduced order modelling*. PhD thesis, 2010.

[102] H.D. Nedjari, O. Guerri, and M. Saighi. Full rotor modelling and generalized actuator disc for wind turbine wake investigation. *Energy Reports*, 6:232–255, 2020. Technologies and Materials for Renewable Energy, Environment and Sustainability.

[103] J. A. Nitsche and A. H. Schatz. Interior estimates for ritz-galerkin methods. *Mathematics of Computation*, 28(128):937–958, 1974.

[104] B. N. Owens. *The Wind Power Story: A Century of Innovation That Reshaped the Global Energy Landscape*. Hoboken, John Wiley & Sons Inc/IEEE Press, 2019.

[105] Ronald L Panton. *Incompressible flow*. John Wiley & Sons, 2013.

[106] J. Papež, U. Rüde, M. Vohralík, and B. Wohlmuth. Sharp algebraic and total a posteriori error bounds for h and p finite elements via a multilevel approach. *hAL Preprint*, 1662944, 2017.

[107] K. Pearson. On lines and planes of closest fir to systems of points in space. 2:559–572, 1901.

[108] W. Prager and J. L. Synge. Approximations in elasticity based on the concept of function space. *Quarterly of Applied Mathematics*, 5(3):241–269, 1947.

[109] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. 124(1):70–80, 2002.

[110] A. Quarteroni, A. Manzoni, and F. Negri. Reduced basis methods for partial differential equations: an introduction. *Springer*, 92, 2015.

[111] A. Quarteroni and S. Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.

[112] D. Rabinovich, D. Givoli, and S. Vigdergauz. Xfem-based crack detection scheme using a genetic algorithm. *International Journal for Numerical Methods in Engineering*, 71(9):1051–1080, 2007.

[113] D. Rajaram, T. G. Puranik, C. Perron, and D. N. Mavris. Non-intrusive parametric reduced order modeling using randomized algorithms. In *AIAA Scitech 2020 Forum*, page p. 23, 01 2020.

[114] M. Randrianarivony. Anisotropic finite elements for the stokes problem: a posteriori error estimator and adaptive mesh. *Journal of computational and applied mathematics*, 169(2):255–275, 2004.

[115] S. I. Repin. *A Posteriori Estimates for Partial Differential Equations*. De Gruyter, 2008.

[116] F. Riesz and B. Nagy. Functional analysis. 1955.

[117] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *Journal of computational physics*, 202(1):346–366, 2005.

[118] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1):75–89, 2009.

[119] D. Ryckelynck, L. Gallimard, and S. Jules. Estimation of the validity domain of hyper-reduction approximations in generalized standard elastoviscoplasticity. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–19, 2015.

[120] Brenner C. S. Multigrid methods and stress intensity factors i: Corner singularities. *Mathematics of Computation*, 68(165):559–583, 1999.

[121] R. Sanchez. *Application des techniques de bases réduites à la simulation des écoulements en milieux poreux*. PhD thesis, 2017.

[122] W. J. Schroeder, L. S. Avila, and W. Hoffman. Visualizing with vtk: a tutorial. *IEEE Computer graphics and applications*, 20(5):20–27, 2000.

[123] S. Sen, K. Veroy, D.B.P. Huynh, S. Deparis, N.C. Nguyen, and A.T. Patera. "natural norm" a posteriori error estimators for reduced basis approximations. *Journal of Computational Physics*, 217(1):37–62, 2006. Uncertainty Quantification in Simulation Science.

[124] Z. Shang. Performance analysis of large scale parallel cfd computing based on code_saturne. *Computer Physics Communications*, 184(2):381–386, 2013.

[125] L Sirovich. Turbulence and the dynamics of coherent structures, parts i-iii. XLV(3):561–571, 1987.

[126] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza. Pod-galerkin reduced order methods for cfd using finite volume discretisation: vortex shedding around a circular cylinder. *Communications in Applied and Industrial Mathematics*, 8(1):210–236, 2017.

[127] B. Streichenberger. *Approches multi-fidélités pour la simulation rapide d'écoulements d'air en milieu urbain*. PhD thesis, 2021. Thèse de doctorat dirigée par Chakir, Rachida, Université Gustave Eiffel.

[128] J. Sumner, G. España, C. Masson, and S. Aubrun. Evaluation of rans/actuator disk modelling of wind turbine wake flow using wind tunnel measurements. *International Journal of Engineering Systems Modelling and Simulation* 47, 5:147 – 158, 01 2013.

[129] Vidar T. *Galerkin finite element methods for parabolic problems*, volume 25. Springer Science & Business Media, 2007.

[130] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[131] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1-3):67–83, 1994.

[132] K. Veroy, C. Prud'homme, and A. Patera. Reduced-basis approximation of the viscous burgers equation: Rigorous a posteriori error bounds. *I*, 337:619–624, 11 2003.

[133] L. B. Wahlbin. On the sharpness of certain local estimates for $\overset{\circ}{H}^1$ projections into finite element spaces: Influence of a reentrant corner. *Mathematics of Computation*, 42(165):1–8, 1984.

[134] Lars B. Wahlbin. Local behavior in finite element methods. pages 353–522, 1991.

[135] P. Xu. Truncated svd methods for discrete linear ill-posed problems. *Geophysical Journal International*, 135(2):505–514, 1998.

[136] X. Zou, M. Conti, P. Díez, and F. Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2):230–251, 2018.

# List of Figures

# List of Tables

Elise Grosjean - Thèse de doctorat - 2021

# Résumé

Le but de cette thèse est l'analyse et le développement d'outils numériques à faible coûts. La première partie présente les méthodes de Bases Réduites (MBR) étudiée pendant cette thèse. Nous avons contribué à l'élaboration d'une bibliothèque open-source en Python sur les MBR et nous présentons plusieurs résultats numériques avec différentes MBR sur un problème modèle en l'utilisant. En particulier, nous introduisons la méthode deux grilles. Comme son nom l'indique, elle utilise deux maillages: un fin et un grossier. Le but est de trouver une bonne approximation d'une solution d'un problème paramétrique, aussi proche de la solution exacte que si nous avions utilisé un code haute-fidélité avec une méthode classique, par exemple avec la Méthode des Eléments Finis (MEF) tout en réduisant considérablement les degrés de liberté. Cela induit donc une réduction des temps de calculs. Après son analyse pour les équations elliptiques dans le contexte MEF, nous présentons plusieurs analyses supplémentaires de la méthode deux grilles comme par exemple son adaptation aux équations paraboliques. Nous appliquons ensuite la méthode à des domaines contenant des singularités. Par la suite, nous réalisons une analyse détaillée de la méthode dans le contexte des volumes finis. La troisième partie de cette thèse porte sur le développement d'outils non intrusifs. Nous présentons deux nouvelles méthodes et les appliquons sur un problème 2D classique d'un écoulement de Navier-Stokes stationnaire. Un des outils permet de considérer des domaines tronqués et ainsi de réduire drastiquement les temps de calculs des simulations. La dernière partie de cette thèse concerne l'application de la méthode non-intrusive de bases réduites à la simulation d'un champ d'éoliennes offshores. Cette partie est une collaboration avec EDF et le but est de tester la méthode deux grilles sur des problèmes paramétriques hautement complexes. Les champs d'éoliennes sont modélisés à l'aide des équations de Navier-Stokes en moyenne de Reynolds. Pour les résoudre, nous utilisons un solveur d'EDF appelé code_saturne. Nous présentons plusieurs résultats numériques en 2D et 3D.
Mots-clés: Méthode de bases réduites, modèle d'ordre réduit, modélisation d'éoliennes offshore, Méthode des volumes finis, méthodes des éléments finis.

# Abstract

The purpose of this thesis is the analysis and the development of low-cost numerical tools. The first chapter is a review on Non-Intrusive Reduced Basis methods (NIRB) studied during this thesis. We contributed to the elaboration of an open library on NIRB methods in Python. We present numerical results with several NIRB on a model problem using this library. In particular, we introduce the two-grid method which is analyzed in the second chapter. As its name indicates, the method uses two grids: one fine mesh and one coarse mesh. Its aims to recover an accurate approximation of the solution of a parameterized problem as if we had used a high-fidelity code for instance with the Finite Element Method (FEM) while significantly reducing the degrees of freedom. Thus, it reduces the complexity. In the second chapter, in addition to its detailed numerical analysis in the context of FEM with elliptic equations, we proceed with several further analyses such as its adaptation to parabolic problems. We then apply the method to domain with singularities. Subsequently, we extend the method to finite volume solvers, and detail its numerical analysis in this context. The third chapter is concerned with the development of new NIRB methods. We present two new approaches and we test them on a model problem which is the classical 2D backward-facing step. One tool allows us to consider truncated domains and to further reduce the runtimes. The last part of this thesis is about an application of the two-grid method on offshore wind farm simulations. This part is a collaboration with EDF and the purpose is to test the two-grid method on highly complex problems. A solver of EDF, named code_saturne, is used to solve the Reynolds Averaged Navier-Stokes (RANS) equations representing the wind around the turbines. We present several numerical results in 2D and 3D.
Key Words: reduced basis method, model order reduction, offshore wind farms modeling, finite volume schemes, finite element method.