

# A NON-INTRUSIVE REDUCED BASIS METHOD FOR URBAN FLOWS SIMULATION

RACHIDA CHAKIR<sup>1</sup>, BENJAMIN STREICHENBERGER<sup>2</sup> AND PATRICE CHATELLIER<sup>3</sup>

<sup>1</sup> Université Gustave Eiffel, IFSTTAR, Marne-la-Vallée, France  
Efficacity, Marne-la-Vallée, France  
rachida.chakir@univ-eiffel.fr

<sup>2</sup> Université Gustave Eiffel, IFSTTAR, Marne-la-Vallée, France  
benjamin.streichenberger@univ-eiffel.fr

<sup>3</sup> Université Gustave Eiffel, IFSTTAR, Marne-la-Vallée, France  
patrice.chatellier@univ-eiffel.fr

**Key words:** Computational Fluid Dynamics (CFD), Reduced Basis Methods, Non-intrusive approach

**Abstract.** In this work, we present a non-intrusive method using the Reduced Basis framework in order to diminish the cost of numerical simulation arising from the computation of parameters-dependent Partial Differential Equations (PDE). This method involves the computation of less expensive (but less accurate) solutions of the PDE during the online stage, and a RB-based rectification step. It represents a good substitute for standard Reduced Basis methods when it is applied to urban flows modelling. This approach speeds up the CFD simulation while remaining non-intrusive in relation to the high fidelity model, which can allow to avoid practical problems (*e.g.* non-affine parametric dependence) associated to model reduction for complex air flows involved in many sophisticated methods of urban air quality modeling. Our focus here is on the validation of the non-intrusive method applied to the backward-facing step 2D benchmark.

## 1 INTRODUCTION

As the population increases, cities must constantly reassess their urban planning. However, this must be done in such a way to preserve the quality of life of its inhabitants. In this context, the study of the different urban flows (pollution, heat) is very important and the effectiveness of simulation tools is essential. Advanced physically based models (such as CFD models) could provide spatially rich small-scale solution, however the use of such models is challenging due to explosive computational times in real-world applications.

Reduced Order Models methods can remedy this problem by decreasing drastically the full-scale model complexity without important loss of information. We are interested in particular in Reduced Basis (RB) methods [1, 2, 3], which allow for a parametrized PDE to construct approximation spaces generated by a few particular solutions of the PDE. Works on reduced basis methods applied to CFD models can be found in [4, 5, 6], but many improvements can still happen in this specific field. One reason is that CFD

models, or urban flow models in general, are often very complex (geometry, non-linearity, non-affine parametric dependence), which makes it difficult to efficiently implement RB methods in these cases. Besides, the intrusive nature of RB methods can be problematic when one does not have access to the simulation code that solves the physical model (industrial codes for example).

For these two main reasons, the Non-Intrusive Reduced Basis (NIRB) method [7, 8, 9] is a good alternative to RB methods in urban CFD applications. We propose in this work to apply the NIRB method to a 2D CFD simulation of the airflow along a backward-facing step, which represents the air recirculation that occurs behind a building. This will validate notably the extension of the NIRB method to this kind of full-scale urban CFD applications.

The paper is organized as follows. In section 2, first we introduce the main ideas of RB methods and the motivations for a non-intrusive procedure; then we explain the NIRB method. Section 3 presents the 2D backward step benchmark chosen, namely the parametric study, the CFD model and a mesh convergence analysis. Finally, in section 4, numerical results and perspectives are given.

## 2 NON-INTRUSIVE REDUCED BASIS METHOD

### 2.1 Context and motivations

To represent the physical phenomena that we are interested in, we consider a parametrized PDE under its following variational formulation : for a given  $s \in \mathcal{D} \subset \mathbb{R}^{N_s}$ , find  $w(s) \in X$  such that

$$a(w(s), v; s) = f(v; s), \quad \forall v \in X \quad (1)$$

$\mathcal{D}$  is the parameter space and  $s$  is a set of  $N_s$  parameters.  $X = X(\Omega)$  is an appropriate functional space, where  $\Omega \subset \mathbb{R}^d$  is the computational domain ( $d = 2$  or  $3$ ).  $a(\cdot, \cdot; s)$  and  $f(\cdot)$  are respectively continuous and bounded bilinear and linear forms, and  $a(\cdot, \cdot; s)$  is assumed to satisfy the inf-sup condition [1, 2].

We denote by  $w^{\text{true}}(s)$  the truth solution of (1). Let  $X_h$  be a discretization space of finite dimension (e.g. a finite element space) in which we seek for a solution of (1), noted  $w_h(s)$ . We suppose that  $X_h$  is a high-fidelity approximation of  $X$ , which means that the dimension  $N_h$  of  $X_h$  is quite large so that  $w_h(s)$  is close enough to  $w^{\text{true}}(s)$ . The high-fidelity problem is formulated as follows : for a given  $s \in \mathcal{D}$ , find  $w_h(s) \in X_h$  such that

$$a(w_h(s), v_h; s) = f(v_h; s), \quad \forall v_h \in X_h \quad (2)$$

If the dimension  $N_h$  is high (and it generally is), solving problem (2) for several values of  $s$  may imply a great computational cost. This is where RB methods come in. RB methods can give a real-time and reliable prediction of the solution  $w_h(s)$  using a low-dimensional approximation space. More generally, they allow to reduce a wide range of physical problems in a significant way and without an important loss of information. The idea is that, when the manifold of solutions of (2)  $\mathcal{M}_h = \{w_h(s) \mid s \in \mathcal{D}\}$  can be approximated by a subspace of dimension  $N \ll N_h$ , we can look for an approximation of  $w_h(s)$  in this new subspace and therefore drastically reduce the computational cost.

We suppose that the manifold  $\mathcal{M}_h$  is regular enough and of low dimension  $N$ . Let first introduce a sample of the parameter space  $S_N = \{s^1, \dots, s^N\} \subset \mathcal{D}$ . The reduced basis approximation space is then  $X_N = \text{span}(w_h(s^k), 1 \leq k \leq N)$ , and the high-fidelity solution  $w_h(s)$  can be replaced by a linear combination of the snapshots  $w_h(s^k), 1 \leq k \leq N$ . One apply a Galerkin projection to get the reduced problem of (2) in its variational form : for a given  $s \in \mathcal{D}$ , find  $w_N(s) \in X_N$  such that

$$a(w_N(s), v_N; s) = f(v_N; s), \quad \forall v_N \in X_N \quad (3)$$

For reason of stability of problem (3), it is important to orthonormalize the snapshots in order to generate another basis of  $X_N$ , called reduced basis and noted  $\{\zeta_1, \dots, \zeta_N\}$ . One immediate question is the way to construct the RB space. Optimal or standard RB methods are based on a greedy algorithm : after initialization, the reduced basis is completed at every step  $n \leq N$  with the solution  $w_h(s)$  which is the least well represented in the space  $X_n$ . Other non-optimal methods to generate the snapshots exist, in particular the "Proper Orthogonal Decomposition" method [1, 10], but it requires the pre-calculation of a full training set  $\mathcal{A}_h = \{w_h(s^1), \dots, w_h(s^{n_{\text{train}}})\}$ .

RB methods are separated into two stages : the *offline* and the *online* stage. In the *offline* stage, one compute and save the reduced basis functions and all the parameter-independent elements (matrices, vectors). This stage is expensive but performed only once. The *online* stage is performed for each new value of  $s$ . One compute the remaining parameters-dependent quantities that are necessary to the implementation of the reduced problem (3), and the latter is solved to find the approximation  $w_N(s)$ . The key of the *online* stage is that all the computations have to be independent of the high dimension  $N_h$ . Otherwise, the reduced basis method is not used to its full extent. If the forms  $a(\cdot, \cdot; s)$  and  $f(\cdot)$  satisfy the property of affine parameter dependance (apd) [1, 3, 11], it is possible to separate all the parameter-dependent quantities from the dimension  $N_h$ , and thus the *online* stage is optimal. If the property of apd is not verified, one can use the Empirical Interpolation Method (EIM) [12, 13, 14] and still be able to implement the method in an optimal way. However, both apd and EIM are very intrusive techniques, because they require to modify the simulation code that computes the high-fidelity solutions. It can be problematic when one does not have access to the code, which can often happen with industrial collaborations.

This is why we are interested in a less intrusive process, namely the NIRB method. The latter offers an alternative to standard RB method. We explain how it works in the following section.

## 2.2 Explication of the NIRB method

The NIRB method is based first on the generation of a reduced basis  $V_N$  composed of  $N$  particular high-fidelity solutions (as the previous section describes), and secondly on the computation of high-fidelity solutions  $w_h(s)$  and low-fidelity solutions  $w_H(s)$ , with  $s$  in a pre-chosen subspace  $\Xi_{\text{train}} = \{s^1, \dots, s^{n_{\text{train}}}\} \subset \mathcal{D}$ . The low-fidelity solutions live in a space  $X_H$  with a low dimension  $N_H$ , which is significantly inferior to  $N_h$ . We get two training sets of size  $n_{\text{train}}$  : a high-fidelity set  $\mathcal{A}_h = \{w_h(s^1), \dots, w_h(s^{n_{\text{train}}})\}$  and a low-fidelity one  $\mathcal{A}_H = \{w_H(s^1), \dots, w_H(s^{n_{\text{train}}})\}$ .

The idea behind the NIRB method, in order to get a reduced approximation of  $w_h(s)$  for any  $s \in \mathcal{D}$ , is to rectify the associated coarse solution  $w_H(s)$ . As standard reduced basis methods, the NIRB method is divided into an *offline* stage and an *online* stage. In the *offline* stage, after obtaining the reduced basis  $V_N$ , one compute and save the coefficients of the projection on the RB space of both the high-fidelity and low-fidelity solutions of the training sets  $\mathcal{A}_h$  and  $\mathcal{A}_H$ . The collection of these coefficients allows to implement a rectification process, in the form of a  $N_h$ -independent rectification matrix. In the *online* stage, for each new value of  $s$ , one compute the coefficients of the projection on  $X_N$  of the low-fidelity solution  $w_H(s)$  and correct them with the rectification matrix. Then the linear combination of the reduced basis functions with the rectified coefficients gives a reduced approximation of  $w_h(s)$ . One can observe that the *online* stage does not depend on  $N_h$ , only the low dimension  $N_H$  interferes in the calculations.

In practice, the solutions  $w_h(s)$  (resp.  $w_H(s)$ ) for  $s \in \Xi_{\text{train}}$  are computed on a fine (resp. coarse) mesh  $\mathcal{T}_h$  (resp.  $\mathcal{T}_H$ ). Besides, the training set  $\mathcal{A}_h$  being already computed during the *offline* stage, the reduced

basis can be calculated using a POD method.

Let us detail precisely the explication of the NIRB method. Let  $B_N^h(s) \in \mathbb{R}^N$  be the optimal high-fidelity coefficient vector of  $w_h(s)$  in the reduced basis  $V_N$ .  $B_N^h(s)$  is optimal in the sense of the  $L^2$ -norm projection on the reduced space (in standard reduced basis methods, one also compute a coefficient vector but it is an approximation of the optimal coefficients of the solution). We define as well  $B_N^H(s)$  as the optimal low-fidelity coefficient vector. To summarize, one has :

$$\beta_j^h(s) = \left( B_N^h(s) \right)_j = \langle u_h(s), \zeta_j \rangle_{L^2}, \quad \beta_j^H(s) = \left( B_N^H(s) \right)_j = \langle u_H(s), \zeta_j \rangle_{L^2}, \quad 1 \leq j \leq N \quad (4)$$

In practice, one has to choose an appropriate maximal number for the dimension of the approximation space, noted  $N_{\max}$ . Then, one can generate the reduced basis  $V_N, 1 \leq N \leq N_{\max}$ . In the *online* stage, for each value of interest of  $s$ , one compute rapidly  $w_H(s)$  and the coefficient vector  $B_H(s)$ . Then one apply a rectification on  $B_H(s)$  to obtain an approximation of  $B_h(s)$ . This means that the reduced solution is an approximation of the  $L^2$ -norm projection of  $w_h(s)$  on the reduced basis  $V_N$ .

We still have to explain the rectification process. The low-fidelity vector  $B_H(s)$  is corrected using a rectification matrix  $R \in \mathbb{R}^{N_{\max} \times N_{\max}}$ , which is computed in the *offline* stage as follows : after producing the vectors  $B_{N_{\max}}^h(s^i)$  and  $B_{N_{\max}}^H(s^i)$  for all  $1 \leq i \leq n_{\text{train}}$ , one try to identify the matrix that links  $B_{N_{\max}}^H(s^i)$  to  $B_{N_{\max}}^h(s^i)$  for all set of parameter  $s^i$ . In other words, find  $R \in \mathbb{R}^{N_{\max} \times N_{\max}}$  such that

$$R B_{N_{\max}}^H(s^i) = B_{N_{\max}}^h(s^i), \quad \forall 1 \leq i \leq n_{\text{train}} \quad (5)$$

It can be written also as

$$\sum_{k=1}^{N_{\max}} R_{jk} \beta_k^H(s^i) = \beta_j^h(s^i), \quad \forall 1 \leq j \leq N_{\max}, \quad \forall 1 \leq i \leq n_{\text{train}} \quad (6)$$

In the *online* stage, for each new value of  $s$ , one replace the optimal high-fidelity coefficients  $\beta_j^h(s), 1 \leq j \leq N_{\max}$  by the following approximations

$$\beta_{j,N_{\max}}^{H,h}(s) = \sum_{k=1}^{N_{\max}} R_{jk} \beta_k^H(s) \quad (7)$$

or equivalently, under matrix form, by writing  $B_{N_{\max}}^{H,h}(s) \in \mathbb{R}^{N_{\max}}$  the approximation vector of the high-fidelity coefficients, *i.e.*  $\left( B_{N_{\max}}^{H,h}(s) \right)_j = \beta_{j,N_{\max}}^{H,h}(s)$

$$B_{N_{\max}}^{H,h}(s) = R B_{N_{\max}}^H(s) \quad (8)$$

Finally, with all this, one can construct a reduced less- intrusive approximation of the solution  $w_h(s)$

$$w_{N_{\max}}^{H,h}(s) = \sum_{j=1}^{N_{\max}} \beta_{j,N_{\max}}^{H,h}(s) \zeta_j \quad (9)$$

Of course, one can want to generate an approximation with only  $N$  reduced basis functions, with  $1 \leq N \leq N_{\max}$ . In that case, the *offline* stage is the same, but in the *online* stage one use a square submatrix

of  $R$  of dimension  $N$ . Therefore, one compute, for  $1 \leq j \leq N$

$$\dot{\beta}_{j,N}^{H,h}(s) = \sum_{k=1}^N R_{jk} \beta_k^H(s) \quad (10)$$

The reduced less intrusive approximation is noted  $\dot{w}_N^{H,h}(s)$  and is equal to

$$\dot{w}_N^{H,h}(s) = \sum_{j=1}^N \dot{\beta}_{j,N}^{H,h}(s) \zeta_j \quad (11)$$

In practice, how can we construct the rectification matrix  $R$ ? The construction process, described in [8], starts with the computation of the following matrix

$$\mathcal{H} = \begin{pmatrix} B_{N_{\max}}^H(s^1) \\ \vdots \\ B_{N_{\max}}^H(s^{n_{\text{train}}}) \end{pmatrix} = \begin{pmatrix} \beta_1^H(s^1) & \cdots & \beta_{N_{\max}}^H(s^1) \\ \vdots & \ddots & \vdots \\ \beta_1^H(s^{n_{\text{train}}}) & \cdots & \beta_{N_{\max}}^H(s^{n_{\text{train}}}) \end{pmatrix} \quad (12)$$

Moreover, we note  $L_{R_1}, \dots, L_{R_{N_{\max}}}$  the lines of the rectification matrix  $R$  and we define, for  $1 \leq j \leq N_{\max}$ , the vector  $\mathcal{S}_j^h \in \mathbb{R}^{n_{\text{train}}}$  par  $(\mathcal{S}_j^h)_k = \beta_j^h(s^k)$ . Then one can see from (6) that construct the matrix  $R$  is equivalent to find the vectors  $\mathcal{R}_j \in \mathbb{R}^N$  which minimizes the following functionals

$$C_j = \|\mathcal{H}\mathcal{R}_j - \mathcal{S}_j^h\|_2^2, \quad \forall 1 \leq j \leq N_{\max} \quad (13)$$

where  $\|\cdot\|_2$  is the Euclidean norm of  $\mathbb{R}^{n_{\text{train}}}$ . It is equivalent to solve the linear systems  $\mathcal{H}\mathcal{R}_j = \mathcal{S}_j^h$ . Finally,  $R$  is assembled by taking  $L_{R_j}^N = (\mathcal{R}_j^N)^T$ , for all  $1 \leq j \leq N_{\max}$ . To implement the NIRB method, we used the open-source PDE solver Freefem++ [15].

### 3 APPLICATION TO THE 2D BENCHMARK OF THE BACKWARD STEP

#### 3.1 Parametrical study

In what follows, we apply the NIRB method to the well-known 2D benchmark of the backward step. We study a turbulent airflow motion along a backward step. We choose to vary only one parameter of the physical model. This parameter is linked to the Reynolds number and is equal to the inlet velocity of the flow into the computational domain. The true flow velocity is designed by  $v(s)$  and the inlet velocity by  $v_{\text{in}}$ , so that one has  $s = v_{\text{in}} \in \mathcal{D} \subset \mathbb{R}$ , where  $\mathcal{D}$  is the parameter set.

Here, the turbulence is defined by the turbulent properties of the inlet flow. It means that, before the step, the flow corresponds to a turbulent channel flow. For such a flow, one can make the approximation  $v_{av} \approx v_{in}$ , where  $v_{av}$  is the averaged velocity (we verified this numerically). Thus, the following definition for the Reynolds number is used :  $Re = \frac{v_{in}h}{\nu}$ , where  $h$  is the step high and  $\nu$  the kinematic viscosity ( $\nu = 1.56 \times 10^{-5} \text{ m}^2/\text{s}$  for the air). It is consistent with the Reynolds number definition used in [16], where turbulent flows along a backward step are studied. An other definition of the step-based Reynolds number is proposed in [17] and [18], but it deals principally with laminar flows. For more information, a review of backward-facing step flow studies is given in [19].

We take  $\mathcal{D} = [0.0156, 1.56]$  (m/s) so that the Reynolds number lives between  $10^3$  and  $10^5$  ( $h = 1\text{m}$ , see the following subsection for details on the computational domain geometry). Let  $\Xi_{\text{train}}$  be a discretised subspace of  $\mathcal{D}$  of dimension  $n_{\text{train}} = 181$ . Those  $n_{\text{train}}$  parameters are associated with a Reynolds number value. It gives us 91 Reynolds number values for  $10^3 \leq Re \leq 10^4$  (every  $Re = 100$ ) and 90 values for  $10^4 < Re \leq 10^5$  (every  $Re = 1000$ ). We discretize more finely for  $Re$  less than  $10^4$  than after because the transition between laminar and turbulent flows happens in the interval  $[10^3; 10^4]$  (see [17]). Therefore, the flows along the backward step are quite different when  $Re$  is less than  $10^4$ , whereas beyond this value the flow behaviours are very similar, only the velocities change.

### 3.2 CFD modelling

The figure (1) illustrates the geometry considered for the CFD simulations. The step height is  $h = 1\text{m}$  and the expansion rate  $E_R = 2$ . The channel length before the step is  $L_1 = 12\text{m}$  and the channel length after the step is  $L_2 = 29\text{m}$ . The total length is  $L = L_1 + L_2 = 41\text{m}$ . The choice of this geometry was inspired by the approach of [16].

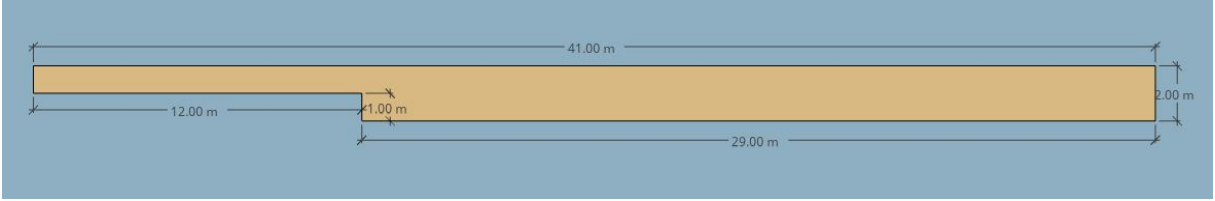


Figure 1: Geometry of the backward-facing step

We followed the recommendations and guidelines found in [20] for the selection of a turbulence model and the choice of a high-fidelity mesh. We suppose that the airflow is Newtonian and incompressible. The governing equations are the 2D Reynolds Averaged Navier-Stokes (RANS) equations with the  $k$ -omega SST model turbulence closure model. For reasons of numerical stability, pseudo-stationary equations are solved instead of stationary ones. The  $k$ -omega SST turbulence model [21], [22], gives accurate prediction near the walls, it deals with reattachment better than the standard  $k - \omega$  model and it is also less sensitive to boundaries conditions. Besides, it is well-suited for low-Reynolds turbulent flow simulations, which corresponds to a wide part of our simulations (for  $10^3 \leq Re \leq 10^4$ ). Thus, this turbulence model seems to be the more robust for this study.

Let  $\Omega$  be the computational domain,  $\Gamma$  the boundary of  $\Omega$ ,  $\Gamma_{\text{in}}$  the inlet boundary,  $\Gamma_{\text{out}}$  the outlet boundary and  $\Gamma_{\text{wall}} = \Gamma \setminus \{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}\}$  the wall boundaries. We enforce a non-homogeneous Dirichlet conditions on  $\Gamma_{\text{in}}$  ( $v = v_{\text{in}}$ ), a homogenous Neumann condition on  $\Gamma_{\text{out}}$  and a no-slip condition on  $\Gamma_{\text{wall}}$ .

Concerning the computational settings, we used as explained previously a  $k$ -omega SST turbulence model to simulate the 2D steady flow. The equations are discretized according to the Finite Volume method using the CFD software CODE\_SATURNE [23]. We chose The SIMPLE algorithm to deal with the pressure-velocity coupling and a Least Squares method based on a partial extended neighborhood for the gradient calculation. The time-schemes are all of first-order accuracy with a constant time step. The stopping criterion is the number of time step  $n_{\text{iter}}$ , which gives us a fixed simulation time. Two different values of  $n_{\text{iter}}$  are considered : one for  $10^4 \leq Re \leq 10^5$ , and one higher for  $10^3 \leq Re \leq 10^4$  (because the velocities are smaller, and the flow needs more time to leave the domain). We ensured ourselves that

these values are both large enough so that the simulated flow is steady. Finally, the solutions considered in the following are averaged over the last 20 time steps, in order to smooth them a little more (this operation seems to be useful only for low Reynolds number close to  $10^3$ ). The averaged velocity solutions are designated by  $u(s)$ .

### 3.3 Mesh convergence analysis

In this part, we performed a mesh convergence analysis to select a high-fidelity mesh. This analysis is specific to separation flows like backward-facing step flows because it is based on the reattachment length  $X_r$  of the flow, which is the average distance from the step where the flow leaves the circulation zone, see [19] for more details. Therefore, we generated five meshes of the backward step geometry using the open-source software SALOME [24]. The meshes characteristics are presented in table 1.  $\mathcal{T}_5$  is the finest mesh and  $\mathcal{T}_1$  the coarsest one. The maximum mesh element size, given in table 1 for each mesh, is a mesh generator parameter that represents the maximum linear dimensions for mesh cells. Besides, for each mesh, we allowed local size to vary on the walls around the step. Those variations are proportionate to the maximum mesh size.

Mesh Name	Maximum mesh size [m]	Local mesh size around step [m]	Number of cells	Time step [s]
$\mathcal{T}_1$	0.2	0.06	6028	0.2
$\mathcal{T}_2$	0.1	0.03	20 315	0.1
$\mathcal{T}_3$	0.05	0.015	72 875	0.05
$\mathcal{T}_4$	0.033	0.01	144 784	0.0333
$\mathcal{T}_5$	0.025	0.075	220 072	0.025

Table 1: Meshes characteristics

We computed, for each mesh and for two Reynolds numbers (the two extremities  $Re = 10^3$  and  $Re = 10^5$ , the ratio  $X_r/h$ ). The results are presented in table 2a. One can see that, for  $Re = 10^3$ , the ratio stabilizes from the third mesh at a value of 7.1, and for  $Re = 10^5$ , the ratio is almost constant from  $\mathcal{T}_1$  to  $\mathcal{T}_5$ . From these results, both meshes  $\mathcal{T}_3$  and  $\mathcal{T}_4$  can be selected as the high-fidelity mesh. To choose between them, let us compute numerical errors relative to a reference solution. Let  $u_i, 1 \leq i \leq 5$ , be the averaged velocities computed on the meshes  $\mathcal{T}_i, 1 \leq i \leq 5$ . For the calculation of the finest solution  $u_5$ , we took a time step equal to  $\Delta t = 0.025$  s and a number of iterations equal to  $n_{\text{iter}} = 160000$  (resp.  $n_{\text{iter}} = 16000$ ) for low (resp. high) Reynolds number. To select the high-fidelity mesh, the relative errors between  $u_1$  and the velocities  $u_i, 2 \leq i \leq 5$ , are compared. We compute the following relative error :

$$e_i = \frac{\sqrt{(I_i(u_1^x) - u_i^x)^2 + (I_i(u_1^y) - u_i^y)^2}}{\sqrt{\|u_i^x\|_{L^2}^2 + \|u_i^y\|_{L^2}^2}} \quad (14)$$

where  $(u_i^x, u_i^y)$  and respectively  $(u_1^x, u_1^y)$  are the (x, y)-components of the velocity vectors  $u_i$  and  $u_1$ . The operator  $I_i$  is an interpolation operator on the mesh  $\mathcal{T}_i$ , for  $2 \leq i \leq 5$ . These relative errors are given in  $L^2$ -norm in table 2b for  $Re = 10^3$  and  $Re = 10^5$ . One can see that, for  $Re = 10^3$ , the errors remains

Meshes	$Re = 10^3$	$Re = 10^5$
$\mathcal{T}_1$	9.1	8.3
$\mathcal{T}_2$	7.7	8.3
$\mathcal{T}_3$	7.1	8.5
$\mathcal{T}_4$	7.0	8.5
$\mathcal{T}_5$	7.0	8.3

(a)

	$Re = 10^3$	$Re = 10^5$
$e_2$	0.222	0.043
$e_3$	0.327	0.071
$e_4$	0.333	0.081
$e_5$	0.312	0.087

(b)

Table 2: Left table : value of the ratio  $X_r/h$  for the 5 meshes and for two extreme Reynolds number. Right table : values of the relative errors  $e_i$  between the solutions computed on meshes  $\mathcal{T}_i, 2 \leq i \leq 5$  and the coarsest mesh  $\mathcal{T}_1$  for two extreme Reynolds number

practically stable as of the second mesh. On the other hand, for  $Re = 10^5$ , the errors growth for each mesh but it seems that it will soon reaches a plateau. Indeed, the difference between  $e_4$  and  $e_5$  is almost twice inferior than the difference between  $e_3$  and  $e_4$ . As a consequence, the mesh  $\mathcal{T}_4$  is considered from now on as the high-fidelity mesh and is noted  $\mathcal{T}_h$ . The high-fidelity velocity solution is noted  $u_h(s)$ .

#### 4 NUMERICAL RESULTS

We recall that  $n_{train} = 181$ . Let  $\mathcal{A}_h = \{u_h(s^1), \dots, u_h(s^{n_{train}})\}$  be the training set associated with the  $n_{train}$  parameters. From this training set, we generated a reduced basis  $V_N = \{\zeta_1, \dots, \zeta_N\}, 1 \leq N \leq N_{max}$ , using a POD method. We set  $N_{max} = 50$ . We introduce the following projection errors on the reduced basis space (projection RB errors) for  $1 \leq N \leq N_{max}$

$$e_N(s^i) = u_h(s^i) - \sum_{j=1}^N \langle u_h(s^i), \zeta_j \rangle_{L^2} \zeta_j, \quad 1 \leq i \leq n_{train}. \quad (15)$$

and the associated relative errors in  $L^2$ -norm  $e_N^{rel}(s^i) = \frac{\|e_N(s^i)\|_{L^2}}{\|u_h(s^i)\|_{L^2}}$ . To verify that the reduced basis is a good approximation of the full solution space, we computed first the averaged relative error over the training set  $err^N = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} e_N^{rel}(s^i)$ . One observe on the figure 2 that, with  $N = N_{max}$ , the averaged relative error is approximatively equal to 0.2%. Yet, it is not enough to conclude on the quality of the reduced basis, because the latter was formed with solutions from the training set. Therefore, we performed a Leave-One-Out Cross Validation (LOOCV) method to test the reduced basis space quality when it is faced with new data :  $n_{train}$  new POD methods were applied on the training set  $\mathcal{A}_h$  lacking each time with a different velocity solution. The missing solution is projected on the new reduced basis and the associated projection RB error is computed. The averaged results are represented on figure 2. One can see that we recover almost entirely the nice projection error of the initial training set. Thus, the reduced basis space is a good approximation of  $X_h$ .



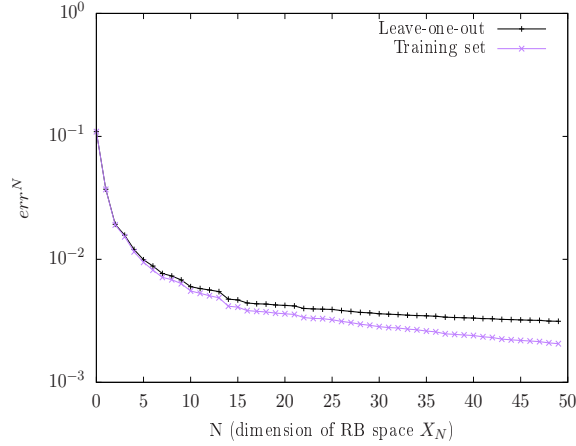


Figure 2: Relative projection errors averaged over the training set and with the LOOCV method with respect to the dimension  $N$  of the reduced space  $X_N$

In the following, the NIRB method is tested on the backward step benchmark using low-fidelity solutions  $u_H(s)$  from the coarse meshes  $\mathcal{T}_i$ ,  $1 \leq i \leq 2$ . We want to compare the projection RB errors with a NIRB projection error, which is the projection error on  $X_N$  of the rectified low-fidelity solutions.

Let us give more details about these errors. On figure 3, we plotted the relative errors  $err^N$  introduced previously and, for the two coarse meshes  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , the following NIRB projection averaged errors

$$\frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \frac{\|u_h(s^i) - \sum_{j=1}^N \beta_{j,N}^{H,h}(s^i) \zeta_j\|_{L^2}}{\|u_h(s^i)\|_{L^2}} \quad (16)$$

We also represented on figure 3 the NIRB projection errors without rectification, *i.e.*

$$\frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \frac{\|u_h(s^i) - \sum_{j=1}^N \langle u_H(s^i), \zeta_j \rangle_{L^2} \zeta_j\|_{L^2}}{\|u_h(s^i)\|_{L^2}} \quad (17)$$

to see how the rectification improves the reduced approximation quality. One can observe that for coarse solutions computed on the coarsest mesh  $\mathcal{T}_1$  (left figure), the NIRB projection error with rectification reaches 1% for  $N = 5$ , 0.7% for  $N = 10$  and approximately 0.65% for  $N = N_{\text{max}}$ . Besides, for the mesh  $\mathcal{T}_2$  (right figure), one see that the NIRB error is marginally weaker than the NIRB error for  $\mathcal{T}_1$ . There is no need to increase the computational cost in the *online* stage for such a poor improvement concerning the NIRB error (indeed, a coarse solution calculation on  $\mathcal{T}_2$  takes about 4 to 5 more time than a calculation on  $\mathcal{T}_1$ ). Thus, the NIRB method is applied with the first mesh, and it is satisfying regarding the averaged errors over the training set. Finally, it is important to mention that the NIRB projection errors are quite close to the RB projection errors.

As for the RB approximation, one need to test the quality of the NIRB approximation. This time, we did not go for the LOOCV method. Instead, we computed  $n_{\text{trial}} = 10$  random trial solutions associated with  $n_{\text{trial}}$  parameters  $s^i$ . The trial set is  $\mathcal{A}_{t,h} = \{u_h(s^i) \mid 1 \leq i \leq n_{\text{trial}}\}$ . The  $n_{\text{trial}}$  parameters, chosen randomly, gave us these values of Reynolds number : (1122, 3333, 6040, 7237, 9647, 22436, 44532, 63462, 82500, 95449). The maximal RB and NIRB (with rectification) projection errors of the training

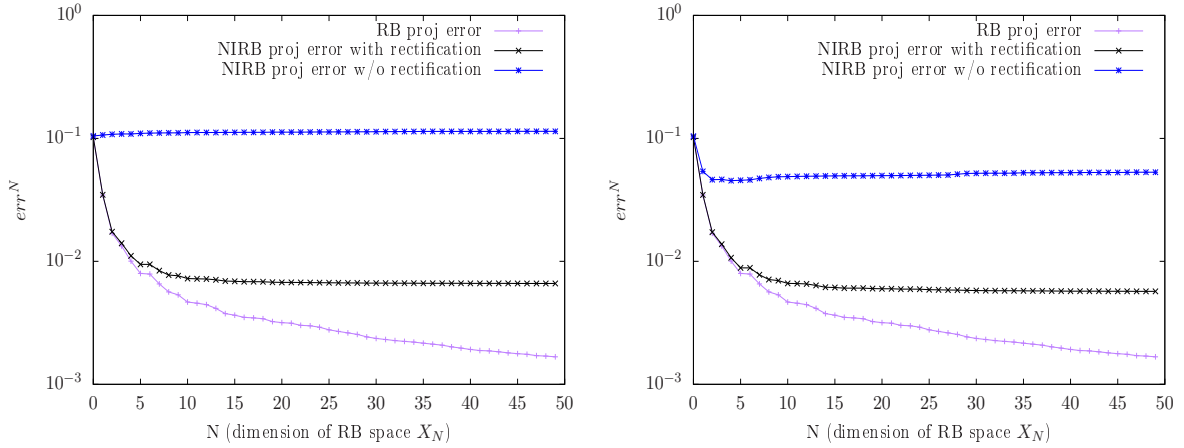


Figure 3: RB and NIRB projection errors (with and without rectification) averaged over the training set with respect to the dimension  $N$  of the approximation space. On left (resp. right) figure, low-fidelity solutions are computed on the mesh  $\mathcal{T}_1$  (resp.  $\mathcal{T}_2$ )

solutions is plotted on figure 4 (left figure), as well as the maximal RB and NIRB projection error of the trial solutions (right figure). First, one observe on right figure that the maximal projection error over

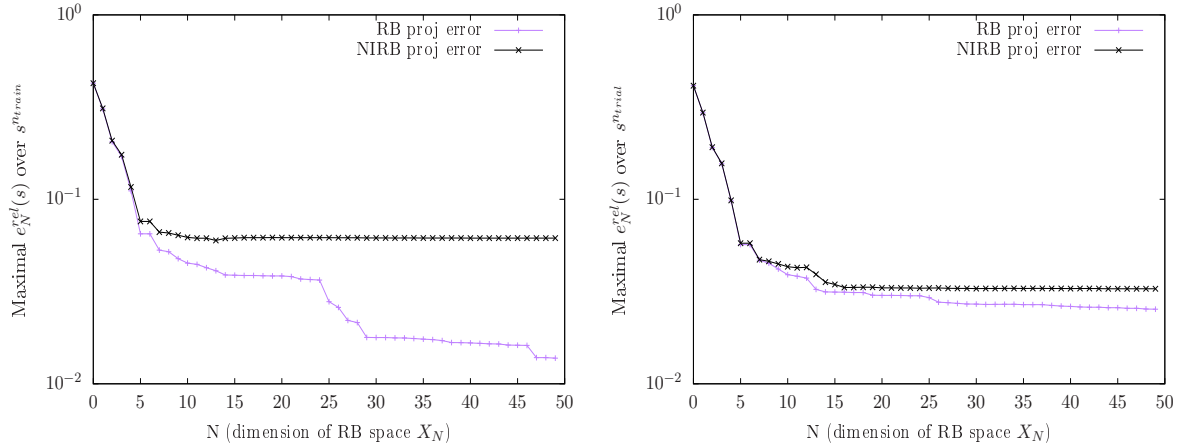


Figure 4: RB and NIRB maximal projection errors over the training set (left figure) and the trial set (right figure) with respect to  $N$ .

the trial set is as good as the maximal RB projection error over the trial set. It means that we recover entirely the approximation quality of the RB method and that the NIRB method works in this case. One can also notice that the maximal NIRB projection error over the training set is superior than with the trial set. It comes from the fact that the reduced basis space hardly represents low Reynolds high-fidelity solutions (this is where the projection error is the highest, in particular with  $Re$  close to  $10^3$ ). This leads to a NIRB error maximal for low-Reynolds numbers, because we still use the reduced basis functions to compute the NIRB error. Since in the trial set, we only have one solution with a Reynolds number close to  $10^3$  (whereas in the training set there are several of them), it is likely that the maximal NIRB error is

smaller for the trial solutions. In practice, we deal with this by separating the parameter space in 2, one for Reynolds numbers inferior to  $10^4$  and one for Reynolds number superior to  $10^4$ . Two reduced basis are then generated and it becomes simpler to capture all the physical phenomena which appears between laminar, transitory and turbulent flows.

## 5 CONCLUSION

In this work we present a non-intrusive RB method applied to the 2D backward step benchmark. This approach, based on a very coarse mesh, allows in this case to win considerable computational time for a wide range of Reynolds number solutions. The lost of accuracy is very acceptable regarding how the reduced approximation is close to the high-fidelity solution : for some random trial solutions associated to a Reynolds number value between  $10^3$  and  $10^5$ , we found a maximal projection error around 0.4%. Future works will included the extension of the NIRB method to real urban flows modelling at the district scale.

## REFERENCES

- [1] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*. Springer, 2016.
- [2] N. N. Cuong, K. Veroy, and A. Patera. Certified real-time solution of parametrized partial differential equations. 2005.
- [3] 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. *J. Fluids Eng*, 124(1):70–80, 2002.
- [4] L. Fick, Y. Maday, A. Patera, and T. Taddei. A reduced basis technique for long-time unsteady turbulent flows. 2017.
- [5] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numerical Methods for PDE*, 23, 2007.
- [6] K. Veroy and A. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Int. J. for Numer. Methods in Fluids*, 47:773–788, 2005.
- [7] R. Chakir and Y. Maday. Une méthode combinée d'éléments finis à deux grilles/bases réduites pour l'approximation des solutions d'une e.d.p. paramétrique. *Comptes Rendus Mathématiques*, 347(7):435–440, 2007.
- [8] 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, 2019.
- [9] R. Chakir and J. K. Hammond. A non-intrusive reduced basis method for elastoplasticity problems in geotechnics. *J. of Comput. App. Math.*, 2018.
- [10] S. Volkwein. Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. *Lecture Notes, University of Konstanz*, 2013.
- [11] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of

- parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.
- [12] M. Barrault, Y. Maday, N. N. Cuong, and A. T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 39(9):667–672, 2004.
- [13] Y. Maday and O. Mula. A generalized empirical interpolation method : application of reduced basis techniques to data assimilation. *Analysis and Numerics of Partial Differential Equations. Springer INdAM Series*, 2013.
- [14] C. Daversin and C. Prud’homme. Simultaneous empirical interpolation and reduced basis method for non-linear problems. *Comptes Rendus Mathematique*, 353(12):1105–1109, 2015.
- [15] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [16] H. Le, P. Moin, and J. Kim. Direct numerical simulation of turbulent flow over a backward-facing step. *J. Fluid Mech*, 330:349–374, 1997.
- [17] B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schönung. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mechanics*, 127:473–496, 1983.
- [18] G. Biswas, M. Breuer, and F. Durst. Backward-Facing Step Flows for Various Expansion Ratios at Low and Moderate Reynolds Numbers. *J. Fluids Eng*, 126(3):362–374, 2004.
- [19] L. Chen, K. Asai, T. Nonomura, G. Xi, and T. Liu. A review of Backward-Facing Step (BFS) flow mechanisms, heat transfer and control. *Thermal Science and Engineering Progress*, 6:194–216, 2018.
- [20] B. Blocken. Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91:219–245, 2015.
- [21] F. R. Menter. Zonal Two Equation  $k - \omega$  Turbulence Models for Aerodynamic Flows. *AIAA Journal*, 1993.
- [22] D. C. Wilcox. Formulation of the  $k - \omega$  Turbulence Model Revisited. *AIAA Journal*, 46(11), 2008.
- [23] 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, 2004.
- [24] A. Ribes and C. Caremoli. Salome platform component model for numerical simulation. In *COMP-SAC 07: Proceeding of the 31st Annual International Computer Software and Applications Conference*, pages 553–564, Washington, DC, USA, 2007. IEEE Computer Society.