



# Reduction of the Shallow Water System by an Error Aware POD-Neural Network Method: Application to Floodplain Dynamics

Mustapha Allabou, Robin Bouclier, Pierre-André Garambois, Jerome Monnier

## ► To cite this version:

Mustapha Allabou, Robin Bouclier, Pierre-André Garambois, Jerome Monnier. Reduction of the Shallow Water System by an Error Aware POD-Neural Network Method: Application to Floodplain Dynamics. 2024. hal-04391872v1

**HAL Id: hal-04391872**

**<https://hal.science/hal-04391872v1>**

Preprint submitted on 12 Jan 2024 (v1), last revised 12 Apr 2024 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reduction of the Shallow Water System by an Error Aware POD-Neural Network Method: Application to Floodplain Dynamics.

M. Allabou<sup>a</sup>, R. Bouclier<sup>a,b</sup>, P.A. Garambois<sup>c</sup> and J. Monnier<sup>a</sup>

<sup>a</sup>*Institut de Mathématiques de Toulouse (IMT), Université de Toulouse, CNRS-INS-UT1-UT2-UPS, 1 R.3, Université Paul Sabatier, 118 Rte de Narbonne, Toulouse, 31400, France*

<sup>b</sup>*Institut Clément Ader (ICA), Université de Toulouse, CNRS-INS-ISAIE-Mines Albi-UPS, Toulouse, France*

<sup>c</sup>*INRAE, UMR Recover, Aix-Marseille Université, Aix-en-Provence, France*

## ARTICLE INFO

### Keywords:

Non-intrusive reduced order models  
Shallow water equations  
Proper orthogonal decomposition  
Artificial neural network  
Error learning  
River hydraulics

## Abstract

In this study, we elaborate on and evaluate a new reduced basis method for model reduction of the shallow water equations using Proper Orthogonal Decomposition (POD) and artificial Neural Networks (NNs). The method begins with the POD technique to construct reduced bases from high-resolution solutions, followed by training two deep NNs to learn associated coefficients in the reduced bases. The approach follows an offline-online strategy: the POD reduced basis, along with the training of the NNs, is performed in an offline stage, and then the surrogate model can be used in an online stage for real-time predictions. The method takes into account the POD-based projection error, enabling the attainment of higher accuracy while preserving a limited number of POD modes, even in the delicate situation of convection-dominated flow problems. This point is crucial in our approach since it enables to limit the output dimension of the NNs, thus providing the opportunity to employ smaller NNs (with less parameters), and therefore smaller dataset (the snapshots) to train the NNs. The process is non-intrusive: it does not require opening the high-resolution model code. The method is evaluated on a real-world test case aimed at simulating inundation of the Aude river (Southern France). The results show that the proposed method provides satisfying accuracy for the hydraulic variables (water elevation, discharge) compared to the reference high-resolution 2D shallow water model. Overall, the method is promising, particularly for performing real-time simulations of flood plain dynamics.

## 1. Introduction

The Reduced Order Model (ROM) theory (see, *e.g.*, [19, 13, 3]) has gained significant attention in the field of computational sciences, offering techniques to tackle the computational challenges posed by complex systems. The goal is to simplify high-dimensional models while preserving their essential features. Model reduction methods can be broadly categorized into intrusive and non-intrusive approaches, each with its advantages and limitations, see *e.g.* [19, 13, 3] and references therein. In this paper, we adopt an offline-online non-intrusive strategy, applied to non-linear hyperbolic problems, specifically to the 2D Shallow Water Equations (SWEs). This PDEs system describes well the behavior of flood plain dynamics. Capturing complex features such as non linear travelling waves, over long time integration periods, using *e.g.* Finite Volume (FV) schemes is CPU-time consuming, which is impractical for real-time

\*Corresponding author:

E-mail address: monnier@insa-toulouse.fr (J. Monnier)

✉ allaboumstapha96@gmail.com (M. Allabou); bouclier@insa-toulouse.fr (R. Bouclier); pierre-andre.garambois@inrae.fr (P.A. Garambois); monnier@insa-toulouse.fr (J. Monnier)

🌐 <https://www.math.univ-toulouse.fr/~jmonnie/> (J. Monnier)

ORCID(s):

simulations or optimization and uncertainty quantification. To overcome these challenges, model reduction techniques offer an alternative approach by reducing the dimensionality of the problem while preserving the crucial dynamics. However, ROMs are particularly challenging in this situation of convection-dominated problems, see e.g. [19, 1, 2] and references therein. Applying ROM techniques aims to strike a balance between computational efficiency and accuracy. The reduction of these complex systems enables faster simulations, facilitates real-time decision-making processes, and opens avenues for optimization and uncertainty quantification tasks.

The literature addressing ROMs for non linear hyperbolic systems like SWEs is already quite large. For intrusive ROM approaches, let us mention for example [4, 23, 10, 24] and for non-intrusive ones [9, 25, 12], see Table 1. To reduce the 2D SWEs model (considered as non-parametrized), [23, 4] propose POD-DEIM (Proper Orthogonal Decomposition - Discrete Empirical Interpolation Method) approaches which are intrusive. [9] proposes a non-intrusive method based on POD and Radial Basis Functions (RBF) interpolation. Considering the 2D SWEs as a parametrized model, [10, 24] propose intrusive approaches based on generalized polynomial chaos and POD-Galerkin, respectively. [25, 12] address non-intrusive POD-NN (Neural Network) based reduction methods to reduce parametrized SWEs. These POD-NN based methods rely on the usual offline-online strategy: a computationally expensive offline phase is performed beforehand to construct the reduced-order model, while the online phase efficiently computes the reduced system's response given a new parameter value, without further involving the full-order model (non-intrusive aspect). In these POD-NN methods, the POD enables to identify the dominant modes of the system, capturing the most significant features, while the NN serves as approximating the coefficients in the POD-based reduced basis. Such a combination of POD and NN seems to be first introduced in [14], and later in [25]. In these pioneering studies, the method showed promising efficiency for relatively simple PDE-based models, but also for the steady incompressible Navier-Stokes equations modeling an academic driven cavity viscous flows. However, such POD-NN based methods turned out to be insufficiently accurate for complex situations such as those presented in the present study (a real-life flooding event).

In this work, we extend the aforementioned POD-NN method by learning, in addition to the reduced (projected) solution, the projection error, through an additional NN. For non-linear convection-dominated or hyperbolic systems such as the one considered here, this enhancement turns out to be essential for constructing an accurate and robust surrogate model. Indeed, it allows for the correction of the reduced solution obtained from the standard POD-NN method while limiting the output dimension of the NNs, *i.e.*, the dimension of the reduced spaces. This limitation enables to maintain reasonably sized datasets (snapshots) for training. Without this, achieving an accurate machine-learning-type method may be infeasible. Ultimately, the proposed method preserves twofold advantages of the original

Paper	Math. model, num. scheme	ROM method	Input variable/dimension	vari-	Test case	Non-intrusive?
Stefanescu et al., IJNMF 2014 [23]	2D SWEs $(h, u, v)$ , FD	Tensorial POD-DEIM	time / 1		Analytical solution	Intrusive
Steinstraesser et al., JCM 2021 [4]	2D SWEs $(h, u, v)$ , FV	POD-DEIM	time / 1		Analytical solution	Intrusive
Dutta et al., JCP 2021 [9]	2D SWEs $(h, Q)$ , FE	RBF-POD	time / 1		Kissimmee river (USA)	✓
El Moçayd et al., EMA 2017 [10]	1D SWEs $(h, Q)$ , FV	Generalized polynomial chaos	$(Q, K_s)$ / 2 or 4		Garonne river (FR)	Intrusive
Strazzullo et al., JNMA 2022 [24]	2D SWEs, FE	Galerkin POD	Affinely parametrized (optimal control context) / 3		Analytical solution	Intrusive
Wang et al., JCP 2019 [25]	1D Euler equations, FV	POD-NN	$(t; \delta_1, \delta_2)$ (IC param.) / 3		Shock tube	✓
Ghorbanidehno et al., AWR 2020 [12]	2D SWEs $(h, u, v)$ , FE	POD-NN	Velocity data (Inverse pb context) / 11		Real rivers portions (USA)	✓

**Table 1**

A few references of ROM applied to SWEs (or Euler equations). (FV) Finite Volumes, (FE) Finite Elements, (FD) Finite Differences.

POD-NN method: (i) it is generic, meaning it can be applied to any PDEs based problems solved by any discretization schemes (FV or FE for example); (ii) it is non-intrusive as it only requires to run, in a black-box fashion, the high-resolution model code multiple times in the offline phase. Due to the incorporation of this error-learning step, we refer to our approach as an Error-Aware POD-NN (EA-POD-NN) reduction method.

The paper is organized as follows. After this introduction, Section 2 introduces the considered parametrized 2D SWEs model, and its numerical resolution. Then, Section 3 presents, in a general manner, the developed EA-POD-NN methodology. It outlines the steps involved in the offline and online phases of the method and describes the corresponding algorithms. Particular care is given in this section to highlight the limitations of the existing POD-NN approach, thereby motivating the integration of the projection error within the process. Section 4 presents the numerical experiments conducted to evaluate the performance and accuracy of the EA-POD-NN method. It discusses the selected SWEs problems, the setup of the simulations, and the comparison of results between the initial high-resolution model and the ROM using the proposed EA-POD-NN approach. Eventually, Section 5 provides a brief conclusion, summarizing the key findings and main contributions of this study, and identifying potential areas for future research based on the proposed approach.

## 2. The $\mu$ -parametrized model and the reference high-resolution solutions

In this section, first, the general concept of High-Resolution (HR) solutions corresponding to a given set of input parameters is recalled. Next, the flow model considered in our study is presented, followed by a discussion of the numerical scheme and computational software employed to solve it.

### 2.1. Basic principle of the high-resolution solutions generation

The reduction method developed in this study can be formally applied to any  $\mu$ -parametrized non-linear PDE-based model with the input parameter  $\mu = (\mu_1, \dots, \mu_{N_\mu}) \in \mathbb{R}^{N_\mu}$ , which typically represents a set of physical parameters in the bulk and/or boundary condition parameters. From a reduction point of view, the important characteristic of  $\mu$  is its dimension  $N_\mu$ :  $N_\mu$  has to be quite small, say  $N_\mu = \mathcal{O}(10)$  at most, otherwise an additional reduction method of  $\mu$  should be considered in a pre-processing step.

More precisely, we set  $\mathcal{H}$  such that  $\mu \in \mathcal{H}$  with  $\dim(\mathcal{H}) = N_\mu$ . A set of  $N_s$  parameters are fixed by sampling in some way the parameter space  $\mathcal{H}$ . We obtain the reference parameter set  $P_s = \{\mu_s\}_{s=1}^{N_s}$ ,  $P_s \in (\mathcal{H})^{N_s}$ . Next, the  $\mu$ -parametrized model, denoted as  $\mathcal{M}_\mu$ , is employed to generate the  $N_s$  corresponding vector solutions  $\mathbf{u}_h$ : this is the classically called HR solutions set, also called the snapshots set. These snapshots  $\mathbf{u}_h$  constitute the set  $\mathbf{S} = [\mathbf{u}_h(\mu_1) | \dots | \mathbf{u}_h(\mu_{N_s})]$ .

### 2.2. The $\mu$ -parametrized mathematical flow model

In the developed application, the considered model  $\mathcal{M}_\mu$  relies on the 2D SWEs, employed in particular to simulate river and floodplain flows dynamics. In this context, the parameter  $\mu$  could be related to boundary conditions, to the initial conditions, or to the friction coefficient, etc.

For a given computational domain  $\Omega \subset \mathbb{R}^2$  and a time interval  $[0, T]$ , the considered 2D SWEs are those written in conservative form as:

$$(\mathcal{M}_\mu) \begin{cases} \partial_t h + \text{div}(\mathbf{q}) & = 0 & \text{in } \Omega \times ]0, T] \\ \partial_t \mathbf{q} + \text{div} \left( \frac{\mathbf{q} \otimes \mathbf{q}}{h} + g \frac{h^2}{2} Id \right) & = S_g(\mu; h) + S_f(\mu; h, \mathbf{q}) & \text{in } \Omega \times ]0, T] \\ \text{Plus Initial Conditions,} & \text{plus Boundary Conditions (B.C.)}(\mu) \end{cases} \quad (1)$$

$\forall (x, t) \in \Omega \times [0, T]$ , the variable  $h(x, t)$  represents the water depth (in  $m$ ) and  $\mathbf{q}(x, t)$  stands for the discharge (in  $m^3 s^{-1}$ ). Then, we have  $\mathbf{q} = h\mathbf{U}$ , where  $\mathbf{U} = (v_x, v_y)^T$  denotes the depth-averaged velocity (in  $ms^{-1}$ ). Moreover:

- $S_g(\mu; h)$  is the gravity source term:  $S_g(\mu; h) = -gh\nabla z_b$ , with  $g$  the gravity magnitude and  $z_b$  the bed elevation;

- $S_f(\mu; h, \mathbf{q})$  is the friction source term:  $S_f(\mu; h, \mathbf{q}) = -g \frac{n^2 \|\mathbf{q}\|}{h^{7/3}} \mathbf{q}$ , with  $n$  the Manning-Strickler roughness coefficient.

The B.C. can be a mix of conditions necessary for real-world applications. At outflow boundaries, different types of flow conditions can be imposed, depending on the available information, see [22] for details. At the inflow boundaries, discharge time series are imposed. In the case of  $N_{in}$  inflow boundaries, the inflow discharge time series  $Q_{in,k}(\mu; t)$ ,  $k \in 1..N_{in}$ , are imposed. For the presentation of our method in a general manner in Section 3, let us denote by  $u_h = (h, \mathbf{q}) = (h, h\mathbf{U}) = (h, q_x, q_y)$  the HR solution field of (1).

In this work, our primary focus is on the most important input parameter for operational users; that is, the inflow discharge signal at upstream (inflow B.C.). Consequently, in the numerical applications, the  $\mu$  parameter will be related to the discharge functions  $Q_{in,k}(\mu, t)$ . Moreover, for simplicity in notation in the following, we will incorporate the time parameter  $t_{sn}$  into the parameter  $\mu$ . This is not only convenient but also scientifically sound, considering that  $t_{sn}$  represents a dimension of  $\mathcal{H}$  for constructing the snapshot set.

### Numerical solver

Given a parameter value  $\mu$ , the system  $\mathcal{M}_\mu$  is numerically solved by the first-order FV method implemented into the open-source computational software **DassFlow 2D** [22, 18]. The solver relies on a well-balanced Godunov-type scheme and the explicit Euler time scheme, with additional treatment to be stable and robust at wet-dry front dynamics. The boundary conditions required for real-world flows are available. The mesh is a mix of triangles and quadrangles. The DassFlow 2D kernel codes are written in Fortran 90, using MPI library. Moreover, the Fortran computational kernels are wrapped in Python. This allows for seamless integration of these physics-based computations with other Python libraries. These libraries can serve various purposes, including simple pre or post-processing of computed fields or data, as well as incorporating Python libraries like PyTorch for deep learning, and so forth.

## 3. The Error-Aware POD-NN reduction method

Let us now detail the proposed reduction method that allows to construct an efficient and accurate surrogate model, here applied to the 2D SWEs-based model  $\mathcal{M}_\mu$ . The presentation in this section is crafted in a general manner as the proposed approach is generic and, consequently, can be applied in various contexts. Specifically, the presentation follows the different aspects of the method, namely (i) the projection onto a RB, (ii) the POD technique for constructing the RB, (iii) the learning of coefficients of the reduced solution using a deep NN, and (iv) the involvement of the projection error.

### 3.1. Reduced basis and projection

Our method being based on a RB approach, let us start by properly defining the complete space, the reduced space through its RB, and the relations between both. For illustration purpose, one can refer to Fig. 1 which will be detailed in the following.

Let us denote  $V_h$  as the initial complete functional space that contains the discrete solution computed using the HR model. Additionally, we introduce the basis of shape functions  $\Phi(x) = \{\varphi_l(x)\}_{l=1}^{N_h}$  that generates  $V_h$ , meaning that  $V_h = \text{span } \Phi(x)$  with  $\dim(V_h) = N_h$ . Then, let us define the reduced space  $V_{rb} \subset V_h$  such that  $V_{rb} = \text{span } \Xi(x)$ , where  $\Xi(x) = \{\xi_n(x)\}_{n=1}^{N_{rb}}$  constitutes the RB. Therefore, we have  $\dim(V_{rb}) = N_{rb}$ , and we expect  $N_{rb} \ll N_h$ . Ultimately, the initial complete solution  $u_h(\mu; x)$  and the reduced one  $u_{rb}(\mu; x)$  can be written in algebraic form as follows:

$$u_h(\mu; x) = \Phi(x)^T \mathbf{u}_h(\mu) \quad ; \quad u_{rb}(\mu; x) = \Xi(x)^T \mathbf{u}_{rb}(\mu). \quad (2)$$

Here,  $\Phi(x)$  (respectively,  $\Xi(x)$ ) and  $\mathbf{u}_h(\mu)$  (respectively,  $\mathbf{u}_{rb}(\mu)$ ) are vectors in  $\mathbb{R}^{N_h}$  (resp. in  $\mathbb{R}^{N_{rb}}$ ) that collect the shape functions and associated coefficients. In the remainder of this Section, since the spaces  $V_h$  and  $V_{rb}$  are the same for any  $\mu \in \mathcal{H}$ , we will omit the dependence of the solutions on these parameters for more simplicity in the notations.

**Remark 1.** Note that the definition of  $V_h$  encompasses all common discretization methods encountered in scientific computing, such as FE, FV, Discontinuous Galerkin, and so on. For instance,  $\Phi(x)$  simply represents the nodal Lagrange shape functions in the FE context or corresponds to constant functions per cell in a FV framework. In our application, we will focus on the FV method, but it is worth noting that the proposed approach is applicable to other contexts, as will be underlined later.

As a first relation between the two spaces, we can construct the matrix  $\mathbf{B}_{rb} \in \mathbb{R}^{N_h \times N_{rb}}$ , which encodes the change of variables from the RB  $\Xi(x)$  to the complete one  $\Phi(x)$ . More precisely, it is defined as:

$$\mathbf{B}_{rb} = \left[ \xi_1 \mid \dots \mid \xi_{N_{rb}} \right], \quad (3)$$

where  $\xi_n$  denotes the vector gathering the coefficients of function  $\xi_n(x)$  in the complete basis  $\Phi(x)$ . Thus, we can write:

$$\Xi(x) = \mathbf{B}_{rb}^T \Phi(x), \quad (4)$$

and it follows that any reduced solution  $u_{rb}(x) \in V_{rb}$  can be represented in the complete basis  $\Phi(x)$  as:

$$u_{rb}(x) = \Xi(x)^T \mathbf{u}_{rb} = \Phi(x)^T \mathbf{B}_{rb} \mathbf{u}_{rb} = \Phi(x)^T \mathbf{u}_{rb}^{N_h}. \quad (5)$$

where

$$\mathbf{u}_{rb}^{N_h} = \mathbf{B}_{rb} \mathbf{u}_{rb} \in \mathbb{R}^{N_h} \quad (6)$$

refers to the vector that contains the coefficients of the reduced solution  $u_{rb}(x)$  in the complete basis  $\Phi(x)$ .

Next, as a second relation, we can express the orthogonal projection from the complete space  $V_h$  onto the reduced space  $V_{rb}$ . In the continuous setting, the projection is defined by the projection operator  $\mathcal{P}_{rb}$  as follows:

$$\begin{aligned} \mathcal{P}_{rb} &: V_h \rightarrow V_{rb} \\ u_h(x) &\mapsto \sum_{n=1}^{N_{rb}} (u_h(x), \xi_n(x))_h \xi_n(x), \end{aligned} \quad (7)$$

where  $(\cdot, \cdot)_h$  represents a scalar product on  $V_h$ . In our application, since the numerical solution is discretized using piecewise constant functions (see Remark 1), we will simply consider the  $L^2$ -scalar product. With this in mind and transitioning to the algebraic setting, the projection matrix  $\mathbf{P}_{rb} \in \mathbb{R}^{N_h \times N_h}$  reads:

$$\forall \mathbf{u}_h \in \mathbb{R}^{N_h}, \quad \mathbf{P}_{rb} \mathbf{u}_h = \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h = \mathbf{B}_{rb} \mathbf{u}_h^{N_{rb}}, \quad (8)$$

where

$$\mathbf{u}_h^{N_{rb}} = \mathbf{B}_{rb}^T \mathbf{u}_h \in \mathbb{R}^{N_{rb}} \quad (9)$$

refers to the vector that collects the coefficients in the RB  $\Xi(x)$  of the projection of  $u_h(x)$  onto the reduced space  $V_{rb}$  (see Fig. 1). With all this in hand, we can eventually define the projection error of  $u_h(x)$  onto  $V_{rb}$  in the algebraic form as follows:

$$\mathbf{e}_h = \mathbf{u}_h - \mathbf{P}_{rb} \mathbf{u}_h = \mathbf{u}_h - \mathbf{B}_{rb} \mathbf{u}_h^{N_{rb}} = \mathbf{u}_h - \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h, \quad (10)$$

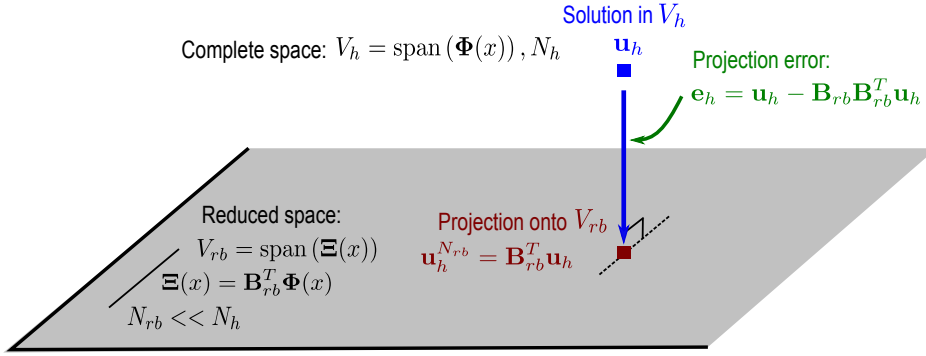
where  $\mathbf{e}_h \in \mathbb{R}^{N_h}$  (see Fig. 1 again).

The key point now is to build a RB, or equivalently the matrix  $\mathbf{B}_{rb}$ , such that  $N_{rb}$  is small and  $\mathbf{P}_{rb} \mathbf{u}_h \approx \mathbf{u}_h$ , meaning  $\mathbf{e}_h \approx 0$  for any  $\mu \in \mathcal{H}$ . This will be performed with the classical POD method, which is reviewed in next Section.

### 3.2. The POD reduced basis

The POD method is far from recent (see, e.g., [17] for an application in turbulent fluid mechanics) and is now widely used in many fields of scientific computing (refer to the following books [19, 13, 3] to name a few). The interest of the





**Figure 1:** Schematic representation of the different spaces:  $V_h$  is the initial complete space (associated basis  $\Phi(x)$ );  $V_{rb} \subset V_h$  is the reduced space (associated basis  $\Xi(x)$ );  $N_h$  and  $N_{rb}$  are the dimensions of  $V_h$  and  $V_{rb}$ , respectively;  $B_{rb}$  is the change of variables matrix;  $\mathbf{u}_h^{N_{rb}}$  is the projection vector in  $\Xi(x)$  of  $\mathbf{u}_h$  onto  $V_{rb}$ ; and  $\mathbf{e}_h$  is the corresponding projection error (expressed in basis  $\Phi(x)$ ).

POD strategy is that it leads to a RB that is optimal in a chosen norm with respect to a collection of complete solutions. In this regard, it may be noted that the Principal Component Analysis (PCA) is essentially the same as the POD method but applied to problems in statistics (the aim of the latter being to convert primitive variables into uncorrelated ones).

To briefly present the construction of the POD RB, let us start by considering the set  $P_s = \{\mu_s\}_{s=1}^{N_s} \in (\mathcal{H})^{N_s}$ , which comprises  $N_s$  parameter samples obtained by sampling the hypercube  $\mathcal{H}$ . These parameters are used to generate a set of complete numerical solutions with the HR model. These solutions are referred to as *snapshots* and their corresponding coefficient vectors are stored in the so-called snapshot matrix  $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$ , defined as:

$$\mathbf{S} = \left[ \mathbf{u}_h(\mu_1) \mid \dots \mid \mathbf{u}_h(\mu_{N_s}) \right]. \quad (11)$$

The principle of the POD method is then to perform a Singular Value Decomposition (SVD) of the rectangular matrix  $\mathbf{S}$  and to retain the most significant left singular vectors, following the Schmidt-Eckart-Young theorem. These retained singular vectors constitute the POD modes that form the RB.

In our application, we will be in the case of  $N_s > N_h$  since numerous snapshots will be necessary to accurately capture the underlying physics of the model (see Section 4). For the SVD, we therefore build the correlation matrix  $\mathbf{C}$  as follows:

$$\mathbf{C} = \mathbf{S} \mathbf{S}^T. \quad (12)$$

Note that, again, we use here the  $L^2$ -scalar product due to the regularity of our discrete fields (see Remark 1). The resulting matrix  $\mathbf{C} \in \mathbb{R}^{N_h \times N_h}$  is the smallest matrix that represents the correlation between the snapshots, and it is obviously dense, symmetric, and positive. The next step is to compute the eigenvalues  $\{\lambda_i\}_{i=1}^{N_h}$  ( $\lambda_i \in \mathbb{R}^+$ ) and

eigenvectors  $\{\xi_i\}_{i=1}^{N_h}$  ( $\xi_i \in \mathbb{R}^{N_h}$  and  $\|\xi_i\|_{L^2} = 1$ ) of  $\mathbf{C}$ , which correspond, respectively, to the squares of the singular values and the left singular vectors of  $\mathbf{S}$ :  $\mathbf{C} \xi_i = \lambda_i \xi_i$ ,  $1 \leq i \leq N_h$ . Finally, as mentioned above,  $\mathbf{B}_{rb}$  (introduced in Eq. (3)) is simply constructed with the  $N_{rb}$  vectors  $\{\xi_i\}_{i=1}^{N_{rb}}$  associated with the  $N_{rb}$  largest  $\{\lambda_i\}_{i=1}^{N_{rb}}$  (and, again, we expect  $N_{rb} \ll N_h$ ).

By construction,  $\mathbf{B}_{rb}$  is an orthonormal basis of  $V_{rb}$ , meaning that  $\mathbf{B}_{rb}^T \mathbf{B}_{rb} = \mathbf{I}_{N_{rb}}$ , where  $\mathbf{I}_{N_{rb}}$  is the identity operator (of dimension  $N_{rb}$ ). However,  $\mathbf{B}_{rb} \mathbf{B}_{rb}^T \neq \mathbf{I}_{N_h}$  since  $\mathbf{P}_{rb} \mathbf{u}_h = \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h$  only approximates  $\mathbf{u}_h$  due to the fact that  $V_{rb} \subset V_h$  (remind Eq. (8) and Fig. 1). An error estimation is actually available in this context. More precisely, let us define the space of semi-orthonormal<sup>1</sup> matrices of dimension  $N_{rb}$ :  $\mathcal{B}_{N_{rb}}^\perp = \{\mathbf{B} \in \mathbb{R}^{N_h \times N_{rb}}, \mathbf{B}^T \mathbf{B} = \mathbf{I}_{N_{rb}}\}$ . The interest of the POD matrix  $\mathbf{B}_{rb}$  is that it minimizes, over all possible  $N_{rb}$ -dimensional semi-orthonormal matrices  $\mathbf{B} \in \mathcal{B}_{N_{rb}}^\perp$ , the projection error (10) onto the whole set of snapshots; that is, the sum of the errors between all the snapshots  $\{\mathbf{u}_h(\mu_s)\}_{s=1}^{N_s}$  and their projection onto the associated subspace spanned by  $\mathbf{B}$ . Mathematically, this means that among the semi-orthonormal matrices of dimension  $N_{rb}$ , the POD one is the optimal one and it satisfies:

$$\sum_{s=1}^{N_s} \|\mathbf{u}_h(\mu_s) - \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h(\mu_s)\|_{L^2}^2 = \min_{\mathbf{B} \in \mathcal{B}_{N_{rb}}^\perp} \sum_{s=1}^{N_s} \|\mathbf{u}_h(\mu_s) - \mathbf{B}^T \mathbf{B} \mathbf{u}_h(\mu_s)\|_{L^2}^2 = \sum_{s=N_{rb}+1}^{N_h} \lambda_s \quad (13)$$

This result derives from the Schmidt-Eckart-Young theorem, see, *e.g.*, [19].

Benefiting from the above error estimation, the dimension of the RB can be fairly defined. In practice, we can compute the following ratio:

$$R(N_{rb}) = \frac{\sum_{i=1}^{N_{rb}} \lambda_i}{\sum_{i=1}^{N_h} \lambda_i}, \quad (14)$$

and choose a target expressed as  $(1 - \epsilon_{POD}^2)$ .  $N_{rb}$  is then defined as the minimal value such that:

$$R(N_{rb}) \geq (1 - \epsilon_{POD}^2), \quad (15)$$

meaning that the signal of the snapshots captured by the most significant  $N_{rb}$  POD modes is conserved (in the  $L^2$ -norm here) within  $(100 - \epsilon_{POD})\%$ . The chosen value for  $\epsilon_{POD}$  will be specified and accounted for in Section 4.

The construction of the RB is obviously performed during the offline phase. This actually constitutes the initial step of the present reduction method (refer to Algorithm 1 for an overview of the complete offline phase carried out in this work).

<sup>1</sup>Semi-orthonormal property denotes here the orthonormal property for rectangular matrices.

Once again, it is worth noting that the strategy employed so far can be applied to any  $\mu$ -parametrized model, solved by any discretization method. It only requires the  $N_s$  HR solutions to form the snapshot matrix  $\mathbf{S}$  (see Eq. (11)).

**Remark 2.** Let us mention that if  $N_h > N_s$ , the alternative correlation matrix  $\mathbf{C} = \mathbf{S}^T \mathbf{S}$  can be built to ensure minimal computational cost and memory storage. In this case, the left singular vectors of  $\mathbf{S}$  have to be computed from the eigenvectors of  $\mathbf{C}$ , see, *e.g.*, [19] for details.

**Remark 3.** It may also be noticed that other scalar products and associated norms like, *e.g.*, the  $H^1(\Omega)$ -norm, can be considered depending on the initial numerical model considered. To do so, one only has to modify the projection (8) and the correlation matrix (12) accordingly.

### 3.3. The POD-NN method

The goal now is to compute, for a new set of parameters  $\mu_{new} \in \mathcal{H}$  which does not belong to  $P_s$ , a good approximation of the complete solution  $u_h(\mu_{new}; x)$  (or equivalently of the complete solution vector  $\mathbf{u}_h(\mu_{new})$ ) in the reduced space  $V_{rb}$ . To achieve this, the novel method proposed in the present study relies on an enriched version of the POD-NN method first introduced in [14]. In the present section, we detail the original POD-NN method before enriching it in the next section.

The POD-NN method basic idea is to start from the projection defined in Eq. (8) (see also Fig. 1) and to learn, using a deep NN, the coefficients in the RB  $\Xi(x)$  of the projection of  $\mathbf{u}_h(\mu_{new})$  onto  $V_{rb}$ ; that is, to learn  $\mathbf{u}_h^{Nrb}(\mu_{new}) \in \mathbb{R}^{N_{rb}}$ , see Eq. (9). More precisely, let us introduce the following application:

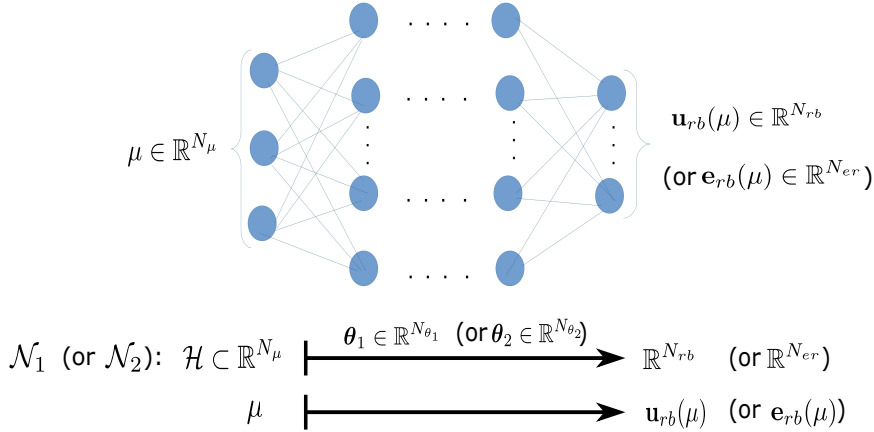
$$\begin{aligned} F_1 : \mathcal{H} \subset \mathbb{R}^{N_\mu} &\rightarrow \mathbb{R}^{N_{rb}} \\ \mu &\mapsto \mathbf{u}_h^{Nrb}(\mu) = \mathbf{B}_{rb}^T \mathbf{u}_h(\mu). \end{aligned} \quad (16)$$

It maps each set of input parameters  $\mu \in \mathcal{H}$  to the coefficients  $\mathbf{B}_{rb}^T \mathbf{u}_h(\mu)$  in the RB  $\Xi(x)$  of the projection of the complete solution  $\mathbf{u}_h(\mu)$  onto  $V_{rb}$ .

In the offline phase, this non-linear mapping  $F_1$  is approximated by employing an artificial NN, particularly a multi-layer perceptron (see Fig. 2). Artificial NNs have proved to be universal approximators, due to their rich interpolation space [15, 5]. In practice, training the aforementioned NN can be performed using supervised learning based on input-output pairs obtained from the snapshots  $\mathbf{S}$  (see Eq. (11)). Denoting by  $\mathcal{N}_1(\theta_1)(\mu)$  the considered NN with associated parameters (weights and biases) gathered in vector  $\theta_1$ , this leads to minimizing the following loss function for the training:

$$\mathcal{J}_1(\theta_1) = \frac{1}{N_s} \sum_{s=1}^{N_s} \|\mathcal{N}_1(\theta_1)(\mu_s) - \mathbf{B}_{rb}^T \mathbf{u}_h(\mu_s)\|_{L_2}^2, \quad (17)$$

where, once again, the  $L^2$ -norm is considered in our work. Training  $\mathcal{N}_1$  constitutes Step 9 of our offline phase, see Algorithm 1.



**Figure 2:** NN approximation of the map  $F_1$ , see Eq. (16) (or  $F_2$ , see Eq. (21)). Note that  $N_\mu$ ,  $N_{rb}$  and  $N_{er}$  are on the order of 10. In practice the hidden layers are taken with the same number of neurons and the latter is larger than the numbers of input and output neurons (see Section 4 for the exact chosen values).

**Remark 4.** It is important to notice at this stage that the reduction step, which transforms  $\mathbf{u}_h \in \mathbb{R}^{N_h}$  into  $\mathbf{u}_h^{N_{rb}} \in \mathbb{R}^{N_{rb}}$  with  $N_{rb} \ll N_h$  is crucial in the process. This step allows to obtain only a few output neurons for the NN (in our application  $N_{rb} = \mathcal{O}(10)$ , see Section 4), making it possible to consider a deep NN with a number of parameters  $N_{\theta_1}$  (i.e.,  $\theta_1 \in \mathbb{R}^{N_{\theta_1}}$ ) reasonable compared to the number of data (i.e., the  $N_s$  snapshots) during training. Of course, for the same reason, the number of input neurons of the network must be low, i.e., the number of physical parameters  $N_\mu$  ( $\mu \in \mathbb{R}^{N_\mu}$ ) must be small, which is also a limitation of more conventional reduced methods based on POD.

Next, during the online phase, all that is required is to perform a forward pass of the NN  $\mathcal{N}_1$  given a new (unseen) set of physical parameters  $\mu_{new}$  and the optimized  $\theta_1$  obtained after training. This process computes the reduced coefficients  $\mathbf{u}_{rb}(\mu_{new})$  and is known to be very fast, as the forward evaluation of a NN simply involves affine and element-wise activation transformations. This makes the online phase achievable in real-time. Finally, the reduced solution can be written in the original basis  $\Phi(x)$  using Eq. (6):  $\mathbf{u}_h^{N_h}(\mu_{new}) = \mathbf{B}_{rb} \mathbf{u}_{rb}(\mu_{new}) \in \mathbb{R}^{N_h}$  (which is also very fast to perform). In particular, this allows the utilization of existing subroutines from the initial computational code for post-processing. The entire online phase performed in this work is summarized in Algorithm 2. The procedures mentioned above correspond to Steps 1 and 2 of the online phase.

**Remark 5.** In the proposed method, we ultimately only need to utilize the initial computational code associated with the HR model  $\mathcal{M}_\mu$  in a black-box fashion to run it multiple times ( $N_s$  times) with different parameters. Subsequently, the method is entirely independent of the considered HR model  $\mathcal{M}_\mu$  and underlying code since, in particular, the online

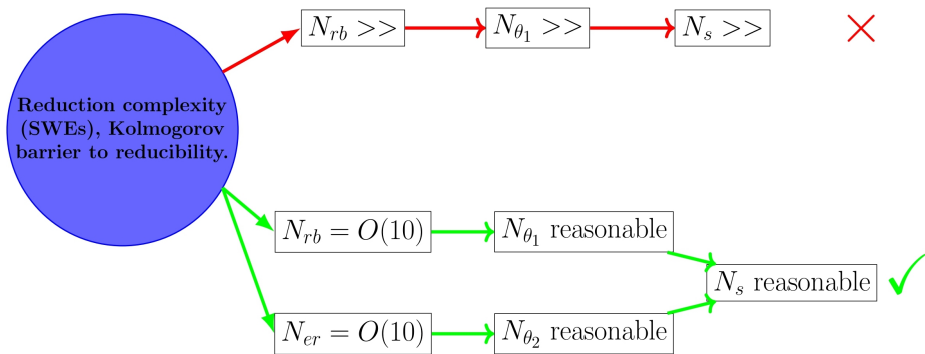
phase is decoupled from  $\mathcal{M}_\mu$ . Consequently, the method has the twofold advantages of being generic (it can be applied to any type of problem solved by any discretization method), and non-intrusive (we never need to delve into the code itself). Here, we leverage on these properties to apply the strategy to a non-linear problem with a non-affine dependence on the parameters and where a Galerkin approach is not used at the discretization level, which is a well-known scenario in which traditional reduced-order methods fail.

### 3.4. The Error-Aware POD-NN method

The POD-NN method considered so far requires the ability to strongly reduce the dimension of the problem. More precisely, we need to have  $N_{rb} = \mathcal{O}(10)$ , so that it is possible to employ a deep NN with a number of parameters  $N_{\theta_1}$  reasonable compared to the number snapshots  $N_s$  (see Fig. 3 that illustrates the different constraints). Therefore, in some situations, especially in the case of convection-dominated flow problems, it is requested to further enhance the accuracy of the method to mitigate the Kolmogorov barrier to reducibility (see, *e.g.*, [2] for the concept of reduction complexity). To achieve this in a straightforward manner with minimal computational overhead, we propose to integrate the ROM error estimation. Note that this generic idea has been developed, *e.g.*, in [26] within the framework of Gaussian processes. The idea is here to learn, in addition to the projection of  $\mathbf{u}_h(\mu)$  onto  $V_{rb}$ , the projection error introduced in Eq. (10) (see also Fig. 1):

$$\mathbf{e}_h(\mu) = \mathbf{u}_h(\mu) - \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h(\mu), \quad (18)$$

and to benefit from the latter to correct the reduced solution obtained so far.



**Figure 3:** Due to a high reduction complexity for convection dominated problems, see, *e.g.*, [2] for the concept of the Kolmogorov barrier to reducibility, it is requested to complement the basic POD-NN method with the projection error learning. This allows to limit the sizes of the reduced bases (here  $N_{rb} = \mathcal{O}(10)$  and  $N_{er} = \mathcal{O}(10)$ ), thus to use smaller NNs, and therefore smaller datasets (here  $N_s$  snapshots).

For this purpose, we follow the same procedure used to construct the reduced solution, but this time starting with the snapshot error matrix  $\mathbf{E} \in \mathbb{R}^{N_h \times N_s}$ :

$$\mathbf{E} = \left[ \mathbf{e}_h(\mu_1) \mid \dots \mid \mathbf{e}_h(\mu_{N_s}) \right], \quad (19)$$

which collects the projection errors of the already computed snapshots  $\{\mathbf{u}_h(\mu_s)\}_{s=1}^{N_s}$  (see Eq. (11)). Consequently, we first apply the POD strategy (same operations as in Section 3) to compute the corresponding error RB  $\{\zeta_n(x)\}_{n=1}^{N_{er}}$  (where we again anticipate  $N_{er} \ll N_h$ ) or, in practice, the error RB matrix  $\mathbf{B}_{er} \in \mathbb{R}^{N_h \times N_{er}}$ :

$$\mathbf{B}_{er} = \left[ \zeta_1 \mid \dots \mid \zeta_{N_{er}} \right]. \quad (20)$$

Note that  $\mathbf{B}_{er}$  also serves as the change-of-variable matrix from basis  $\{\zeta_n(x)\}_{n=1}^{N_{er}}$  to the initial complete one  $\Phi(x)$ . At this stage, similarly as in Eq. (16), we introduce the following application:

$$\begin{aligned} F_2 : \mathcal{H} \subset \mathbb{R}^{N_\mu} &\rightarrow \mathbb{R}^{N_{er}} \\ \mu &\mapsto \mathbf{e}_h^{N_{er}}(\mu) = \mathbf{B}_{er}^T \mathbf{e}_h(\mu), \end{aligned} \quad (21)$$

which maps each set of input parameters  $\mu \in \mathcal{H}$  to the coefficients  $\mathbf{B}_{er}^T \mathbf{e}_h(\mu)$  of the projection of the error  $\mathbf{e}_h(\mu)$  onto the subspace spanned by  $\{\zeta_n(x)\}_{n=1}^{N_{er}}$ . The remaining task of the extended offline phase is therefore to train a new artificial NN, denoted by  $\mathcal{N}_2(\theta_2)(\mu)$ , to approximate the non-linear mapping  $F_2$  (refer to Fig. 2 once again). As for the reduced solution, the latter is performed by minimizing the following loss function:

$$\mathcal{J}_2(\theta_2) = \frac{1}{N_s} \sum_{s=1}^{N_s} \|\mathcal{N}_2(\theta_2)(\mu_s) - \mathbf{B}_{er}^T \mathbf{e}_h(\mu_s)\|_{L^2}^2. \quad (22)$$

For a better understanding and conciseness, the additional operations to learn the projection error in the offline phase are outlined in Algorithm 1 (see Steps 6, 7 and 9).

Next, during the online phase, it is possible to estimate the projection error at a new set of physical parameters  $\mu_{new}$  by simply performing a forward pass of the NN  $\mathcal{N}_2$  using the optimized  $\theta_2$  obtained after training. The reduced projection error can also be written in the original basis  $\Phi(x)$  such as:  $\mathbf{e}_{rb}^{N_h}(\mu_{new}) = \mathbf{B}_{er} \mathbf{e}_{rb}(\mu_{new}) \in \mathbb{R}^{N_h}$ . In this basis, we eventually complement the former reduced solution  $\mathbf{u}_{rb}^{N_h}(\mu_{new})$  of Section 3 with the present projection error, which

yields the final corrected reduced solution as follows:

$$\left( \mathbf{u}_{rb}^{N_h}(\mu_{new}) + \mathbf{e}_{rb}^{N_h}(\mu_{new}) \right) = \tilde{\mathbf{u}}_{rb}^{N_h}(\mu_{new}) \approx \mathbf{u}_h(\mu_{new}). \quad (23)$$

The overall online phase is outlined in Algorithm 2; more precisely, Steps 3 to 5 constitute the additional operations to perform to correct the initial reduced solution with a learning of the projection error. This phase can be carried out very quickly since it merely consists in performing forward passes of trained NNs.

As mentioned in the introduction, because our approach incorporates NNs with POD and takes into account the projection error, we refer to it as an *Error-Aware POD-NN reduction method*. Complementing the POD-NN strategy with the projection error enables to attain higher accuracy while preserving a limited number of POD modes. This is not equivalent as setting the basic POD-NN method with  $(N_{rb} + N_{er})$  modes. Indeed, the two NN outputs are uncorrelated; more importantly, the Error Aware POD-NN method enables to limit the output dimension of the NNs, thus providing

the opportunity to employ smaller NNs (*i.e.* with less parameters), and therefore smaller datasets (the snapshots) to train the NNs (see Fig. 3).

---

**Algorithm 1:** The Error-Aware POD-NN reduction method: the offline phase

---

**function**  $[\mathbf{B}_{rb}, \mathcal{N}_1(\theta_1), \mathbf{B}_{er}, \mathcal{N}_2(\theta_2)] = \text{POD-NN-OFFLINE}(N_s, \varepsilon_{POD})$ .

1. Generate the reference parameters values by sampling the hypercube  $\mathcal{H}$ :  $P_s = \{\mu_s\}_{s=1}^{N_s} \in (\mathcal{H})^{N_s}$ .
  2. Compute the associated complete numerical solutions with the HR model and form the snapshot matrix:  

$$\mathbf{S} = [\mathbf{u}_h(\mu_1) | \dots | \mathbf{u}_h(\mu_{N_s})] \in \mathbb{R}^{N_h \times N_s}.$$
  3. Build up the correlation matrix  $\mathbf{C} = \mathbf{S} \mathbf{S}^T \in \mathbb{R}^{N_h \times N_h}$ .
  4. Compute its eigenelements  $(\lambda_i, \xi_i)_{i=1}^{N_h}$ , with  $\lambda_i \in \mathbb{R}_+$  and  $\xi_i \in \mathbb{R}^{N_h}$  ( $\|\xi_i\|_{L^2} = 1$ ):  

$$\mathbf{C} \xi_i = \lambda_i \xi_i, \quad 1 \leq i \leq N_h.$$
  5. Take the  $N_{rb}$  vectors  $\{\xi_i\}_{i=1}^{N_{rb}}$  associated with the  $N_{rb}$  largest  $\{\lambda_i\}_{i=1}^{N_{rb}}$  (by using criterion (15) based on  $\varepsilon_{POD}$ ) to form the RB matrix:  $\mathbf{B}_{rb} = [\xi_1 | \dots | \xi_{N_{rb}}] \in \mathbb{R}^{N_h \times N_{rb}}$  with  $N_{rb} \ll N_h$ .
  6. By making use of the POD projection error defined as  $\mathbf{e}_h(\mu) = \mathbf{u}_h(\mu) - \mathbf{B}_{rb} \mathbf{B}_{rb}^T \mathbf{u}_h(\mu)$ , form the snapshot error matrix:  $\mathbf{E} = [\mathbf{e}_h(\mu_1) | \dots | \mathbf{e}_h(\mu_{N_s})] \in \mathbb{R}^{N_h \times N_s}$ .
  7. Apply the POD procedure to  $\mathbf{E}$  (see Steps 3 to 5 above) to compute the error RB matrix:  

$$\mathbf{B}_{er} = [\zeta_1 | \dots | \zeta_{N_{er}}] \in \mathbb{R}^{N_h \times N_{er}} \quad \text{with} \quad N_{er} \ll N_h.$$
  8. Train artificial NN  $\mathcal{N}_1$  which approximates application  $F_1$  defined in (16), by minimizing the loss function  

$$\mathcal{J}_2(\theta_1) = \frac{1}{N_s} \sum_{s=1}^{N_s} \|\mathcal{N}_1(\theta_1)(\mu_s) - \mathbf{B}_{rb}^T \mathbf{u}_h(\mu_s)\|_{L^2}^2.$$
  9. Train artificial NN  $\mathcal{N}_2$  which approximates application  $F_2$  defined in (21), by minimizing the loss function  

$$\mathcal{J}_2(\theta_2) = \frac{1}{N_s} \sum_{s=1}^{N_s} \|\mathcal{N}_2(\theta_2)(\mu_s) - \mathbf{B}_{er}^T \mathbf{e}_h(\mu_s)\|_{L^2}^2.$$
- 

**Algorithm 2:** The Error-Aware POD-NN reduction method: the online phase (real-time computations)

---

**function**  $[\mathbf{u}_{rb}^{N_h}(\mu_{new})] = \text{POD-NN-ONLINE}(\mu_{new}; \mathbf{B}_{rb}, \mathcal{N}_1(\theta_1), \mathbf{B}_{er}, \mathcal{N}_2(\theta_2))$ .

1. Given a new set of input physical parameters gathered in  $\mu_{new}$ , and taking the optimal  $\theta_1$  obtained from Algorithm 1 (Step 9), evaluate the output  $\mathbf{u}_{rb}(\mu_{new})$  of artificial NN  $\mathcal{N}_1$ .
  2. Deduce the reduced solution in basis  $\Phi(x)$ :  $\mathbf{u}_{rb}^{N_h}(\mu_{new}) = \mathbf{B}_{rb} \mathbf{u}_{rb}(\mu_{new}) \in \mathbb{R}^{N_h}$ .
  3. Given the same new set of input parameters  $\mu_{new}$ , and taking the optimal  $\theta_2$  obtained from Algorithm 1 (Step 10), evaluate the output  $\mathbf{e}_{rb}(\mu_{new})$  of artificial NN  $\mathcal{N}_2$ .
  4. Deduce the reduced projection error in basis  $\Phi(x)$ :  $\mathbf{e}_{rb}^{N_h}(\mu_{new}) = \mathbf{B}_{er} \mathbf{e}_{rb}(\mu_{new}) \in \mathbb{R}^{N_h}$ .
  5. Deduce the final corrected reduced solution:  $\tilde{\mathbf{u}}_{rb}^{N_h}(\mu_{new}) = \mathbf{u}_{rb}^{N_h}(\mu_{new}) + \mathbf{e}_{rb}^{N_h}(\mu_{new})$ .
-



#### 4. Numerical experiments and discussions

In this section, we numerically evaluate the proposed EA-POD-NN method on a real-world flooding event of relatively high magnitude occurred on the Aude River in the southeastern region of France in 2018. The flood hydraulic model was built in [11] based on HR terrain elevation data provided by IGN (French geographical national institute) and hydrometeorological data from SCHAPI (French national flood forecasting center) and Météo France. The [DassFlow 2D](#) software [22, 18] is used for numerically solving the 2D SWEs (see Section 2.2) over an unstructured mesh covering the spatial domain  $\Omega$  that contains a floodplain located at the confluence of the Aude and Fresquel rivers. The latter is inflowed by two real hydrographs  $Q_{in,k}(\mu, t)$  at  $k = 2$  upstream points on  $\partial\Omega$  the border of  $\Omega$  (see Fig. 4).

Again, the primary focus is on the  $\mu$  parameter related to the inflow discharge functions which is the most important factor controlling flooding dynamics, particularly crucial in operational context where fast and accurate hydraulic simulations of flooding are needed after hydro-meteorological forecasts of discharge. In this context, two test cases are carried out, representative of real complexity of concomitant or not flood inflows:

1. Case with simultaneous inflows where the classic POD-NN method only is sufficient to reduce the problem.
2. Case with non-simultaneous inflows, hence which introduces more complex non-linear waves interactions, where the POD-NN method struggles to accurately reproduce the solution. This is where the proposed projection error correction approach becomes essential to enhance the accuracy of the ROM.

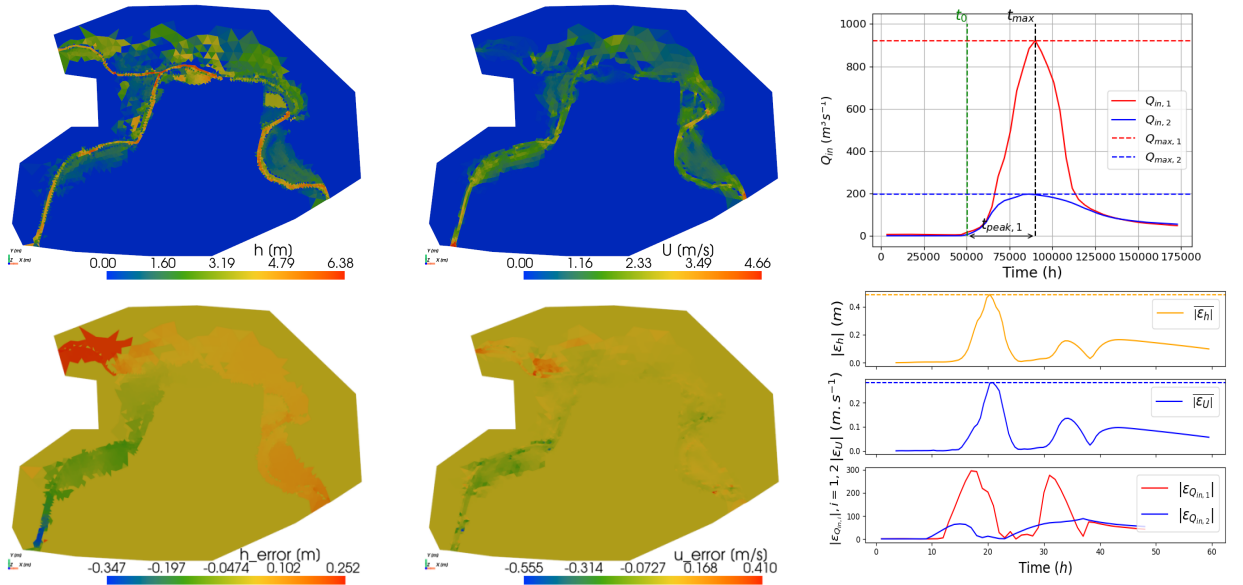
Therefore, the investigated test cases demonstrate the limitations of the classic POD-NN method for reducing a SW model and showcase the significant improvement in the ROM's quality achieved through the proposed EA-POD-NN approach.

Let us introduce at this stage some notations for the physical and numerical quantities that will be used throughout this section, especially in the majority of the figures:

- $U = \sqrt{v_x^2 + v_y^2}$  denotes the velocity norm in  $(ms^{-1})$ .
- $h_{ref}$  in  $(m)$  (resp.  $U_{ref}$  in  $(ms^{-1})$ ) denotes the reference 2D hydraulic model water depth (resp. the reference 2D hydraulic model velocity norm).
- $h_{\square}$  in  $(m)$  (resp.  $U_{\square}$  in  $(ms^{-1})$ ) denotes the approximated water depth (resp. the approximated velocity norm), where  $\square$  stands for POD-NN or EA-POD-NN depending on the considered method for the computation.
- $\varepsilon_h(x, t) = h_{ref} - h_{\square}$  (resp.  $\varepsilon_U(x, t) = U_{ref} - U_{\square}$ ) denotes the difference between  $h_{ref}$  (resp.  $U_{ref}$ ) and  $h_{\square}$  (resp.  $U_{\square}$ ).
- Finally,  $\overline{\varepsilon_h}(t)$  (resp.  $\overline{\varepsilon_U}(t)$ ) denotes the spatial average over  $\Omega$  of the error on water depth (resp. of the error on velocity norm).

#### 4.1. General design of the numerical experiments

To start with, we outline here the design of the numerical experiments considered to reduce the 2D hydraulic simulation of the 2018 flood event that occurred in the southeastern region of France. This event was characterized by intense rainfall on upstream catchments, river overflow, and extensive flooding over the studied domain  $\Omega$ . The maximal submersion depths and flow velocity norm over the floodplain simulated by the reference 2D hydraulic model are shown in Fig 4(Top), along with the real hydrographs from gauging stations that are used as inflows. A  $T = 60$ h simulation time period is chosen for the reference 2D SW model numerical solution, to enable sufficient time for flood propagation and recession over the domain  $\Omega$ .



**Figure 4:** (Top) The reference 2D hydraulic model of the 2018 flood on the Aude River domain  $\Omega$  at  $t = 25.2$ h around maximum flooding. (Left) The water depth  $h_{ref}$  in (m). (Middle) The velocity norm  $U_{ref}$  in ( $m s^{-1}$ ). (Right) The real input hydrographs from gauging stations; (Red) for the Aude river  $Q_{in,1}$  with  $t_{peak,1} = 11.94$ h and  $Q_{max,1} = 922 m^3 s^{-1}$ , (Blue) for the Fresnel river  $Q_{in,2}$  in  $m^3 s^{-1}$  with  $t_{peak,2} = 9.16$ h and  $Q_{max,2} = 198 m^3 s^{-1}$ ; the black line corresponds to maximal flooding time at which spatial maps are plotted. (Bottom) Comparison between the reference 2D hydraulic model inflowed by the real ("ref") or triangular ("tri") input hydrographs, in terms of (Left) difference  $\epsilon_h$  on water depth  $h$ , (Middle)  $\epsilon_U$  on velocity norm  $U$ . (Right) Temporal variation of the spatially averaged absolute difference  $|\epsilon_h|$  for  $h$ ,  $|\epsilon_U|$  for  $U$ , and the errors  $\epsilon_{Q_{in,1}}$  and  $\epsilon_{Q_{in,2}}$  of reproduction of real inflow hydrographs with triangular ones.

##### 4.1.1. Representation of the inflows signals

A necessary condition to apply the reduction method without additional pre-processing is to have an input parameter  $\mu$  of quite small dimension:  $N_\mu = \mathcal{O}(10)$  (see sections 2 and 3). Hence, an effective representation of inflows is performed using a minimal number of key parameters: the maximum peak flow discharge  $Q_{max}$  ( $m^3 s^{-1}$ ), and the corresponding time of peak  $t_{peak}$  (h), defined as  $t_{peak} = (t_{max} - t_0)$  the duration between  $t_0$  the onset of rising water levels and  $t_{max}$  the corresponding attainment of  $Q_{max}$ , see Fig. 4 (Top right) for the first inflow (red curve). Hence

we have  $Q_{in,k}(t_{\text{peak},k}, Q_{\text{max},k}; t)$  for  $k = 1, 2$ . The concept of utilizing  $Q_{\text{max}}$  and  $t_{\text{peak}}$  as representative parameters is rooted in their ability to characterize the fundamental aspects of flood initiation.  $Q_{\text{max}}$  quantifies the peak discharge intensity, reflecting the maximum flow rate reached during the flood. On the other hand,  $t_{\text{peak}}$  encapsulates the time it takes for the river to transition from its initial state to the point of maximum discharge, offering insights into the flood's temporal evolution. This simple parameterization enables to represent the inflow hydrographs while preserving their essential features, *i.e.* peak flow magnitude and timing, therefore ensuring consistent flood inundation modeling as shown by the limited errors in space and time on submersion depth and velocity norm between the reference 2D SW model run with real or triangular hydrographs, see Fig. 4(Bottom). In particular, the errors are relatively lower at peak time and maximum flooding rather than during more dynamic phases of submersion and recessions *i.e.* corresponding to hydrograph rising limb and also during recession.

As mentioned previously, two test cases are considered with different input parameters, corresponding to increasingly complex inflow signals and resulting SW model responses over the floodplain:

1. Simultaneous inflow hydrographs:  $\mu = (t_{\text{peak}}, Q_{\text{max},1}, Q_{\text{max},2}; t_{sn})$ , so  $N_\mu = 4$ .
2. Non-simultaneous inflow hydrographs:  $\mu = (t_{\text{peak},1}, Q_{\text{max},1}, t_{\text{peak},2}, Q_{\text{max},2}; t_{sn})$ , so  $N_\mu = 5$ .

For both cases, the different steps of Algorithm 1 are performed to achieve the offline phase of the reduction method (see Section 3). Case-specific parameters values are detailed for each experiment in the following paragraphs.

#### 4.1.2. Snapshots matrix construction

The starting point of the learning phase (offline phase) is the set of parameters  $P_s$  and the corresponding snapshots matrix  $\mathbf{S}$ . Here, the array  $P_s = \{\mu_s\}_{s=1}^{N_s}$  of  $N_s$  parameters sets is obtained by uniformly sampling  $M$  times the discharge related parameters in  $\mu$  considering upper and lower bounds around the reference values of the 2018 real flood hydrographs (the detailed values will be specified later). The last parameter  $t_{sn}$  defines the  $N_t$  physical times at which the snapshots of the reference 2D SW model are computed. It is fixed *a priori* in function of the response dynamics of the physical model to reduce. Overall, we have  $N_s = M \times N_t$ .

**Remark 6.** It is important to note that the snapshot time parameter  $t_{sn}$  defines the temporal repartition of the used physical model outputs which is therefore conditioning the capture of information in the reduction process of the physical model. Here hydraulic simulations were performed over a span of 60 hours. However, in the context of SW model reduction, we select the time interval starting from the point corresponding to the onset of rising water levels for the inflows. This selection allows us to concentrate on the period where the flood dynamics exhibit significant changes, aligning with the goals of SW model reduction. Also, this approach ensures that the NNs focus on capturing meaningful flow variations that occur starting from the rising phase.

Next, for each pair of inflow hydrographs  $Q_{in;k=1,2}(\mu_s, t)$  resulting from the parameter set  $\{\mu_s\}_{s=1}^{N_s}$ , the  $\mu$ -parameterized hydraulic model  $\mathcal{M}_\mu$  is run to generate the HR solution  $\mathbf{u}_h(\mu_s)$ . In the discrete setting, the dimension of the output vectors  $\mathbf{u}_h(\mu_s) = [\mathbf{h}^s, \mathbf{v}_x^s, \mathbf{v}_y^s]^T$  is  $N_h = 3 \times N_x = 21801$  where  $N_x = 7267$  counts the number of cells of the mesh covering  $\Omega$ . Note eventually that each model output  $\mathbf{h}$ ,  $\mathbf{v}_x$  or  $\mathbf{v}_y$  consists in  $N_t$  temporal snapshots of model outputs spatial fields. This enables to obtain the  $(N_h \times N_s)$  snapshot matrix  $\mathbf{S}$  as follows:

$$\mathbf{S} = [\mathbf{h}, \mathbf{v}_x, \mathbf{v}_y]^T = \begin{bmatrix} \mathbf{h}(\mu_1; x) & \cdots & \mathbf{h}(\mu_{N_s}; x) \\ \mathbf{v}_x(\mu_1; x) & \cdots & \mathbf{v}_x(\mu_{N_s}; x) \\ \mathbf{v}_y(\mu_1; x) & \cdots & \mathbf{v}_y(\mu_{N_s}; x) \end{bmatrix}.$$

Once  $\mathbf{S}$  is built, one can perform the other steps of the offline phase for the POD-NN and EA-POD-NN algorithms (see Algorithm 1).

#### 4.1.3. RB matrix construction

The second step involves constructing the reduced matrix  $\mathbf{B}_{rb} \in \mathbf{R}^{N_{rb} \times N_h}$  by applying the POD method to the snapshots matrix  $\mathbf{S}$ . Indeed, since we have  $N_h \leq N_s$ , we compute the POD modes of the correlation matrix  $\mathbf{C} = \mathbf{S}\mathbf{S}^T$ , see Eq. (12). To truncate the first  $N_{rb}$  modes, we ask to conserve 99.99% of the energy of the system. In other words, given a tolerance  $\epsilon_{POD}^2 = 10^{-2}$  and by using Eq. 15, we obtain the reduced dimension  $N_{rb} = 5$ . Following the same approach for the error matrix  $\mathbf{E}$ , by taking  $\epsilon_{er}^2 = 10^{-1}$ , the reduced dimension for errors is  $N_{er} = 17$ .

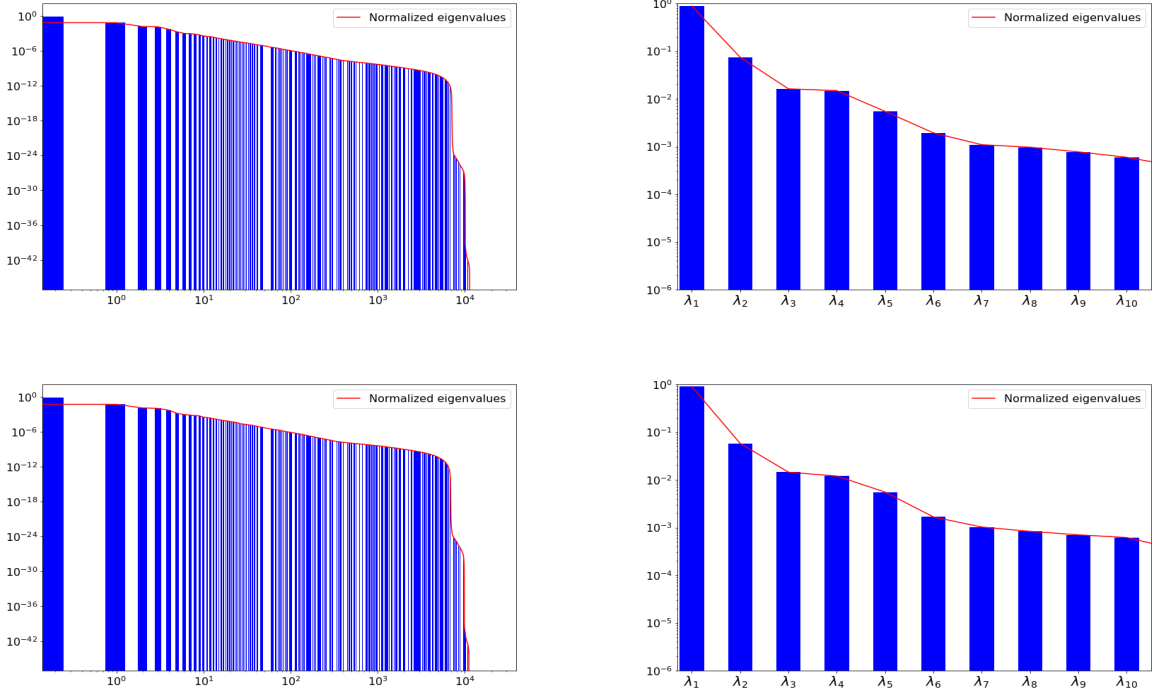
The plot in log-scale of the eigenvalues of the correlation matrix  $\mathbf{C}$  is given in Fig. 5. Note that the behaviour of the eigenvalues of the correlation matrix of the errors ( $\mathbf{C}_{er} = \mathbf{E} \cdot \mathbf{E}^T$ ) is similar for both tested cases with simultaneous and non-simultaneous inflows.

**Remark 7.** The Kolmogorov  $N_{rb}$ -width [8, 19] enables to measure the degree of reductibility of model. A link between the Kolmogorov  $N_{rb}$ -width and the POD model construction is presented in [19]. This link relies on the decay of the eigenvalues. In Fig. 5, we observe a slow decay of the eigenvalues, which is typical in the case of non-linear convection-dominated problems [2]. That is, the error in the right hand side of Eq.(13) is large. This turns the SWEs reduction problem to be challenging.

**Remark 8.** For the EA-/POD-NN approach, certain numerical constraints must be adhered to. These constraints are summarized in Tab 2.

## 4.2. Case with simultaneous inflows

Let us now focus on the results obtained for the first test case involving simultaneous inflow hydrographs.



**Figure 5:** (Left) The decay of eigenvalues of the correlation matrix  $\mathbf{C}$ . (Right) The zoom on the first 10 eigenvalues. (Top) For the case with simultaneous hydrographs ( $N_\mu = 4$ ). (Bottom) For the case with non-simultaneous hydrographs ( $N_\mu = 5$ ).

Constraint	POD-NN Approach
Case 1: $N_s < N_h$	Correlation matrix $\mathbf{C} = \mathbf{S}^T \mathbf{S}$ of size $N_s \times N_s$
Case 2: $N_s > N_h$	Correlation matrix $\mathbf{C} = \mathbf{S} \mathbf{S}^T$ of size $N_h \times N_h$
Constraint 3:	$N_\theta < N_s$
Constraint 4:	$N_L > N_{rb}$

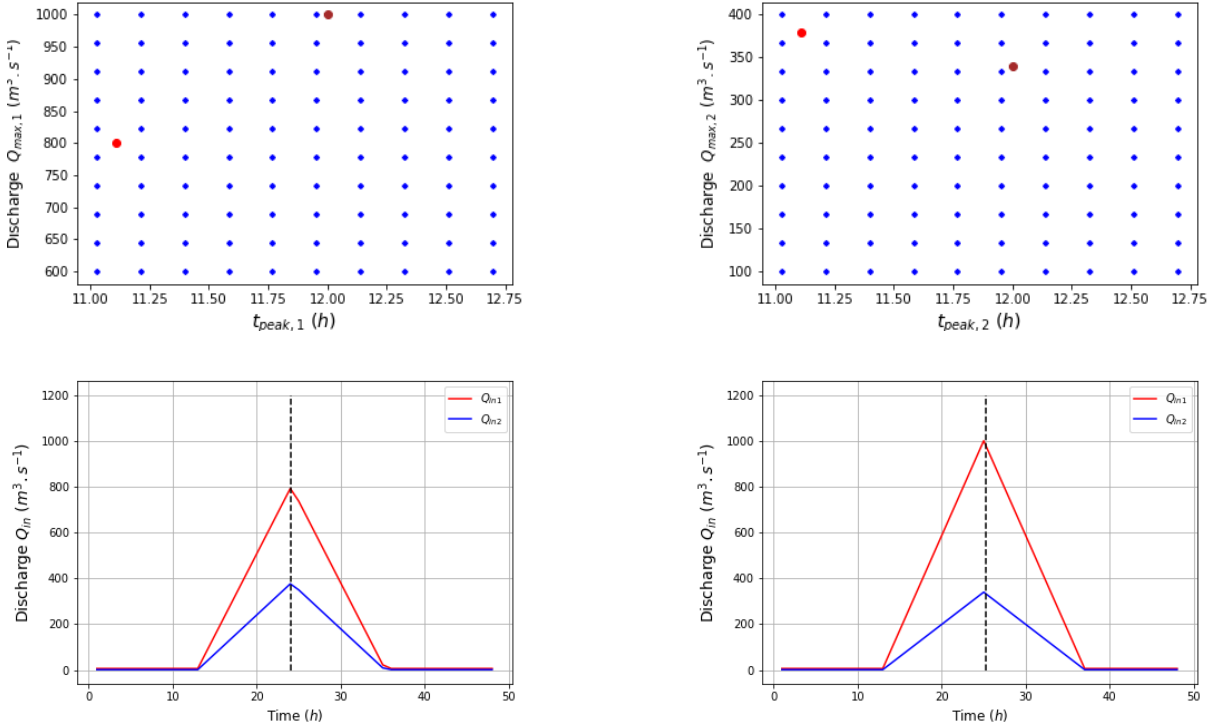
**Table 2**

The conditions that must be adhered to in the the offline phase for both numerical test cases. These conditions concern the following parameters:  $N_s$  representing the number of snapshots,  $N_{\text{param}}$  denoting the number of parameters of the NN, and  $N_L$  indicating the number of neurons in the last layer of the NN.

#### 4.2.1. The parameter space

As mentioned above, the same  $t_{\text{peak}}$  for both inflows is considered here, so that the parameter reads  $\mu = (t_{\text{peak}}, Q_{\text{max},1}, Q_{\text{max},2}, t_{\text{sn}})$  and is of dimension  $N_\mu = 4$ . Following previous discussions, the parameters ranges are defined as follows:  $t_{\text{peak}} \in [11.2, 12.69]$ ,  $Q_{\text{max},1} \in [600, 1200]$ , and  $Q_{\text{max},2} \in [100, 400]$ . Then, the snapshots time input parameter is chosen as  $t_{\text{sn}} \in [3.6, 60]$ . Recapitulating, one has  $\mathcal{H} = [11.2, 12.69] \times [600, 1200] \times [100, 400] \times [3.6, 60]$ . A visual representation of the parameter set sample  $P_s = \{\mu_s\}_{s=1}^{N_s}$  with a focus around  $t_{\text{peak}}$ ,  $Q_{\text{max},1}$ , and  $Q_{\text{max},2}$  is given in Fig. 6.

The snapshot matrix  $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$  is constructed with a number of snapshots fixed as  $N_s = M \times N_t = 189\,000$  (with  $M = 1\,000$  simulations using the DassFlow software and  $N_t = 189$  writing times, *i.e.* a  $dt_w = 18\text{min}$  writing time step). Note that quasi similar results could be obtained with  $dt_w = 36\text{min}$ , as will be chosen for the second case with non-simultaneous hydrographs because of memory limitations. Note finally that such writing time steps remain significantly lower than the flood propagation time that is around  $3h$  over the studied zone.



**Figure 6:** Simultaneous inflows case. (Top) The sampled parameter sets  $P_s$ , blue points for offline phase, red and brown points being random parameter sets that will be used for online prediction; (left) For the first hydrograph  $Q_{in,1}$  on the Aude River and (right) for the second hydrograph  $Q_{in,2}$  on the Fresquel River (corresponding to red and blue hydrographs, respectively, in Fig. 4). (Bottom) Input inflow hydrographs for (left) red and (right) brown points that will be used for online prediction (vertical dashed lines correspond to the time of maximum flooding at which spatial error analysis will be performed).

#### 4.2.2. The training phase (during the offline phase)

For the training of the NNs, the data of size  $N_s = 189\,000$  is classically partitioned into 80% for the actual training, 10% for the test and 10% for the validation. To ensure that the number of NNs parameters remains small relative to the data dimension, we determine the number of parameters of the NN  $\mathcal{N}_1$  as 3% (resp. of the NN  $\mathcal{N}_2$  as 5%) of the training data dimension  $N_s$ , yielding  $N_{\theta_1} = 4405$  for  $\mathcal{N}_1$  with  $N_{rb} = 5$  for the output layer, and  $N_{\theta_2} = 9977$  with  $N_{er} = 17$  for the output layer of  $\mathcal{N}_2$  in the case of simultaneous inflows (resp.  $N_{er} = 16$  in the case of non simultaneous inflows). The number of hidden layers for  $\mathcal{N}_1$  is 10 with 20 neurons per layer and for  $\mathcal{N}_2$  it is 10 with 30 neurons per layer. The

chosen activation function is 'RELU'. We train the NNs over 10000 epochs which leads to  $\varepsilon_{NN} = 10^{-7}$  in terms of loss function. The loss function on the validation data set is observed to well decrease too. An *Adam* optimizer [16] is used for the gradient descent to minimize the loss function. The latter computes the loss between the HR projected output  $\mathbf{u}_h^{N_{rb}}(\mu_s) = \mathbf{B}_{rb}^T \mathbf{u}_h(\mu_s)$  (resp.  $\mathbf{e}_h^{N_{er}}(\mu_s) = \mathbf{B}_{er}^T \mathbf{e}_h(\mu_s)$ ),  $1 \leq s \leq N_s$ , and the NN output  $\mathcal{N}_1(\theta_1)(\mu_s)$  (resp.  $\mathcal{N}_2(\theta_2)(\mu_s)$ ) using the Mean Square Error (MSE), in line with Eqs. (17) and (22).

#### 4.2.3. The online prediction phase

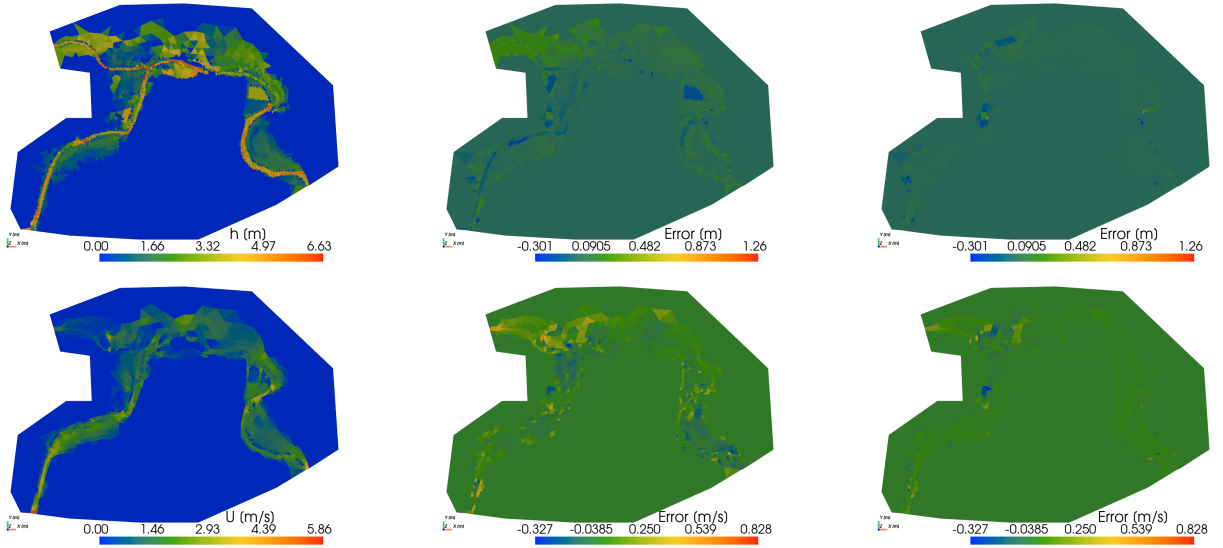
After successfully completing the training phase and optimizing the NNs parameters, we move to the prediction stage, where the model demonstrates its ability to rapidly forecast hydraulic variables  $(h, v_x, v_y)(x, t)$  for new parameter values, denoted as  $\mu_{new}$ . These new parameter values are in the parameter space range ( $\mu_{new} \in \mathcal{H}$ ) but distinct from all the points in  $P_s$  (let us recall that these new parameter values correspond to the red and brown points in Fig. 6 (Top), which are different from the blue points that form  $P_s$ ). In contrast, the chosen parameter  $t_{sn}$  defines the times at which the snapshots are performed in the offline phase and the prediction time in online phase. These two points  $\mu_{new}$  are carefully selected: the first (brown point in Fig. 6(Top)) is chosen to be in close proximity to the training phase data points, while the second (red point in Fig. 6(Top)) is taken at a considerable distance from the training data points. This deliberate selection allows us to rigorously evaluate the predictive capabilities and precision of our approximation model. The performances obtained in reducing the 2D SW model are depicted in their spatio-temporal dimensions with several graphes (see Figs. 7 to 10), especially at maximal flooding which is a crucial instant for operational forecasting, with related spatial statistics of  $\overline{\varepsilon}_h(x, t_{peak})$  and  $\overline{\varepsilon}_U(x, t_{peak})$ , but also in terms of temporal variability of  $\overline{\varepsilon}_h(t)$  and  $\overline{\varepsilon}_U(t)$ .

More precisely, spatial performances of the online solution  $\mathbf{u}_{\square}(\mu_{new})$  at maximum flooding are first shown in Figs. 7 and 8 for  $\mu_{new,1} = (11.11, 800, 380; 24)$  (red point in Fig. 6) and  $\mu_{new,2} = (12, 1000, 340; 25.2)$  (brown point in Fig. 6). Then, the spatial values of misfit to the reference 2D SW simulation  $\mathbf{u}_h(\mu_{new})$  are depicted at peak time in histograms in Fig 9 and in terms of Cumulative Distribution Function (CDF) of their spatial variability in Fig. 10. In particular, the median value of  $|\varepsilon_h(x, t_{peak})|$  (or  $|\varepsilon_U(x, t_{peak})|$ ) is  $10^{-3} m$  (or  $0.7 \times 10^{-3} ms^{-1}$ ) with EA-POD-NN and  $3 \times 10^{-3} m$  (or  $10^{-3} ms^{-1}$ ) with POD-NN for  $\mu_{new,1}$ . Also, the median value of  $|\varepsilon_h(x, t_{peak})|$  (or  $|\varepsilon_U(x, t_{peak})|$ ) is  $2 \times 10^{-3} m$  (or  $10^{-3} ms^{-1}$ ) with EA-POD-NN and  $5 \times 10^{-3} m$  (or  $4 \times 10^{-3} ms^{-1}$ ) with POD-NN for  $\mu_{new,2}$ .

Next, the temporal variation of errors spatial averages  $|\overline{\varepsilon}_h|(t)$  (or  $|\overline{\varepsilon}_U|(t)$ ) in Fig. 10 shows that submersion depth (or flow velocity magnitude) stays below  $0.1m$  (or  $0.1ms^{-1}$ ) over the whole spatio-temporal domain for both reduction approaches. For instance, the median error value is  $0.006 m$  (or  $0.005 ms^{-1}$ ) with EA-POD-NN and  $0.0252 m$  (or  $0.016 ms^{-1}$ ) with POD-NN for  $\mu_{new,1}$  and  $0.007 m$  (or  $0.007 ms^{-1}$ ) with EA-POD-NN and  $0.025 m$  (or  $0.019 ms^{-1}$ ) with POD-NN for  $\mu_{new,2}$ . Also, the maximum value of the temporal variation of errors spatial averaged  $|\overline{\varepsilon}_h|(t)$  (or  $|\overline{\varepsilon}_U|(t)$ )

is  $0.04 \text{ m}$  (or  $0.03 \text{ ms}^{-1}$ ) with EA-POD-NN and  $0.12 \text{ m}$  (or  $0.08 \text{ ms}^{-1}$ ) with POD-NN for  $\mu_{new,1}$  and  $0.04 \text{ m}$  (or  $0.03 \text{ ms}^{-1}$ ) with EA-POD-NN and  $0.10 \text{ m}$  (or  $0.08 \text{ ms}^{-1}$ ) with POD-NN for  $\mu_{new,2}$ .

Overall, both the classic POD-NN and proposed EA-POD-NN demonstrate good performance in reproducing flow depth and velocity in this moderately complex dynamic flooding scenario. The accuracy achieved in both spatial and temporal aspects of flow variables highlights the applicability of both methods for reducing the 2D SW model in this case with simultaneous inflows. Nevertheless, a slight improvement in accuracy is already noticeable with the proposed EA-POD-NN, suggesting that this approach may also prove high-performance in more complex flooding scenarios, as demonstrated below.



**Figure 7:** Simultaneous inflows case. Comparison of the model outputs at maximum flooding time for  $\mu_{new} = (11.11, 800, 380, 24.0)$  (red point in Fig. 6 (Top)), (Top) For the water depth  $h$ . (Bottom) For the velocity norm  $U$ . (Left) The reference solution: 2D SW model with triangular inflow, i.e.  $\mathbf{u}_h(\mu_{new})$ . (Middle) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for the classic POD-NN  $\mathbf{u}_{h,POD-NN}(\mu_{new})$ . (Right) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for the proposed EA-POD-NN  $\mathbf{u}_{h,EA-POD-NN}(\mu_{new})$ .

### 4.3. Case with non-simultaneous inflows

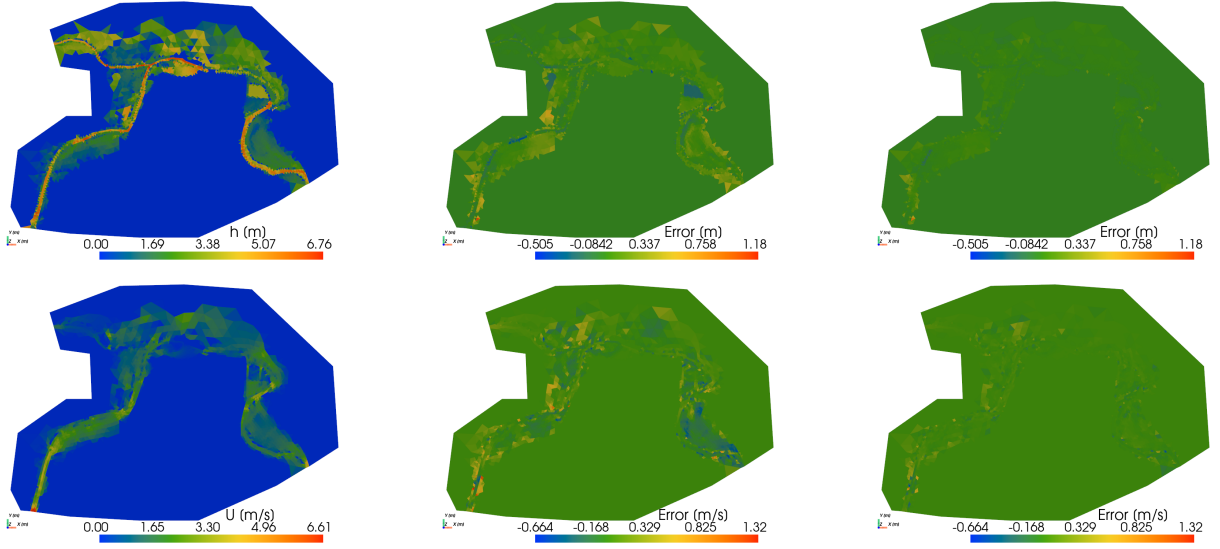
#### 4.3.1. The parameter space and training phase

This case is for non-simultaneous inflow hydrographs and the parameter reads  $\mu = (t_{peak,1}, Q_{max,1}, t_{peak,2}, Q_{max,2}; t_{sn})$  and is of dimension  $N_\mu = 5$ . Again, parameters ranges are defined around the values of the reference hydrograph and are as follows:  $t_{peak,1} \in [7.83, 14.77]$ ,  $Q_{max,1} \in [600, 1200]$ ,  $t_{peak,2} \in [7.44, 9.22]$ , and  $Q_{max,2} \in [100, 400]$ . In summary, one has  $\mathcal{H} = [7.83, 14.77] \times [600, 1200] \times [7.44, 9.22] \times [100, 400] \times [7.2, 60]$ . A visual representation of  $P_s$ , with a focus on  $t_{peak,1}$ ,  $Q_{max,1}$ ,  $t_{peak,2}$ , and  $Q_{max,2}$  is given in Fig. 11.

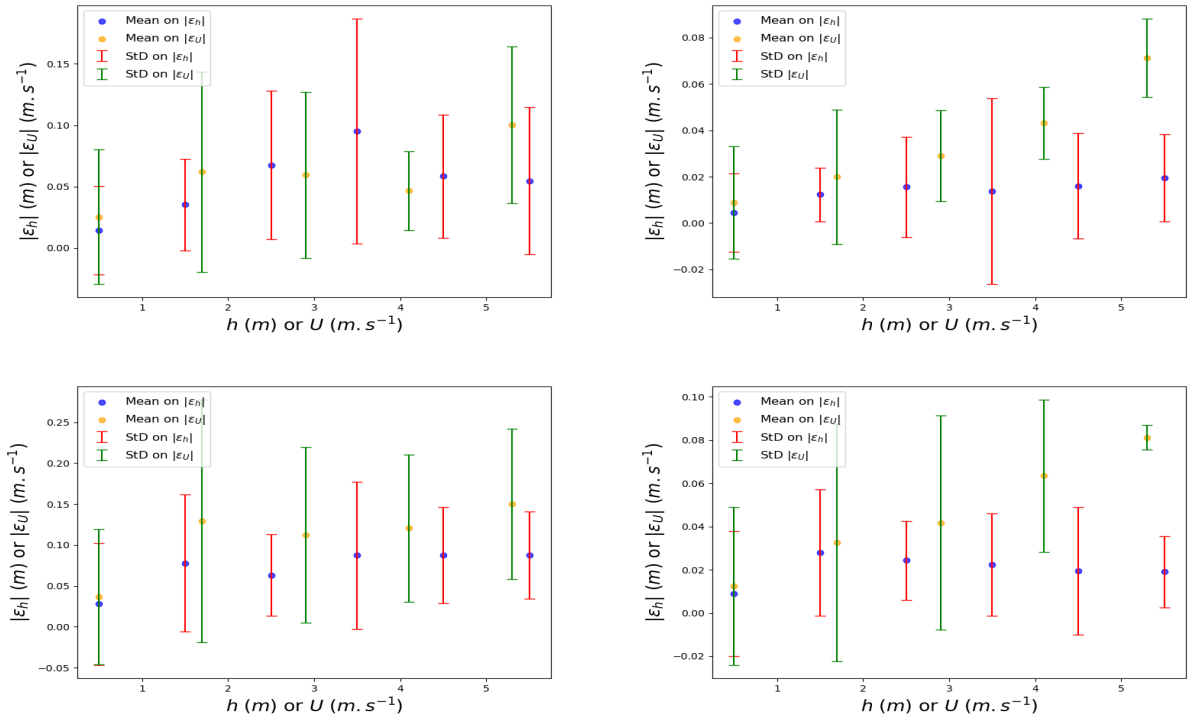
The snapshot matrix  $\mathbf{S} \in \mathbb{R}^{N_h \times N_s}$ , with a number of snapshots fixed as  $N_s = M \times N_t = 180\,225$ , with  $M = 2\,025$  simulations using DassFlow software and  $N_t = 89$  writing times, i.e.  $dt_w = 36\text{min}$  writing time steps - chosen due to



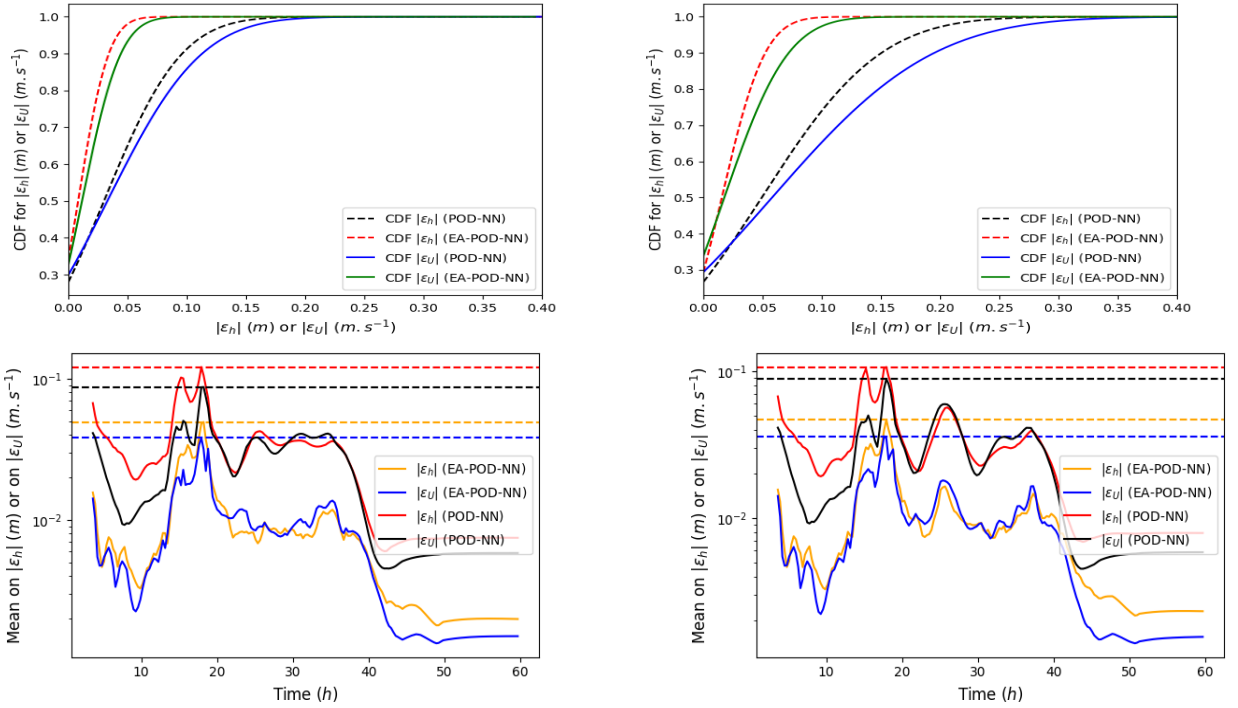
## ROM of SWEs



**Figure 8:** Simultaneous inflows case. Comparison of the model outputs at maximum flooding time for  $\mu_{new} = (12, 1000, 340, 25.2)$  (brown point in Fig. 6 (Top)), (Top) For the water depth  $h$ . (Bottom) For the velocity norm  $U$ . (Left) The reference solution: 2D SW model with triangular inflow, i.e.  $u_h(\mu_{new})$ . (Middle) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for the classic POD-NN  $u_{h,POD-NN}(\mu_{new})$ . (Right) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for the proposed EA-POD-NN  $u_{h,EA-POD-NN}(\mu_{new})$ .



**Figure 9:** Simultaneous inflows case. The histogram of the mean and the StD of spatial values of  $|\varepsilon_h|$  and  $|\varepsilon_U|$  at maximum flooding time. (Left) For the classic POD-NN method. (Right) For the proposed EA-POD-NN method. (Top) For  $\mu_{new} = (11.11, 800, 380, 24.0)$  (red point in Fig. 6. (Bottom) For  $\mu_{new} = (12, 1000, 340, 25.2)$  (brown point in Fig. 6).



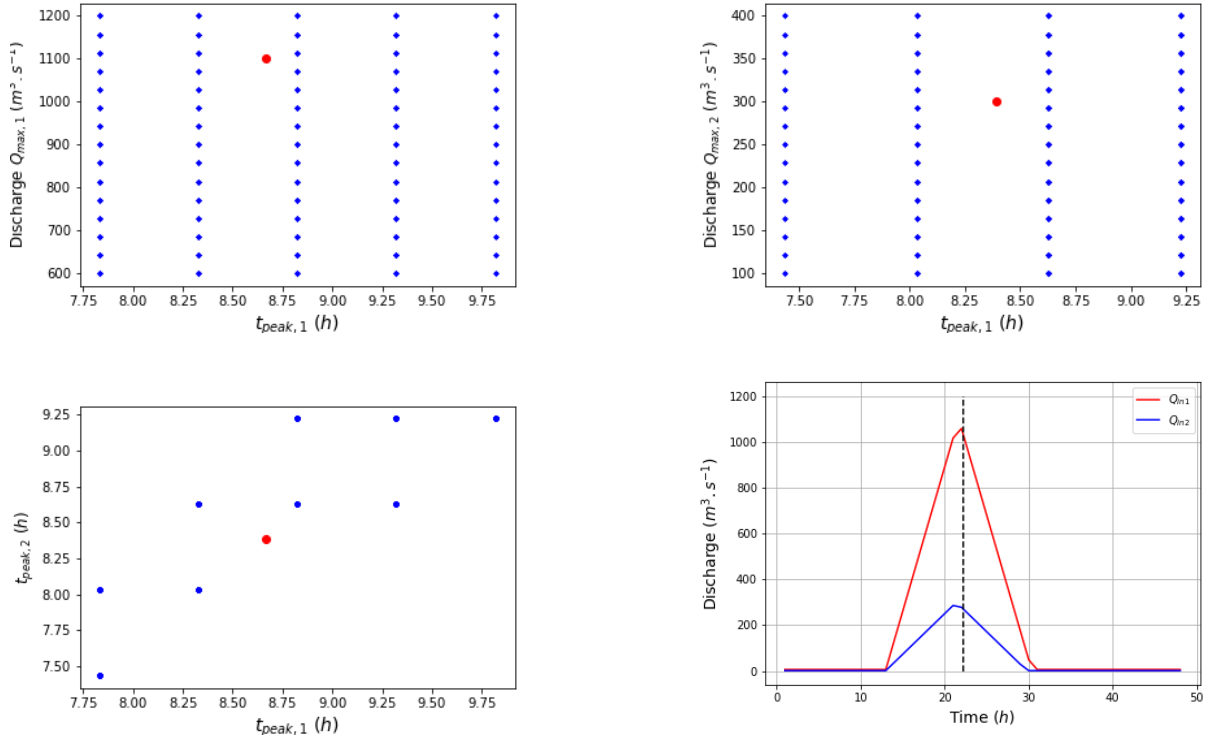
**Figure 10:** Simultaneous inflows case. Comparison between the classic POD-NN and the proposed EA-POD-NN methods performances. (Top) The CDF of the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$ . (Bottom) The mean of the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$  with respect to the real time. (Left) For  $\mu_{new} = (11.11, 800, 380, 24.0)$  (red point in Fig. 6). (Right) For  $\mu_{new} = (12, 1000, 340, 25.2)$  (brown point in Fig. 6).

numerical memory capacity constraints but this  $dt_w$  remains significantly lower than the flood propagation time that is around  $3h$  over the studied zone. Note that with  $dt_w = 18min$  as in the previous case (results not presented): (i) the POD-NN method leads to an inaccurate reduction in terms of predicted physical variables, (ii) the EA-POD-NN gives model reduction performances that are comparable to those obtained with  $dt_w = 36min$ .

Note that the training phase is performed in this case with the same tolerances and reduced dimension  $N_{rb} = 5$  as in first case (cf. section 4.1.3). The only difference is that the reduced dimension for errors reduced matrix  $\mathbf{B}_{er} \in \mathbb{R}^{N_h \times N_{er}}$  is  $N_{er} = 16$ .

#### 4.3.2. The prediction phase (online phase)

**Evaluation against reference SW model with triangular inflows.** For the prediction stage, again, we perform a prediction for a new parameter value, denoted as  $\mu_{new}$ . This new parameter value is in the parameter space range, i.e.  $\mu_{new} \in \mathcal{H}$ , but is distinct from all the calibration parameter sets that compose  $P_s$ . This new value  $\mu_{new}$  corresponds to the red point in Fig. 11 and is selected at a considerable distance from the training data points. Fig. 12 shows a



**Figure 11:** Non-simultaneous inflows case. (Top) The sampled parameter sets  $P_s$ , blue points for offline phase, red point being random parameter sets for online prediction; (left) For the first hydrograph  $Q_{in,1}$  on the Aude River and (right) for the second hydrograph  $Q_{in,2}$  on the Fresquel River (red and blue hydrographs in Fig. 4). (Bottom left) The sampled parameter sets  $P_s$  with focus on time of peak  $t_{peak,1}$  for the first hydrograph on the Aude River and  $t_{peak,2}$  for the second hydrograph on the Fresquel River. (Bottom right) Input inflow hydrographs for red point used for online prediction - vertical dashed lines correspond to the time of maximum flooding at which spatial error analysis are performed in each case.

comparison between the reference HR solution (DassFlow solution) and the predicted solution by the EA-POD-NN method.

The prediction performances with this online phase are analyzed, in terms of  $\epsilon_h(x, t)$  and  $\epsilon_U(x, t)$ , for both POD-NN and EA-POD-NN methods, with similar graphes as before: spatial fields of errors at maximum flooding are shown in Fig. 12 and the corresponding values are depicted by histograms of spatial values of  $\epsilon$  at maximum flooding time in function of physical variable values in Fig. 13. Again, reduction error is significant with the proposed EA-POD-NN method over this more complex case with non-simultaneous hydrographs. This is confirmed by the temporal variation and the Cumulative Distribution Function (CDF) of absolute error spatial average shown in Fig. 13 which clearly shows that the error with EA-POD-NN is 10 times lower than with POD-NN. Namely, the median value of error on  $h$  (or  $U$ ) is  $0.02 \text{ m}$  (or  $0.012 \text{ ms}^{-1}$ ) with POD-NN and  $1.6 \times 10^{-3} \text{ m}$  (or  $9 \times 10^{-4} \text{ ms}^{-1}$ ) with EA-POD-NN. Also, the temporal variation of errors spatial averages  $|\overline{\epsilon_h}|(t)$  (or  $|\overline{\epsilon_U}|(t)$ ) are shown in Fig. 14. In the latter, the maximum values of the temporal variation of errors spatial averaged is  $0.002 \text{ m}$  (or  $0.002 \text{ ms}^{-1}$ ) with EA-POD-NN and  $0.44 \text{ m}$  (or  $0.27 \text{ ms}^{-1}$ )

with POD-NN. The maximum value of the temporal variation of errors spatial averaged  $\overline{|\epsilon_h|}(t)$  (or  $\overline{|\epsilon_U|}(t)$ ) is  $0.02\ m$  (or  $0.02\ ms^{-1}$ ) with EA-POD-NN and  $1.11\ m$  (or  $0.79\ ms^{-1}$ ) with POD-NN.

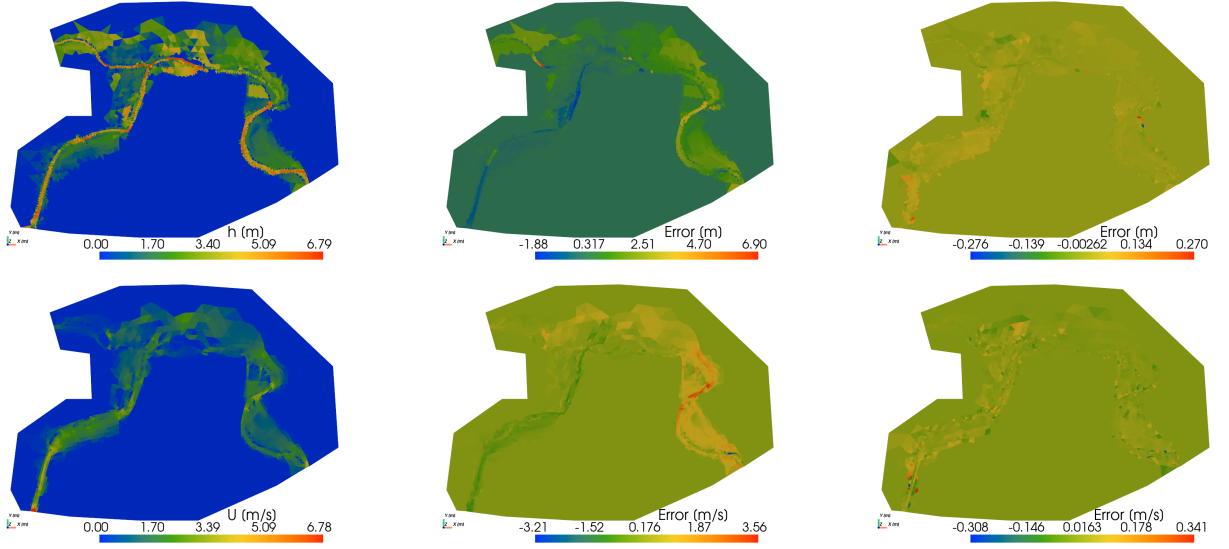
These relatively good performances show the capability of EA-POD-NN method to accurately reduce the 2D SW model in this case with non-simultaneous inflows. Results also show the limitations of the classical POD-NN to reduce of the SW model for this more complex case with non-simultaneous hydrographs.

**Evaluation against reference SW model with real inflows.** Now let us go further and test the predictability of our model first when taking the real hydrographs in Fig. 4 (right) as input inflows for the reference 2D HR hydraulic model; second when taking triangular hydrographs corresponding to a parameter  $\mu$ . The latter is given by  $\mu_{new} = (11.94, 922, 9.16, 198, 25)$  and it is out of the range of the space parameter points  $P_s$  given in Fig. 11. The predicted hydraulic variables at peak flow are comparable to those of the reference HR hydraulic model as shown in spatial maps of Fig. 15 with real like hydrographs (resp. in 18 with triangular hydrographs corresponding to the parameter  $\mu_{new} = (11.94, 922, 9.16, 198, 25)$ ). The associated error histogram is shown in Fig. 16 with real like hydrographs (resp. in 19 with triangular hydrographs). The temporal variation of spatially averaged error analysis is presented in Fig. 17 with real like hydrographs (resp. in 20 with triangular hydrographs). These results show the relatively good accuracy of the reduction compared to the reference model. For instance, in the case of real input hydrographs, the maximum value of the temporal variation of errors spatial averaged with POD-NN on  $h$  (or  $U$ ) is  $0.99\ m$  (or  $0.73\ ms^{-1}$ ) and equal to 3 times those one obtained by the EA-POD-NN ( $0.34\ m$  (or  $0.22\ ms^{-1}$ )). Moreover, with triangular hydrographs corresponding to the parameter  $\mu_{new} = (11.94, 922, 9.16, 198, 25)$ , the maximum value of the temporal variation of errors spatial averaged with POD-NN on  $h$  (or  $U$ ) is  $1.03\ m$  (or  $0.74\ ms^{-1}$ ) and equal to 5 times those one obtained by the EA-POD-NN ( $0.20\ m$  (or  $0.11\ ms^{-1}$ )). Again, the EA-POD-NN is able to reduced the SW model in this complex case with non-simultaneous inflows where the parameter  $\mu_{new} = (11.94, 922, 9.16, 198, 25)$  is taken out of the learning set ( $\mu_{new} \notin P_s$ ). While, the POD-NN is no more applicable to reduce the SW model.

## 5. Conclusion

From the obtained results, and in line with the objectives of this research, the following conclusions can be made:

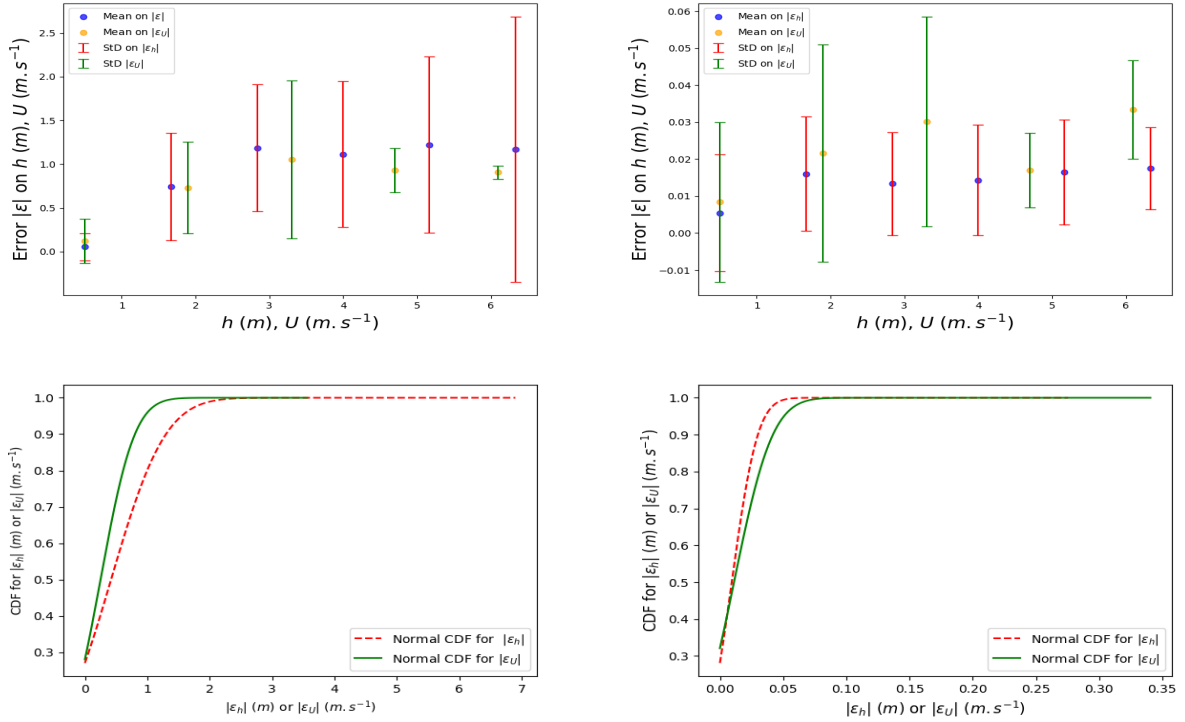
- The basic POD-NN method as proposed in [14, 25], see also Section 3, enables to reduce the 2D SW model on a moderately complex case in the sense of the tested dynamics with simultaneous inflow hydrographs, parameterized in low dimension.
- In the case of more complex non-linear waves, here produced by the two non-simultaneous inflows hydrographs, the basic POD-NN method fails to be sufficiently accurate.



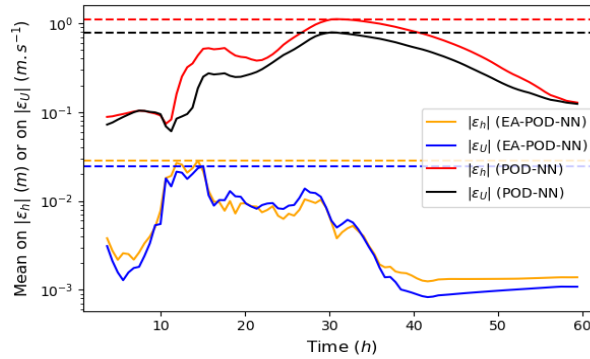
**Figure 12:** Non-Simultaneous inflows case. Comparison of the model outputs at maximum flooding time for  $\mu_{new} = (8.66, 1100, 8.38, 300, 22.2)$  (red point in Fig. 11), (Top) For the water depth  $h$ . (Bottom) For the velocity norm  $U$ . (Left) The reference solution: 2D SW model with triangular inflow, i.e.  $\mathbf{u}_h(\mu_{new})$ . (Middle) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for POD-NN  $\mathbf{u}_{h,POD-NN}(\mu_{new})$ . (Right) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for EA-POD-NN  $\mathbf{u}_{h,EA-POD-NN}(\mu_{new})$ .

- The proposed EA-POD-NN method, that additionally learns the projection error onto the RB, enables to reduce much more accurately the 2D SW model (comparisons based on the same parameter sampling). In the present application, this enables to well simulate the complex flood case with the two non simultaneous hydrographs. The spatio-temporal hydraulic variables are quite accurately predicted.
- The proposed EA-POD-NN method, is able to predict with a good accuracy ROM solution of SW model in extrapolation case.

The EA-POD-NN method paves the way for immediate application to highly non linear PDEs systems, like the 2D SWEs in complex configurations. The obtained surrogate model could then be used as a digital twin [7, 21], allowing for rapid flood forecasts given the registered scenario. Immediate research work will pertain in improving the capability of the method to be applicable in higher dimensional input parameters. In a flood modeling context, this may be more complex inflow hydrographs or spatially distributed friction fields. This should enable to generalize its applicability to the variety of real hydrological signals and rivers-floodplains connectivities, as well as to optimization or uncertainty quantification tasks. Moreover, the required snapshots number may be reduced by enforcing the model residual to vanish at some points, as in the Physics Informed Neural Networks methods for instance [20, 6].



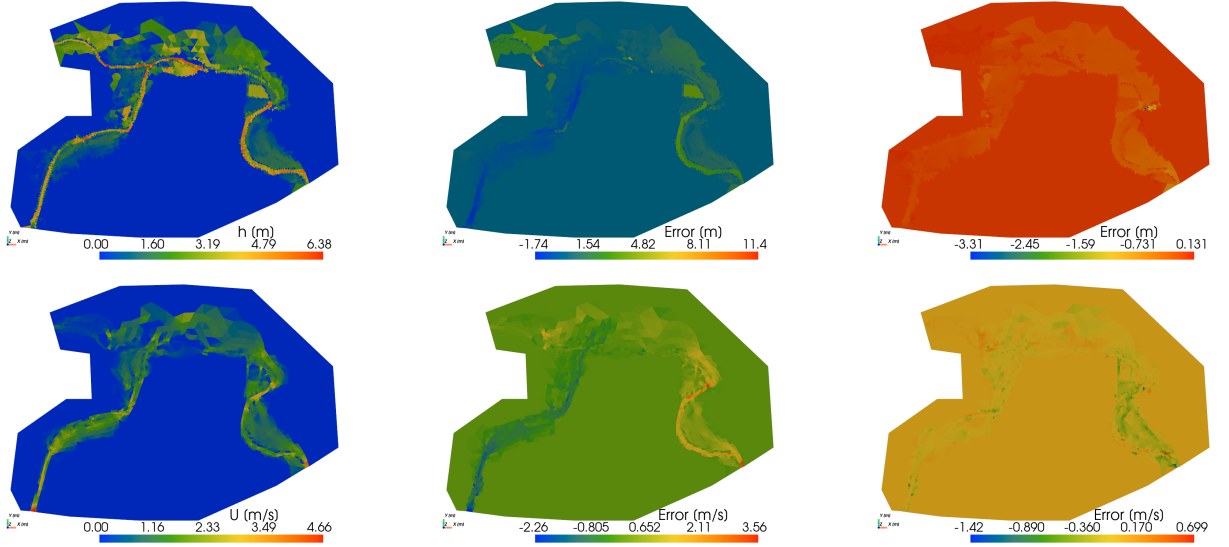
**Figure 13:** Non-simultaneous inflows case. (Top) The histogram of the mean and the StD of spatial values of the absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$  at maximum flooding time. (Left) with POD-NN method. (Right) with EA-POD-NN method for  $\mu_{new} = (8.66, 1100, 8.38, 300, 22.2)$  (red point in Fig. 11). (Bottom) The CDF of the absolute absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$ . (Left) with POD-NN method. (Right) with EA-POD-NN method for  $\mu_{new} = (8.66, 1100, 8.38, 300, 22.2)$  (red point in Fig. 11).



**Figure 14:** Non-simultaneous inflows case. The mean on the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$  with respect to the real time with POD-NN and EA-POD-NN methods for  $\mu_{new} = (8.66, 1100, 8.38, 300, 22.2)$  (red point in Fig. 6). The horizontal dashed lines correspond to the maximum value of the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$ .

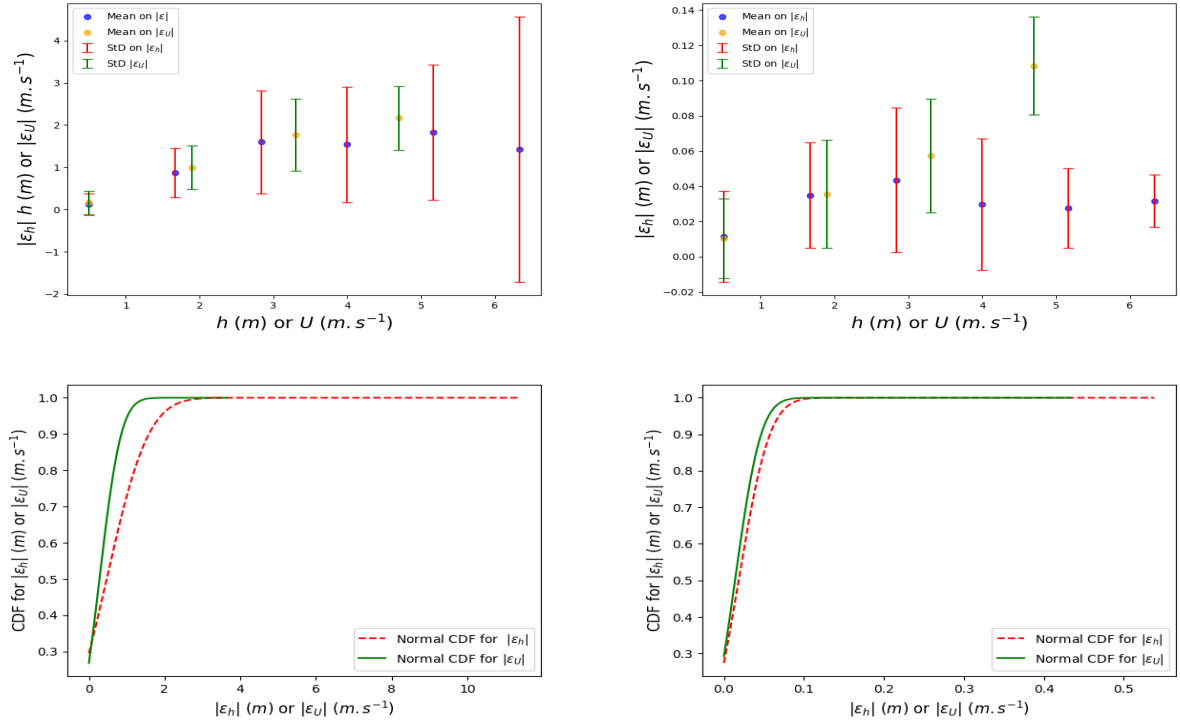
## Acknowledgments

The application of this computational sciences study has been inspired by scientific discussions led in the MUFFINS ANR project, "MULTi- scale Flood Forecasting with INnovating Solutions", ANR-21-CE04-0021-01. The authors

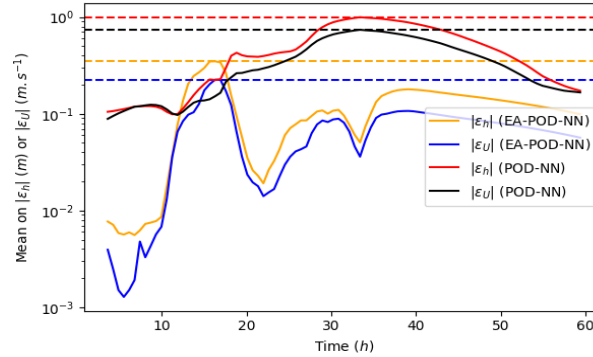


**Figure 15:** Non-simultaneous inflows case (real hydrographs). Comparison of the model outputs at maximum flooding time. (Top) For the water depth  $h$ . (Bottom) For the velocity norm  $U$ . (Left) The reference solution: 2D SW model with real inflow hydrographs (Fig. 4 right), i.e.  $\mathbf{u}_h(\mu_{new})$ . (Middle) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for POD-NN  $\mathbf{u}_{h,POD-NN}(\mu_{new})$ . (Right) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for EA-POD-NN  $\mathbf{u}_{h,EA-POD-NN}(\mu_{new})$ .

acknowledge SCHAPI-DGPR, Météo-France and IGN for providing data used in this work.



**Figure 16:** Non-simultaneous inflows case (real hydrographs). (Top) The histogram of the mean and the StD of spatial values of the absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$  at maximum flooding time with real inflow hydrographs (Fig. 4 right). (Left) with POD-NN method. (Right) with EA-POD-NN method. (Bottom) The CDF of the absolute absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$ . (Left) with POD-NN method. (Right) with EA-POD-NN method

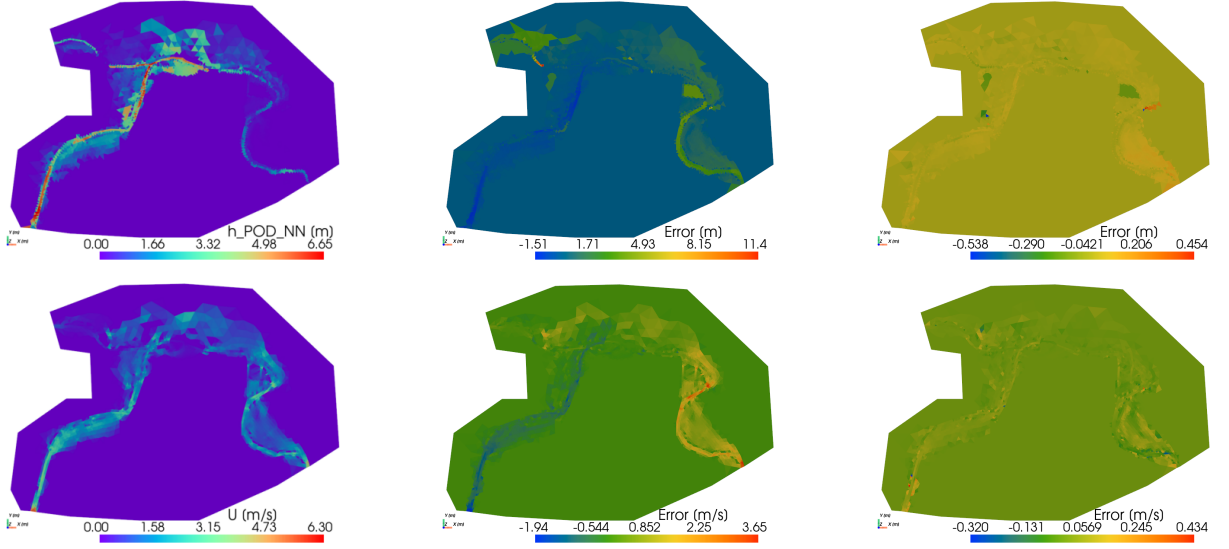


**Figure 17:** Non-simultaneous inflows case (real hydrographs). The mean on the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$  with respect to the real time with POD-NN and EA-POD-NN methods with real inflow hydrographs (Fig. 4 right). The horizontal dashed lines correspond to the maximum value of the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$ .

## References

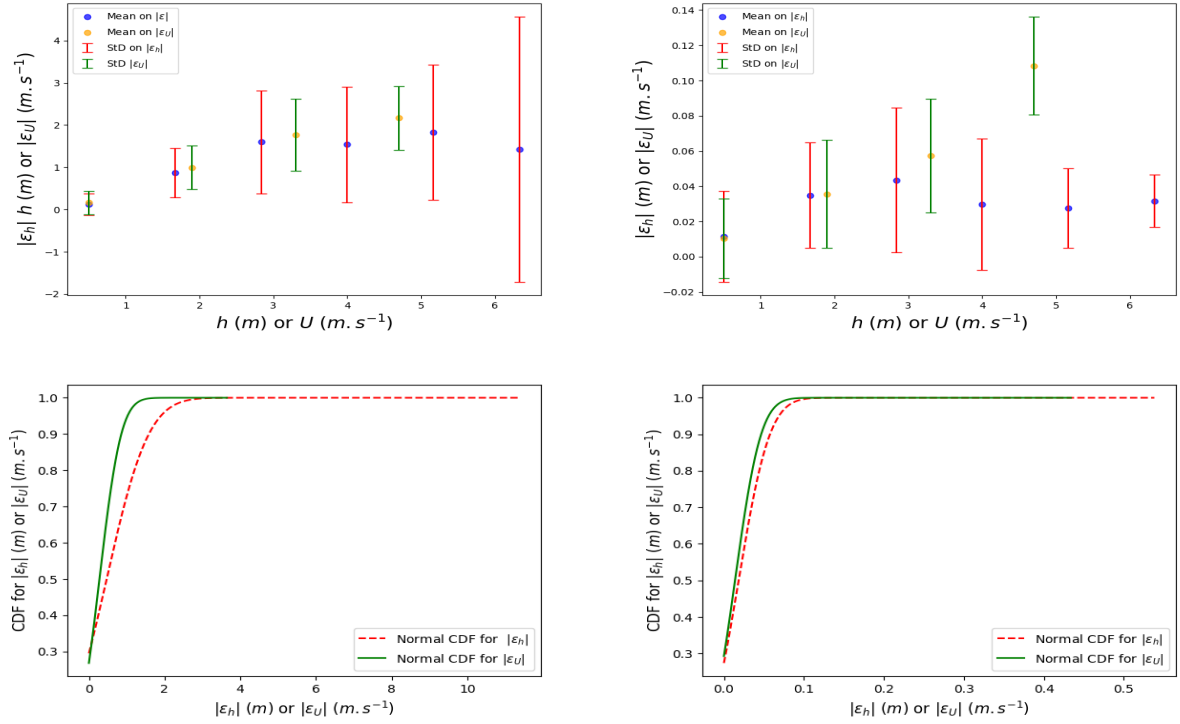
- [1] Shady E Ahmed, Omer San, Adil Rasheed, and Traian Iliescu. Nonlinear proper orthogonal decomposition for convection-dominated flows. *Physics of Fluids*, 33(12), 2021.



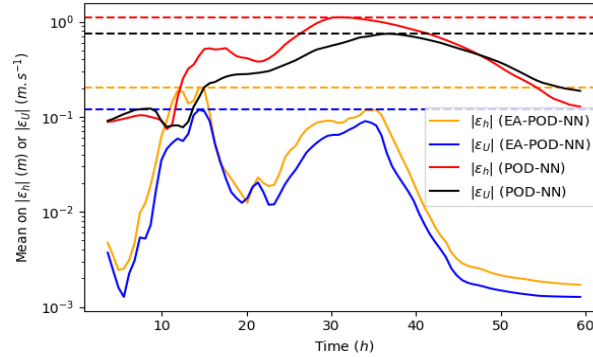


**Figure 18:** Non-simultaneous inflows case (triangle hydrographs). Comparison of the model outputs at maximum flooding time. (Top) For the water depth  $h$ . (Bottom) For the velocity norm  $U$ . (Left) The reference solution: 2D SW model with triangle inflow hydrographs for  $\mu_{new} = (11.94, 922, 9.22, 198, 25)$ , i.e.  $\mathbf{u}_h(\mu_{new})$ . (Middle) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for POD-NN  $\mathbf{u}_{h,POD-NN}(\mu_{new})$ . (Right) The difference  $\varepsilon_h$  or  $\varepsilon_U$  for EA-POD-NN  $\mathbf{u}_{h,EA-POD-NN}(\mu_{new})$ .

- [2] Joshua Barnett, Charbel Farhat, and Yvon Maday. Neural-network-augmented projection-based model order reduction for mitigating the kolmogorov barrier to reducibility. *Journal of Computational Physics*, 492:112420, 2023.
- [3] P. Benner, W. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, and L. Miguel Silveira. *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*. Model Order Reduction ; Volume 2. De Gruyter., Berlin :, 2020-2021.
- [4] Joao G Caldas Steinstraesser, Vincent Guinot, and Antoine Rousseau. Modified parareal method for solving the two-dimensional nonlinear shallow water equations using finite volumes. *The SMAI journal of computational mathematics*, 7:159–184, 2021.
- [5] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6(4):911–917, 1995.
- [6] Wenqian Chen, Qian Wang, Jan S Hesthaven, and Chuhua Zhang. Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of computational physics*, 446:110666, 2021.
- [7] Francisco Chinesta, Elias Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, and Fouad El Khaldi. Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of computational methods in engineering*, 27:105–134, 2020.
- [8] Albert Cohen and Ronald DeVore. Approximation of high-dimensional parametric pdes. *Acta Numerica*, 24:1–159, 2015.
- [9] Sourav Dutta, Matthew W Farthing, Emma Perracchione, Gaurav Savant, and Mario Putti. A greedy non-intrusive reduced order model for shallow water equations. *Journal of Computational Physics*, 439:110378, 2021.
- [10] Nabil El Mocayd, Sophie Ricci, Nicole Goutal, Mélanie C Rochoux, Sébastien Boyaval, Cédric Goeury, Didier Lucor, and Olivier Thual. Polynomial surrogates for open-channel flows in random steady state. *Environmental Modeling & Assessment*, 23:309–331, 2018.
- [11] Pierre-André Garambois, Jérôme Monnier, and Villenave Lilian. Coupled 2D hydrologic-hydraulic catchment scale flood modeling with data assimilation capabilities: the DassHydro platform. In *Colloque SHF - "Prévision des crues et inondations, avancées, valorisations et perspectives"*, Toulouse,, France, 2023.



**Figure 19:** Non-simultaneous case (triangle hydrographs). (Top) The histogram of the mean and the StD of spatial values of the absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$  at maximum flooding time with triangle inflow hydrographs for  $\mu_{new} = (11.94, 922, 9.22, 198, 25)$ . (Left) with POD-NN method. (Right) with EA-POD-NN method. (Bottom) The CDF of the absolute absolute difference  $|\epsilon_h|$  and  $|\epsilon_U|$ . (Left) with POD-NN method. (Right) with EA-POD-NN method



**Figure 20:** Non-simultaneous inflows case (triangle hydrographs). The mean on the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$  with respect to the real time with POD-NN and EA-POD-NN methods with triangle inflow hydrographs for  $\mu_{new} = (11.94, 922, 9.22, 198, 25)$ . The horizontal dashed lines correspond to the maximum value of the absolute difference  $|\epsilon_h|$  or  $|\epsilon_U|$ .

- [12] Hojat Ghorbanidehno, Jonghyun Lee, Matthew Farthing, Tyler Hesser, Eric F Darve, and Peter K Kitanidis. Deep learning technique for fast inference of large-scale riverine bathymetry. *Advances in Water Resources*, 147:103715, 2021.
- [13] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

- [14] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [15] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [17] John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.
- [18] Jerome Monnier, Frédéric Couderc, Denis Dartus, Kévin Larnier, Ronan Madec, and J-P Vila. Inverse algorithms for 2D shallow water equations in presence of wet dry fronts: Application to flood plain dynamics. *Advances in Water Resources*, 97:11–24, 2016.
- [19] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. 2015.
- [20] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [21] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *Ieee Access*, 8:21980–22012, 2020.
- [22] Open source computational software DassFlow (Data Assimilation for Free Surface Flows). Math. institute of toulouse (imt) and inrae and insa and icube strasbourg. <https://github.com/dasshydro>, 2023.
- [23] Răzvan Ștefănescu, Adrian Sandu, and Ionel M Navon. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 76(8):497–521, 2014.
- [24] Maria Strazzullo, Francesco Ballarin, and Gianluigi Rozza. POD-Galerkin model order reduction for parametrized nonlinear time-dependent optimal flow control: an application to shallow water equations. *Journal of Numerical Mathematics*, 30(1):63–84, 2022.
- [25] Qian Wang, Jan S Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of computational physics*, 384:289–307, 2019.
- [26] Dunhui Xiao. Error estimation of the parametric non-intrusive reduced order model using machine learning. *Computer Methods in Applied Mechanics and Engineering*, 355:513–534, 2019.