Combination of intrusive POD-based reduced-order models and augmented Riemann solvers applied to unsteady 2D shallow water equations

P. Solán-Fustero [a],*, J.L. Gracia [b], A. Navas-Montilla [a], P. García-Navarro [a]

[a] *I3A and Fluid Mechanics Department, University of Zaragoza, Spain*
[b] *IUMA and Applied Mathematics Department, University of Zaragoza, Spain*

ARTICLE INFO

ABSTRACT

The shallow water equations (SWEs) can be used to model the spatio-temporal evolution of free surface flows. The numerical resolution of realistic problems based on the 2D SWEs by means of augmented Roe-based (ARoe) methods requires the inclusion of certain numerical corrections to avoid non-physical results in presence of irregular topography and wet dry fronts. Besides that, their complex and transient nature involves high computational costs. In this direction, intrusive reduced-order models (ROMs) based on the proper orthogonal decomposition (POD) are presented as alternative to speed up computational calculations without compromising the accuracy of the solutions. The main objective of this article is to study whether the inclusion of numerical corrections in the ROM strategy of the 2D SWEs for non trivial situations is necessary to obtain accurate solutions or not, and, if necessary, to present their reduced version. In addition to this, it is proposed to solve problems with Dirichlet-type boundary conditions (BCs) by means of ROMs using a technique whereby the BCs are directly integrated into the on-line phase of ROM solving. The efficiency of the ARoe-based ROM has been tested with respect to the full-order model by comparing their computational cost and the accuracy of their solutions in different numerical cases.

1. Introduction

Shallow water models are mathematically defined as those in which the horizontal scale is much larger than the depth of the layer of fluid [1]. They can be modelled by means of the shallow water equations (SWE) in which the dimensionality of the problem is reduced with respect to the Navier–Stokes equations by averaging the variables in the vertical direction and assuming hydrostatic pressure. They are formulated as a system of hyperbolic partial differential equations [2] of mass and momentum and they govern the spatio-temporal evolution of the water depth and depth averaged velocity of the fluid.

There is no analytical solution for the SWE, so it is necessary to solve them numerically. The Finite Volume (FV) method, independently introduced by McDonald in 1971 [3] and MacCormack and Paullay in 1972 [4] and extended by Rizzi and Inouye in 1973 [5], is based on the direct discretization of the integral form of the conservation laws and this form does not require the fluxes to be continuous. The FV method being closer to the physical flow conservation laws is the reason why it is very useful when solving Fluid Mechanics equations and, in particular, the SWEs.

---

* Corresponding author.
  E-mail address: psolfus@unizar.es (P. Solán-Fustero).

Godunov-type FV methods discretize the information of the hyperbolic conservation systems by assuming a piecewise constant distribution of the conserved variables within computational cells [1,6,7]. The variables are updated in the computational cells based on the calculation of numerical fluxes at the walls of each cell. In order to calculate these fluxes, so-called Riemann problems (RP) consist of the hyperbolic equation together with special initial data defined by a piecewise constant function with a single jump discontinuity at some point [1]. Approximate Riemann solvers, such as the Roe solver, are used to solve these RPs [1,8].

The source terms that model the effects of variable bed level and friction need to be included in the SWE to capture the dynamics of realistic shallow flows. The use of the numerical flux as defined for homogeneous equations is not adequate to solve situations involving source terms, as indicated by several works [9–11]. In this way, the augmented Roe method [10] has been chosen since it represents the complete eigenstructure of the Jacobian matrix of the flux, i.e., all waves, and it has been reported to produce robust and stable solutions in many situations. It is designed to preserve the well-balanced property, i.e., the equilibrium in presence of source terms with machine precision [12,13]. In addition to that, it includes some numerical corrections to fix unphysical numerical solutions that may appear under certain circumstances, such as the entropy and the wet-dry front problems [1,8,14,15].

The numerical resolution of the SWE using augmented Roe methods may involve large computational resources. It is therefore worth exploring alternatives to speed up the calculations. There has been a huge development in recent years of a wide range of mathematical strategies and tools in the scientific literature to facilitate, improve and increase the calculation capacity of the classical methods used in the framework of Fluid Mechanics. The main examples of these are the (discrete) empirical interpolation method [16,17], dynamic mode decomposition [18–21], Krylov subspaces method [22,23] and artificial neural networks [18,24,25]. Among all of them, intrusive reduced-order models (ROMs) based on the proper orthogonal decomposition (POD, [26]) are particularly suitable for application to the SWE, as they speed up computational times with respect to classical schemes while maintaining the accuracy of the solutions. In addition, they allow a detailed and well understood development to emulate the augmented solvers due to their intrusive character.

The intrusive POD-based ROM strategy consists of two phases: the off-line phase, in which the ROM is trained; and the on-line phase, in which the ROM is numerically solved. The training of the ROM starts by computing a set of high-fidelity reference solutions with a classic numerical scheme, also called full-order models (FOMs), following the snapshot method [27]. The solutions calculated by the FOM are called training snapshots and they are used to train the ROM by means of the singular value decomposition (SVD) [27].

In short, ROMs act as faster alternative numerical schemes to replace the calculations performed by FOMs. They reside in a reduced dimensional space much smaller than the physical space, which is the reason why they are more efficient than FOMs. This is done by means of the Galerkin method [28], according to which the numerical solution is projected orthogonally from the physical space to the reduced space defined by the POD method. Thanks to the proper interval decomposition (PID) [29–31], nonlinear problems such as the SWEs can be successfully and efficiently approximated by linearized ROMs, as proposed in this paper.

Because of their intrusive condition, ROMs should be able to preserve the good features since they arise as transformations of already known numerical schemes. The main objective of this paper is to study the necessity of including the numerical corrections that are necessary to the ROM to obtain accurate solutions. This study can be framed within the concept of ROM consistency as proposed by Ingimarson, Rebholz and Iliescu in a very recent work [32,33], in which they investigate theoretically and numerically how the performance of the ROM is affected by the discretization model used with respect to the FOM.

Dirichlet-type boundary conditions (BCs), depending on the time or not, require special treatment when integrated into ROM solving. There are different ways to deal with it in the literature, such as the *lifting functions* used to homogenize the snapshots so that they become independent of the BCs [34] or the explicit-boundary POD method proposed in [35,36], according to which part of the ROM is replaced by equations for the original BCs. In this paper, it is proposed to solve problems with Dirichlet-type BCs by means of ROMs using a technique whereby the BCs are directly integrated into the on-line phase of ROM solving.

The study proposed in this paper is illustrated by means of several 2D numerical results that provide examples of the inclusion of the aforementioned numerical corrections and the resolution of Dirichlet-type BCs with ROMs.

The remainder of the paper is organized as follows. Section 2 introduces the 2D SWE, their discretization using the FV method and the numerical corrections. Section 3 outlines the POD-based ROM applied to the SWE and the reduced version of the numerical corrections. Section 4 presents several numerical test cases to prove the inclusion of the numerical corrections in the ROM; Section 5 presents three problems to test the efficiency of the ROM in realistic scenarios. Finally, concluding remarks are drawn in Section 6.

2. Governing equations and full-order model

The 2D version of the SWE with source terms reads

$$\partial_t h + \partial_x q_x + \partial_y q_y = 0,$$
$$\partial_t q_x + \partial_x \left(q_x^2/h + gh^2/2 \right) + \partial_y \left(q_x q_y/h \right) = gh \left(S_{z_x} - S_{f_x} \right), \tag{1}$$
$$\partial_t q_y + \partial_x \left(q_x q_y/h \right) + \partial_y \left(q_y^2/h + gh^2/2 \right) = gh \left(S_{z_y} - S_{f_y} \right),$$

where $h = h(x, y, t)$ is the depth and $q_x = q_x(x, y, t)$ and $q_y = q_y(x, y, t)$ are the discharges in the x- and y-directions, with $q_x = hu_x$ and $q_y = hu_y$, and $u_x = u_x(x, y, t)$ and $u_y = u_y(x, y, t)$, the depth averaged velocities in the x- and y-directions; g is the gravity acceleration; S_{z_x} and S_{z_y} are the source terms due to the gradient of the bed elevation $z = z(x, y)$

$$S_{z_x} = -\partial_x z, \quad S_{z_y} = -\partial_y z,$$

and S_{f_x} and S_{f_y} are the source terms due to the channel friction

$$S_{f_x} = \frac{n_b^2 u_x \sqrt{u_x^2 + u_y^2}}{h^{4/3}}, \quad S_{f_y} = \frac{n_b^2 u_y \sqrt{u_x^2 + u_y^2}}{h^{4/3}},$$

where n_b is the Manning coefficient. The problem is posed in the domain $(x, y, t) \in [0, L_x] \times [0, L_y] \times [0, T]$ and the initial condition (IC) and the boundary conditions (BCs) for h, q_x and q_y will be defined in each numerical case in Section 4.

All the physical variables that appear in this paper are in units of the International System. They have been omitted in all cases for the sake of simplicity.

2.1. Finite volume method

The spatial domain $[0, L_x] \times [0, L_y]$ is discretized using the FV method by means of I_c cells of volume S_i whose centre positions are $\mathbf{x}_i = (x_i, y_i)$, with $i = 1, \ldots, I_c$. In general, the volume cells can be defined as triangles or rectangles with $I_f = 3$ or 4 polygonal faces, respectively, of length $l_{i,e}$ and with outward-pointing normal vector $\mathbf{n}_{i,e} = (n_x, n_y)_{i,e}$. The index e denotes the eth edge of the cell ith cell, with $e = 1, \ldots, I_f$.

The boundary of the domain Γ is split into

$$\Gamma = \Gamma^D \cup \Gamma^N, \; \Gamma^D \cap \Gamma^N = \emptyset,$$

where Dirichlet BCs are specified on Γ^D and Neumann BCs are specified on Γ^N, which include free and wall type conditions, i.e. transmissive and non-transmissive. The vertices are numbered with index sets J^I, J^D and J^N such that $i \in J^I$ if $(x_i, y_i) \in (0, L_x) \times (0, L_y)$, $i \in J^D$ if $(x_i, y_i) \in \Gamma^D$ and $i \in J^N$ if $(x_i, y_i) \in \Gamma^N$; and $J = J^I \cup J^D \cup J^N$.

The conserved variables are

$$\mathbf{U}_i^n = \left(h_i^n, (q_x)_i^n, (q_y)_i^n\right)^T,$$

where $h_i^n \approx h(x_i, y_i, t^n)$, $(q_x)_i^n \approx q_x(x_i, y_i, t^n)$ and $(q_y)_i^n \approx q_y(x_i, y_i, t^n)$ are the cell-averaged values of the water depth and the discharges in the x- and y-directions, respectively.

The FOM is obtained by discretizing (1) with the Godunov-type scheme based on the Roe method as a sum of waves [37]

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} \sum_{m=1}^{3} \left(\tilde{\lambda}_m^- \tilde{\gamma}_m^- \mathbf{e}_m\right)_{i,e}^n l_{i,e}, \; i = 1, \ldots, I_c, \tag{2}$$

where $\left(\tilde{\lambda}_m^{\pm}\right)_{i,e}^n = \left(\tilde{\lambda}_m \pm |\tilde{\lambda}_m|\right)_{i,e}^n / 2$, with $m = 1, 2, 3$. The numerical eigenvalues and eigenvectors of the Jacobian matrix of a transformation of (2) [9] are

$$\left(\tilde{\lambda}_1\right)_{i,e}^n = (\tilde{\mathbf{u}} \cdot \mathbf{n} - \tilde{c})_{i,e}^n, \quad \left(\tilde{\lambda}_2\right)_{i,e}^n = (\tilde{\mathbf{u}} \cdot \mathbf{n})_{i,e}^n, \quad \left(\tilde{\lambda}_3\right)_{i,e}^n = (\tilde{\mathbf{u}} \cdot \mathbf{n} + \tilde{c})_{i,e}^n,$$

$$\left(\tilde{\mathbf{e}}_1\right)_{i,e}^n = \begin{pmatrix} 1 \\ \tilde{u}_x - \tilde{c} n_x \\ \tilde{u}_y - \tilde{c} n_y \end{pmatrix}_{i,e}^n, \quad \left(\tilde{\mathbf{e}}_2\right)_{i,e}^n = \begin{pmatrix} 0 \\ -\tilde{c} n_y \\ \tilde{c} n_x \end{pmatrix}_{i,e}^n, \quad \left(\tilde{\mathbf{e}}_3\right)_{i,e}^n = \begin{pmatrix} 1 \\ \tilde{u}_x + \tilde{c} n_x \\ \tilde{u}_y + \tilde{c} n_y \end{pmatrix}_{i,e}^n,$$

with the following numerical velocities $\tilde{\mathbf{u}}_{i,e}^n = (\tilde{u}_x, \tilde{u}_y)_{i,e}^n$

$$\left(\tilde{u}_x\right)_{i,e}^n = \frac{(u_x)_i^n \sqrt{h_i^n} + (u_x)_j^n \sqrt{h_j^n}}{\sqrt{h_i^n} + \sqrt{h_j^n}}, \quad \left(\tilde{u}_y\right)_{i,e}^n = \frac{(u_y)_i^n \sqrt{h_i^n} + (u_y)_j^n \sqrt{h_j^n}}{\sqrt{h_i^n} + \sqrt{h_j^n}}, \quad \tilde{c}_{i,e}^n = \sqrt{g \tilde{h}_{i,e}^n},$$

where the ith and the jth are the adjacent cells separated by the eth edge; $\tilde{h}_{i,e}^n = \left(h_i^n + h_j^n\right)/2$; and $\left(\tilde{\gamma}_m^-\right)_{i,e}^n = \left(\tilde{\alpha}_m - \tilde{\beta}_m^- / \tilde{\lambda}_m^-\right)_{i,e}^n$. The wave strengths are defined as

$$\left(\tilde{\alpha}_{1,3}\right)_{i,e}^n = \frac{\delta h_{i,e}^n}{2} \mp \frac{1}{2\tilde{c}_{i,e}^n} (\delta \mathbf{q} - \tilde{\mathbf{u}} \delta h)_{i,e}^n \mathbf{n}_{i,e}, \quad \left(\tilde{\alpha}_2\right)_{i,e}^n = \frac{1}{\tilde{c}_{i,e}^n} \left[(\delta q_y - \tilde{u}_y \delta h) n_x - (\delta q_x - \tilde{u}_x \delta h) n_y\right]_{i,e}^n,$$

and the source terms $\left(\tilde{\beta}_m^-\right)_{i,e}^n = \left(\tilde{\beta}_m \tilde{\lambda}_m^- / \tilde{\lambda}_m\right)_{i,e}^n$, with

$$\left(\tilde{\beta}_1\right)_{i,e}^n = -\left(\tilde{\beta}_3\right)_{i,e}^n = \left(\tilde{\beta}_f + \tilde{\beta}_z\right)_{i,e}^n, \quad \left(\tilde{\beta}_2\right)_{i,e}^n = 0.$$

The discrete friction and bed source terms are

$$\left(\tilde{\beta}_f\right)_{i,e}^n = \frac{1}{2} \frac{g \tilde{h}_{i,e}^n}{\tilde{c}_{i,e}^n} (d_n)_{i,e} \frac{(\tilde{n}_b)_{i,e}^2 \tilde{\mathbf{u}}_{i,e}^n \cdot \mathbf{n}_{i,e} |\tilde{\mathbf{u}}^n|_{i,e}}{\max\left(h_i^n, h_j^n\right)^{4/3}}, \quad \left(\tilde{\beta}_z\right)_{i,e}^n = \frac{1}{2} \frac{g \tilde{h}_{i,e}^n}{\tilde{c}_{i,e}^n} (\delta z)_{i,e}, \tag{3}$$

where $(d_n)_{i,e} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the distance between the centres of the ith and the jth cells, $(\tilde{n}_b)_{i,e} = \left[(n_b)_i + (n_b)_j\right]/2$, and

$$\delta h_{i,e}^n = h_j^n - h_i^n, \quad \delta z_{i,e} = z_j - z_i, \quad \delta \left(q_x\right)_{i,e}^n = (q_x)_j^n - (q_x)_i^n, \quad \delta \left(q_y\right)_{i,e}^n = (q_y)_j^n - (q_y)_i^n.$$

The time step $\Delta t = t^{n+1} - t^n$ of the explicit schemes is selected dynamically using the Courant–Friedrichs–Lewy (CFL) condition [38]

$$\Delta t = CFL \frac{\min\left\{S_i\right\}}{\max\left\{\left(\tilde{\lambda}_1\right)_{i,e}^n, \left(\tilde{\lambda}_2\right)_{i,e}^n, \left(\tilde{\lambda}_3\right)_{i,e}^n\right\}}, \tag{4}$$

where the CFL number satisfies $0 < CFL \leq 0.5$.

The ARoe-based FOM is shown in Appendix A.

2.2. Numerical corrections

In the augmented Roe, it is necessary to take into account a series of numerical corrections that are applied to the scheme to ensure its robustness and avoid the appearance of results of unphysical nature [9,14,15,39–42].

The friction correction to avoid reverse flow, the entropy fix and the wet/dry treatment are based on a thorough understanding of Riemann problems and thus arise as results of the superposition of travelling waves whose velocities are the eigenvalues of the problem. They are explained in detail below.

2.2.1. Friction correction to avoid reverse flow

The numerical friction force may reverse the sign of the flow velocity, which is not physically admissible, so it has to be restricted. With this purpose, define the water discharge intermediate

$$\begin{aligned}
\left(q^*\right)_{i,e}^n &= \mathbf{q}_i^n \cdot \mathbf{n}_{i,e} + \left(\tilde{\alpha}_1 \tilde{\lambda}_1\right)_{i,e}^n - \left(\tilde{\beta}_z\right)_{i,e}^n - \left(\tilde{\beta}_f\right)_{i,e}^n, \\
\left(q^{**}\right)_{i,e}^n &= \mathbf{q}_i^n \cdot \mathbf{n}_{i,e} + \left(\tilde{\alpha}_1 \tilde{\lambda}_1\right)_{i,e}^n - \left(\tilde{\beta}_z\right)_{i,e}^n.
\end{aligned} \tag{5}$$

If $\left(q^{**}\right)_{i,e}^n < 0$ and $\left(q^*\right)_{i,e}^n > 0$, then $\left(q^*\right)_{i,e}^n \left(q^{**}\right)_{i,e}^n < 0$, which means that the numerical friction might reverse the flow. In this case, the friction contribution of the source term is redefined as

$$\left(\tilde{\beta}_f\right)_{i,e}^n = \left(q^{**}\right)_{i,e}^n. \tag{6}$$

For example, suppose that at the first wall ($e = 1$) of the ith cell the condition indicated in (6) is fulfilled, then the friction term $\left(\tilde{\beta}_f\right)_{i,1}^n$ from (A.1) is replaced by the intermediate state $\left(q^{**}\right)_{i,1}^n$ defined in (5). In this way, the water depth is updated as follows

$$\begin{aligned}
h_i^{n+1} &= h_i^n + \cdots + \frac{\Delta t}{S_i}\left[\left(\tilde{\beta}_f\right)_{i,1}^n \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^n l_{i,1} + \sum_{e=2}^{I_f}\left(\tilde{\beta}_f\right)_{i,e}^n \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,e}^n l_{i,e}\right] \\
&= h_i^n + \cdots + \frac{\Delta t}{S_i}\left[\left(q^{**}\right)_{i,1}^n \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^n l_{i,1} + \sum_{e=2}^{I_f}\left(\tilde{\beta}_f\right)_{i,e}^n \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,e}^n l_{i,e}\right].
\end{aligned} \tag{7}$$

The other two Eqs. (A.2) and (A.3) associated to with the discharges q_x and q_y are modified correspondingly. For more details, see [9,15].

2.2.2. Entropy fix

Augmented Riemann solvers, such as the ARoe solver used here, may lead to physically meaningless solutions due to the entropy problem [8,43]. The regime of the flow can be sub- or supercritical depending on the relative sign of the eigenvalues, $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$, such that, if $\tilde{\lambda}_1 \tilde{\lambda}_2 < 0$, it is subcritical and, if $\tilde{\lambda}_1 \tilde{\lambda}_2 > 0$, supercritical. It can also be possible that one of the eigenvalues might be zero for some values of (h, q). In transcritical rarefactions the continuous fan of intermediate states is represented using a Roe eigenvalue with approximately zero velocity [44] and this would imply numerical problems to properly update the variables.

To solve this problem, a numerical correction, known as entropy fix, have to be included in the numerical scheme [37,44]. The entropy fix is implemented under the following conditions: if $\left(\lambda_m\right)_i^n < 0 < \left(\lambda_m\right)_j^n$, then the eigenvalues $\left(\tilde{\lambda}_m^-\right)_{i,e}^n$ and $\left(\tilde{\lambda}_m^+\right)_{i,e}^n$ have to be replaced by the following left and right states, respectively

$$\left(\tilde{\lambda}_m^\leftarrow\right)_{i,e}^n = \left(\lambda_m\right)_i^n \frac{\left(\lambda_m\right)_j^n - \left(\tilde{\lambda}_m\right)_{i,e}^n}{\left(\lambda_m\right)_j^n - \left(\lambda_m\right)_i^n}, \quad \left(\tilde{\lambda}_m^\rightarrow\right)_{i,e}^n = \left(\lambda_m\right)_j^n \frac{\left(\tilde{\lambda}_m\right)_{i,e}^n - \left(\lambda_m\right)_i^n}{\left(\lambda_m\right)_j^n - \left(\lambda_m\right)_i^n}, \tag{8}$$

with $m = 1, 2, 3$. The eigenvalues are evaluated in the cells as follows

$$\left(\lambda_1\right)_i^n = \left(\sqrt{u_x^2 + u_y^2} - c\right)_i^n, \quad \left(\lambda_2\right)_i^n = \left(\sqrt{u_x^2 + u_y^2}\right)_i^n, \quad \left(\lambda_3\right)_i^n = \left(\sqrt{u_x^2 + u_y^2} + c\right)_i^n,$$

with $\left(u_x\right)_i^n = \left(q_x/h\right)_i^n$, $\left(u_y\right)_i^n = \left(q_y/h\right)_i^n$ and $(c)_i^n = \sqrt{gh_i^n}$. Some of the source terms have to be cancelled as follows

$$\left(\tilde{\beta}_1^\rightarrow\right)_{i,e}^n = 0, \quad \left(\tilde{\beta}_3^\leftarrow\right)_{i,e}^n = 0. \tag{9}$$

2.2.3. Wet/dry treatment

Realistic scenarios may involve cases in which the water elevation in the ith cell may be smaller than the bed elevation of the neighbouring jth cell, i.e., $h_i + z_i < h_j$. This needs a special treatment to avoid unphysical solutions that may lead to the wetting of areas that must not be wet. This treatment consists of two steps embedded in the resolution of the FOM.

First, it is necessary to bounce the information that updates the variables back to the wet cell and not to the dry cell. For this purpose, the following intermediate states are defined for the water depth [9,15]

$$(h^*)^n_{i,e} = h^n_i + (\tilde{\alpha}_1)^n_{i,e} - \left(\frac{\tilde{\beta}_1}{\tilde{\lambda}_1}\right)^n_{i,e}, \qquad (h^{**})^n_{i,e} = h^n_j - (\tilde{\alpha}_3)^n_{i,e} + \left(\frac{\tilde{\beta}_3}{\tilde{\lambda}_3}\right)^n_{i,e}. \tag{10}$$

The general wet/dry treatment is written as follows:

- If $h^n_j = 0$ and $(h^{**})^n_{i,e} < 0$, then the contribution of neighbouring jth cell is cancelled in the equation associated to the ith cell in (2)

$$\sum_{m=1}^{3} \left(\tilde{\lambda}_m^- \tilde{\gamma}^- \mathbf{e}_m\right)^n_{j,e} = 0.$$

- If $h^n_i = 0$ and $(h^*)^n_{i,e} < 0$, then the contribution of neighbouring ith cell is cancelled in the equation associated to the jth cell in (2)

$$\sum_{m=1}^{3} \left(\tilde{\lambda}_m^- \tilde{\gamma}^- \mathbf{e}_m\right)^n_{i,e} = 0.$$

Secondly, it is important to impose a zero value on the velocities normal to the walls in cases where the flux is not crossing the cell wall. In 2D problems, when working with meshes that are not aligned with the x- and y axes (i.e. is unstructured), such as in the example presented in Fig. 1, this second step is not trivial. It is necessary to carefully cancel the perpendicular component to the wet/dry wall and keep the parallel component. Once the water discharges in both directions have been calculated using the numerical scheme, they are redirected to cancel the components perpendicular to the walls that meet the wet/dry conditions. For this purpose, as many intermediate update steps are necessary as the number of walls in each cell

$$\begin{aligned}
(q_x)^{n+1,m}_i &= a^{WD}_{i,m} (q_x)^{n+1,m-1}_i + b^{WD}_{i,m} (q_y)^{n+1,m-1}_i, \\
(q_y)^{n+1,m}_i &= c^{WD}_{i,m} (q_x)^{n+1,m-1}_i + d^{WD}_{i,m} (q_y)^{n+1,m-1}_i,
\end{aligned} \tag{11}$$

where $m = 1, \ldots, I_f$ and $i = 1, \ldots, I_c$; with $(q_x)^{n+1,0}_i$ and $(q_y)^{n+1,0}_i$ are the update states given by the numerical scheme (A.2) and (A.3). The coefficients are computed as follows

$$a^{WD}_{i,m} = \begin{cases} \sin^2\left(\alpha'_{i,m}\right), & \text{if treated}, \\ 1, & \text{otherwise}, \end{cases} \qquad b^{WD}_{i,m} = \begin{cases} \sin\left(\alpha'_{i,m}\right)\cos\left(\alpha'_{i,m}\right), & \text{if treated}, \\ 0, & \text{otherwise}, \end{cases}$$

$$c^{WD}_{i,m} = \begin{cases} \cos\left(\alpha'_{i,m}\right)\sin\left(\alpha'_{i,m}\right), & \text{if treated}, \\ 0, & \text{otherwise}, \end{cases} \qquad d^{WD}_{i,m} = \begin{cases} \cos^2\left(\alpha'_{i,m}\right), & \text{if treated} \\ 1, & \text{otherwise}. \end{cases} \tag{12}$$

where the α' angle is

$$\alpha'_{i,m} = \begin{cases} \alpha_{i,m}, & \text{if } \alpha_{i,m} = \frac{n}{2}\pi, \text{ with } n = 0,1,2,3,4, \\ \alpha_{i,m} + \frac{\pi}{2}, & \text{otherwise}. \end{cases}$$

and

$$\alpha_{i,m} = \begin{cases} \arctan\left(\frac{n_y}{n_x}\right)_{i,m}, & \text{if } (n_y)_{i,m} \geq 0, \\ \pi + \arctan\left(\frac{n_y}{n_x}\right)_{i,m}, & \text{if } (n_y)_{i,m} < 0. \end{cases}$$

3. Reduced-order model

The POD-based ROM strategy consists of two phases: (I) the off-line phase, in which the ROM is trained using solutions computed with the FOM; and (II) the on-line phase, in which the ROM is solved.

The set of N_T time numerical solutions computed by the FOM, or *training solutions*, are assembled in the so-called *snapshot matrices*
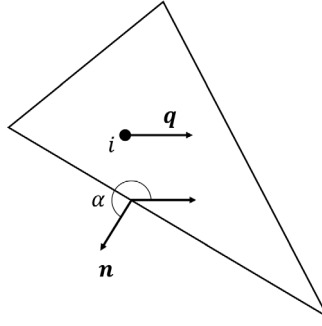
Fig. 1. Wet/dry treatment: angle α between water discharge q and normal vector n to the eth wall of the ith cell.

$$M_h = \begin{pmatrix} h_1^1 & \cdots & h_1^{N_T} \\ \vdots & \vdots & \vdots \\ h_{I_c}^1 & \cdots & h_{I_c}^{N_T} \end{pmatrix}, \quad M_{q_x} = \begin{pmatrix} (q_x)_1^1 & \cdots & (q_x)_1^{N_T} \\ \vdots & \vdots & \vdots \\ (q_x)_{I_c}^1 & \cdots & (q_x)_{I_c}^{N_T} \end{pmatrix}, \quad M_{q_y} = \begin{pmatrix} (q_y)_1^1 & \cdots & (q_y)_1^{N_T} \\ \vdots & \vdots & \vdots \\ (q_y)_{I_c}^1 & \cdots & (q_y)_{I_c}^{N_T} \end{pmatrix}.$$

The POD [27] of these matrices decomposes them into orthogonal components, also called *POD modes*, by means of the *singular value decomposition* (SVD, [45])

$$M_h = \mathbf{\Phi} \mathbf{\Sigma} \mathbf{\Psi}^T,$$

where $\mathbf{\Sigma} \in \mathbb{R}^{I_c \times N_T}$ is a diagonal matrix whose entries of the main diagonal are the singular values of M_h and represent the magnitude of each POD mode; $\mathbf{\Phi} \in \mathbb{R}^{I_c \times I_c}$ and $\mathbf{\Psi} \in \mathbb{R}^{N_T \times N_T}$ are orthogonal matrices. Matrix $\mathbf{\Phi} = (\phi_1, \ldots, \phi_{I_c})$ with $\phi_k = (\phi_{1,k}, \ldots, \phi_{I_c,k})^T$ consists of the orthogonal eigenvectors of $M_h M_h^T$ which are used to define the reduced space. The snapshot matrices related to the water discharges q_x and q_y are decomposed in a similar way.

Due to the accumulation of most of the energy/information in the principal POD modes (sorted in descending order), it is feasible to truncate the dimension of the reduced space without losing much precision with respect to the snapshot matrix. Let M_{POD} be a positive integer such that $M_{\text{POD}} \ll \min\left(N_T, I_c\right)$ and it will be chosen as small as possible without significantly affecting the accuracy of the computed solution with the reduced-order method [46–50].

The intrusive POD-based ROM is an alternative numerical scheme that needs to be developed from a standard numerical scheme by projecting the numerical variables from the physical space to the reduced space. The Galerkin method [28] acts as the projection between these two spaces

$$h_i^n \approx \sum_{k=1}^{M_{\text{POD}}} \hat{h}_k^n \phi_{i,k}, \quad (q_x)_i^n \approx \sum_{k=1}^{M_{\text{POD}}} (\hat{q}_x)_k^n \varphi_{i,k}, \quad (q_y)_i^n \approx \sum_{k=1}^{M_{\text{POD}}} (\hat{q}_y)_k^n \psi_{i,k}, \tag{13}$$

where $\{\phi_{i,k}\}, \{\varphi_{i,k}\}, \{\psi_{i,k}\}$ are the reduced space basis functions provided by the POD/SVD; and $\hat{h}_k^n$, $(\hat{q}_x)_k^n$ and $(\hat{q}_y)_k^n$ are the reduced coefficients that depend on time. If the maximum number of POD modes is used, the reconstruction of the Galerkin method recovers the solution in the physical space exactly, i.e., (13) it is an equality.

As explained in the previous work [51], the Galerkin decomposition method cannot be applied directly to the ARoe-based FOM given in (A.1), (A.2) and (A.3), because of the presence of the conserved variables in denominators and within square roots. The *proper interval decomposition* (PID, [29–31]) allows the development of the linearized ARoe-based ROM to speed-up computational times at the same time that enables the generation of shocks and rarefaction waves in the solutions computed by the ROM. According to the PID, the snapshot matrix is divided into different time windows from each of which a different reduced space is defined. Once the ROM completes a time window, it is projected back to the physical space and then projected to the following reduced space. For more information, see [51].

3.1. Dirichlet BCs

When considering Dirichlet-type BCs, whether or not they are time dependent, the development of the ROM is not directly feasible. The combination of time and space dependence makes it impossible to model it only during the training phase: it is necessary that the ROM receives information about the time evolution of the boundary instantaneously. There are some proposals in the literature for dealing with this type of BCs [35,36]. However, in this paper, it is proposed a resolution of the Dirichlet BCs integrated in the ROM that allows to solve it in time.

For example, if Dirichlet BCs are considered for h such that $h = h_0$ on Γ^D and the rest of the cells of the boundary $i \in J^N$ are imposed free BCs, the following approximation is used in (A.1)

$$\sum_{i \in J} \left(\sum_{k=1}^{M_{\text{POD}}} \hat{h}_k^n \phi_{i,k} \right) \phi_{i,p} = \sum_{i \in J^D} \left(\sum_{k=1}^{M_{\text{POD}}} \hat{h}_k^n \phi_{i,k} \right) \phi_{i,p} + \sum_{i \in J^I \cup J^N} \left(\sum_{k=1}^{M_{\text{POD}}} \hat{h}_k^n \phi_{i,k} \right) \phi_{i,p}$$

$$\approx \sum_{i \in J^D} \left(h_0^n\right)_i^n \phi_{i,p} + \sum_{i \in J^I \cup J^N} \left(\sum_{k=1}^{M_{POD}} \hat{h}_k^n \phi_{i,k}\right) \phi_{i,p}. \tag{14}$$

The same splitting procedure applies to the water discharges q_x and q_y in case Dirichlet BCs are imposed on them. Eqs. (15), (16) and (17) show how these BCs are integrated into the ROM.

The Neumann BCs are implemented in the ROM as indicated in Appendix B.

3.2. Augmented Roe-based reduced-order model

The ROM is developed from the numerical expression (A.1), (A.2) and (A.3) of the FOM by applying the Galerkin method (13) to the variables of interest and projecting it to the reduced space (first by multiplying it by a basis function $\phi_{i,p}$ and then by integrating it to the whole physical space).

Conversely, the velocities $(u_x)_i^n$ and $(u_y)_i^n$ are not reduced, so that computational time is saved by not projecting back to the physical space to update its values from those of the conserved variables. Instead, they are time-averaged following the PID method.

The **2D ARoe-based ROM** is obtained by applying the Galerkin decomposition (13) to the 2D ARoe-based FOM given in (A.1), (A.2) and (A.3)

$$\hat{h}_p^{n+1} = \sum_{i \in J^D} \left(h_0\right)_i^n \phi_{i,p} + \sum_{k=1}^{M_{POD}} A_{pk}^h \hat{h}_k^n + \Delta t \sum_{k=1}^{M_{POD}} B_{pk}^h \hat{h}_k^n$$
$$+ \Delta t \sum_{k=1}^{M_{POD}} C_{pk}^h \left(\hat{q}_x\right)_k^n + \Delta t \sum_{k=1}^{M_{POD}} D_{pk}^h \left(\hat{q}_y\right)_k^n, \tag{15}$$

$$\left(\hat{q}_x\right)_p^{n+1} = \sum_{i \in J^D} \left(q_{x,0}\right)_i^n \varphi_{i,p} + \sum_{k=1}^{M_{POD}} A_{pk}^{q_x} \left(\hat{q}_x\right)_k^n + \Delta t \sum_{k=1}^{M_{POD}} B_{pk}^{q_x} \hat{h}_k^n$$
$$+ \Delta t \sum_{k=1}^{M_{POD}} C_{pk}^{q_x} \left(\hat{q}_x\right)_k^n + \Delta t \sum_{k=1}^{M_{POD}} D_{pk}^{q_x} \left(\hat{q}_y\right)_k^n, \tag{16}$$

$$\left(\hat{q}_y\right)_p^{n+1} = \sum_{i \in J^D} \left(q_{y,0}\right)_i^n \psi_{i,p} + \sum_{k=1}^{M_{POD}} A_{pk}^{q_y} \left(\hat{q}_y\right)_k^n + \Delta t \sum_{k=1}^{M_{POD}} B_{pk}^{q_y} \hat{h}_k^n$$
$$+ \Delta t \sum_{k=1}^{M_{POD}} C_{pk}^{q_y} \left(\hat{q}_x\right)_k^n + \Delta t \sum_{k=1}^{M_{POD}} D_{pk}^{q_y} \left(\hat{q}_y\right)_k^n, \tag{17}$$

where h_0, $q_{x,0}$, $q_{y,0}$ are the functions defining the Dirichlet BCs and the rest of the coefficients A_{pk}^h, B_{pk}^h, … are given in Appendix B. Note that these reduced coefficients are time independent within each time window and also independent of the Dirichlet BCs.

The *accuracy* obtained by the solutions calculated by the ROM will be evaluated. This is measured by means of the differences with respect to the solutions calculated by the FOM at each time step using the weighted L_1 norm defined as

$$\|d^h\|_1 = \frac{1}{L_x \times L_y} \sum_{i \in J} S_i |h_i^{FOM} - h_i^{ROM}|. \tag{18}$$

The differences of the water discharges $\|d^{q_x}\|_1$ and $\|d^{q_y}\|_1$ are computed in a similar way.

It will be also possible to study the *speed-up* achieved by the ROM by dividing the CPU time required by the FOM, τ_{CPU}^{FOM}, by that of the ROM, τ_{CPU}^{ROM}, both measured in seconds.

3.3. Numerical corrections in the reduced space

The ARoe-based ROM in (15), (16) and (17) requires the inclusion of the numerical corrections explained in Section 2.2, as evidenced in the test cases in Section 4. For a detailed discussion of this, see [52]. The other three numerical corrections need a specialized treatment during the ROM off-line phase and even in its on-line phase in the case of the wet/dry treatment, as shown below.

3.3.1. Friction correction to avoid reverse flow

The modification proposed by the friction correction to avoid reverse flows in (7) replaces β_f by q^{**}. While β_f depends only on h, the intermediate state q^{**} depends on all variables h, q_x and q_y. Because of this multiple dependence, the contribution of the reduced version of q^{**} has to be split into the corresponding ROM coefficients of (B.1), as indicated in Appendix C. In addition to this, the contribution of $\left(\tilde{\beta}_f\right)_{i,e}^n$ has to be removed from the corresponding ROM coefficient.

All these modifications of the reduced coefficients are applied to them during the off-line phase and after they have been computed as indicated in (B.1). If, for example, the friction source term needs to be corrected on the first wall of the ith cell, the reduced coefficients of the reduced water depth $\hat{h}$ in (15) are modified as follows

$$B_{pk}^h \mapsto B_{pk}^h + \frac{l_{i,1}}{S_i} \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{h}}^{RF}\right)_{i,1}^w \phi_{i,p} - \frac{l_{i,1}}{S_i} \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_f\right)_{i,1}^w \phi_{i,p},$$

$$C_{pk}^h \mapsto C_{pk}^h + \frac{l_{i,1}}{S_i}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_x}^{RF}\right)_{i,1}^w \phi_{i,p},$$

$$D_{pk}^h \mapsto D_{pk}^h + \frac{l_{i,1}}{S_i}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_y}^{RF}\right)_{i,1}^w \phi_{i,p},$$

with

$$\left(\hat{\beta}_{\hat{h}}^{RF}\right)_{i,1}^w = \frac{1}{2}\left[\delta\phi_{i,1,k}\left(\lambda_1 + \frac{\tilde{\mathbf{u}}\mathbf{n}}{\tilde{c}}\right)_{i,1}^w - \tilde{\phi}_{i,1,k}\frac{g\delta z_{i,1}}{\tilde{c}_{i,1}^w}\right], \quad \left(\hat{\beta}_{\hat{q}_x}^{RF}\right)_{i,1}^w = \left[\varphi_{i,k} - \delta\varphi_{i,1,k}\frac{1}{2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{c}}\right)_{i,1}^w\right](n_x)_{i,1},$$

$$\left(\hat{\beta}_{\hat{q}_y}^{RF}\right)_{i,1}^w = \left[\psi_{i,k} - \delta\psi_{i,1,k}\frac{1}{2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{c}}\right)_{i,1}^w\right](n_y)_{i,1}, \quad \left(\beta_f\right)_{i,1}^w = \tilde{\phi}_{i,1,k}\frac{\tilde{\mathbf{u}}_{i,1}^w \cdot \mathbf{n}_{i,1}|\tilde{\mathbf{u}}_{i,1}^w|\left(\tilde{n}_b^2 d_n\right)_{i,1}}{\tilde{c}_{i,1}^w \max\left(h_i^w, h_j^w\right)^{4/3}}.$$

The reduced coefficients of the reduced water discharges $\hat{q}_x$ and $\hat{q}_y$ in (16) and (17), respectively, are modified as indicated in (C.1).

3.3.2. Entropy fix

The entropy fix in the reduced domain consists of a redefinition of the numerical eigenvalues during the off-line phase following (8). Furthermore, it is necessary to cancel the contribution of $\tilde{\beta}_3^-$ to the ROM coefficients (B.1) according to (9).

Again, these two modifications of the reduced coefficients are applied to them during the off-line phase and after they have been computed as indicated in (B.1). If, for example, the entropy needs to be fixed for the third component of the eigenvectors $\tilde{\lambda}_3$ on the first wall of the ith cell, the reduced coefficients of the reduced water depth $\hat{h}$ in (15) are modified as follows

$$B_{pk}^h \mapsto B_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\phi_{i,1,k}\left[\tilde{\lambda}_3^-\left(1 - \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \phi_{i,p} - \frac{l_{i,1}}{2S_i}\delta\phi_{i,1,k}\left[\tilde{\lambda}_3^\leftarrow\left(1 - \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \phi_{i,p} - \frac{g}{2}\frac{l_{i,1}}{S_i}\left(\hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \phi_{i,p},$$

$$C_{pk}^h \mapsto C_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_x)_{i,1}\phi_{i,p} - \frac{l_{i,1}}{2S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde{\lambda}_3^\leftarrow}{\tilde{c}}\right)_{i,1}^w (n_x)_{i,1}\phi_{i,p},$$

$$D_{pk}^h \mapsto D_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\psi_{i,1,k}\left(\frac{\tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_y)_{i,1}\phi_{i,p} - \frac{l_{i,1}}{2S_i}\delta\psi_{i,1,k}\left(\frac{\tilde{\lambda}_3^\leftarrow}{\tilde{c}}\right)_{i,1}^w (n_y)_{i,1}\phi_{i,p}.$$

The reduced coefficients of the reduced water discharges $\hat{q}_x$ and $\hat{q}_y$ in (16) and (17), respectively, are modified as indicated in (C.2).

3.3.3. Wet/dry treatment

The wet/dry treatment, as explained in Section 2, is applied to the ARoe-based FOM given in (A.1), (A.2) and (A.3) to avoid unphysical solutions in which volume cells that should stay dry get wet.

Let consider the reduced coefficients of the ARoe-based ROM (15) and assume that at the first wall of the ith cell that connects it to the jth cell the wet/dry condition is met, i.e. $h_j + z_j < z_i$. In that case, all information that would be sent to the ith cell is returned to the jth cell. This is translated in the ROM in such a way that in the construction of its coefficients the component that would be projected on the ith cell is projected on the jth cell; so that, the reduced coefficients of the reduced water depth $\hat{h}$ in (15) are modified as follows

$$\begin{aligned}
B_{pk}^h \mapsto {} & B_{pk}^h + \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\tilde{\lambda}_1^-\left(1 + \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right) + \tilde{\lambda}_3^-\left(1 - \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \phi_{i,p} \\
& - \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\tilde{\lambda}_1^-\left(1 + \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right) + \tilde{\lambda}_3^-\left(1 - \frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \phi_{j,p} \\
& - \frac{g}{2}\frac{l_{i,1}}{S_i}\left(\hat{\beta}_1\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} + \hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \phi_{i,p} + \frac{g}{2}\frac{l_{i,1}}{S_i}\left(\hat{\beta}_1\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} + \hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \phi_{j,p} \\
C_{pk}^h \mapsto {} & C_{pk}^h - \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_x)_{i,1}\phi_{i,p} + \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_x)_{i,1}\phi_{j,p}, \\
D_{pk}^h \mapsto {} & D_{pk}^h - \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_y)_{i,1}\phi_{i,p} + \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_{i,1}^w (n_y)_{i,1}\phi_{j,p}.
\end{aligned}$$

This is evaluated during the off-line phase and is extended to the rest of the coefficients of the ARoe-based FOM to complete the first wet/dry step as indicated in (C.3).

As explained in Section 2.2, once the updating of the numerical scheme has been completed, it is necessary to correct both components of the water discharge in the x and y directions to cancel the component perpendicular to the wall involved in the wet/dry. Since this is done by two additional Eqs. (11) that are solved in each time step, it is necessary to obtain a reduced version as follows

$$\left(\hat{q}_x\right)_p^{n+1} = \sum_{k=1}^{M_{POD}} \hat{a}_{pk}^{WD}\left(\hat{q}_x\right)_p^{n+1}\varphi_{i,k}\varphi_{i,p} + \sum_{k=1}^{M_{POD}} \hat{b}_{pk}^{WD}\left(\hat{q}_y\right)_p^{n+1}\psi_{i,k}\varphi_{i,p},$$
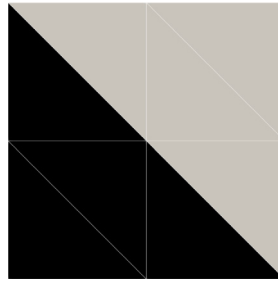
Fig. 2. Case 1. 2D mesh with initial water depth.

$$\left(\hat{q}_y\right)_p^{n+1} = \sum_{k=1}^{M_{\text{POD}}} \hat{c}_{pk}^{WD} \left(\hat{q}_x\right)_p^{n+1} \varphi_{i,k}\psi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} \hat{d}_{pk}^{WD} \left(\hat{q}_y\right)_p^{n+1} \psi_{i,k}\psi_{i,p}. \tag{19}$$

where

$$\begin{pmatrix} \hat{a}_{pk}^{WD} & \hat{b}_{pk}^{WD} \\ \hat{c}_{pk}^{WD} & \hat{d}_{pk}^{WD} \end{pmatrix} = \prod_{m=1}^{I_f} \begin{pmatrix} a_{i,m}^{WD} & b_{i,m}^{WD} \\ c_{i,m}^{WD} & d_{i,m}^{WD} \end{pmatrix}, \tag{20}$$

with the coefficients $a_{i,m}^{WD}$, $a_{i,m}^{WD}$, $a_{i,m}^{WD}$ and $a_{i,m}^{WD}$ given in (12) that depend only on space. This is evaluated during the on-line phase, after the updating of the reduced water discharges according to (16) and (17).

4. Test cases to evaluate each of the numerical corrections

These numerical cases serve to test whether numerical corrections need to be taken into account in the ROM. They have been designed to be studied individually without the other corrections intervening at the same time.

The objective is to recover the training solution that has been previously corrected in the three scenarios previously described during the computation of the FOM. To do this, two first solutions are obtained, one with and one without correction in the ROM. In addition, it is necessary to use the maximum number of POD modes and as many time windows as time steps to obtain the training solution with machine accuracy.

An additional solution has also been included in each case to evaluate the effect of including the correction while using small values of the ROM parameters. In this way, it is intended to check whether the error introduced by the ROM completely screens out those of the corrections and, therefore, whether they are worth taking into account or not.

Case 1. Friction correction to avoid reverse flow

A 2D dam-break problem is to be solved, where the initial conditions are defined as follows

$$h(x, y, 0) = \begin{cases} 0.05, & \text{if } x + y \le 20, \\ 0, & \text{if } 20 < x + y, \end{cases} \quad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0. \tag{21}$$

There is no bed slope (i.e., $z \equiv 0$) and the Manning coefficient is $n_b = 0.03$. The time–space domain $(x, y, t) \in [0, 20] \times [0, 20] \times [0, 20]$ is divided into $I_c = 8$ triangular cells, as shown in Fig. 2, where the initial water depth is represented in greyscale according to (21). All boundaries are considered as solid walls.

The three different results presented in this test case are used to illustrate the need to include the correction of the friction term. As shown in Table 1, in Case 1.1 the ROM is solved without the friction correction; while in the rest of them, it is corrected.

Apart from that, Cases 1.2 and 1.3 are used to study the effect of the setting parameters of the ROM, namely the number of POD modes M_{POD} and the number of time windows M_W. While in Case 1.2 the ROM uses the maximum number of POD modes, 8, and one time window per time step, so that the training solution can be recovered with machine precision; in Case 1.3 only 3 POD modes are used, and the number of time windows is reduced from 11 to 4.

Errors introduced by not including the friction correction in the ROM (Case 1.1) grow immediately, as it can be seen in Fig. 3(a). The ROM solution computed in Case 1.2, on the contrary, matches with machine precision the FOM solution, as indicated by the differences shown in Fig. 3(b).

If the number of POD modes and the number of time windows are decreased to 3 and 4, respectively, as in Case 1.3, the error remains in the same order of magnitude (Fig. 3(c)), still smaller than that of Case 1.1, as indicated in Table 2. In all these figures, the vertical grey lines represent the limits of the time windows. On the one hand, the ROM in Cases 1.1 and 1.2 is much slower

Table 1
Case 1. Problem subcases and settings.

Case	M_{POD}	M_W	Friction correction
1.1	8	11	No
1.2	8	11	Yes
1.3	3	4	Yes

$L_x \times L_y = 20 \times 20$, $T = 20$, IC: Eq. (21), BCs: Wall
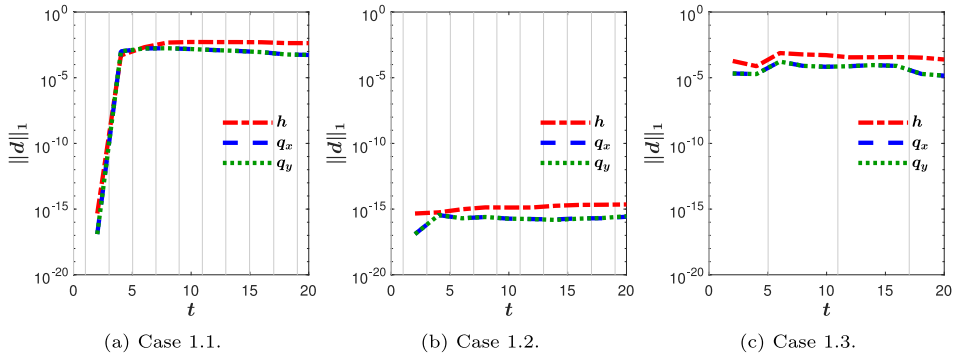z: 0, $n_b = 0.03$, CFL $= 0.4$, $I_c = 8$, $N_T = 11$

(a) Case 1.1.   (b) Case 1.2.   (c) Case 1.3.

Fig. 3. Case 1. Differences $\|d\|_1$ between the FOM and the ROM solutions.

Table 2
Case 1. Results of efficiency.

Case	1.1	1.2	1.3
$\|d^h\|_1$	$4.30 \cdot 10^{-3}$	$2.25 \cdot 10^{-15}$	$2.45 \cdot 10^{-4}$
$\|d^{q_x}\|_1$	$5.40 \cdot 10^{-4}$	$2.62 \cdot 10^{-16}$	$1.44 \cdot 10^{-5}$
$\|d^{q_y}\|_1$	$5.40 \cdot 10^{-4}$	$2.88 \cdot 10^{-16}$	$1.43 \cdot 10^{-5}$
$\tau_{\text{CPU}}^{\text{FOM}}$	$3.00 \cdot 10^{-4}$	$3.00 \cdot 10^{-4}$	$3.00 \cdot 10^{-4}$
$\tau_{\text{CPU}}^{\text{ROM}}$	$3.00 \cdot 10^{-3}$	$2.00 \cdot 10^{-3}$	$4.00 \cdot 10^{-6}$
Speed-up	$\times 0$	$\times 0$	$\times 75$

than the FOM, because it is solved using the maximum number of POD modes, just as it were the FOM itself, but with the addition of the change between time windows. On the other hand, in Case 1.3 the ROM is 75 times faster than the FOM (see Table 2).

Taking into account these results and the fact that the inclusion of the friction correction is only involved in the off-line phase, it is advisable to take it into account in order to avoid possible erroneous solutions.

Case 2. Entropy fix.

A 2D dam-break problem is to be solved, where the initial conditions are defined as follows

$$h(x, y, 0) = \begin{cases} 2, & \text{if } x \leq 10, \\ 0.1, & \text{if } 10 < x, \end{cases} \qquad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0. \tag{22}$$

There is no bed slope (i.e., $z \equiv 0$) and the Manning coefficient is $n_b = 0$. The time–space domain $(x, y, t) \in [0, 20] \times [0, 20] \times [0, 5]$ is divided into $I_c = 36$ cells, as shown in Fig. 4, where the initial water depth is represented in greyscale according to (22). All boundaries are considered as solid walls.

In Case 2.1 the ROM does not include the entropy fix, whereas in the rest of the cases it does, as shown in Table 3. Again, Case 2.3 is used to check how this correction interacts with the error introduced by the ROM using small values of its setting parameters.

The inclusion of the entropy fix in the resolution of the ROM allows the obtention of the training solution with machine accuracy, as shown in the results of Case 2.2 in Fig. 5(b).

Similarly to Case 1, the error of the ROM in Case 2.3 is smaller than in Case 2.1, as indicated in Table 4. It can be therefore concluded that this correction should be included in the resolution of the ROM as it does not involve loss of time calculation and improves the results.
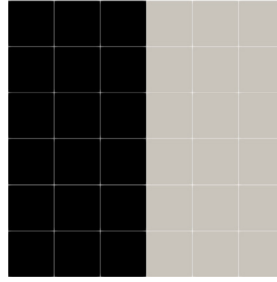
Fig. 4. Case 2. 2D mesh with initial water depth.

Table 3
Case 2. Problem subcases and settings.

Case	M_{POD}	M_W	Entropy fix
2.1	36	21	No
2.2	36	21	Yes
2.3	3	7	Yes

$L_x \times L_y = 20 \times 20$, $T = 5$, IC: Eq. (22), BCs: Wall
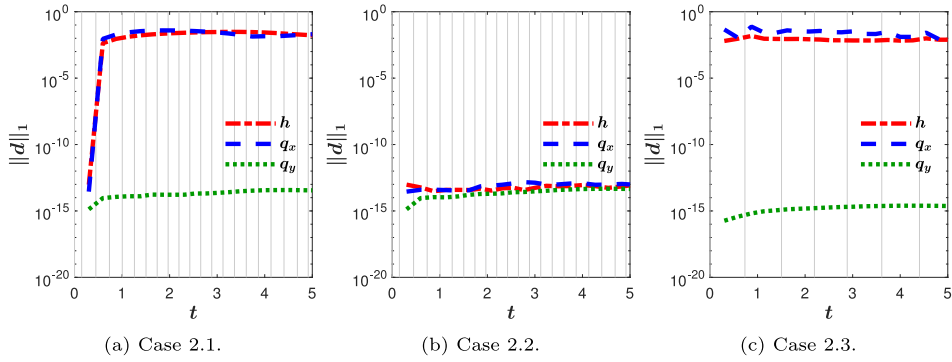z: 0, $n_b = 0$, CFL = 0.4, $I_c = 36$, $N_T = 21$

(a) Case 2.1. (b) Case 2.2. (c) Case 2.3.

Fig. 5. Case 2. Differences $\|d\|_1$ between the FOM and the ROM solutions.

Table 4
Case 2. Results of efficiency.

Case	2.1	2.2	2.3
$\|d^h\|_1$	$1.67 \cdot 10^{-2}$	$8.36 \cdot 10^{-14}$	$7.80 \cdot 10^{-3}$
$\|d^{q_x}\|_1$	$2.03 \cdot 10^{-2}$	$9.85 \cdot 10^{-14}$	$8.26 \cdot 10^{-3}$
$\|d^{q_y}\|_1$	$3.56 \cdot 10^{-14}$	$4.53 \cdot 10^{-14}$	$2.35 \cdot 10^{-15}$
$\tau_{\text{CPU}}^{\text{FOM}}$	$2.50 \cdot 10^{-3}$	$2.50 \cdot 10^{-3}$	$2.50 \cdot 10^{-3}$
$\tau_{\text{CPU}}^{\text{ROM}}$	$7.44 \cdot 10^{-4}$	$7.51 \cdot 10^{-4}$	$1.20 \cdot 10^{-5}$
Speed-up	$\times 3$	$\times 3$	$\times 208$

Case 3. Wet/dry treatment.

A 2D dambreak is proposed to study how the inclusion of the wet/dry treatment affects the solution of the ARoe-based ROM. The time–space domain is defined as $(x, y, t) \in [0, 6] \times [0, 6] \times [0, 10]$. The initial condition is defined as

$$h(x, y, 0) = \begin{cases} 1, & \text{if } x \leq 3, \\ 0, & \text{if } 3 < x, \end{cases} \quad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0, \tag{23}$$

with the following bed level

$$z(x, y) = \begin{cases} 0, & \text{if } x \leq 3, \\ 2, & \text{if } 3 < x. \end{cases} \tag{24}$$

Fig. 6. Case 3. 2D mesh with the two subregions of the bed elevation.

Table 5
Case 3. Problem subcases and settings.

Case	M_{POD}	M_W	Wet/dry treatment
3.1	16	31	No
3.2	16	31	Yes
3.3	3	11	Yes

$L_x \times L_y = 6 \times 6$, $T = 510$, IC: Eq. (23), BCs: Eq. (25)
z: Eq. (24), $n_b = 0$, CFL $= 0.4$, $I_c = 16$, $N_T = 31$

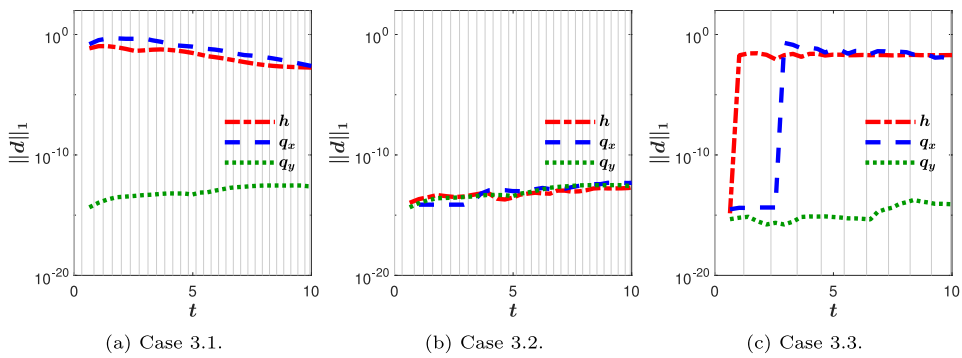(a) Case 3.1.   (b) Case 3.2.   (c) Case 3.3.

Fig. 7. Case 3. Differences $\|d\|_1$ between the FOM and the ROM solutions.

A constant water discharge is imposed along the west boundary ($x = 0$)

$$q_x(0, y, t) = 4, \tag{25}$$

with $0 < y < 6$ and $t > 0$; and free boundary conditions on the rest of the boundary.

The west boundary of the spatial domain is filled up until the water level rises above the obstacle and continues to flow freely through the east boundary. In the process of filling, the wet/dry treatment acts so that the cells to the right of the obstacle do not get wet, as the water level has not yet reached its height.

The spatial domain has been discretized using 16 rectangular cells, as shown in Fig. 6. The black volume cells indicate the obstacle that water must overtop.

Cases 3.1 and 3.2 are used to study the consequences if the wet/dry treatment is not included in the ROM. It is worth remembering that this correction, in addition to the off-line phase, is also involved in the on-line phase when resolving the ROM. On the other hand, Case 3.3 is used to test whether it is worth including or not. All the settings of the problem are shown in Table 5.

As indicated by the differences shown in Fig. 7 for Cases 3.1 and 3.2, the ARoe-based ROM in (15), (16) and (17) is able to recover the training solution with machine precision only in the case the wet/dry treatment is applied. Due to the special treatment proposed in (14), the Dirichlet BCs are also reconstructed with machine precision.

Even though the error at the final time is smaller in Case 3.1 than in Case 3.3, the solution may be wetting areas that should not be wetted. So it can be concluded that, due to the inclusion of the wet/dry treatment, ROMs can obtain physically satisfactory results. In addition, this correction, far from what it might seem, does not add a significant overhead to the on-line ROM calculation. Table 6 shows that the speed-ups reach two orders of magnitude in Case 3.3.

Table 6

Case 3. Results of efficiency.

Case	3.1	3.2	3.3
$\|d^h\|_1$	$1.79 \cdot 10^{-3}$	$2.09 \cdot 10^{-13}$	$1.96 \cdot 10^{-2}$
$\|d^{q_x}\|_1$	$2.46 \cdot 10^{-3}$	$4.89 \cdot 10^{-13}$	$8.18 \cdot 10^{-3}$
$\|d^{q_y}\|_1$	$2.45 \cdot 10^{-13}$	$2.62 \cdot 10^{-13}$	$8.08 \cdot 10^{-15}$
$\tau_{\text{CPU}}^{\text{FOM}}$	$2.40 \cdot 10^{-3}$	$2.40 \cdot 10^{-3}$	$2.40 \cdot 10^{-3}$
$\tau_{\text{CPU}}^{\text{ROM}}$	$1.00 \cdot 10^{-4}$	$1.00 \cdot 10^{-4}$	$1.00 \cdot 10^{-5}$
Speed-up	$\times 24$	$\times 24$	$\times 240$

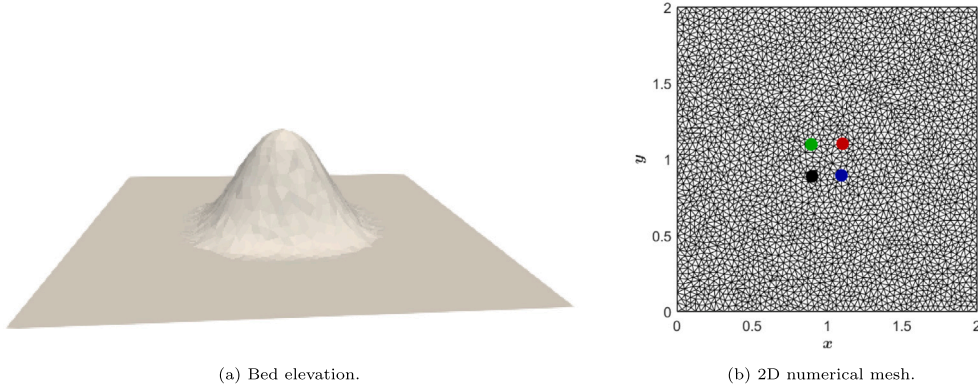(a) Bed elevation.   (b) 2D numerical mesh.

Fig. 8. Case 4. 2D bed elevation (left) and numerical mesh (right). Coloured dots represent the position of the probes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 7

Case 4. Problem settings.

$L_x \times L_y$	T	IC	BC	z	n_b	CFL	I_c	N_T	M_{POD}	M_W
2×2	0.15	Eq. (26)	Wall	Eq. (27)	0	0.6	6195	709	Eq. (28)	Eq. (28)

5. Application cases

In this section, four numerical application cases set in realistic scenarios are presented in which the aforementioned numerical corrections are applied both to the FOM and the ROM when necessary.

Case 4. 2D circular dam-break on a non-flat bed.

This case, designed as a 2D circular dam break on a non-flat bottom, is inspired by the one solved in [53–56]. The position of the discontinuity in the water depth and the bottom hump have been centred to evaluate the symmetry of the problem. The time–space domain of the case is defined as $(x, y, t) \in [0, 2] \times [0, 2] \times [0, 0.15]$. Initially, the water depth is defined as a circular discontinuity at rest

$$h(x, y, 0) = \begin{cases} 1.1 - z(x, y), & \text{if } (x - 1)^2 + (y - 1)^2 \leq 0.01, \\ 0.6 - z(x, y), & \text{otherwise,} \end{cases} \qquad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0, \tag{26}$$

where the bed is

$$z(x, y) = \begin{cases} z_1(x, y), & \text{if } (x - 1)^2 + (y - 1)^2 \leq 0.25, \\ 0, & \text{otherwise,} \end{cases} \tag{27}$$

with

$$z_1(x, y) = \frac{1}{8} \left[1 + \cos\left(2\pi\left(x - 1\right)\right) \right] \left[\cos\left(2\pi\left(y - 1\right)\right) \right].$$

The bed elevation is shown in Fig. 8(a) and the IC in Fig. 10(a). Closed walls in all boundaries and no friction are considered. The spatial domain is discretized using $I_c = 6195$ unstructured elements, as shown in Fig. 8(b). All these settings are shown in Table 7.

The ROM has been solved using the values indicated in (28) for the number of POD modes M_{POD} and time windows M_W. From the combination of all the values of M_{POD} and M_W, 64 different results have been obtained for this problem. In this way, the optimal

Table 8
Case 4. Differences between FOM and ROM solutions and CPU times required.

Case	M_{POD}	M_W	$\|d^h\|_1$	$\|d^{q_x}\|_1$	$\|d^{q_y}\|_1$	$\tau_{\text{CPU}}^{\text{FOM}}$	$\tau_{\text{CPU}}^{\text{ROM}}$	Speed-up
4a	1	7	$2.85 \cdot 10^{-3}$	$5.00 \cdot 10^{-3}$	$4.98 \cdot 10^{-3}$	$2.20 \cdot 10^1$	$5.00 \cdot 10^{-4}$	×43 946
4b	3	7	$1.88 \cdot 10^{-4}$	$4.51 \cdot 10^{-4}$	$4.16 \cdot 10^{-4}$	$2.20 \cdot 10^1$	$1.50 \cdot 10^{-3}$	×14 649

(a) Differences of h.

(b) Differences of q_x.

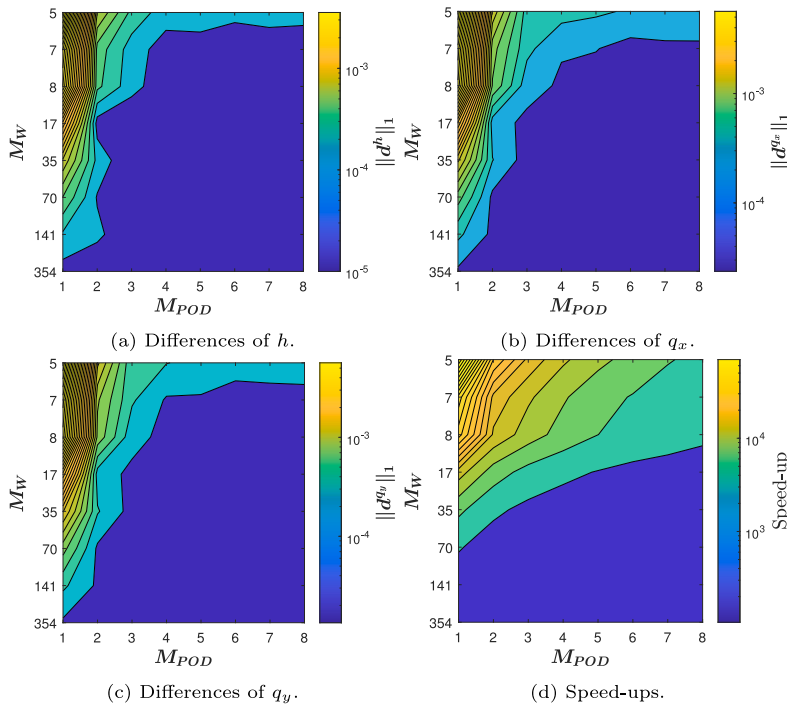(c) Differences of q_y.

(d) Speed-ups.

Fig. 9. Case 4. Differences between FOM and ROM solutions and speed-ups.

values of these ROM parameters are studied in terms of the accuracy of the solution measured using the differences (18) between the FOM and the ROM solutions, and speed-up achieved by the ROM.

$$M_{\text{POD}} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$M_W \in \{354, 141, 70, 35, 17, 8, 7, 5\} \tag{28}$$

Figs. 9(a), 9(b) and 9(c) show the differences of h, q_x and q_y at the final time, respectively. These figures indicate that the accuracy of the solution computed by the ROM is adequate when using at least 3 POD modes and at least 7 time windows. As for the speed-up, as can be seen in Fig. 9(d), it achieves very large values, reaching up to four orders of magnitude when 7 time windows and 4 or less POD modes are used. From this it can be concluded that the optimal values of the parameters defining the ROM resolution are 3 POD modes and 7 time windows. The total number of time steps N_T may change from one problem to another, so the number of time windows, which depends on the first, is not directly generalizable. In this sense, it is useful to know the number of snapshots per time window, which in this particular case is 100. Therefore, in all other numerical cases, 3 POD modes and around 100 snapshots per time window will be used.

Four probes have been set at the points

$$P_1 = (0.9, \ 0.9), \quad P_2 = (1.1, \ 0.9), \quad P_3 = (1.1, \ 1.1), \quad P_4 = (0.9, \ 1.1), \tag{29}$$

to study how the ROM performs the symmetry of the problem. The position of these probes is indicated with black, red, blue and green dots within the spatial domain in Fig. 8(b). Fig. 11 shows the results of Cases 4a (1 POD modes and 7 time windows) and 4b (3 POD modes and 7 time windows) at these probes, where it can be seen that there are large differences between using a single POD mode, with which non-smooth solutions are obtained, and using 3 POD modes, with which the ROM reproduces the solution of the FOM very accurately. The differences and the speed-ups obtained for these two cases are shown in Table 8. The solution of the water depth h computed by the ROM with the optimal values of its settings (Case 4b) is shown in Fig. 10 at different time steps.
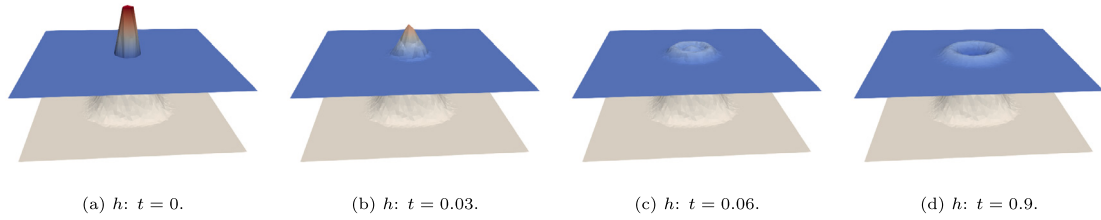
(a) h: $t = 0$.   (b) h: $t = 0.03$.   (c) h: $t = 0.06$.   (d) h: $t = 0.9$.

Fig. 10. Case 4b. ROM solutions of h at different time steps.

(a) Case 4a.   (b) Case 4b.
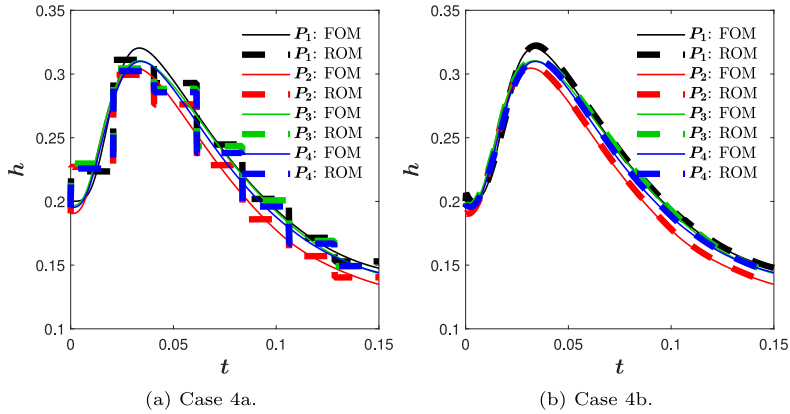
Fig. 11. Case 4. Water depth at probes.

Table 9
Case 5. Problem settings.

$L_x \times L_y$	T	IC	BC	z	n_b	CFL	I_c	N_T	M_{POD}	M_W
6×6	5	Eq. (30)	Eq. (32) + wall	Eq. (31)	0.01	0.5	3763	3346	3	34

Case 5. 2D water depth source.

This test case is designed to prove that ROMs are very useful to reproduce solutions with injections at inner-domain points, such as point sources of water depth. The time–space domain of the case is defined as $(x, y, t) \in [0, 6] \times [0, 6] \times [0, 5]$. Initially, all the domain is dry

$$h(x, y, 0) = 0, \quad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0, \tag{30}$$

and the bed is flat

$$z(x, y) = 0, \tag{31}$$

with $0 \le x, y \le 6$. From $t > 0.1$, water starts to enter according to the following function

$$h(x, y) = \begin{cases} 0, & \text{if } t < 0.1, \\ 0.5 - 0.56(1 - t), & \text{if } 0.1 \le t < 1, \\ 0.5, & \text{if } 1 \ge t, \end{cases} \tag{32}$$

and $(x, y) \in \Omega^S$, with $\Omega^S = \{(x, y), (x - 3)^2 + (y - 3)^2 \le 4\}$, as shown in Fig. 12(a). This injection in the centre of the domain is considered as Dirichlet points when solving the ROM. Closed walls are considered in all boundaries and the Manning coefficient is $n_b = 0.01$. All these settings are shown in Table 9.

The spatial domain is discretized using $I_c = 3763$ unstructured elements, as shown in Fig. 12(b), where the injection points in which the water depth is imposed are plotted in blue.

The numerical results obtained with the ROM can be seen in Fig. 14 for different times. The quasi-stationary regime at the beginning of the simulation is solved by the ARoe-based ROM (15), (16) and (17) with a wide time window and then 34 time windows with a fixed number of 100 snapshots per window. There are no significant differences between its solutions and those of
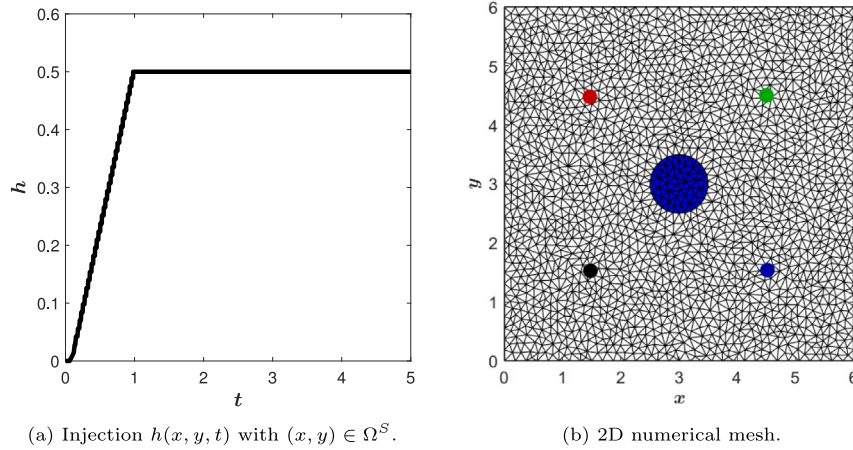
(a) Injection $h(x, y, t)$ with $(x, y) \in \Omega^S$.

(b) 2D numerical mesh.

Fig. 12. Case 5. Water depth injected in the centre of the domain (left) and 2D numerical mesh (right) with injection volume cells indicated in blue. Coloured dots represent the position of the probes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
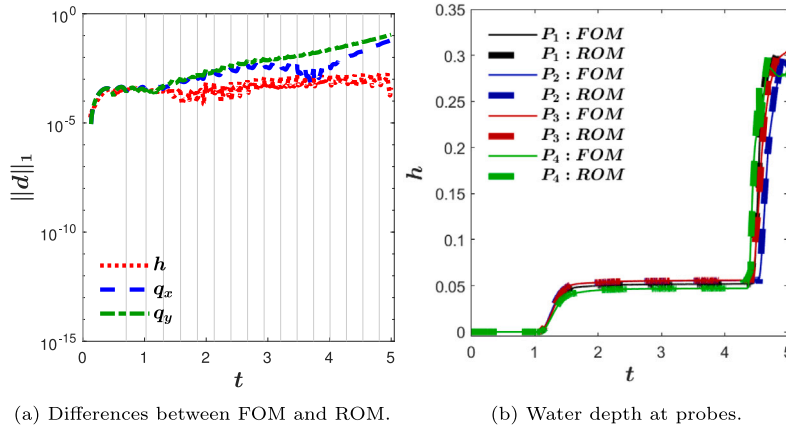

(a) Differences between FOM and ROM.

(b) Water depth at probes.

Fig. 13. Case 5. Differences $\|d\|_1$ between FOM and ROM solutions and time evolution of water depth at probes (33).

Table 10

Case 5. Differences between FOM and ROM solutions and CPU times required.

$\|d^h\|_1$	$\|d^{q_x}\|_1$	$\|d^{q_y}\|_1$	$\tau_{\text{CPU}}^{\text{FOM}}$	$\tau_{\text{CPU}}^{\text{ROM}}$	Speed-up
$4.15 \cdot 10^{-4}$	$6.19 \cdot 10^{-2}$	$1.09 \cdot 10^{-1}$	$3.29 \cdot 10^{1}$	$7.50 \cdot 10^{-3}$	$\times 4382$

the FOM. Furthermore, the differences between the solutions of the FOM and the ROM have been computed to test the accuracy of the latter. Fig. 13(a) shows the time evolution of the differences, alongside the limits of time windows depicted by the vertical grey lines.

Four probes have been set at the points

$$P_1 = (1.5, \ 1.5), \quad P_2 = (4.5, \ 1.5), \quad P_3 = (1.5, \ 4.5), \quad P_4 = (4.5, \ 4.5), \tag{33}$$

to study how the ROM performs the symmetry of the problem. The position of these probes is indicated with black, red, blue and green dots within the spatial domain in Fig. 12(b). Fig. 13(b) shows strong agreement between the solutions of the FOM (solid lines) and the ROM (dashed lines) in all probes.

Finally, the CPU times required by the FOM and the ROM are shown in the Table 10, as well as the speed-up achieved in this case, which reaches 3 orders of magnitude. That table also shows the differences between the FOM and the ROM solutions computed at the final time.
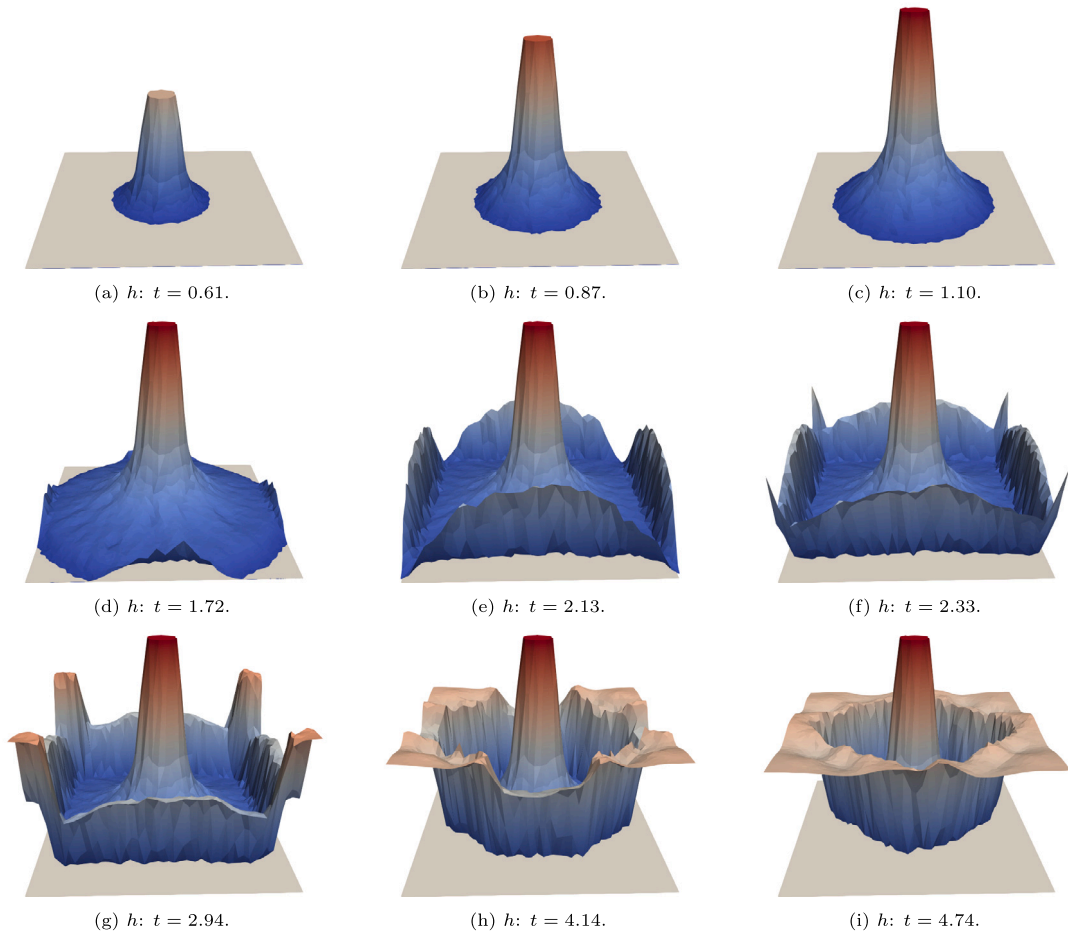
(a) h: $t = 0.61$.   (b) h: $t = 0.87$.   (c) h: $t = 1.10$.

(d) h: $t = 1.72$.   (e) h: $t = 2.13$.   (f) h: $t = 2.33$.

(g) h: $t = 2.94$.   (h) h: $t = 4.14$.   (i) h: $t = 4.74$.

Fig. 14. Case 5. ROM solutions of h at different time steps.

Case 6. 2D filling of floodplain depressions.

This numerical case is the second benchmark proposed in [57], where a floodplain with multiple depressions is filled by a water depth hydrograph entering a region in the western boundary $(0, y)$ with $y \in (1900, 2000)$.

The time–space domain of the case is defined as $(x, y, t) \in [0, 2000] \times [0, 2000] \times [0, 6000]$. The bed elevation of the case is shown in Fig. 15(a) with the space discretization. The BCs are defined as the following water discharge hydrograph

$$q(0, y, t) = \begin{cases} 0, & \text{if } 0 < t \le 300, \\ 20, & \text{if } 300 < t \le 5100, \\ 0, & \text{if } 5100 < t \le 6000, \end{cases} \tag{34}$$

with $y \in (1900, 2000)$, shown in Fig. 16(a); and wall BCs are imposed on the rest of the domain. The entire domain is dry at the beginning

$$h(x, y, 0) = 0, \quad q_x(x, y, 0) = 0, \quad q_y(x, y, 0) = 0, \tag{35}$$

and the Manning coefficient is $n_b = 0.03$.

The spatial domain is discretized using an unstructured mesh $I_c = 6243$ volume cells, as shown in Fig. 16(b). In Fig. 16(c) the boundary points in which the water depth is imposed are plotted in blue.

The ROM is solved using 3 POD modes and 53 time windows. These data and the rest of the settings are shown in Table 11.

Fig. 17 shows the time evolution of h at different times by the ROM. The water enters the domain from the north-west corner and fills the floodplain depressions as it advances. At $t = 5100$, the discharge inflow is cut off and the water is properly distributed until it stops.
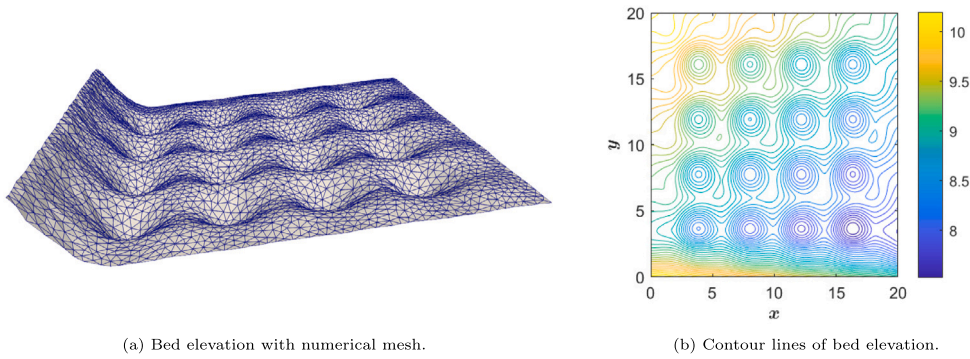
(a) Bed elevation with numerical mesh.

(b) Contour lines of bed elevation.

Fig. 15. Case 6. Bed elevation.

(a) Injection h at $(x, y) \in J^D$.

(b) 2D mesh.
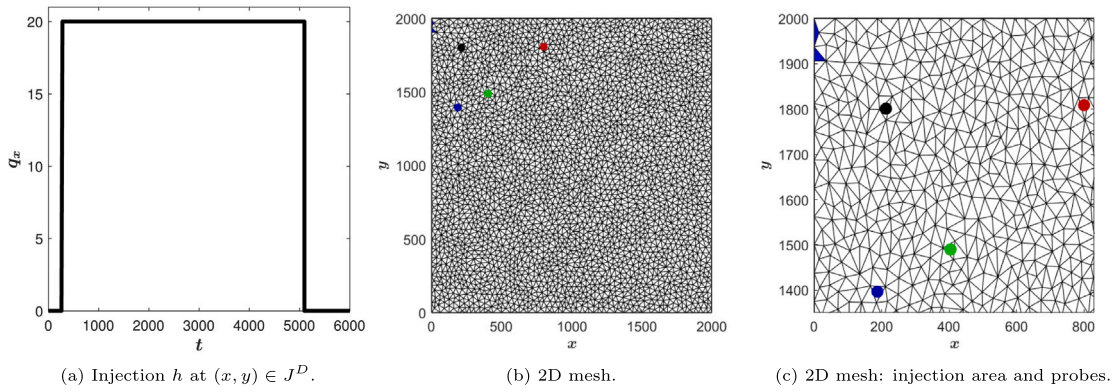

(c) 2D mesh: injection area and probes.

Fig. 16. Case 6. Water depth injected in the western boundary (left) and 2D numerical mesh (centre and right) with injection volume cells indicated in blue. Coloured dots represent the position of the probes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 11
Case 6. Problem settings.

$L_x \times L_y$	T	IC	BC	z	n_b	CFL	I_c	N_T	M_{POD}	M_W
2000×2000	6000	Eq. (35)	Eq. (34) + wall	Fig. 15(a)	0.03	0.4	6243	10 570	3	53

Table 12
Case 6. Differences between FOM and ROM solutions and CPU times required.

$\|d^h\|_1$	$\|d^{q_x}\|_1$	$\|d^{q_y}\|_1$	$\tau_{\text{CPU}}^{\text{FOM}}$	$\tau_{\text{CPU}}^{\text{ROM}}$	Speed-up
$1.80 \cdot 10^{-3}$	$7.68 \cdot 10^{-4}$	$4.69 \cdot 10^{-3}$	$2.25 \cdot 10^{1}$	$1.45 \cdot 10^{-2}$	$\times 1555$

The differences between the solutions computed with the FOM and the ROM are shown in Fig. 18(a). It is shown that they present a good agreement between the solutions computed with the FOM and the ROM and, moreover, the ROM is 1555 times faster than the FOM, as it is shown in Table 12.

Four probes have been set at the points

$$P_1 = (200, \; 1800), \quad P_2 = (200, \; 1400), \quad P_3 = (800, \; 1800), \quad P_4 = (400, \; 1500), \tag{36}$$

to study how the ROM performs with respect to the FOM. The position of these probes is indicated with black, red, blue and green dots within the spatial domain in Fig. 16(c), respectively. Fig. 18(b) shows a strong agreement between the solutions of the FOM and the ROM in all probes.

Case 7. 2D tsunami laboratory test case.

In this final case, the method presented in this work is tested in a 1/400 scale laboratory tsunami test case done by [58], and included in many other works [59–61]. The time–space domain of the case is defined as $(x, y, t) \in [0, 5.488] \times [0, 3.388] \times [0, 22.5]$.
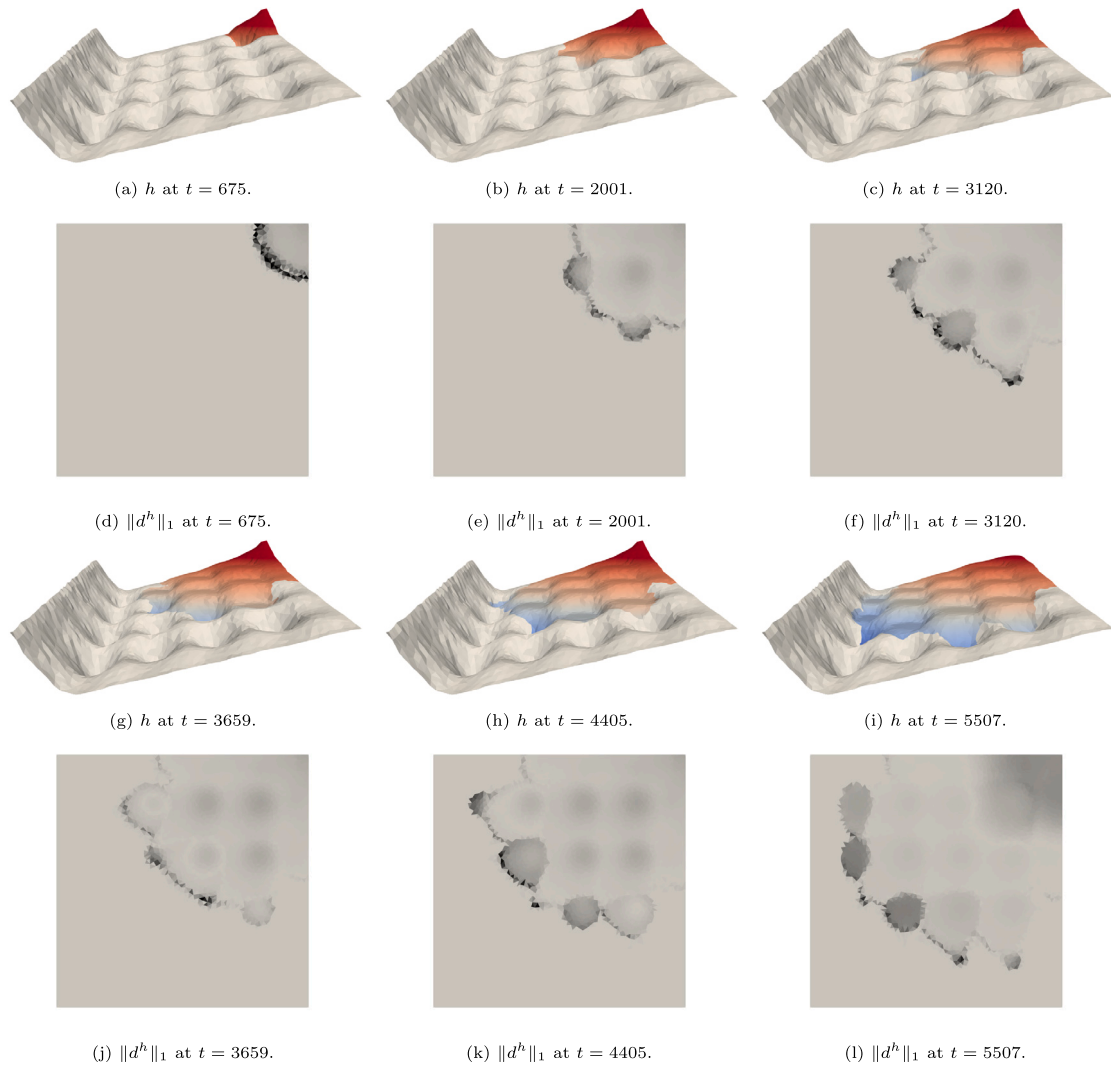
(a) h at $t = 675$.

(b) h at $t = 2001$.

(c) h at $t = 3120$.

(d) $\|d^h\|_1$ at $t = 675$.

(e) $\|d^h\|_1$ at $t = 2001$.

(f) $\|d^h\|_1$ at $t = 3120$.

(g) h at $t = 3659$.

(h) h at $t = 4405$.

(i) h at $t = 5507$.

(j) $\|d^h\|_1$ at $t = 3659$.

(k) $\|d^h\|_1$ at $t = 4405$.

(l) $\|d^h\|_1$ at $t = 5507$.

Fig. 17. Case 6. ROM solutions of h at different time steps and differences $\|d\|_1$ between FOM and ROM solutions.
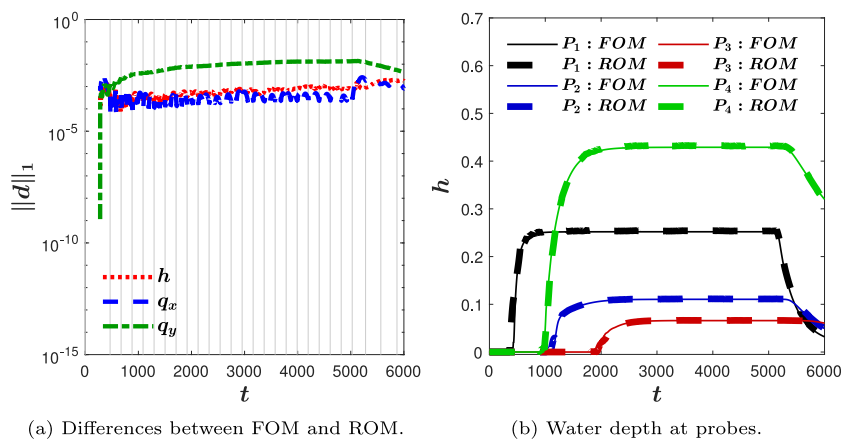

(a) Differences between FOM and ROM.

(b) Water depth at probes.

Fig. 18. Case 6. Differences $\|d\|_1$ between FOM and ROM solutions and time evolution of water depth at probes (36).
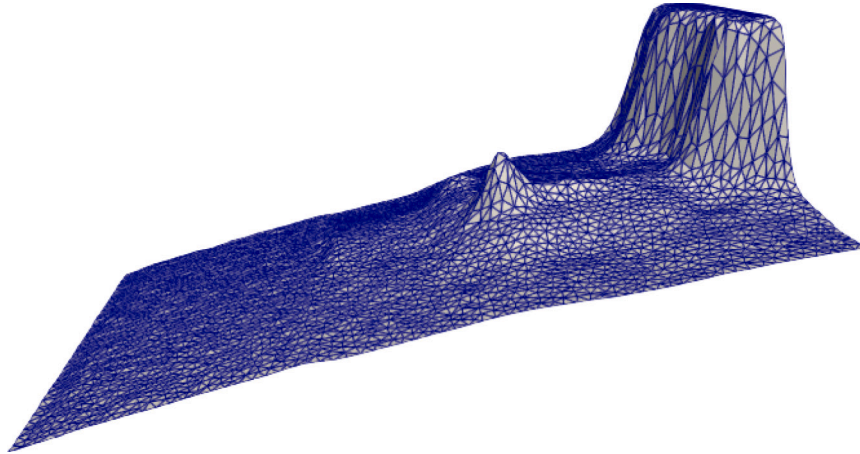
Fig. 19. Case 7. Bed elevation with numerical mesh.

Table 13
Case 7. POD setting, ROM CPU time and speed-ups achieved of each subcase.

$L_x \times L_y$	T	IC	BC	z	n_b	CFL	I_c	N_T	M_{POD}	M_W
5.488×3.388	22.5	Eq. (37)	Fig. 20(a) + wall	Fig. 19	0.01	0.5	5863	5447	3	36

Table 14
Case 7. Differences $\|d\|_1$ between FOM and ROM solutions and CPU times required.

$\|d\|_1^h$	$\|d\|_1^{q_x}$	$\|d\|_1^{q_y}$	τ_{CPU}^{FOM}	τ_{CPU}^{ROM}	Speed-up
$3.86 \cdot 10^{-4}$	$1.78 \cdot 10^{-3}$	$2.68 \cdot 10^{-4}$	$1.46 \cdot 10^{1}$	$1.05 \cdot 10^{-2}$	×1388

The bed elevation of the case is shown in Fig. 19. The boundary conditions are defined as walls in the north, south and east boundaries and as the water depth hydrograph shown in Fig. 20(a). The ICs are set as water at rest

$$h(x, y, 0) + z(x, y) = 0, \ 0 \le x \le 5.488, \ 0 \le y \le 3.388, \tag{37}$$

as shown in Fig. 23(a), and the Manning coefficient is $n_b = 0.01$.

The spatial domain is discretized using an unstructured mesh that consists of $I_c = 5863$ volume cells, as shown in Fig. 20(b), where the boundary points in which the water depth is imposed are plotted in blue. The volume cells represented in grey are those in which the bed level is positive, i.e., $z > 0$ m; and the red dots indicate the position of the three gauging points where the experimental data were measured, given by these coordinates

$$P_1 = (4.52, \ 1.196), \ P_2 = (4.52, \ 1.696), \ P_3 = (4.52, \ 2.196).$$

The ARoe-based ROM has been solved using 3 POD modes and 36 time windows, as shown in Table 13 with the rest of the settings of the problem.

The time evolution of the numerical solutions computed by the ARoe-based ROM (15), (16) and (17) for the water depth is shown in Fig. 23, represented in blue. From time instant $t = 10.81$ s (Fig. 23(b)) to $t = 14.98$ s (Fig. 23(c)), the water wave enters the left domain and advances towards the coast. At $t = 16.51$ s (Fig. 23(g)) it can be seen that the water rises along the dry coastal land. And then, once the wave has bounced, it returns to the inner sea (Figs. 23(h) and 23(i)).

Fig. 21 shows the good agreement between the FOM and the ROM solutions and the experimental data measured in the three gauging points. Mass is conserved as shown in Fig. 22(a), where some discrepancies between the solutions of the FOM and the ROM are observed. Using a non-maximum number of POD modes when solving the ROM can imply losses or gains in mass that have no physical significance. However, when comparing both numerical solutions with experimental data in Fig. 21, it can be seen that the general trend is well achieved.

The time windows have been homogeneously defined starting from the instant at which the hydrograph of the water depth varies. Until then, as can be seen in Fig. 22(b), a single wide window is defined that covers all the time in which the state of the problem does not change significantly. The differences, as can be seen here, show good levels of accuracy. Finally, as indicated in Table 14, the ARoe-based ROM is 1388 times faster than the ARoe-based FOM.
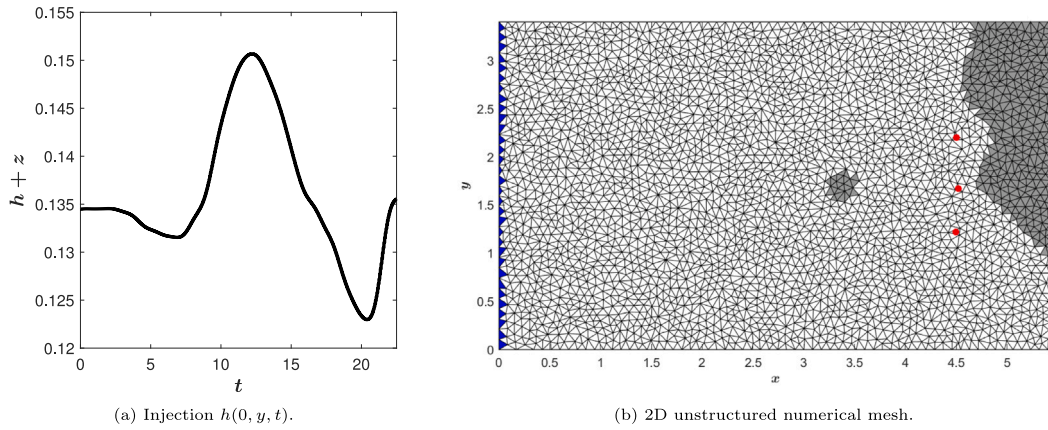
(a) Injection $h(0, y, t)$.

(b) 2D unstructured numerical mesh.

Fig. 20. Case 7. Water depth injected in the western boundary (left) and 2D numerical mesh (right) with injection volume cells indicated in blue. Coloured dots represent the position of the probes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
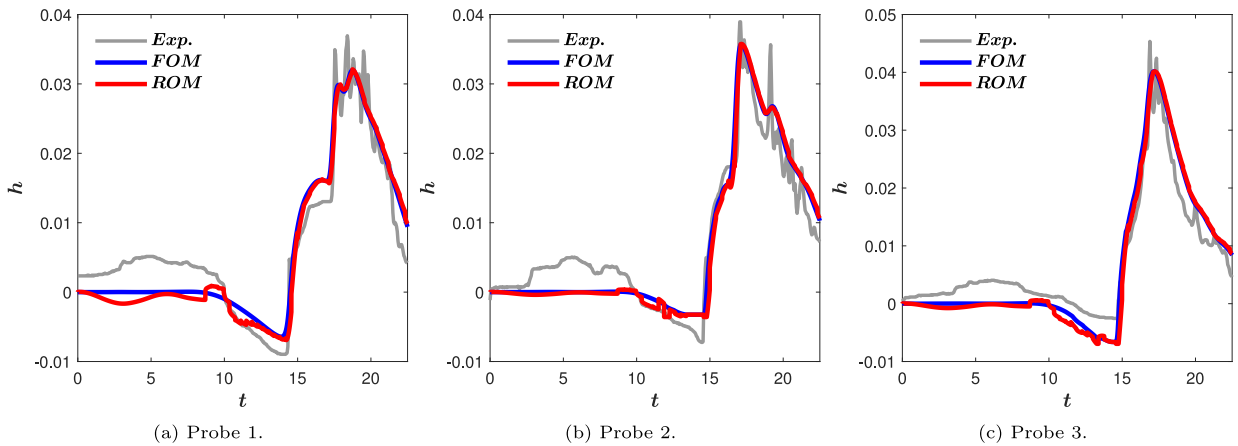

(a) Probe 1.

(b) Probe 2.

(c) Probe 3.

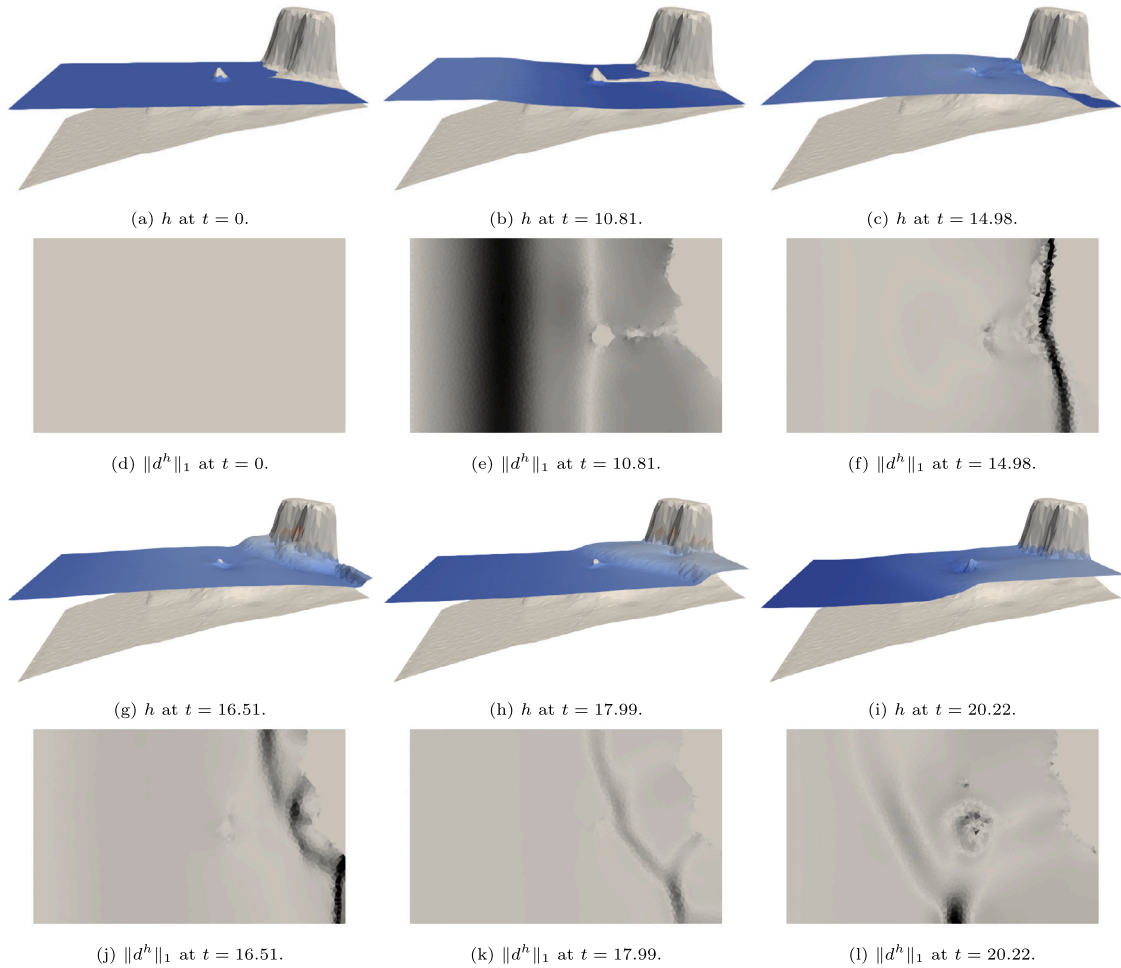**Fig. 21.** Case 7. Time evolution of water depth at 3 different probes.

(a) Mass conservation.

(b) $\|d\|_1$.
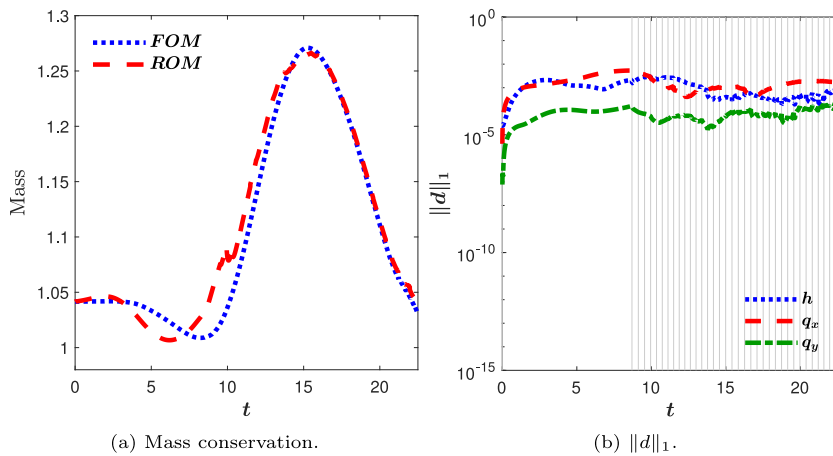
Fig. 22. Case 7. Time evolution of mass conservation and solution differences.

(a) h at $t = 0$.

(b) h at $t = 10.81$.

(c) h at $t = 14.98$.

(d) $\|d^h\|_1$ at $t = 0$.

(e) $\|d^h\|_1$ at $t = 10.81$.

(f) $\|d^h\|_1$ at $t = 14.98$.

(g) h at $t = 16.51$.

(h) h at $t = 17.99$.

(i) h at $t = 20.22$.

(j) $\|d^h\|_1$ at $t = 16.51$.

(k) $\|d^h\|_1$ at $t = 17.99$.

(l) $\|d^h\|_1$ at $t = 20.22$.

Fig. 23. Case 7. Time evolution of water depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Concluding remarks

In this paper, the application of intrusive POD-based ROMs to the 2D SWE has been carried out. The ARoe-based ROM has been developed in a linearized formulation by means of the PID method.

The necessity of including some numerical corrections in the ROM strategy to obtain proper solutions has been demonstrated. In particular, the friction source term correction to avoid reverse flow, the entropy fix and the wet/dry treatment. All of them operate in the off-line phase when solving 2D problems, except for the wet/dry treatment that is also included in the on-line phase, so it does contribute to the ROM efficiency computation.

The numerical results proposed (Cases 1, 2 and 3) for each of these corrections isolated show that they allow the recovery of the training solutions with machine accuracy when using the maximum number of POD modes and time windows. However, their inclusion in the ROM does not improve the accuracy of the solutions when using few POD modes, but it does ensure physics fulfilment. The three last numerical results (Cases 4, 5, 6 and 7) show large speed-ups achieved by the ROM in 2D problems with respect to the FOM.

In addition to this, it has been proposed a direct treatment of the Dirichlet-type BCs that allows the satisfactory resolution of ROMs while imposing time dependent BCs. These have been properly validated by means of Cases 3, 4, 5 and 6.

CRediT authorship contribution statement

P. Solán-Fustero: Writing – original draft, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **J.L. Gracia:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **A. Navas-Montilla:** Writing – review & editing, Writing – original draft, Supervision, Methodology,

Investigation, Data curation, Conceptualization. **P. García-Navarro:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Appendix A. ARoe-based FOM

The ARoe-based FOM of the 2D SWEs is

$$
\begin{aligned}
h_i^{n+1} = h_i^n &- \frac{\Delta t}{S_i} \frac{1}{2} \sum_{e=1}^{I_f} \delta h_{i,e}^n \left[\tilde{\lambda}_1^- \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) + \tilde{\lambda}_3^- \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \frac{1}{2} \sum_{e=1}^{I_f} \left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}} \right)_{i,e}^n \left[(\delta q_x)_{i,e}^n (n_x)_{i,e} + (\delta q_y)_{i,e}^n (n_y)_{i,e} \right] l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\tilde{\beta}_z + \tilde{\beta}_f)_{i,e}^n \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} \right)_{i,e}^n l_{i,e},
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
(q_x)_i^{n+1} = (q_x)_i^n &- \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} \delta h_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c} n_x}{2} \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right. \\
&\left. + \tilde{\lambda}_2^- (\tilde{u}_y n_x - \tilde{u}_x n_y) n_y + \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c} n_x}{2} \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\delta q_x)_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c} n_x}{2\tilde{c}} n_x - \tilde{\lambda}_2^- n_y n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c} n_x}{2\tilde{c}} n_x \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\delta q_y)_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c} n_x}{2\tilde{c}} n_y + \tilde{\lambda}_2^- n_x n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c} n_x}{2\tilde{c}} n_y \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\tilde{\beta}_z + \tilde{\beta}_f)_{i,e}^n \left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} (\tilde{u}_x - \tilde{c} n_x) - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} (\tilde{u}_x + \tilde{c} n_x) \right]_{i,e}^n l_{i,e},
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
(q_y)_i^{n+1} = (q_y)_i^n &- \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} \delta h_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c} n_y}{2} \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right. \\
&\left. - \tilde{\lambda}_2^- (\tilde{u}_y n_x - \tilde{u}_x n_y) n_x + \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c} n_y}{2} \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\delta q_x)_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c} n_y}{2\tilde{c}} n_x + \tilde{\lambda}_2^- n_x n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c} n_y}{2\tilde{c}} n_x \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\delta q_y)_{i,e}^n \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c} n_y}{2\tilde{c}} n_y - \tilde{\lambda}_2^- n_x n_x - \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c} n_y}{2\tilde{c}} n_y \right]_{i,e}^n l_{i,e} \\
&+ \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} (\tilde{\beta}_z + \tilde{\beta}_f)_{i,e}^n \left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} (\tilde{u}_y - \tilde{c} n_y) - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} (\tilde{u}_y + \tilde{c} n_y) \right]_{i,e}^n l_{i,e},
\end{aligned}
\tag{A.3}
$$

with $i \in J^I$ and where the source terms $(\tilde{\beta}_f)_{i,e}^n$ and $(\tilde{\beta}_f)_{i,e}^n$ are given in (3).

Appendix B. Coefficients of the ARoe-based ROM

The coefficients of the ARoe-based ROM (15), (16) and (17) are the following

$$
A_{pk}^h = \sum_{i \in J^I \cup J^N} \phi_{i,k} \phi_{i,p}, \quad A_{pk}^{q_x} = \sum_{i \in J^I \cup J^N} \varphi_{i,k} \varphi_{i,p}, \quad A_{pk}^{q_y} = \sum_{i \in J^I \cup J^N} \psi_{i,k} \psi_{i,p},
$$

$$B_{pk}^h = \sum_{i \in J^N} b_i^h \phi_{i,p} - \frac{1}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\phi_{i,e,k} \left[\tilde{\lambda}_1^- \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) + \tilde{\lambda}_3^- \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^w l_{i,e} \phi_{i,p}$$

$$+ \frac{g}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \left(\hat{\beta}_1 \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} + \hat{\beta}_3 \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} \right)_{i,e}^w l_{i,e} \phi_{i,p},$$

$$C_{pk}^h = \sum_{i \in J^N} c_i^h \phi_{i,p} + \frac{1}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\varphi_{i,e,k} \left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}} \right)_{i,e}^w (n_x)_{i,e} \, l_{i,e} \phi_{i,p},$$

$$D_{pk}^h = \sum_{i \in J^N} d_i^h \phi_{i,p} + \frac{1}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\psi_{i,e,k} \left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}} \right)_{i,e}^w (n_y)_{i,e} \, l_{i,e} \phi_{i,p},$$

$$B_{pk}^{q_x} = \sum_{i \in J^N} b_i^{q_x} \varphi_{i,p} - \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\phi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c}n_x}{2} \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right.$$

$$\left. + \tilde{\lambda}_2^- \left(\tilde{u}_y n_x - \tilde{u}_x n_y \right) n_y + \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c}n_x}{2} \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^w l_{i,e} \varphi_{i,p}$$

$$+ \frac{g}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \left[\hat{\beta}_1 \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} \left(\tilde{u}_x - \tilde{c}n_x \right) + \hat{\beta}_3 \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} \left(\tilde{u}_x + \tilde{c}n_x \right) \right]_{i,e}^w l_{i,e} \varphi_{i,p},$$

$$C_{pk}^{q_x} = \sum_{i \in J^N} c_i^{q_x} \varphi_{i,p} + \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\varphi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c}n_x}{2\tilde{c}} n_x - \tilde{\lambda}_2^- n_y n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c}n_x}{2\tilde{c}} n_x \right]_{i,e}^w l_{i,e} \varphi_{i,p},$$

$$D_{pk}^{q_x} = \sum_{i \in J^N} d_i^{q_x} \varphi_{i,p} + \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\psi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_x - \tilde{c}n_x}{2\tilde{c}} n_y + \tilde{\lambda}_2^- n_x n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_x + \tilde{c}n_x}{2\tilde{c}} n_y \right]_{i,e}^w l_{i,e} \varphi_{i,p},$$

$$B_{pk}^{q_y} = \sum_{i \in J^N} b_i^{q_y} \psi_{i,p} - \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\phi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c}n_y}{2} \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right.$$

$$\left. - \tilde{\lambda}_2^- \left(\tilde{u}_y n_x - \tilde{u}_x n_y \right) n_x + \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c}n_y}{2} \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^w l_{i,e} \psi_{i,p}$$

$$+ \frac{g}{2} \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \left[\hat{\beta}_1 \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} \left(\tilde{u}_y - \tilde{c}n_y \right) + \hat{\beta}_3 \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} \left(\tilde{u}_y + \tilde{c}n_y \right) \right]_{i,e}^w l_{i,e} \psi_{i,p},$$

$$C_{pk}^{q_y} = \sum_{i \in J^N} c_i^{q_y} \psi_{i,p} + \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\varphi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c}n_y}{2\tilde{c}} n_x + \tilde{\lambda}_2^- n_x n_y - \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c}n_y}{2\tilde{c}} n_x \right]_{i,e}^w l_{i,e} \psi_{i,p},$$

$$D_{pk}^{q_y} = \sum_{i \in J^N} d_i^{q_y} \psi_{i,p} + \sum_{i \in J^I} \frac{1}{S_i} \sum_{e=1}^{I_f} \delta\psi_{i,e,k} \left[\tilde{\lambda}_1^- \frac{\tilde{u}_y - \tilde{c}n_y}{2\tilde{c}} n_y - \tilde{\lambda}_2^- n_x n_x - \tilde{\lambda}_3^- \frac{\tilde{u}_y + \tilde{c}n_y}{2\tilde{c}} n_y \right]_{i,e}^w l_{i,e} \psi_{i,p}, \tag{B.1}$$

where $\delta\phi_{i,e,k} = \phi_{j,k} - \phi_{i,k}$, $\delta\varphi_{i,e,k} = \varphi_{j,k} - \varphi_{i,k}$, $\delta\psi_{i,e,k} = \psi_{j,k} - \psi_{i,k}$; and $(\hat{\beta}_1)_{i,e}^w = -(\hat{\beta}_3)_{i,e}^w = (\hat{\beta}_z + \hat{\beta}_f)_{i,e}^w$, with

$$(\hat{\beta}_z)_{i,e}^w = \bar{\phi}_{i,e,k} \frac{\delta z_{i,e}}{\tilde{c}_{i,e}^w}, \qquad (\hat{\beta}_f)_{i,e}^w = \bar{\phi}_{i,e,k} \frac{\tilde{\mathbf{u}}_{i,e}^w \cdot \mathbf{n}_{i,e} |\tilde{\mathbf{u}}_{i,e}^w| \, (\tilde{n}_b^2 d_n)_{i,e}}{\tilde{c}_{i,e}^w \max \left(h_i^w, h_j^w \right)^{4/3}},$$

and $\bar{\phi}_{i,e,k} = (\phi_{j,k} + \phi_{i,k})/2$. The index w indicates that the variable is averaged in the wth time window and the coefficients $b^h, c^h, d^h, b^{q_x}, c^{q_x}, d^{q_x}, b^{q_y}, c^{q_y}, d^{q_y}$ depend on the BCs imposed. For example, if free BCs are imposed at the first wall of the ith cell, then

$$b_i^h = -\frac{1}{2} \frac{1}{S_i} \sum_{e=2}^{I_f} \delta\phi_{i,e,k} \left[\tilde{\lambda}_1^- \left(1 + \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) + \tilde{\lambda}_3^- \left(1 - \frac{\tilde{\mathbf{u}} \cdot \mathbf{n}}{\tilde{c}} \right) \right]_{i,e}^w l_{i,e} \phi_{i,p}$$

$$+ \frac{1}{S_i} \sum_{e=2}^{I_f} \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3} \right)_{i,e}^w (\hat{\beta}_z + \hat{\beta}_f)_{i,e}^w l_{i,e} \phi_{i,p}.$$

The rest of the coefficients related to the BCs are determined similarly.

Appendix C. Modifications of ROM coefficients due to numerical corrections

Friction correction to avoid reverse flow

If, for example, the friction source term is corrected in the FOM on the first wall of the ith cell, the reduced coefficients presented in (B.1) need to include the following additional terms

$$
B_{pk}^h \mapsto B_{pk}^h + \frac{l_{i,1}}{S_i}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{h}}^{\mathrm{RF}} - \hat{\beta}_f\right)_{i,1}^w \phi_{i,p},
$$

$$
C_{pk}^h \mapsto C_{pk}^h + \frac{l_{i,1}}{S_i}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_x}^{\mathrm{RF}}\right)_{i,1}^w \phi_{i,p}, \quad D_{pk}^h \mapsto D_{pk}^h + \frac{l_{i,1}}{S_i}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_y}^{\mathrm{RF}}\right)_{i,1}^w \phi_{i,p},
$$

$$
B_{pk}^{q_x} \mapsto B_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{h}}^{\mathrm{RF}} - \hat{\beta}_f\right)_{i,1}^w \varphi_{i,p},
$$

$$
C_{pk}^{q_x} \mapsto C_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_x}^{\mathrm{RF}}\right)_{i,1}^w \varphi_{i,p}, \quad D_{pk}^{q_x} \mapsto D_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_y}^{\mathrm{RF}}\right)_{i,1}^w \varphi_{i,p},
$$

$$
B_{pk}^{q_y} \mapsto B_{pk}^{q_y} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{h}}^{\mathrm{RF}} - \hat{\beta}_f\right)_{i,1}^w \psi_{i,p},
$$

$$
C_{pk}^{q_y} \mapsto C_{pk}^{q_y} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_x}^{\mathrm{RF}}\right)_{i,1}^w \psi_{i,p}, \quad D_{pk}^{q_y} \mapsto D_{pk}^{q_y} + \frac{l_{i,1}}{S_i}\left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)_{i,1}^w \left(\hat{\beta}_{\hat{q}_y}^{\mathrm{RF}}\right)_{i,1}^w \psi_{i,p}, \tag{C.1}
$$

with

$$
\left(\hat{\beta}_{\hat{h}}^{\mathrm{RF}}\right)_{i,1}^w = \frac{1}{2}\left[\delta\phi_{i,1,k}\left(\lambda_1 + \frac{\bar{\mathbf{u}}\mathbf{n}}{\tilde{c}}\right)_{i,1}^w - \tilde{\phi}_{i,1,k}\frac{g\delta z_{i,1}}{\tilde{c}_{i,1}^w}\right], \quad \left(\hat{\beta}_{\hat{q}_x}^{\mathrm{RF}}\right)_{i,1}^w = \left[\varphi_{i,k} - \delta\varphi_{i,1,k}\frac{1}{2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{c}}\right)_{i,1}^w\right](n_x)_{i,1},
$$

$$
\left(\hat{\beta}_{\hat{q}_y}^{\mathrm{RF}}\right)_{i,1}^w = \left[\psi_{i,k} - \delta\psi_{i,1,k}\frac{1}{2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{c}}\right)_{i,1}^w\right](n_y)_{i,1}.
$$

Entropy fix

If, for example, the entropy needs to be fixed for the third component of the eigenvectors $\tilde{\lambda}_3$ on the first wall of the ith cell, the reduced coefficients presented in (B.1) need to include the following additional terms

$$
B_{pk}^h \mapsto B_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\phi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\left(1 - \frac{\bar{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \phi_{i,p} - \frac{g}{2}\frac{l_{i,1}}{S_i}\left(\hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_{i,1}^w \phi_{i,p},
$$

$$
C_{pk}^h \mapsto C_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}}{\tilde{c}}\right)_{i,1}^w (n_x)_{i,1}\phi_{i,p}, \quad D_{pk}^h \mapsto D_{pk}^h + \frac{l_{i,1}}{2S_i}\delta\psi_{i,1,k}\left(\frac{\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}}{\tilde{c}}\right)_{i,1}^w (n_y)_{i,1}\phi_{i,p},
$$

$$
B_{pk}^{q_x} \mapsto B_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_x + \tilde{c}n_x}{2}\left(1 - \frac{\bar{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \varphi_{i,p} - \frac{g}{2}\frac{l_{i,1}}{S_i}\left[\hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_x + \tilde{c}n_x\right)\right]_{i,1}^w \varphi_{i,p},
$$

$$
C_{pk}^{q_x} \mapsto C_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_x + \tilde{c}n_x}{2\tilde{c}}n_x\right]_{i,1}^w \varphi_{i,p},
$$

$$
D_{pk}^{q_x} \mapsto D_{pk}^{q_x} + \frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_x + \tilde{c}n_x}{2\tilde{c}}n_y\right]_{i,1}^w \varphi_{i,p},
$$

$$
B_{pk}^{q_y} \mapsto B_{pk}^{q_y} + \frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_y + \tilde{c}n_y}{2}\left(1 - \frac{\bar{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_{i,1}^w \psi_{i,p} - \frac{g}{2}\frac{l_{i,1}}{S_i}\left[\hat{\beta}_3\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_y + \tilde{c}n_y\right)\right]_{i,1}^w \psi_{i,p},
$$

$$
C_{pk}^{q_y} \mapsto C_{pk}^{q_y} + \frac{l_{i,e}}{S_i}\delta\varphi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_y + \tilde{c}n_y}{2\tilde{c}}n_x\right]_{i,1}^w \psi_{i,p},
$$

$$
D_{pk}^{q_y} \mapsto D_{pk}^{q_y} + \frac{l_{i,e}}{S_i}\delta\psi_{i,1,k}\left[\left(\tilde{\lambda}_3^- - \tilde{\lambda}_3^{\leftarrow}\right)\frac{\tilde{u}_y + \tilde{c}n_y}{2\tilde{c}}n_y\right]_{i,1}^w \psi_{i,p}. \tag{C.2}
$$

Wet/dry treatment

Let consider the reduced coefficients of the ARoe-based ROM (15) and assume that at the first wall of the ith cell that connects it to the jth cell the wet/dry condition is met, i.e., $h_j + z_j < z_i$. In that case, the reduced coefficients presented in (B.1) need to include the following additional terms

$$B_{pk}^h \mapsto B_{pk}^h + \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\tilde\lambda_1^-\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde c}\right)+\tilde\lambda_3^-\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde c}\right)\right]_{i,1}^w(\phi_{i,p}-\phi_{j,p})$$
$$-\frac{g}{2}\frac{l_{i,1}}{S_i}\left(\hat\beta_1\frac{\tilde\lambda_1^-}{\tilde\lambda_1}+\hat\beta_3\frac{\tilde\lambda_3^-}{\tilde\lambda_3}\right)_{i,1}^w(\phi_{i,p}-\phi_{j,p}),$$

$$C_{pk}^h \mapsto C_{pk}^h - \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left(\frac{\tilde\lambda_1^- - \tilde\lambda_3^-}{\tilde c}\right)_{i,1}^w(n_x)_{i,1}(\phi_{i,p}-\phi_{j,p}),$$

$$D_{pk}^h \mapsto D_{pk}^h - \frac{1}{2}\frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left(\frac{\tilde\lambda_1^- - \tilde\lambda_3^-}{\tilde c}\right)_{i,1}^w(n_y)_{i,1}(\phi_{i,p}-\phi_{j,p}),$$

$$B_{pk}^{q_x}\mapsto B_{pk}^{q_x}-\frac{g}{2}\frac{l_{i,1}}{S_i}\left[\hat\beta_1\frac{\tilde\lambda_1^-}{\tilde\lambda_1}(\tilde u_x-\tilde c n_x)+\hat\beta_3\frac{\tilde\lambda_3^-}{\tilde\lambda_3}(\tilde u_x+\tilde c n_x)\right]_{i,1}^w(\varphi_{i,p}-\varphi_{j,p})$$
$$+\frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_x-\tilde c n_x}{2}\left(1+\frac{\tilde{\mathbf u}\cdot\mathbf n}{\tilde c}\right)+\tilde\lambda_2^-(\tilde u_y n_x-\tilde u_x n_y)n_y+\tilde\lambda_3^-\frac{\tilde u_x+\tilde c n_x}{2}\left(1-\frac{\tilde{\mathbf u}\cdot\mathbf n}{\tilde c}\right)\right]_{i,1}^w(\varphi_{i,p}-\varphi_{j,p}),$$

$$C_{pk}^{q_x}\mapsto C_{pk}^{q_x}-\frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_x-\tilde c n_x}{2\tilde c}n_x-\tilde\lambda_2^- n_y n_y-\tilde\lambda_3^-\frac{\tilde u_x+\tilde c n_x}{2\tilde c}n_x\right]_{i,1}^w(\varphi_{i,p}-\varphi_{j,p}),$$

$$D_{pk}^{q_x}\mapsto D_{pk}^{q_x}-\frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_x-\tilde c n_x}{2\tilde c}n_y+\tilde\lambda_2^- n_x n_y-\tilde\lambda_3^-\frac{\tilde u_x+\tilde c n_x}{2\tilde c}n_y\right]_{i,1}^w(\varphi_{i,p}-\varphi_{j,p}),$$

$$B_{pk}^{q_y}\mapsto B_{pk}^{q_y}-\frac{g}{2}\frac{l_{i,1}}{S_i}\left[\hat\beta_1\frac{\tilde\lambda_1^-}{\tilde\lambda_1}(\tilde u_y-\tilde c n_y)+\hat\beta_3\frac{\tilde\lambda_3^-}{\tilde\lambda_3}(\tilde u_y+\tilde c n_y)\right]_{i,1}^w(\psi_{i,p}-\psi_{j,p})$$
$$+\frac{l_{i,1}}{S_i}\delta\phi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_y-\tilde c n_y}{2}\left(1+\frac{\tilde{\mathbf u}\cdot\mathbf n}{\tilde c}\right)-\tilde\lambda_2^-(\tilde u_y n_x-\tilde u_x n_y)n_x+\tilde\lambda_3^-\frac{\tilde u_y+\tilde c n_y}{2}\left(1-\frac{\tilde{\mathbf u}\cdot\mathbf n}{\tilde c}\right)\right]_{i,1}^w(\psi_{i,p}-\psi_{j,p}),$$

$$C_{pk}^{q_y}\mapsto C_{pk}^{q_y}-\frac{l_{i,1}}{S_i}\delta\varphi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_y-\tilde c n_y}{2\tilde c}n_x+\tilde\lambda_2^- n_x n_y-\tilde\lambda_3^-\frac{\tilde u_y+\tilde c n_y}{2\tilde c}n_x\right]_{i,1}^w(\psi_{i,p}-\psi_{j,p}),$$

$$D_{pk}^{q_y}\mapsto D_{pk}^{q_y}-\frac{l_{i,1}}{S_i}\delta\psi_{i,1,k}\left[\tilde\lambda_1^-\frac{\tilde u_y-\tilde c n_y}{2\tilde c}n_y-\tilde\lambda_2^- n_x n_x-\tilde\lambda_3^-\frac{\tilde u_y+\tilde c n_y}{2\tilde c}n_y\right]_{i,1}^w(\psi_{i,p}-\psi_{j,p}).\tag{C.3}$$

Data availability

Data will be made available on request.

References

[1] R.J. LeVeque, Finite volume methods for hyperbolic problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002, http://dx.doi.org/10.1017/CBO9780511791253.

[2] J.D. Anderson, Computational Fluid Dynamics: The Basics with Applications, McGraw-Hill Education, ISBN: 9780070016859, 1995, URL https://books.google.es/books?id=dJceAQAAIAAJ.

[3] P.W. McDonald, The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades, in: Turbo Expo: Power for Land, Sea, and Air, ASME 1971 International Gas Turbine Conference and Products Show, 1971, V001T01A089, http://dx.doi.org/10.1115/71-GT-89.

[4] W. Maccormack, A.J. Paullay, Computational efficiency achieved by time splitting of finite difference operators, in: 10th Aerospace Sciences Meeting, AIAA, 1972, http://dx.doi.org/10.2514/6.1972-154, URL https://arc.aiaa.org/doi/abs/10.2514/6.1972-154.

[5] A.W. Rizzi, M. Inouye, Time-split finite-volume method for three-dimensional blunt-body flow, AIAA J. 11 (11) (1973) 1478–1485, http://dx.doi.org/10.2514/3.50614.

[6] S.K. Godunov, I. Bohachevsky, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Matematičeskij Sbornik 47 (89) (1959) 271–306, URL https://hal.archives-ouvertes.fr/hal-01620642.

[7] C. Hirsch, Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics, Elsevier, 2007.

[8] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, Springer Berlin Heidelberg, ISBN: 9783540498346, 2009, URL https://books.google.es/books?id=SqEjX0um8o0C.

[9] I. Echeverribar, M. Morales-Hernández, P. Brufau, P. García-Navarro, 2D numerical simulation of unsteady flows for large scale floods prediction in real time, Adv. Water Resour. (ISSN: 0309-1708) 134 (2019) 103444, http://dx.doi.org/10.1016/j.advwatres.2019.103444, URL https://www.sciencedirect.com/science/article/pii/S0309170819304786.

[10] P. García-Navarro, M.E. Vázquez-Cendón, On numerical treatment of the source terms in the shallow water equations, Comput. & Fluids (ISSN: 0045-7930) 29 (8) (2000) 951–979, http://dx.doi.org/10.1016/S0045-7930(99)00038-9, URL https://www.sciencedirect.com/science/article/pii/S0045793099000389.

[11] R.J. LeVeque, Wave Propagation Algorithms for Multidimensional Hyperbolic Systems, J. Comput. Phys. (ISSN: 0021-9991) 131 (2) (1997) 327–353, http://dx.doi.org/10.1006/jcph.1996.5603, URL https://www.sciencedirect.com/science/article/pii/S002199619965603X.

[12] A. Navas-Montilla, P. Solán-Fustero, J. Murillo, P. García-Navarro, Discontinuous Galerkin well-balanced schemes using augmented Riemann solvers with application to the shallow water equations, J. Hydroinformatics (ISSN: 1464-7141) 22 (5) (2020) 1038–1058, http://dx.doi.org/10.2166/hydro.2020.206.

[13] C. Parés, M. Castro, On the well-balance property of roe's method for nonconservative hyperbolic systems. Applications to shallow-water systems, ESAIM Math. Model. Numer. Anal. 38 (5) (2004) 821–852, http://dx.doi.org/10.1051/m2an:2004041.

[14] A. Harten, J.M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws, J. Comput. Phys. 50:2 (1983) http://dx.doi.org/10.1016/0021-9991(83)90066-9, URL https://www.osti.gov/biblio/5730902.

[15] J. Murillo, P. García-Navarro, Augmented versions of the HLL and HLLC Riemann solvers including source terms in one and two dimensions for shallow flow applications, J. Comput. Phys. (ISSN: 0021-9991) 231 (20) (2012) 6861–6906, http://dx.doi.org/10.1016/j.jcp.2012.06.031, URL https://www.sciencedirect.com/science/article/pii/S0021999112003464.

[16] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. (ISSN: 1631-073X) 339 (9) (2004) 667–672, http://dx.doi.org/10.1016/j.crma.2004.08.006, URL https://www.sciencedirect.com/science/article/pii/S1631073X04004248.

[17] S. Chaturantabut, D.C. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, 2009, pp. 4316–4321, http://dx.doi.org/10.1109/CDC.2009.5400045.

[18] S. Ahmed, O. San, D. Bistrian, I.M. Navon, Sampling and resolution characteristics in reduced order models of shallow water equations: Intrusive vs nonintrusive, Internat. J. Numer. Methods Fluids 92 (8) (2020) 992–1036, http://dx.doi.org/10.1002/fld.4815, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4815.

[19] S. Le Clainche, D. Rodríguez, V. Theofilis, J. Soria, Flow around a hemisphere-cylinder at high angle of attack and low Reynolds number. Part II: POD and DMD applied to reduced domains, Aerosol Sci. Technol. (ISSN: 1270-9638) 44 (2015) 88–100, http://dx.doi.org/10.1016/j.ast.2014.10.009, URL https://www.sciencedirect.com/science/article/pii/S1270963814002077 Instability and Control of Massively Separated Flows.

[20] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28, http://dx.doi.org/10.1017/S0022112010001217.

[21] R. Ştefănescu, A. Sandu, I.M. Navon, Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations, Int. J. Numer. Methods Fluids 76 (8) (2014) 497–521, http://dx.doi.org/10.1002/fld.3946, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3946.

[22] P. Feldmann, R.W. Freund, Efficient linear circuit analysis by padé approximation via the lanczos process, in: Proceedings of the Conference on European Design Automation, in: EURO-DAC '94, IEEE Computer Society Press, Washington, DC, USA, ISBN: 0897916859, 1994, pp. 170–175.

[23] E.J. Yoo, Parametric Model Order Reduction for Structural Analysis and Control (Ph.D. thesis), Technische Universität München, 2010, p. 165, URL https://mediatum.ub.tum.de/997277.

[24] A. Peyvan, V. Oommen, A.D. J., G.E. Karniadakis, RiemannONets: Interpretable neural operators for Riemann problems, Comp. Meth. App. Mech. Eng. (ISSN: 0045-7825) 426 (2024) 116996, http://dx.doi.org/10.1016/j.cma.2024.116996, URL https://www.sciencedirect.com/science/article/pii/S0045782524002524.

[25] D. Rim, R. Baraldi, C.M. Liu, R.J. LeVeque, K. Terada, Tsunami early warning from global navigation satellite system data using convolutional neural networks, Geophys. Res. Lett. 49 (20) (2022) e2022GL099511, http://dx.doi.org/10.1029/2022GL099511, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022GL099511e2022GL099511 2022GL099511.

[26] J.L. Lumley, The structure of inhomogeneous turbulent flows, Atmospheric Turbul. Radio Wave Propagation (1967) 166–176.

[27] L. Sirovich, Turbulence and the dynamics of coherent structures. I - Coherent structures. II - symmetries and transformations. III - dynamics and scaling, Quart. Appl. Math. 45 (1987) 561–571, http://dx.doi.org/10.1090/qam/910463, 573–582, 583–590.

[28] B.G. Galerkin, Rods and plates. Series occurring in various questions concerning the elastic equilibrium of rods and plates, Eng. Bull. 19 (1915) 897–908.

[29] W. IJzerman, Signal Representation and Modeling of Spatial Structures in Fluids (Ph.D. thesis), Universiteit Twente, 2000.

[30] O. San, J. Borggaard, Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows, Int. J. Numer. Methods Fluids 78 (1) (2015) 37–62, http://dx.doi.org/10.1002/fld.4006, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4006.

[31] J.M. Zokagoa, A. Soulaïmani, A POD-based reduced-order model for uncertainty analyses in shallow water flows, Int. J. Comput. Fluid Dyn. 32 (6–7) (2018) 278–292, http://dx.doi.org/10.1080/10618562.2018.1513496, URL https://doi.org/10.1080/10618562.2018.1513496.

[32] S. Ingimarson, L.G. Rebholz, T. Iliescu, Full and reduced order model consistency of the nonlinearity discretization in incompressible flows, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 401 (2022) 115620, http://dx.doi.org/10.1016/j.cma.2022.115620, URL https://www.sciencedirect.com/science/article/pii/S0045782522005758.

[33] B. García-Archilla, J. Novo, S. Rubino, On the influence of the nonlinear term in the numerical approximation of Incompressible Flows by means of proper orthogonal decomposition methods, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 405 (2023) 115866, http://dx.doi.org/10.1016/j.cma.2022.115866, URL https://www.sciencedirect.com/science/article/pii/S0045782522008222.

[34] S. Georgaka, G. Stabile, G. Rozza, M. Bluck, Parametric POD-Galerkin model order reduction for unsteady-state heat transfer problems, Commun. Comput. Phys. 27 (2018) http://dx.doi.org/10.4208/cicp.OA-2018-0207.

[35] M. Gosses, W. Nowak, T. Wöhling, Explicit treatment for Dirichlet, Neumann and Cauchy boundary conditions in POD-based reduction of groundwater models, Adv. Water Resour. (ISSN: 0309-1708) 115 (2018) 160–171, http://dx.doi.org/10.1016/j.advwatres.2018.03.011, URL https://www.sciencedirect.com/science/article/pii/S0309170817307467.

[36] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 196 (4) (2007) 1030–1047, http://dx.doi.org/10.1016/j.cma.2006.08.004, URL http://www.sciencedirect.com/science/article/pii/S0045782506002337.

[37] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: Application to the shallow water equations, J. Comput. Phys. (ISSN: 0021-9991) 229 (11) (2010) 4327–4368, http://dx.doi.org/10.1016/j.jcp.2010.02.016, URL https://www.sciencedirect.com/science/article/pii/S0021999110000896.

[38] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen differenzengleichungen der mathematischen physik, Math. Ann. (ISSN: 0025-5831) 100 (1) (1928) 32–74, http://dx.doi.org/10.1007/BF01448839.

[39] Q. Liang, F. Marche, Numerical resolution of well-balanced shallow water equations with complex source terms, Adv. Water Resour. (ISSN: 0309-1708) 32 (6) (2009) 873–884, http://dx.doi.org/10.1016/j.advwatres.2009.02.010, URL https://www.sciencedirect.com/science/article/pii/S0309170809000396.

[40] M. Morales-Hernández, M. Hubbard, P. García-Navarro, A 2D extension of a Large Time Step explicit scheme (CFL>1) for unsteady problems with wet/dry boundaries, J. Comput. Phys. (ISSN: 0021-9991) 263 (2014) 303–327, http://dx.doi.org/10.1016/j.jcp.2014.01.019, URL https://www.sciencedirect.com/science/article/pii/S0021999114000461.

[41] J. Murillo, A. Navas-Montilla, A comprehensive explanation and exercise of the source terms in hyperbolic systems using roe type solutions. Application to the 1D-2d shallow water equations, Adv. Water Resour. (ISSN: 0309-1708) 98 (2016) 70–96, http://dx.doi.org/10.1016/j.advwatres.2016.10.019, URL https://www.sciencedirect.com/science/article/pii/S0309170816305917.

[42] J. Murillo, A. Navas-Montilla, P. García-Navarro, Formulation of exactly balanced solvers for blood flow in elastic vessels and their application to collapsed states, Comp. Flu. (ISSN: 0045-7930) 186 (2019) 74–98, http://dx.doi.org/10.1016/j.compfluid.2019.04.008, URL https://www.sciencedirect.com/science/article/pii/S0045793019301185.

[43] J. Murillo, P. García-Navarro, Wave Riemann description of friction terms in unsteady shallow flows: Application to water and mud/debris floods, J. Comput. Phys. (ISSN: 0021-9991) 231 (4) (2012) 1963–2001, http://dx.doi.org/10.1016/j.jcp.2011.11.014, URL https://www.sciencedirect.com/science/article/pii/S0021999111006693.

[44] J. Mairal, J. Murillo, P. García-Navarro, The entropy fix in augmented Riemann solvers in presence of source terms: Application to the Shallow Water Equations, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 417 (2023) 116411, http://dx.doi.org/10.1016/j.cma.2023.116411, URL https://www.sciencedirect.com/science/article/pii/S0045782523005352.

[45] G.H. Golub, C.F. Van Loan, Matrix computations, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, ISBN: 9781421407944, 2013.

[46] I. Akhtar, Z. Wang, J. Borggaard, T. Iliescu, A New Closure Strategy for Proper Orthogonal Decomposition Reduced-Order Models, J. Comput. Nonlinear Dyn. (ISSN: 1555-1415) 7 (3) (2012) 034503.

[47] P. Astrid, Reduction of Process Simulation Models: a Proper Orthogonal Decomposition Approach (Ph.D. thesis), Technische Universiteit Eindhoven Electrical Engineering, 2004, http://dx.doi.org/10.6100/IR581728.

[48] D. Ha, P. Tkalich, E.S. Chan, Tsunami forecasting using proper orthogonal decomposition method, J. Geo. Res.: Oceans 113 (C6) (2008) http://dx.doi.org/10.1029/2007JC004583, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007JC004583.

[49] H. Lu, D.M. Tartakovsky, Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena, J. Comput. Phys. (ISSN: 0021-9991) 407 (2020) 109229, http://dx.doi.org/10.1016/j.jcp.2020.109229, URL https://www.sciencedirect.com/science/article/pii/S0021999120300036.

[50] J.M. Zokagoa, A.S. mani, A POD-based reduced-order model for free surface shallow water flows over real bathymetries for Monte-Carlo-type applications, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 221–222 (2012) 1–23, http://dx.doi.org/10.1016/j.cma.2011.11.012, URL https://www.sciencedirect.com/science/article/pii/S0045782511003586.

[51] P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, P. García-Navarro, Development of POD-based Reduced-Order Models applied to shallow water equations using augmented Riemann solvers, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 410 (2023) 116038, http://dx.doi.org/10.1016/j.cma.2023.116038, URL https://www.sciencedirect.com/science/article/pii/S0045782523001627.

[52] P. Solán-Fustero, Reduced-Order Models Based on the Proper Orthogonal Decomposition Applied to Hyperbolic Problems (Ph.D. thesis), University of Zaragoza, 2024, p. 252.

[53] M.T. Capilla, A. Balaguer-Beser, A new well-balanced non-oscillatory central scheme for the shallow water equations on rectangular meshes, J. Comp. App. Math. (ISSN: 0377-0427) 252 (2013) 62–74, http://dx.doi.org/10.1016/j.cam.2013.01.014, URL https://www.sciencedirect.com/science/article/pii/S0377042713000538 Selected papers on Computational and Mathematical Methods in Science and Engineering (CMMSE).

[54] M. Castro, E. Fernández-Nieto, A. Ferreiro, J. García-Rodríguez, C. Madroñal, High Order Extensions of Roe Schemes for Two-Dimensional Nonconservative Hyperbolic Systems, J. Sci. Comp. 39 (2009) 67–114, http://dx.doi.org/10.1007/s10915-008-9250-4.

[55] A. Prieto-Arranz, L. Ramirez, I. Couceiro, I. Colominas, X. Nogueira, A well-balanced SPH-ALE scheme for shallow water applications, J. Sci. Comp. 88 (2021) http://dx.doi.org/10.1007/s10915-021-01600-1.

[56] Z. Wang, J. Zhu, N. Zhao, A new fifth-order finite difference well-balanced multi-resolution WENO scheme for solving shallow water equations, Comput. Appl. Math. (ISSN: 0898-1221) 80 (5) (2020) 1387–1404, http://dx.doi.org/10.1016/j.camwa.2020.07.003, URL https://www.sciencedirect.com/science/article/pii/S0898122120302698.

[57] S. Neelz, G. Pender, Benchmarking of 2D Hydraulic Modelling Packages, SC080035/SR2 Environment Agency, 2010.

[58] P. Liu, H. Yeh, C. Synolakis, Advanced Numerical Models for Simulating Tsunami Waves and Runup, WORLD SCIENTIFIC, 2008, http://dx.doi.org/10.1142/6226, URL https://www.worldscientific.com/doi/abs/10.1142/6226.

[59] J. Burguete, P. García-Navarro, J. Murillo, Preserving bounded and conservative solutions of transport in one-dimensional shallow-water flow with upwind numerical schemes: Application to fertigation and solute transport in rivers, Internat. J. Numer. Methods Fluids 56 (9) (2008) 1731–1764.

[60] J. Fernández-Pato, M. Morales-Hernández, P. García-Navarro, Implicit finite volume simulation of 2D shallow water flows in flexible meshes, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 328 (2018) 1–25, http://dx.doi.org/10.1016/j.cma.2017.08.050, URL https://www.sciencedirect.com/science/article/pii/S0045782517302724.

[61] M. Morales-Hernández, A. Lacasta, J. Murillo, P. García-Navarro, A Large Time Step explicit scheme (CFL>1) on unstructured grids for 2D conservation laws: Application to the homogeneous shallow water equations, Appl. Math. Model. (ISSN: 0307-904X) 47 (2017) 294–317, http://dx.doi.org/10.1016/j.apm.2017.02.043, URL https://www.sciencedirect.com/science/article/pii/S0307904X17301397.