



Well-balanced POD-based reduced-order models for finite volume approximation of hyperbolic balance laws

I. Gómez-Bueno ^a ^{*}, E.D. Fernández-Nieto ^b, S. Rubino ^c

^a Dpto. Matemática Aplicada, Universidad de Málaga, 29010, Málaga, Spain

^b Dpto. Matemática Aplicada I & IMUS, Universidad de Sevilla, 41012, Sevilla, Spain

^c Dpto. EDAN & IMUS, Universidad de Sevilla, 41012, Sevilla, Spain

ARTICLE INFO

MSC:

76M12

35L60

65M08

Keywords:

Finite volume method

Proper orthogonal decomposition

Reduced order modeling

Hyperbolic balance laws

Well-balanced property

ABSTRACT

This paper introduces a reduced-order modeling approach based on finite volume methods for hyperbolic systems, combining Proper Orthogonal Decomposition (POD) with the Discrete Empirical Interpolation Method (DEIM) and Proper Interval Decomposition (PID). Applied to systems such as the transport equation with source term, non-homogeneous Burgers equation, and shallow water equations with non-flat bathymetry and Manning friction, this method achieves significant improvements in computational efficiency and accuracy compared to previous time-averaging techniques. A theoretical result justifying the use of well-balanced Full-Order Models (FOMs) is presented. Numerical experiments validate the approach, demonstrating its accuracy and efficiency. Furthermore, the question of prediction of solutions for systems that depend on some physical parameters is also addressed, and a sensitivity analysis on POD parameters confirms the model's robustness and efficiency in this case.

1. Introduction

The numerical resolution of hyperbolic equations for realistic applications could require high computational costs, due to the use of very fine grids to properly capture complex shock phenomena or discontinuities, and also integration over long periods in time. Reduced-Order Models (ROMs) applied to numerical design of complex physics problems are a proper mathematical tool that is wide-spreading in the scientific community in the recent years in order to save computational time without so much loss of accuracy with respect to classical methods such as Finite Difference (FD), Finite Element (FE) or Finite Volume (FV) methods [1].

Among the most popular ROMs approaches, Proper Orthogonal Decomposition (POD) strategy provides optimal (from the energetic point of view) basis or modes to represent the dynamics from a given database (snapshots) obtained by a Full-Order Model (FOM). Onto these reduced bases, a Galerkin projection of the governing equations can be employed to obtain a low-order dynamical system for the basis coefficients. The resulting low-order model is named standard POD-(Galerkin) ROM, which thus consists in the projection of high-fidelity (full-order) representations of physical problems onto low-dimensional spaces of solutions, with a dramatically reduced dimension. These low-dimensional spaces are capable of capturing the dominant characteristics of the solution, their main advantage being that the computations in the low-dimensional space can be done at a reduced computational cost [2,3]. This has led researchers to apply POD-ROMs to a variety of physical and engineering problems, including Computational Fluid Dynamics (CFD) problems in order to model advection-diffusion equations, see e.g. [4,5], the Navier–Stokes Equations (NSE), see e.g. [6–11], and also hyperbolic equations, such as Shallow Water (SW) equations [12–16].

* Corresponding author.

E-mail address: igomezbueno@uma.es (I. Gómez-Bueno).

<https://doi.org/10.1016/j.cam.2025.116735>

Received 16 December 2024; Received in revised form 7 April 2025

Available online 4 May 2025

0377-0427/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Although POD-ROMs can be very computationally efficient and relatively accurate in some flow configurations, they also present several drawbacks, when applied for instance to strongly nonlinear systems. In this direction, several techniques have been presented for hyperbolic systems, such as the dynamic mode decomposition [12,17], the Discrete Empirical Interpolation Method (DEIM) [18–20], the tensorial POD [21], and also non-intrusive hyper-reduction techniques [12]. Furthermore, to address this issue, a recent paper [22] uses the time averaging approach in combination with the Proper Interval Decomposition (PID, [16,23,24]) applied to SW using augmented Riemann solvers. Moreover, it should be emphasized that, in the context of hyperbolic problems, reduced order modeling techniques suffer from fundamental limitations when used to predict solutions beyond the training time window. This issue arises due to the highly nonlinear and propagative nature of the solutions to systems of balance laws, which may develop discontinuities or rapidly evolving features whose dynamics are not adequately represented by the dominant modes extracted from limited-time snapshots. Consequently, the reliability of reduced models in extrapolative regimes may be affected, and care should be taken when applying them outside the training interval.

In this work, we focus on the use of POD-based ROMs for generic hyperbolic systems with source terms. In particular, we show how the use of DEIM technique to deal with nonlinearities combined with PID outperforms the time averaging approach in combination with PID proposed in [22], especially in terms of physical accuracy.

These PDE systems have been extensively used to model waves caused by small disturbances in steady solutions. A notable example is tsunami waves in the ocean. A numerical method that can solve all steady states of the PDE exactly, or with improved accuracy, or at least a significant subset of them, is referred to as *exactly Well-Balanced* (EWB). Additionally, the method is classified as *fully exactly well-balanced* if it exactly preserves all the stationary solutions of the hyperbolic system (see [25]). The design of well-balanced schemes is a highly active area of research, and many studies focus on this topic, specially for the case of finite volume methods. For example, see [26] for the Burgers equation; [27–32] for the shallow water equations; and [33–37] for other hyperbolic systems, among others.

In this paper we present an original theoretical result of consistency at ROM level, to justify the choice of well-balanced methods to build the FOM, which is independent of the considered hyperbolic balance law. Three examples of well-balanced POD-based methods are presented, for the transport equation with a source term, non-homogeneous Burgers equation and for the SW equations with topography and Manning friction. Finally, following other works focused on reduced order modeling for parameterized problems (see [19]), the question of prediction of solutions for physical parameter-dependent systems is also addressed in this paper, as well a sensitivity analysis to specific parameters of the proposed POD-ROM, such as the number of POD modes and time windows in terms of the errors between FOM and ROM solutions and the CPU times they require.

The rest of the paper is organized as follows: first, we introduce in Section 2 the finite volume full-order models for general one-dimensional hyperbolic PDE systems, detailing the numerical schemes and discussing the construction of exactly well-balanced schemes. In Section 2.1, we specify the FOMs for three examples of balance laws: the linear transport equation with source term, the Burgers equation with a nonlinear source term, and the shallow water equations with and without Manning friction. In all cases, we use PVM-0 type numerical fluxes [38], and for the shallow water case, we additionally apply the HLL scheme [39].

Section 3 outlines the strategy for deriving the POD-based ROMs for the FOMs presented in this article. Special attention is given to the treatment of nonlinearities, where we employ the PID method to adapt the basis functions over time by partitioning the total time interval into subintervals, known as time windows. This method is combined with DEIM, achieving improved results compared to other approaches, such as time-averaging, especially in handling discontinuities. In Section 4, we illustrate the above procedure for the three balance law examples introduced in Section 2.1. Section 5 introduces a theorem proving that if the initial FOM is exactly well-balanced, the associated ROM will also satisfy this property. In Section 6, we briefly address the problem of solution prediction for systems depending on physical parameters, such as the Manning friction coefficient in the shallow water system. Section 7 presents several numerical tests to support the theoretical findings of the study. Finally, concluding remarks are provided in Section 8.

2. Hyperbolic PDE systems: full-order models

Let us consider, for ease of presentation, a general one-dimensional hyperbolic system of balance laws of the form

$$W_t + F(W)_x = S(W)H_x + R(W), \quad x \in I, t > 0, \tag{1}$$

where $I \subset \mathbb{R}$; the unknown $W(x; t) = (w_1(x, t), \dots, w_N(x, t))^T$ takes values in Ω , being Ω an open convex set of \mathbb{R}^N , called set of states; F is a regular function from Ω to \mathbb{R}^N called flux function; S and R are functions from Ω to \mathbb{R}^N and $H(x)$ is a known bounded function from I to \mathbb{R} .

The technical extension to higher dimension (2D/3D) could be performed by following the guidelines given in this work.

First-order FV numerical methods will be used to discretize system (1). The domain I is divided into N_x computational cells $I_i = [x_{i-1/2}, x_{i+1/2}]$. For simplicity, a uniform spatial step size Δx is assumed. The center of each cell I_i is denoted by x_i , where $x_i = (i - 1/2)\Delta x$, and the inter-cell boundary is located at $x_{i+1/2} = i\Delta x$. Similarly, Δt denotes the time step size, where $t^n = (n - 1)\Delta t$ for $n = 1, 2, \dots$. The cell-averaged piecewise approximation of the solution $W(x, t)$ in cell I_i at time t^n is denoted by W_i^n :

$$W_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} W(x, t^n) dx, \tag{2}$$

and its j th component is denoted by $w_{j,i}^n, j = 1, \dots, N$. The mid-point rule is considered to approximate the integrals. A first-order FV full-order model can be written in terms of a general operator \mathcal{L} as follows:

$$W_i^{n+1} = W_i^n - \frac{\Delta t}{\Delta x} \mathcal{L} (W_{i-1}^n, W_i^n, W_{i+1}^n). \tag{3}$$

Moreover, the systems of balance laws of the form (1) admit non-trivial stationary solutions satisfying

$$F(W)_x = S(W)H_x + R(W). \tag{4}$$

In this paper we consider numerical methods that are able to preserve such stationary solutions in the following sense:

Definition 1. A numerical method is said to be Exactly Well-Balanced (EWB) for a stationary solution W^* of (1) if the sequence of its cell-averages $\{W_i^*\}$ (or the sequence of their approximations if a quadrature formula is used to compute them) is an equilibrium of the method.

The development of well-balanced methods has been extensively investigated, with various strategies proposed in the literature. For example, within the path-conservative framework, a family of paths can be selected to satisfy the generalized Rankine–Hugoniot condition, thereby ensuring the well-balanced property for all stationary solutions—or at least for a significant subset of them (see [40]). For instance, in the context of the shallow water equations, when the geometric source term is approximated using a family of straight segments, the resulting numerical method is exactly well-balanced for stationary solutions representing water at rest. This corresponds to cases where the velocity is zero, and the free surface elevation remains constant.

Another strategy for constructing well-balanced schemes involves employing well-balanced reconstruction operators, as described in [25,35]. For example, a first-order, exactly well-balanced reconstruction operator can be defined as:

$$P_i^n(x) = W_i^{*,n}(x) + W_i^n - W_i^{*,n}(x_i), \tag{5}$$

where $W_i^{*,n}(x)$ represents the stationary solution of (1) such that $W_i^{*,n}(x_i) = W_i^n$, since the mid-point rule is used to approximate the integrals.

With regard to the numerical fluxes, we consider $\mathbb{F}(W_L, W_R)$, where \mathbb{F} is a numerical flux consistent with F , i.e., $\mathbb{F}(W, W) = F(W)$. In this work we consider numerical fluxes written under the following viscous form:

$$\mathbb{F}(W_L, W_R) = \frac{1}{2} (F(W_L) + F(W_R)) - \frac{1}{2} D(W_L, W_R) (W_R - W_L), \tag{6}$$

where $D(W_L, W_R)$ is a viscosity matrix, defined in terms of Roe matrix $\mathcal{A}(W_L, W_R)$, verifying

$$\mathcal{A}(W_L, W_R)(W_R - W_L) = F(W_R) - F(W_L).$$

Following [38], Polynomial Viscosity Matrix (PVM) methods have been considered, where $D(W_L, W_R)$ is defined by evaluating a polynomial of degree $l, p^l(W_L, W_R)(x)$, in a linearized Roe matrix $\mathcal{A}(W_L, W_R)$:

$$D(W_L, W_R) = p^l(W_L, W_R)(\mathcal{A}(W_L, W_R)), \tag{7}$$

with

$$p^l(W_L, W_R)(x) = \sum_{k=0}^l \alpha^k(W_L, W_R)x^k.$$

For simplicity, given the sequence of cell-averages $\{W_i\}$, we will use the notation $p^l(W_i, W_{i+1}) = p_{i+1/2}^l, \alpha^k(W_i, W_{i+1}) = \alpha_{i+1/2}^k$ and $\lambda_j(W_i, W_{i+1}) = \lambda_{j,i+1/2}$. In particular, let $\lambda_{1,i+1/2} < \dots < \lambda_{N,i+1/2}$ be the eigenvalues of $\mathcal{A}(W_i, W_{i+1})$. For stability demands one needs

$$|\lambda_{j,i+1/2}| \leq p_{i+1/2}^l(\lambda_{j,i+1/2}) \leq \frac{\Delta x}{\Delta t}, \quad \forall j = 1, \dots, N, \tag{8}$$

with the CFL condition

$$\frac{\Delta t}{\Delta x} \max_j |\lambda_j(W_i, W_{i+1})| \leq \gamma,$$

being $\gamma \leq 1$.

The simplest choice for a PVM method corresponds to the constant polynomial (PVM-0 method):

$$p_{i+1/2}^0(x) = \alpha_{i+1/2}^0,$$

For a PVM-0 flux, the stability condition (8) reads as

$$\max_j |\lambda_{j,i+1/2}| \leq \alpha_{i+1/2}^0 \leq \frac{\Delta x}{\Delta t}. \tag{9}$$

The Lax–Friedrichs flux corresponds $\alpha_{i+1/2}^0 = \frac{\Delta x}{\Delta t}$, the modified Lax–Friedrichs represents the choice $\alpha_{i+1/2}^0 = \gamma \frac{\Delta x}{\Delta t}$ and the Rusanov numerical flux corresponds to $\alpha_{i+1/2}^0 = \max_j |\lambda_{j,i+1/2}|$.

We also consider the HLL numerical flux introduced in [39], which corresponds to the first degree polynomial

$$p_{i+1/2}^1(x) = \alpha_{i+1/2}^0 + \alpha_{i+1/2}^1 x,$$

where the coefficients $\alpha_{i+1/2}^0$ and $\alpha_{i+1/2}^1$ are defined as

$$\alpha_{i+1/2}^0 = \frac{S_{i+1/2}^R |S_{i+1/2}^L| - S_{i+1/2}^L |S_{i+1/2}^R|}{S_{i+1/2}^R - S_{i+1/2}^L}, \quad \alpha_{i+1/2}^1 = \frac{|S_{i+1/2}^R| - |S_{i+1/2}^L|}{S_{i+1/2}^R - S_{i+1/2}^L}. \tag{10}$$

Here, $S_{i+1/2}^L$ and $S_{i+1/2}^R$ are, respectively, approximations of the minimum and maximum wave speed propagation.

2.1. Full-order models: examples

Although this work introduces a strategy for developing ROMs for general 1D systems of balance laws, it is important to remember that the derivation of the ROM depends on the chosen numerical flux and the explicit form of the scheme is utilized to construct the reduced model. Consequently, we consider three different 1D balance laws to evaluate the proposed strategy: the linear transport equation with source term, the Burgers equation with nonlinear source term and SW system with topography and Manning friction.

The numerical schemes used for each system will be explicitly detailed in this subsection in order to subsequently derive the corresponding ROMs.

2.1.1. Transport equation with a linear source term

Let us consider the 1D transport equation with a linear source term

$$w_t + cw_x = \beta w, \quad c, \beta \in \mathbb{R}, \tag{11}$$

which corresponds to the choices

$$W = w, \quad F(w) = cw, \quad S(w) = \beta w, \tag{12}$$

for (1). Notice that, given a cell-average w_i^n , the stationary solution $w_i^{*,n}$ verifying $w_i^{*,n}(x_i) = w_i^n$ is as follows:

$$w_i^{*,n}(x) = w_i^n e^{(\beta/c)(x-x_i)}. \tag{13}$$

Exactly well-balanced reconstruction operators will be applied as in [35] to preserve the stationary solutions. Therefore, if a PVM-0 method is considered, the first-order EWB method reads as follows:

$$\begin{aligned} w_i^{n+1} = & w_i^n - \frac{\Delta t}{2\Delta x} c \left(w_{i+1}^n e^{-\frac{\beta\Delta x}{2c}} + w_i^n \left(e^{\frac{\beta\Delta x}{2c}} - e^{-\frac{\beta\Delta x}{2c}} \right) - w_{i-1}^n e^{\frac{\beta\Delta x}{2c}} \right) \\ & + \frac{\Delta t}{2\Delta x} \left(\alpha_{i+1/2}^{0,n} (w_{i+1}^n e^{-\frac{\beta\Delta x}{2c}} - w_i^n e^{\frac{\beta\Delta x}{2c}}) - \alpha_{i-1/2}^{0,n} (w_i^n e^{-\frac{\beta\Delta x}{2c}} - w_{i-1}^n e^{\frac{\beta\Delta x}{2c}}) \right) \\ & + \frac{\Delta t}{\Delta x} c \left(w_i^n e^{\frac{\beta\Delta x}{2c}} - w_i^n e^{-\frac{\beta\Delta x}{2c}} \right). \end{aligned} \tag{14}$$

With the purpose of simplicity only one PVM-0 method has been considered, namely the modified Lax–Friedrichs method, corresponding to

$$\alpha_{i+1/2}^{0,n} = \gamma \frac{\Delta x}{\Delta t}.$$

2.1.2. Burgers equation with a nonlinear source term

Let us consider the 1D Burgers equation with a nonlinear source term

$$w_t + \left(\frac{w^2}{2} \right)_x = \beta w^2, \quad \beta \in \mathbb{R}, \tag{15}$$

which corresponds to the choices

$$W = w, \quad F(w) = \frac{w^2}{2}, \quad S(w) = \beta w^2, \tag{16}$$

in (1). Notice that it can be regarded as a convective model taking into account a centrifugal force, defined by $\beta = -C \theta_x$, where θ represents the angle of the domain. Then, by considering a constant value of β implies to consider a domain with a constant curvature. Let us remark that $\beta > 0$ implies a negative curvature, then the flow accelerates by the centrifugal acceleration. In this case, given a cell-average w_i^n , the stationary solution verifying $w_i^{*,n}(x_i) = w_i^n$ is:

$$w_i^{*,n}(x) = w_i^n e^{\beta(x-x_i)}. \tag{17}$$

Again, we have considered the approach in [35] to build exactly well-balanced methods, in which the property is transferred to the reconstruction operators. If a PVM-0 method is considered, the following first-order EWB is obtained:

$$\begin{aligned} w_i^{n+1} = & w_i^n - \frac{\Delta t}{4\Delta x} \left((w_{i+1}^n)^2 e^{-\beta\Delta x} + (w_i^n)^2 (e^{\beta\Delta x} - e^{-\beta\Delta x}) - (w_{i-1}^n)^2 e^{\beta\Delta x} \right) \\ & + \frac{\Delta t}{2\Delta x} \left(\alpha_{i+1/2}^{0,n} (w_{i+1}^n e^{-\frac{\beta\Delta x}{2}} - w_i^n e^{\frac{\beta\Delta x}{2}}) - \alpha_{i-1/2}^{0,n} (w_i^n e^{-\frac{\beta\Delta x}{2}} - w_{i-1}^n e^{\frac{\beta\Delta x}{2}}) \right) \\ & + \frac{\Delta t}{2\Delta x} (w_i^n)^2 (e^{\beta\Delta x} - e^{-\beta\Delta x}). \end{aligned} \tag{18}$$

As in previous case, we consider the modified Lax–Friedrichs method.

2.1.3. Shallow water system with non-flat bathymetry and manning friction

Let us consider the 1D SW system with non-flat bathymetry and Manning friction. The one-dimensional shallow water equations (SWE), also known as the Saint-Venant equations, are a system of hyperbolic partial differential equations that describe free-surface flows in rivers and open channels. Originally derived by Barré de Saint-Venant in the 19th century [41], these equations express the fundamental principles of mass conservation and momentum conservation for an incompressible, inviscid fluid under the hydrostatic pressure assumption. The conservative form of the SWE with bottom slope and Manning friction source terms reads:

$$\begin{cases} h_t + q_x = 0, \\ q_t + \left(\frac{q^2}{h} + \frac{1}{2}gh^2 \right)_x = -ghz_x - g \frac{n_b^2 q |q|}{h^{7/3}}, \end{cases} \tag{19}$$

which correspond to the choices $H(x) = -z(x)$ and

$$W = \begin{pmatrix} h \\ q \end{pmatrix}, F(W) = \begin{pmatrix} q \\ \frac{q^2}{h} + \frac{g}{2}h^2 \end{pmatrix}, S(W) = \begin{pmatrix} 0 \\ gh \end{pmatrix}, R(W) = \begin{pmatrix} 0 \\ -g \frac{n_b^2 q |q|}{h^{7/3}} \end{pmatrix}. \tag{20}$$

where the first equation corresponds to mass conservation, and the second equation represents momentum conservation, with source terms for the bed slope and Manning’s friction law. In (19), the unknowns h and q are the water depth of the water layer and discharge, respectively; the function $z(x)$ is the depth function; g is the gravity acceleration and n_b is the Manning friction coefficient. We denote by $u = q/h$ the depth-averaged velocity and $c = \sqrt{gh}$ the celerity.

The eigenvalues of the system are $\lambda^\pm = u \pm c$ and the Froude number defined by

$$Fr(W) = \frac{|u|}{c}, \tag{21}$$

characterizes the flow regime: subcritical ($Fr < 1$), critical ($Fr = 1$) or supercritical ($Fr > 1$).

In this work, we are interested in the preservation of the steady states corresponding to the water at rest, which are of the form

$$u = 0, \quad \eta = constant, \tag{22}$$

where

$$\eta = h + z,$$

denotes the free surface. Following [40], if the path-conservative framework is considered and the family of segments is chosen to approximate the geometrical source term, the method is exactly well-balanced for water at rest. Choosing a PVM-0 method as numerical flux, the first-order EWB method for non-moving steady states reads as follows:

$$h_i^{n+1} = h_i^n - \frac{\Delta t}{2\Delta x} (q_{i+1}^n - q_{i-1}^n) + \frac{\Delta t}{2\Delta x} \left(\alpha_{i+1/2}^{0,n} (\eta_{i+1}^n - \eta_i^n) - \alpha_{i-1/2}^{0,n} (\eta_i^n - \eta_{i-1}^n) \right), \tag{23}$$

$$\begin{aligned} q_i^{n+1} &= q_i^n - \frac{\Delta t}{2\Delta x} \left((u_{i+1}^n)^2 h_{i+1}^n + \frac{1}{2}g(h_{i+1}^n)^2 - (u_{i-1}^n)^2 h_{i-1}^n - \frac{1}{2}g(h_{i-1}^n)^2 \right) \\ &+ \frac{\Delta t}{2\Delta x} \left(\alpha_{i+1/2}^{0,n} (q_{i+1}^n - q_i^n) - \alpha_{i-1/2}^{0,n} (q_i^n - q_{i-1}^n) \right) \\ &- \frac{g\Delta t}{4\Delta x} \left((h_{i+1}^n + h_i^n)(z_{i+1} - z_i) + (h_i^n + h_{i-1}^n)(z_i - z_{i-1}) \right) - \Delta t g \frac{n_b^2 q_i^n |q_i^n|}{(h_i^n)^{7/3}}. \end{aligned} \tag{24}$$

As in previous cases, we consider the case corresponding to the modified Lax–Friedrichs method, defined by

$$\alpha_{i+1/2}^{0,n} = \gamma \frac{\Delta x}{\Delta t}.$$

On the other hand, if the HLL flux is selected, the following first-order EWB method for water-at-rest is considered:

$$h_i^{n+1} = h_i^n - \frac{\Delta t}{2\Delta x} (q_{i+1}^n - q_{i-1}^n) + \frac{\Delta t}{2\Delta x} \alpha_{i+1/2}^{0,n} (\eta_{i+1}^n - \eta_i^n) - \frac{\Delta t}{2\Delta x} \alpha_{i-1/2}^{0,n} (\eta_i^n - \eta_{i-1}^n) + \frac{\Delta t}{2\Delta x} \alpha_{i+1/2}^{1,n} (q_{i+1}^n - q_i^n) - \frac{\Delta t}{2\Delta x} \alpha_{i-1/2}^{1,n} (q_i^n - q_{i-1}^n), \tag{25}$$

$$\begin{aligned} q_i^{n+1} &= q_i^n - \frac{\Delta t}{2\Delta x} \left((u_{i+1}^n)^2 h_{i+1}^n + \frac{1}{2}g(h_{i+1}^n)^2 - (u_{i-1}^n)^2 h_{i-1}^n - \frac{1}{2}g(h_{i-1}^n)^2 \right) + \frac{\Delta t}{2\Delta x} \left(\alpha_{i+1/2}^{1,n} \left(-(\tilde{u}_{i+1/2}^n)^2 + g\tilde{h}_{i+1/2}^n \right) (\eta_{i+1}^n - \eta_i^n) \right) \\ &- \frac{\Delta t}{2\Delta x} \left(\alpha_{i-1/2}^{1,n} \left(-(\tilde{u}_{i-1/2}^n)^2 + g\tilde{h}_{i-1/2}^n \right) (\eta_i^n - \eta_{i-1}^n) \right) + \frac{\Delta t}{2\Delta x} \alpha_{i+1/2}^{0,n} (q_{i+1}^n - q_i^n) - \frac{\Delta t}{2\Delta x} \alpha_{i-1/2}^{0,n} (q_i^n - q_{i-1}^n) \\ &+ \frac{\Delta t}{\Delta x} \alpha_{i-1/2}^{1,n} \tilde{u}_{i+1/2}^n (q_{i+1}^n - q_i^n) - \frac{\Delta t}{\Delta x} \alpha_{i-1/2}^{1,n} \tilde{u}_{i-1/2}^n (q_i^n - q_{i-1}^n) - \frac{g\Delta t}{2\Delta x} \left(\tilde{h}_{i+1/2}^n (z_{i+1} - z_i) + \tilde{h}_{i-1/2}^n (z_i - z_{i-1}) \right) - \Delta t g \frac{n_b^2 q_i^n |q_i^n|}{(h_i^n)^{7/3}}, \end{aligned} \tag{26}$$

where

$$\tilde{h}_{i+1/2}^n = \frac{h_{i+1}^n + h_i^n}{2}, \quad \tilde{u}_{i+1/2}^n = \frac{\sqrt{h_{i+1}^n u_{i+1}^n} + \sqrt{h_i^n u_i^n}}{\sqrt{h_{i+1}^n} + \sqrt{h_i^n}}, \tag{27}$$

(see [42]).

3. POD-based reduced-order models for hyperbolic PDE systems

Let us describe the general strategy to develop a POD-based reduced-order method for a given FOM previously introduced. Given the j th component of W , w_j , we consider the following snapshot matrix M_{w_j} whose columns are the set of N_T numerical solutions at each time step t^n , $n = 1, \dots, N_T$, where $t^{N_T} = T_f$, being T_f the final computational time:

$$M_{w_j} = \begin{pmatrix} w_{j,1}^1 & w_{j,1}^2 & \dots & w_{j,1}^{N_T} \\ w_{j,2}^1 & w_{j,2}^2 & \dots & w_{j,2}^{N_T} \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,N_x}^1 & w_{j,N_x}^2 & \dots & w_{j,N_x}^{N_T} \end{pmatrix}, \tag{28}$$

i.e., $w_{j,i}^n$ is the numerical approximation of w_j at the i th cell at time t^n .

First, we compute the functions of the POD basis by applying the singular value decomposition to the snapshot matrix for each variable w_j to obtain

$$M_{w_j} M_{w_j}^T = \Phi_j \Sigma_j \Sigma_j^T \Phi_j^T, \tag{29}$$

where

$$\Sigma_j = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{N_x \times N_T} \tag{30}$$

is a diagonal matrix whose entries are the singular values of M_{w_j} and the matrix $\Phi_j = (\Phi_{j,1}, \dots, \Phi_{j,N_x}) \in \mathbb{R}^{N_x \times N_x}$ consists of the orthogonal eigenvectors of $M_{w_j} M_{w_j}^T$, with $\Phi_{j,k} = (\Phi_{j,1,k}, \dots, \Phi_{j,N_x,k})^T$. Now, let us apply the Galerkin decomposition to each component of W :

$$w_{j,i}^n \approx w r_{j,i}^n = \sum_{k=1}^M \hat{w}_{j,k}^n \Phi_{j,i,k}, \tag{31}$$

where $M \ll \min\{N_x, N_T\}$ is the number of POD modes. In order to develop the ROM, we first introduce the Galerkin decomposition (31) into de FOM, multiply each equation by the p th component of the corresponding basis functions $\Phi_{j,i,p}$, $p = 1, \dots, M$, and sum up over the cells to complete the projection. Following the previous steps, the system to obtain the coefficients for the ROM reads

$$\hat{w}_{j,p}^{n+1} = \hat{w}_{j,p}^n - \frac{\Delta t}{\Delta x} \left(\sum_{i=1}^{N_x} \mathcal{L}_j (W r_{i-1}^n, W r_i^n, W r_{i+1}^n) \Phi_{j,i,p} \right), \tag{32}$$

where $p = 1, \dots, M$, \mathcal{L}_j is the j th component of the operator \mathcal{L} , and $W r_i^n$ is the vector whose components are $w r_{j,i}^n$.

The number of POD modes M is chosen following [43], which suggests a criterion to select the POD dimension $M \leq r$ as the smaller integer such that

$$I(M) = \frac{\sum_{i=1}^M \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \epsilon_{\text{POD}}^2, \tag{33}$$

where ϵ_{POD} is a desired tolerance.

3.1. Dealing with nonlinearities

Accurate solutions can often be obtained with a limited number of POD modes when dealing with smooth flow dynamics. Nevertheless, for highly nonlinear and irregular flows, traditional POD-based techniques may encounter limitations. This issue stems from the use of time-invariant basis functions, as discussed in [16,44]. Various approaches have been introduced to tackle these issues.

In this work, we propose adopting the PID approach. Rather than employing a single set of basis functions throughout the entire simulation duration, this method divides the time domain into multiple subintervals, within which the standard POD method is applied independently. This approach allows the basis functions to vary over time.

The entire simulation interval $[0, T_f]$ is divided into N_v non-overlapping subintervals, referred to as *time windows*, organized as follows:

$$[0, T_f] = [t^{N_1}, t^{N_2}] \cup [t^{N_3}, t^{N_4}] \cup \dots \cup [t^{2N_v-1}, t^{2N_v}], \tag{34}$$

where $t^{N_1} = t^1 = 0$ and $t^{2N_v} = t^{N_T} = T_f$. Each time window is denoted by

$$T^v = [t^{N_{2v-1}}, t^{N_{2v}}].$$

Thus, each snapshot matrix M_{w_j} is also divided into N_v submatrices $M_{w_j}^v \in \mathbb{R}^{N_x \times N_{sv}}$, $v = 1, \dots, N_v$, to which the POD procedure is performed. Here, N_{sv} is the number of snapshots per time window. As a result, each time window will have its own POD basis

composed of the functions $\Phi_{j,k}^v$, $v = 1, \dots, N_v$. For simplicity, a uniform partition of the time domain into windows is considered in this work.

At the interface of each time window, in order to switch from one time interval to the other without requiring an online computation of the full-order model and therefore reducing the computational cost, we impose continuity in the projection, considering the following jump condition between two times windows T^v and T^{v+1} :

$$\hat{w}_{j,k}^{n,v+1} = \sum_{i=1}^{N_x} \sum_{l=1}^M \hat{w}_{j,i,l}^{n,v} \Phi_{j,i,l}^v \Phi_{j,i,k}^{v+1} \tag{35}$$

This approach allows to reduce the number of POD modes when dealing with nonlinear terms, so that, considering a much smaller number of POD modes and applying the PID method, a similar accuracy to the results obtained with a standard POD method can be obtained.

In addition, sometimes the POD method cannot be directly applied to some systems. Consider, for example, the Manning friction term in the SW equations (19):

$$-g \frac{n_b^2 q |q|}{h^{7/3}} = -g \frac{n_b^2 u |u|}{h^{1/3}}. \tag{36}$$

This is a highly nonlinear term, in which we find quotients and powers of functions, among others. When solving the ROM, a system to update the POD coefficients \hat{w}_j^n for the conserved variables is obtained. However, sometimes other auxiliary variables are also involved, to which the Galerkin decomposition method is applied. This is the case, for instance, of the velocity u in the SW equations. These additional snapshot matrices, built offline, allows to improve the efficiency of the standard POD method. The main problem is the fact that we do not have a direct expression that allows us to update the POD coefficients for these auxiliary variables. In each iteration of the ROM, once the POD coefficients of the conserved variables have been obtained, it would not be efficient to calculate these variables in the physical space in order to be able to carry out the projection in the corresponding POD basis functions and obtain the POD coefficients of the auxiliary variables. To tackle this problem, two different strategies are studied and compared in this paper: the time-averaging approach (see [16,22]) and DEIM (see [1,43]).

Remark 1. Note that for the shallow water system, in addition to considering the modified Lax–Friedrichs flux, the HLL method is also used, which allows a higher accuracy to be obtained. However, considering the HLL flux poses a significant challenge in deriving the ROM due to the presence of numerous nonlinearities. This will allow us to verify that the strategies considered in the article to tackle this problem (time-averaging and DEIM) are adequate and compare them to see which one is more suitable.

On the one hand, the time-averaging approach described in [16,22] is based on the PID method described above. When developing the ROM, instead of considering the reduced variable $w_{r,j}$, the following approximation is considered at each time window:

$$\overline{w_{j,i}^v} = \frac{1}{N_{sv}} \sum_{s=v_1}^{v_{N_{sv}}} w_{j,i}^s, \quad i = 1, \dots, N_x, \tag{37}$$

where $t^{v_1} = t^{N_{2v}-1}$ and $t^{v_{N_{sv}}} = t^{N_{2v}}$ are the initial and final times of the v th time window T^v , $v = 1, \dots, N_v$.

On the other hand, DEIM offers a way to update the POD coefficients for auxiliary variables, as discussed in [14,16,20], among others. The main concept involves substituting the full-space orthogonal projection with a projection onto a reduced-dimensional subspace. This subspace is formed by a basis constructed through a greedy algorithm, which iteratively minimizes the interpolation error across the set of snapshots, evaluated in the maximum norm. As a result, this projection technique requires computing only a few specific components of the nonlinear terms, thereby significantly reducing computational costs. Let us consider, for instance, the velocity $u = q/h$ for the SW equations. This variable can be approximated in the ROM as follows:

$$u_i^n \approx \sum_{k=1}^M \hat{u}_k^n \Phi_{3,i,k}, \tag{38}$$

where $\Phi_{3,i,k}$ is the i th component of the basis function

$$\Phi_{3,k} = \left(\Phi_{3,1,k}, \dots, \Phi_{3,N_x,k} \right)^T, \tag{39}$$

which is the k th column of the matrix Φ_3 in (29). A pseudo-code to update the POD coefficients \hat{u} at each time iteration is given in Algorithms 1 and 2, where the former is devoted to the offline process and the latter to the online stage. In Algorithm 1, U_I and $\Phi_{3,I,m}$ refers to the vectors formed by the elements at the positions of the indices I of U and $\Phi_{3,m}$, respectively.

4. Reduced-order models: some examples

Let us illustrate the above procedure by applying it to the transport equation with a linear source term, the Burgers equation with a nonlinear source term, and the SW system with topography and Manning friction introduced in Section 2.1. Notice that, since the latter two systems have nonlinear terms, models using the PID strategy are considered. For the sake of simplicity, the superindices corresponding to the time windows have been removed.

Algorithm 1: DEIM: offline stage

Input: Snapshot matrix $\Phi_3 \in \mathbb{R}^{N_x \times M}$, number of DEIM points M
Output: Interpolation matrix U and index set I
 $i_1 \leftarrow \arg \max_{i=1, \dots, N_x} |\Phi_{3,i,1}|;$
 $U \leftarrow \Phi_{3,1};$
 $I \leftarrow \{i_1\};$
for $m = 2$ **to** M **do**
 $r \leftarrow \Phi_{3,m} - U (U_I^{-1} \Phi_{3,I,m});$
 $i_m \leftarrow \arg \max_{i=1, \dots, N_x} |r_i|;$
 $U \leftarrow [U \ \Phi_{3,m}];$
 $I \leftarrow I \cup \{i_m\};$
return U, I

Algorithm 2: DEIM: online stage

Input: $U_I \in \mathbb{R}^{M \times M}$, index set $I = \{i_1, \dots, i_M\}$
Output: Approximated vector \hat{u}
 Evaluate the nonlinear function at interpolation points: $\hat{u}_I \leftarrow \frac{q}{h} \Big|_I;$
 Solve the linear system: $U_I \hat{u} = \hat{u}_I;$
return \hat{u}

4.1. Transport equation with a linear source term

The ROM is developed by applying the Galerkin decomposition to w , obtaining the following vector formulation:

$$\hat{w}^{n+1} = \hat{w}^n - \frac{c \Delta t}{2 \Delta x} A \hat{w}^n + \frac{\gamma}{2} B \hat{w}^n + \frac{c \Delta t}{\Delta x} C \hat{w}^n, \tag{40}$$

where the training matrices $A = (A_{pk})$, $B = (B_{pk})$, $C = (C_{pk})$ are given by

$$A_{pk} = a_1 \Phi_{1,p} + \sum_{i=2}^{N_x-1} (\Phi_{i+1,k} e^- + \Phi_{i,k} (e^+ - e^-) - \Phi_{i-1,k} e^+) \Phi_{i,p} + a_{N_x} \Phi_{N_x,p}, \tag{41}$$

$$B_{pk} = b_1 \Phi_{1,p} + \sum_{i=2}^{N_x-1} (\Phi_{i+1,k} e^- - \Phi_{i,k} (e^+ + e^-) + \Phi_{i-1,k} e^+) \Phi_{i,p} + b_{N_x} \Phi_{N_x,p}, \tag{42}$$

$$C_{pk} = \sum_{i=1}^{N_x} (\Phi_{i,k} (e^+ - e^-)) \Phi_{i,p}, \tag{43}$$

with $e^- = e^{-\frac{\beta \Delta x}{2c}}$, $e^+ = e^{\frac{\beta \Delta x}{2c}}$ and $a_1, b_1, a_{N_x}, b_{N_x}$ the boundary terms. For instance, if free boundary conditions are considered, one has:

$$\begin{aligned} a_1 &= \Phi_{2,k} e^- - \Phi_{1,k} e^-, & a_{N_x} &= \Phi_{N_x,k} e^+ - \Phi_{N_x-1,k} e^+, \\ b_1 &= \Phi_{2,k} e^- - \Phi_{1,k} e^-, & b_{N_x} &= -\Phi_{N_x,k} e^+ + \Phi_{N_x-1,k} e^+. \end{aligned}$$

4.2. Burgers equation with a nonlinear source term

The ROM is developed by applying the Galerkin decomposition to w , obtaining the following vector formulation:

$$\hat{w}^{n+1} = \hat{w}^n - \frac{\Delta t}{4 \Delta x} (\hat{w}^n)^T A \hat{w}^n + \frac{\gamma}{2} B \hat{w}^n + \frac{\Delta t}{2 \Delta x} (\hat{w}^n)^T C \hat{w}^n, \tag{44}$$

where the training 3D linear tensors $A = (A_{plk})$ and $C = (C_{plk})$ and the matrix $B = (B_{pk})$ are given by

$$A_{plk} = a_1 \Phi_{1,p} + \sum_{i=2}^{N_x-1} (\Phi_{i+1,k} \Phi_{i+1,l} E^- + \Phi_{i,k} \Phi_{i,l} (E^+ - E^-) - \Phi_{i-1,k} \Phi_{i-1,l} E^+) \Phi_{i,p}, \tag{45}$$

$$B_{pk} = b_1 \Phi_{1,p} + \sum_{i=2}^{N_x-1} (\Phi_{i+1,k} e^- - \Phi_{i,k} (e^+ + e^-) + \Phi_{i-1,k} e^+) \Phi_{i,p} + b_{N_x} \Phi_{N_x,p}, \tag{46}$$

$$C_{plk} = \sum_{i=1}^{N_x} (\Phi_{i,k} \Phi_{i,l} (E^+ - E^-)) \Phi_{i,p}, \tag{47}$$

with $E^- = e^{-\beta \Delta x}$, $E^+ = e^{\beta \Delta x}$, $e^- = e^{-\frac{\beta \Delta x}{2}}$, $e^+ = e^{\frac{\beta \Delta x}{2}}$ and $a_1, b_1, a_{N_x}, b_{N_x}$ the boundary terms.

4.3. Shallow water system with non-flat bathymetry manning friction

Let us consider the SW with bottom and Manning friction (19). Using the general notation in (1), $h = w_1$ and $q = w_2$. Moreover, the POD method will be also applied to the velocity u , applying DEIM to update it when solving the system at each iteration. Each variable will be approximated as follows:

$$h_i^n \approx \sum_{k=1}^M \hat{h}_k^n \Phi_{1,i,k}, \quad q_i^n \approx \sum_{k=1}^M \hat{q}_k^n \Phi_{2,i,k}, \quad u_i^n \approx \sum_{k=1}^M \hat{u}_k^n \Phi_{3,i,k}. \quad (48)$$

As described in Section 2.1, the modified Lax–Friedrichs and HLL numerical fluxes have been considered, leading to two different ROMs.

4.3.1. Modified lax–friedrichs ROM

If the modified Lax–Friedrichs numerical flux (23)–(24) is considered, the vector formulation of the ROM model reads as follows:

$$\begin{aligned} \hat{h}^{n+1} &= \hat{h}^n - \frac{\Delta t}{2\Delta x} A \hat{q}^n + \frac{\gamma}{2} B \hat{h}^n + \frac{\gamma}{2} C, \\ \hat{q}^{n+1} &= \hat{q}^n - \frac{\Delta t}{2\Delta x} (\hat{u}^n)^T D \hat{q}^n - g \frac{\Delta t}{4\Delta x} (\hat{h}^n)^T E \hat{h}^n + \frac{\gamma}{2} F \hat{q}^n - g \frac{\Delta t}{4\Delta x} G \hat{h}^n - \Delta t Fric, \end{aligned} \quad (49)$$

where the training matrices $A = (A_{pk})$, $B = (B_{pk})$, $F = (F_{pk})$ and $G = (G_{pk})$, the vector $C = (C_p)$, and the 3D linear tensors $D = (D_{plk})$ and $E = (E_{plk})$ are given by

$$A_{pk} = a_1 \Phi_{1,1,p} + \sum_{i=2}^{N_x-1} (\Phi_{2,i+1,k} - \Phi_{2,i-1,k}) \Phi_{1,i,p} + a_{N_x} \Phi_{1,N_x,p}, \quad (50)$$

$$B_{pk} = b_1 \Phi_{1,1,p} + \sum_{i=2}^{N_x-1} (\Phi_{1,i+1,k} - 2\Phi_{1,i,k} + \Phi_{1,i-1,k}) \Phi_{1,i,p} + b_{N_x} \Phi_{1,N_x,p}, \quad (51)$$

$$C_p = c_1 \Phi_{1,1,p} + \sum_{i=2}^{N_x-1} (z_{i+1} - 2z_i + z_{i-1}) \Phi_{1,i,p} + c_{N_x} \Phi_{1,N_x,p}, \quad (52)$$

$$D_{plk} = d_1 \Phi_{2,1,p} + \sum_{i=2}^{N_x-1} (\Phi_{2,i+1,k} \Phi_{3,i+1,l} - \Phi_{2,i-1,k} \Phi_{3,i-1,l}) \Phi_{2,i,p} + d_{N_x} \Phi_{2,N_x,p}, \quad (53)$$

$$E_{plk} = e_1 \Phi_{2,1,p} + \sum_{i=2}^{N_x-1} (\Phi_{1,i+1,k} \Phi_{1,i+1,l} - \Phi_{1,i-1,k} \Phi_{1,i-1,l}) \Phi_{2,i,p} + e_{N_x} \Phi_{2,N_x,p}, \quad (54)$$

$$F_{pk} = f_1 \Phi_{2,1,p} + \sum_{i=2}^{N_x-1} (\Phi_{2,i+1,k} - 2\Phi_{2,i,k} + \Phi_{2,i-1,k}) \Phi_{2,i,p} + f_{N_x} \Phi_{2,N_x,p}, \quad (55)$$

$$G_{pk} = g_1 \Phi_{2,1,p} + g_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} ((\Phi_{1,i+1,k} + \Phi_{1,i,k})(z_{i+1} + z_i) + (\Phi_{1,i,k} + \Phi_{1,i-1,k})(z_i + z_{i-1})) \Phi_{2,i,p}, \quad (56)$$

with $a_1, b_1, c_1, d_1, e_1, f_1, g_1, a_{N_x}, b_{N_x}, c_{N_x}, d_{N_x}, e_{N_x}, f_{N_x}, g_{N_x}$ the boundary terms.

In order to build the friction term $Fric$ in (49), the two strategies described in the previous section (time windows and DEIM) could be considered. This leads to different expressions of the friction term $Fric$ in (49):

- If time-averaging is applied to all the variables, one has:

$$Fric = gn_b^2 H, \quad (57)$$

where $H = (H_p)$ reads as follows:

$$H_p = \sum_{i=1}^{N_x} \frac{\overline{|u_i^n|} \overline{|u_i^n|}}{(\overline{h_i^n})^{1/3}} \Phi_{2,i,p}. \quad (58)$$

Here, $\overline{|u_i^n|}$ and $\overline{h_i^n}$ are the time-averaging approximations introduced in (37) applied to u_i^n and h_i^n , respectively.

- If time-averaging is only applied to the term

$$\frac{|u|}{h^{4/3}},$$

one has

$$Fric = gn_b^2 H \hat{q}^n, \quad (59)$$

where $H = (H_{pk})$ reads as follows:

$$H_{pk} = \sum_{i=1}^{N_x} \frac{|\overline{u_i^n}|}{(\overline{h_i^n})^{4/3}} (\Phi_{2,i,p})^2. \tag{60}$$

• If the auxiliary variable

$$f = \frac{|q|}{h^{7/3}} \tag{61}$$

is considered, and the Galerkin decomposition method is applied to it

$$f_i^n \approx \sum_{k=1}^M \hat{f}_k^n \Phi_{4,i,k}, \tag{62}$$

using DEIM to update the coefficients \hat{f}_k^n , then one has

$$Fric = gn_b^2 (\hat{q}^n)^T H \hat{f}^n, \tag{63}$$

where $H = (H_{plk})$ reads as follows:

$$H_{plk} = \sum_{i=1}^{N_x} \Phi_{4,i,k} \Phi_{2,i,l} \Phi_{2,i,p}. \tag{64}$$

4.3.2. HLL ROM

Again, the time-averaging approach and DEIM will be applied if the HLL numerical flux (25)–(26) is considered. Here, $\overline{\hat{u}_{i+1/2}^n}$ and $\overline{\hat{h}_{i+1/2}^n}$ denote the time-averaging approximations of $\hat{u}_{i+1/2}^n$ and $\hat{h}_{i+1/2}^n$, respectively (see (27)), introduced in (37). The vector formulation reads as follows:

$$\begin{aligned} \hat{h}^{n+1} &= \hat{h}^n - \frac{\Delta t}{2\Delta x} A \hat{q}^n + \frac{\Delta t}{2\Delta x} M_1 + \frac{\Delta t}{2\Delta x} M_2 + \frac{\Delta t}{2\Delta x} M_3, \\ \hat{q}^{n+1} &= \hat{q}^n - \frac{\Delta t}{2\Delta x} (\hat{u}^n)^T D \hat{q}^n - g \frac{\Delta t}{4\Delta x} (\hat{h}^n)^T E \hat{h}^n + \frac{\Delta t}{2\Delta x} M_4 + \frac{\Delta t}{2\Delta x} M_5 + \frac{\Delta t}{2\Delta x} M_6 + \frac{\Delta t}{2\Delta x} M_7 - g \frac{\Delta t}{4\Delta x} G \hat{h}^n - \Delta t Fric, \end{aligned} \tag{65}$$

where the training matrices A and G and the training tensors D and E are the same as in (56), and the three previous possibilities (57)–(59)–(63) for the friction term $Fric$ can be considered. Two different approaches are considered to define the terms M_m , $m = 1, \dots, 6$, coming from the diffusion part of the HLL numerical flux. On the one hand, if the time-averaging technique is applied to approximate the values $\alpha_{i+1/2}^{s,n}$, $s = 1, 2$, denoting these approximations by $\overline{\alpha_{i+1/2}^{s,n}}$, $s = 1, 2$, one has:

$$M_1 = U_1 \hat{h}^n, \quad M_2 = U_2 \hat{q}^n, \quad M_3 = U_3, \quad M_4 = U_4 \hat{h}^n, \quad M_5 = U_5 \hat{q}^n, \quad M_6 = U_6 \hat{q}^n, \quad M_7 = U_7, \tag{66}$$

where the training matrices $U_1 = ((U_1)_{pk})$, $U_2 = ((U_2)_{pk})$, $U_4 = ((U_4)_{pk})$, $U_5 = ((U_5)_{pk})$, $U_6 = ((U_6)_{pk})$ and $U_7 = ((U_7)_p)$, and the training vector $U_3 = ((U_3)_p)$ are given by

$$(U_1)_{pk} = (u_1)_1 \Phi_{1,1,p} + \dots + (u_1)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{0,n}} (\Phi_{1,i+1,k} - \Phi_{1,i,k}) - \overline{\alpha_{i-1/2}^{0,n}} (\Phi_{1,i,k} - \Phi_{1,i-1,k}) \right) \Phi_{1,i,p}, \tag{67}$$

$$(U_2)_{pk} = (u_2)_1 \Phi_{1,1,p} + (u_2)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{1,n}} (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \overline{\alpha_{i-1/2}^{1,n}} (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{1,i,p}, \tag{68}$$

$$(U_3)_p = (u_3)_1 \Phi_{1,1,p} + (u_3)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{0,n}} (z_{i+1} - z_i) - \overline{\alpha_{i-1/2}^{0,n}} (z_i - z_{i-1}) \right) \Phi_{1,i,p}, \tag{69}$$

$$\begin{aligned} (U_4)_{pk} &= (u_4)_1 \Phi_{2,1,p} + (u_4)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{1,n}} \left(-\overline{(\hat{u}_{i+1/2}^n)^2} + g \overline{\hat{h}_{i+1/2}^n} \right) (\Phi_{1,i+1,k} - \Phi_{1,i,k}) \right) \Phi_{2,i,p} \\ &\quad - \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i-1/2}^{1,n}} \left(-\overline{(\hat{u}_{i-1/2}^n)^2} + g \overline{\hat{h}_{i-1/2}^n} \right) (\Phi_{1,i,k} - \Phi_{1,i-1,k}) \right) \Phi_{2,i,p}, \end{aligned} \tag{70}$$

$$(U_5)_{pk} = (u_5)_1 \Phi_{2,1,p} + (u_5)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{0,n}} (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \overline{\alpha_{i-1/2}^{0,n}} (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{2,i,p}, \tag{71}$$

$$(U_6)_{pk} = (u_6)_1 \Phi_{2,1,p} + (u_6)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} 2 \left(\overline{\alpha_{i+1/2}^{1,n} \overline{\hat{u}_{i+1/2}^n}} (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \overline{\alpha_{i-1/2}^{1,n} \overline{\hat{u}_{i-1/2}^n}} (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{2,i,p}, \tag{72}$$

$$\begin{aligned} (U_7)_p &= (u_7)_1 \Phi_{2,1,p} + (u_7)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i+1/2}^{1,n}} \left(-\overline{(\hat{u}_{i+1/2}^n)^2} + g \overline{\hat{h}_{i+1/2}^n} \right) (z_{i+1} - z_i) \right) \Phi_{2,i,p} \\ &\quad - \sum_{i=2}^{N_x-1} \left(\overline{\alpha_{i-1/2}^{1,n}} \left(-\overline{(\hat{u}_{i-1/2}^n)^2} + g \overline{\hat{h}_{i-1/2}^n} \right) (z_i - z_{i-1}) \right) \Phi_{2,i,p}, \end{aligned} \tag{73}$$

and $(u_m)_r$, $m = 1, \dots, 7$, $r = 1, N_x$, are the boundary terms. This choice will be denoted in the numerical experiments as HLL - Tav.

On the other hand, if the Galerkin decomposition method is applied to $\hat{\alpha}_{i+1/2}^{s,n}$, $s = 1, 2$

$$\alpha_{i+1/2}^{s,n} \approx \sum_{k=1}^M \hat{\alpha}_k^{s,n} \mathcal{A}_{i+1/2,k}^s, \quad s = 1, 2, \tag{74}$$

and DEIM is used to update the coefficients $\hat{\alpha}_k^{s,n}$, $s = 1, 2$, one has:

$$\begin{aligned} M_1 &= (\hat{\alpha}^{0,n})^T U_1 \hat{h}^n, & M_2 &= (\hat{\alpha}^{1,n})^T U_2 \hat{q}^n, & M_3 &= U_3 \hat{\alpha}^{0,n}, \\ M_4 &= (\hat{\alpha}^{1,n})^T U_4 \hat{h}^n, & M_5 &= (\hat{\alpha}^{0,n})^T U_5 \hat{q}^n, & M_6 &= (\hat{\alpha}^{1,n})^T U_6 \hat{q}^n, & M_7 &= U_7 \hat{\alpha}^{1,n}, \end{aligned} \tag{75}$$

where the training 3D linear tensors $U_1 = ((U_1)_{plk})$, $U_2 = ((U_2)_{plk})$, $U_4 = ((U_4)_{plk})$, $U_5 = ((U_5)_{plk})$ and $U_6 = ((U_6)_{plk})$, and the training matrices $U_3 = ((U_3)_{pk})$ and $U_7 = ((U_7)_{pk})$ are given by

$$(U_1)_{plk} = (u_1)_1 \Phi_{1,1,p} + (u_1)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,l}^0 (\Phi_{1,i+1,k} - \Phi_{1,i,k}) - \mathcal{A}_{i-1/2,l}^0 (\Phi_{1,i,k} - \Phi_{1,i-1,k}) \right) \Phi_{1,i,p}, \tag{76}$$

$$(U_2)_{plk} = (u_2)_1 \Phi_{1,1,p} + (u_2)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,l}^1 (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \mathcal{A}_{i-1/2,l}^1 (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{1,i,p}, \tag{77}$$

$$(U_3)_{pk} = (u_3)_1 \Phi_{1,1,p} + (u_3)_{N_x} \Phi_{1,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,k}^0 (z_{i+1} - z_i) - \mathcal{A}_{i-1/2,k}^0 (z_i - z_{i-1}) \right) \Phi_{1,i,p}, \tag{78}$$

$$\begin{aligned} (U_4)_{plk} &= (u_4)_1 \Phi_{2,1,p} + (u_4)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,l}^1 \left(-\overline{(\hat{u}_{i+1/2}^n)^2} + \overline{g \hat{h}_{i+1/2}^n} \right) (\Phi_{1,i+1,k} - \Phi_{1,i,k}) \right) \Phi_{2,i,p} \\ &\quad - \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i-1/2,l}^1 \left(-\overline{(\hat{u}_{i-1/2}^n)^2} + \overline{g \hat{h}_{i-1/2}^n} \right) (\Phi_{1,i,k} - \Phi_{1,i-1,k}) \right) \Phi_{2,i,p}, \end{aligned} \tag{79}$$

$$(U_5)_{plk} = (u_5)_1 \Phi_{2,1,p} + (u_5)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,l}^0 (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \mathcal{A}_{i-1/2,l}^0 (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{2,i,p}, \tag{80}$$

$$(U_6)_{plk} = (u_6)_1 \Phi_{2,1,p} + (u_6)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} 2 \left(\mathcal{A}_{i+1/2,l}^1 \overline{\hat{u}_{i+1/2}^n} (\Phi_{2,i+1,k} - \Phi_{2,i,k}) - \mathcal{A}_{i-1/2,l}^1 \overline{\hat{u}_{i-1/2}^n} (\Phi_{2,i,k} - \Phi_{2,i-1,k}) \right) \Phi_{2,i,p}, \tag{81}$$

$$\begin{aligned} (U_7)_{pk} &= (u_7)_1 \Phi_{2,1,p} + (u_7)_{N_x} \Phi_{2,N_x,p} + \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i+1/2,k}^1 \left(-\overline{(\hat{u}_{i+1/2}^n)^2} + \overline{g \hat{h}_{i+1/2}^n} \right) (z_{i+1} - z_i) \right) \Phi_{2,i,p} \\ &\quad - \sum_{i=2}^{N_x-1} \left(\mathcal{A}_{i-1/2,k}^1 \left(-\overline{(\hat{u}_{i-1/2}^n)^2} + \overline{g \hat{h}_{i-1/2}^n} \right) (z_i - z_{i-1}) \right) \Phi_{2,i,p}, \end{aligned} \tag{82}$$

and $(u_m)_s$, $m = 1, \dots, 7$, $s \in \{1, N_x\}$, are the boundary terms. This choice will be denoted in the numerical experiments as HLL - DEIM.

5. Well-balanced property

The choice of well-balanced methods to build the FOMs is justified by the following result:

Theorem 1. *Let us suppose that the numerical scheme (3) for the FOM is well-balanced for a stationary solution W^* , i.e., $W_i^n = W_i^1, \forall i, \forall n$, where W_i^1 are the cell-averages of W^* . Then, the associated ROM is also well-balanced for W^* .*

Proof. Let us first show that, if and the initial cell-averages W_i^1 belong to a stationary solution W^* for which the FOM is well-balanced, the matrices $M_{w_j} M_{w_j}^T$ have only one non-zero eigenvalue and, consequently, the ROM has only one mode, i.e., $M = 1$. Indeed, all the columns of M_{w_j} are identical:

$$\begin{aligned} M_{w_j} &= \begin{pmatrix} w_{j,1}^1 & w_{j,1}^2 & \dots & w_{j,1}^{N_T} \\ w_{j,2}^1 & w_{j,2}^2 & \dots & w_{j,2}^{N_T} \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,N_x}^1 & w_{j,N_x}^2 & \dots & w_{j,N_x}^{N_T} \end{pmatrix} = \begin{pmatrix} w_{j,1}^1 & w_{j,1}^1 & \dots & w_{j,1}^1 \\ w_{j,2}^1 & w_{j,2}^1 & \dots & w_{j,2}^1 \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,N_x}^1 & w_{j,N_x}^1 & \dots & w_{j,N_x}^1 \end{pmatrix} \\ &= (w_j^1 | \dots | w_j^1). \end{aligned} \tag{83}$$

Let us consider a vector $v \in \mathbb{R}^{N_x}$. Since

$$M_{w_j}^T v = (w_j^1)^T v \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{84}$$

one has

$$\ker(M_{w_j}^T) = (w_j^1)^\perp. \tag{85}$$

Let us compute the eigenvalues of $M_{w_j} M_{w_j}^T$:

- If $v \in (w_j^1)^\perp$:

$$M_{w_j} M_{w_j}^T v = (w_j^1)^T v M_{w_j} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0. \tag{86}$$

- If $v = w_j^1$:

$$M_{w_j} M_{w_j}^T w_j^1 = \|(w_j^1)\|^2 M_{w_j} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = N_x \|(w_j^1)\|^2 w_j^1. \tag{87}$$

Thus, the matrices $M_{w_j} M_{w_j}^T$, $j = 1, \dots, N$ have only one non-zero eigenvalue $\sigma = N_x \|(w_j^1)\|^2$ with eigenvector w_j^1 .

Consequently, only one POD mode is selected to build the associated ROM. If we denote by $w P_j^n$ the projection of w_j^n into the functions of the POD basis, one has:

$$\frac{1}{N_T} \|w_j^n - w P_j^n\|^2 = \sum_{m=M+1}^r \sigma_m. \tag{88}$$

Notice that, since $\sigma_m = 0, \forall m > 1$, one has the equality $w_j^1 = w P_j^1$. Due to the fact that the initial condition for the ROM is computed from the FOM by projecting in the POD basis, at the initial time t^1 , one has

$$w r_{j,i}^1 = \hat{w}_{j,i}^1 \Phi_{j,i,1} = w P_{j,i}^1 = w_{j,i}^1. \tag{89}$$

This implies that

$$\mathcal{L}_j (W r_{i-1}^1, W r_i^1, W r_{i+1}^1) = \mathcal{L}_j (W_{i-1}^1, W_i^1, W_{i+1}^1), \tag{90}$$

and thus

$$\hat{w}_{j,i,1}^2 = \hat{w}_{j,i,1}^1 \Rightarrow w r_{j,i}^2 = \hat{w}_{j,i,1}^2 \Phi_{j,i,1} = w r_{j,i}^1, \forall i. \tag{91}$$

Reasoning in this way, one has:

$$w r_{j,i}^n = w r_{j,i}^1, \forall i, \forall n, \forall j. \quad \square \tag{92}$$

Corollary 1. *The ROMs considered in Section 4 are fully exactly well-balanced, i.e., they preserve all stationary solutions for the transport equation with a linear source term and the Burgers equation with a nonlinear source term, while in the shallow water model case the ROM is exactly well-balanced for the family of stationary solutions corresponding to water at rest.*

6. Simulation in parameter-dependent systems

Let us suppose that the source term in (1) depends on a physical parameter μ , i.e., the PDE can be written as follows:

$$W_t + F(W)_x = G(W, \mu). \tag{93}$$

Notice that the three systems of balance laws introduced in Section 2.1 are of this type. Indeed, in the transport equation with linear source term $S(w) = \beta w$ and the Burgers equation with nonlinear source term $S(w) = \beta w^2$, the parameter is $\mu = \beta$, whereas in the SW system with Manning friction (19),

$$R(W) = \left(0, -g \frac{n_b^2 q |q|}{h^{7/3}} \right)^T,$$

the parameter μ is the friction coefficient ($\mu = n_b$).

Sometimes, our interest is to quickly obtain good approximations of the solutions when considering variations of the parameter μ . Then, our next goal is to efficiently provide accurate approximations of the solutions for different values of this parameter, enhancing the flexibility and speed of the model in capturing variations in physical scenarios. The POD-based strategy described in the previous sections will be used to build the predictive ROMs. The key to these prediction models is to include a training step, that is, numerical solutions computed with the FOM for a set of values of the parameter $\mu_l, l = 1, \dots, N_\mu$, called *training set*, will be considered in the offline process. In particular, the POD basis functions must contain information of these solutions obtained for the different values of the training set. Thus, if a new value of μ which does not belong to the training set is considered, the predictive ROM will give acceptable approximations.

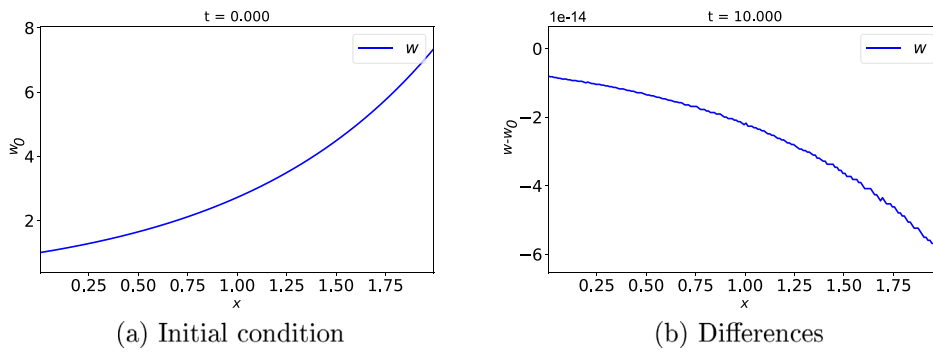


Fig. 1. Test 7.1.1. Well-balanced property check: transport equation. Initial condition (left) and differences between the initial and final numerical solutions for a 200-cell mesh at $T_f = 10$ s with the ROM (right).

Given the variable $w_j, j = 1, \dots, N$, its snapshot matrix M_{w_j} reads now as follows:

$$M_{w_j} = \left(M_{w_j}^{\mu_1} | \dots | M_{w_j}^{\mu_{N_\mu}} \right) \in \mathbb{R}^{N_x \times (N_l \cdot N_\mu)}, \tag{94}$$

where $M_{w_j}^{\mu_l}$ is the snapshot matrix obtained with the FOM with $\mu = \mu_l, l = 1, \dots, N_\mu$. A previous work [19] studies reduced order modeling techniques for hyperbolic conservation laws with applications in uncertainty quantification. In that study, a POD-Greedy approach combined with an empirical interpolation method (EIM) is employed to create a reduced basis space. It should be noted that this is a different approach from that presented in this section, where the technique described in Section 3 is used to build ROMs that can be used to obtain good approximations of the solutions for parameterized systems trained with a set of different parameter values when considering other parameter values outside the training set. Unlike in the paper [19], here we consider hyperbolic PDE systems with source terms (in fact, the parameters considered in this paper belong to this term).

7. Numerical experiments

In all the experiments, free boundary conditions are considered and the CFL parameter is set to 0.9.

7.1. Validation of the well-balanced property

This first experiment is devoted to the validation of the well-balanced property proved in Theorem 1. Stationary solutions for the three systems described in Section 2.1 have been considered, checking that the corresponding discrete stationary solution defined by its projection into the POD basis function is preserved in the ROM. Since a stationary solution is chosen as initial condition, the number of POD modes is $M = 1$. Thus, only one time window $N_t = 1$ is considered.

7.1.1. Transport equation with a linear source term

Let us consider the transport equation with linear source term (11), with $c = 1$ and $\beta = 1$. Let us consider the space interval $[0, 2]$ and the time interval $[0, 10]$. As initial condition, we consider discrete stationary solution given by the projection into the POD basis of the stationary solution

$$w_0(x) = e^x$$

for the FOM (see Fig. 1, left). L^1 -errors between the initial and final cell-averages of the ROM have been computed for different meshes (see Table 1). In particular, Fig. 1 right shows the differences between the initial and final solutions for a 200-cell mesh at $T_f = 10$ s with the ROM. CPU times in seconds required by the FOM and ROM to obtain the solution at final time are also shown in Table 1, with the corresponding speed-ups also reported. As expected, the ROM is faster than the FOM, obtaining a significant reduction in computational time if the number of cells is high.

7.1.2. Burgers equation with a nonlinear source term

Let us consider the Burgers equation with a nonlinear source term (15), with $\beta = 1$. Let us consider the space interval $[0, 2]$ and the time interval $[0, 10]$. As initial condition, we consider discrete stationary solution given by the projection into the POD basis of the stationary solution

$$w_0(x) = 0.1e^x$$

for the FOM. L^1 -errors between the initial and final cell-averages of the ROM have been computed for different meshes (see Table 2). CPU times in seconds required by the FOM and ROM to obtain the solution at final time and speed-ups are also shown in Table 2. Again, the ROM is faster than the FOM, specially if fine meshes are considered.

Table 1

Test 7.1.1. Well-balanced property check: transport equation. L^1 -errors between the initial and final cell-averages of the ROM, CPU times in seconds required by the FOM and ROM to obtain the solution at $T_f = 10$ s and speed-ups.

Cells	Errors with ROM	CPU time		Speed-up
		FOM	ROM	
200	5.17e-14	1.24	1.13	1.10
400	1.26e-15	6.01	4.35	1.38
800	6.40e-14	34.23	16.65	2.06
1600	0.0	201.24	68.02	2.95

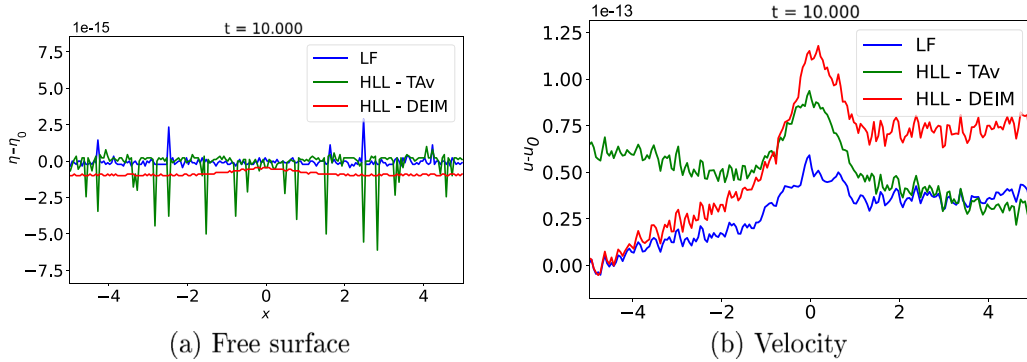


Fig. 2. Test 7.1.3. Well-balanced property check: shallow water equations. Differences between the initial and final numerical solutions for a 200-cell mesh at $T_f = 10$ s with the ROM: free surface η (left) and velocity u (right).

Table 2

Test 7.1.2. Well-balanced property check: Burgers equation. L^1 -errors between the initial and final cell-averages of the ROM and CPU times in seconds required by the FOM and ROM to obtain the solution at $T_f = 10$ s and speed-ups.

Cells	Errors with ROM	CPU time		Speed-up
		FOM	ROM	
200	0.0	0.85	0.8	1.06
400	2.33e-15	4.38	3.21	1.36
800	1.09e-16	22.88	11.74	1.95
1600	8.21e-14	130.46	50.12	2.60

7.1.3. Frictionless shallow water equations with non-flat bathymetry

Let us consider the SW system (19) with $n_b = 0$. Let us consider the space interval $[-5, 5]$ and the time interval $[0, 10]$. As initial condition, we consider discrete stationary solution given by the projection into the POD basis of the stationary solution

$$h_0(x) = -z(x), \quad u_0(x) = 0, \quad z(x) = -1 + 0.5e^{-x^2},$$

for the FOM. The gravity is $g = 9.81$. The modified Lax–Friedrichs flux and the HLL scheme (considering the two versions HLL - TAv and HLL - DEIM) have been applied. L^1 -errors between the initial and final cell-averages of the ROM have been computed for different meshes (see Tables 3 and 4 for modified Lax–Friedrichs and HLL, respectively). In particular, Fig. 2 shows the differences between the initial and final solutions for a 200-cell mesh at $T_f = 10$ s with the ROM for the different schemes. CPU times in seconds required by the FOM and ROM to obtain the solution at final time are also shown in Tables 3 and 4, which also include the speed-up. For this test case, the ROM is much faster than the FOM: particularly remarkable is the reduction obtained for the fine mesh with 1600 cells, where the CPU time is reduced by almost 6 times for the modified Lax–Friedrichs scheme and more than 26 times for the HLL.

7.2. Relationship between the number of POD modes, time windows and errors

7.2.1. Transport equation with a linear source term

Let us first consider the transport equation with linear source term (11), with $c = 1$ and $\beta = 1$. The space interval is $[0, 2]$ and the time interval $[0, 0.8]$. As initial condition, we consider discrete stationary solution given by the projection into the POD basis of

Table 3

Test 7.1.3. Well-balanced property check: shallow water equations. Modified Lax–Friedrichs scheme. L^1 -errors between the initial and final cell-averages of the ROM and CPU times in seconds required by the FOM and ROM to obtain the solution at $T_f = 10$ s, and the speed-ups.

Cells	Errors with ROM		CPU time		Speed-up
	h	q	FOM	ROM	
200	3.00e-14	2.41e-13	2.72	1.31	2.07
400	2.94e-13	5.91e-13	7.31	4.01	1.82
800	1.59e-13	5.10e-13	24.38	8.62	2.82
1600	4.32e-15	7.35e-14	96.79	16.70	5.80

Table 4

Test 7.1.3. Well-balanced property check for the shallow water equations using the HLL scheme. The table reports the L^1 -errors between the initial and final cell-averages for both ROM-TAv and ROM-DEIM, the CPU times (in seconds) required by the FOM and each ROM variant to compute the solution at $T_f = 10$ s, and the corresponding speed-ups obtained by each reduced model.

Cells	Errors (TAv)		Errors (DEIM)		CPU time			Speed-up	
	h	q	h	q	FOM	TAv	DEIM	TAv	DEIM
200	2.92e-14	4.72e-14	3.02e-14	4.78e-13	8.16	2.11	2.13	3.87	3.83
400	3.50e-13	3.99e-13	2.77e-13	2.14e-13	23.92	4.08	4.18	5.86	5.72
800	2.74e-14	2.87e-13	6.29e-14	3.03e-13	90.70	7.76	7.95	11.69	11.41
1600	4.19e-15	1.18e-13	7.58e-14	4.22e-14	438.20	15.34	16.50	28.57	26.56

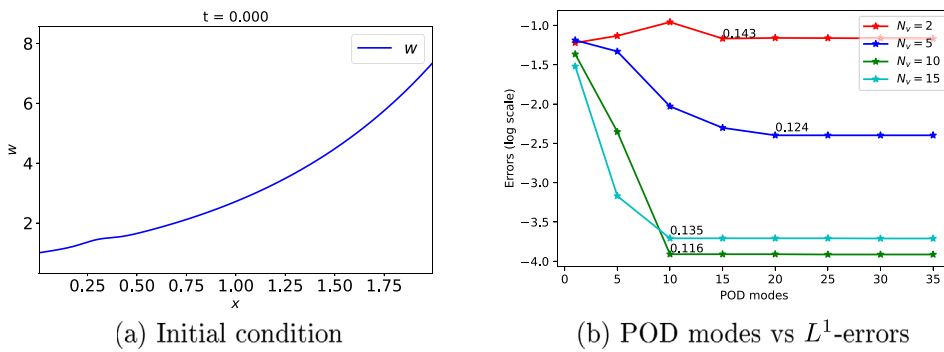


Fig. 3. Test 7.2.1. Transport equation: small perturbation of a stationary solution. Initial condition (left) and number of POD modes vs. the L^1 -errors in logarithmic scale for different number of time windows (right).

a small perturbation of a stationary solution

$$w_0(x) = e^x + 0.1e^{-100(x-0.3)^2}$$

(see Fig. 3 (left)). We consider a mesh with 200 cells.

In order to study the relationship between the number of POD modes M and the time windows N_v , L^1 -errors between the numerical solutions of the FOM and ROM have been computed in the final time $T_f = 0.8$ s considering different numbers of modes for several values of N_v . Fig. 3 (right) shows the results obtained: the number of POD modes vs. the L^1 -errors in logarithmic scale are plotted. As expected, it is observed that for each value of N_v , the errors start to decay with increasing M and end up stabilizing. In addition, the numbers appearing at each line for the value M at which the errors stagnate are the CPU times obtained. Note that a large number of modes and time windows are not necessary: in this case, taking $M = 10$ with $N_v = 10$, we obtain errors of the order of $1e-4$. The CPU time required by the FOM to reach the final solution at time $t = 0.8$ s was 0.43 s. In Table 5, the speed-ups between the FOM and the ROM are reported for the different time windows, considering the number of modes at which the errors have stabilized, as observed in Fig. 3 (right).

Moreover, since a perturbation of a steady state is selected as initial condition, let us check what happens when the perturbation leaves the domain. For this purpose, we now consider a total simulation time $T_f = 10$ s. and $N_v = 100$. Fig. 4 shows the numerical solutions obtained for FOM and ROM at times $t = 0.8$ s. and $t = 10$ s. The difference obtained in L^1 -norm between both solutions is equal to $7.97e-8$. In addition, we have measured the error between the last two iterations of the ROM to check that a steady state has been reached: indeed, the error obtained is $5.62e-16$.

7.2.2. Burgers equation with a nonlinear source term

Let us consider the Burgers equation with a nonlinear source term (15), with $\beta = 1$. The space interval is $[0, 2]$ and the time interval $[0, 3]$. As initial condition, we consider discrete stationary solution given by the projection into the POD basis of a perturbation of

Table 5

Test 7.2.1. Transport equation: speed-ups obtained by the ROM for different numbers of time windows N_v , computed as the ratio between the FOM time (0.43 s) and the ROM CPU times. The number of modes corresponds to the value at which the error was observed to stabilize in Fig. 3 (right).

N_v	M	ROM CPU time	Speed-up
2	15	0.143	3.01
5	20	0.124	3.47
10	10	0.116	3.71
15	10	0.135	3.19

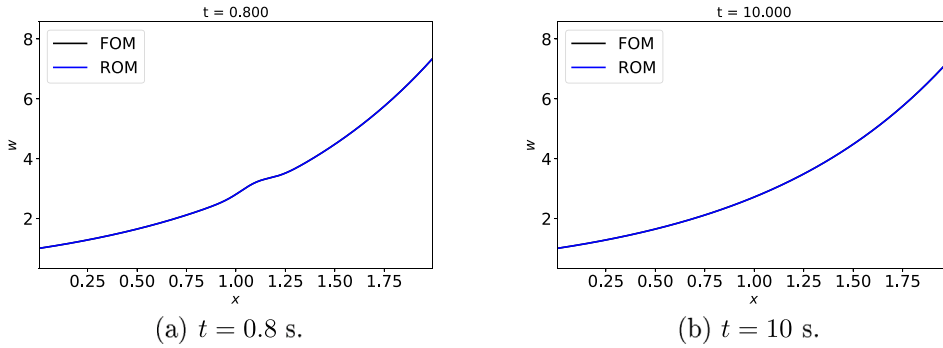


Fig. 4. Test 7.2.1. Transport equation: small perturbation of a stationary solution. Numerical solutions of FOM and ROM at $t = 0.8$ s. and $t = 10$ s.

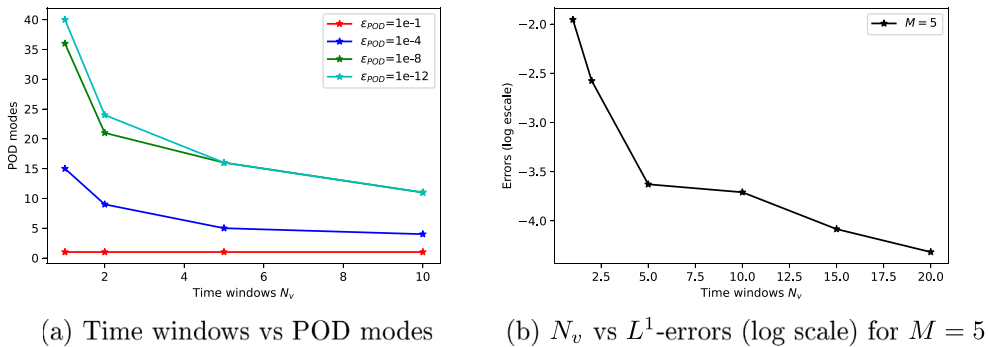


Fig. 5. Test 7.2.2. Burgers equation: perturbation of a stationary solution. Time windows vs. POD modes (left) and Time windows vs. L^1 -errors in logarithmic scale choosing 5 POD modes (right).

a stationary solution

$$u_0(x) = 0.1e^x + 0.1e^{-100(x-0.3)^2}.$$

We consider a mesh with 200 cells.

Let us study the relationship between the number of time windows N_v and the number of POD modes M given a tolerance ϵ_{POD} for the criterion to choice M in (33). Given different tolerances ϵ_{POD} , Fig. 5 (left) shows the number of POD modes obtained for some values of N_v . As expected, the number of POD modes is considerably reduced as N_v increases.

Since this is a nonlinear system, the use of time windows plays a fundamental role. To reinforce this statement, we have also checked that, if we fix the number of POD modes and increase N_v , we obtain better approximations of the solution. Fig. 5 (right) shows the L^1 -errors in logarithmic scale between the numerical solutions of the FOM and ROM for 6 different values of N_v taking 5 POD modes. The CPU time required by the FOM to reach the final solution at $T_f = 3$ s is 0.277 s. Table 6 reports the speed-ups obtained with 5 modes and different numbers of time windows N_v . These results show that the speed-up remains relatively stable across different values of N_v when the number of modes is fixed. In Fig. 6 the solutions obtained with the ROM for these values of N_v are plotted and compared with the solution of the FOM (in black).

Table 6

Test 7.2.2. Burgers equation: perturbation of a stationary solution. Speed-ups obtained by the ROM using 5 modes for different numbers of time windows N_v .

N_v	ROM Time	Speed-up
1	0.158	1.75
2	0.162	1.71
5	0.167	1.66
10	0.164	1.69
15	0.166	1.67
20	0.167	1.66

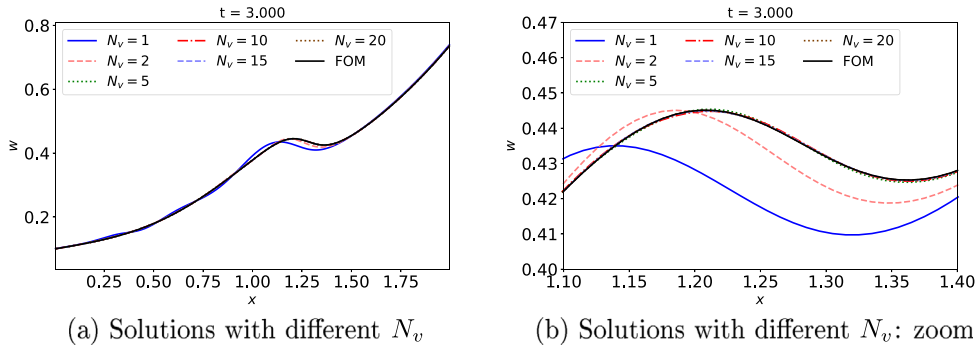


Fig. 6. Test 7.2.2. Burgers equation: perturbation of a stationary solution. Numerical solutions at $T_f = 3$ s. for different time windows choosing 5 POD modes (left) and zoom (right).

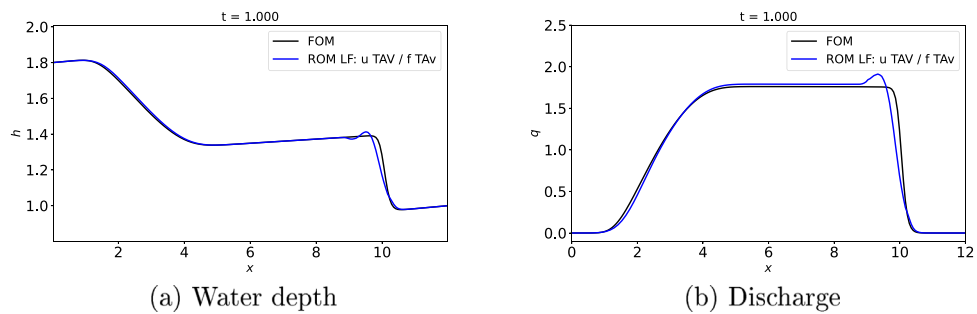


Fig. 7. Test 7.3. SW equations: dam-break. Lax–Friedrichs numerical flux and time-averaging for u and $f = |q|/h^{7/3}$. Water depth h (left) and discharge q (right) at $T_f = 1$.

7.3. Shallow water equations: dam-break problem

We consider the SW system with $n_b = 0.1$. The space interval is $[0, 12]$ and the time interval is $[0, 1]$. A mesh with 200 cells is considered. The initial condition is a dam-break with a mild slope:

$$h_0(x) = \begin{cases} 2 - z(x) & \text{if } 0 \leq x \leq 6 \\ 1 - z(x) & \text{if } 6 < x \leq 12 \end{cases}, \quad u_0(x) = 0, \tag{95}$$

where

$$z(x) = 0.2(1 - x/12). \tag{96}$$

The number of time windows is $N_v = 5$ and $\epsilon_{\text{POD}} = 1e-10$. Our goal is to compare the time-averaging and the DEIM approaches. Let us first apply the modified Lax–Friedrichs numerical flux. Following Section 4.3.1, we can consider different choices to linearize the friction term in order to build the ROM.

Firstly, we consider time-averaging for u and $f = |q|/h^{7/3}$, i.e., the expression of the friction term is given by (57)–(58). With the previous conditions, the selected number of POD modes is $M = 21$. Fig. 7 shows, from left to right, the water height h and the discharge q . Notice that significant spurious oscillations appear near the front shock.

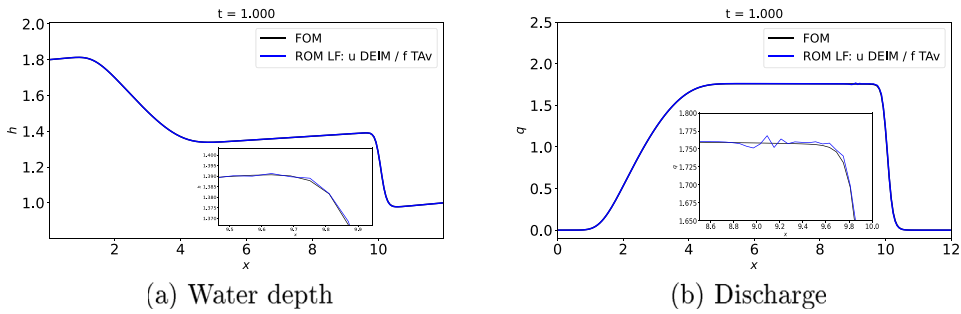


Fig. 8. Test 7.3. SW equations: dam-break. Lax–Friedrichs numerical flux, DEIM approach for u and time-averaging for $f = |q|/h^{7/3}$. Water depth h (up) and discharge q (right) at $T_f = 1$ s. Inner figures: zooms.

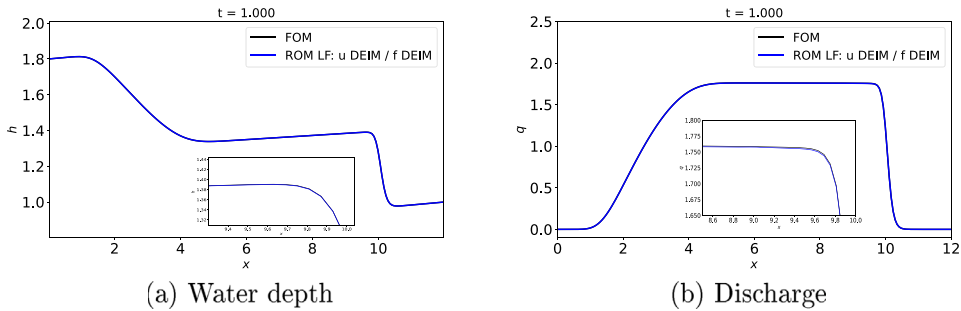


Fig. 9. Test 7.3. SW equations: dam-break. Lax–Friedrichs numerical flux, DEIM approach for u and $f = |q|/h^{7/3}$. Water depth h (left), discharge q (right) at $T_f = 1$ s. Inner figures: zooms.

Secondly, we consider the DEIM approach for u and time-averaging for $f = |q|/h^{7/3}$, leading to the expression (59)–(60) for the friction term in the ROM. Given the previous conditions, the selected number of POD modes is $M = 17$. Fig. 8 (up) shows, from left to right, the water depth h of the water and the discharge q . Observe that, for the variable q , some spurious oscillations appear near the front shock, although they are significantly smaller than in the previous case: Fig. 8 (down) zooms in near the shock for q .

Finally, let us consider DEIM for u and $f = |q|/h^{7/3}$, which gives the expression (62)–(63)–(64) of the friction term in the ROM. Given the previous choices, the selected number of POD modes is $M = 20$. Fig. 9 (up) shows, from left to right, the variables h and q . Note that the spurious oscillations which appear near the front shock in the previous case are no longer in this plot, unlike the previous cases. Fig. 9 (down) zooms in near the shock for q .

Moreover, errors in L^1 -norm between the solution of the ROM and a numerical solution computed with the FOM with the parameter $n_b = 0.1$ are shown in Table 7 for these three possible choice for the linearization of u and f . We can observe that errors are reduced when using the DEIM approach.

Let us now consider the HLL numerical flux. As a result of the above findings, the friction term is linearized using the DEIM approach for u and $f = |q|/h^{7/3}$ as in the last of the previous cases. In order to verify that the DEIM strategy does indeed give better results than the time-averaging technique, these two approaches have been applied now to update the coefficients of the diffusive term of the numerical flux $\alpha_{i+1/2}^{0,n}$ and $\alpha_{i+1/2}^{1,n}$, (see (23)–(24)).

First, let us consider DEIM for u and $f = |q|/h^{7/3}$ and the time-averaging approach for the coefficients $\alpha_{i+1/2}^{0,n}$ and $\alpha_{i+1/2}^{1,n}$. Considering the previous choices, the selected number of POD modes is $M = 20$. Fig. 10 (up) shows, from left to right, the height of the water h and the discharge q . Again, spurious oscillations appear near the front shock. Fig. 10 (down) zooms in near the shock for q .

Lastly, the DEIM approach is considered for u , $f = |q|/h^{7/3}$ and the coefficients $\alpha_{i+1/2}^{0,n}$ and $\alpha_{i+1/2}^{1,n}$. Considering the previous choices, the selected number of POD modes is again $M = 20$. Fig. 11 (up) shows, from left to right, the height of the water h and the discharge q . Spurious oscillations are reduced using DEIM near the front shock. Fig. 11 (down) zooms in near the shock for q .

Errors in L^1 -norm between the solution of the ROM and a numerical solution computed with the FOM with the parameter $n_b = 0.1$ are shown in Table 8. It can be observed that by considering DEIM linearization errors are reduced.

This experiments shows that the DEIM approach introduced in this work produces better results in the presence of discontinuities than other linearization techniques previously described in the literature such as time-averaging.

Table 9 summarizes the CPU times and speed-ups for different ROM configurations. For Lax–Friedrichs, full DEIM yields the best accuracy, while, as expected, the full time-averaging configuration is fastest. The mixed approach balances both. A similar pattern appears with HLL: applying DEIM to u and f and time-averaging the coefficients $(\alpha_{i+1/2}^{0,n}, \alpha_{i+1/2}^{1,n})$ achieves the highest speed-up,

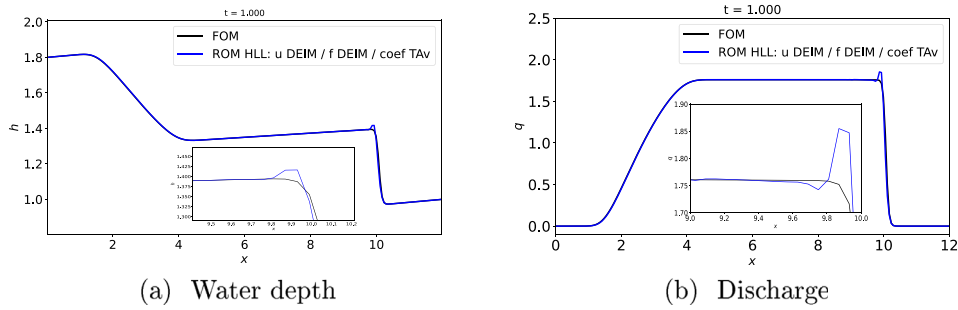


Fig. 10. Test 7.3. SW equations: dam-break. HLL numerical flux, DEIM approach for u and $f = |q|/h^{7/3}$ and the time-averaging approach for the coefficients $\alpha_{i+1/2}^{s,n}$, $s = 1, 2$. Water depth h (left), discharge q (right) at $T_f = 1$ s. Inner figures: zooms.

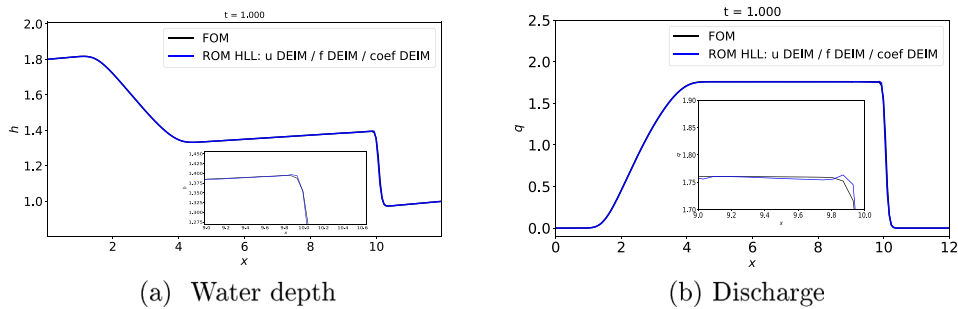


Fig. 11. Test 7.3. SW equations: dam-break. HLL numerical flux, DEIM approach for u , $f = |q|/h^{7/3}$ and the coefficients $\alpha_{i+1/2}^{s,n}$, $s = 1, 2$. Water depth h (left), discharge q (right) at $T_f = 1$ s. Inner figures: zooms.

Table 7

Test 7.3. SW equations: dam-break, Lax–Friedrichs numerical flux. Errors in L^1 -norm between the solution of the ROM and a numerical solution computed with the FOM with the parameter $n_b = 0.1$ for h and q at $T_f = 1$ s. by considering time-averaging or DEIM linearization for u or $f = |q|/h^{7/3}$.

Lax–Friedrichs	Linearization		
	u time-averaging f time-averaging	u DEIM f time-averaging	u DEIM f DEIM
h	8.13e–02	1.85e–03	9.48e–04
q	5.25e–01	1.81e–02	9.47e–03

Table 8

Test 7.3. SW equations: dam-break, HLL numerical flux. Errors in L^1 -norm between the solution of the ROM and a numerical solution computed with the FOM with the parameter $n_b = 0.1$ for h and q at $T_f = 1$ s. by considering time-averaging or DEIM linearization for $\alpha_{i+1/2}^{0,n}$ and $\alpha_{i+1/2}^{1,n}$ and DEIM for u and $f = |q|/h^{7/3}$.

HLL	Linearization	
L^1 error	u DEIM, ($\alpha_{i+1/2}^{0,n}$, $\alpha_{i+1/2}^{1,n}$) f DEIM, time averaging	u DEIM, ($\alpha_{i+1/2}^{0,n}$, $\alpha_{i+1/2}^{1,n}$) f DEIM, DEIM
h	1.52e–02	8.33e–03
q	6.55e–02	4.80e–02

whereas full DEIM improves accuracy slightly. Overall, the results suggest that combining DEIM and time-averaging offers the best trade-off between precision and performance.

7.4. Simulation in parameter-dependent systems

We consider again the SW system with friction (19). The space interval is $[0, 12]$ and the time interval is $[0, 1]$. A mesh with 200 cells is taken. Again, the initial condition is the dam-break given by (95)–(96). Our goal is to build a predictive ROM following Section 6, where the parameter to be predicted is the friction coefficient n_b . We apply the modified Lax–Friedrichs numerical flux with the linearization (63)–(64) of the friction term. It is worth noting that, in this experiment, the cost of constructing the reduced bases from the snapshot matrices (formed by assembling submatrices corresponding to different values of the parameter in the training

Table 9

Test 7.3. SW equations: dam-break. CPU times and speed-ups for different ROM configurations using the Lax–Friedrichs and HLL schemes. TAv stands for time-averaging. The FOM CPU times are 0.367 s for Lax–Friedrichs and 0.723 s for HLL.

Scheme	u	f	$(\alpha_{i+1/2}^{0,n}, \alpha_{i+1/2}^{1,n})$	ROM CPU time	Speed-up
Lax–Friedrichs	TAv	TAv	–	0.212	1.73
	DEIM	TAv	–	0.215	1.71
	DEIM	DEIM	–	0.224	1.64
HLL	DEIM	DEIM	TAv	0.251	2.88
	DEIM	DEIM	DEIM	0.377	1.92

Table 10

Test 7.4. SW equations with friction. Errors in L^1 -norm between numerical solution computed with the FOM with the parameter $n_b = 0.035$ and the solution of the ROM with the same parameter when projecting onto the entire space (maximum number of modes) and when using only 5 modes at $T_f = 1$ s.

	Projection	5 POD modes
h	1.27e–04	1.70e–04
q	3.86e–04	7.14e–04

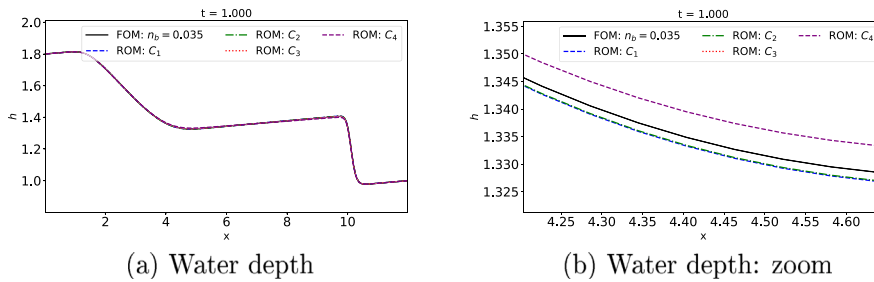


Fig. 12. Test 7.4. Predictive ROMs for the SW equations with friction. Prediction for $n_b = 0.035$ using the training sets C_k , $k = 1, 2, 3, 4$. Water depth h (left) and zoom in (right) at $T_f = 1$ s.

set) is performed during the offline phase and therefore does not contribute to the computational cost. As a result, the speed-up achieved in this case is expected to be of the same order as that obtained in the previous test, considering the Lax–Friedrichs scheme and the same linearization of u and the friction term. Let us consider the Manning’s friction coefficient as $n_b = 0.035$. Before building predictive ROMs, we have first evaluated the errors obtained by using the FOM for this value and then deriving the associated ROM. We consider $N_v = 25$ time windows and $\epsilon_{\text{POD}} = 1e-10$. Table 10 presents the L^1 -norm errors when projecting onto the entire space (maximum number of modes) and when using only 5 modes. In both cases, the errors are of the order of $1e-4$.

Now, we have applied the procedure introduced in Section 6 to construct different ROMs using bases derived from 4 distinct training sets, ensuring that none of the sets includes parameter $n_b = 0.035$. The training sets, labeled as C_1, C_2, C_3 and C_4 , are the following:

$$C_1 = \{0, 1\}, C_2 = \{0.01, 0.05, 0.09\}, C_3 = \{0.03, 0.04\}, C_4 = \{0.07, 0.09\}.$$

Figs. 12–14 display the solutions obtained from the 4 predictive ROMs and the absolute value of the differences between the FOM numerical solution for $n_b = 0.035$ and the ROMs for the training sets C_k , $k = 1, 2, 3, 4$. The L^1 -errors between the FOM solution and the ones computed with the four ROMs are summarized in Table 11.

Notice that when using C_1 , which contains extreme values of the friction parameter, we obtain a good approximation of the solution. Furthermore, as expected, the error decreases when the training sets are based on more information around the parameter being predicted $n_b = 0.035$, as seen with sets C_2 and C_3 . In fact, with the latter, the error decreases by an entire order of magnitude. Finally, when considering training set C_4 , which is distant from the parameter being predicted, we observe that although the error increases, the solutions still provide reasonably accurate approximations: the predicted ROM matches well with the FOM considering the exact Manning friction parameter $n_b = 0.035$, obtaining small errors. Therefore, in view of the results, the technique for designing ROMs that allows us to make parameter predictions described in Section 6 seems to provide accurate approximations in the considered framework.

8. Conclusions

This work addresses the resolution of hyperbolic systems of balance laws by implementing POD-ROMs. The use of these methods has proven to be computationally efficient, offering significantly less expensive solutions compared to traditional FOMs based on

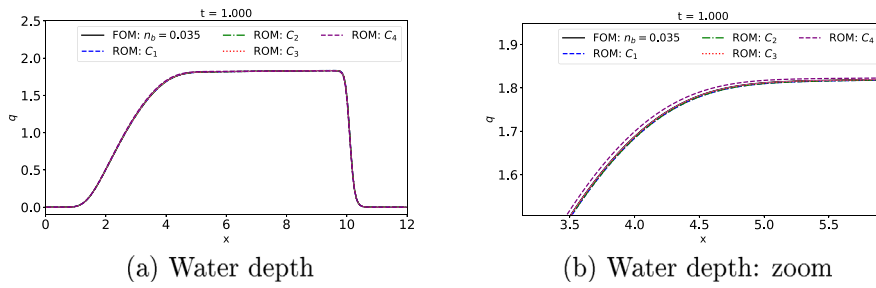


Fig. 13. Test 7.4. Predictive ROMs for the SW equations with friction. Prediction for $n_b = 0.035$ using the training sets C_k , $k = 1, 2, 3, 4$. Discharge q (left) and zoom in (right) at $T_f = 1$ s.

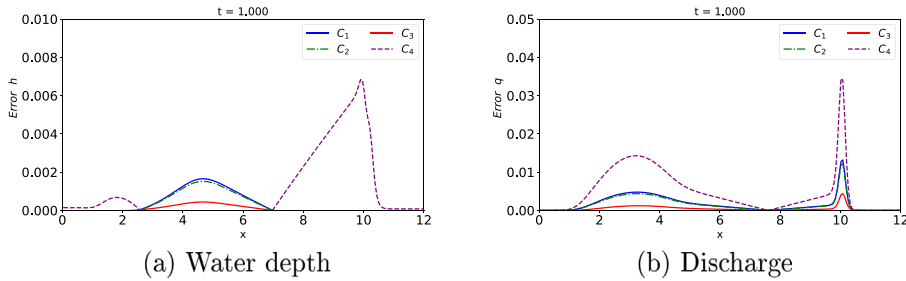


Fig. 14. Test 7.4. Predictive ROMs for the SW equations with friction. Prediction for $n_b = 0.035$ using the training sets C_k , $k = 1, 2, 3, 4$. Differences in absolute value between the FOM computed with $n_b = 0.035$ and the predictive ROMs. Water depth h (left) and discharge q (right) at $T_f = 1$ s.

Table 11

Test 7.4. Predictive ROMs for the SW equations with friction. Prediction for $n_b = 0.035$ using the training sets C_k , $k = 1, 2, 3, 4$. Errors in L^1 -norm between the numerical solution computed with the FOM with the parameter $n_b = 0.035$ and the solution of the predictive ROMs at $T_f = 1$ s.

	C_1	C_2	C_3	C_4
h	8.56e-03	7.87e-03	2.33e-03	2.49e-02
q	2.09e-02	1.92e-03	5.63e-03	6.17e-02

FV schemes. However, since these systems of partial differential equations often include nonlinear terms, dealing with them using ROMs poses a challenge. Appropriate techniques are required to handle these nonlinearities while maintaining accuracy.

Previous works, such as [22], have applied the PID approach in combination with time averaging techniques to address the nonlinearities when deriving ROMs. In this work, we introduce an alternative approach based on DEIM to overcome this challenge, once again in combination with the PID approach. The influence of the number of modes and time windows on the results in terms of error and computational time has been investigated through several numerical experiments involving the transport equations and Burgers' equation, both with source terms. Additionally, we compared the time-averaging strategy with the DEIM-based linearization strategy for the SW system with Manning friction. The results demonstrate that the novel DEIM-based technique presented in this article is more accurate and reduces spurious oscillations near discontinuities.

A key finding of this study is the demonstration of an important result concerning the well-balanced property of ROMs. A theorem is introduced, showing that if a ROM is derived from an exactly well-balanced FOM, the ROM inherits this property. To the best of our knowledge, it is the first time that a result on the well-balancedness of the reduced order models has been proved. The key demonstration is that if the original scheme is well-balanced, and the initial conditions are cell-averages of a stationary solution, the snapshot matrix contains only one nonzero eigenvector, meaning the ROM is fully represented by a single POD mode. Numerically, the well-balanced property has been confirmed in various hyperbolic problems, consistently yielding machine precision errors in the preservation of steady-state solutions.

Finally, given the computational advantages of ROMs and their application to hyperbolic systems dependent on certain physical parameters, this work presents a methodology for constructing predictive ROMs. These models are designed using a training set that incorporates different values of the parameter, aiming to deliver fast and accurate results for new parameter values not included in the training set. A sensitivity analysis of this methodology was conducted for the shallow water system, considering the Manning's friction as the parameter to be estimated, and the results obtained are highly promising.

As possible future work, it would be interesting to apply these approach to other geophysical fluids and also extend it to the two-dimensional case.

Acknowledgments

This work is partially supported by the Grant QUALIFICA (programa: ayudas a acciones complementarias de I+D+i) by Junta de Andalucía, Spain grant number QUAL21 005 USE. This research has also been partially supported by Junta de Andalucía, Spain research project ProyExcel_00525, by the European Union - NextGenerationEU program and by grants PID2022-137637NB-C21 and PID2022-137637NB-C22 funded by MCIN/AEI/10.13039/501100011033 and “ERDF A way of making Europe” and by the Spanish Government Project PID2021-123153OB-C21, funded by MCIN/AEI/10.13039/501100011033, UE. Funding for open access charge: Universidad de Málaga/CBUA.

Data availability

No data was used for the research described in the article.

References

- [1] J.S. Hesthaven, G. Rozza, B. Stamm, et al., *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, 2016, p. 590.
- [2] J.L. Lumley, The structure of inhomogeneous turbulent flows, *Atmospheric Turbul. Radio Wave Propag.* (1967) 166–178.
- [3] L. Sirovich, Turbulence and the dynamics of coherent structures I. Coherent structures, *Quart. Appl. Math.* 45 (3) (1987) 561–571.
- [4] M. Azaïez, T. Chacón, S. Rubino, A cure for instabilities due to advection-dominance in POD solution to advection-diffusion-reaction equations, *J. Comput. Phys.* 425 (2021) 109916.
- [5] S. Giere, T. Iliescu, V. John, D. Wells, SUPG reduced order models for convection-dominated convection–diffusion–reaction equations, *Comput. Methods Appl. Mech. Engrg.* 289 (2015) 454–474.
- [6] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Fluids* 72 (12) (2013) 1219–1243.
- [7] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Engrg.* 102 (5) (2015) 1136–1161.
- [8] B. Galletti, C.H. Bruneau, L. Zannetti, A. Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, *J. Fluid Mech.* 503 (2004) 161–170.
- [9] J. Novo, S. Rubino, Error analysis of proper orthogonal decomposition stabilized methods for incompressible flows, *SIAM J. Numer. Anal.* 59 (1) (2021) 334–369.
- [10] T. Chacón Rebollo, S. Rubino, M. Oughelou, C. Allery, Error analysis of a residual-based stabilization-motivated POD-ROM for incompressible flows, *Comput. Methods Appl. Mech. Engrg.* 401 (2022) 115627.
- [11] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison, *Comput. Methods Appl. Mech. Engrg.* 237 (2012) 10–26.
- [12] S.E. Ahmed, O. San, D.A. Bistrrian, 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.
- [13] M. Dehghan, M. Abbaszadeh, The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations, *J. Comput. Phys.* 351 (2017) 478–510.
- [14] R. Ștefănescu, I.M. Navon, POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model, *J. Comput. Phys.* 237 (2013) 95–114.
- [15] X. Zeng, G. Stabile, E.N. Karatzas, G. Scovazzi, G. Rozza, Embedded domain reduced basis models for the shallow water hyperbolic equations with the shifted boundary method, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115143.
- [16] 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.
- [17] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.* 656 (2010) 5–28.
- [18] S. Chaturantabut, D.C. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, IEEE, 2009, pp. 4316–4321.
- [19] R. Crisovan, D. Torlo, R. Abgrall, S. Tokareva, Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification, *J. Comput. Appl. Math.* 348 (2019) 466–489.
- [20] H. Fu, H. Wang, Z. Wang, POD/DEIM reduced-order modeling of time-fractional partial differential equations with applications in parameter identification, *J. Sci. Comput.* 74 (2018) 220–243.
- [21] R. Ștefănescu, A. Sandu, I.M. Navon, Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations, *Internat. J. Numer. Methods Fluids* 76 (8) (2014) 497–521.
- [22] 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.* 410 (2023) 116038.
- [23] M. Ahmed, O. San, Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks, *Comput. Appl. Math.* 37 (5) (2018) 6870–6902.
- [24] W.L. IJzerman, *Signal Representation and Modeling of Spatial Structures in Fluids* (Ph.D. thesis), 2000.
- [25] I. Gómez-Bueno, M.J. Castro, C. Parés, G. Russo, Collocation methods for high-order well-balanced methods for systems of balance laws, *Mathematics* 9 (15) (2021) 1799.
- [26] J.M. Greenberg, A.Y. LeRoux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, *SIAM J. Numer. Anal.* 33 (1) (1996) 1–16.
- [27] R. Abgrall, Y. Liu, A new approach for designing well-balanced schemes for the shallow water, *Appl. Math. Comput.* 5 (2023) 370–402.
- [28] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, *SIAM J. Sci. Comput.* 25 (6) (2004) 2050–2065.
- [29] Y.N. Cheng, A. Kurganov, Moving-water equilibria preserving central-upwind schemes for the shallow water equations, *Commun. Math. Sci.* 14 (2016) 1643–1663.
- [30] M. Ciallella, D. Torlo, M. Ricchiuto, Arbitrary high order WENO finite volume scheme with flux globalization for moving equilibria preservation, *J. Sci. Comput.* 96 (2) (2023) 53.
- [31] V. Michel-Dansac, C. Berthon, S. Clain, F. Foucher, A well-balanced scheme for the shallow-water equations with topography or manning friction, *J. Comput. Phys.* 335 (2017) 115–154.

- [32] S. Noelle, N. Pankratz, G. Puppo, J.R. Natvig, Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows, *J. Comput. Phys.* 213 (2) (2006) 474–499.
- [33] R. Abgrall, M. Ricchiuto, Hyperbolic balance laws: Residual distribution, in: *Local and Global Fluxes*, Springer, 2022, pp. 177–222.
- [34] C. Berthon, S. Bulteau, F. Foucher, M. M'Baye, V. Michel-Dansac, A very easy high-order well-balanced reconstruction for hyperbolic systems with source terms, *SIAM J. Sci. Comput.* 44 (4) (2022) A2506–A2535.
- [35] M.J. Castro, C. Parés, Well-balanced high-order finite volume methods for systems of balance laws, *J. Sci. Comput.* 82 (2020) 42.
- [36] M. Dumbser, O. Zanotti, E. Gaburro, I. Peshkov, A well-balanced discontinuous Galerkin method for the first-order Z4 formulation of the Einstein–Euler system, *J. Comput. Phys.* (2024) 112875.
- [37] C. Klingenberg, G. Puppo, M. Semplice, Arbitrary order finite volume well-balanced schemes for the Euler equations with gravity, *SIAM J. Sci. Comput.* 41 (2) (2019) A695–A721.
- [38] M.J. Castro, E. Fernández-Nieto, A class of computationally fast first order finite volume solvers: PVM methods, *SIAM J. Sci. Comput.* 34 (4) (2012) A2173–A2196.
- [39] A. Harten, P.D. Lax, B. Van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Rev.* 25 (1) (1983) 35–61.
- [40] M.J. Castro, T. Morales de Luna, C. Parés, Well-balanced schemes and path-conservative numerical methods, in: *Handbook of Numerical Analysis*, vol. 18, Elsevier, 2017, pp. 131–175.
- [41] A.J.C.B. de Saint-Venant, Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit, *C R Acad Sci* 73 (1871) 147–154.
- [42] P. Glaister, Approximate Riemann solutions of the shallow water equations, *J. Hydraul. Res.* 26 (3) (1988) 293–306.
- [43] A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92, Springer, 2015.
- [44] J.-F. Gerbeau, D. Lombardi, Approximated lax pairs for the reduced order integration of nonlinear evolution equations, *J. Comput. Phys.* 265 (2014) 246–269.