Politecnico di Torino

Politecnico di Torino

Master's Degree in Mathematical Engineering

A.y. 2024/2025

Graduation Session November 2025

Structure-preserving reduced order models for conservation laws with source terms

Supervisors:

Fabio Vicini

Camilla Fiorini

Alessia Del Grosso

Candidate:

Giorgio Musso

Abstract

Numerical simulation of partial differential equations is fundamental for studying complex physical phenomena. However, the high computational cost of high-fidelity methods, such as finite element and finite volume schemes, makes them prohibitively expensive for parametrized problems in multi-query contexts. Reduced Order Models (ROMs) address this issue through dimensionality reduction, maintaining reasonable accuracy. Nevertheless, this promising approach shows significant limitations in the context of hyperbolic conservation laws, where classical ROMs often fail due to the presence of discontinuities and spurious oscillations that generate physically inadmissible values (e.g., negative density or water height).

This work addresses these challenges by introducing an alternative framework named the collocated Reduced Order Model (cROM), which differs from common projection-based model (pROM). We investigate strategies for preserving the structure of conservation laws, with a specific focus on positivity and conservation properties, in the context of ROMs.

To guarantee positivity, the cROM is combined with two transformations that are positivity-preserving by construction: the Logarithmic-Exponential (LE) and the Square-Root (SR). Numerical analysis demonstrates that the SR transformation provides satisfactory results for the linear transport equation and shows strong applicability for the shallow water equations, whereas the LE transformation exhibits limitations that require further investigation.

For the conservation, the introduction of a specific offset on standard cROM allows us to theoretically achieve such property. The numerical validation of this approach exhibits significant improvements for both linear advection and SW equations.

In conclusion, this study demonstrates that a suitable transformation, such as the square-root, can effectively preserve the positivity of physical variables at the numerical level, opening the way for applying these techniques to 2D or 3D problems. The recovery of the conservation property on cROM, while still yielding preliminary results, represents a promising area for future development.

A mia nonna, Pina

Acknowledgements

I would like to express my gratitude to Professor Fabio Vicini for his guidance and for the great availability he has shown me throughout the production of this work.

My sincere thanks go to Camilla Fiorini and Alessia Del Grosso, who welcomed me and made it possible for me to undertake my internship in Paris. They guided me through both the internship itself and the writing of this thesis. They have been incredible mentors, always ready to resolve my doubts and extremely open and kind from the very first day we met.

I also wish to thank the Conservatoire National des Arts et Métiers for providing such a positive and exciting environment over these past months, and the Institut de Mathématiques de Bordeaux for the brief yet intense week of collaboration. A special thanks to all the colleagues in M2N and each person at CNAM who helped me integrate and gradually revealed to me what it means to work in a supportive environment based on constant mutual aid.

Thank you to all my university classmates and friends who have accompanied me on this journey, offering their constant support and help. You have made these years deeply meaningful and filled with wonderful memories.

Finally, infinite thanks to my family, for always being by my side, supporting my every choice, and providing me with immense encouragement.

Table of Contents

List of Figures

Chapter 1

Introduction

Numerical simulation has gained increasingly interest in the fields of engineering and applied sciences. Advances in computational performance have enabled the treatment of complex phenomena modelled by Partial Differential Equations (PDEs). Given the frequent absence of analytical solutions and the practical limitations of real-world experiments, solving PDEs through numerical methods has become essential. These methods have been designed to construct an appropriate discrete formulation that corresponds to the original continuous problem. Their goal is to accurately reproduce the observed phenomenon by approximating the exact solution, which is why they are commonly referred to as high-fidelity (HF) schemes. Numerous HF methods have been developed, such as finite volume schemes, finite element schemes [1, 2, 3], spectral methods [4, 5, 6], and discontinuous Galerkin methods [7, 8, 9]. This manuscript focuses solely on finite volume methods and a specific class of problems: hyperbolic PDEs, which comprehends conservation and balance laws (conservations laws with source terms). The fundamental schemes for these problems are extensively covered in [10], which provides a general introduction to Riemann solvers and numerical methods for fluid dynamics. Further key references include [11, 12], which detail the application of these methods for conservation laws. The primary limitation of HF methods is their prohibitive computational cost, often deriving from systems with an enormous number of degrees of freedom.

To overcome this restriction, Reduced Order Models (ROMs) have been developed. Valid references on ROMs, including their variational formulation and analysis, can be found in [13, 14]. A more general presentation and study is provided in [15]. The general ROM framework typically consists of two main stages: an offline phase and an online phase. The goal of the offline phase is to construct a low-dimensional subspace capable of accurately approximating the HF solutions. This is typically the most computationally expensive part of the process. The most common technique for building this reduced basis is Proper Orthogonal Decomposition (POD) [16, 17]. This method extracts the dominant behaviour

from a collection of snapshots (HF solutions), allowing us to build a basis that retains the most significant information from the original system while achieving a substantial reduction in its dimensionality.

In the subsequent online phase, the goal is to efficiently compute the coefficients for the reduced basis such that the solution can be represented as a linear combination of the basis functions. These coefficients are typically obtained at a low computational cost, often using a Petrov-Galerkin projection. In this method, the test subspace may differ from the trial subspace. If they coincide, the method is referred to as a Galerkin projection [18, 19]. An alternative approach involves residual minimization [20].

In this work, we primarily focus on the projection-based Reduced Order Model (pROM) approach. Furthermore, we introduce an additional layer of approximation by considering a collocated ROM (cROM) framework, as described in [21].

At the continuous level, the governing equations are characterized by physical properties, such as the conservation of fundamental variables, the preservation of their positivity, and the satisfaction of entropy stability conditions. Maintaining these properties at the discrete level is crucial for the numerical solution to be mathematically consistent and physically meaningful. Consequently, numerous HF methods have been designed specifically to enforce these constraints. The preservation of positivity, for instance, is essential in problems where obtaining negative values for physical quantities like density or water height is inadmissible. Significant research has been dedicated to this goal. One approach is presented by [22], who proposed a positivity-preserving scheme for ordinary differential equations. Another key contribution is the work of [23], which introduces a scheme that maintains both positivity and the well-balanced property (a condition where stationary solutions are preserved over time) for shallow water equations.

Finite volume schemes are inherently conservative by construction. They are derived from the integral form of PDEs, and their formulation ensures that the conservation property is preserved at the discrete level [10, 11, 12]. However, classical ROMs often struggle with advection-dominated problems characterized by non-smooth solutions and discontinuities. Various strategies have been therefore provided to enforce physical properties like positivity. For instance, [24] introduced techniques designed to preserve positivity in cross-diffusion systems. Referring to this work, we analyze the variable transformation they proposed within the cROM framework and introduce an alternative transformation which, to the best of our knowledge, has not yet been explored. Regarding conservation, relevant approaches include the works of [25], who developed a ROM scheme based on the conservative structure of finite volume methods for hyperbolic laws, and [26], who proposed a conservative ROM for fluid flows. In the context of cROMs, this work proposes an approach aimed at conserving the global quantity of the system.

1.1 Structure of the manuscript

The structure of this work is therefore divided into the following chapters, excluding this introductory chapter and the final chapter containing the conclusions and future perspectives:

- **Chapter 2**: hyperbolic PDEs and their main features are presented mathematically, with a particular focus on conservation laws with source terms, where the absence of this term constitutes classical conservation laws, while its introduction constitutes balance laws;
- **Chapter 3**: this chapter analyses the essential framework of ROMs, introducing key concept such as Proper Orthogonal Decomposition. It first presents the classical projection-based ROM (pROM) formulation before detailing the derivation and implementation of collocated ROM (cROM).
- **Chapter 4**: this chapter focuses on positivity. Numerous tests are carried out on linear transport equation and shallow water equations, presenting the positivity-preserving approaches adopted in combination with cROM.
- **Chapter 5**: this chapter is related to conservation. We present numerical tests that incorporate a specific formulation designed to maintain the global conservation of the system. Two distinct cases are studied: one where the conserved quantity has a constant integral over the domain, and another where this integral varies in time.

Chapter 2

Hyperbolic Partial Differential Equations

In this section, we introduce the class of hyperbolic Partial Differential Equations (PDEs), with a particular focus on conservation laws. The presentation follows the framework and definitions provided by Toro in [10]. We restrict our discussion to the one-dimensional case, even though the framework can be extended to multi-dimensional problems.

2.1 Hyperbolic PDEs

We define a first order PDEs system such as

$$\frac{\partial u_i}{\partial t} + \sum_{j=1}^{m} a_{ij}(x, t, u_1, ..., u_m)\frac{\partial u_j}{\partial x} + b_i(x, t, u_1, ..., u_m) = 0, \quad i = 1, ..., m. \tag{2.1}$$

This system consists of m equations in m unknowns u_i which depend on the space variable x and the time variable t. These two last variables are the *independent variables*, whereas the u_i are the *dependent variables*. The dependence is expressed via the notation $u_i = u_i(x, t)$, in which $\partial u_i/\partial t$ and $\partial u_i/\partial x$ denote, respectively, the partial derivative of $u_i(x, t)$ with respect to t and x. System (2.1) can be compactly written in matrix form:

$$\mathbf{U_t} + \mathbf{A}\mathbf{U_x} + \mathbf{B} = 0, \tag{2.2}$$

with

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ a_{21} & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{bmatrix}.$$

The system (2.2) is *linear with constant coefficients* if the entries a_{ij} of the matrix $\mathbf{A}$ and the components b_i of the vector $\mathbf{B}$ are both constant. The system is *linear with variable coefficients* if $a_{ij} = a_{ij}(x,t)$ and $b_i = b_i(x,t)$. The system is still linear if $\mathbf{B}$ depends linearly on $\mathbf{U}$ and is called *quasi-linear* if the coefficient matrix $\mathbf{A}$ is a function of the vector $\mathbf{U}$, that is $\mathbf{A} = \mathbf{A}(\mathbf{U})$. Note that quasi-linear systems are typically systems of nonlinear equations. System (2.2) is called *homogeneous* if $\mathbf{B} = \mathbf{0}$.

For a set of PDEs of the form (2.2), the range of variation of the independent variables x and t needs to be specified. Usually x lies in a subinterval of $\mathbb{R}$, namely $x_L \leq x \leq x_R$; this subinterval is called the *spatial domain* of the PDEs, or simply *domain*. We may need to impose *Boundary Conditions (BCs)* at the values x_L, x_R. Moreover, at the initial time one needs to specify an *Initial Condition (IC)*, normally chosen to be at $t^0 = 0$.

Two scalar ($m = 1$) examples of PDEs of the form (2.1) are given by the *linear advection equation*

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{2.3}$$

and the *inviscid Burgers equation*

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{2.4}$$

In both equations, the conservative variable is the transported quantity u, while the coefficient a and the value $a(u) = u$ represent, respectively, the wave propagation speed.

As mentioned, we will first focus on conservation laws and later we extend the definition to general case of balance laws.

Definition 2.1 (Conservation Laws) *Conservation laws are systems of partial differential equations that can be written in the form*

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0}, \tag{2.5}$$

where

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix},$$

$\mathbf{U}$ *is the vector of conserved variables,* $\mathbf{F} = \mathbf{F}(\mathbf{U})$ *is called the vector of physical fluxes and each of its components* f_i *and is a function of the components* u_j *of* $\mathbf{U}$*.*

Definition 2.2 (Jacobian Matrix) *The Jacobian of the flux function* $\mathbf{F}(\mathbf{U})$ *in equation (2.5) is the matrix*

$$\mathbf{A}(\mathbf{U}) = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \cdots & \frac{\partial f_2}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial u_1} & \cdots & \frac{\partial f_m}{\partial u_m} \end{bmatrix}. \tag{2.6}$$

The entries a_{ij} *of* $\mathbf{A}(\mathbf{U})$ *are partial derivatives of the components* f_i *of the vector* $\mathbf{F}$ *with respect to the components* u_j *of the vector of conserved variables* $\mathbf{U}$*, that is,*

$$a_{ij} = \frac{\partial f_i}{\partial u_j}.$$

Note that one can write in quasi-linear form (2.2), with $\mathbf{B} \equiv 0$, a conservation laws of the form (2.5) by applying the chain rule to the second term in system (2.5), namely

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x}.$$

Hence system (2.5) becomes

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_\mathbf{x} = \mathbf{0},$$

which is a specific case of system (2.2). Thanks to the above observations, the scalar PDEs (2.3) and (2.4) can be reformulated as conservation laws, namely

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad f(u) = au, \tag{2.7}$$

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad f(u) = \frac{1}{2}u^2. \tag{2.8}$$

To study conservation laws, it is necessary to introduce two fundamental concepts: eigenvalues and eigenvectors.

Definition 2.3 (Eigenvalues) *The eigenvalues* λ_i *of a matrix* $\mathbf{A}$ *are the solutions of the characteristic polynomial*

$$|\mathbf{A} - \lambda \mathbf{I}| = det(\mathbf{A} - \lambda \mathbf{I}) = 0,$$

where $\mathbf{I}$ *is the identity matrix. The eigenvalues of the coefficient matrix* $\mathbf{A}$ *of a system of the form (2.5) are called the eigenvalues of the system.*

Physically, eigenvalues represent speeds of propagation of information. Speeds will be measured positive in the direction of increasing x and negative otherwise.

Definition 2.4 (Eigenvectors) *A right eigenvector of a matrix* $\mathbf{A}$ *corresponding to an eigenvalue* λ_i *of* $\mathbf{A}$ *is a vector* $\mathbf{K}^{(i)} = [k_1^{(i)}, k_2^{(i)}, ..., k_m^{(i)}]^T$ *satisfying* $\mathbf{A}\mathbf{K}^{(i)} = \lambda_i \mathbf{K}^{(i)}$. *Similarly, a left eigenvector of a matrix* $\mathbf{A}$ *corresponding to an eigenvalue* λ_i *of* $\mathbf{A}$ *is a vector* $\mathbf{L}^{(i)} = [l_1^{(i)}, l_2^{(i)}, ..., l_m^{(i)}]^T$ *such that* $\mathbf{L}^{(i)}\mathbf{A} = \lambda_i \mathbf{L}^{(i)}$.

For the examples (2.7)-(2.8), the eigenvalues are simply found to be $\lambda = a$ and $\lambda = u$ respectively.

Now we have all the ingredients to formally define the hyperbolicity concept.

Definition 2.5 (Hyperbolic System) *A system (2.5) is said to be hyperbolic at a point (x,t) if* $\mathbf{A}$ *has m real eigenvalues* $\lambda_1, ..., \lambda_m$ *and a corresponding set of m linearly independent right eigenvectors* $\mathbf{K}^{(1)}, ..., \mathbf{K}^{(m)}$. *The system is said to be strictly hyperbolic if the eigenvalues* λ_i *are all distinct.*

Note that strict hyperbolicity implies hyperbolicity, because real and distinct eigenvalues ensure the existence of a set of linearly independent eigenvectors. The system (2.5) is said to be *elliptic* at a point (x, t) if none of the the eigenvalues λ_i of $\mathbf{A}$ are real. Both scalar examples (2.7)-(2.8) are trivially hyperbolic.

Before delving deeper into the topic of conservation laws and describing some essential properties, we present a more general definition of conservation laws introducing additionally a source term in the form (2.5). Formally, this definition broadens the class of conservation laws, which constitute a special case of the so-called balance laws.

Definition 2.6 (Balance laws) *Balance laws are systems of partial differential equations that can be written in the form*

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{S}(\mathbf{U}), \tag{2.9}$$

where

$$\mathbf{S}(\mathbf{U}) = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix},$$

$\mathbf{S} = \mathbf{S}(\mathbf{U})$ *is the vector of source terms and each component* s_i *is a function of the components* u_j *of* $\mathbf{U}$. *If this vector vanishes,* $\mathbf{S}(\mathbf{U}) \equiv \mathbf{0}$, *the system (2.9) reduces to a conservation law.*

A nonlinear system ($m = 2$) example of balance laws is provided by the *shallow water (SW) equations*

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial}{\partial x}(hu) = 0, \\ \dfrac{\partial (hu)}{\partial t} + \dfrac{\partial}{\partial x}\Big(hu^2 + \frac{1}{2}gh^2\Big) = -gh\,\partial_x z, \end{cases} \tag{2.10}$$

where $z(x)$ is the bottom topography, g is the gravitational acceleration, and $h(x,t)$ and $u(x,t)$ refer to the water height and the water velocity respectively. The system of equations can be compactly written in the form (2.9) with

$$\mathbf{U} = \begin{bmatrix} h \\ hu \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} hu \\ hu^2 + 1/2gh^2 \end{bmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{bmatrix} 0 \\ -ghz_x \end{bmatrix}. \tag{2.11}$$

This model is also hyperbolic, with eigenvalues given by $\lambda_{1,2} = u \pm \sqrt{gh}$. Note that the water height $h(x,t)$ must be nonnegative: $h = 0$ corresponds to a dry state, while $h < 0$ has no physical meaning. Mathematically, negative water heights would result in undefined values due to the square root term $\sqrt{gh}$. In the case $h = 0$, the velocity is physically meaningless since it cannot be defined in the absence of water. A common approach is therefore to impose $u = 0$ in dry states.

In addition, the system (2.10) admits the steady state known as the *lake at rest*:

$$z(x) + h(x,t) = \text{constant}, \quad u(x,t) = 0, \tag{2.12}$$

which represents an equilibrium configuration. This condition will be discussed in detail in the next chapters since it can lead to numerical issues.

2.2 Conservation Laws

The purpose of this section is to draw attention to some mathematical properties of hyperbolic conservation laws. We restrict our attention to those properties considered necessary for the development and application of numerical methods for conservation laws. In the previous section, we introduced the formal definition of a system of m conservation laws

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0}.$$

According to the previous section, the system is assumed to be hyperbolic with eigenvalues and eigenvectors ordered as

$$\lambda_1(\mathbf{U}) < \lambda_1(\mathbf{U}) <, ..., < \lambda_m(\mathbf{U}),$$

$$\mathbf{K}^{(1)}(\mathbf{U}), \mathbf{K}^{(2)}(\mathbf{U}), ..., \mathbf{K}^{(m)}(\mathbf{U}).$$

It is essential to note that now eigenvalues and eigenvectors depend on $\mathbf{U}$, even though sometimes the argument $\mathbf{U}$ can be omitted. We now focus on describing how to derive the integral and the differential form of conservation laws, taking as references [10], [11] and [12].

2.2.1 Integral Forms of Conservation Laws

To show how conservation laws derive from physical principles, we will begin by evaluating one of the simplest fluid dynamics problem: a gas or fluid that flows through a one-dimensional pipe with some known velocity $v(x,t)$. Let $\rho(x,t)$ be the concentration or the density of some chemical tracer present in this fluid; $\rho(x,t)$ is the unknown function that we wish to determine. The density is typically measured in units of mass per unit volume, but, since we are studying a one-dimensional problem, it is reasonable to assume that $\rho(x,t)$ is measured in units of mass per unit length instead.

Let us consider a given section from x_L to x_R, then

$$\int_{x_L}^{x_R} \rho(x,t)dx \tag{2.13}$$

represents the total mass of the tracer in that section at time t. If we assume that the walls of the pipe are impermeable and the mass is neither created nor destroyed, then the mass in this interval can change only because of gas flowing across the endpoints x_L or x_R. Let $F_i(t)$ be the rate at which the tracer flows past the extreme points x_i for $i = L, R$. Using the common convention where $F_i(t) > 0$ corresponds to rightward flux and $F_i(t) < 0$ indicates leftward flux, we have

$$\frac{d}{dt}\int_{x_L}^{x_R} \rho(x,t)dx = F_L(t) - F_R(t). \tag{2.14}$$

This is one formulation of the conservation law in **integral form**. This relation is the basis of *conservation*, namely the rate of change of the total mass only depends on fluxes through the endpoints. To proceed further, we need to establish the relation between the flux functions $F_i(t)$ and the density $\rho(x,t)$. In the case of fluid described previously, the rate of flow or flux of gas past at any point x and time t is given by

$$F(x,t) = \rho(x,t)v(x,t). \tag{2.15}$$

From initial assumptions $v(x,t)$ is a known function, so we can write

$$F(x,t) = f(\rho, x, t) = v(x,t)\rho. \tag{2.16}$$

Moreover, if $v(x,t) = \bar{v}$ is a constant, so it is independent of x and t, we have

$$F(x,t) = f(\rho, x, t) = \bar{v}\rho. \tag{2.17}$$

The equation is called *autonomous* since in this case the flux at any point and time can be determined from the value of the conserved quantity at that point.

For a general autonomous flux $f(\rho)$ which depends only on the variable ρ, it is possible to recast the conservation law (2.14) as

$$\frac{d}{dt}\int_{x_L}^{x_R} \rho(x,t)dx = f(\rho(x_L,t)) - f(\rho(x_R,t)). \tag{2.18}$$

Integrating over any time interval $[t', t'']$, this finally yields to

$$\int_{x_L}^{x_R} \rho(x,t'')dx = \int_{x_L}^{x_R} \rho(x,t')dx + \int_{t'}^{t''} f(\rho(x_L,t))dt - \int_{t'}^{t''} f(\rho(x_R,t))dt. \tag{2.19}$$

To derive the differential form of the conservation law, we must assume that $\rho(x,t)$ and $v(x,t)$ are differentiable functions. This smoothness hypothesis is very important to keep in mind when we face nonsmooth solution to these equations. Based on this assumptions, the equation (2.18) can be rewritten as

$$\int_{x_L}^{x_R} \frac{\partial}{\partial t}\rho(x,t)dx = -\int_{x_L}^{x_R} \frac{\partial}{\partial x} f(\rho(x,t))dx. \tag{2.20}$$

or, with further step, as

$$\int_{x_L}^{x_R} \left[\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x} f(\rho(x,t))\right] dx = 0. \tag{2.21}$$

Since this must hold for any interval $[x_L, x_R]$, we can conclude that the integrand in (2.21) must be necessarily equal to zero. This finally leads to the differential equation

$$\rho_t + f(\rho)_x = 0. \tag{2.22}$$

This is the desired **differential form** of the conservation laws.

Remark. In this section, we considered the density $\rho(x,t)$ as conserved variable for all the equations. The argument, however, is completely general and can be applied to any conserved quantity. From now on, we will use the generic notation $u(x,t)$ to denote the conserved variable of interest.

2.2.2 Classical and Weak Solutions

The definition of weak solution is relevant since in practice many interesting solutions are not smooth or not continuously differentiable, but contain discontinuities such

as shock waves. Nonlinear conservations laws typically exhibit discontinuities that can easily develop spontaneously even from smooth initial data. If a discontinuity appears in u, then the partial differential equation (2.22) does not hold in the classical sense, whereas the form (2.14) still holds.

Let us begin by introducing what is meant by a solution in the classical sense, giving its definition

Definition 2.7 (Classical Solution) *A function $u(x,t)$ is a classical solution of the conservation laws if equation (2.22) holds for all x and t over the domain.*

To motivate the weak form, we start by supposing that $u(x,t)$ is a smooth function. Integrating the equation (2.21) in time between two selected times t' and t'', we obtain

$$\int_{t'}^{t''} \int_{x_L}^{x_R} \Big[u_t + f(u)_x \Big] dxdt = 0. \tag{2.23}$$

Instead of considering this latter integral for arbitrary choices of x_L, x_R, t' and t'', we can select, without loss of generality,

$$\int_0^{\infty} \int_{-\infty}^{\infty} \Big[u_t + f(u)_x \Big] \phi(x,t) dxdt = 0, \tag{2.24}$$

for a certain class of functions $\phi(x,t)$. Note that if we chose $\phi(x,t)$ such that

$$\phi(x,t) = \begin{cases} 1, & \text{if } (x,t) = [x_L, x_R] \times [t', t''] =: \Omega, \\ 0, & \text{otherwise,} \end{cases} \tag{2.25}$$

then the integral reduces to the previous one in formula (2.23). We can extend this notion considering $\phi(x,t)$ any function that has compact support, meaning it is identically equal to zero outside of some bounded region of $x-t$ plane.

Assuming that $\phi(x,t)$ is now a smooth function, we can integrate by parts in (2.24) and obtain

$$\int_0^{\infty} \int_{-\infty}^{\infty} \Big[u\phi_t + f(u)\phi_x \Big] dxdt = -\int_0^{\infty} u(x,0)\phi(x,0) dx. \tag{2.26}$$

A relevant feature of (2.26) is that the derivatives moved on ϕ, and no longer involve u and $f(u)$. Even if u is discontinuous, the relation (2.26) is still valid. This motivates the following definition

Definition 2.8 *The function $u(x,t)$ is a weak solution of the conservation law (2.22) with given initial data $u(x,t)$ if relation (2.26) holds for all functions ϕ in $C_c^1(\Omega)$.*

The functions space $C_c^1(\Omega)$ includes the set of all functions that are $C^1(\Omega)$ (continuously differentiable) and have compact support. It can be shown that the integral conservation law is also satisfied by any weak solution and vice versa. See [11] and [12] for further discussions.

2.3 Numerical Methods

Our interest is the application of numerical methods for solving partial differential equations (PDEs). Numerical methods replace the *continuous* problem represented by the PDEs by a *finite* set of *discrete* values. These are obtained by first discretizing the domain of the PDEs into a finite set of points or volumes via a mesh or grid. The corresponding discretization of the PDEs on the grid results in discrete values. In the **Finite Difference approach** one regards these values as *point values* defined at grid points. The **Finite Volume approach** regards these discrete values as *averages over finite volumes*. We are mostly interested in the second one for the purpose of the forthcoming chapters.

2.3.1 Finite Volume Method

To introduce the finite volumes method, we begin by taking into account an initial-values problem, also known as *Cauchy problem*, in one-dimensional domain,

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\,\dfrac{\partial u}{\partial x} = 0, & x \in (x_L, x_R),\ t \in (0, T], \\ u(x,0) = u_0(x), & x \in [x_L, x_R]. \end{cases} \tag{2.27}$$

in which $f(u) = au$. By recalling the formula in (2.19) and adjusting it for our problem, we obtain

$$\int_{x_L}^{x_R} u(x,t'')dx = \int_{x_L}^{x_R} u(x,t')dx + \int_{t'}^{t''} f(u(x_L,t))dt - \int_{t'}^{t''} f(u(x_R,t))dt. \tag{2.28}$$

We discretize the $x-t$ plane by choosing a **mesh width** $\Delta x > 0$ and a **time step** $\Delta t > 0$ and define the discrete mesh points (x_j, t^n) by

$$\begin{aligned} x_j &= j\Delta x, \qquad j = 0,1,\ldots,N \\ t^n &= n\Delta t, \qquad n = 0,1,\ldots,N_T, \end{aligned}$$

where $x_0 = x_L$, $x_N = x_R$ and $T = N_T\Delta t$. In addition, we define the interface points as follows:

$$x_{j+1/2} = x_j + \frac{\Delta x}{2} = \left(j + \frac{1}{2}\right)\Delta x.$$

For the sake of simplicity, we take a uniform mesh that does not change over time, although most of the numerical methods can be extended to spatially and temporally variable meshes. After performing this discretization, we can rewrite

the formula (2.28) for each cell $\mathcal{V}_j = [x_{j-1/2}, x_{j+1/2}]$ and $t' = t^n$, $t'' = t^{n+1}$, obtaining

$$\begin{aligned}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t^{n+1})dx = &\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t^n)dx \\ &-\left[\int_{t^n}^{t^{n+1}} f(u(x_{j+1/2},t))dt - \int_{t^n}^{t^{n+1}} f(u(x_{j-1/2},t))dt\right],\end{aligned} \tag{2.29}$$

Let us now introduce two essential definitions as the cell average of the exact solution $u(x,t^n)$, defining by

$$u_j^n = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t^n)dx, \tag{2.30}$$

and the average of the flux at the interface $x_{j+1/2}$ over the time interval Δt as

$$F_{j+1/2}^n = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{j+1/2}),t)dt. \tag{2.31}$$

The integral form of the conservation law then gives us the exact evolution equation for the cell averages:

$$\Delta x u_j^{n+1} = \Delta x u_j^n - \Delta t(F_{j+1/2}^n - F_{j-1/2}^n). \tag{2.32}$$

A finite volume numerical method is directly based on equation (2.32). The core of the method is approximate the quantity $F_{j+1/2}^n$ along $x = x_{j+1/2}$ through *numerical fluxes* $f_{j+1/2}^n$

$$f_{j+1/2}^n \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{j+1/2}),t)dt. \tag{2.33}$$

Each numerical method is defined by a particular choice of this numerical flux. For a hyperbolic problem information propagates with finite speed, so it is reasonable to first suppose that we can obtain $f_{j+1/2}^n$ based only on cell averages u_j^n and u_{j+1}^n on either side of the interface $x_{j+1/2}$, namely

$$f_{j+1/2}^n = \mathcal{F}(u_j^n, u_{j+1}^n). \tag{2.34}$$

$\mathcal{F}(u,v)$ is a function that depend on two real variables u and v.

Once the numerical flux chosen, the numerical scheme is given by

$$\Delta x\, u_j^{n+1} = \Delta x\, u_j^n - \Delta t\,(f_{j+1/2}^n - f_{j-1/2}^n); \tag{2.35}$$

introducing the constant

$$\lambda = \frac{\Delta t}{\Delta x} > 0, \tag{2.36}$$

the **finite volume method** can be written as

$$u_j^{n+1} = u_j^n - \lambda \, (f_{j+1/2}^n - f_{j-1/2}^n), \tag{2.37}$$

or, including the dependency in the fluxes, as

$$u_j^{n+1} = u_j^n - \lambda \left[f_{j+1/2}^n(u_j^n, u_{j+1}^n) - f_{j-1/2}^n(u_{j-1}^n, u_j^n) \right]. \tag{2.38}$$

In particular, this last equation emphasizes the fact that this type-method is an *explicit method with a three-point stencil*, since u_j^{n+1} depends on u_{j-1}^n, u_j^n and u_{j+1}^n.

Let us now briefly introduce some notable methods:

- **Lax-Friedrichs method**

  The numerical flux $\mathcal{F}(u, v)$ is given by

  $$\mathcal{F}_{LF}(u, v) = \frac{a}{2}(u + v) + \frac{1}{2\lambda}(u - v), \tag{2.39}$$

  and the resulting scheme is

  $$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{1}{2}\lambda a(u_{j+1}^n - u_{j-1}^n). \tag{2.40}$$

- **Upwind method**

  The numerical flux is given by

  $$\mathcal{F}_U(u, v) = \frac{a}{2}(u + v) + \frac{|a|}{2}(u - v) = \begin{cases} au & \text{if } a > 0, \\ av & \text{if } a < 0. \end{cases} \tag{2.41}$$

  and the resulting scheme

  $$\begin{aligned} u_j^{n+1} &= u_j^n - \frac{1}{2}\lambda a(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\lambda|a|(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \\ &= \begin{cases} u_j^n - \lambda a(u_j^n - u_{j-1}^n) & \text{if } a > 0, \\ u_j^n - \lambda a(u_{j+1}^n - u_j^n) & \text{if } a < 0. \end{cases} \end{aligned} \tag{2.42}$$

- **Lax-Wendroff method**

  The numerical flux $\mathcal{F}(u, v)$ is given by

  $$\mathcal{F}_{LW}(u, v) = \frac{a}{2}(u + v) + \frac{1}{2}\lambda a^2(u - v), \tag{2.43}$$

  and the resulting scheme is

  $$u_j^{n+1} = u_j^n - \frac{1}{2}\lambda a(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}(\lambda a)^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \tag{2.44}$$

For the next chapters, we will refer to a fully discretized scheme (2.37) under the following compact form

$$u_j^{n+1} = \mathcal{L}(u_{j-1}^n, u_j^n, u_{j+1}^n), \tag{2.45}$$

where $\mathcal{L}$ is an operator which depends on the solution at the previous time step u^n and the selected discretization for the numerical flux.

2.3.2 Convergence

Let us now introduce some fundamental concepts concerning the convergence of numerical methods. These concepts certainly deserve more in-depth and specific treatment (see [10], [12]), but they are not the main focus of this work. We will therefore limit ourselves to providing some simple definitions that will be useful for the following discussions. One essential requirement is that the resulting method should be convergent, i.e., the numerical solution should converge to the true solution of the differential equation as the grid is refined (as $\Delta x, \Delta t \to 0$). This generally requires two conditions:

- **Consistency**: the method approximates the differential equation well locally;
- **Stability**: the method produces small errors at each time step that do not blow-up or grow too rapidly.

Consistency

The integral in (2.33) should be approximated by the numerical flux. In particular, if the function $u(x,t) \equiv \bar{u}$ is constant in x, then u will not vary in time and the integral in (2.33) simply reduces to $f(\bar{u})$. As a result, if $u_{j-1}^n = u_j^n = \bar{u}$, then we expect the numerical flux function $\mathcal{F}$ of (2.34) to reduce to $f(\bar{u})$, so we require

$$\mathcal{F}(\bar{u}, \bar{u}) = f(\bar{u}) \tag{2.46}$$

for any value $\bar{u}$. This is part of the basic **consistency condition**. It is easy to verify that all the methods introduced above satisfy this condition and are therefore consistent.

Stability

In order to properly describe the concept of stability, it is important to first define what is meant by the domain of dependence of a PDE.

Let $(\bar{x}, \bar{t})$ be some fixed point in space–time. The solution $u(\bar{x}, \bar{t})$ is affected only by the initial data u_0 at m particular points $\bar{x} - \lambda_p \bar{t}$ for $p = 1, 2, \ldots, m$. This set of points,

$$\mathcal{D}(\bar{x}, \bar{t}) = \{\bar{x} - \lambda_p \bar{t} : p = 1,2, \ldots, m\} \tag{2.47}$$

is called the *domain of dependence* of the point $(\bar{x}, \bar{t})$ (see Figure 2.1). The numerical domain of dependence, $\mathcal{D}_k(\bar{x}, \bar{t})$, for a particular method is similarly defined. It is the set of points x for which the numerical solution $u(\bar{x}, \bar{t})$ depends on the initial data $u_0(x)$.

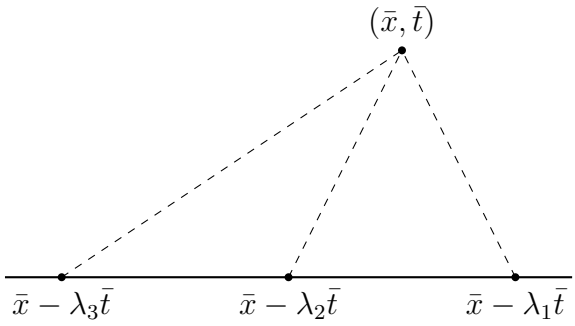

Figure 2.1: Domain of dependence of the point $(\bar{x}, \bar{t})$ for a general hyperbolic system of three equations with eigenvalues $\lambda_1 < 0 < \lambda_2 < \lambda_3$.

In 1928, Courant, Friedrichs and Lewy recognized a necessary stability condition for any numerical method.This one was named CFL condition in honour of them and it states as follows:

Definition 2.9 (CFL condition) *A numerical method can be stable only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as Δt and Δx go to zero.*

It is crucial to note that the CFL condition is only a necessary condition for stability, and indeed it is not always sufficient to guarantee stability. However, for the linear methods we have previously introduced (such as the Upwind, Lax-Friedrichs, and Lax-Wendroff schemes), it can be mathematically proved that the CFL condition is also sufficient for stability in appropriate norms.

For the linear advection equation (2.27), let us introduce the *Courant number*

$$\text{Cour} = |a|\lambda = |a|\frac{\Delta t}{\Delta x}. \tag{2.48}$$

The CFL condition simply results in $\text{Cour} \leq 1$, namely

$$\Delta t \leq \frac{\Delta x}{|a|}. \tag{2.49}$$

In practice, the Courant number is set such that $0 < \text{Cour} \leq 1$ and, after choosing the mesh width Δx, we can determine the time step Δt through the following expression

$$\Delta t = \frac{\text{Cour}}{|a|}\Delta x. \tag{2.50}$$

In this way, the CFL condition is automatically satisfied.

If the coefficient a is not constant, the Courant number will be defined as

$$\text{Cour}^n = \max_j |a_{j+1/2}|\lambda.$$

Now it depends on the current time step and the CFL condition is given by $\text{Cour}^n \leq 1$ for each $n \geq 0$.

2.3.3 Conservation property

As mentioned above, if we are studying a substance that is neither created nor destroyed within a given section, then the total mass within this section can change only due to the flux or flow of particles through the endpoints of the section at x_L and x_R. For clarity, let us recall the formula (2.14), which expresses the concept of conservation of physical variables

$$\frac{d}{dt}\int_{x_L}^{x_R} u(x,t)dx = F_L(t) - F_R(t).$$

The same conservation property that we expressed in the integral continuous form finds a counterpart in the discrete scheme: the finite volume method (2.37) is in fact built in order to preserve the total quantity exactly, as we show below. This property is related to its formulation since it is written in **conservation form**. In fact, note that summing $u_j^{n+1}\Delta x$ over N cells, we have

$$\sum_{j=1}^{N} u_j^{n+1}\Delta x = \sum_{j=1}^{N} u_j^n \Delta x - \sum_{j=1}^{N} \frac{\Delta t}{\Delta x}\Big(f_{j+1/2}^n - f_{j-1/2}^n\Big)\Delta x.$$

The sum of the flux differences leads to cancel all the terms except for fluxes at the extreme edges

$$\sum_{j=1}^{N} \frac{\Delta t}{\Delta x}\Big(f_{j+1/2}^n - f_{j-1/2}^n\Big)\Delta x = \Delta t\Big(f_{N+1/2}^n - f_{1-1/2}^n\Big),$$

where the boundary conditions have to be imposed, resulting in the final expression

$$\sum_{j=1}^{N} u_j^{n+1}\Delta x = \sum_{j=1}^{N} u_j^n \Delta x + \Delta t\Big(f_{N+1/2}^n - f_{1-1/2}^n\Big). \tag{2.51}$$

We can guarantee that the numerical method is conservative in a way that mimics the true solution. This is because $\sum_{j=1}^{N} u_j^n \Delta x$ represents the integral of u over the entire interval $[x_L, x_R]$, and if we use a method that is in conservation

form, then this discrete sum will evolve only due to fluxes at the boundaries $x = x_L$ and $x = x_R$.

If the inflow and the outflow coincide, we can also eliminate the last term in (2.51) yielding to

$$\sum_{j=1}^{N} u_j^{n+1} \Delta x = \sum_{j=1}^{N} u_j^n \Delta x. \tag{2.52}$$

This latter equation shows that the integral of u at time $n+1$ is equal to the integral of the same at the previous step n.

To conclude, the total "mass" within the computational domain will be therefore preserved, or at least will vary correctly if the boundary conditions are properly imposed.

2.3.4 Positivity preservation

The preservation of positivity plays a crucial role in the study of conservation laws, in particular for equations involving physical quantities that do not allow negative values, such as water height, pressure, and density. Mathematically speaking, the models used to study these quantities have been designed to verify this property on a continuous basis. The same goal is pursued by numerical methods, including finite volume methods; as we have seen, the latter depend strictly on the choice of numerical flux and this choice should be done in order to preserve the physical feature of positivity for conserved variables.

For a numerical method, this definition can be express as

Definition 2.10 (Positivity preservation) *Assume $u_j^0 \geq 0$, $\forall j$, then the positivity preservation is satisfied only if*

$$u_j^n \geq 0, \ \forall j, \ n = 1, ..., N_T.$$

To present the concept, we consider the linear advection equation (2.27) with the initial condition

$$u(x,0) = u_0(x) = \begin{cases} 1, & x \in [0.25,0.5], \\ 0, & \text{otherwise.} \end{cases}$$

In this simple test case, we consider an initial condition $u_0 \geq 0$ and we also know the analytical solution of the equation (2.7) which is simply given by a translation of the initial profile, namely

$$u(x,t) = u_0(x - at). \tag{2.53}$$

From the expression (2.53), note that the solution remains greater than or equal to zero over time since the positivity of u is closely linked to the positivity of the initial

condition chosen for our problem. In more complex problems such as the shallow water system (2.10), the question of positivity is closely related to the physics of the problem instead. Negative values of water height h do not correspond to any real state and are therefore not physically admissible.

The wave propagation speed is set to $a = 1/2$ and the Courant number is chosen as $\text{Cour} = 0.8 \leq 1$. The spatial discretization uses a mesh width $\Delta x = 1/128$, while the time step Δt is selected according to the CFL stability condition (2.50).

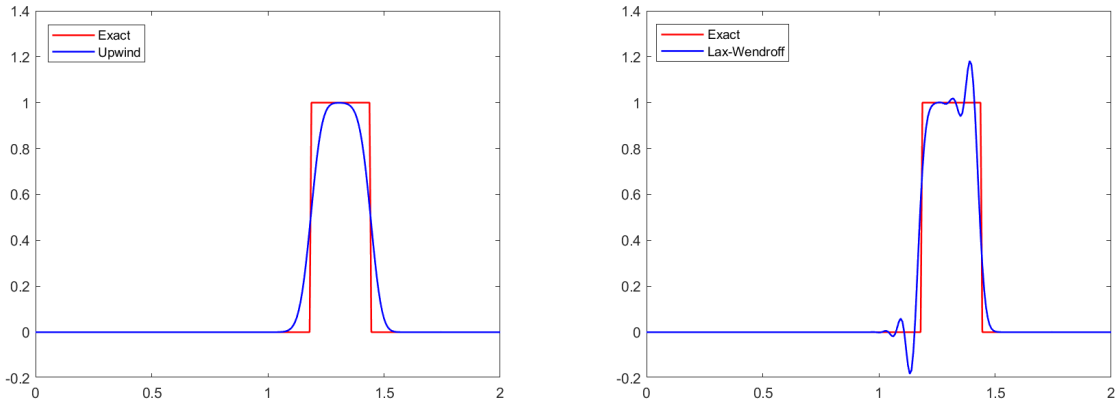

Figure 2.2: Comparison between the exact solution and the numerical approximation with the Upwind scheme (left) and the Lax–Wendroff scheme (right).

Figure 2.2 highlights the different behaviour of the two schemes. The Upwind method does not produce negative values and thus preserves positivity, although the profile suffers from numerical diffusion. In contrast, the Lax–Wendroff scheme yields non-physical spurious oscillations nearby the discontinuities, which in turn lead to non-negative values of the approximate solution u.

However, this elementary test emphasizes that positivity preservation is not automatically guaranteed for all the methods and a fundamental role is played by the choice of the numerical flux.

Chapter 3

Reduced Order Models

3.1 Introduction

A wide range of physical problems can be described by parametrized PDEs. To obtain their numerical solutions, high-fidelity discretization techniques such as the FV method, presented above, are typically employed. However, these methods are often computationally expensive, particularly when applied to large-scale systems or multi-query simulations. To overcome this issue, Reduced Order Models (ROMs) have been developed in recent years. This approach allows for a significant reduction in computational cost while maintaining a balanced level of accuracy.

The ROM construction process is divided into two main stages:

- **Offline (training) phase**: A set of HF solutions (snapshots) is computed for selected parameter values. These snapshots are used to capture the general behaviour of the system and to build a low-dimensional trial subspace that accurately represents the HF solutions.
- **Online (prediction) phase**: For new parameter instances, the solution is rapidly computed by exploiting the information gathered in the previous stage.

In this chapter, we first introduce the mathematical framework of parametrized PDEs and the concept of solution manifold. We then analyze the Proper Orthogonal Decomposition (POD) technique for computing the reduced basis. Finally, we compare the classic projection ROM (pROM) and the more recent collocated ROM (cROM) technique.

3.2 Parametrized PDEs

Let $\mathbb{P} \subset \mathbb{R}^p$ be a closed and bounded *parametric space* with $p \in \mathbb{N}^*$, and let $\Omega \subset \mathbb{R}^d$ be the spatial domain with $d \in \{1,2,3\}$. In this work, we restrict our attention to one-dimensional problems; thus, the spatial domain corresponds to a open interval $\Omega = (x_L, x_R)$.

We consider a parametric, time-dependent PDE of the form [15]:

$$\frac{\partial}{\partial t} u(x,t;\vartheta) + \mathcal{G}[u(x,t;\vartheta)] = 0, \quad x \in \Omega,\ t \in \mathbb{R}^*_+,\ \vartheta \in \mathbb{P}, \tag{3.1}$$

where x denotes the spatial coordinate, t the time variable, and ϑ the parameter vector. The problem is combined with suitable initial and boundary conditions. The unknown $u(x,t;\vartheta)$ is the exact solution of the problem, which we assume belongs to an appropriate functional space $\mathbb{V}(\Omega)$ enabled with the inner product $\langle \cdot, \cdot \rangle_{\mathbb{V}}$ and the associated norm $\| \cdot \|_{\mathbb{V}} = \sqrt{< \cdot, \cdot >_{\mathbb{V}}}$. The operator $\mathcal{G}$ corresponds to the spatial differential operator that contains the flux and source terms when we consider conservation and balance laws.

Since analytical solutions to (3.1) are typically not available, we search for a discrete counterpart, which is appropriate for its resolution. Indeed, we seek numerical approximation in a discrete subspace $\mathbb{V}_h \subset \mathbb{V}$ of dimension N_h (the number of Degrees of Freedom (DOF)), in which the approximate solution is sought. The discrete solution

$$u_h(x,t;\vartheta)$$

approximates the exact solution at point x, time instance t and input parameter ϑ.

3.2.1 Solution Manifold and Reduced Basis Approximation

A fundamental concept in ROMs is the *solution manifold* [13], which comprises all solutions of the parametric problem (3.1) under variation of the parameters

$$\mathcal{M} = \{u(\vartheta) \mid \vartheta \in \mathbb{P}\} \subset \mathbb{V}. \tag{3.2}$$

Each $u(\vartheta) \in \mathbb{V}$ corresponds to the solution of the exact problem.

Following this definition for the continuum problem, we can similarly define the discrete form of the solution manifold

$$\mathcal{M}_h = \{u_h(\vartheta) \mid \vartheta \in \mathbb{P}\} \subset \mathbb{V}_h, \tag{3.3}$$

where each $u_h(\vartheta) \in \mathbb{V}_h$ corresponds to the solution of the discrete problem. We require that the solution manifold is of low dimension, which means that the span of a low number of appropriately chosen basis functions represents the solution

manifold with small error. Assuming N_r-dimensional reduced basis, denoted as $\{\phi^k\}_{k=1}^{N_r} \subset \mathbb{V}_h$, then the associated reduced basis space is given by

$$\mathbb{V}_{N_r} = \text{span}\{\phi^1, \phi^2, \ldots, \phi^{N_r}\} \subset \mathbb{V}_h.$$

The assumption of the low dimensionality of the manifold implies that $N_r \ll N_h$, and thus the solution is approximated by

$$\tilde{u}(x, t; \vartheta) = u_0(x) + \sum_{k=1}^{N_r} \alpha_k(t; \vartheta)\phi^k(x), \tag{3.4}$$

where $\tilde{u}$ is the approximated solution, $u_0(x)$ is the offset and $\{\alpha_k\}_{k=1}^{N_r}$ are the coefficients which represent the solution in the reduced space $\mathbb{V}_h$. Common choices for the offset include:

- **No offset**: $u_0(x) = 0$,
- **"Mass" offset**: $u_0(x)$ is the mean value over the domain.

Remark. This latter case, however, requires an important clarification. For now, we will consider the offset constant, which is valid as long as the "mass" within the domain remains unchanged over time. We will later focus on cases where the "mass" is governed by inflow and outflow at the boundaries; in those specific scenarios, the offset becomes time-dependent. A more detailed analysis will be provided when we discuss conservation within the ROMs in Chapter 5.

3.2.2 Finite Volume Discretization

Even if we can discretize the spatial operator $\mathcal{G}$ by applying several approaches, i.e. Finite Difference (FD), Finite Element (FE), Finite Volume (FV) or Discontinuous Galerkin (DG), we rely on the FV method discussed in the previous part. In this context, the domain Ω is typically partitioned into a set of non-overlapping $\{\mathcal{V}_j\}_{j=1}^{N}$ cells such that

$$\Omega = \bigcup_j \mathcal{V}_j, \qquad \mathcal{V}_j \cap \mathcal{V}_i = \emptyset \text{ for } j \neq i.$$

Through the cell-averages approximation,

$$u_j(t; \vartheta) \approx \frac{1}{|\mathcal{V}_j|} \int_{\mathcal{V}_j} u(x, t; \vartheta) dx,$$

and employing a linear step scheme for time discretization with approximated fluxes at the cell interfaces, this leads to the discrete evolution equation:

$$\mathbf{u}^{n+1} = \mathcal{L}(\mathbf{u}^n), \tag{3.5}$$

where $\mathcal{L}$ represents the discrete operator introduced in expression (2.45). The discrete high-fidelity solution $u_h(x,t;\vartheta)$ is here represented by the N-dimensional vector

$$\mathbf{u}^n(\vartheta) = \begin{bmatrix} u_1^n(\vartheta) \\ u_2^n(\vartheta) \\ \vdots \\ u_N^n(\vartheta) \end{bmatrix} \in \mathbb{R}^N, \tag{3.6}$$

where $u_j^n(\vartheta) \simeq u(x_j, t^n; \vartheta)$. The discrete manifold is hence a subspace of $\mathbb{V}_h = \mathbb{R}^N$ where $N_h \equiv N$. In this subspace, the inner product is induced by a Symmetric Positive-Definite (SPD) matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$:

$$\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_{\mathbf{W}} := \mathbf{v}_1^T \mathbf{W} \mathbf{v}_2,$$

where $\mathbf{v} = [v_1, v_2, \dots, v_N]^T \in \mathbb{R}^N$.

Let $\phi^m = [\phi_1^m, \phi_2^m, \dots, \phi_N^m]^T \in \mathbb{R}^N$ be a column vector comprising the coefficients of the m-th basis function ϕ^m, $m = 1, \dots, N_r$. By introducing the offset vector $\mathbf{u}_0 \in \mathbb{R}^N$, the reduced coefficients $\boldsymbol{\alpha}^n \in \mathbb{R}^{N_r}$ and the basis functions matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N\times N_r}$ such that

$$\mathbf{u}_0 = \begin{bmatrix} u_{10} \\ u_{20} \\ \vdots \\ u_{N0} \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \phi_1^1 & \phi_1^2 & \cdots & \phi_1^{N_r} \\ \phi_2^1 & \phi_2^2 & \cdots & \phi_2^{N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N^1 & \phi_N^2 & \cdots & \phi_N^{N_r} \end{bmatrix}, \quad \boldsymbol{\alpha}^n = \begin{bmatrix} \alpha_1^n \\ \alpha_2^n \\ \vdots \\ \alpha_{N_r}^n \end{bmatrix},$$

we can define the approximated discrete solution (3.4) in compact form as follows

$$\tilde{\mathbf{u}}^n(\vartheta) = \mathbf{u}_0 + \boldsymbol{\Phi}\boldsymbol{\alpha}^n(\vartheta), \tag{3.7}$$

To summarize, the ROM procedure is divided in two main steps: the offline phase requires the computation of $\boldsymbol{\Phi}$, while the online phase computes the coefficients $\boldsymbol{\alpha}^n$.

3.3 Proper Orthogonal Decomposition

This section details the computation of ROM basis functions ϕ^m from a set of high-fidelity PDE solutions, known as snapshots.

For a given time instance $t_{k(l)}$ and input parameter $\vartheta_{j(l)}$, a snapshot is defined as $\mathbf{s}_l = \mathbf{u}^{k(l)}(\vartheta_{j(l)})$ for $l = 1, \dots, N_s$, where $u_i^{k(l)}(\vartheta_{j(l)}) \simeq u(x_i, t_{k(l)}; \vartheta_{j(l)})$. Once N_s snapshots are collected into a snapshot matrix $\mathbf{S}$, assembled as:

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N_s} \\ s_{21} & s_{22} & \cdots & s_{2N_s} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{NN_s} \end{bmatrix} \in \mathbb{R}^{N\times N_s}, \tag{3.8}$$

the basis functions are obtained by solving a Least-Squares (LS) minimization problem [15]. Let $\hat{\mathbf{s}}_l$ be the orthogonal projection of the snapshot onto the N_r-dimensional subspace spanned by the basis functions. The basis is found by solving:

$$\begin{aligned} \min_{\mathbf{\Phi}\in\mathbb{R}^{N\times N_r}} \quad & \sum_{l=1}^{N_s} \|\mathbf{s}_l - \hat{\mathbf{s}}_l\|_{\mathbf{W}}^2 \\ \text{subject to } & \langle \phi^n, \phi^m \rangle_{\mathbf{W}} = \delta_{n,m}, \quad \forall n, m \in \{1, \dots, N_r\}. \end{aligned} \tag{3.9}$$

The objective function, in the LS sense, measures the difference between the snapshots and their orthogonal projections onto the affine subspace of rank N_r. Here, δ denotes the Kronecker delta.

By centring the snapshots $\mathbf{s}_l = \mathbf{s}_l - \mathbf{u}_0$, the minimization problem (3.9) can be cast in matrix form as

$$\begin{aligned} \min_{\Phi\in\mathbb{R}^{N\times N_r}} \quad & \|\mathbf{S} - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{W}\mathbf{S}\|_{F_W}^2 \\ \text{subject to} \quad & \mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi} = \mathbf{I}_{N_r}, \end{aligned} \tag{3.10}$$

where $\mathbf{I}_{N_r}$ corresponds to the $N_r \times N_r$ identity matrix and

$$\|\mathbf{A}\|_{F_{\mathbf{W}}}^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{W}\mathbf{A})$$

denotes the Frobenius norm associated with the inner product induced by $\mathbf{W}$. In order to employ a factorization of the symmetric positive definite matrix $\mathbf{W}$, we briefly recall the definition of *Cholesky decomposition.*

Definition 3.1 (Cholesky Decomposition [15]) *The Cholesky decomposition of a Hermitian positive-definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a factorization of the form*

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T,$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is a lower triangular matrix with real and positive diagonal entries. Every Hermitian positive-definite matrix (and thus every real symmetric positive-definite matrix) admits a unique Cholesky decomposition.

The matrix $\mathbf{W}$ can be decomposed as

$$\mathbf{W} = \mathbf{W}^{1/2}(\mathbf{W}^{1/2})^T.$$

By introducing the following change of variables

$$\tilde{\mathbf{S}} = (\mathbf{W}^{1/2})^T\mathbf{S}, \qquad \tilde{\mathbf{\Phi}} = (\mathbf{W}^{1/2})^T\mathbf{\Phi},$$

we recast the problem into the standard low-rank approximation form:

$$\begin{aligned} \min_{\tilde{\mathbf{\Phi}}\in\mathbb{R}^{N\times N_r}} \quad & \|\tilde{\mathbf{S}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^T\tilde{\mathbf{S}}\|_F^2 \\ \text{subject to } & \tilde{\mathbf{\Phi}}^T\tilde{\mathbf{\Phi}} = \mathbf{I}_{N_r}, \end{aligned} \tag{3.11}$$

where $\|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{A})$ denotes now the classical Frobenius norm.

The fundamental Schmidt–Eckart–Young–Mirsky theorem provides the best rank-N_r approximation of $\tilde{\mathbf{S}}$ for the low-rank approximation problem (3.11). Before formally state the theorem, we recall the definition of Singular Value Decomposition.

Definition 3.2 (Singular Value Decomposition) *Let $\mathbf{M} \in \mathbb{K}^{n\times m}$, where $\mathbb{K}$ is either the field of real or complex numbers. The* singular value decomposition *(SVD) of $\mathbf{M}$ is the factorization*

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{K}^{n\times n}$ and $\mathbf{V} \in \mathbb{K}^{m\times m}$ are orthogonal (or unitary) matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix whose nonnegative entries are the singular values of $\mathbf{M}$ [15].

Theorem 3.1 (Schmidt–Eckart–Young–Mirsky theorem) *Let $\tilde{\mathbf{S}} \in \mathbb{R}^{n\times m}$ be a real rectangular matrix. Suppose that the singular value decomposition (SVD) of $\tilde{\mathbf{S}}$ is*

$$\tilde{\mathbf{S}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}_+^{n\times m}$ is a diagonal matrix whose entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,m)} \geq 0$ are the singular values of $\tilde{\mathbf{S}}$.

For any integer $k \leq \min(n, m)$, the best rank-k approximation to $\tilde{\mathbf{S}}$ in the Frobenius norm is given by [27, 28, 29]:

$$\min_{\mathrm{rank}(\mathbf{X})\leq k} \|\tilde{\mathbf{S}} - \mathbf{X}\|_F^2 = \|\tilde{\mathbf{S}} - \tilde{\mathbf{S}}_k\|_F^2 = \sum_{i=k+1}^{\min(n,m)} \sigma_i^2,$$

where $\mathbf{X} = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T\tilde{\mathbf{S}}$ and $\tilde{\mathbf{S}}_k$ denotes the truncated singular value decomposition of $\tilde{\mathbf{S}}$:

$$\tilde{\mathbf{S}}^* = \begin{bmatrix} U_{1,1} & \cdots & U_{1,k} \\ \vdots & & \vdots \\ U_{n,1} & \cdots & U_{n,k} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} V_{1,1} & \cdots & V_{1,m} \\ \vdots & & \vdots \\ V_{k,1} & \cdots & V_{k,m} \end{bmatrix}^T \in \mathbb{R}^{n\times k}.$$

According to the theorem (3.1) and recovering $\boldsymbol{\Phi} = (\mathbf{W}^{1/2})^{-T}\tilde{\boldsymbol{\Phi}}$ via inverse transformation, the basis functions ϕ^m are finally obtained as:

$$\boldsymbol{\Phi} = (\mathbf{W}^{1/2})^{-T} \begin{bmatrix} U_{1,1} & \cdots & U_{1,N_r} \\ \vdots & & \vdots \\ U_{N,1} & \cdots & U_{N,N_r} \end{bmatrix}, \tag{3.12}$$

where $\mathbf{U}$ is provided by the SVD of $\tilde{\mathbf{S}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$.

The dimension N_r of the ROM is chosen by relying on a popular indicator, called Relative Information Content (RIC) [15]:

$$\mathrm{RIC}(N_r) = \frac{\sum_{n=1}^{N_r} \sigma_n^2}{\sum_{n=1}^{\min(N,N_s)} \sigma_n^2} > 1 - \epsilon, \tag{3.13}$$

where σ_n corresponds to the n-th singular value. This rate indicates the percentage of total energy captured by the basis functions. Note that if the singular values decrease rapidly, we need to consider a small number N_r of modes in order to globally assimilate the solution behaviour.

3.4 Projection-Based ROM (pROM)

Having computed the reduced basis $\boldsymbol{\Phi} = [\phi^1, \dots, \phi^{N_r}] \in \mathbb{R}^{N \times N_r}$ via POD, we now address the online phase with the calculation of the reduced coefficients $\boldsymbol{\alpha}^n = [\alpha_1^n, \dots, \alpha_{N_r}^n]^T \in \mathbb{R}^{N_r}$.

Consider the continuum L^2 projection operator $\mathcal{P}_{\mathbb{V}_{N_r}} : L^2(\Omega) \to \mathbb{V}_{N_r}$ defined in [21] by:

$$g \mapsto \mathcal{P}_{\mathbb{V}_{N_r}} g := \sum_{j=1}^{N_r} \left(\int_\Omega g \phi^j dx \right) \phi^j. \tag{3.14}$$

Assuming $u_0 = 0$ for simplicity and using the approximation (3.4), we have:

$$\int_\Omega u \phi^j dx = \langle u, \phi^j \rangle_{L^2(\Omega)} \simeq \left\langle \sum_{i=1}^{N_r} \alpha_i \phi^i, \phi^j \right\rangle_{L^2(\Omega)} = \alpha_j, \tag{3.15}$$

since the basis functions are orthonormal by hypothesis.

The discrete manifold $\mathcal{M}_h \subset \mathbb{R}^N$ inherits the inner product $\langle \cdot, \cdot \rangle_{\mathbf{W}}$ associated with the norm $\| \cdot \|_{\mathbf{W}} = \sqrt{\langle \cdot, \cdot \rangle_{\mathbf{W}}}$ from the SPD matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$.

The integral is therefore approximated by:

$$\int_\Omega u(x, t; \vartheta) \phi^j dx \simeq \sum_{s=1}^{N} w_s u_s(t; \vartheta) \phi_s^j, \quad \forall j = 1, \dots, N, \tag{3.16}$$

where the w_s corresponds to the weights associated with the quadrature points of the grid. The formula (3.16) can be compactly written as:

$$\int_\Omega u \phi^j dx \simeq \langle \mathbf{u}, \phi^j \rangle_{\mathbf{W}} \equiv \phi^j \mathbf{W} \mathbf{u}. \tag{3.17}$$

The vector $\boldsymbol{\alpha}$ of reduced coefficients is given by:

$$\boldsymbol{\alpha}(t; \vartheta) = \boldsymbol{\Phi}^T \mathbf{W} \mathbf{u}(t; \vartheta). \tag{3.18}$$

Finally, the reduced solution onto the subspace $\mathbb{V}_{N_r}$ is:

$$\tilde{\mathbf{u}}(t;\vartheta) = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{W}\mathbf{u}(t;\vartheta). \tag{3.19}$$

This approach is a first layer of ROM approximation. We now investigate an alternative method that proposes a second approximation layer.

3.5 Collocated ROM (cROM)

We present this method and emphasize that it will be used for all subsequent simulations in this manuscript. This approach adds empirical quadrature as a second layer of approximation to increase computational efficiency. The main idea is to use a sparse empirical rule that only employs a carefully chosen subset of quadrature points in place of the full quadrature rule.

We approximate the projection integral as:

$$\int_\Omega u\phi^j dx \approx \sum_{\kappa=1}^{N_m} \tilde{w}_\kappa u_{m(\kappa)}\varphi^j_{m(\kappa)}, \quad j = 1,\dots,N_r, \tag{3.20}$$

where $\{\tilde{w}_\kappa\}$ are empirical weights and $m(\kappa)$ maps empirical indices to physical grid points.

The empirical quadrature points with the corresponding weights give the best approximation of the contiuum projection integral. In order to compute them, we define the matrix for $i = 1,\dots,N$:

$$\mathbf{G}_i = \begin{bmatrix} s_{11}\phi^i_1 & \cdots & s_{1N}\phi^i_N \\ \vdots & \ddots & \vdots \\ s_{N_s1}\phi^i_1 & \cdots & u_{N_sN}\phi^i_N \end{bmatrix}, \quad \mathbf{d}_i = \begin{bmatrix} \langle s_1, \phi^i\rangle \\ \vdots \\ \langle s_{N_s}, \phi^i\rangle \end{bmatrix}. \tag{3.21}$$

Assembling the global matrix $\mathbf{G} = [\mathbf{G}_1^T,\dots,\mathbf{G}_{N_r}^T]^T \in \mathbb{R}^{N_rN_s\times M}$ and vector $\mathbf{d} = [\mathbf{d}_1^T,\dots,\mathbf{d}_{N_r}^T]^T \in \mathbb{R}^{N_rN_s}$, we solve the Non-Negative Least Squares (NNLS) minimization problem [30] [31] [32]:

$$\bar{\boldsymbol{\omega}} = \arg\min_{\mathbf{w}\in\mathbb{R}^M_+} \|\mathbf{G}\mathbf{w} - \mathbf{d}\|_2^2, \tag{3.22}$$

using the Lawson-Hanson algorithm [33]. This yields sparse nonnegative weights $\bar{\boldsymbol{\omega}}$ that accurately approximate the projection integrals while using only a small subset of quadrature points.

By adapting the formula (3.16) for the coefficient $\boldsymbol{\alpha}$, this yields to:

$$\tilde{\boldsymbol{\alpha}}(t;\vartheta) = \boldsymbol{\Phi}\mathbf{W}_\epsilon \mathbf{u}(t;\vartheta), \tag{3.23}$$

where $\tilde{\boldsymbol{\alpha}}$ is the vector approximating the vector of reduced coefficients $\boldsymbol{\alpha}$ and $\mathbf{W}_\epsilon \in \mathbb{R}^{N\times N}$ is the matrix containing all the non-zeros entries obtained by solving the NNLS problem (3.22). The ϵ subscript refers to the approximation error made. This matrix is diagonal and, moreover, is sparse along the diagonal since only few entries are not zero. Let $\mathcal{I}_c$ be the set of indexes where $\bar{\omega}_j > 0$ such that:

$$\omega_j = (\mathbf{W}_\epsilon)_{jj}, \quad j \in \mathcal{I}_c, \tag{3.24}$$

corresponding to the indexes of the so-called *collocation points* or *magic points*. Solving the problem means finding the best approximation of the empirical weights, taking into account the following error committed due to the NNLS algorithm:

$$\|\boldsymbol{\Phi}\mathbf{W}\mathbf{u}(t;\vartheta) - \boldsymbol{\Phi}\mathbf{W}_\epsilon \mathbf{u}(t;\vartheta)\|_2 \leq \epsilon_{\text{NNLS}} \|\boldsymbol{\Phi}\mathbf{W}\mathbf{u}(t;\vartheta)\|_2. \tag{3.25}$$

The reduced solution is now a collocated solution, expressed as:

$$\mathbf{u}_c(\vartheta) = \sum_{m=1}^{N_r} \left(\sum_{j\in\mathcal{I}_c} \varphi_j^m \omega_j^\epsilon \mathbf{u}(\vartheta) \right) \phi^m, \tag{3.26}$$

or simply,

$$\mathbf{u}_c(\vartheta) = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{W}_\epsilon \mathbf{u}(\vartheta). \tag{3.27}$$

By computing the reduced solution at the discrete time t^n, the collocated solution at the next time t^{n+1} is updated via the high-fidelity discretization $\mathcal{L}(\mathbf{u}_c^n)$ as follows:

$$\mathbf{u}_c^{n+1}(\vartheta) = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{W}_\epsilon \mathcal{L}(\mathbf{u}_c^n(\vartheta)). \tag{3.28}$$

We can define this approach *collocated* or *collocation-based Reduced Order Model* (cROM) since it is not a simple projection onto the trial subspace, but we consider a new layer of approximation based directly on the collocation points.

Since only N_m points need to be employed in order to compute the update collocated solution, then this latter equation may be manipulated further, leading to:

$$\mathbf{u}_c^{n+1}(\theta) = \boldsymbol{\Phi}\overline{\boldsymbol{\Phi}}^T \overline{\mathbf{W}}_\epsilon \overline{\mathcal{L}(\mathbf{u}_c^n(\theta))}. \tag{3.29}$$

The notation via bar $\bar{\cdot}$ over the variables is introduced to specify that the quantities are computed only at the N_m collocated points. A last step can be finally done; indeed, since the HF discretization is typically made by considering three-stencil type-method, we need to know the solution in the magic points and their neighbours. We define N_{cn} the total amount of collocated points with their neighbours. Note

that a few collocated points may share the same neighbour points or simply be the neighbour of one of them. In order to restrict the computation to N_{nc}, we introduce an additional notation, namely the $\tilde{\cdot}$ over a matrix or a vector. The final form of our cROM is now given by:

$$\tilde{\mathbf{u}}_c^{n+1}(\vartheta) = \tilde{\mathbf{\Phi}}\overline{\mathbf{\Phi}}^T\overline{\mathbf{W}}_\epsilon\overline{\mathcal{L}}(\tilde{\mathbf{u}}_c^n(\vartheta)), \tag{3.30}$$

where $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{N_{cn}\times N_m}$.

3.6 cROM implementation

We have described the structure of the cROM in greater mathematical detail, but we will now briefly summarize the numerical steps needed to implement this algorithm. The cROM implementation follows the standard offline-online decomposition:

Algorithm: The cROM

Input: N, N_s, N_r, N_m and ϵ_{NNLS}.
Output: The collocated solution $\mathbf{u}_c^{n+1}(\vartheta)$ at the time t^{n+1}.

Offline phase.

1. Compute the HF snapshots
$$\mathbf{s}_l = \mathbf{u}_h^{k(l)}(\vartheta_{j(l)}), \quad i = 1, \dots, N_s,$$
via FV method.
2. Assemble the snapshot matrix $\mathbf{S} \in \mathbb{R}^{N\times N_s}$, whose i-th column is $\mathbf{s}_i$, and compute POD basis $\mathbf{\Phi} \in \mathbb{R}^{N\times N_r}$ via SVD.
3. Solve NNLS problem to obtain the set of indices $\mathcal{I}_c$ of the collocation points and the weights $\omega_j^\varepsilon = (\mathbf{W}_\varepsilon)_{jj}$, $j \in \mathcal{I}_c$.

Online phase.

1. Given $\mathbf{u}_c^n(\vartheta)$, compute the HF discretization at collocated points:
$$\tilde{\mathbf{u}}_c^n(\vartheta) = \mathcal{L}(\tilde{\mathbf{u}}_c^n(\vartheta)).$$
2. Compute the cROM solution in the collocation points and neighbouring cells:
$$\tilde{\mathbf{u}}_c^{n+1}(\vartheta) = \tilde{\mathbf{\Phi}}\overline{\mathbf{\Phi}}^T\overline{\mathbf{W}}_\epsilon\overline{\mathcal{L}}(\tilde{\mathbf{u}}_c^n(\vartheta)).$$
3. Reconstruct full discrete solution at the desired time:
$$\mathbf{u}_c^{n+1}(\vartheta) = \mathbf{\Phi}\overline{\mathbf{\Phi}}^T\overline{\mathbf{W}}_\epsilon\overline{\mathcal{L}}(\tilde{\mathbf{u}}_c^n(\vartheta)).$$

3.7 NNLS Algorithm and Computational Speedup Analysis

3.7.1 NNLS Tolerance

The selection of magic points is performed via NNLS using MATLAB's `lsqnonneg` function. The default tolerance ϵ_{NNLS} is defined as:

$$\epsilon_{\text{NNLS}} = 10 \cdot \texttt{eps} \cdot \texttt{norm(G,1)} \cdot \texttt{length(G)}$$

where `eps` is MATLAB's machine precision and `G` is the coefficient matrix $\mathbf{G}$ in the NNLS formulation (3.22). In the next chapters, we will specify the tolerance set for each simulation performed. The NNLS algorithm is applied with a preconditioning strategy that implicitly bounds the maximum number of selectable magic points. While this upper bound is set *a priori*, the actual number of points identified by the algorithm may be lower, depending on the tolerance and the specific problem.

3.7.2 Computational Complexity Comparison

In the Full Order Model (FOM), the computational cost of the finite volume scheme scales as:

$$\mathcal{C}_{\text{FOM}} = \mathcal{O}(N_t \cdot N \cdot C_{\text{FV}})$$

where N_t is the number of time steps, N the number of spatial cells and C_{FV} the cost per cell of the finite volume update.

The collocated Reduced Order Model (cROM) attempts to alleviate the computational effort through:

$$\mathcal{C}_{\text{cROM}} = \mathcal{O}(\tilde{N}_t \cdot (N_m \cdot C_{\text{FV}} + N_{cn} \cdot N_m))$$

where $\tilde{N}_t$ denotes the number of time steps in reduced model (typically $\tilde{N}_t \leq N_t$), N_m the number of magic points and N_{cn} the number of magic points plus neighbours.

At each time step, the term $N_m \cdot C_{\text{FV}}$ represents the cost of updating the solution at magic points, while $N_{cn} \cdot N_m$ accounts for the extension operator application. Indeed, the extended operator $\tilde{\mathbf{\Phi}}\overline{\mathbf{\Phi}}^T\overline{\mathbf{W}}_\epsilon$ can be pre-allocated into a matrix of dimension $N_{cn} \times N_m$ before computing the online stage.

3.7.3 Time steps analysis

Let M be the number of magic points where we can compute the maximum wave speed necessary for the CFL condition. We may obtain the following inequality:

$$\max_{i \in P} |a_{i+1/2}| \geq \max_{i \in M} |a_{i+1/2}| \tag{3.31}$$

where $M \subset P$, thus the total number of cells contains a restricted set of cells related to the magic points.

If the mesh is not uniform, then it is possible to estimate with an additional inequality the space width Δx_i which is now dependent from the cells. The CFL condition is now variable over the time domain and the time step is given by:

$$\Delta t^n = \frac{\Delta x^M_{\min}}{\max_{i \in M} |a_i|} \text{Cour.} \tag{3.32}$$

If we consider the minimum over the M cells, in general this minimum will be greater than the minimum over all the cells, so:

$$\Delta x^M_{\min} \geq \Delta x^P_{\min}.$$

The time step can become greater as for the non uniform meshes since the collocated points the minimum could not be the global minimum as for the inequality seen before. This explains why in general $\tilde{N}_t \leq N$ and this is considerably valuable with respect to the speedup.

3.7.4 Speedup Definition

The computational speedup is quantified as:

$$\text{Speedup} = \frac{T_{\text{FOM}}}{T_{\text{cROM}}},$$

where T_{FOM} and T_{cROM} represent the total execution times of the full and reduced order models, respectively.

Remark. All numerical simulations were run using MATLAB R2024b on a personal computer equipped with an Intel Core i5-1035G1 processor and 8 GB RAM. The reported speedup values are still useful for comparing the full and reduced order models, even though absolute timing measurements depend on the hardware.

Chapter 4

Positivity-preserving ROMs

Conservation laws are a class of PDEs that can exhibit discontinuous or non-smooth solutions with advection-dominated behaviour, therefore classical ROMs are likely to fail in this scenario. Hence, physical properties are easily lost. On the contrary, there exist many HF schemes capable of preserving relevant properties, such as positivity, entropy stability, conservation, etc. In general, the research aims to transfer them to ROMs. In this chapter, we focus on positivity-preservation seeking a ROM approach that inherits this property.

The previous section has shown a general description of ROM and discussed two different approaches: pROM and cROM. In the next sections, we examine some numerical test cases employing techniques applied directly to the cROM.

4.1 Test case 1: Linear Advection Equation

Let us recall the linear advection equation (2.27):

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\,\dfrac{\partial u}{\partial x} = 0, & x \in (0,l),\ t \in (0,T], \\ u(x,0) = u_0(x), & x \in [0,l]. \end{cases} \tag{4.1}$$

We consider the following test case with a smooth initial condition, defined as:

$$u_0(x) = \begin{cases} A \sin^2\Big(\dfrac{\pi}{L}(x - x_c) + \dfrac{\pi}{2}\Big), & x \in (x_c - L/2, x_c + L/2) \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

The length interval is $l = 1$, while the final time is $T = 0.4$. The coefficients of the initial solution u_0 are set to $A = 1$, $x_c = 0.2$ and $L = 0.2$. We perform a discretization over the domain [0,1] by considering $N = 2000$ cells ($N+1$ interfaces).

The mesh width is equal to $\Delta x = (1-0)/N = 5 \cdot 10^{-4}$. To approximate the fluxes at the interfaces, the upwind scheme (2.42) is implemented. As mentioned, this method is consistent and, moreover, stable (under certain norms), only if the CFL condition (2.48) is satisfied. Setting the Courant number Cour $= 0.8$, the expression (2.50) provides the time step $\Delta t = 4 \cdot 10^{-4}$. We recall that the analytical solution for this problem is expressed in formula (2.53) and corresponds to the initial profile shifted in according to the sign of the propagation speed; in our case, it is on the right ($a > 0$).

While the parameter ϑ is not directly involved in the formulation, the problem can be considered as parametric. We can treat time t as a parameter by collecting snapshots at discrete times during the simulation to capture the dynamical evolution. The key is that these snapshots are not captured at each time step of the HF scheme, but typically on a larger scale. This practice of sub-sampling the time domain is precisely what allows us to consider t as a discrete parameter, thus permitting to deal with a parametric problem. We collected $N_s = 40$ snapshots at intervals of $\Delta t^{\text{snap}} = 0.0103s$, which are reported in the following figure.

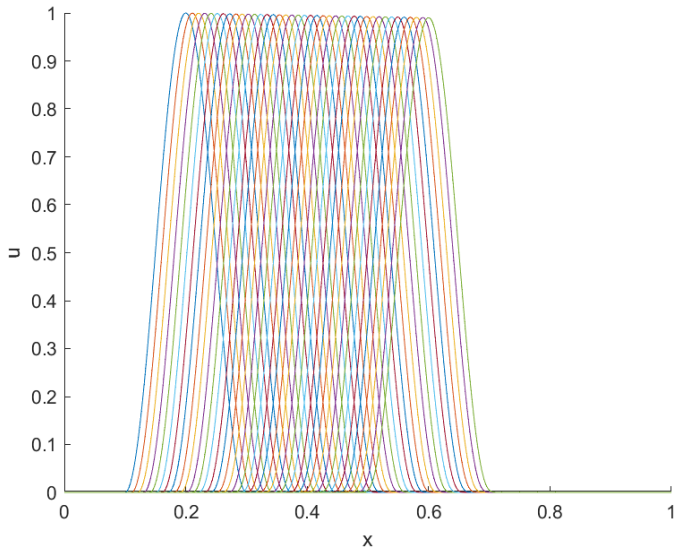

Figure 4.1: 40 snapshots for the problem (4.1) with space step $\Delta x = 5 \cdot 10^{-4}$, time step $\Delta t = 4 \cdot 10^{-4}$ and Courant number Cour $= 0.8$ for the upwind scheme.

The number of snapshots can be increased to obtain a deeper understanding of the system's general dynamics. While a large set of HF solutions allows the ROM to better assimilate the system's behaviour through the basis functions, it also implies a computational effort of the offline phase. This stage requires the calculation of a significant number of HF solutions, followed by performing a SVD and solving a NNLS problem for high-dimensional systems. The offline phase is,

by definition, the most computationally expensive stage of a ROM. Although it is reasonable to invest computational resources in this stage, the cost of the HF simulations can be prohibitively high. In this example, our focus is not primarily on this aspect, even though it warrants more attention for higher-dimensional problems.

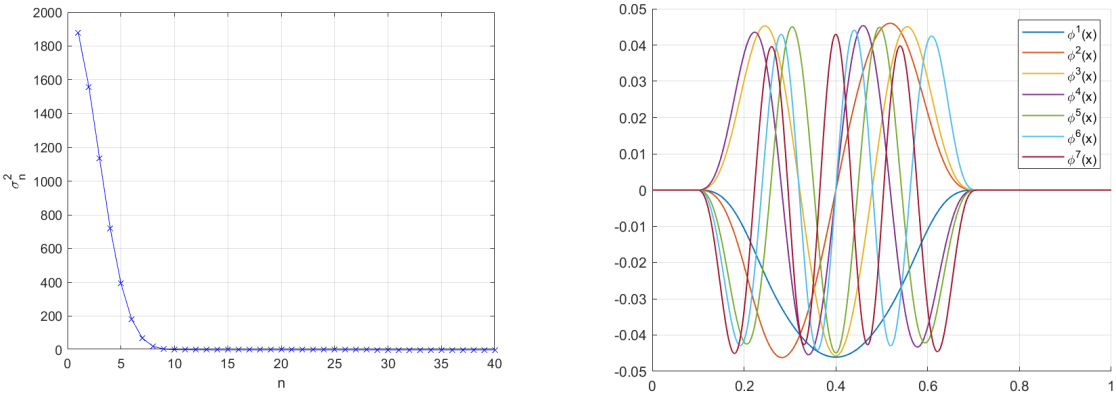

Figure 4.2: Plot of squared singular values σ_n^2 (left) and basis functions ϕ^m for $m = 1, \dots, 7$ (right) for standard cROM.

The SVD of the snapshots matrix (3.8) is performed. Plotting the squared singular values (see Figure 4.2) and employing the RIC indicator with $\epsilon = 0.01$, we discover that only 7 out of 40 basis are required to capture the 99% of the system's energy, highlighting the effective compressibility of the system. Within the offline phase, the NNLS algorithm identifies a number of collocated or magic points $N_m = 119$ with NNLS tolerance ϵ_{NNLS} set to default, as described in Section 3.7.1.

For the online stage, a new solution is computed by advancing the simulation to a chosen final time. It is crucial to note that in this specific test case, distinguishing between in-sample and out-of-sample parameters is not straightforward. Since the discrete solution is updated at each time step, it accumulates an approximation error. Therefore, even if the final time corresponds to the time of an in-sample snapshot, the resulting ROM solution will not perfectly match the original high-fidelity solution due to this error accumulation.

Figure 4.3 shows a comparison of the HF solution, the pROM solution, and the cROM solution at $t^* = t_{30}$, which corresponds to the temporal instant of the 30th snapshot. In the legend, u^{HF} denotes the high-fidelity solution, u_r the pROM solution, and u_c the cROM solution. The result highlights a significant issue: oscillations on the left side of the domain, where the solution should form a flat plateau. This condition is shared by both the pROM and cROM, underscoring that the problem is not restricted to the cROM approach but is already present

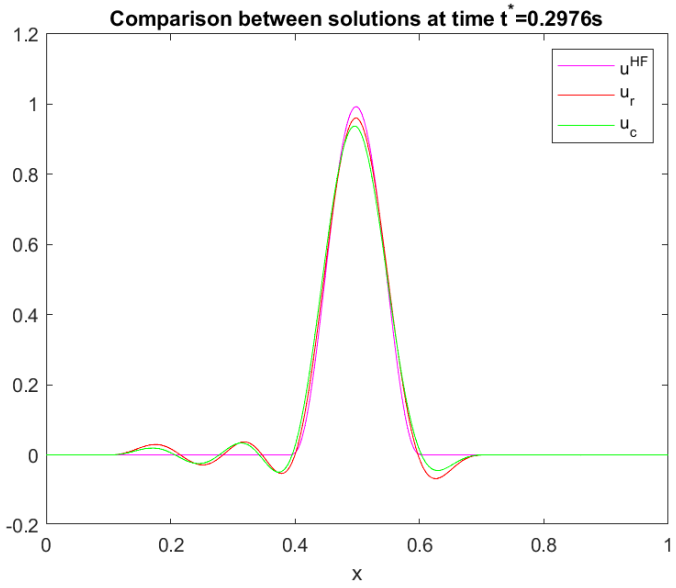

Figure 4.3: Comparison between HF, pROM and cROM solutions at $t_{30} = 0.2976s$.

in the classical pROM. While these oscillations leading to negative values are mathematically feasible in this toy example, they motivate the exploration of techniques to ensure positivity. Such methods should be developed for preserving the physical positivity of conservative variables when applied to systems such as the SW equations, where negative values are not admissible. To guarantee positivity solutions, the next section investigates two nonlinear transformations within the cROM framework: Logarithm-Exponential (LE) and Square-Root (SR).

4.1.1 Positivity and Non-Negativity-Preserving Transformations

Let $\mathbf{u}_c$ be the collocated solution. The core idea is to transform the updated solution applying a given function, perform the cROM procedure and then go back to the original solution with the inverse function.

Let us begin with the **Logarithm-Exponential (LE)** approach, whose idea has been proposed for cross-diffusion systems in [24]. The standard cROM update is:

$$\mathbf{u}_c^{n+1} = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{W}_\epsilon\mathcal{L}(\mathbf{u}_c^n). \tag{4.3}$$

The modified LE approach acts on the updated solution by the $\mathcal{L}$ operator:

$$\mathbf{u}_c^{n+1} = \exp[\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^T\hat{\mathbf{W}}_\epsilon\log(\mathcal{L}(\mathbf{u}_c^n))]. \tag{4.4}$$

where the hat notation indicates that the matrices $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{W}}_\epsilon$ are derived from transformed snapshots. Indeed, it is essential to note that these transformations modify the input of the cROM architecture, as the model acts on transformed solutions. Consequently, the computation of matrices $\mathbf{\Phi}$ and $\mathbf{W}_\epsilon$ differs from the classical cROM approach due to the snapshot matrix $\mathbf{S}$, which now contains transformed vectors. In particular, this transformation implies that the POD basis is optimal for representing $\log(u)$ instead of the original variable u. This formulation enforces positivity by construction, as the exponential function provides strictly positive values. However, a critical implementation issue arises: the logarithm is undefined for non-positive arguments; in this example, the HF solution is zero on both sides of the domain. To overcome this, we introduce a small upward shift before applying the logarithm and a corresponding downward shift after the exponential, yielding finally to:

$$\mathbf{u}_c^{n+1} = \exp[\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^T\hat{\mathbf{W}}_\epsilon \log(\mathcal{L}(\mathbf{u}_c^n) + s)] - s \tag{4.5}$$

We perform the simulations with the shift parameter set to $s = 0.05$. However, it is crucial to note that this formulation does not guarantee strict positivity. While the exponential function ensures positive values, the final downward shift can produce negative results if $\exp[\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^T\hat{\mathbf{W}}_\epsilon \log(\mathcal{L}(\mathbf{u}_c^n) + s)] < s$.

We propose the **Square-Root (SR)** transformation, which represents, to the best of our knowledge, an unexplored alternative for ensuring positivity. This approach only requires the solution to be nonnegative and is computationally simpler, as it avoids the need for shift in this example. The corresponding updated solution is given by:

$$\mathbf{u}_c^{n+1} = \left\{\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^T\hat{\mathbf{W}}_\epsilon \sqrt{(\mathcal{L}(\mathbf{u}_c^n))}\right\}^2. \tag{4.6}$$

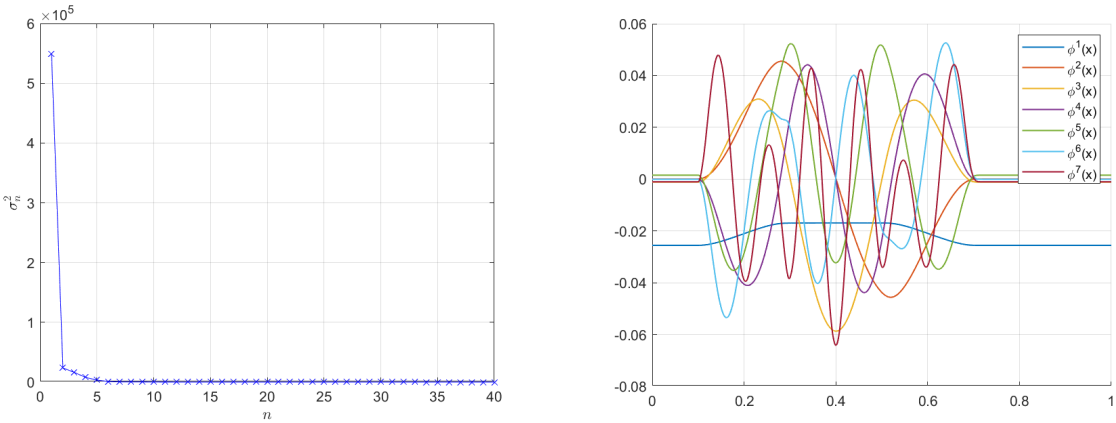

Figure 4.4: Plot of squared singular values σ_n^2 (left) and basis functions ϕ^m for $m = 1, \ldots, 7$ (right) for cROM+LE.

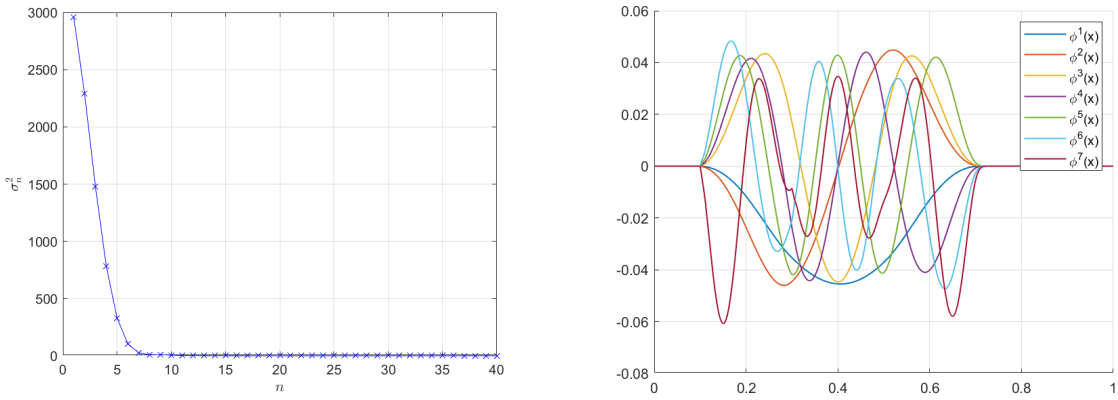

Figure 4.5: Plot of squared singular values σ_n^2 (left) and basis functions ϕ^m for $m = 1, \ldots, 7$ (right) for cROM+SR.

As a result, the squared singular values and basis functions are modified, as shown in Figures 4.4 and 4.5. In particular, the first squared singular values for LE approach is significantly higher than the subsequent one. We tested different values for the shift parameter and observed that this parameter compresses the bump in the snapshots dataset. This compression causes the first POD mode to mostly capture the general behaviour of the solution, which results in a large gap between the first and second squared singular values.

The first $N_r = 4$ basis provide the 99% of the system's total energy for LE approach, whereas $N_r = 6$ basis are required for SR approach. This indicates that most energy is contained in a few modes and, furthermore, that the transformations do not alter significantly the compressibility of the system. Examining the first basis function ϕ^1, Figures (4.4)-(4.5) reveal a notable difference: the SR approach generates the similar bump behaviour observed in the classical cROM, while the LE approach shifts it downward without preserving this shape. This stems precisely from the introduction of the shift in the transformation. If the snapshots are not centred at the beginning of the cROM procedure, i.e. the offset $u_0(x) = 0$, then the first basis function corresponds to the average of all snapshots. Since the snapshots represent the same solution propagating rightward over the domain, ϕ^1 effectively approximates the initial solution's profile in classical cROM and SR approach.

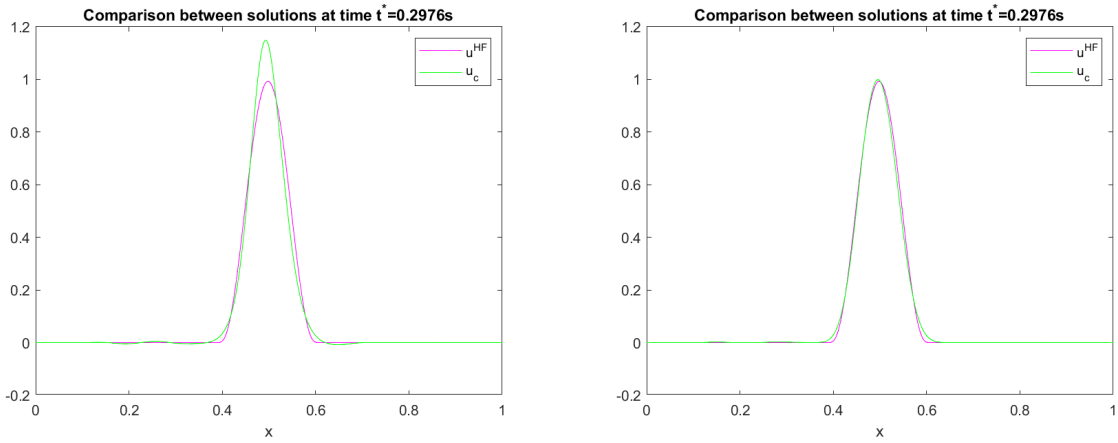

Figure 4.6: Comparison between solutions obtained with LE transformation (left) and SR transformation (right).

In Figure 4.6 we show the effect of transformations on the problem solution, with simulations run until $t_{30} = 0.2976s$ and set *a priori* $N_r = 7$ and $N_m = 200$. The LE transformation successfully damps oscillations, even though it produces slightly negative values and overestimates the maximum of the bump. The SR method, on the other hand, not only damps oscillations effectively but also captures the correct bump profile.

4.1.2 Error analysis

We compare the relative approximation errors across three approaches: the classical cROM, and cROM combined with the LE and SR transformations. The error is computed as follows:

$$\frac{\|u_{HF}(T) - u_{ROM}(T)\|_{L^1}}{\|u_{HF}(T)\|_{L^1}}, \tag{4.7}$$

where $T = t_{40}$. The discrete norm is defined as:

$$\|\mathbf{v}\|_{L^1} := \sum_{i=1}^{N} |v_i| \Delta x.$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_N]^T \in \mathbb{R}^N$.

Table 4.1 presents the number of magic points, percentage relative error, and speedup for the three approaches across different numbers of basis functions. Since we increase the number of basis functions, the NNLS algorithm needs more iterations to reach the tolerance, which explains why the number of magic points increases. The modified cROM approaches require more magic points than the standard method. Both the classical and SR approaches show similar accuracy when using

many basis functions, while LE has the highest error in these cases. Regarding computational efficiency, the speedup generally shows a decreasing trend across all methods.

	N_m			Err u [%]			Speedup [×]		
N_r	cROM	LE	SR	cROM	LE	SR	cROM	LE	SR
5	107	137	117	72.89	40.06	41.60	3.62	2.21	3.91
7	107	145	124	27.68	13.91	7.18	2.57	1.47	2.70
10	119	153	125	3.60	8.97	5.14	2.72	1.49	2.90
15	129	161	138	1.50	2.70	1.64	1.80	1.10	2.17
20	139	167	146	1.06	1.88	1.24	1.94	1.39	1.99
25	144	174	147	0.95	1.80	1.40	1.10	1.34	1.78
30	150	183	153	0.94	1.16	1.10	1.98	1.33	1.77

Table 4.1: N_m, realtive L^1 error and speedup varying N_r at time $t_{40} \equiv 0.4s$.

We now fix $N_r = 15$ basis functions, previously identified as the best trade-off, and conduct a further analysis on the number of magic points. Interestingly, the error slightly decreases when using fewer points (e.g., 110 or 90), while still maintaining a good speedup for this 1D problem. The results in the last row demonstrate that an insufficient number of magic points (2.5%) leads to increased error, thus providing a lower bound for the required number of collocation points. In particular, we observe an error saturation for N_m ranging from 90 to 160.

	Err u [%]			Speedup [×]		
N_m	cROM	LE	SR	cROM	LE	SR
160	1.50	2.70	1.64	2.27	1.50	2.23
150	1.49	2.69	1.64	2.21	1.48	2.40
130	1.50	2.67	1.61	2.10	1.71	2.42
110	1.51	2.59	1.48	2.79	2.11	3.32
90	1.50	7.19	1.47	3.63	2.73	4.62
70	1.80	7.47	3.34	4.25	2.82	4.89
50	4.89	62.26	15.37	6.91	4.39	8.57

Table 4.2: Error and speedup varying N_m at time $t_{40} \equiv 0.4s$, fixed $N_r = 15$.

We now analyze the percentage of negative solution values during the simulation. Setting $N_r = 10$ basis functions, we quantify the proportion of solution cells that become negative over the entire time evolution up to $T = 0.4s$. Figure 4.7 shows the percentage of negative cells $N_{<0}$ relative to the total number N, plotted against the time step ($t^n = n\Delta t$). The results reveal contrasting behaviours: cROM shows oscillatory patterns with a period of approximately 200 time steps and

negative values ranging from 20% to 30%. The cROM+LE approach exhibits unsatisfactory values, with negative cells percentage reaching up to 70%. In contrast, the cROM+SR approach fully maintains positivity, demonstrating its effectiveness in preserving this physical property.

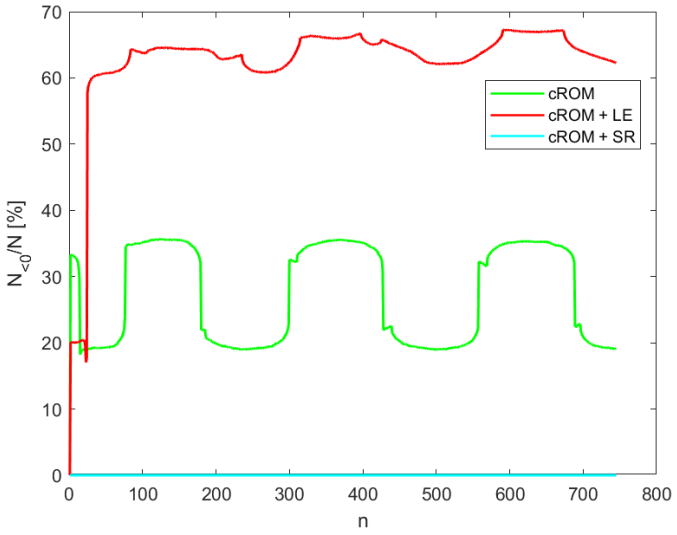

Figure 4.7: Percentage $N_{<0}/N$ of negative cells until final time $T = 0.4s$ across the three approaches

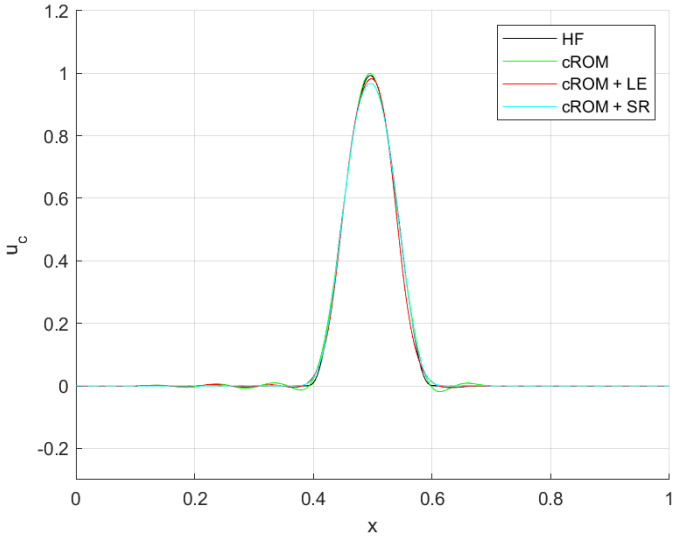

Figure 4.8: Comparison of solutions across three different approaches, fixing $N_r = 10$, $N_m = 200$ and $T = 0.4s$.

To investigate the high percentage of negative values in cROM+LE approach, we compare the solutions at time $t_{30} = 0.2976s$ corresponding to the 30th snapshot. Since Figure 4.8 does not straightforwardly highlight the reason of this issue, we therefore report two lateral regions in Figure 4.9. These zoomed views reveal that cROM+LE underestimates the flat plateau with values dropping below zero, which account for the greater proportion of points involved in Figure 4.7. We attribute this behaviour to the shift introduced in our transformation, which is necessary for this test case. Although the shift is small, it becomes critical in problems requiring strictly positive variable values.

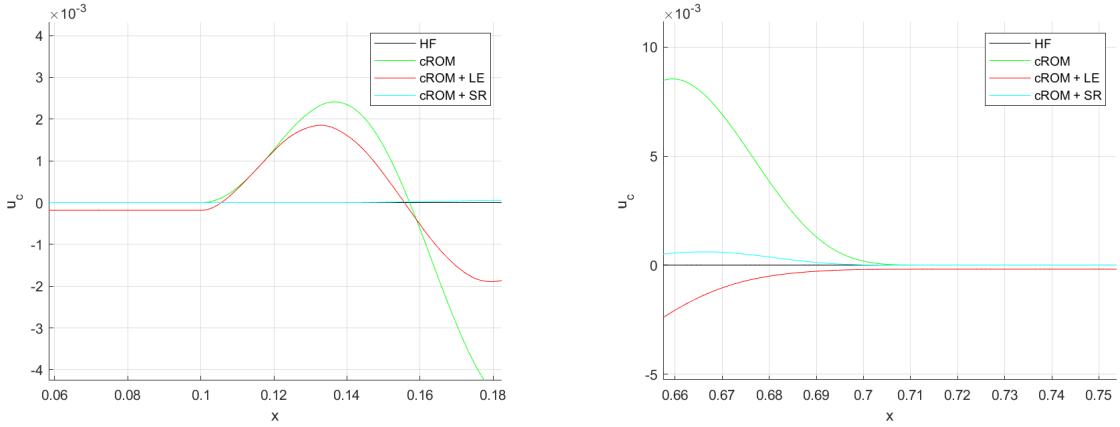

Figure 4.9: Zoom of solutions showed in Figure 4.8.

In summary, cROM+SR achieves performance comparable to classical cROM while preserving positivity by construction, which was the central objective of this section. Although cROM+LE requires further development, it may represents a promising alternative for shift-free scenarios. We now turn to a more challenging physical test case where non-negativity of the conservative variable is essential.

4.2 Test case 2: Shallow Water Equations

In this section, we present some numerical examples based on the Shallow Water (SW) equations (2.10), where $h \geq 0$ by definition. This system of PDEs can be rewritten using the conservative variable $q = hu$ as:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q}{\partial x} = 0, \\ \dfrac{\partial q}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q^2}{h} + \dfrac{1}{2}gh^2\right) = -gh\,\partial_x z(x). \end{cases} \tag{4.8}$$

Since we are dealing with a nonlinear system, we should adapt a transformation

architecture accordingly. Given the greater results of the SR approach highlighted for the linear advection equation, we extend this technique to SW system. Let us first examine the update for the collocated solution of h:

$$\mathbf{h}_c^{n+1} = \left\{ \hat{\mathbf{\Phi}}_h \hat{\mathbf{\Phi}}_h^T \hat{\mathbf{W}}_\epsilon [\hat{\mathcal{L}}(\mathbf{U}_c^n)_1] \right\}^2 = \left\{ \hat{\mathbf{\Phi}}_h \hat{\mathbf{\Phi}}_h^T \hat{\mathbf{W}}_\epsilon \sqrt{(\mathcal{L}(\mathbf{U}_c^n))_1} \right\}^2, \tag{4.9}$$

where $\mathcal{L}(\mathbf{U}_c^n)$ advances the solution $\mathbf{U}_c^n = [\boldsymbol{h}_c^n, \boldsymbol{q}_c^n]^T$ to the next time step using the chosen numerical scheme. The subscript denotes the first component of $\mathcal{L}(\mathbf{U}_c^n)$.

While the LE transformation for q is not strictly necessary, we chose to extend the SR transformation to the second variable. Noting that:

$$\mathbf{q}_c^{n+1} = \mathbf{h}_c^{n+1} \mathbf{u}_c^{n+1} = \sqrt{\mathbf{h}_c^{n+1}} \left(\sqrt{\mathbf{h}_c^{n+1}} \mathbf{u}_c^{n+1} \right). \tag{4.10}$$

we can express the update for q as:

$$\mathbf{q}_c^{n+1} = \hat{\mathbf{\Phi}}_h \hat{\mathbf{\Phi}}_h^T \hat{\mathbf{W}}_\epsilon \sqrt{\mathcal{L}(\mathbf{U}_c^n)_1} \hat{\mathbf{\Phi}}_q \hat{\mathbf{\Phi}}_q^T \hat{\mathbf{W}}_\epsilon \frac{\mathcal{L}(\mathbf{U}_c^n)_2}{\sqrt{\mathcal{L}(\mathbf{U}_c^n)_1}} \tag{4.11}$$

Having described the SR transformation framework for the one-dimensional SW equations, we now discuss the numerical flux used for snapshot computation. We implemented the HLL solver from [10] for flux approximation. Since the wave speeds vary over the time due to their dependence from the conservative variables, the CFL condition must be updated at each time step [10] as follows:

$$S_{\max}^n = \text{cour} \frac{\Delta x}{\Delta t}, \tag{4.12}$$

where $S_{\max}^n$ is the maximum wave speed at time n:

$$S_{\max}^n = \max_j \left(|u_j| + \sqrt{g h_j} \right). \tag{4.13}$$

To validate our implementation of the HLL numerical flux, we provided a convergence test using a double rarefaction waves problem (see Appendix A). In the following sections, we present two distinct test cases: the first without source terms (flat bottom topography), and the second incorporating a non-flat basin.

4.2.1 Singular non-transonic wave

To construct a singular non-transonic rarefaction wave test case, we follow the treatment in [34]. By analyzing the intersection of rarefaction and shock curves in the phase plane, we derived the correct left and right states for the Riemann problem. This analysis yields the following initial conditions:

$$\mathbf{U}(x,0) = \begin{cases} (h_L,\, u_L) = (0.2,\, -1), & x_{in} \leq x < x_c, \\ (h_R,\, u_R) = (0.0368717,\, 0.598579), & x_c < x \leq x_{end}. \end{cases} \tag{4.14}$$

where $x_{in} = -1.5$, $x_{end} = 1.5$, and $x_c = 0.5$. The domain of length 3 is discretized using $N = 2000$ cells, with a CFL number set to 0.9. Figure 4.10 exhibits a comparison of the exact solution of this Riemann problem with the numerical one computed through HLL solver until the final time chosen, i.e. $T = 0.4s$. The

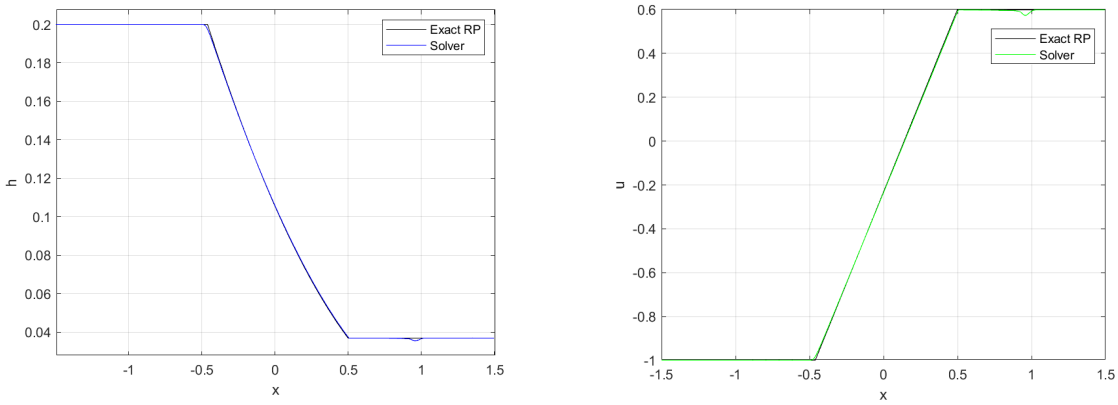

Figure 4.10: Comparison between solutions at final time T = 0.4

Figure reveals a spurious wave typical of the HLL scheme, although the method captures the exact solution with good global accuracy. This motivates the choice of this numerical flux for snapshot collection. Setting $N_s = 40$, Figure 4.11 displays the squared singular values for h and q with respect to cROM and cROM+SR approaches, both computed via SVD.

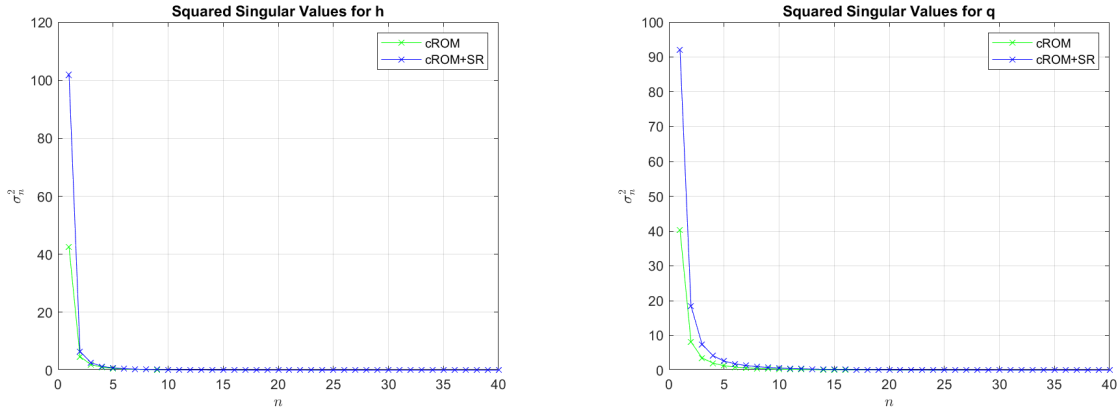

Figure 4.11: Comparison of squared singular values for h (left) and q (right) between classical cROM and cROM+SR approaches.

Overall, the SR transformation does not alter the compressibility of the system. Specifically, 99% of the system's energy is captured by 2 modes for h and 3 modes for q in standard cROM, compared to just 1 mode for h and 2 modes for q when using cROM+SR. Since basis functions showed similar trends, they are not reported. From these observations, we initially set $N_r = 3$ and $N_m = 200$ to compute the reduced order solution up to the final time t_{30}. The ϵ_{NNLS} tolerance is set to its default value. Figure 4.12 shows the comparison of the two approaches with the exact Riemann problem solution.

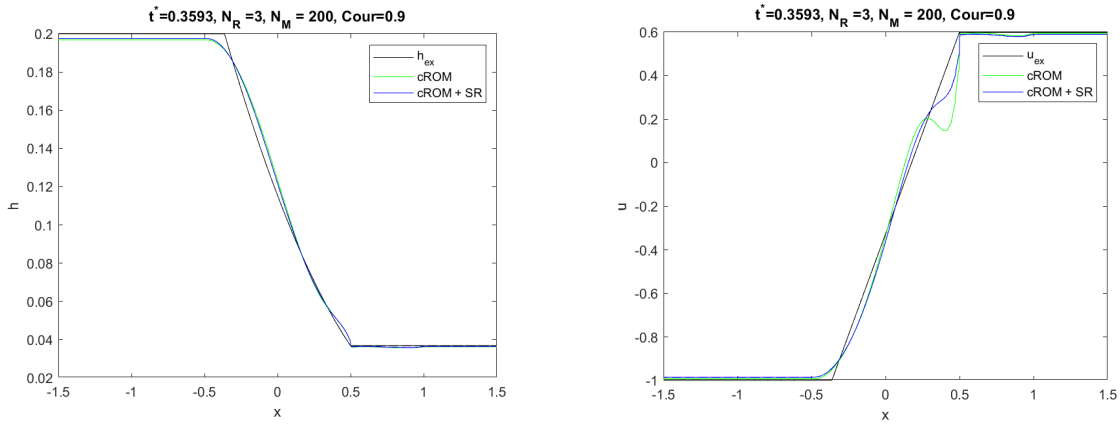

Figure 4.12: Comparison of h (left) and u (right) at time t_{30} between classical cROM, SR approaches, and the exact solution.

Both methods represent the general solution behaviour using an extremely small number of modes, i.e. $N_r = 3$. However, the flat plateau on the left for h is not perfectly captured, and the solution for u is less accurate near the initial

discontinuity at x_c. The relative L^1 error produces the following results:

	cROM		**SR**	
	L^1 Error	Rel Error [%]	L^1 Error	Rel Error [%]
h	0.0088	2.46	0.0072	2.00
u	0.0153	4.27	0.0128	3.56

Table 4.3: Error with $N_r = 3$, $N_m = 200$ at $T = 0.4s$.

Table 4.3 demonstrates that both methods achieve an acceptable approximation, even though the SR transformation yields superior results in terms of relative error.

Let us continue studying the error for h and u varying the number of basis functions selected for the reduced space approximation. We compare different choices of N_r and their relative speedup. The first column displays the actual number of magic points computed through NNLS. We performed the simulation until the time T $= t_{40} \equiv 0.4s$.

	N_m		Err h [%]		Err u [%]		Speedup [×]	
N_r	cROM	SR	cROM	SR	cROM	SR	cROM	SR
3	122	127	4.35	3.83	9.39	7.19	9.35	8.77
5	132	139	2.23	1.78	4.59	3.18	8.42	7.82
7	143	147	1.84	0.99	3.16	1.92	7.37	7.25
9	146	157	0.85	0.61	1.88	1.32	8.10	6.86
10	148	160	0.55	0.43	1.77	1.13	7.17	6.08
15	170	174	0.37	0.31	0.80	0.59	7.72	6.30
20	173	182	0.34	0.33	0.64	0.59	6.92	5.94

Table 4.4: Error and speedup varying N_r at time $T = t_{40} \equiv 0.4s$.

Analyzing the results column by column, we observe that the number of magic points increases with for both methods. This is expected, as a larger basis captures more complex dynamics, requiring more points to effectively minimize the NNLS residual. The SR approach generally yields lower errors than the classical cROM. However, the error appears to converge and saturate as N_r increases. Regarding computational efficiency, the speedup is calculated based on the total time required to update the solution at each time step.

In summary, the SR approach consistently outperforms standard cROM. The best trade-off between accuracy, compressibility and efficiency for this test case is achieved with $N_r = 7$. We therefore perform a error analysis by fixing $N_r = 7$ and varying the number of magic points N_m.

	Err h [%]		Err u [%]		Speedup [×]	
N_m	cROM	SR	cROM	SR	cROM	SR
140	1.89	1.03	3.24	1.97	7.64	7.28
130	1.93	1.03	3.29	1.98	7.92	7.98
120	1.92	0.99	3.40	1.91	9.22	8.16
110	1.86	0.93	3.34	1.93	9.43	8.95
100	1.88	1.15	3.31	1.97	9.65	8.95
90	1.81	1.17	3.28	1.98	11.98	10.58
80	NaN	1.18	NaN	2.04	NaN	11.56

Table 4.5: Error and speedup varying N_m ($N_r = 7$ fixed).

Table 4.5 demonstrates that reducing the number of magic points is significative, as the error increases only slightly compared to the significant speedup gained. Moreover, the last row is particularly relevant: with only 80 magic points, the standard cROM fails (producing negative values for h), while the SR approach remains stable, yielding less accurate but still acceptable values.

This test case demonstrates the strong potential of the ROM technique. Since nonnegative water height values occurs marginally in this introductory example, we now extend our analysis to a more challenging test case with a non-flat basin.

4.2.2 Topography: nonflat basin

In this section, we introduce the study of the wet-dry transition, involving non smooth topography. This problem, originally proposed in [35] and also studied in [36], involves the propagation of two rarefaction waves moving in opposite directions. Their interaction produces discontinuities and leads to the formation of a dry bed in the middle of the domain. The spatial domain is [0,25] with a nonflat bottom topography defined as:

$$z(x) = \begin{cases} 7, & x \in (\frac{25}{3}, \frac{25}{2}), \\ 6, & \text{elsewhere.} \end{cases} \tag{4.15}$$

Concerning the initial conditions, we have

$$q(x,0) = \begin{cases} -300, & \text{if } x \geq \frac{50}{3}, \\ 300, & \text{if } x < \frac{50}{3} \end{cases}$$

and

$$h(x,0) := \eta(x,0) - z(x) = 16 - z(x),$$

where z is the bottom topography, h is the total water height (from bottom to the free surface) and η is the free surface elevation ($z + h$). According to [35, 36, 23],

the simulation is run until the final time $T = 0.25$. The spatial discretization is performed using $N = 1000$ cells, and the CFL parameter is set to 0.9. The FV discretization is chosen implementing a hydrostatic reconstruction [37] and the HLL method [10]. The hydrostatic reconstruction, obtained in the asymptotic limit, provides an approximation of the source term. This approach slightly modifies the fluxes, which are then resolved via the HLL method. Overall, this reconstruction leads to a well-balanced scheme that preserves the lake-at-rest condition 2.12 (see Appendix B for details). This test case represents a first example of a problem that is parametric in a strict sense, as it now depends on a physical parameter, distinct from time. We consider two potential options for the analysis of parameter $p \in \mathbb{P}$: the height of the bottom step and its width.

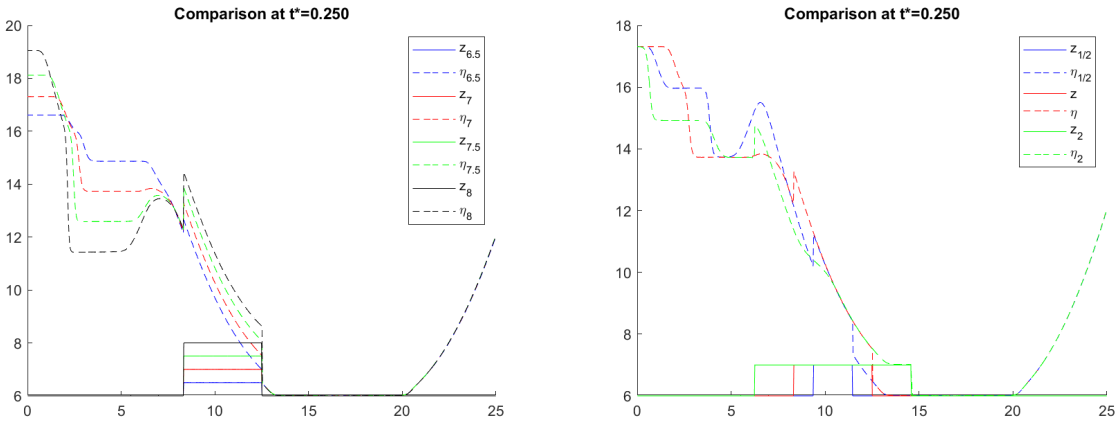

Figure 4.13: High-fidelity solutions for different topography parameters: water surface elevation η for different step heights z (left) and for varying step widths (right).

Figure 4.13 displays the behaviour of HF solutions for $\eta = h + z$ at final time T varying the topography settings. In particular, the left panel varies the central step height within $\{6.5, 7, 7.5, 8, 8.5\}$, while the right panel modifies the step width by halving or doubling the original interval. The qualitative similarity among solutions in the left panel stems from the fixed positions of the discontinuities in the domain, suggesting this parameter range is suitable for ROM applications. We therefore focus on varying the step height within the parametric space $\mathbb{P} = [6,9]$, discretized as:

$$\mathbb{P}_h = \{6, 6.5, 6.75, 7, 7.25, 7.5, 7.75, 8, 8.25, 8.5, 9\}.$$

The parameter space is partitioned into in-sample and out-of-sample values:

$$\mathbb{P}_{\text{in}} = \{6.5, 7, 7.5, 8, 8.5\}, \quad \mathbb{P}_{\text{out}} = \{6, 6.75, 7.25, 7.75, 8.25, 9\}.$$

We collect $N_s = 100$ snapshots for each in-sample parameter value.

Figure 4.14 compares solutions at $t_{30} = 0.2249$s (the 30th snapshot) for $N_r = 10$ and $N_m = 300$, showing the high-fidelity solution η, its projection η_{proj}, and the SR-approximated solution η_{SR}. The projection is evaluated as:

$$\eta_{\text{proj}} = \mathbf{\Phi}\mathbf{\Phi}^{\boldsymbol{T}}\mathbf{W}_{\epsilon}\eta_{\text{snap}},$$

where η_{snap} corresponds to the HF solution at the selected time snapshot (t_{30}). The classical cROM fails due to negative values, while the LE transformation for h causes unbounded growth that diverges from the physical solution. In contrast, the SR method accurately captures the basin emptying and reproduces the solution behaviour on the left of the step.

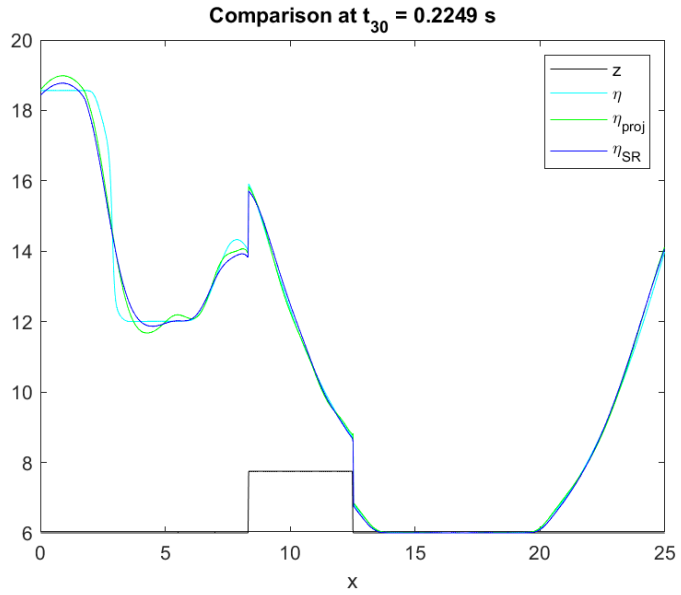

Figure 4.14: High-fidelity solution η, its projection η_{proj}, and the SR-approximated solution η_{SR} at time $t_{30} = 0.2249s$.

Figure 4.15 illustrates the relative L^1 error for h. The projection error provides a theoretical lower bound, while the cROM+SR error remains low within the green region (in-sample parameters interval) but increases in the red region (outside in-sample parameters interval). An interesting and counterintuitive observation concerns parameters within the red regions. Flat topography, i.e. $p = 6$, yields lower error than $p = 9$ despite the POD basis being constructed from nonflat topography snapshots.

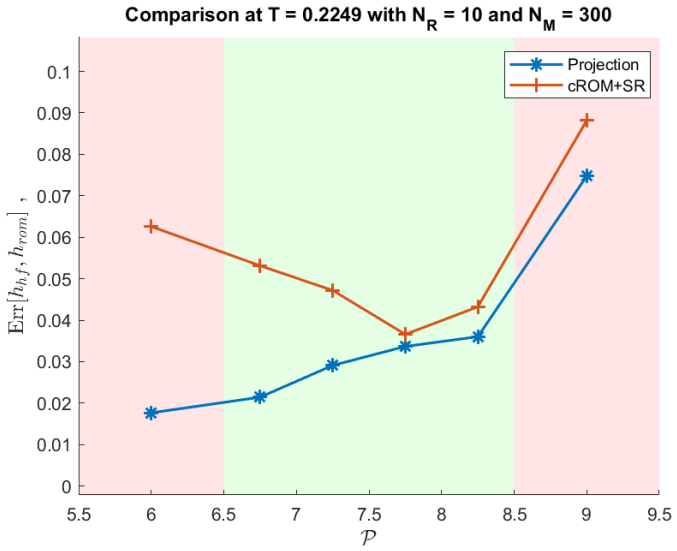

Figure 4.15: Relative L^1 error for h varying height step within $\mathbb{P}_{\text{out}}$ set.

The following Tables present an error analysis for the out-of-sample parameter $p = 7.75$ within the green region.

Table 4.6 shows results for varying N_r with $N_m = 300$, obtained via NNLS. In this specific case, the NNLS tolerance is set $\epsilon_{\text{NNLS}} = 10^-6$ in order to reach convergence. In the cROM+SR approach, 99.9% of the system energy is captured by just 8 modes for both h and q. With $N = 1000$ cells, $N_m = 300$ represents a substantial fraction of the domain, necessary to achieve satisfactory accuracy but resulting in modest speedups that remain below $2\times$. Note that we omit the column for maximum magic points since the NNLS algorithm produces exactly 300 points for each N_r.

As mentioned, the projection error provides a theoretical lower bound, representing the optimal reconstruction achievable. In contrast, the actual cROM approach accumulates errors at each time step. Analysis of the relative L^1 errors for h and q reveals that $N_r = 10$ provides the best trade-off between accuracy and computational efficiency. Interestingly, if we consider a large number of basis functions, this leads to higher relative errors. This behaviour originates from

the higher modes: due to their high-frequency nature, they introduce spurious oscillations that deteriorate the solution accuracy.

	Err h [%]		Err q [%]		
N_r	Proj	cROM	Proj	cROM	Speedup [×]
8	6.58	8.10	6.16	7.51	1.22
10	3.46	4.01	3.16	3.55	1.20
15	2.13	4.58	1.96	3.58	1.46
20	1.59	9.15	1.39	7.83	1.10
30	1.09	8.86	0.86	8.16	1.17

Table 4.6: Error and speedup varying N_r at time $T = 0.25s$, set $p = 7.75$ and $N_m = 300$.

Fixing $N_r = 10$, Table 4.7 shows the effect of reducing the number of magic points. As expected, decreasing N_m improves computational speedup but results in increased approximation error. For $N_m = 150, 200$, the algorithm remains functional but produces significantly larger errors.

	Err h [%]		Err q [%]		
N_m	Proj	cROM	Proj	cROM	Speedup [×]
250	3.46	4.63	3.17	4.07	1.49
200	3.46	15.63	3.17	10.64	1.50
150	3.46	16.07	3.17	11.25	2.33

Table 4.7: Error and speedup varying N_m at time $T = 0.25s$, set $p = 7.75$ and $N_r = 10$.

These analyses highlight the complexity of wet-dry transition problems and underscore some relevant observations. First, this test case exhibits sensitivity to parameter settings, producing degradation with increasing basis functions due to introduced oscillations. However, our study demonstrates that the SR approach remains satisfactory and stable compared to both classical cROM (which fails due to negative values) and LE transformation (which suffers from solution blow-up). This test case clearly deserves deeper investigation, particularly regarding the development of enhanced ROM. The SR method emerges as a promising approach for such future work with 2D and 3D problems.

Chapter 5

Conservation property in ROMs

5.1 Conservation property

In Chapter 2, we derived FV methods from the integral form of PDEs. While such discretization techniques numerically enforce conservation over each control volume, standard ROMs often fail to preserve this property, being inherently non-conservative. Key works in the literature addressing this issue include [25], which proposed a conservative model reduction framework that assimilates the structure intrinsic to FV methods, and [26], which applied conservative ROM techniques directly to fluid flows. This section investigates an alternative approach to guarantee global conservation in the collocated reduced-order model (cROM). It is crucial to note that we exclusively consider standard cROM formulation, not involving the LE and SR transformations described previously. Introducing such nonlinear transformations is problematic, as it can further complicate the already persistent non-conservative state. A perspective future direction is to develop techniques that combine cROM to preserve properties like positivity with the essential structure-preserving aspect of these governing laws, namely the conservation.

In the present discussion, we employ a specific choice for the offset term in equation (3.4), which we now proceed to explain. Let us consider the approximate solution:

$$u_h(x,t;\vartheta) \simeq M_0 + \sum_{k=1}^{N_r} \alpha_k(t;\vartheta)\phi^k(x), \tag{5.1}$$

where M_0 is the offset quantity to be determined.

Integrating over the spatial domain of the equation Ω, this yields to:

$$\int_\Omega u_h dx \simeq M_0|\Omega| + \sum_{k=1}^{N_r} \alpha_k \int_\Omega \phi^k dx. \tag{5.2}$$

To enforce that the integral of the discrete solution remains invariant over time, the integral of each basis function ϕ^k must vanish:

$$\int_\Omega \phi^k dx = 0, \qquad \text{for } k = 1, \dots, N_r.$$

This requirement is satisfied by choosing the offset such that:

$$\int_\Omega u_h dx - M_0|\Omega| \simeq 0. \tag{5.3}$$

Performing a FV discretization over a one-dimensional domain of length L , at a generic time step t^n, this leads to:

$$\sum_{i=1}^{N} (\mathbf{u}_{\text{HF}})_i^n \Delta x \simeq M_0 L. \tag{5.4}$$

Thus, the initial mass offset is selected as:

$$M_0 \simeq \frac{1}{L} \sum_{i=1}^{N} (\mathbf{u}_{\text{HF}})_i^n \Delta x \tag{5.5}$$

Since the high-fidelity solution is computed using a conservative scheme, and assuming the total "mass" remains unchanged, the discrete integral:

$$\sum_{i=1}^{N} (\mathbf{u}_{\text{HF}})_i^n \Delta x = \text{constant} \tag{5.6}$$

is constant in time. Consequently, we can compute the snapshot adjusted removing the offset as:

$$\mathbf{s} = \mathbf{u}_{\text{HF}}^n - \frac{1}{L} \sum_{i=1}^{N} (\mathbf{u}_{\text{HF}}^n)_i \Delta x = \mathbf{u}_{\text{HF}}^n - \mathbf{M}_0 \tag{5.7}$$

where $\mathbf{M}_0$ is the vector whose entries are all equal to the computed mean value.

This formulation directly influences the architecture of the cROM, where the updated solution is now computed as follows:

$$\mathbf{u}_c^{n+1} = \mathbf{M}_0 + \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{W}_\epsilon [\mathcal{L}(\mathbf{u}_c^n) - \mathbf{M}_0] \tag{5.8}$$

In this procedure, we remove the offset quantity before computing the updated solution through the $\mathcal{L}$ and finally we add the same quantity to recover the updated cROM solution at the next time t^{n+1}.

However, the assumption of constant integral deserves more attention, as the integral may vary if the conserved quantity depends on the inflow and outflow at the domain boundaries. We therefore distinguish between two main test cases: constant "mass" and variable "mass". In particular, we analyze the linear advection and SW equations under the constant "mass" assumption, while also performing a SW simulation where the mass changes over the domain.

5.1.1 Linear Advection Equation

Initially we carry out a conservation analysis on the linear advection equation. We employ the same parameters as in Section 4.1, i.e. number of cells $N = 2000$, number of snapshots $N_s = 40$, number of magic points $N_m = 200$ and number of basis functions $N_r = 10$. Since the integral over the domain is constant equal to 0.1 and the domain length $l = 1$, thus the offset corresponds to $M_0 = 0.1$. The final time is set to $T = 0.4$ and cour $= 0.8$.

The effect of the offset is firstly provided by Table 5.1 which compares the integral $\int_\Omega$ for the first 5 basis functions with or without offset. Although the introduction of the offset is significant, the integrals are not close to the machine precision which is a primary requirement to validate our analysis.

$\int dx$	No Offset	Offset
ϕ^1	-0.014010309174975	-8.981729068862609e-09
ϕ^2	2.179715073278799e-05	4.559809686161300e-06
ϕ^3	0.005282810172803	8.385788242941091e-06
ϕ^4	2.357551603267199e-06	5.063161192765235e-09
ϕ^5	-0.003980955565185	4.742157153662557e-06

Table 5.1: Comparison of the integral of the first 5 basis functions including no offset and offset on linear advection equation.

To visualize the temporal evolution of the integral up to final time $0.4s$, we plot in Figure 5.1 the comparison of HF and cROM solutions when the offset is considered or not. While the HF integral of u is equal to 0.1 accordingly to previous discussion, we observe a significant improvement, gaining approximately three orders of magnitude, when we introduce the offset. This first example provides a preliminary application of the idea, showing that the order of accuracy in terms of conservation can be significantly improved. This promising validation on the linear advection equation encourages the application of the method to more complex systems, such as the SW equations, where conservation is still critical.

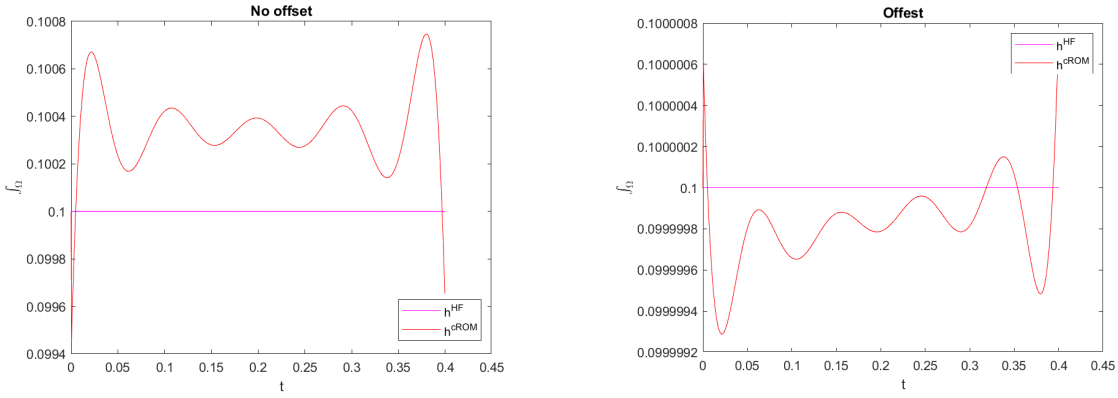

Figure 5.1: Comparison of the integral varying over time without offset (left) and with offset (right).

5.1.2 Shallow Water: constant "mass"

In this part, we rely on SW equations (2.10) without source term. We present a Riemann problem with the following initial conditions:

$$h(x,0) = \begin{cases} 0.5, & x_{\text{in}} < x < x_c^-, \\ 1, & x_c^- < x < x_c^+, \\ 0.5, & x_c^+ < x < x_{\text{end}}, \end{cases} \tag{5.9}$$

and

$$q(x,0) = 0, \qquad x_{\text{in}} < x < x_{\text{end}}. \tag{5.10}$$

The spatial domain is $(x_{\text{in}}, x_{\text{end}}) = (-3,3)$, while the coefficients $x_c^- = -0.5$ and $x_c^+ = 0.5$. The final time is $T = 0.4s$. Employing a spatial discretization with $N = 3000$ cells and cour $= 0.9$, we run the simulation for the standard cROM approach using $N_r = 50$, $N_m = 300$ and $N_s = 120$. Since the domain length is 6 and the integral of the initial condition for h is 3.5, the constant offset is $M_0 = 3.5/6$. In the following Table 5.2, we compare the integral of basis functions with or without the introduction of the offset.

$\int dx$	No Offset	Offset
ϕ^1	0.108146569958359	1.584010700383942e-16
ϕ^2	0.007688588823427	8.645861804268407e-17
ϕ^3	0.005158369892580	3.842065554593432e-17
ϕ^4	0.003702278536896	2.748842820032849e-17
ϕ^5	0.001614328223451	1.897960955066225e-17

Table 5.2: Comparison of the integral of the first 5 basis functions for h including no offset and offset on SW equations (constant mass).

Table 5.2 shows that the presence of the offset leads to basis function integrals that are numerically zero, on the order of machine epsilon. Let us analyze the time evolution of the integral computed for the HF solution and the cROM solution in Figure 5.2. Without offset, the global quantity h exhibits a relevant "mass" loss over time; conversely, the right view demonstrates the effectiveness of the offset, with variations on the order of $10^{-13}/10^{-14}$. Note that the HF solution integral is not exact (the real value is 3.5), resulting in a slight error.

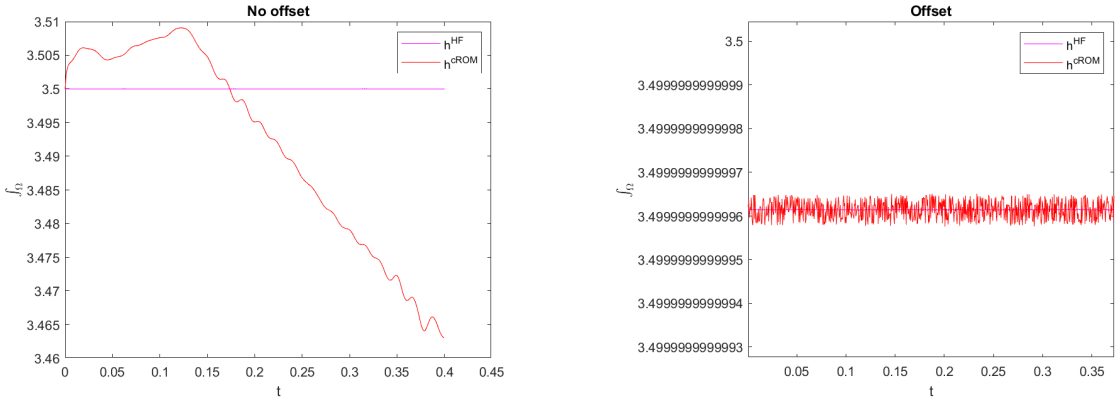

Figure 5.2: Comparison of the integral varying over time without offset (left) and with offset (right).

This second example demonstrates that the introduction of the offset yields excellent results and opens a possible application involving non constant offset.

5.1.3 Shallow water: variable "mass"

In this test, we consider again SW equations with a suitable problem varying its "mass" inside the domain. We examine a Riemann problem that generates two

rarefaction waves. The initial data are given by:

$$h(x,0) = 1, \qquad x_{\text{in}} < x < x_{\text{end}} \tag{5.11}$$

and

$$u(x,0) = \begin{cases} u_L = -2, & x_{\text{in}} < x < x_c \\ u_R = 2, & x_c < x < x_{\text{end}} \end{cases} \tag{5.12}$$

where $x_{\text{in}} = -1.5$, $x_{\text{end}} = 1.5$ and $x_c = 0$. The final time is set to $T = 0.2s$. In this scenario, we define the snapshots similar to formula (5.7):

$$\mathbf{s} = \mathbf{u}^n_{\text{HF}} - \frac{1}{N}\sum_{i=1}^{N}(\mathbf{u}^n_{\text{HF}})_i \tag{5.13}$$

It is essential to note that the integral over the domain of the HF solution changes at each time step. Therefore, the mean value subtracted from each snapshot is not a global constant but is computed individually for each snapshot. This process ensures that all the basis functions have a zero mean. For the online stage of the cROM, the offset is now time-dependent. To determine its evolution, we start from the definition of the total mass M. Its rate of change is given by:

$$\frac{dM}{dt} = \frac{d}{dt}\int_\Omega h(x,t)dx = \int_\Omega \partial_t h(x,t)dx = -\int_\Omega \partial_x F(x,t)dx \tag{5.14}$$

Applying the divergence theorem to the last term, we obtain:

$$\frac{dM}{dt} = -\int_{\partial\Omega} F(x,t)dx. \tag{5.15}$$

Since we work on one-dimensional domain, we consider $\Omega = (a,b)$; thus, this results in

$$\frac{dM}{dt} = F(a,t) - F(b,t). \tag{5.16}$$

Performing a discretization of the latter equation in time with a simple Euler scheme and introduction a flux function provides the update rule for the discrete mass:

$$M^{n+1} = M^n + (f^n_{1/2} - f^n_{N+1/2}), \tag{5.17}$$

where $f^n_{1/2}$ and $f^n_{N+1/2}$ are the numerical fluxes at the left and right domain boundaries, respectively, at time step $t^n = n\Delta t$.

In this variable "mass" scenario, the solution is computed as follows:

$$\mathbf{u}^{n+1}_c = \mathbf{M}^{n+1} + \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{W}_\epsilon[\mathcal{L}(\mathbf{u}^n_c) - \mathbf{M}^{n+1}] \tag{5.18}$$

where $\mathbf{M}^n$ is a vector whose entries are all equal to the computed mean value at time t^n.

$\int dx$	No Offset	Offset
ϕ^1	-0.065967062287350	3.280709037767337e-17
ϕ^2	-0.009224251340041	2.810252031082428e-18
ϕ^3	-0.005839060643247	-4.292399768957011e-17
ϕ^4	0.003797630976401	-3.699124340172944e-17
ϕ^5	0.002773251029157	-7.635558851859514e-17

Table 5.3: Comparison of the integral of the first 5 basis functions for h including no offset and offset on SW equations (variable mass).

The introduction of the offset is extremely significative, as shown in Table 5.3. The integrals of the first basis functions are reduced to the order of machine precision. Recalling formula (5.16), the rate of change of "mass" for this problem is given by:

$$\frac{dM}{dt} = F(h(x_{\text{in}}, t)) - F(h(x_{\text{end}}, t)). \tag{5.19}$$

The flux function for h is $F_h = hu$. Given the constant initial states at the boundaries $h = 1$, $u_L = -2$ and $u_R = 2$, we obtain the boundary fluxes:

$$\begin{aligned} F_h(x_{\text{in}}, t) &= -2, \\ F_h(x_{\text{end}}, t) &= 2. \end{aligned} \tag{5.20}$$

Consequently, the exact "mass" evolution is described by the linear function:

$$M(t) = 3 - 4t.$$

Figure 5.3 confirms the expected trend: a significant difference exists between the cases with and without the offset. Specifically, the simulation without the offset overestimates the true integral of h, whereas the simulation with offset matches the linear decreasing evolution. To quantify this effect, we essentially examine the integral deviation between h^{HF} and h^{cROM} in Figure 5.4. This analysis reveals that introducing the offset produces approximately one order of magnitude in terms of conservation accuracy compared to the case without it.

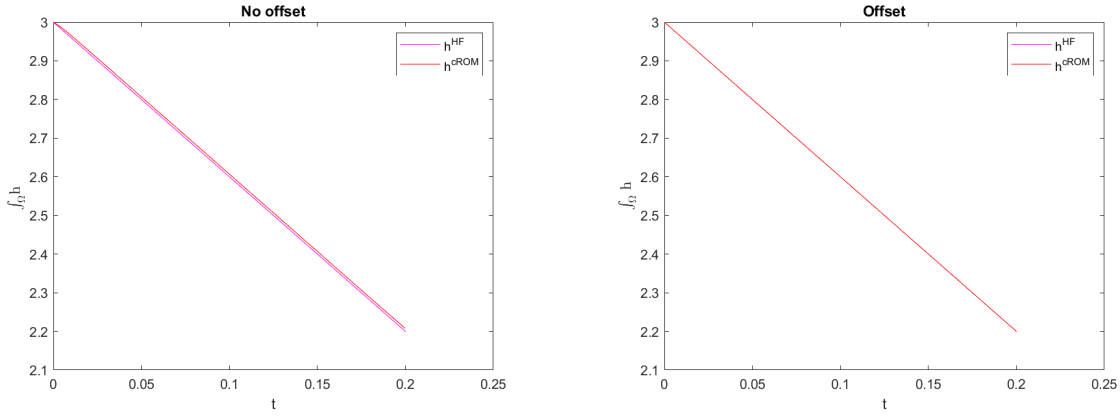

Figure 5.3: Comparison of the h integral varying over time without offset (left) and with offset (right).

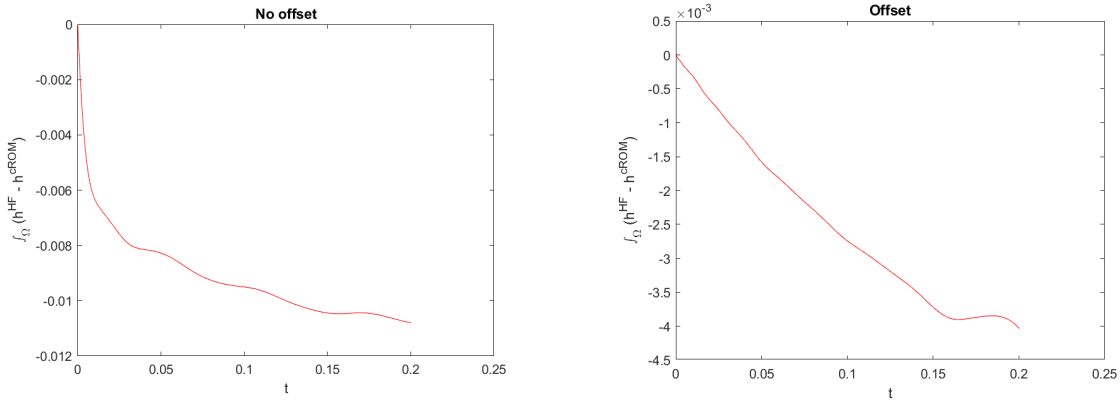

Figure 5.4: Comparison of $\int_\Omega (h^{HF} - h^{cROM})$ varying over time without offset (left) and with offset (right).

This final example exhibits promising feedback for future works and discussions. Although the current improvement is marginal, this offset introduction deserves deeper investigation to evaluate its effectiveness on complex problems such as higher-dimensional SW systems or the Euler equations.

To summarize, introducing the offset achieved the outcome of driving the integral of basis functions close of zero. Despite not obtaining integral values on the order of machine precision for linear advection equation, we attained notable results for SW equations. The accuracy of the conservation is certainly improved by the inclusion of the offset, but it only provided excellent results in the case of SW, where the error was close to the order of machine precision. The conservation improvement with offset showed initial and promising findings but it deserves several tests to effectively validate this introductory hypothesis.

Chapter 6

Conclusions

In this work, we addressed the challenge of maintaining intrinsic properties of hyperbolic PDE systems, such as positivity and conservation.

The application of transformations such as Logarithm-Exponential (LE) and square-root (SR) has produced significant results: while for the linear transport equation the SR approach yields results comparable to those of cROM, successfully maintaining a positive solution over time, the LE approach shows that the introduction of a shift is inappropriate for preserving positivity. For the shallow water equations in a non-transonic rarefaction scenario, we model the SR transformation on both conservative variables h and $q = hu$. This approach demonstrates improved accuracy and computational speed compared to classical ROM, particularly in an example where issues related to near-zero values do not directly arise. In the final test, the SR strategy proves effective even in a more complex topography setting involving a non-flat basin. Here, the SR transformation enables accurate approximation of the solution, whereas the classical ROM fails due to the presence of negative values and the LE approach produces numerical blow-up.

Regarding conservation, the first tests carried out with the introduction of this offset verified some expected properties, such as the vanishing integral of the first basis functions, and showed a noticeable but still limited improvement in overall conservation.

6.1 Future Perspectives

These promising initial results pave the way for several research directions. Future work will focus on different primary fronts:

- **Structure-Preserving ROMs**: this task pursues the initial contribution of this work, aiming to develop ROMs capable of preserving sharp fronts and discontinuities, as well as well-balanced and asymptotic-preserving ROMs. A

key objective is to extend these goals on 2D-3D dimensional problems, such as the shallow water or Euler equations;

- **Improvement of Reduced Basis Construction**: this task aims to to enhance the offline phase by studying the influence of different norms employed in the POD, choosing the most appropriate function space for the case of conservation laws, minimizing the number of basis functions needed whilst keeping a good level of expressivity, and optimise the choice of parameter values used to obtain the snapshots;
- **Error estimation and Uncertainty Quantification**: this final front aims to develop error estimators for ROMs of conservation laws, providing confidence intervals for ROM output and studying the interaction between different types of error.

Appendix A

Convergence test on shallow water

In this appendix, we perform a convergence test for a shallow water Riemann problem characterized by two rarefaction waves. The initial conditions are defined as:

$$h(x,0) = 0.5, \qquad x \in [x_{\text{in}}, x_{\text{end}}] \tag{A.1}$$

and

$$u(x,0) = \begin{cases} u_L = -2, & x_{\text{in}} \leq x < x_c \\ u_R = 2, & x_c < x \leq x_{\text{end}} \end{cases} \tag{A.2}$$

where $x_{\text{in}} = 0$, $x_{\text{end}} = 1$ and $x_c = 0.5$. The final time is set to $T = 0.1s$. We apply a finite volume discretization, refining the mesh over the following set of cells:

$$N = \{500, 1000, 2000, 4000\}.$$

Figure A.1 compares the L^1 error for the conservative variable h and q. The results confirm the suitability of the HLL solver, as the error decreases systematically with mesh refinement, demonstrating an expected convergence behaviour as Δx is reduced.

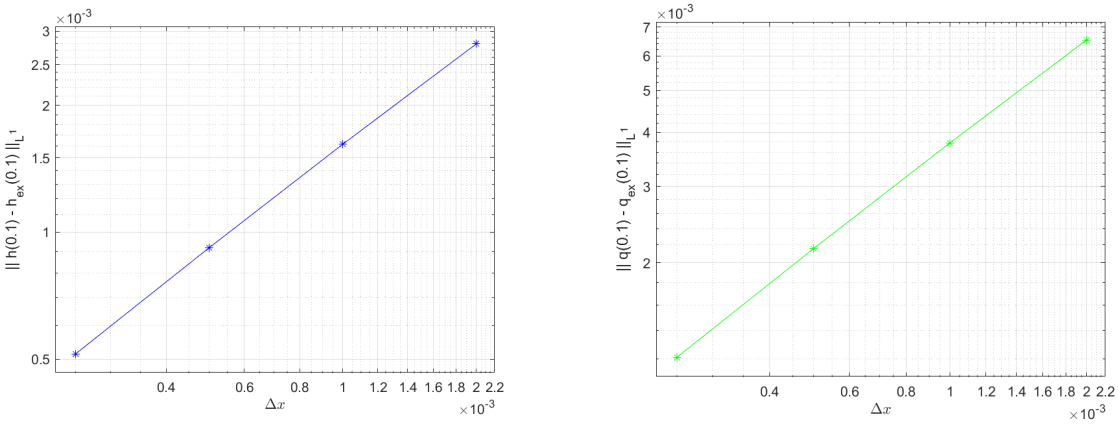

Figure A.1: L^1 error comparison under mesh refinement for the conservative variables h (left) and $q = hu$ (right).

Δx	Error L^1 h	Error L^1 q
0.0020	0.002795871778442	0.006535502103863
1e-03	0.001616166276042	0.003784732742197
5e-04	9.183423662733807e-04	0.002156915988136
2.5e-04	5.144137776586987e-04	0.001211968370247

Table A.1: Numerical values corresponding to the convergence study in Figure A.1.

Through the MATLAB command `polyfit`, we estimate the experimental order of convergence based on the data in Table A.1. The resulting order is approximately 0.81 for both conservative variables. This value indicates a reasonable convergence rate, especially considering that the HLL solver is applied to a non-trivial test case involving two rarefaction waves. These results support the choice of the HLL scheme for the simulations discussed in this work, particularly in the context of shallow water equations.

Appendix B

Lake at rest

In this appendix, we test the preservation of the "lake at rest" steady state, a fundamental property for numerical schemes solving the SW equations. We present three test cases with different topographies (linear, smooth, and bump) to demonstrate the effectiveness of hydrostatic reconstruction implementation combined with HLL solver.

B.1 Linear Topography

For the first test, we consider the following initial conditions and a liner bottom with slope:

$$\eta(x,0) = 1.1, \quad u(x,0) = 0, \quad z(x) = 0.5x. \tag{B.1}$$

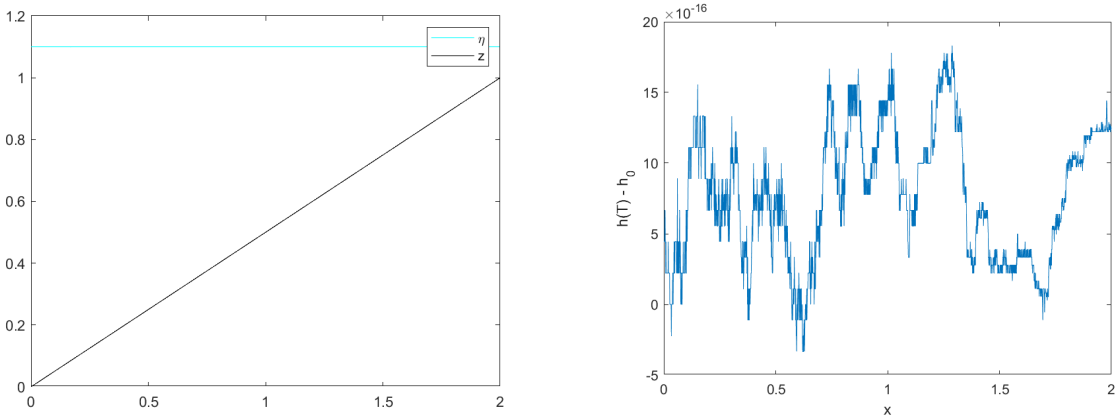

Figure B.1: Lake at rest state (left) and deviation from initial state $h(T) - h(0)$ (right) at time $T = 2s$ with linear topography (hydrostatic reconstruction + HLL solver).

The computational domain is [0,2], simulated until a final time $T = 2s$ with a CFL number of 0.8. We perfrom spatial discretization using $N = 2000$ cells. Figure B.1 shows the numerical solution at $T = 2s$. The free surface elevation $\eta = h + z$ remains perfectly flat. The right panel shows that the deviation from the initial water height is on the order of 10^{-16}, confirming that the lake-at-rest condition is preserved to machine precision.

B.2 Smooth Topography

This test features a continuous piecewise topography with a smooth quadratic section:

$$\eta(x,0) = 2.1, \quad u(x,0) = 0, \quad z(x) = \begin{cases} 0.5x^2, & x > 0 \\ 0, & x < 0. \end{cases} \tag{B.2}$$

The computational domain is $[-2,2]$, with final time $T = 2$s, CFL $= 0.8$, and $N = 2000$ cells. As shown in Figure B.2, the free surface remains level. The error

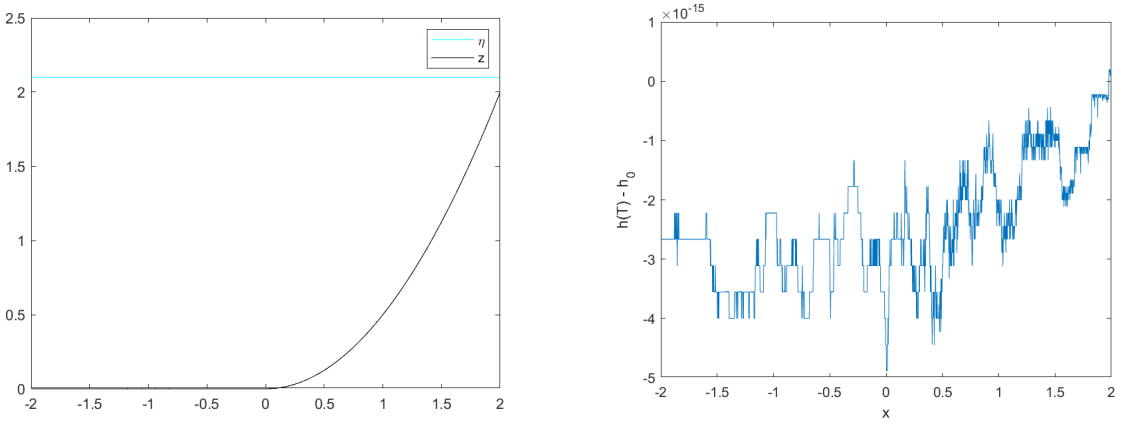

Figure B.2: Lake at rest state (left) and deviation from initial state $h(T) - h(0)$ (right) at time $T = 2s$ with smooth topography (hydrostatic reconstruction + HLL solver).

plot (right) confirms preservation of the steady state, with deviations on the order of 10^{-15}.

B.3 Bump topography

The final test involves a discontinuous bottom topography with a bump, referring to [38]:

$$\eta(x,0) = 3, \quad u(x,0) = 0, \quad z(x) = \begin{cases} 2 + f(x), & 1.4 < x < 1.6 \\ 2, & \text{otherwise.} \end{cases} \tag{B.3}$$

where $f(x) = 0.25(\cos(10\pi(x - 0.5)) + 1)$. The computational domain is [0,2], with final time $T = 0.2s$, CFL $= 0.8$, and $N = 2000$ cells. Figure B.3 demonstrates the

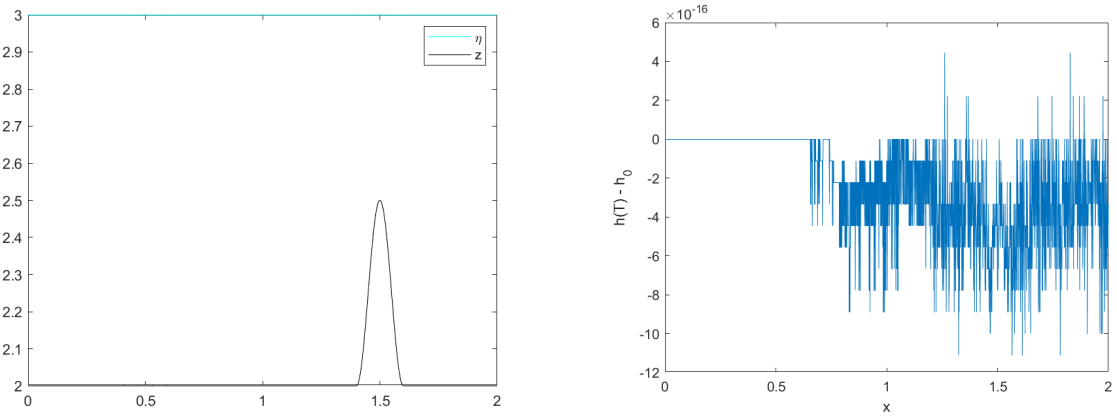

Figure B.3: Lake at rest state (left) and deviation from initial state $h(T) - h(0)$ (right) at time $T = 2s$ with bump topography (hydrostatic reconstruction + HLL solver).

scheme is still capable to handle this complex bottom profile. The error remains on the order of 10^{-16}, proving that the steady state is perfectly preserved.

All three tests consistently show that HLL solver comined with the hydrostatic reconstruction preserves the lake-at-rest state to machine precision ($\sim 10^{-15}$–10^{-16}). This is achieved even in the presence of different topographies, validating its effectiveness as a robust and well-balanced scheme.

Bibliography

[1] Susanne C Brenner and L Ridgway Scott. *The mathematical theory of finite element methods*. Springer, 2008 (cit. on p. 1).

[2] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012 (cit. on p. 1).

[3] Robert Leroy Taylor and JZ Zhu. *The Finite Element Method: Its Basis and Fundamentals: Its Basis and Fundamentals*. Butterworth-Heinemann, 2005 (cit. on p. 1).

[4] Claudio Canuto, M Youssuff Hussaini, Alfio Quarteroni, and Thomas A Zang. *Spectral methods: fundamentals in single domains*. Springer, 2006 (cit. on p. 1).

[5] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001 (cit. on p. 1).

[6] David Gottlieb and Steven A Orszag. *Numerical analysis of spectral methods: theory and applications*. SIAM, 1977 (cit. on p. 1).

[7] Bernardo Cockburn, George E Karniadakis, and Chi-Wang Shu. *Discontinuous Galerkin methods: theory, computation and applications*. Vol. 11. Springer Science & Business Media, 2012 (cit. on p. 1).

[8] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer, 2008 (cit. on p. 1).

[9] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*. Vol. 69. Springer Science & Business Media, 2011 (cit. on p. 1).

[10] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 2nd. Berlin, Heidelberg: Springer, 1997. ISBN: 978-3-540-65966-3 (cit. on pp. 1, 2, 4, 9, 15, 42, 47).

[11] Randall J. LeVeque. *Numerical Methods for Conservation Laws*. 2nd. Lectures in Mathematics ETH Zürich. Basel: Birkhäuser, 1992. ISBN: 3-7643-2723-5. DOI: 10.1007/978-3-0348-8629-1 (cit. on pp. 1, 2, 9, 11).

[12] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press, 2002. ISBN: 978-0-521-00924-9 (cit. on pp. 1, 2, 9, 11, 15).

[13] Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. 1st ed. Springer Briefs in Mathematics. Switzerland: Springer, 2015, p. 135. ISBN: 978-3-319-22469-5. DOI: 10.1007/978-3-319-22470-1 (cit. on pp. 1, 21).

[14] Jan S Hesthaven, Cecilia Pagliantini, and Gianluigi Rozza. «Reduced basis methods for time-dependent problems». In: *Acta Numerica* 31 (2022), pp. 265–345 (cit. on p. 1).

[15] Sébastien Riffaud. «Reduced-order models : convergence between scientific computing and data for fluid mechanics». Theses. Université de Bordeaux, Dec. 2020. URL: https://theses.hal.science/tel-03156427 (cit. on pp. 1, 21, 24–26).

[16] Anindya Chatterjee. «An introduction to the proper orthogonal decomposition». In: *Current science* (2000), pp. 808–817 (cit. on p. 1).

[17] YC Liang, HP Lee, SP Lim, WZ Lin, KH Lee, and CG1237 Wu. «Proper orthogonal decomposition and its applications—Part I: Theory». In: *Journal of Sound and vibration* 252.3 (2002), pp. 527–544 (cit. on p. 1).

[18] Bruce Moore. «Principal component analysis in linear systems: Controllability, observability, and model reduction». In: *IEEE transactions on automatic control* 26.1 (2003), pp. 17–32 (cit. on p. 2).

[19] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. «Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics». In: *Archives of Computational Methods in Engineering* 15.3 (2008), pp. 229–275 (cit. on p. 2).

[20] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. «Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations». In: *International Journal for numerical methods in engineering* 86.2 (2011), pp. 155–181 (cit. on p. 2).

[21] Michel Bergmann, Michele Giuliano Carlino, and Angelo Iollo. «Model Order Reduction Using a Collocation Scheme on Chimera Meshes: Addressing the Kolmogorov-Width Barrier». In: *SIAM Journal on Scientific Computing* 47.4 (2025), A2272–A2298 (cit. on pp. 2, 26).

[22] Sergio Blanes, Arieh Iserles, and Shev Macnamara. «Positivity-preserving methods for ordinary differential equations». In: *ESAIM: Mathematical Modelling and Numerical Analysis* 56.6 (2022), pp. 1843–1870 (cit. on p. 2).

[23] Emmanuel Audusse, Christophe Chalons, and Philippe Ung. «A simple well-balanced and positive numerical scheme for the shallow-water system». In: *Communications in Mathematical Sciences* 13.5 (2015), pp. 1317–1332 (cit. on pp. 2, 46).

[24] Jad Dabaghi and Virginie Ehrlacher. «Structure-preserving reduced order model for parametric cross-diffusion systems». In: *ESAIM: Mathematical Modelling and Numerical Analysis* 58.3 (2024), pp. 1201–1227 (cit. on pp. 2, 35).

[25] Kevin Carlberg, Youngsoo Choi, and Syuzanna Sargsyan. «Conservative model reduction for finite-volume models». In: *Journal of Computational Physics* 371 (2018), pp. 280–314 (cit. on pp. 2, 51).

[26] Babak Maboudi Afkham, Nicolo Ripamonti, Qian Wang, and Jan S Hesthaven. «Conservative model order reduction for fluid flow». In: *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*. Springer, 2020, pp. 67–99 (cit. on pp. 2, 51).

[27] Erhard Schmidt. «Zur Theorie der linearen und nichtlinearen Integralgleichungen». In: *Mathematische Annalen* 63.4 (1907), pp. 433–476 (cit. on p. 25).

[28] Leon Mirsky. «Symmetric gauge functions and unitarily invariant norms». In: *The quarterly journal of mathematics* 11.1 (1960), pp. 50–59 (cit. on p. 25).

[29] Carl Eckart and Gale Young. «The approximation of one matrix by another of lower rank». In: *Psychometrika* 1.3 (1936), pp. 211–218 (cit. on p. 25).

[30] Todd Chapman, Philip Avery, Pat Collins, and Charbel Farhat. «Accelerated mesh sampling for the hyper reduction of nonlinear computational models». In: *International Journal for Numerical Methods in Engineering* 109.12 (2017), pp. 1623–1654 (cit. on p. 27).

[31] Charbel Farhat, Philip Avery, Todd Chapman, and Julien Cortial. «Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency». In: *International Journal for Numerical Methods in Engineering* 98.9 (2014), pp. 625–662 (cit. on p. 27).

[32] Sebastian Grimberg, Charbel Farhat, Radek Tezaur, and Charbel Bou-Mosleh. «Mesh sampling and weighting for the hyperreduction of nonlinear Petrov–Galerkin reduced-order models with local reduced-order bases». In: *International Journal for Numerical Methods in Engineering* 122.7 (2021), pp. 1846–1874 (cit. on p. 27).

[33] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995 (cit. on p. 27).

[34] Camilla Fiorini. «Sensitivity analysis for nonlinear hyperbolic systems of conservation laws». PhD thesis. Université Paris Saclay, 2018 (cit. on p. 43).

[35] Thierry Gallouët, Jean-Marc Hérard, and Nicolas Seguin. «Some approximate Godunov schemes to compute shallow-water equations with topography». In: *Computers & Fluids* 32.4 (2003), pp. 479–513 (cit. on p. 46).

[36] Manuel Castro, Alberto Pardo, Carlos Parés, and E Toro. «On some fast well-balanced first order solvers for nonconservative systems». In: *Mathematics of computation* 79.271 (2010), pp. 1427–1472 (cit. on p. 46).

[37] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoı t Perthame. «A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows». In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2050–2065 (cit. on p. 47).

[38] Christophe Chalons and Alessia Del Grosso. «Exploring different possibilities for second-order well-balanced Lagrange-projection numerical schemes applied to shallow water Exner equations». In: *International Journal for Numerical Methods in Fluids* 94.6 (2022), pp. 505–535 (cit. on p. 65).