TESIS DE LA UNIVERSIDAD DE ZARAGOZA
2025
59

Pablo Solán Fustero

# Reduced-order models based on the proper orthogonal decomposition applied to hyperbolic problems

Director/es
Gracia Lozano, José Luis
Navas Montilla, Adrián

http://zaguan.unizar.es/collection/Tesis

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

# REDUCED-ORDER MODELS BASED ON THE PROPER ORTHOGONAL DECOMPOSITION APPLIED TO HYPERBOLIC PROBLEMS


Autor

Pablo Solán Fustero

Director/es

Gracia Lozano, José Luis
Navas Montilla, Adrián

**UNIVERSIDAD DE ZARAGOZA**
**Escuela de Doctorado**

Programa de Doctorado en Mecánica de Fluidos


2024


**Repositorio de la Universidad de Zaragoza – Zaguan http://zaguan.unizar.es**

Universidad Zaragoza
1542

# Tesis Doctoral

## Reduced-order models based on the proper orthogonal decomposition applied to hyperbolic problems


Autor

Pablo Solán Fustero

Director/es

Adrián Navas Montilla
José Luis Gracia Lozano

Supervisora

Pilar García Navarro

Escuela de Ingeniería y Arquitectura


2024


Repositorio de la Universidad de Zaragoza – Zaguan http://zaguan.unizar.es

*A Carmen, Ignacio y Carlota*

# Agradecimientos

Mi principal agradecimiento va dirigido a mis tutores de la tesis: A Pilar, que, aunque oficialmente ocupa la posición de supervisora, es tutora de facto de esta tesis, por ser mi mentora en cada paso de mi vida investigadora, desde que me recogió de la Facultad de Ciencias y me propuso hacer el TFG. A Adrián, por enseñarme qué es la investigación y cómo se hace, y, sobre todo, por hacerlo desde el cuidado, la confianza y la amistad. A José Luis, por sostener las Matemáticas en esta tesis y por su incansable trabajo, porque si yo he le echado todas las horas posibles, él ha estado ahí atento en todas y cada una de ellas.

Su amplia experiencia, su enorme intuición (que, a día de hoy, me sigue sorprendiendo) y sus actitudes complementarias han permitido levantar una tesis de la nada dentro de un tema del que los cuatro éramos completos desconocedores al principio. Estoy muy satisfecho con el trabajo conjunto que hemos realizado y me siento orgulloso de formar parte de sus brillantes carreras.

Les quiero agradecer al resto de componentes del grupo, Pilar Brufau, Javier Murillo, Mario Morales, Javier Fernández y Sergio Martínez, todo el apoyo y la cercanía que me han brindado estos cuatro años. Tanto directamente, con sus comentarios, sugerencias, códigos, etc., como indirectamente, pues me he leído todas sus tesis y muchos de sus artículos, su trabajo ha sido esencial para poder desarrollar el mío.

He estado muy bien rodeado en la sala de colaboradores durante toda la tesis. En primer lugar, por mis compañeros doctorandos, Juan Mairal y Pablo Vallés, y la ya doctora Isabel Echeverribar, a quienes observo cuidadosamente como guías de conducta en ámbitos que van más allá de la investigación. Les agradezco con todo mi cariño lo mucho que me han enseñado y, por supuesto, las estancias en congresos que hemos podido disfrutar juntos. Y, en segundo lugar, por las nuevas incorporaciones, José Segovia, Nacho Ojer, Víctor Loras y Javier Orera, los cuales han llenado la sala con su buen humor (y buen trabajo) en el momento en que más necesario era.

# Acknowledgements

The author would like to express his gratitude to Prof. Dr. Donsub Rim for interesting discussions on reduced-order models and his valuable review of this thesis; to Prof. Dr. Tommaso Taddei for all his help during his stay at INRIA, Bordeaux; and to Prof. Dr. Mathew Hubbard for his valuable review of this thesis.

This work was funded by the Spanish Ministry of Science and Innovation under the research projects PGC2018-094341-B-I00 and PID2022-141051NA-I00. This work has also been partially funded by Gobierno de Aragón through Fondo Social Europeo (T32-20R and E24-17R, Feder 2019-2021 "Construyendo Europa desde Aragón").

*A-tom in Greek is the same as In-di-viduum in Latin - unsplittable. Those who invented these words knew neither nuclear fission nor schizophrenia - whence only the modern compulsion to splitting into ever smaller parts, splitting off entire parts of the personality from that ancient person conceived as indivisible.*

Christa Wolf, *Accident: A Day's News.*

*Science, being called upon to henceforth represent society's collective consciousness, must really become the property of everybody. Thereby, without losing anything of its universal character, of which it can never divest itself without ceasing to be science, and while continuing to concern itself exclusively with general causes, the conditions and fixed relations of individuals and things, it will become one in fact with the immediate and real life of all individuals.*

Mikhail Bakunin, *God and the State.*

*Break the dam! Release the river!*

Treebeard, *The Two Towers.*

# Contents

**Abstract** **1**

**Resumen** **3**

**Acronyms** **5**

**1 Introduction** **7**

1.1 Motivation . . . . . 7

1.2 State of the art . . . . . 9

1.3 Objectives of this thesis . . . . . 12

1.4 Outline . . . . . 13

1.5 Acknowledgements . . . . . 15

**2 Equations and full-order models** **17**

2.1 1D advection-diffusion equation . . . . . 17

2.1.1 First-order upwind scheme . . . . . 19

2.1.2 Optimized Lax-Friedrichs scheme . . . . . 20

2.1.3 Lax-Wendroff scheme . . . . . 21

2.2 2D advection-diffusion equation . . . . . 21

2.3 1D Burgers' equation . . . . . 23

2.4 1D shallow water equations . . . . . 25

2.4.1 Augmented Roe-based FOM . . . . . 26

2.4.2 Optimized Lax-Friedrichs-based FOM . . . . . 28

2.4.3 Well-balanced Lax-Friedrichs FOM . . . . . 29

2.4.4 Numerical corrections . . . . . 30

2.5 2D shallow water equations . . . . . 34

**3 Reduced-order models for linear problems** **43**

3.1 Basic aspects . . . 43
3.1.1 Reduced-order modelling strategy . . . 43
3.1.2 Detailed example of the development of a ROM . . . 47
3.1.3 Extension to 2D problems . . . 51
3.1.4 Preliminary numerical results . . . 51
3.2 ROMs and high-order schemes . . . 63
3.2.1 Lax-Friedrichs-based ROM . . . 63
3.2.2 Lax-Wendroff-based ROM . . . 63
3.2.3 Test case 4. Analysis of the convergence . . . 64
3.3 ROMs and parametrized linear problems . . . 68
3.3.1 Modification of the standard ROM strategy . . . 69
3.3.2 Test case 5. Input parameters: advection velocity and diffusion coefficient 70
3.3.3 Test case 6. Input parameters: initial Gaussian profile . . . 74
3.3.4 Test case 7. Input parameters: initial discontinuity . . . 76
3.3.5 Test case 8. Input parameters: boundary Gaussian profile . . . 77
3.4 Beyond the training time (I) . . . 79
3.4.1 Coordinate transform-based ROM strategy . . . 80
3.4.2 1D advection-diffusion equation . . . 82
3.4.3 1D advection equation with time-varying velocity . . . 89
3.4.4 1D reactive transport of two coupled solutes . . . 92
3.4.5 2D extension of the CTROM strategy using Radon transform . . . 96
3.5 Concluding remarks . . . 99

**4 ROMs and non-linear problems 101**
4.1 Why and how to approximate non-linearity with linearized ROMs . . . 101
4.1.1 Development of a fully non-linear ROM . . . 102
4.1.2 PID method . . . 105
4.1.3 Development of a linearized ROM . . . 106
4.1.4 A posteriori error/efficiency criterion . . . 109
4.2 ROMs and the parametrized Burgers equation . . . 115
4.2.1 Combination of the standard ROM strategy & PID . . . 115
4.2.2 Test case 18. Predicting the levels of a discontinuous IC . . . 115

4.2.3 Test case 19. Predicting the position of a discontinuous IC . . . . . 118
4.2.4 Test case 20. Predicting the shape of a Gaussian profile as IC . . . . . 119
4.2.5 Test case 21. Predicting the position of a Gaussian profile . . . . . 122
4.3 Beyond the training time (II) . . . . . 123
4.3.1 Development of the CTROM of the Burgers equation . . . . . 123
4.3.2 Test case 22. 1D shock generation . . . . . 126
4.3.3 Test case 23. 1D rarefaction generation . . . . . 128
4.4 Concluding remarks . . . . . 130

**5 ROMs applied to the SWE 133**

5.1 Development of ROMs . . . . . 133
5.1.1 ARoe-based ROMs . . . . . 134
5.1.2 1D Lax-Friedrichs-based ROM . . . . . 137
5.1.3 1D well-balanced Lax-Friedrichs-based ROM . . . . . 138
5.2 Numerical corrections . . . . . 139
5.2.1 Well-balancing . . . . . 139
5.2.2 Source term correction to ensure positive water depth . . . . . 148
5.2.3 Friction correction to avoid reverse flow . . . . . 149
5.2.4 Entropy fix . . . . . 153
5.2.5 Wet/dry treatment . . . . . 157
5.3 ROMs and parametrized SWEs . . . . . 171
5.3.1 Test case 32. Input parameter: Manning coefficient . . . . . 171
5.3.2 Test case 33. Input parameter: shape of the bed slope . . . . . 175
5.3.3 Test case 34. Input parameter: defining position of the bed slope . . . . 178
5.3.4 Test case 35. Input parameters: initial discontinuity . . . . . 180
5.3.5 Test case 36. Input parameters: position of the initial discontinuity . . . 183
5.3.6 Test case 37. Input parameter: boundary condition . . . . . 185
5.4 Beyond the training time (III) . . . . . 188
5.4.1 1D linearized shallow water equations . . . . . 188
5.4.2 2D linearized shallow water equations . . . . . 190
5.5 Final numerical cases . . . . . 193
5.5.1 Test case 40. 1D/2D dam-break over a triangular obstacle . . . . . 193

5.5.2 Test case 41. 2D water depth source. . . . . 201
5.5.3 Test case 43. 2D tsunami test case. . . . . 203
5.6 Concluding remarks . . . . . 208

**6 Concluding remarks** **209**
6.1 Final conclusions . . . . . 209
6.2 Future work . . . . . 210
6.3 Main contributions . . . . . 211
6.3.1 General contributions . . . . . 211
6.3.2 Articles and other publications . . . . . 212
6.3.3 Talks and other communications . . . . . 213

**Conclusiones** **215**

# Abstract

The shallow water equations (SWEs) are a system of partial differential equations (PDEs) that model the spatio-temporal evolution of numerous problems that can be found in nature characterised by having a much smaller vertical dimension than the horizontal ones. Unfortunately, SWEs have no analytical solution, so they have to be solved numerically. Among the different numerical methods available, the Finite Volume (FV) method is widely used to solve this type of problem because its integral discretisation of the conservation laws governing SWEs faithfully reflects the physics behind these laws. Godunov-type FV method based on the Augmented Roe (ARoe) scheme have been used in this thesis due to its robustness. The numerical resolution of realistic scenarios of a transient and complex nature by means of validated, robust and competitive numerical schemes involves high computational costs. Because of this, in recent years there has been a large field of research devoted to the exploration and development of numerical tools to speed up computations.

In this direction, intrusive reduced-order models (ROMs) based on the proper orthogonal decomposition (POD) are presented as alternative numerical schemes to the classical ones that allow to speed up computational calculations without compromising the accuracy of the solutions. They require a training phase that consists of the application of the POD method to the training solutions (called snapshots) calculated by means of the classical numerical schemes (or full-order models (FOMs)) to define the reduced space of the variables of interest. The acceleration achieved by ROMs is based on the reduction of the dimensionality of this reduced space with respect to the discretised physical space. The ROM is developed from a given FOM by means of the Galerkin method, which is essentially based on projecting the high-fidelity solutions of the FOM onto the reduced space.

In this thesis, the application of intrusive POD-based ROMs to different problems is presented, such as the advection-diffusion equation, the Burgers equation and with the main focus on the SWEs. In the case of non-linear problems, linearisation of ROMs using the proper interval decomposition (PID) method has been proposed to speed-up computation times. The efficiency of ROMs has been tested with respect to FOMs by comparing their computational cost and the accuracy of their solutions. Furthermore, it is of great interest to study which properties the ROM inherits from the FOM and which does not, such as the convergence order, the well-balanced property and the numerical corrections required by the ARoe-based FOM for the resolution of SWEs. In addition to this, the possibilities for ROMs to overcome the limitations imposed by the training snapshots have been studied. On the one hand, a modification of the standard ROM methodology has been developed based on an appropriate coordinate transformation to obtain solutions beyond the training time. Although this transformation is developed for 1D problems, it is extended to 2D problems by using the Radon transform. On

the other hand, a modification of the classical ROM training strategy has been proposed to solve parameterized problems and the set of parameters appear in the coefficients of the PDE, in the initial condition or the boundary conditions.

# Resumen

Las ecuaciones de aguas poco profundas (SWE) son un sistema de ecuaciones diferenciales parciales (EDPs) que modelizan la evolución espacio-temporal de numerosos problemas que pueden encontrarse en la naturaleza caracterizados por tener una dimensión vertical mucho menor que la horizontal. Desafortunadamente, las EDPs no tienen solución analítica, por lo que deben resolverse numéricamente. Entre los diferentes métodos numéricos disponibles, el método de Volúmenes Finitos (VF) es ampliamente utilizado para resolver este tipo de problemas debido a que su discretización integral de las leyes de conservación que gobiernan las SWE refleja fielmente la física que subyace a estas leyes. En esta tesis se ha utilizado el método FV de tipo Godunov basado en el esquema de Roe aumentado (ARoe) debido a su robustez. La resolución numérica de escenarios realistas de naturaleza transitoria y compleja mediante esquemas numéricos validados, robustos y competitivos implica elevados costes computacionales. Debido a ello, en los últimos años se ha generado un amplio campo de investigación dedicado a la búsqueda y desarrollo de herramientas numéricas que permitan acelerar los cálculos.

En este sentido, los modelos de orden reducido (ROMs) intrusivos basados en la descomposición ortogonal adecuada (POD) se presentan como esquemas numéricos alternativos a los métodos clásicos que permiten acelerar los cálculos computacionales sin perder precisión en las soluciones. Estos requieren una fase de entrenamiento que consiste en la aplicación del método POD a las soluciones de entrenamiento (denominadas *snapshots*) calculadas mediante los esquemas numéricos clásicos (o modelos de orden completo (FOMs)) para definir el espacio reducido de las variables de interés. La aceleración lograda por los ROMs se basa en la reducción de la dimensionalidad de este espacio reducido con respecto al espacio físico discretizado. El ROM se desarrolla a partir de un FOM dado mediante el método de Galerkin, que se basa esencialmente en proyectar las soluciones de alta fidelidad del FOM sobre el espacio reducido.

En esta tesis se presenta la aplicación de ROMs intrusivos basados en el método POD a diferentes problemas, como la ecuación de advección-difusión, la ecuación de Burgers y con especial atención a las SWE. En el caso de los problemas no lineales, la linealización de los ROM mediante el método de descomposición en intervalos adecuados (PID) para acelerar los tiempos de cálculo. Se ha comprobado la eficacia de los ROMs con respecto a los FOMs comparando su coste computacional y la precisión de sus soluciones. Además, es de gran interés estudiar qué propiedades hereda el ROM del FOM y cuáles no, tales como el orden de convergencia, la propiedad *well-balanced* y las correcciones numéricas requeridas por el FOM basado en ARoe para la resolución de las SWE. Además, se han estudiado las posibilidades de que los ROMs superen las limitaciones impuestas por los *snapshots* de entrenamiento. Por un lado, se ha desarrollado una modificación de la metodología ROM estándar basada en una transformación de

coordenadas apropiada para obtener soluciones más allá del tiempo de entrenamiento. Aunque esta transformación está desarrollada para problemas 1D, se extiende a problemas 2D utilizando la transformada de Radon. Por otro lado, se ha propuesto una modificación de la estrategia clásica de entrenamiento ROM para resolver problemas parametrizados y el conjunto de parámetros aparecen en los coeficientes de la EDPs, en la condición inicial o en las condiciones de contorno.

# Acronyms

| | |
|---|---|
| **ARoe** | Augmented Roe |
| **BC** | Boundary condition |
| **CTFOM** | Coordinate transform full-order model |
| **CTROM** | Coordinate transform reduced-order model |
| **FOM** | Full-order model |
| **FV** | Finite Volume |
| **IC** | Initial condition |
| **LF** | Lax-Friedrichs |
| **LW** | Lax-Wendroff |
| **PDE** | Partial differential equation |
| **PID** | Proper interval decomposition |
| **POD** | Proper orthogonal decomposition |
| **RP** | Riemann problem |
| **ROM** | Reduced-order model |
| **SVD** | Singular value decomposition |
| **SWE** | Shallow water equation |
| **TC** | Test case |
| **WLF** | Well-balanced Lax-Friedrichs |

# Chapter 1

# Introduction

## 1.1 Motivation

Many phenomena in nature can be mathematically modelled by means of **partial differential equations (PDEs)**. These equations allow us to establish relationships between the involved variables and to approximate the underlying physical principles governing these phenomena. In many cases there are no analytical solutions for PDEs, so they have to be solved by means of **numerical methods**. Computational results offer the advantage of providing extensive and accurate information when coming from a well-verified numerical model [11].

Within the framework of Fluid Mechanics it is possible to identify **free surface flows**, which are of great interest for sciences and engineering (see for example [22, 30, 47, 170, 187, 190, 196]). A free-surface flow is mathematically defined as one in which the horizontal scale is much larger than the depth of the layer of fluid [95, 102]. The **shallow water equations** (SWE) modelling free surface flows arise as an approximation of the Navier-Stokes equations in which the dimensionality of the problem is reduced by averaging the variables in the vertical direction and assuming hydrostatic pressure. Their mathematical formulation is based on fundamental conservation laws of mass and momentum, which are expressed in the form of a system of hyperbolic partial differential equations [95, 11]. They govern the spatio-temporal evolution of the water depth, flow and velocity of the fluid.

There is no analytical solution for SWE, so it is necessary to solve them numerically. There are numerous techniques for discretising PDEs that can be divided in different main groups based on their discretising technique, such as the Finite Difference method, which is the oldest one and is attributed to Leonhard Euler in 1768, based on estimating the derivatives of the PDEs by the ratio of two differences according to their theoretical definition [23, 135]; the Finite Element method, which was initially proposed by Courant in 1943 [37] and in which the structure of an object is cut into several elements, and then the elements are reconnected at points called nodes. The Discontinuous-Galerkin method, originally attributed to [152] and extended in [35, 36], have been one of the most popular FE methods in recent years [28, 53, 80, 82, 132]; and the **Finite Volume (FV) method**, independently introduced by McDonald in 1971 [116] and MacCormack and Paullay in 1972 [113] and extended by Rizzi and Inouye in 1973 [157]. The FV method is based on the direct discretization of the integral form of the conservation laws and this form does not require the fluxes to be continuous. The FV method being closer

to the physical flow conservation laws is the reason why it is very useful when solving Fluid Mechanics equations and, in particular, the SWEs.

Godunov-type FV methods discretize the information of the hyperbolic conservation systems by assuming a piece-wise constant distribution of the conserved variables within computational cells [64, 78, 102]. This leads to the appearance of discontinuities at the walls between the cells. The updating of the variables in the cells is based on the calculation of numerical fluxes at the walls. In order to calculate these fluxes, so-called **Riemann problems (RP)** are formulated that consist of the hyperbolic equation together with special initial data defined by a piecewise constant function with a single jump discontinuity at some point [102]. In the solution of a non-linear Riemann problem, different types of waves may appear, such as shock and rarefaction waves. Depending on the initial conditions on both sides of a given point, compression zones can be produced that generate shock waves; or regions of expansion, with rarefaction waves. The physical phenomenon of dam-break is a widespread example where such waves can be found. The essential feature of an instantaneous dam-break is a large discontinuity that occurs when the water accumulated at a higher elevation is released and flows over the lower channel. Dam-breaks summarise the essential features of RPs, so that are commonly treated as benchmark test cases when solving SWEs. In this thesis they play a significant role, as will be seen in Chapter 5. RPs are solved by means of **Riemann solvers** and there are two types: exact and approximate [102, 184]. On the one hand, the exact Riemann solvers are based on iterative methods and are, therefore, very expensive. On the other hand, approximate Riemann solvers are the most used due to their greater efficiency. Within the approximate ones, there are the linearised solvers, such as the Roe solver used in this thesis, and the non-linear ones, such as the HLL and all its variants [48, 77, 125, 185].

When trying to capture the dynamics of realistic shallow flows, the inclusion of source terms in the equations that represent the effects of variable bed level and the friction is mandatory. Several works indicate that the use of the numerical flux as defined for homogeneous equations is not adequate to solve situations involving source terms [44, 59, 85, 101, 125, 127]. Among the many options that exist for constructing numerical schemes of the SWE, **the augmented Roe method** [59] has been chosen since it has been reported to produce robust and stable solutions in many situations. It uses averaged values of the variables at walls to compute the numerical fluxes. Apart from that, it represents the complete eigenstructure of the Jacobian matrix of the flux, i.e. all waves, unlike HLL [48, 77, 125, 185] , which is an incomplete solver and does not represent the contact wave, generating a larger numerical diffusion in the transverse component of the momentum. Furthermore, this first-order method has been chosen instead of other more accurate methods, such as methods based on WENO reconstructions [89, 107], for the sake of simplicity. The key feature of augmented Riemann solvers is that they are designed to preserve equilibrium in presence of source terms [132]. This is also called the **well-balancing property** and ensures discrete equilibrium with machine precision [18, 57, 70, 75, 142]. In addition to well-balancing, it is well known that other **numerical corrections** are necessary to fix some unphysical numerical solutions that may appear under certain circumstances, such as the entropy and the wet-dry front problems [76, 84, 102, 184, 199].

The **Fluid Dynamic Technologies (TFD)** group at the Aragon Institute for Engineering

Research (I3A) of the University of Zaragoza specialises works on the study and development of numerical schemes based on the Finite Volume method applied to problems governed by SWEs and much of the work of the TFD is focused on the resolution of realistic scenarios involving water flows on coasts, in large river basins or channels [43, 58, 104, 131, 188], in close collaboration with the Ebro River Basin Authority (Confederación Hidrográfica del Ebro). In all these works, the numerical models are applied to **realistic problems** of a complex and variable nature, so that the computational costs are very high. As a result, the predictive capacity of the mathematical tools can be lost. Several lines of research have been opened either to improve the classical numerical schemes, such as high-order schemes [130, 132, 175]; and to accelerate the computations, such as GPU programming [43, 45, 46]; or to study alternatives to these with objectives such as the calibration of parameters with the adjoint method [66, 67].

This thesis opens a new of approach line that aims to combine several improvements on the existing tools. In short, **intrusive reduced-order models (ROMs)** are intended to be alternative numerical schemes to replace the calculations performed by classical schemes, also called full-order models (FOMs), to save computational costs without losing accuracy in the solutions. These ROMs reside in a reduced dimensional space much smaller than the physical space, which is the reason why they are more efficient than FOMs, and they can be applied to several kinds of problems.

ROMs need to be developed from a certain FOM, so that each problem needs a different ROM; but this makes them able to preserve many of the good features of the developing FOM and allows the resolution of problems of great complexity such as those modelled by the SWEs.

ROMs are data-driven methods, which means that they need a training phase prior to their resolution. The training phase defines the framework in which ROMs can be solved, setting limitations on the values of the parameters that define the problem and on the final simulation time. The standard ROM strategy can be modified as proposed in this thesis, so that ROMs can be applied to parametrized problems and overcome the training limitations. And, by means of an appropriate coordinate transformation, they can even predict solutions in time, without compromising their computational cost savings.

## 1.2 State of the art

The large number and diversity of problems requiring computational cost improvement has led in recent years to the development of a wide range of mathematical strategies and tools in the scientific literature to facilitate, improve and increase the calculation capacity of the classical methods used in the framework of Fluid Mechanics. The main examples of these are the (discrete) empirical interpolation method [15, 33], dynamic mode decomposition [4, 97, 100, 109, 167, 179, 189], Krylov subspaces method [51, 197] and artificial neural networks [4, 19, 145, 155].

In addition to the above methods, reduced-order modelling is one of the most popular in the field. It was originally developed as the Reduced basis strategy for predicting the non-linear static response of structures [8, 129, 136]. This strategy essentially states that the

variable of interest resides on a low-dimensional manifold within the infinite-dimensional solution space associated with the partial differential equation [148]. Within the framework of reduced-order modelling, intrusive reduced order models based on the proper orthogonal decomposition (POD) are one of the most interesting tools for Fluid Mechanics [5]. The POD method was introduced originally by Lumley in 1967 [111] to approach the turbulence problem by random field of velocities of turbulent flows into a set of deterministic functions [191]. The POD method is also known as Karhunen-Loève expansions [92, 108], principal component analysis [81, 90, 121, 143].

ROMs have been developed for many equations in the literature, such as elliptic equations [148], parabolic equations [32, 71, 79, 146, 159] and hyperbolic equations including the linear scalar equation [112, 149, 161], as well as for the Burgers equation [3, 16, 87, 93, 96, 134, 154, 156, 165], the Navier-Stokes equations [20, 42, 60, 79, 162], the shallow water equations (SWE) [4, 73, 179, 200, 202] and other nonlinear problems involving discontinuous solutions [182].

ROMs have been combined with other strategies. An interesting analysis of the POD-based ROM methodology in comparison with other non-intrusive methods applied to 2D SWE is presented in [4, 140]. Non-intrusive POD-based ROMs have also been applyed to the Navier-Stokes equations [29, 193, 194] and to the SWEs [91]. It is also worth mentioning interesting recent works investigating the combination of neural networks with POD methods applied to Navier-Stokes equations [118, 144] or even deep-learning techniques [1, 39, 55] and Physics-informed neural networks [198]. Other works on the combination of SWE and ROMs are focused on reducing the computational complexity by using discrete empirical interpolation method with implicit numerical schemes [178], parametric sensitivity analysis [200, 201], the meshless radial basis functions technique [41] or different recent works on ROMs based on high order dynamic mode decomposition combined [94, 99, 98, 189].

The intrusive POD-based ROM strategy consists of two parts: the off-line phase, in which the ROM is trained; and the on-line phase, in which the ROM is numerically solved. The training of the ROM starts by computing a set of high-fidelity reference solutions with a classic numerical scheme or FOM, following the snapshot method [171]. The solutions calculated by the FOM are called snapshots and they are used to train the ROM by means of the singular value decomposition (SVD) [171]. Once trained, the ROM can be solved up to the same training time. The main advantage of ROMs is that they are supposed to be faster than FOMs and can achieve speed-ups of several orders of magnitude [3].

Since intrusive ROMs are in fact numerical schemes, they need to be developed as such from classical schemes in a process of dimension reduction. This is done by means of the Galerkin method [56], according to which an orthogonal projection is made from the physical space of the FOM to the reduced space defined by the POD method. In other words, ROMs arise as transformations of already known numerical schemes. This means that the FOMs are of intrinsic importance in the development of the ROMs and their good construction can in turn determine the success or failure of the ROM that is developed from them.

Solving problems in realistic scenarios competitively and faithfully according to physics requires a detailed knowledge of the tools used. This means that the numerical schemes must be well

tuned and calibrated, as if they were the finest of orchestras. This exhaustive knowledge can be maintained and luckily extended to ROMs because of their intrusive condition, whereas with other black-box type methodologies it would be completely lost. This answers the question of why choosing POD-based intrusive ROMs out of all the possible tools that exist in the field of Fluid Mechanics.

In the field of free surface flows, this exhaustive knowledge translates into the construction of numerical schemes that solve SWEs in a way that preserves equilibrium solutions, respects dry/wet fronts, does not cause negative water depths, etc. One of the main objectives of this thesis is to study whether ROMs need to take into account all these numerical corrections that improve the FOMs or whether simply being trained with corrected solutions is enough. This study can be framed within the concept of ROM consistency as proposed by Ingimarson, Rebholz and Iliescu in a very recent work [88], in which they investigate theoretically and numerically how the performance of the ROM is affected by the discretisation model used with respect to the FOM. Another very recent example dealing with the study of the concept of ROM consistency is that of García-Archilla, Novo and Rubino [61], in which the Navier-Stokes equations are considered.

The resolution of nonlinear features such as the generation of shock waves and rarefaction is not trivial when using ROMs. Thus, the SWEs are a particularly interesting challenge when applying ROMs. As will be seen below, it is possible to develop ROMs from nonlinear FOMs that are also nonlinear. However, the increase in complexity also means an increase in the computational cost [179, 202]. It is therefore interesting to consider the linearization of ROMs to speed up their resolution of non-linear models. There are many alternatives in the literature, such as tensorial POD [179], goal-oriented POD [21], the proper generalized decomposition [50], the trust-region proper orthogonal decomposition [17, 49], the shifted POD [153], the balanced POD [160, 192], the spectral POD [2, 117, 186], stabilization techniques [31, 87, 63, 137, 161, 162, 163, 166, 168, 195], the weighted POD [34] or the local reduced-order bases [10]. Among all of them, the **proper interval decomposition (PID)** [86, 164, 202] has proven to be very useful when solving the SWEs. According to this method, different time windows are defined within the snapshot matrix, which implies the definition of multiple reduced spaces. Thanks to this, nonlinear problems such as the SWEs can be successfully and efficiently approximated by linearized ROMs, as proposed in this thesis.

Dirichlet-type boundary conditions (BCs) require special treatment when integrated into ROM solving. There are different ways to deal with it in the literature, such as the *lifting functions* used to homogenize the snapshots so that they become independent of the BCs [62], or the explicit-boundary POD method proposed in [68, 72], according to which part of the ROM is replaced by equations for the original BCs. In this thesis, it is proposed to solve problems with Dirichlet-type BCs by means of ROMs using a technique whereby the BCs are directly integrated into the on-line phase of ROM solving. To the best of the author's knowledge, it has not been presented previously.

Training solutions impose computational limits that ROMs cannot exceed. These limits affect multiple aspects of the resolution and can be divided into two large groups: the final simulation

time and the parameters that define the problem. However, it is highly interesting to study what can be done to overcome these limits by means of ROMs. In this thesis the possibilities to obtain solutions to parametrized problems and beyond the training time will be explored.

Time prediction is not always possible and represents one of the major limitations of the POD method (and, in general, SVD-based methods, such as DMD) when dealing with advection-dominated equations [3, 97, 110], this being a challenging problem of recent interest. Computing predicted solutions with a ROM for times longer than the training time would suppose a major step in the field of computational hydraulics. Different works have been carried out in the field of temporal prediction, such as the application to the diffusion equation [109], the application to the dominant convection equation based on a Lagrangian formulation of the ROM [110] and the application to hyperbolic equations based on auto-encoder type algorithms [120]. In the present thesis, a ROM strategy based on the coordinate transformation proposed in [69] and called CTROM is proposed with the aim of predicting solutions beyond the training time in the Eulerian framework. Following this method, the computational grid evolves in time following a main characteristic curve. Other works propose different mapping strategies, such as the Lagragian framework approach in [110, 119, 120].

The correct resolution of PDEs by means of numerical schemes requires a thorough calibration of the parameters on which the mathematical model depends. This calibration involves high computational costs in cases where the aim is to emulate realistic problems by means of classical schemes. It is therefore of great interest to try to find alternatives that help in this calibration objective in parametrized problems [150]. In the framework of SWEs, elements of the source terms such as the Manning coefficient modelling the bed friction or the shape of the bed itself are usually considered as those input parameters that require calibration. In this sense, ROMs can be useful to launch multiple simulations in a less expensive way to map different values of the input parameter. There are different strategies available in the literature, such as those based on the reduced-basis methods [54, 74, 138, 139, 182, 180, 181, 183], the interpolation in Matrix manifolds [9, 141, 158] or the extrapolation technique carried out in [202]. The extension of intrusive POD-based ROMs to the parametrized versions of the equations solved in this thesis is proposed, which is based on a modification of the training of the ROM, as presented in [14, 13, 62, 177]. In addition to the aforementioned input parameters, the advection velocity and the diffusion coefficient have also been considered in the diffusion-advection equation, as well as different parameters that define the initial (IC) and boundary conditions of this equation, the Burgers equation and the SWEs.

## 1.3 Objectives of this thesis

The improvement of the efficiency of the calculations is one of the most fundamental objectives. ROMs have to be faster than FOMs while keeping accuracy in their solutions to be useful. The efficiency is measured as the trade-off between the acceleration achieved by the ROM in computational CPU time and the error introduced with reference to the FOM solution. Efficiency is used as the final criterion to assess the performance of ROMs in each of the

numerical results of this thesis.

The **main objective** of this thesis is the development and application of instrusive POD-based ROMs to hyperbolic problems, with special emphasis on the SWEs, to speed-up computational times without compromising the accuracy of the solutions. This can be divided into the following partial objectives

- Fully non-linear ROMs are as slow or slower than their developing FOMs. In this way, an objective of this thesis is the development of competitive ROMs applied to non-linear problems by means of their linearization using the PID method.
- A necessary step in the application of ROMs to SWEs is the study of their consistency with respect to the numerical corrections required by the FOM (wet/dry treatment, entropy fix, etc.). Thus, an objective is to uniquely evaluate each numerical correction needed when solving the SWEs in relation to the ROM.
- FOMs need to be well-balanced to preserve equilibrium states when solving the SWEs. Another objective regarding ROM consistency is study whether ROMs need to be well-balanced or they just need well-balanced training snapshots.
- Higher-order schemes provide more accurate solutions than first-order schemes. It is proposed here to develop and solve ROMs from higher-order schemes in order to test whether their combination maintains the benefits of both.
- Given the need for training that ROMs require, it is great importance to study as an objective of this thesis the possibility of calculating solutions beyond the training time. To overcome the time limitation imposed by the training snapshots, the standard ROM strategy has to be modified to include an appropriate coordinate transformation that allows time prediction.
- Numerical schemes require the calibration of certain parameters to accurately represent reality. The computational costs of this calibration are high and ROMs can help. To this end, the aim is to solve parametrized problems by means of ROMs to obtain solutions with parameter values that do not belong to the training set.
- The final aim of this thesis is to show the real capacity of ROMs. For this purpose, they are applied to a series of test cases that represent challenging scenarios of difficult resolution. These scenarios involve complex geometries, several numerical corrections and very fine unstructured meshes. As a result of all of this, problems of this kind imply high computational costs when using FOMs; and the high efficiency of intrusive POD-based ROMs has been proven.

## 1.4 Outline

The thesis is structured according to the following chapter distribution:

In **Chapter 2**, a summary of all the mathematical models used in this thesis is presented. The equations that help to better understand the POD-based ROMs are included, in increasing order of complexity, starting with the 1D and 2D advection-diffusion equation, through the 1D nonlinear Burgers equation and ending with the 1D and 2D SWEs. This same order is used to define the following chapters, in which the ROM methodology is applied to each of these equations. In addition to that, the numerical schemes for solving these equations numerically are also presented. As explained before, FOMs play a principal role in the development of ROMs. This is why they have been given a chapter of their own, and are not included in an appendix, where they would lose the relevance they actually have. However, the truly new content of this thesis is contained in Chapters 3, 4 and 5.

In **Chapter 3**, the most fundamental theoretical aspects of POD-based ROMs are first presented. It begins with a detailed explanation of its strategy and the phases into which it is divided. This is followed by an example of a step-by-step development of the 1D linear equation-based ROM and its extension to 2D problems. The first section closes with preliminary numerical results that serve to illustrate these basic aspects. The extension of POD-based ROMs to higher-order numerical schemes and how this affected the numerical results is studied, which will be presented with a corresponding test case. The following sections present the first parts of the studies on solving ROMs beyond the limits imposed by training, first with parameters and then with training time. In this second study, a modification of the ROM strategy based on an appropriate coordinate transformation is presented. A series of test cases illustrate the good results obtained in both studies.

In **Chapter 4**, POD-based ROMs are extended to non-linear problems through their application to the 1D Burgers equation. Theoretical aspects of this extension are first discussed, such as the need to modify the POD methodology by making use of time windows, following PID to face the generation of shock and rarefaction waves. Also, the linearization of ROMs based on non-linear equations is proposed for efficiency reasons, which will be properly justified by means of numerical results. Apart from that, an a posteriori criterion is introduced to select the number of POD modes and time windows to use by the ROM, since the PID disables the a priori criterion presented in Chapter 3. To conclude this chapter, the second part of the sections on solving ROMs beyond parameters and training time is presented. In both cases, the particular modifications required to cope with the non-linearity are presented.

In **Chapter 5**, the intrusive POD-based ROM strategy is applied to the SWEs. The 1D and 2D versions of the ARoe-based ROMs are presented. In order to study whether ROMs need to be well-balanced to obtain well-balanced solutions, a couple of test cases are presented that investigate this, also making use of the Lax-Friedrichs scheme and its well-balanced version to perform a consistency study related to this property. Following this consistency direction, it has been tested whether other numerical corrections required by the ARoe-based FOM should be included in the ROM or whether it is sufficient to train it with corrected snapshots. The extension of the technique that allows to obtain solutions beyond the training time to the linearised version of the SWEs is presented. In addition, the parametric limits imposed by snapshots are also explored by applying ROMs to parametrized SWEs. Finally, a series of numerical results are presented in which the real capacity of ROMs is evaluated. In these

test cases, large 2D problems have been solved using very fine meshes, with time-varying boundary conditions and complex topographic geometries. All the conclusions obtained from the consistency study carried out in previous sections of the same chapter are applied.

**Chapter 6** outlines all the conclusions reached after the completion of this thesis and sets out potential future work. It also shows all the works derived from this thesis, including articles, talks and posters.

## 1.5 Acknowledgements

This work was funded by the Spanish Ministry of Science and Innovation under the research projects PGC2018-094341-B-I00 and PID2022-137334NB-I00. This work has also been partially funded by Gobierno de Aragón through Fondo Social Europeo (T32-20R and E24-17R, Feder 2019-2021 "Construyendo Europa desde Aragón").

# Chapter 2

# Equations and full-order models

In this chapter, the equations in the continuous domain considered in this thesis are introduced, as well as the classical numerical schemes used to solve them in the discrete domain. All of them represent the initial boundary value problems on which the intrusive POD-based ROMs will be applied and tested in the later chapters. The following will be considered, with special emphasis on the resolution of hyperbolic problems,

- 1D and 2D linear advection-diffusion equations, used to understand the most fundamental aspects of ROMs.
- 1D Burgers equation, as a preliminary step to the SWEs.
- 1D and 2D SWE, as the main objective of the thesis is the application of ROMs to these equations to reproduce realistic problems.

## 2.1 1D advection-diffusion equation

Consider the following partial differential equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \tag{2.1}$$

where $u = u(x,t)$ is the conserved variable and $f(u) = au$ is the physical flux, with $a$ the advection velocity; and $\nu \geq 0$ is the diffusion coefficient. To find $u(x,t)$ in the domain $(x,t) \in [0,L] \times [0,T]$, equation (2.1) has to be solved subject to appropiate initial conditions (ICs) and boundary conditions (BCs).

The analysis will be focused on transport problems (i.e. $\nu = 0$) and advection-dominated advection-diffusion problems (i.e. $\nu \ll |a(x,t)|$, for all $x,t$). The solution of the transport problem is constant along the characteristic curves. For example, if $a$ is constant, $u(x,t)$ remains constant along any time-space line that satisfies $x - at =$constant [102] and can be approximated along them from the solution given by the ICs and the BCs. The advection velocity $a$ determines the slope of the characteristic and all the characteristic curves are parallel to each other, as shown in Figure 2.1.

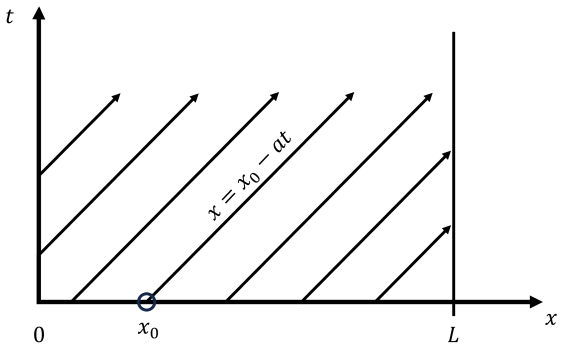



Figure 2.1: Characteristic curves of the 1D advection equation.

On the one hand, in all the examples of this thesis, the IC will be defined by a Gaussian function or a piecewise polynomial function. And, on the other hand, the BC at $x = 0$ (and at the end of the domain $x = L$) will be one of the following possibilities

- Dirichlet BC

$$u(0,t) = u_0(t),\ t > 0, \tag{2.2}$$

  where $u_0$ is a given function.

- Periodic BC

$$u(0,t) = u(L,t),\ t > 0. \tag{2.3}$$

- Neumann BC

$$\left.\frac{\partial u}{\partial x}\right|_{(0,t)} = 0,\ t > 0. \tag{2.4}$$

Equation (2.1) can be numerically approximated by means of the FV method [83, 103, 142]. The computational domain is discretized using volume cells of uniform length $\Delta x$ where the positions of the center and left and right interfaces of $i$-th cell are $x_i$, $x_{i-1/2}$ and $x_{i+1/2}$, respectively, with $i = 1, ..., I_x$, being $I_x$ the number of volume cells.

Regarding the time discretization, the time step $\Delta t = t^{n+1} - t^n$ with $n = 0, ..., N_T - 1$, and $N_T$ is the total number of time steps, is selected dynamically using the Courant-Friedrichs-Lewy (CFL) condition [38]

$$\Delta t = CFL \frac{\Delta x^2}{\Delta x \max(\lambda) + 2\nu}, \tag{2.5}$$

where $\lambda = \dfrac{\partial f}{\partial u} = a$ and $0 < \text{CFL} \leq 1$.

By means of the FV method, the variables are integrated into each of the volume cells, shown in Figure 2.2. In this way, the discretised variables are averaged over the entire spatial volume occupied by the cell.

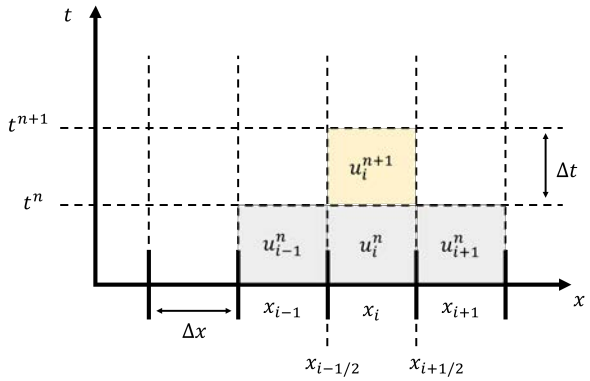



Figure 2.2: Discretization of physical space into a numerical mesh.

Following the FV method, the 1D advection-diffusion equation is integrated inside the $i$-th volume cell $\left(x_{i-1/2}, x_{i+1/2}\right)$ with $2 \leq i \leq I_x - 1$ in the time interval $(t^n, t^{n+1})$ as follows

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial u}{\partial t} dx dt + \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial f(u)}{\partial x} dx dt = \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \nu \frac{\partial^2 u}{\partial x^2} dx dt,$$

and this yields

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{f_{i+1/2}^{n,-,*} - f_{i-1/2}^{n,+,*}}{\Delta x} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}, \tag{2.6}$$

where $u_i^n \approx u(x_i, t^n)$ is the cell average value over the cell $(x_{i-1/2}, x_{i+1/2})$ at $t^n$. The diffusion term is discretized using central differences and, regarding the advection term, the numerical fluxes,

$$f_{i+1/2}^{n,-,*} = \int_{t^n}^{t^{n+1}} f(u(x_{i+1/2}, t)) dt, \quad f_{i-1/2}^{n,+,*} = \int_{t^n}^{t^{n+1}} f(u(x_{i-1/2}, t)) dt,$$

are reconstructed by means of the first-order upwind (FOU) method. In some cases, other methods are considered, such as the Lax-Friedrichs scheme, that has some advantages when using ROMs although it is less accurate, and the Lax-Wendroff methods, that is second-order of accuracy. All of them are explained below.

### 2.1.1 First-order upwind scheme

Following the first order upwind (FOU) method [59, 124, 132], the numerical fluxes are discretized as

$$f_{i+1/2}^{n,+,*} = f(u_{i+1}^n) - \left(\tilde{\lambda}^+\right)_{i+1/2}^n \delta u_{i+1/2}^n,$$
$$f_{i+1/2}^{n,-,*} = f(u_i^n) + \left(\tilde{\lambda}^-\right)_{i+1/2}^n \delta u_{i+1/2}^n,$$

where $f(u_i^n) = a u_i^n$, $\delta u_{i+1/2}^n = u_{i+1}^n - u_i^n$ and $\left(\tilde{\lambda}^\pm\right)_{i+1/2}^n = \frac{1}{2}\left(a \pm |a|\right)$.

By introducing these numerical fluxes in (2.6), the FOU-based FOM is obtained

$$\begin{aligned} u_i^{n+1} = u_i^n - \frac{1}{2}a\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\ + \frac{\Delta t}{\Delta x^2}\nu\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right),\ i = 2, ..., I_x - 1. \end{aligned} \tag{2.7}$$

As can be seen in Equation (2.7), this method (and all those used in this thesis) is explicit, i.e. the update of the variable in the next time step $t^{n+1}$ is done with information exclusively from the immediately preceding step $t^n$. Likewise, the $i$-th cell uses information from itself and its neighbouring cells $i-1$ and $i+1$ for this update, as can be seen in the Figure 2.2, where the grey volume cells are used to calculate the yellow cell in the next time step.

All the variables of interest belonging to the volume cells inside the domain are integrated in time following the same updating relation indicated in (2.7). However, those on the boundary require a special treatment that takes into account the boundary conditions imposed on them.

Boundary conditions are very relevant when developing ROMs, as will be seen in the following chapters. It is therefore necessary to establish from the beginning a clear division of the boundary conditions into two distinct groups in the discrete domain that address the way in which they will later be included in the development of ROMs.

- On the one hand, the periodic BC (2.3) is discretized as follows

$$\begin{aligned} u_1^{n+1} = u_1^n - \frac{1}{2}a\frac{\Delta t}{\Delta x}\left(u_2^n - u_{I_x}^n\right) + \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\left(u_2^n - 2u_1^n + u_{I_x}^n\right) \\ + \frac{\Delta t}{\Delta x^2}\nu\left(u_2^n - 2u_1^n + u_{I_x}^n\right). \end{aligned} \tag{2.8}$$

  In addition, the following discretization of the homogeneous Neumann BC (2.4) at $x = 0$ is considered

$$u_1^{n+1} = u_1^n - \frac{1}{2}a\frac{\Delta t}{\Delta x}\left(u_2^n - u_1^n\right) + \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\left(u_2^n - u_1^n\right) + \frac{\Delta t}{\Delta x^2}\nu\left(u_2^n - u_1^n\right); \tag{2.9}$$

- On the other hand, the time-dependent Dirichlet-type BC given in (2.2) at $x = 0$ is discretized as follows

$$u_1^n = \left(u_0\right)^n, \tag{2.10}$$

  where $\left(u_0\right)^n = u_0(t^n)$ is a given function.

The discretization of all these boundary conditions at $x = L$ are analogous.

### 2.1.2 Optimized Lax-Friedrichs scheme

The optimized version of the first-order Lax-Friedrichs scheme proposed in [26, 115] is characterized by the following numerical fluxes

$$f_{i+1/2}^{n,-,*} = f_{i+1/2}^{n,+,*} = \frac{1}{2}\left[f\left(u_i^n\right) + f\left(u_{i+1}^n\right)\right] - \frac{1}{2}\xi\frac{\Delta x}{\Delta t}\left(u_{i+1}^n - u_i^n\right),$$

with $0 < \xi \leq CFL$. So that, the standard Lax-Friedrichs-based FOM (LF-based FOM) reads

$$\begin{aligned} u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\frac{1}{2}a\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{1}{2}\xi\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\ + \frac{\Delta t}{\Delta x^2}\nu\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right),\ i = 2, ..., I_x - 1. \end{aligned} \tag{2.11}$$

### 2.1.3 Lax-Wendroff scheme

The Lax-Wendroff solver consists of a second-order numerical scheme following [78, 115, 147]

$$f_{i+1/2}^{n,-,*} = f_{i+1/2}^{n,+,*} = f(u_i^n) + a\frac{1}{2}\left(u_{i+1}^n - u_i^n\right) - \frac{1}{2}a^2\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right).$$

Thus, the Lax-Wendroff-based FOM (LW-based FOM) reads

$$\begin{aligned} u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\frac{1}{2}a\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{1}{2}\left(\frac{\Delta t}{\Delta x}a\right)^2\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\ + \nu\frac{\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right),\ i = 2, ..., I_x - 1. \end{aligned} \tag{2.12}$$

The boundary conditions indicated in each test case for the Lax-Friedrichs and the Lax-Wendroff schemes determine the component equations of the boundary volume cells, $i = 1, I_x$, in a similar way as indicated in (2.3), (2.4) and (2.2) for the FOU-based FOM (2.7).

## 2.2 2D advection-diffusion equation

Consider the 2D version of the advection-diffusion equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = \nu_x\frac{\partial^2 u}{\partial x^2} + \nu_y\frac{\partial^2 u}{\partial y^2}, \tag{2.13}$$

where $u = u(x, y, t)$ is the conserved variable; $f(u) = a_x u$ and $g(u) = a_y u$ are the physical fluxes, with $a_x$ and $a_y$ the advection velocities in the $x$- and $y$-directions, respectively; and $\nu_x$ and $\nu_y$ are the diffusion coefficients in the $x$- and $y$-directions, respectively. The problem $u(x, t)$ is defined on the domain $(x, y, t) \in [0, L_x] \times [0, L_y] \times [0, T]$ and initial and boundary conditions are imposed. In the case that 2D problems are considered in this thesis, the boundary of the domain $\Gamma$ is split into

$$\Gamma = \Gamma^D \cup \Gamma^S,\ \Gamma^D \cap \Gamma^S = \emptyset,$$

where Dirichlet BCs are specified on $\Gamma^D$ and Neumann or periodic BCs are specified on $\Gamma^S$.

The FOM is formulated by means of the FV method and considering a rectangular structured mesh $\{(x_i, y_j),\, x_i = i\Delta x,\, y_j = j\Delta y\}_{i,j=1}^{I_x,I_y}$ it is given by

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + \frac{f_{i+1/2,j}^{n,-,*} - f_{i-1/2,j}^{n,+,*}}{\Delta x} + \frac{g_{i,j+1/2}^{n,-,*} - g_{i,j-1/2}^{n,+,*}}{\Delta y} = \nu_x \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \nu_y \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2},$$

where $u_{i,j}^{n} \approx u(x_i, y_j, t^n)$ is the cell average value over the cell $(x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2})$ at $t^n$. Based on the first-order upwind method, the numerical fluxes can be formulated as follows

$$f_{i+1/2,j}^{n,\mp,*} = f(u_{i,j}^{n}) \pm \left(\tilde{\lambda}_x^{\mp}\right)_{i+1/2,j}^{n} \delta u_{i+1/2,j}^{n},$$
$$g_{i,j+1/2}^{n,\mp,*} = f(u_{i,j}^{n}) \pm \left(\tilde{\lambda}_y^{\mp}\right)_{i,j+1/2}^{n} \delta u_{i,j+1/2}^{n},$$

where $f(u_{i,j}^{n}) = a_x u_{i,j}^{n}$; $g(u_{i,j}^{n}) = a_y u_{i,j}^{n}$; $\delta u_{i+1/2,j}^{n} = u_{i+1,j}^{n} - u_{i,j}^{n}$; $\delta u_{i,j+1/2}^{n} = u_{i,j+1}^{n} - u_{i,j}^{n}$; and $\left(\tilde{\lambda}_x^{\pm}\right)_{i+1/2,j}^{n} = \frac{1}{2}\left(a_x \pm |a_x|\right)$ and $\left(\tilde{\lambda}_y^{\pm}\right)_{i,j+1/2}^{n} = \frac{1}{2}\left(a_y \pm |a_y|\right)$.

The final expression of the FOU-based FOM of the 2D advection-diffusion equation is

$$\begin{aligned} u_{i,j}^{n+1} = u_{i,j}^{n} &- \frac{1}{2} a_x \frac{\Delta t}{\Delta x}\left(u_{i+1,j}^{n} - u_{i-1,j}^{n}\right) + \frac{1}{2}\left|a_x\right| \frac{\Delta t}{\Delta x}\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right) \\ &- \frac{1}{2} a_y \frac{\Delta t}{\Delta y}\left(u_{i,j+1}^{n} - u_{i,j-1}^{n}\right) + \frac{1}{2}\left|a_y\right| \frac{\Delta t}{\Delta y}\left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right) \\ &+ \nu_x \frac{\Delta t}{\Delta x^2}\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right) + \nu_y \frac{\Delta t}{\Delta y^2}\left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right), \end{aligned} \tag{2.14}$$

for $1 \le i \le I_x$ and $1 \le j \le I_y$.

Two examples of free boundary conditions are given below for a point placed at the corner $(0, 0)$ and for a point placed at the boundary $y = 0$ of the rectangular domain, respectively

$$\begin{aligned} u_{1,1}^{n+1} = u_{1,1}^{n} &- \frac{1}{2} a_x \frac{\Delta t}{\Delta x}\left(u_{2,1}^{n} - u_{1,1}^{n}\right) + \frac{1}{2}\left|a_x\right| \frac{\Delta t}{\Delta x}\left(u_{2,1}^{n} - u_{1,1}^{n}\right) \\ &- \frac{1}{2} a_y \frac{\Delta t}{\Delta y}\left(u_{1,2}^{n} - u_{1,1}^{n}\right) + \frac{1}{2}\left|a_y\right| \frac{\Delta t}{\Delta y}\left(u_{1,2}^{n} - u_{1,1}^{n}\right) \\ &+ \nu_x \frac{\Delta t}{\Delta x^2}\left(u_{2,1}^{n} - u_{1,1}^{n}\right) + \nu_y \frac{\Delta t}{\Delta y^2}\left(u_{1,2}^{n} - u_{1,1}^{n}\right), \\ u_{i,1}^{n+1} = u_{i,1}^{n} &- \frac{1}{2} a_x \frac{\Delta t}{\Delta x}\left(u_{i+1,1}^{n} - u_{i-1,1}^{n}\right) + \frac{1}{2}\left|a_x\right| \frac{\Delta t}{\Delta x}\left(u_{i+1,1}^{n} - 2u_{i,1}^{n} + u_{i-1,1}^{n}\right) \\ &- \frac{1}{2} a_y \frac{\Delta t}{\Delta y}\left(u_{i,2}^{n} - u_{i,1}^{n}\right) + \frac{1}{2}\left|a_y\right| \frac{\Delta t}{\Delta y}\left(u_{i,2}^{n} - u_{i,1}^{n}\right) \\ &+ \nu_x \frac{\Delta t}{\Delta x^2}\left(u_{i+1,1}^{n} - 2u_{i,1}^{n} + u_{i-1,1}^{n}\right) + \nu_y \frac{\Delta t}{\Delta y^2}\left(u_{i,2}^{n} - u_{i,1}^{n}\right), \end{aligned}$$

with $2 \le i \le I_x - 1$.

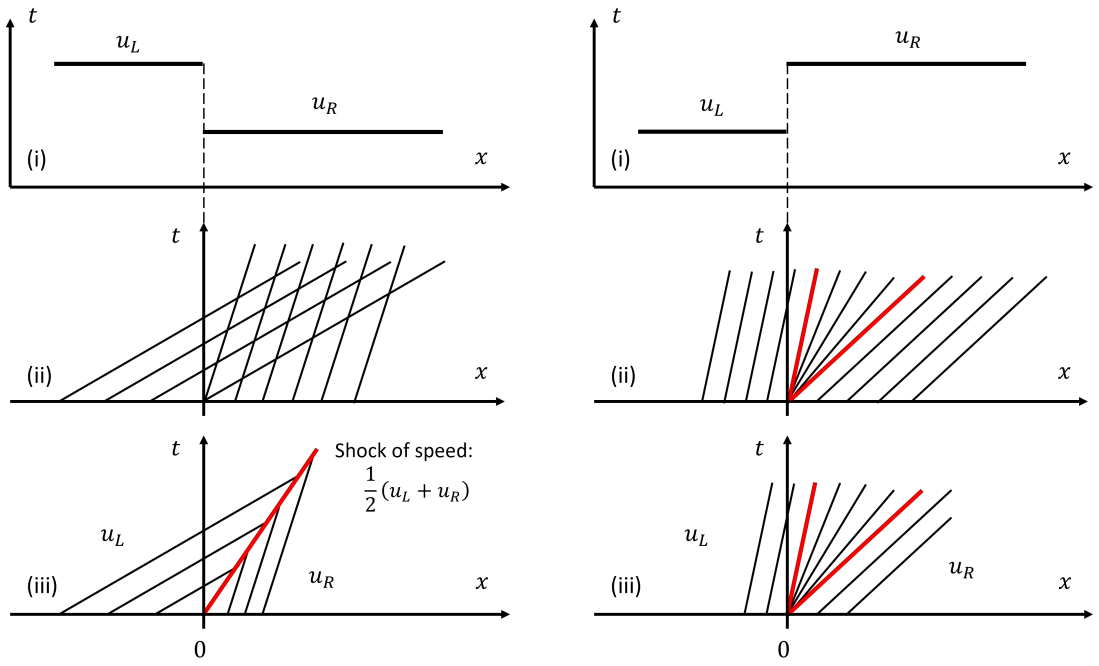
## 2.3 1D Burgers' equation

The Burgers' equation was introduced by Burgers in 1948 [25] as a one-dimensional compressible analogue of the Navier-Stokes equations, since it emulates their convective dynamic terms in a simplified way and provides valuable information on some of the properties of turbulence.

The 1D inviscid Burgers equation with geometrical source term reads

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = -u\frac{\partial z}{\partial x}, \tag{2.15}$$

where $u = u(x,t)$ is the conserved variable and $f(u) = u^2/2$ is the physical flux, so that the conserved variable acts as the advection velocity itself $u = \partial f/\partial u$; and $z = z(x)$ represents the geometrical source term. Boundary and initial conditions are also imposed and the solution $u(x,t)$ is defined on the domain $(x,t) \in [0,L] \times [0,T]$.




(a) Shock wave.

(b) Rarefaction wave.

Figure 2.3: (i) Discontinuous data. (ii) Characteristic curves. (iii) Waves on $x-t$ plane. See [184].

Analogously to the linear equation (2.1), $u$ is constant along the characteristic curves traveling at speed $u$ [102]. Because of non-linearity, solutions of the Burgers equation (2.15) which are initially smooth may become discontinuous within a finite time, which is called loss of regularity [132]. This is related to the generation of compressive or shock waves and rarefaction or fan waves. For example, consider a piecewise constant initial condition with values $u_L$ and $u_R$ as shown in Figure 2.3. Shock waves appear when a discontinuity is formed with $u_L > u_R$, so that the characteristic curves are compressed at one point, as also indicated in Figure 2.3a. Regarding the rarefaction waves, since $u_L < u_R$, a fan of diverging characteristic curves is

formed, as shown in Figure 2.3b. These are two of the main features of non-linearity and they will play an important role in the study carried out in this thesis on the application of ROMs to this type of problem. For more information about the dynamics of the Burgers equation, see [102, 184].

For a numerical scheme, the preservation of equilibrium solutions is a feature of great importance. For this reason, source terms must be properly discretized ensuring the preservation of equilibrium solutions (i.e. the well-balanced property) [132]. The equilibrium condition for (2.15) is obtained when $\partial u/\partial t$ vanishes, yielding

$$\frac{\partial}{\partial x}\left(u+z\right)=0, \tag{2.16}$$

which allows definition of an equilibrium variable as $e=u+z$. Steady solutions of (2.15) will satisfy that $e$ is constant.

The numerical solution of the Riemann problem involved in this type of problem generates a contact wave due to the geometric source term $z$. As explained in [132], augmented Riemann solvers were introduced in an attempt to represent the effect of the source term in the solution of the Riemann problem and automatically fulfill the Rankine-Hugoniot conditions [127].

The FOM of the 1D Burgers equation (2.15) is formulated by means of the FV method and the same relationship is obtained as in the linear case (2.6). Following the first-order upwind method, the numerical fluxes in (2.6) are discretized as indicated below [59, 124, 132], where it is worth noting that the source term is included in the numerical fluxes

$$\begin{aligned}
f^{n,+,*}_{i+1/2} &= f(u^n_{i+1}) - \left(\tilde{\lambda}^{+}\right)^n_{i+1/2}\left(\delta u^n_{i+1/2} + \delta z_{i+1/2}\right),\\
f^{n,-,*}_{i+1/2} &= f(u^n_{i}) + \left(\tilde{\lambda}^{-}\right)^n_{i+1/2}\left(\delta u^n_{i+1/2} + \delta z_{i+1/2}\right),
\end{aligned}$$

where $f(u^n_i)=\left(u^n_i\right)^2/2$, $\delta u^n_{i+1/2}=u^n_{i+1}-u^n_i$, $\delta z_{i+1/2}=z_{i+1}-z_i$ with $z_i\approx z(x_i)$, and $\left(\tilde{\lambda}^{\pm}\right)^n_{i+1/2}=\frac{1}{2}\left(u^n_i+u^n_{i+1}\pm|u^n_i+u^n_{i+1}|\right)$. This FOU-method preserves equilibrium, i.e., it is well-balanced. The final formulation of the FOU-based FOM of the Burgers' equation for the interior points of the domain is

$$\begin{aligned}
u^{n+1}_i = u^n_i &- \frac{\Delta t}{\Delta x}\left[\frac{1}{4}\left(u^n_i+u^n_{i-1}+|u^n_i+u^n_{i-1}|\right)\left(u^n_i-u^n_{i-1}\right)\right.\\
&\left.+\frac{1}{4}\left(u^n_{i+1}+u^n_i-|u^n_{i+1}+u^n_i|\right)\left(u^n_{i+1}-u^n_i\right)\right]\\
&- \frac{\Delta t}{\Delta x}\left[\frac{1}{4}\left(u^n_i+u^n_{i-1}+|u^n_i+u^n_{i-1}|\right)\left(z_i-z_{i-1}\right)\right.\\
&\left.+\frac{1}{4}\left(u^n_{i+1}+u^n_i-|u^n_{i+1}+u^n_i|\right)\left(z_{i+1}-z_i\right)\right],
\end{aligned} \tag{2.17}$$

with $2\leq i\leq I_x-1$. The boundary volume cells are computed according to the conditions imposed in each case.

## 2.4 1D shallow water equations

The 1D SWE with source terms assuming a rectangular channel of unit width

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} &= 0, \\ \frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q^2}{h} + \frac{1}{2}gh^2\right) &= gh(S_z - S_f), \end{aligned} \tag{2.18}$$

where $h = h(x,t)$ is the depth and $q = q(x,t)$ is the unit discharge, with $q = hu$ and $u = u(x,t)$, the depth averaged velocity with $(x,t) \in [0,L] \times [0,T]$. The first source term of (2.18) is due to the gradient of the bed elevation $z = z(x)$

$$S_z = -\frac{\partial z}{\partial x}; \tag{2.19}$$

as can be seen in Figure 2.4.

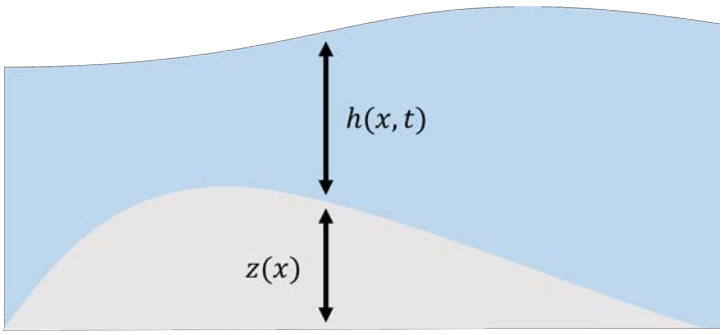



Figure 2.4: Flow model in shallow water equations.

The second source term is due to bed friction

$$S_f = \frac{n_b^2 u\,|u|}{h^{4/3}},$$

where $n_b$ is the Manning coefficient. In addition, initial and boundary conditions for $h$ and $q$ have to be defined. The hyperbolic system of partial differential equations (2.20) can be written in a conservative form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S}, \tag{2.20}$$

where

$$\mathbf{U} = \begin{pmatrix} h \\ q \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} q \\ \frac{q^2}{h} + \frac{1}{2}gh^2 \end{pmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ gh(S_z - S_f) \end{pmatrix},$$

$\mathbf{U}$ is the vector of conserved variables, $\mathbf{F}(\mathbf{U})$ are the physical fluxes and $\mathbf{S}(\mathbf{U})$ are the source terms. System (2.20) can be written in quasi-linear form

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}\frac{\partial \mathbf{U}}{\partial x} = \mathbf{S},$$

and the Jacobian matrix $\mathbf{J}(\mathbf{U})$ of the convective fluxes is given by

$$\mathbf{J}(\mathbf{U}) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix},$$

with $c = \sqrt{gh}$ the surface wave velocity. Assuming that this system of equations is strictly hyperbolic, the Jacobian matrix $\mathbf{J}(\mathbf{U})$ is diagonizable with real eigenvalues. Its eigenvalues $\lambda_1$ and $\lambda_2$ and the corresponding eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$ are

$$\begin{cases} \lambda_1 = u - c, \\ \lambda_2 = u + c, \end{cases} \quad \mathbf{e}_m = (1, \lambda_m)^T, \quad m = 1, 2.$$

The equilibrium state of the 1D SWE (2.18) is also known as lake at rest condition, and, similarly to the Burgers equation, satisfies that

$$\frac{\partial}{\partial x}(h + z) = 0. \tag{2.21}$$

This must be preserved at the discrete level when solving the 1D SWE by means of numerical schemes, so they have to be well-balanced [130].

The FOM is obtained by discretizing (2.18) with the FV method

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+1/2}^{n,-,*} - \mathbf{F}_{i-1/2}^{n,+,*}\right), \tag{2.22}$$

where $\mathbf{U}_i^n = (h_i^n, q_i^n)^T$ and considering explicit schemes, the time step $\Delta t = t^{n+1} - t^n$ is selected dynamically using the Courant-Friedrichs-Lewy (CFL) condition [38]

$$\Delta t = CFL\frac{\Delta x}{\max\{\lambda_1, \lambda_2\}}, \tag{2.23}$$

where the CFL number satisfies $0 < CFL \leq 1$. Then, $h_i^n \approx h(x_i, t^n)$, $u_i^n \approx u(x_i, t^n)$ and $q_i^n \approx q(x_i, t^n)$ are the cell-averaged values of the water depth, velocity and discharge over the cell $(x_{i-1/2}, x_{i+1/2})$. The numerical fluxes $\mathbf{F}_{i+1/2}^{n,-,*}$ and $\mathbf{F}_{i-1/2}^{n,+,*}$ are constructed following two different spatial discretisations: one based on the Roe numerical scheme and the other two based on the Lax-Friedrichs numerical scheme, the optimized version and a new well-balanced version instroduced. In these three discretizations, the friction and bed slope source terms have been defined upwind, so they are included in the numerical flows. They are explained in detail below.

### 2.4.1 Augmented Roe-based FOM

The Augmented Roe-based FOM is obtained by computing the numerical fluxes $\mathbf{F}_{i+1/2}^{n,-,*}$ and $\mathbf{F}_{i-1/2}^{n,+,*}$ as in [59, 127, 132]

$$\begin{aligned}\mathbf{F}^{n,+,*}_{i+1/2} &= \mathbf{F}(\mathbf{U}^n_i) - \sum_{m=1}^{2} \left(\tilde{\lambda}^+_m \tilde{\gamma}^+_m \tilde{\mathbf{e}}_m\right)^n_{i+1/2},\\ \mathbf{F}^{n,-,*}_{i-1/2} &= \mathbf{F}(\mathbf{U}^n_i) + \sum_{m=1}^{2} \left(\tilde{\lambda}^-_m \tilde{\gamma}^-_m \tilde{\mathbf{e}}_m\right)^n_{i-1/2},\end{aligned} \tag{2.24}$$

where

$$\left(\tilde{\lambda}^{\mp}_m\right)^n_{i+1/2} = \frac{1}{2}\left[\left(\tilde{\lambda}_m\right)^n_{i+1/2} \mp \left|\tilde{\lambda}_m\right|^n_{i+1/2}\right],\ m = 1, 2.$$

The approximate eigenvalues $(\tilde{\lambda}_m)^n_{i+1/2}$ and eigenvectors $(\tilde{\mathbf{e}}_m)^n_{i+1/2}$ are

$$\begin{cases} \left(\tilde{\lambda}_1\right)^n_{i+1/2} = \tilde{u}^n_{i+1/2} - \tilde{c}^n_{i+1/2}, \\ \left(\tilde{\lambda}_2\right)^n_{i+1/2} = \tilde{u}^n_{i+1/2} + \tilde{c}^n_{i+1/2}, \end{cases} \qquad (\tilde{\mathbf{e}}_m)^n_{i+1/2} = \left(1, (\tilde{\lambda}_m)^n_{i+1/2}\right)^T, \quad m = 1, 2,$$

with the following expressions for the Roe average velocity $\tilde{u}^n_{i+1/2}$ and celerity $\tilde{c}^n_{i+1/2}$

$$\begin{aligned}\tilde{u}^n_{i+1/2} &= \frac{q^n_{i+1}\sqrt{h^n_{i+1}} + q^n_i\sqrt{h^n_i}}{\sqrt{h^n_{i+1}h^n_i}\left(\sqrt{h^n_{i+1}} + \sqrt{h^n_i}\right)},\\ \tilde{c}^n_{i+1/2} &= \sqrt{g\tilde{h}^n_{i+1/2}},\end{aligned}$$

and $\tilde{h}^n_{i+1/2} = (h^n_{i+1} + h^n_i)/2$. The coefficients $(\tilde{\gamma}^{\pm}_m)^n_{i+1/2}$ in (2.24) are designed to preserve the well-balanced property

$$(\tilde{\gamma}^{\pm}_m)^n_{i+1/2} = \left(\tilde{\alpha}_m - \frac{\tilde{\beta}^{\pm}_m}{\tilde{\lambda}^{\pm}_m}\right)^n_{i+1/2}, \quad m = 1, 2,$$

where the wave strengths are

$$\begin{aligned}(\tilde{\alpha}_1)^n_{i+1/2} &= \frac{\delta h^n_{i+1/2}\left(\tilde{\lambda}_2\right)^n_{i+1/2} - \delta q^n_{i+1/2}}{\left(\tilde{\lambda}_2\right)^n_{i+1/2} - \left(\tilde{\lambda}_1\right)^n_{i+1/2}},\\ (\tilde{\alpha}_2)^n_{i+1/2} &= \frac{\delta q^n_{i+1/2} - \delta h^n_{i+1/2}\left(\tilde{\lambda}_1\right)^n_{i+1/2}}{\left(\tilde{\lambda}_2\right)^n_{i+1/2} - \left(\tilde{\lambda}_1\right)^n_{i+1/2}},\end{aligned}$$

and the source strengths

$$\left(\tilde{\beta}^{\pm}_m\right)^n_{i+1/2} = \left(\tilde{\beta}_m \frac{\tilde{\lambda}^{\pm}_m}{\tilde{\lambda}_m}\right)^n_{i+1/2},$$

with

$$(\tilde{\beta}_1)^n_{i+1/2} = (\tilde{\beta}_f)^n_{i+1/2} + (\tilde{\beta}_z)^n_{i+1/2} = -(\tilde{\beta}_2)^n_{i+1/2},$$

and

$$(\tilde{\beta}_f)^n_{i+1/2} = \Delta x \frac{1}{2} \frac{g}{\tilde{c}^n_{i+1/2}} \tilde{h}^n_{i+1/2} \frac{n_b^2 \tilde{u}^n_{i+1/2} |\tilde{u}^n_{i+1/2}|}{\max\left(h^n_{i+1}, h^n_i\right)^{4/3}},$$
$$(\tilde{\beta}_z)^n_{i+1/2} = \frac{1}{2} \frac{g}{\tilde{c}^n_{i+1/2}} \tilde{h}^n_{i+1/2} \delta z_{i+1/2},$$

and $\delta z_{i+1/2} = z_{i+1} - z_i$ and $\tilde{u}^n_{i+1/2} = (u^n_{i+1} + u^n_i)/2$.

The ARoe-based FOM of (2.18) is

$$\begin{aligned}
h_i^{n+1} = h_i^n &- \frac{\Delta t}{\Delta x} \left[ \left( \delta h \frac{\tilde{\lambda}_1^- \tilde{\lambda}_2 - \tilde{\lambda}_2^- \tilde{\lambda}_1}{2\tilde{c}} \right)^n_{i+1/2} + \left( \delta h \frac{\tilde{\lambda}_1^+ \tilde{\lambda}_2 - \tilde{\lambda}_2^+ \tilde{\lambda}_1}{2\tilde{c}} \right)^n_{i-1/2} \right] \\
&+ \frac{\Delta t}{\Delta x} \left[ \left( \delta q \frac{\tilde{\lambda}_1^- - \tilde{\lambda}_2^-}{2\tilde{c}} \right)^n_{i+1/2} + \left( \delta q \frac{\tilde{\lambda}_1^+ - \tilde{\lambda}_2^+}{2\tilde{c}} \right)^n_{i-1/2} \right] \\
&+ \Delta t \frac{g n_b^2}{2} \left[ \frac{\left( \tilde{h}\tilde{u}|\tilde{u}| \right)^n_{i+1/2} \left( \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2} \right)^n_{i+1/2}}{(\tilde{c})^n_{i+1/2} \max\left(h^n_{i+1}, h^n_i\right)^{4/3}} + \frac{\left( \tilde{h}\tilde{u}|\tilde{u}| \right)^n_{i-1/2} \left( \frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^+}{\tilde{\lambda}_2} \right)^n_{i-1/2}}{(\tilde{c})^n_{i-1/2} \max\left(h^n_i, h^n_{i-1}\right)^{4/3}} \right] \\
&+ \frac{\Delta t}{\Delta x} \frac{g}{2} \left[ \left( \frac{\tilde{h}}{\tilde{c}} \right)^n_{i+1/2} (\delta z)_{i+1/2} \left( \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2} \right)^n_{i+1/2} + \left( \frac{\tilde{h}}{\tilde{c}} \right)^n_{i-1/2} (\delta z)_{i-1/2} \left( \frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^+}{\tilde{\lambda}_2} \right)^n_{i-1/2} \right], \\
q_i^{n+1} = q_i^n &- \frac{\Delta t}{\Delta x} \left[ \left( \delta h \frac{\tilde{\lambda}_1^- - \tilde{\lambda}_2^-}{\tilde{\lambda}_2 - \tilde{\lambda}_1} \tilde{\lambda}_1 \tilde{\lambda}_2 \right)^n_{i+1/2} + \left( \delta h \frac{\tilde{\lambda}_1^+ - \tilde{\lambda}_2^+}{\tilde{\lambda}_2 - \tilde{\lambda}_1} \tilde{\lambda}_1 \tilde{\lambda}_2 \right)^n_{i-1/2} \right] \\
&+ \frac{\Delta t}{\Delta x} \left[ \left( \delta q \frac{\tilde{\lambda}_1^- \tilde{\lambda}_1 - \tilde{\lambda}_2^- \tilde{\lambda}_2}{\tilde{\lambda}_2 - \tilde{\lambda}_1} \right)^n_{i+1/2} + \left( \delta q \frac{\tilde{\lambda}_1^+ \tilde{\lambda}_1 - \tilde{\lambda}_2^+ \tilde{\lambda}_2}{\tilde{\lambda}_2 - \tilde{\lambda}_1} \right)^n_{i-1/2} \right] \\
&+ \Delta t \frac{g n_b^2}{2} \left[ \frac{\left( \tilde{h}\tilde{u}|\tilde{u}| \right)^n_{i+1/2} \left( \tilde{\lambda}_1^- - \tilde{\lambda}_2^- \right)^n_{i+1/2}}{(\tilde{c})^n_{i+1/2} \max\left(h^n_{i+1}, h^n_i\right)^{4/3}} + \frac{\left( \tilde{h}\tilde{u}|\tilde{u}| \right)^n_{i-1/2} \left( \tilde{\lambda}_1^+ - \tilde{\lambda}_2^+ \right)^n_{i-1/2}}{(\tilde{c})^n_{i-1/2} \max\left(h^n_i, h^n_{i-1}\right)^{4/3}} \right] \\
&+ \frac{\Delta t}{\Delta x} \frac{g}{2} \left[ \left( \frac{\tilde{h}}{\tilde{c}} \right)^n_{i+1/2} (\delta z)_{i+1/2} \left( \tilde{\lambda}_1^- - \tilde{\lambda}_2^- \right)^n_{i+1/2} + \left( \frac{\tilde{h}}{\tilde{c}} \right)^n_{i-1/2} (\delta z)_{i-1/2} \left( \tilde{\lambda}_1^+ - \tilde{\lambda}_2^+ \right)^n_{i-1/2} \right].
\end{aligned} \tag{2.25}$$

for $2 \leq i \leq I_x - 1$.

### 2.4.2 Optimized Lax-Friedrichs-based FOM

The 1D SWEs are here discretized following an optimized version of the Lax-Friedrichs method proposed in [26] is obtained by defining the numerical fluxes in (2.22) as

$$\mathbf{F}^{n,\mp,*}_{i\pm1/2} = \mathbf{F}^{n}_{i\pm1/2} \mp \frac{1}{2}\mathbf{S}^{n}_{i\pm1/2} - \frac{1}{2}\xi\frac{\Delta x}{\Delta t}\delta\mathbf{U}^{n}_{i\pm1/2},$$

with CFL $\leq \xi \leq 1$; and

$$\mathbf{U}^{n}_{i} = \begin{pmatrix} e^{n}_{i} \\ q^{n}_{i} \end{pmatrix}, \quad \mathbf{F}^{n}_{i} = \begin{pmatrix} q^{n}_{i} \\ \dfrac{(q^{n}_{i})^2}{h^{n}_{i}} + \dfrac{1}{2}g(h^{n}_{i})^2 \end{pmatrix};$$

where the height of the water surface is $e^{n}_{i} = h^{n}_{i} + z_i$; and the source terms are

$$\mathbf{S}^{n}_{z,i+1/2} = \begin{pmatrix} 0 \\ -g\tilde{h}^{n}_{i+1/2}\delta z_{i+1/2} \end{pmatrix}, \quad \mathbf{S}^{n}_{f,i+1/2} = \begin{pmatrix} 0 \\ -g\Delta x\tilde{h}^{n}_{i+1/2}\dfrac{n_b^2\tilde{u}^{n}_{i+1/2}\left|\tilde{u}^{n}_{i+1/2}\right|}{\max\left(h^{n}_{i+1}, h^{n}_{i}\right)^{4/3}} \end{pmatrix},$$

with $\tilde{h}^{n}_{i+1/2} = (h^{n}_{i+1} + h^{n}_{i})/2$, $\tilde{u}^{n}_{i+1/2} = (u^{n}_{i+1} + u^{n}_{i})/2$ and $\delta z_{i+1/2} = z_{i+1} - z_i$.

The optimized Lax-Friedrichs-based FOM is

$$\begin{aligned}
h^{n+1}_{i} &= h^{n}_{i} - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(q^{n}_{i+1} - q^{n}_{i-1}\right) + \frac{1}{2}\xi\left(h^{n}_{i+1} - 2h^{n}_{i} + h^{n}_{i-1}\right) + \frac{1}{2}\xi\left(z_{i+1} - 2z_i + z_{i-1}\right), \qquad (2.26)\\
q^{n+1}_{i} &= q^{n}_{i} - \frac{1}{2}\frac{\Delta t}{\Delta x}\left[q^{n}_{i+1}u^{n}_{i+1} + \frac{1}{2}g\left(h^{n}_{i+1}\right)^2 - q^{n}_{i-1}u^{n}_{i-1} - \frac{1}{2}g\left(h^{n}_{i-1}\right)^2\right]\\
&\quad + \frac{1}{2}\xi\left[q^{n}_{i+1} - 2q^{n}_{i} + q^{n}_{i-1}\right] - \frac{1}{2}g\frac{\Delta t}{\Delta x}\left[\tilde{h}^{n}_{i+1}\delta z_{i+1/2} + \tilde{h}^{n}_{i-1}\delta z_{i-1/2}\right]\\
&\quad - \frac{1}{2}g\Delta t\left[\tilde{h}^{n}_{i+1}\frac{n_b^2\tilde{u}^{n}_{i+1/2}|\tilde{u}^{n}_{i+1/2}|}{\max\left(h^{n}_{i+1}, h^{n}_{i}\right)^{4/3}} + \tilde{h}^{n}_{i-1}\frac{n_b^2\tilde{u}^{n}_{i-1/2}|\tilde{u}^{n}_{i-1/2}|}{\max\left(h^{n}_{i}, h^{n}_{i-1}\right)^{4/3}}\right].
\end{aligned}$$

### 2.4.3 Well-balanced Lax-Friedrichs FOM

The augmented Roe method used in this work is defined as an augmented Riemann solver. On the contrary, the optimized Lax-Friedrichs method proposed in [26] is not well-balanced, since

$$\begin{aligned}
\mathbf{F}^{n,-,*}_{i+1/2} - \mathbf{F}^{n,+,*}_{i+1/2} &= \mathbf{F}^{n}_{i+1/2} - \frac{1}{2}\mathbf{S}^{n}_{i+1/2} - \frac{1}{2}\xi\frac{\Delta x}{\Delta t}\delta\mathbf{U}^{n}_{i+1/2} - \mathbf{F}^{n}_{i+1/2} - \frac{1}{2}\mathbf{S}^{n}_{i+1/2} + \frac{1}{2}\xi\frac{\Delta x}{\Delta t}\delta\mathbf{U}^{n}_{i+1/2}\\
&= -\mathbf{S}^{n}_{i+1/2}\\
&\neq 0;
\end{aligned}$$

Therefore, a correction term must be added to reproduce properly equilibrium problems. The steady state solution satisfies $\partial q/\partial t = 0$, so the momentum equation (2.18) yields

$$\frac{\partial h}{\partial x} = -\frac{\partial z}{\partial x} - \frac{1}{c^2}\frac{\partial(hu^2)}{\partial x},$$

where we used that $q = hu$ and $c^2 = gh$. This equality motivates the well-balanced Lax-Friedrichs-based FOM (WLF), which is obtained by introducing into (2.22) the following numerical fluxes

$$\mathbf{F}^{n,\mp,*}_{i\pm1/2} = \mathbf{F}^{n}_{i\pm1/2} \mp \frac{1}{2}\mathbf{S}^{n}_{i\pm1/2} - \frac{1}{2}\xi\frac{\Delta x}{\Delta t}\left(\delta\mathbf{U}^{n}_{i\pm1/2} - \mathbf{L}^{n}_{i+1/2}\right), \tag{2.27}$$

with the following correction term

$$\mathbf{L}^{n}_{i+1/2} = \begin{pmatrix} -\delta z^{n}_{i+1/2} - \dfrac{\delta\left(hu^2\right)^{n}_{i+1/2}}{\left(c^{n}_{i+1/2}\right)^2} \\ 0 \end{pmatrix}.$$

### 2.4.4 Numerical corrections

The resolution of realistic scenarios by means of numerical methods requires a continuous effort to develop tools to deal with them competitively. It is also essential that they are able to solve these problems according to the laws of physics in order to obtain quality solutions, respecting their complex and variable nature.

For this reason, in many of these high-performance schemes, such as the augmented one mentioned above, it is also necessary to take into account a series of numerical corrections that are applied to the scheme itself to ensure its robustness. This avoids the appearance of results of unphysical nature.

The following four numerical corrections:

- source term correction to ensure positive water depths,
- friction correction to avoid reverse flow,
- wet/dry treatment,
- and entropy fix

are based on a thorough understanding of Riemann problems and thus arise as results of the superposition of travelling waves whose velocities are the eigenvalues of the problem [44, 76, 105, 122, 125, 127, 128].

These corrections are explained for 1D problems, but their extension to 2D problems is immediate, except in the case of dry-wetting, where the geometry must be taken into account when working with unstructured meshes.

#### 2.4.4.1 Source term correction to ensure positive water depths

The source terms defined according to the ARoe-based FOM (2.25) may cause negative water depths to appear, which does not make physical sense. It is therefore necessary to ensure that

this does not happen. For this purpose, the following intermediate states are defined for the water depth [125]

$$\begin{aligned}(h^*)^n_{i+1/2} &= h^n_i + (\tilde{\alpha}_1)^n_{i+1/2} - \left(\frac{\tilde{\beta}_1}{\tilde{\lambda}_1}\right)^n_{i+1/2} \geq 0,\\ (h^{**})^n_{i+1/2} &= h^n_{i+1} - (\tilde{\alpha}_2)^n_{i+1/2} + \left(\frac{\tilde{\beta}_2}{\tilde{\lambda}_2}\right)^n_{i+1/2} \geq 0.\end{aligned} \tag{2.28}$$

If any of these quantities (2.28) is negative, then the source terms would need to be modified as follows

$$\begin{aligned}\left(\tilde{\beta}_1\right)^n_{i+1/2} = -\left(\tilde{\beta}_2\right)^n_{i+1/2} &= \begin{cases}\left[h^n_i + (\tilde{\alpha}_1)^n_{i+1/2}\right]\left(\tilde{\lambda}_1\right)^n_{i+1/2}, & \text{if } (h^*)^n_{i+1/2} < 0,\\ \left(\tilde{\beta}_1\right)^n_{i+1/2}, & \text{otherwise.}\end{cases}\\ \left(\tilde{\beta}_2\right)^n_{i+1/2} = -\left(\tilde{\beta}_1\right)^n_{i+1/2} &= \begin{cases}-\left[h^n_{i+1} - (\tilde{\alpha}_2)^n_{i+1/2}\right]\left(\tilde{\lambda}_2\right)^n_{i+1/2}, & \text{if } (h^{**})^n_{i+1/2} < 0,\\ \left(\tilde{\beta}_2\right)^n_{i+1/2}, & \text{otherwise.}\end{cases}\end{aligned} \tag{2.29}$$

For more details, see [44, 125].

#### 2.4.4.2 Friction correction to avoid reverse flow

The friction force may reverse the sign of the flow velocity, so it has to be restricted. With this purpose, define the water discharge intermediate

$$\begin{aligned}(q^*)^n_{i+1/2} &= q^n_i + \left(\tilde{\alpha}_1\tilde{\lambda}_1\right)^n_{i+1/2} - \left(\tilde{\beta}_z\right)^n_{i+1/2} - \left(\tilde{\beta}_f\right)^n_{i+1/2},\\ (q^{**})^n_{i+1/2} &= q^n_i + \left(\tilde{\alpha}_1\tilde{\lambda}_1\right)^n_{i+1/2} - \left(\tilde{\beta}_z\right)^n_{i+1/2},\end{aligned} \tag{2.30}$$

where $\left(\tilde{\beta}^z_1\right)^n_{i+1/2}$ and $\left(\tilde{\beta}^f_1\right)^n_{i+1/2}$ are the contributions of the bed slope and the friction, respectively.

If $(q^{**})^n_{i+1/2} < 0$ and $(q^*)^n_{i+1/2} > 0$, then $(q^*)^n_{i+1/2}\,(q^{**})^n_{i+1/2} < 0$, which means that the numerical friction might reverse the flow. This case is not physically possible, so the friction contribution of the source term is redefined as

$$\left(\tilde{\beta}_f\right)^n_{i+1/2} = -\left(\tilde{\beta}_f\right)^n_{i+1/2} = \begin{cases}(q^{**})^n_{i+1/2}, & \text{if } (q^*)^n_{i+1/2}\,(q^{**})^n_{i+1/2} \leq 0,\\ \left(\tilde{\beta}_f\right)^n_{i+1/2}, & \text{otherwise.}\end{cases} \tag{2.31}$$

For more details, see [44, 125].

#### 2.4.4.3 Wet/dry treatment

Realistic scenarios may involve cases in which the water elevation in one cell may be smaller than the bed elevation of a neighbouring cell, i.e. $h_i + z_i < h_{i+1}$, such as the example shown in Figure 2.5. This needs a special treatment to avoid unphysical solutions that may lead to the wetting of areas that must not be wet. This treatment consists of two distinct steps embedded in the resolution of the FOM.

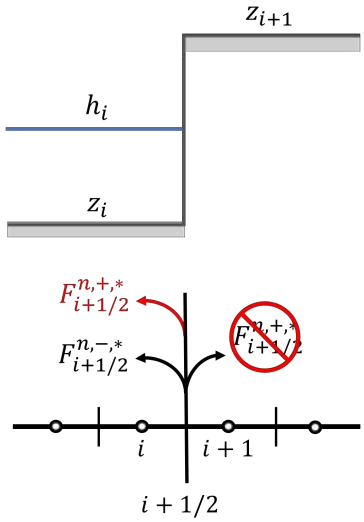



Figure 2.5: Wet/dry treatment example.

First, it is necessary to bounce the information that updates the variables back to the wet cell and not to the dry cell. The positive numerical flux $\mathbf{F}^{n,+,*}_{i+1/2}$ can not be sent to the right $(i+1)$-th cell, since it should not be wet. The wet/dry treatment proposed in [44, 125] sends the positive numerical flux backwards to the $i$-th cell, so the water gradually accumulates, causing the water depth to grow up to $z_{i+1}$. Taking into account the water depth intermediate states (2.28), the general wet/dry treatment is written as follows

$$\begin{cases} \mathbf{F}^{n,-,*}_{i+1/2} = \mathbf{F}^{n,-,*}_{i+1/2} + \mathbf{F}^{n,+,*}_{i+1/2}, \quad \mathbf{F}^{n,+,*}_{i+1/2} = 0, & \text{if } h^n_{i+1} = 0 \text{ and } h^{**}_i < 0, \\ \mathbf{F}^{n,+,*}_{i+1/2} = \mathbf{F}^{n,+,*}_{i+1/2} + \mathbf{F}^{n,-,*}_{i+1/2}, \quad \mathbf{F}^{n,-,*}_{i+1/2} = 0, & \text{if } h^n_i = 0 \quad \text{and } h^*_i < 0. \end{cases} \tag{2.32}$$

Secondly, it is important to impose a zero value on the velocities normal to wall $i+1/2$ in cases where the flux is not crossing the cell wall. In the 1D case, there is only a component perpendicular to the walls, so the full velocity $u^n_\perp$ is cancelled

$$u^n_\perp = u^n_i = u^n_{i+1} = 0, \tag{2.33}$$

and the water discharge $q^n_\perp$ is also cancelled

$$q^n_\perp = q^n_i = q^n_{i+1} = 0$$

In 2D cases approximated on unstructured meshes, as will be seen in Section 2.5, this second step of the wet/dry treatment requires further development, as there is a parallel component of the velocity.

#### 2.4.4.4 Entropy fix

Augmented Riemann solvers, such as the ARoe solver used here, may lead to physically meaningless solutions due to the entropy problem [126, 184]. The regime of the flow can be sub- or supercritical depending on the relative sign of the eigenvalues, $\lambda_1$ and $\lambda_2$, such that, if $\lambda_1\lambda_2 < 0$, it is supercritical and if $\lambda_1\lambda_2 > 0$. It can also be possible that one of the eigenvalues might be zero for some values of $(h, q)$. In transcritical rarefactions the continuous fan of intermediate states is represented using a Roe eigenvalue with approximately zero velocity [114] and this would imply numerical problems to properly update the variables.

The Froude number is a dimensionless number that determines the regime of the flow by balancing the inertial movement and the gravity driven movement

$$Fr = \sqrt{\frac{u}{gh}}; \tag{2.34}$$

so that

$$\begin{cases} Fr < 1, & \text{subcritical } (\lambda_1\lambda_2 < 0)\,, \\ Fr = 1, & \text{critical } (\lambda_1\lambda_2 = 0)\,, \\ Fr > 1, & \text{supercritical } (\lambda_1\lambda_2 > 0)\,. \end{cases}$$

To solve this problem, a numerical correction, known as entropy fix, have to be included in the numerical scheme [114, 124]. The entropy fix is implemented under the following conditions:

- If $\left(\tilde{\lambda}_1\right)_i^n < 0 < \left(\tilde{\lambda}_1\right)_{i+1}^n$, then the eigevalues and the source terms have to be modified as

$$\begin{aligned} \left(\tilde{\lambda}_1^-\right)_{i+1/2}^n &= \left(\tilde{\lambda}_1^{\leftarrow}\right)_{i+1/2}^n, \quad \left(\tilde{\lambda}_1^+\right)_{i+1/2}^n = \left(\tilde{\lambda}_1^{\rightarrow}\right)_{i+1/2}^n, \\ \left(\tilde{\beta}_1^{\leftarrow}\right)_{i+1/2}^n &= \left(\tilde{\beta}_1^-\right)_{i+1/2}^n, \quad \left(\tilde{\beta}_1^{\rightarrow}\right)_{i+1/2}^n = 0. \end{aligned} \tag{2.35}$$

- If $\left(\tilde{\lambda}_2\right)_i^n < 0 < \left(\tilde{\lambda}_2\right)_{i+1}^n$, then the eigenvalues and the source terms have to be modified as

$$\begin{aligned} \left(\tilde{\lambda}_2^-\right)_{i+1/2}^n &= \left(\tilde{\lambda}_2^{\leftarrow}\right)_{i+1/2}^n, \quad \left(\tilde{\lambda}_2^+\right)_{i+1/2}^n = \left(\tilde{\lambda}_2^{\rightarrow}\right)_{i+1/2}^n, \\ \left(\tilde{\beta}_2^{\leftarrow}\right)_{i+1/2}^n &= 0, \qquad\qquad \left(\tilde{\beta}_2^{\rightarrow}\right)_{i+1/2}^n = \left(\tilde{\beta}_2^+\right)_{i+1/2}^n. \end{aligned} \tag{2.36}$$

The modified eigenvalues are defined as

$$\begin{aligned}
\left(\tilde{\lambda}_1^{\leftarrow}\right)_{i+1/2}^n &\approx (\lambda_1)_i^n \frac{(\lambda_1)_{i+1}^n - \left(\tilde{\lambda}_1\right)_{i+1/2}^n}{(\lambda_1)_{i+1}^n - (\lambda_1)_i^n}, \quad \left(\tilde{\lambda}_1^{\rightarrow}\right)_{i+1/2}^n \approx (\lambda_1)_{i+1}^n \frac{\left(\tilde{\lambda}_1\right)_{i+1/2}^n - (\lambda_1)_i^n}{(\lambda_1)_{i+1}^n - (\lambda_1)_i^n}, \\
\left(\tilde{\lambda}_2^{\leftarrow}\right)_{i+1/2}^n &\approx (\lambda_2)_i^n \frac{(\lambda_2)_{i+1}^n - \left(\tilde{\lambda}_2\right)_{i+1/2}^n}{(\lambda_2)_{i+1}^n - (\lambda_2)_i^n}, \quad \left(\tilde{\lambda}_2^{\rightarrow}\right)_{i+1/2}^n \approx (\lambda_2)_{i+1}^n \frac{\left(\tilde{\lambda}_2\right)_{i+1/2}^n - (\lambda_2)_i^n}{(\lambda_2)_{i+1}^n - (\lambda_2)_i^n}.
\end{aligned} \tag{2.37}$$

## 2.5 2D shallow water equations

The 2D version of the SWE with source terms assuming a rectangular channel reads

$$\begin{aligned}
\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} &= 0, \\
\frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h} + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right) &= gh\left(S_{z_x} - S_{f_x}\right), \\
\frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_y^2}{h} + \frac{1}{2}gh^2\right) &= gh\left(S_{z_y} - S_{f_y}\right),
\end{aligned} \tag{2.38}$$

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

$$S_{z_x} = -\frac{\partial z}{\partial x}, \quad S_{z_y} = -\frac{\partial z}{\partial y},$$

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

$$S_{f_x} = \frac{n_b^2 u_x \left|\mathbf{u}\right|}{h^{4/3}}, \quad S_{f_y} = \frac{n_b^2 u_y \left|\mathbf{u}\right|}{h^{4/3}},$$

where $n_b$ is the Manning coefficient. In addition, initial and boundary conditions for $h$, $q_x$ and $q_y$ have to be defined. The problem is posed in the domain $(x, y, t) \in [0, L_x] \times [0, L_y] \times (0, T]$ and the boundary $\Gamma$ of the domain is split into $\Gamma = \Gamma^D \cup \Gamma^S$ similarly to the 2D advection-diffusion problem in Section 2.2.

The hyperbolic system of partial differential equations (2.38) can be written in a conservative form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S}, \tag{2.39}$$

or

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{E} = \mathbf{S},$$

being $\mathbf{E} = (\mathbf{F}, \mathbf{G})$, where

$$\mathbf{U} = \begin{pmatrix} h \\ q_x \\ q_y \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} q_x \\ \frac{q_x^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_x q_y}{h} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{1}{2}gh^2 \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ gh(S_{z_x} - S_{f_x}) \\ gh(S_{z_y} - S_{f_y}) \end{pmatrix},$$

$\mathbf{U}$ is the vector of conserved variables, $\mathbf{F}$ and $\mathbf{G}$ are the physical fluxes and $\mathbf{S}(\mathbf{U})$ are the source terms. The Jacobian matrices of the numerical fluxes are

$$\mathbf{J}_x = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u_x^2 & 2u_x & 0 \\ -u_x u_y & u_y & u_x \end{pmatrix}, \quad \mathbf{J}_y = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 0 & 1 \\ -u_x u_y & u_y & u_x \\ c^2 - u_y^2 & 0 & 2u_y \end{pmatrix},$$

in terms of the flow velocities and the surface wave speed $c = \sqrt{gh}$. This can be projected onto the vector normal to the cell interface to obtain the normal Jacobian matrix

$$\mathbf{J}_n = \mathbf{J}_x n_x + \mathbf{J}_y n_y = \begin{pmatrix} 0 & n_x & n_y \\ c^2 n_x - u_x (\mathbf{u} \cdot \mathbf{n}) & \mathbf{u} \cdot \mathbf{n} + u_x n_x & u_x n_y \\ c^2 n_y - u_y (\mathbf{u} \cdot \mathbf{n}) & u_y n_x & \mathbf{u} \cdot \mathbf{n} + u_y n_y \end{pmatrix}.$$

Its eigenvalues $\lambda_1$, $\lambda_2$ and $\lambda_3$ and the corresponding eigenvectors $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ are

$$\lambda_1 = \mathbf{v} \cdot \mathbf{n} - c, \quad \lambda_2 = \mathbf{v} \cdot \mathbf{n}, \quad \lambda_3 = \mathbf{v} \cdot \mathbf{n} + c,$$

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ u_x - c n_x \\ u_y - c n_y \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ -c n_y \\ c n_x \end{pmatrix}, \quad \mathbf{e}_3 = \begin{pmatrix} 1 \\ u_x + c n_x \\ u_y + c n_y \end{pmatrix},$$

so that the matrices that diagonalize the Jacobian are

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 1 \\ u_x - c n_x & -c n_y & u_x + c n_x \\ u_y - c n_y & c n_x & u_y + c n_y \end{pmatrix},$$

$$\mathbf{P}^{-1} = -\frac{1}{2c} \begin{pmatrix} -\mathbf{u} \cdot \mathbf{n} - c & n_x & n_y \\ 2(u_y n_x - u_x n_y) & 2n_y & -2n_x \\ \mathbf{u} \cdot \mathbf{n} - c & -n_x & -n_y \end{pmatrix},$$

$$\mathbf{J}_n = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1}, \quad \mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}.$$

These properties are of interest for the type of methods used to discretize the equations in this thesis.

**Computational domain**

The spatial domain $[0, L_x] \times [0, L_y]$ is discretized using the FV method by means of $I_c$ volume cells of volume $S_i$ whose center positions $(x_i, y_i)$, with $i = 1, ..., I_c$. Some realistic problems require the use of unstructured grids with triangular cells due to complex terrain topographies. In Chapter 5, numerical results for several 2D examples are shown and the volume cells will have variable area and shape. In general, the volume cells can be defined as triangles or rectangles with $I_f = 3$ or 4 polygonal faces, respectively, of length $l_e$ and with outward-pointing normal vector $\mathbf{n}_e$ from each cell. The index $e$ denotes the $e$-th edge separating elements $i$ and $j$, with $e = 1, ..., I_e$, being $I_e$ the total number of edges.

In the 2D case, the discretized domain is divided into volume cells that belong to $(0, L_x)\times(0, L_y)$ and to the boundary $\Gamma = \Gamma^D \cup \Gamma^S$. The vertices are numbered with index sets $J^I, J^D$ and $J^S$ such that $i \in J^I$ if $(x_i, y_i) \in (0, L_x) \times (0, L_y)$, $i \in J^D$ if $(x_i, y_i) \in \Gamma^D$ and $i \in J^S$ if $(x_i, y_i) \in \Gamma^S$.

**Augmented Roe-based full-order model**

The full-order model (FOM) is obtained by discretizing (2.38) with the Godunov-type scheme based on the Roe method as a sum of waves [6]

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{S_i} \sum_{e=1}^{I_f} \sum_{m=1}^{3} \left( \tilde{\lambda}_m^- \tilde{\gamma}_m^- \mathbf{e}_m \right)_e^n l_e, \ i = 1, ..., I_c, \tag{2.40}$$

where $\left(\tilde{\lambda}_m^{\pm}\right)_e^n = \left(\tilde{\lambda}_m \pm |\tilde{\lambda}_m|\right)_e^n /2$, with $m = 1, 2, 3$, and the conserved variables are

$$\mathbf{U}_i^n = \begin{pmatrix} h_i^n \\ (q_x)_i^n \\ (q_y)_i^n \end{pmatrix}.$$

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

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

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

with the following numerical velocities

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

where the $i$-th and the $j$-th are the two volume cells separated by the $e$-th edge; and $\tilde{h}_e^n = \left(h_i^n + h_j^n\right)/2$; and

$$\left(\tilde{\gamma}_m^-\right)_e^n = \left(\tilde{\alpha}_m - \frac{\tilde{\beta}_m^-}{\tilde{\lambda}_m^-}\right)_e^n.$$

The wave strengths are defined as

$$\left(\tilde{\alpha}_{1,3}\right)_e^n = \frac{\delta h_e^n}{2} \mp \frac{1}{2\tilde{c}_e^n}\left[\delta q_x n_x + \delta q_y n_y - \tilde{\mathbf{u}}\cdot\mathbf{n}\delta h\right]_e^n = \frac{\delta h_e^n}{2} \mp \frac{1}{2\tilde{c}_e^n}\left(\delta\mathbf{q} - \tilde{\mathbf{u}}\delta h\right)_e^n \mathbf{n}_e,$$
$$\left(\alpha_2\right)_e^n = \frac{1}{\tilde{c}_e^n}\left[\left(\delta q_y - \tilde{u}_y\delta h\right)n_x - \left(\delta q_x - \tilde{u}_x\delta h\right)n_y\right]_e^n,$$

and the source terms

$$\left(\tilde{\beta}_m^-\right)_e^n = \left(\tilde{\beta}_m\frac{\tilde{\lambda}_m^-}{\tilde{\lambda}_m}\right)_e^n,$$

and

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

The discrete friction and bed source terms are

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

where $\left(d_n\right)_e = ||\mathbf{x}_i - \mathbf{x}_j||_2$ is the distance between the $i$-th and the $j$-th cells separated by the $e$-th edge and

$$\left(\tilde{n}_b\right)_e = \left[\left(n_b\right)_i + \left(n_b\right)_j\right]/2,$$

and

$$\begin{aligned}
&\delta h_e = h_j^n - h_i^n,\\
&\delta\left(q_x\right)_e = \left(q_x\right)_j^n - \left(q_x\right)_i^n,\\
&\delta\left(q_y\right)_e = \left(q_y\right)_j^n - \left(q_y\right)_i^n,\\
&\delta z_e = z_j - z_i
\end{aligned}$$

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

$$\Delta t = CFL\frac{\min\left\{S_i\right\}}{\max\left\{\lambda_1, \lambda_2, \lambda_3\right\}}, \tag{2.41}$$

where the CFL number satisfies $0 < CFL \leq 1$.

The ARoe-based FOM of the 2D SWEs is

$$
\begin{aligned}
h_i^{n+1} = h_i^n &- \frac{\Delta t}{S_i}\frac{1}{2}\sum_{e=1}^{I_f}\delta h_e^n\left[\tilde{\lambda}_1^-\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)+\tilde{\lambda}_3^-\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\frac{1}{2}\sum_{e=1}^{I_f}\left(\delta q_x\right)_e^n\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_e^n\left(n_x\right)_e l_e \\
&+\frac{\Delta t}{S_i}\frac{1}{2}\sum_{e=1}^{I_f}\left(\delta q_y\right)_e^n\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_3^-}{\tilde{c}}\right)_e^n\left(n_y\right)_e l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_e^n\frac{\delta z_e}{\tilde{c}_e^n}l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\frac{\tilde{\mathbf{u}}_e^n\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^n|}{\tilde{c}_e^n\max\left(h_i^n,h_j^n\right)^{4/3}}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_e^n\left(\tilde{n}_b^2 d_n\right)_e l_e, \\
\left(q_x\right)_i^{n+1} = \left(q_x\right)_i^n &- \frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\delta h_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_x-\tilde{c}n_x}{2}\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right. \\
&\left.+\tilde{\lambda}_2^-\left(\tilde{u}_y n_x-\tilde{u}_x n_y\right)n_y+\tilde{\lambda}_3^-\frac{\tilde{u}_x+\tilde{c}n_x}{2}\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\left(\delta q_x\right)_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_x-\tilde{c}n_x}{2\tilde{c}}n_x-\tilde{\lambda}_2^- n_y n_y-\tilde{\lambda}_3^-\frac{\tilde{u}_x+\tilde{c}n_x}{2\tilde{c}}n_x\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\left(\delta q_y\right)_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_x-\tilde{c}n_x}{2\tilde{c}}n_y+\tilde{\lambda}_2^- n_x n_y-\tilde{\lambda}_3^-\frac{\tilde{u}_x+\tilde{c}n_x}{2\tilde{c}}n_y\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\frac{\delta z_e}{\tilde{c}_e^n}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_x-\tilde{c}n_x\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_x+\tilde{c}n_x\right)\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\frac{\tilde{\mathbf{u}}_e^n\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^n|\left(\tilde{n}_b^2 d_n\right)_e}{\tilde{c}_e^n\max\left(h_i^n,h_j^n\right)^{4/3}}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_x-\tilde{c}n_x\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_x+\tilde{c}n_x\right)\right]_e^n l_e, \\
\left(q_y\right)_i^{n+1} = \left(q_y\right)_i^n &- \frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\delta h_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2}\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right. \\
&\left.-\tilde{\lambda}_2^-\left(\tilde{u}_y n_x-\tilde{u}_x n_y\right)n_x+\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2}\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\left(\delta q_x\right)_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2\tilde{c}}n_x+\tilde{\lambda}_2^- n_x n_y-\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2\tilde{c}}n_x\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\sum_{e=1}^{I_f}\left(\delta q_y\right)_e^n\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2\tilde{c}}n_y-\tilde{\lambda}_2^- n_x n_x-\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2\tilde{c}}n_y\right]_e^n l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_y-\tilde{c}n_y\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_y+\tilde{c}n_y\right)\right]_e^n\frac{\delta z_e}{\tilde{c}_e^n}l_e \\
&+\frac{\Delta t}{S_i}\frac{g}{2}\sum_{e=1}^{I_f}\tilde{h}_e^n\frac{\tilde{\mathbf{u}}_e^n\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^n|\left(\tilde{n}_b^2 d_n\right)_e}{\tilde{c}_e^n\max\left(h_i^n,h_j^n\right)^{4/3}}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_y-\tilde{c}n_y\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_y+\tilde{c}n_y\right)\right]_e^n l_e,
\end{aligned}
\tag{2.42}
$$

with $i \in J^I$.

**Numerical corrections**

Most of the numerical corrections proposed for the 1D SWE in the Section 2.4.4 have a straightforward extension from 1D to 2D. For the sake of simplicity, these considerations will be omitted, referring the interested reader to the bibliography [44], except for the wet/dry treatment that requires a special modification.

In 2D problems, the information is bounced back to the wet cell in the same way as explained in the 1D version of the first step (2.32) of the wet/dry treatment. However, when working with meshes that are not aligned with the $x$- and $y$ axes (i.e. is unstructured), the extension of the second step (2.33) of the treatment presented in Section 2.4.4 is not trivial. It is necessary to carefully cancel the perpendicular component to the wet/dry wall and keep the parallel component. For this purpose, the strategy considered in this thesis consists of a geometrical transformation of the flow discharges defined in the $x$ and $y$ directions, which is applied after the time update step. This procedure is detailed below.

An example can be found in the Figure 2.6, where an arbitrary velocity direction is defined as

$$\vec{u} = u_x\hat{x} + u_y\hat{y},$$

and a normal vector to the wall that points outwards expressed in the OXY axis as

$$\hat{n} = n_x\hat{x} + n_y\hat{y}.$$

Angle $\alpha$ from Figure 2.6 can be computed by means of the trigonometric identities as follows

$$\alpha = \begin{cases} \arctan\left(\dfrac{n_y}{n_x}\right), & \text{if } n_y \geq 0, \\ \pi + \arctan\left(\dfrac{n_y}{n_x}\right), & \text{if } n_y < 0. \end{cases}$$

Once this angle is obtained, a new reference system can be defined as

$$\begin{cases} \hat{n}_\perp = \cos(\alpha)\,\hat{x} + \sin(\alpha)\,\hat{y}, \\ \hat{n}_\parallel = \sin(\alpha')\,\hat{x} + \cos(\alpha')\,\hat{y}; \end{cases}$$

with

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

The velocity and normal vectors can be expressed in this new reference system as

$$\begin{cases} \vec{u}_n = u_\perp\hat{n}_\perp + u_\parallel\hat{n}_\parallel, \\ \hat{n}_n = \hat{n}_\perp; \end{cases}$$

where

$$\begin{cases} u_\perp = \hat{n}_\perp \vec{u} = u_x \cos(\alpha) + u_y \sin(\alpha) \\ u_\parallel = \hat{n}_\parallel \vec{u} = u_x \sin(\alpha') + u_y \cos(\alpha') \,. \end{cases}$$

The wet/dry treatment imposes a zero velocity in the perpendicular direction to the wall, i.e. $u_\perp = 0$, so that $\vec{u}_n = u_\parallel \hat{n}_\parallel$. The velocity has to be redefined including only the parallel component

$$\vec{u}^{WD} = u_\parallel \hat{n}_\parallel = \left(u_\parallel\right)_x \hat{x} + \left(u_\parallel\right)_y \hat{y},$$

where

$$\begin{cases} \left(u_\parallel\right)_x = \hat{x} u_\parallel \hat{n}_\parallel = u_\parallel \sin(\alpha) = \hat{n}_\parallel \vec{u} = u_x \sin^2(\alpha) + u_y \cos(\alpha)\sin(\alpha) \\ \left(u_\parallel\right)_y = \hat{y} u_\parallel \hat{n}_\parallel = u_\parallel \cos(\alpha') = \hat{n}_\parallel \vec{u} = u_x \sin(\alpha)\cos(\alpha') + u_y \cos^2(\alpha') \end{cases}$$

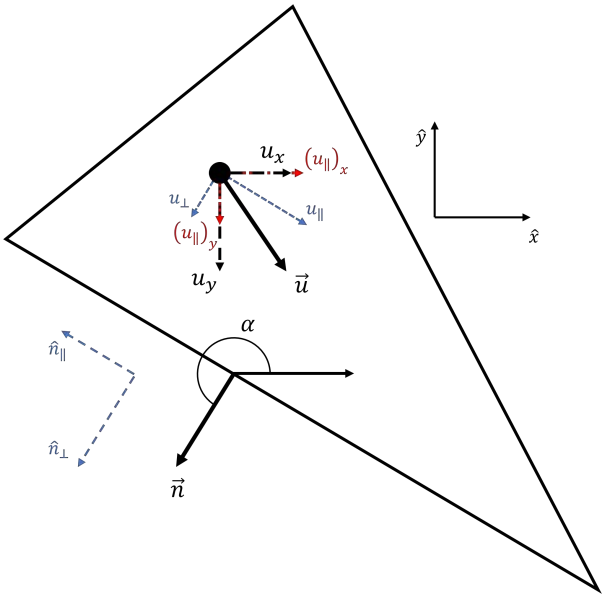



Figure 2.6: Wet/dry treatment: cancellation of the normal velocity in 2D cases.

This second part of the wet/dry treatment is incorporated into the FOM resolution after the update of the variables. Once the water discharges in both directions have been calculated using the numerical scheme, they are redirected to cancel the components perpendicular to the walls that meet the wet/dry conditions. For this purpose, as many intermediate update steps are necessary as the number of walls in each cell as follows

$$\begin{aligned} \left(q_x\right)_i^{n+1,m} &= a_{i,m}^{WD} \left(q_x\right)_i^{n+1,m-1} + b_{i,m}^{WD} \left(q_y\right)_i^{n+1,m-1}, \\ \left(q_y\right)_i^{n+1,m} &= c_{i,m}^{WD} \left(q_x\right)_i^{n+1,m-1} + d_{i,m}^{WD} \left(q_y\right)_i^{n+1,m-1}, \end{aligned} \tag{2.43}$$

where $m = 1, ..., I_f$ and $i = 1, ..., I_e$; with $\left(q_x\right)_i^{n+1,0}$ and $\left(q_y\right)_i^{n+1,0}$ are the update states given by the numerical scheme; and where the coefficients are

$$
\begin{aligned}
a_{i,m}^{WD} &= \begin{cases} \sin(\alpha')\sin(\alpha'), & \text{if treated,} \\ 1, & \text{otherwise,} \end{cases} \quad b_{i,m}^{WD} = \begin{cases} \sin(\alpha')\cos(\alpha'), & \text{if treated,} \\ 0, & \text{otherwise,} \end{cases} \\
c_{i,m}^{WD} &= \begin{cases} \cos(\alpha')\sin(\alpha'), & \text{if treated,} \\ 0, & \text{otherwise,} \end{cases} \quad d_{i,m}^{WD} = \begin{cases} \cos(\alpha')\cos(\alpha'), & \text{if treated} \\ 1, & \text{otherwise.} \end{cases}
\end{aligned}
\tag{2.44}
$$

and where the $\alpha$ angle is computed as indicated before.

# Chapter 3

# Reduced-order models for linear problems

The application of POD-based ROMs to time-dependent linear problems is presented in this chapter.

In Section 3.1, theoretical aspects of ROMs are presented. First, the general strategy for training and solving ROMs and the POD methodology are explained. Secondly, a step-by-step example of the development of the ROM numerical scheme applied to the linear advection-diffusion equation (2.1) and (2.7) are considered. Finally, three preliminary test cases are presented in which the process of solving the ROM is detailed, showing the results obtained in detail.

Hereafter, Section 3.2 deals with the combination of ROMs with high-order numerical schemes, paying special attention to the order of convergence achieved by the ROM solutions. This study allows checking, by means of combinations between high and low order, whether ROMs maintain the accuracy of the solutions with which they have been trained or the FOM from which they have been trained determines it.

In Section 3.3, the ability of the ROMs presented above to solve parametrized problems is studied. To achieve this, it is necessary to generate a large training sample with solutions computed by FOMs. How large such a sample has to be is studied. The numerical results included in this section illustrate the range of parametric application that ROMs can cover, including advection velocity, diffusion coefficient, initial and boundary conditions.

Section 3.4 in turn tests the possibility of solving ROMs beyond the training time by means of an appropriate coordinate transformation. It is detailed in the different problems proposed together with their results.

## 3.1 Basic aspects

### 3.1.1 Reduced-order modelling strategy

The reduced order modeling strategy consists of two phases: 1) the off-line phase, in which the ROM is trained and whose computational cost is not accounted for. The training phase itself can be divided into three sub-phases: 1.1) resolution of the FOM; 1.2) construction/definition of the reduced space by means of the POD; and 1.3) development of the ROM; and 2) the

on-line phase, in which the ROM is solved and whose low computational time justifies the use of ROMs, since it accelerates that of the FOM by several orders of magnitude.

The following is a detailed explanation of each of these phases and their component parts, since they contain the theoretical principles on which the reduced order models are based.

**1.1) Resolution of the full-order model**

The FOM, which in this thesis is chosen from those presented in Chapter 2, is numerically integrated in time step-by-step up to the training time, as illustrated by Figure 3.1. A general formulation of a explicit-in-time FOM can be expressed as follows

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\mathcal{L}(u_{i-1}^n, u_i^n, u_{i+1}^n),$$

being $u_i^n$ the variable of interest, with $i = 2, ..., I_x - 1$, with a special treatment for the boundary volume cells. The operator $\mathcal{L}(u_{i-1}^n, u_i^n, u_{i+1}^n)$ represents the method chosen for the spatial discretisation, namely the Godunov method [64].

In general, in this thesis it is assumed that the training time is equal to the final time for which the ROM is solved, $T_{\text{train}} = T$. However, it is very interesting to try to take the ROM beyond the training time. This is the focus of Sections 3.4, 4.3 and 5.4.

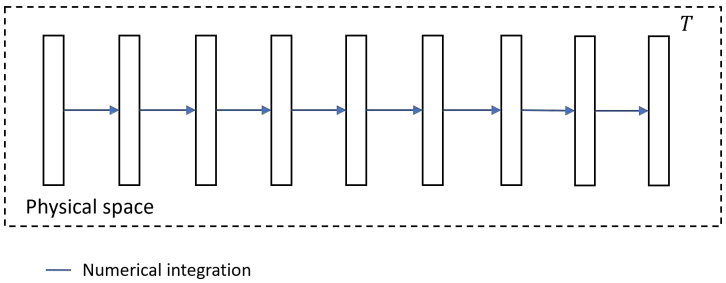



Figure 3.1: Phase 1.1). Resolution of the FOM.

**1.2) Construction/definition of the reduced space**

The set of $N_T$ time numerical solutions computed by the FOM, or *training solutions*, are assembled in the so-called *snapshot matrix*

$$\mathbf{U} = \begin{pmatrix} u_1^1 & u_1^2 & \cdots & u_1^{N_T} \\ u_2^1 & u_2^2 & \cdots & u_2^{N_T} \\ \vdots & \vdots & \ddots & \vdots \\ u_{I_x}^1 & u_{I_x}^2 & \cdots & u_{I_x}^{N_T} \end{pmatrix} \in \mathbb{R}^{I_x \times N_T}$$

The *proper orthogonal decomposition* (POD, [171]) of $\mathbf{U}$ by means of the *singular value decomposition* (SVD, [65]) decomposes the snapshot matrix into orthogonal components, also called *POD modes*

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

where $\mathbf{\Sigma} \in \mathbb{R}^{I_x \times N_T}$ is a diagonal matrix whose entries of the main diagonal are the singular values of $\mathbf{U}$ and represent the magnitude of each POD mode; and $\mathbf{\Phi} \in \mathbb{R}^{I_x \times I_x}$ and $\mathbf{\Psi} \in \mathbb{R}^{N_T \times N_T}$ are orthogonal matrices. The matrix $\mathbf{\Phi} = (\phi_1, \ldots, \phi_{I_x})$ with $\phi_k = (\phi_{1,k}, \ldots, \phi_{I_x,k})^T$ consists of the orthogonal eigenvectors of $\mathbf{U}\mathbf{U}^T$ which are used to define the reduced space, and satisfy

$$(\phi_i, \phi_j) = \delta_{ij}, \quad \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

The POD modes that are sorted in descending order according to the amount of *information* they represent in relation to the total solution, so that, the singular values

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_K > 0,$$

with $K$ the maximum number of POD modes, which is equal to the rank of the snapshot matrix and therefore $K \leq \min(I_x, N_T)$.

Due to this accumulation of most of the energy/information in the principal POD modes, it is feasible to truncate the dimension of the reduced space without losing much precision with respect to the snapshot matrix. Let $M_{\text{POD}}$ be a positive integer such that $M_{\text{POD}} \ll K$ and it will be chosen as small as possible without significantly affecting the accuracy of the computed solution with the reduced-order method [5, 12, 73, 110].

The number of POD modes $M_{\text{POD}}$ determines the percentage of captured information $M_{\text{POD}}^{\text{MAX}}$

$$P(M_{\text{POD}}) = \sum_{k=1}^{M_{\text{POD}}} \sigma_k^2 / \sum_{k=1}^{K} \sigma_k^2 \times 100. \tag{3.1}$$

Thus, if $M_{\text{POD}} = K$, all the information is recovered (i.e., $P(K) = 100\%$). This can be used as an a priori criterion to select $M_{\text{POD}}$ to solve the ROM with a given accuracy.

In short, the construction/definition of the reduced space consists of applying the SVD to the snapshot matrix and truncating the number of basis functions obtained so that it is sufficiently smaller than the physical space, as illustrated in Figure 3.2

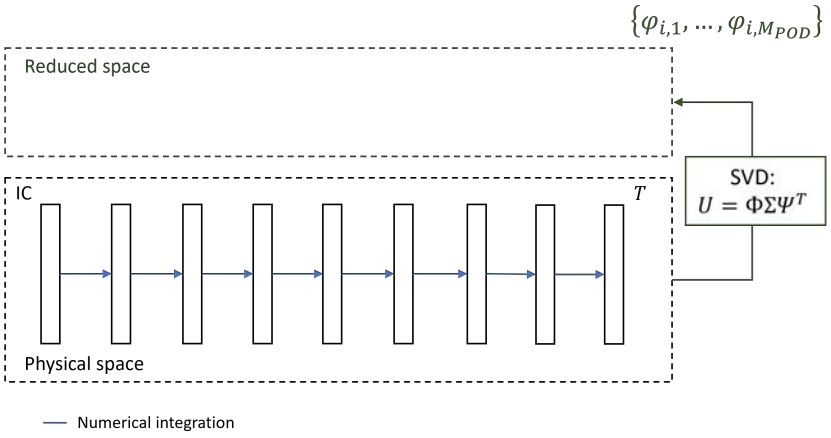



Figure 3.2: Phase 1.2). Construction/definition of the reduced space.

**1.3) Development of the reduced-order model**

The intrusive reduced-order model (ROM) based on the POD method is an alternative numerical scheme that needs to be developed from a standard numerical scheme by projecting it from the physical space to the reduced space. The Galerkin method [56] acts as the projection between these two spaces

$$u_i^n \approx \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n \phi_{i,k}, \;\; i = 1, ..., I_x. \tag{3.2}$$

where $\{\phi_1, ..., \phi_{M_{\mathrm{POD}}}\}$ are the reduced space basis functions provided by the POD/SVD; and $\hat{u}_k^n$ with $k = 1, ..., M_{\mathrm{POD}}$ are the reduced coefficients that depend on time. If the maximum number of POD modes is used, $M_{\mathrm{POD}} = K$, the reconstruction of the Galerkin method recovers the solution in the physical space exactly, i.e. (3.2) it is an equality.

The ROM is developed from the numerical expression of the FOM by applying the Galerkin method (3.2) to the variables of interest and projecting it to the reduced space (first by multiplying it by a basis function $\phi_{i,p}$ and then by integrating it to the whole physical space). An example of the step-by-step development of a ROM is shown in detail below.

The author would like to point out at this point that this development process is highly complex and contains many of the subtleties and conditions that determine how well ROMs perform. It is therefore one of the main interests of this thesis. A great effort has been put into the development of ROMs from different equations and FOMs in order to explore all the possibilities that they offer and the limitations that they present and how to overcome them.

Once the general formulation of the FOM given above has been projected into the reduced space, the following general formulation of the ROM is obtained

$$\begin{aligned}
\hat{u}_i^{n+1} &= \hat{u}_i^n - \frac{\Delta t}{\Delta x} \sum_{i=1}^{I_x} \mathcal{L}(u_{i-1}^n, u_i^n, u_{i+1}^n) \phi_{i,p} \\
&= \hat{u}_i^n - \frac{\Delta t}{\Delta x} \sum_{i=1}^{I_x} \sum_{k=1}^{M_{\mathrm{POD}}} \mathcal{L}(\phi_{i-1,k}, \phi_{i,k}, \phi_{i+1,k}) \hat{u}_k^n \phi_{i,p} \\
&= \hat{u}_i^n - \frac{\Delta t}{\Delta x} \hat{\mathcal{L}} \left( \{\phi_{i,k}\}_{i=1,...,I_x}^{k=1,...,M_{\mathrm{POD}}}, \{\hat{u}_k^n\}^{k=1,...,M_{\mathrm{POD}}} \right)
\end{aligned}$$

The ROM training, in other words, consists, essentially, in constructing operator $\hat{\mathcal{L}}$ that acts as the transfer function of the numerical method in the reduced space. This discrete operator consists of all the basis functions obtained from the SVD, $\left(\{\phi_{i,k}\}_{i=1,...,I_x}^{k=1,...,M_{\mathrm{POD}}}\right)$, and the reduced solution evaluated in all POD modes, $\left(\{\hat{u}_k^n\}^{k=1,...,M_{\mathrm{POD}}}\right)$. Therefore, it depends on the physical and reduced spaces, but not on time. Thus, the ROM, which inhabits the reduced space, is integrated in time, but is decoupled from the physical space.

**2) Resolution of the reduced-order model**

Once the reduced space is properly defined and the ROM is developed, it can be integrated in time, as shown in Figure 3.3. First, the IC has to be projected from the physical space to the

reduced space by means of the Galerkin method (3.2). Then, it is evolved in time up to the final time. And, finally, the final solution is projected back to the physical space.

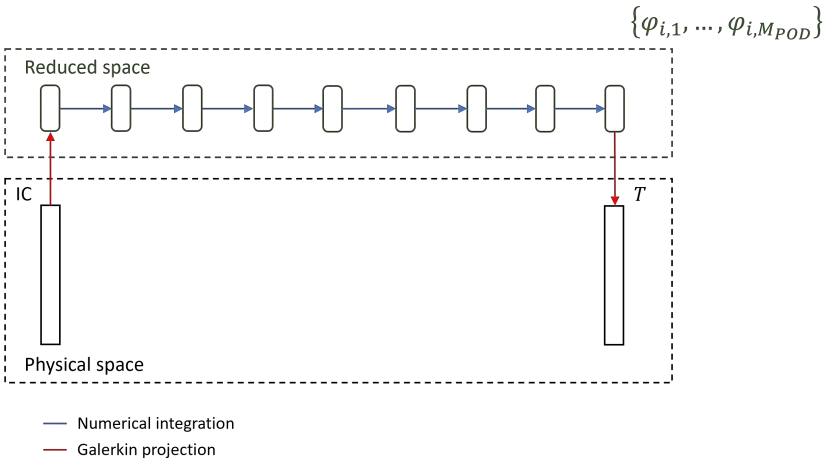



Figure 3.3: Phase 2). Resolution of the reduced-order model.

### 3.1.2 Detailed example of the development of a ROM

In this section, the detailed development of a ROM step by step is presented. For the sake of simplicity, the 1D linear advection-diffusion equation (2.1) will be the starting point. This equation is discretized by means of the Godunov scheme to obtain the first-order upwind FOM (2.7)-(2.7), or FOU-based FOM.

Even though the FOM is discretized by means of the FVM, a special formulation of it is required to develop the ROM which is more similar to the Finite Difference method.

**Development of the ROM setp by step**

The intrusive POD-based ROM is developed following the steps below.

i) Application of the Galerkin method.

The conserved variable in the FOU-based FOM is replaced with the Galerkin method (3.2). In the case of the inner cells, this yields

$$\begin{aligned}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{i,k} = \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{i,k} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k} - \phi_{i-1,k}\right) \\ + \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right) \\ + \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right),\end{aligned} \tag{3.3}$$

with $2 \leq i \leq I_x - 1$.

If, for example, free BCs are considered, the equations of the boundary volume cells would be

$$\begin{aligned}
\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{1,k} = &\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{1,k} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right) \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right) \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right), \\
\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{I_x,k} = &\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{I_x,k} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right) \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right) \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right).
\end{aligned} \tag{3.4}$$

ii) Projection of the equation into the reduced space.

For this purpose, equation (3.3) is multiplied by $\phi_{i,p}$ (the $p$-th component of the vector $\phi_i$ of the POD basis) and (3.4) by $\phi_{1,p}$ and $\phi_{I_x,p}$, respectively

$$\begin{aligned}
\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{1,k}\phi_{1,p} = &\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{1,k}\phi_{1,p} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p} \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p} \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}, \\
\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{i,k}\phi_{i,p} = &\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{i,k}\phi_{i,p} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p} \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}, \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}, \\
\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{I_x,k}\phi_{I_x,p} = &\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{I_x,k}\phi_{I_x,p} - \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p} \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p} \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}.
\end{aligned} \tag{3.5}$$

with $2 \leq i \leq I_x - 1$.

iii) Integration to the entire physical space.

This results from summing all the previous equations (3.5), including the inner and the boundary volume cells

$$\begin{aligned}
\sum_{i=1}^{I_x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{i,k}\phi_{i,p} = &\sum_{i=1}^{I_x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n}\phi_{i,k}\phi_{i,p} \\
&- \frac{1}{2}a\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{u}_k^n\Bigg[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p} \\
&\qquad + \sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p} + \left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\Bigg] \\
&+ \frac{1}{2}|a|\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{u}_k^n\Bigg[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p} \\
&\qquad + \sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p} - \left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\Bigg] \\
&+ \nu\frac{\Delta t}{\Delta x^2}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{u}_k^n\Bigg[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p} \\
&\qquad + \sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p} - \left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\Bigg]
\end{aligned} \tag{3.6}$$

Because of this spatial integration, it is essential to include the discrete equations of the boundary volume cells. If Neumann or periodic BCs are imposed, then they can be easily incorporated to this development, as explained. However Dirichlet BCs are not that easy, as it will be shown in the next section.

Since the POD basis is orthogonal by definition, $(\phi_i, \phi_j) = \delta_{ij}$

$$\sum_{i=1}^{I_x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\phi_{i,k}\phi_{i,p} = \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^{n+1}\sum_{i=1}^{I_x}\phi_{i,k}\phi_{i,p} = \hat{u}_p^{n+1};$$

and the second term of (3.6) is treated similarly. As a result of the projection, the rest of the terms of (3.6) are not simplified in the same way and the following expression for the ROM is obtained

$$\hat{u}_p^{n+1} = \hat{u}_p^n + \Delta t \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n A_{pk},$$

where the coefficient is

$$
\begin{aligned}
A_{pk} = &-\frac{1}{\Delta x}\frac{1}{2}a\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p}+\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right]\\
&+\frac{1}{\Delta x}\frac{1}{2}|a|\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}-\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right]\\
&+\frac{1}{\Delta x^2}\nu\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}-\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right].
\end{aligned}
$$

Since the ROM use the same numerical grid for the physical space, the width of the cells $\Delta x$ can be included in the coefficients that are calculated during the off-line phase.

**Dirichlet BCs**

When considering time-dependent Dirichlet boundary conditions, the development of the ROM is not directly feasible. The combination of time and space dependence makes it impossible to model it only during the training phase: it is necessary that the ROM receives information about the time evolution of the boundary instantaneously. There are some proposals in the literature for dealing with this type of BCs [68, 72]. However, in this thesis, it is proposed a resolution of the Dirichlet boundary conditions integrated in the ROM that allows to solve it without losing the imposed space-time dependencies and speed-up. In addition, it achieves an optimal level of accuracy.

If a Dirichlet BC is considered for (2.1) at $x = 0$ (in the left boundary) and a free BC is imposed at $x = L$ (in the right boundary), the following applies

$$
\sum_{i=1}^{I_x} u_i^n \phi_{i,p} = v^n \phi_{1,p} + \sum_{i=2}^{I_x}\sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n \phi_{i,k}\phi_{i,p}.
$$

where $v^n$ is the time-dependent function imposed on the left boundary. Thus, the FOU-based ROM of the 1D advection-diffusion equation reads

$$
\hat{u}_p^{n+1} = v^n \phi_{1,p} + \sum_{q=1}^{M_{\mathrm{POD}}} A_{pk}\hat{u}_q^n + \Delta t \sum_{k=1}^{M_{\mathrm{POD}}} B_{pk}\hat{u}_k^n, \tag{3.7}
$$

where the coefficients are

$$
\begin{aligned}
A_{pk} &= \sum_{i=2}^{I_x}\phi_{i,k}\phi_{i,p},\\
B_{pk} &= b_{I_x}\phi_{I_x,p} - \frac{1}{\Delta x}\frac{a}{2}\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p}\\
&\quad+\frac{1}{\Delta x}\frac{|a|}{2}\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}\\
&\quad+\frac{1}{\Delta x^2}\nu\sum_{i=2}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}.
\end{aligned} \tag{3.8}
$$

From now on, Neumann and periodic BCs will be given the following notation $b_i$, where $i \in \{1, I_x\}$, depending on either they are considered to be of Dirichlet or not, and are reduced as mentioned before. If, for example, free BCs are imposed on the right, they are given by

$$\begin{aligned} b_{I_x} = &-\frac{a}{2\Delta x}\left(\phi_{I_x,k} - \phi_{I_x-1,k}\right)\phi_{I_x,p} \\ &-\frac{|a|}{2\Delta x}\left(\phi_{I_x,k} - \phi_{I_x-1,k}\right)\phi_{I_x,p} \\ &-\frac{\nu}{\Delta x^2}\left(\phi_{I_x,k} - \phi_{I_x-1,k}\right)\phi_{I_x,p}. \end{aligned} \tag{3.9}$$

### 3.1.3 Extension to 2D problems

An example of extending the development of a ROM to the 2D advection-diffusion equation (2.13) is shown in this section.

It is necessary to take into account that the BC of 2D problems increases the difficulty considerably. It has to be distinguished if the volume cell is in the interior of the domain or in the boundary $\Gamma^D$ or $\Gamma^S$.

If all this is taken into account and the Galerkin method is applied to the FOU-based FOM of the 2D advection-diffusion equation (2.14), the final expression of the FOU-based ROM reads

$$\hat{u}_p^{n+1} = \sum_{i,j\in J_D} u_0^n \phi_{i,j,p} + \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n A_{pk} + \Delta t \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n B'_{pk}, \tag{3.10}$$

where the coefficients are

$$\begin{aligned} A_{pk} = &\sum_{i,j\in J^I\cup J^S} \phi_{i,j,k}\phi_{i,j,p}, \\ B_{pk} = &\frac{1}{\Delta x}\sum_{i,j\in J^S} b_{i,j}\phi_{i,j,p} - \frac{1}{2}a_x \sum_{i,j\in J^I}\left(\phi_{i+1,j,k} - \phi_{i-1,j,k}\right)\phi_{i,j,p} \\ &+\frac{1}{2}|a_x| \sum_{i,j\in J^I}\left(\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}\right)\phi_{i,j,p} \\ &-\frac{1}{2}a_y \sum_{i,j\in J^I}\left(\phi_{i,j+1,k} - \phi_{i,j-1,k}\right)\phi_{i,j,p} \\ &+\frac{1}{2}|a_y| \sum_{i,j\in J^I}\left(\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}\right)\phi_{i,j,p}; \end{aligned}$$

where the coefficients $b_{i,j}$ are given by the BCs considered, similarly to 1D cases (3.9).

### 3.1.4 Preliminary numerical results

This last part of the section will serve to explain the above-mentioned properties of the ROM in more detail with the examples of a series of numerical results. Aspects such as the magnitude

of the singular values and what percentage of the total solution they represent, according to (3.1), as well as the form of the basis functions corresponding to the main POD modes, will be analysed.

The *accuracy* obtained by the solutions $u^{\text{ROM}}$ calculated by the ROM will be evaluated. This is measured by means of the differences with respect to the solutions $u^{\text{FOM}}$ calculated by the FOM measured using the $L_1$ and $L_\infty$ norms. Recall that, the differences between them in terms of the aforementioned norms are defined as

$$\|d\|_1 = \Delta x \sum_{i=1}^{I_x} |u_i^{\text{FOM}} - u_i^{\text{ROM}}|, \quad \|d\|_\infty = \max_{1 \leq i \leq I_x} \left( |u_i^{\text{FOM}} - u_i^{\text{ROM}}| \right). \tag{3.11}$$

These differences can be computed at each time step, so the time evolution of the errors can be visualized. Throughout the rest of thesis the accuracy is always measured like this.

It will be also possible to study the *speed-up* achieved by the ROM by means of diving by its required CPU time, $\tau_{\text{CPU}}^{\text{ROM}}$, that of the FOM,$\tau_{\text{CPU}}^{\text{FOM}}$, both measured in seconds. The concept of *efficiency* will refer then to relation between these two previous magnitudes.

For this purpose, three problems have been proposed:

- Test case 1 (TC1). A Gaussian profile is defined as the initial condition (IC) of this 1D case. It is advected with constant velocity.
- Test case 2 (TC2). A Gaussian profile is imposed as a boundary condition (BC) in the left boundary. The 1D domain is initially at rest.
- Test case 3 (TC3). A Gaussian profile is defined as the IC in a 2D domain. It is advected in space with constant velocity in a diagonal direction.

**Test case 1. 1D advection of an initial Gaussian profile**

In this test case, an initial 1D Gaussian profile is transported along the $x-$axis with constant speed, $a = 1$, and no diffusion, $\nu = 0$.

The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, 8]$, so that $T_{\text{train}} = T$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-(x-6)^2}, \tag{3.12}$$

Free BCs are considered. The IC is shown in Figure 3.4.

To facilitate the understanding of the problem, Table 3.1 contains the essential data about the setting of either the physical problem, the numerical parameters of the FOM, or those of the ROM. From now on, all test cases will present a similar table.

| $L$ | $T$ | $a$ | $\nu$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 20 | 8 | 1 | 0 | Eq. (3.12) | Free | 200 | 1 | 82 | Tab. 3.2 |

Table 3.1: TC1. Problem settings.

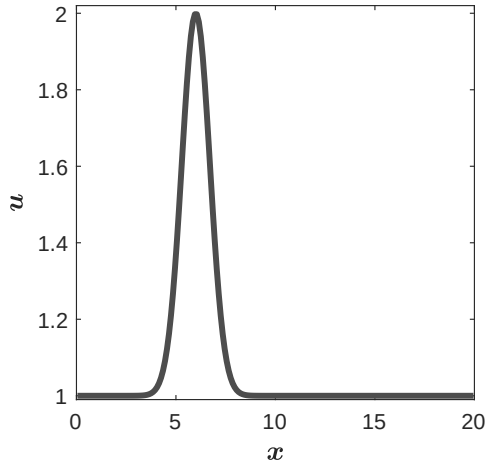



Figure 3.4: TC1. Initial condition and bed elevation.

*Resolution of the FOM*

In this case, an exact solution exists which can be used to prove how good the solutions computed by the FOM and by the ROM are. Therefore, it is necessary to set the CFL number to 1 so that the numerical method does not introduce numerical diffusion. The time steps are dynamically computed according to the CFL condition (2.5).

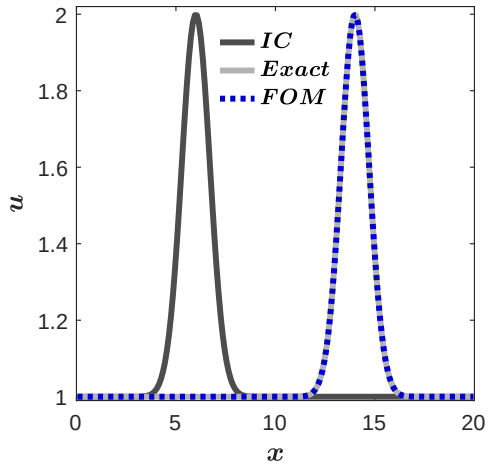



Figure 3.5: TC1. FOM and exact solutions.

The spatial domain is discretized in $I_x = 200$ volume cells. Figure 3.5 shows the solution computed by the FOU-based FOM (2.7) at the final time $T$. Different time steps are plotted in thin grey solid lines together with the solution at the final time $T = 8$ with a thick blue dotted line. The exact solution, represented by the thick pale gray solid line proves the good performance of the FOM.

*Definition of the reduced space*

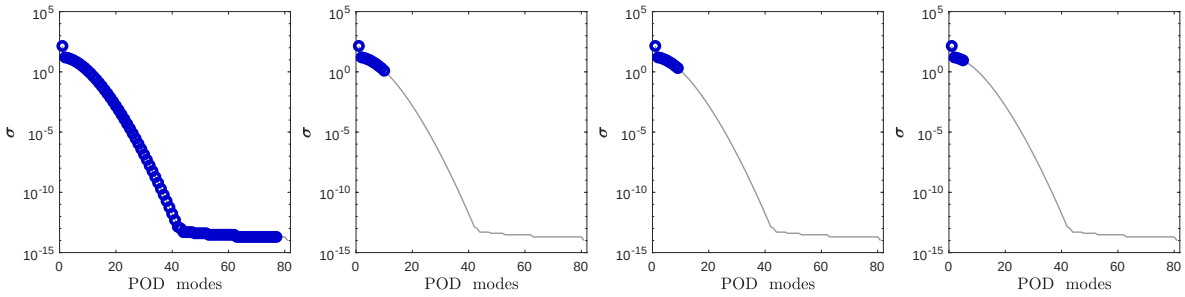
As seen in Section 3.1.1, the number of elements of the POD basis depends on the number of POD modes chosen for the ROM to solve. According to the criterion (3.1), Table 3.2 shows the

number of POD modes necessary to recover different percentages of information, $P(M_{\text{POD}})$. Notice that the 100% of the information is recovered using a non-maximum POD modes; i.e., $M_{\text{POD}} = 77 = \text{rank}\,(\mathbf{U}) < \min\,(I_x, N_T)$. This means that the rank of the snapshot matrix is smaller than its dimension.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P(M_{\text{POD}})$ | 100 | 99.9960 | 99.9884 | 99.5957 |
| $M_{\text{POD}}$ | 77 | 10 | 9 | 5 |

Table 3.2: TC1. Number of POD modes necessary to recover different percentages of information.

Figure 3.7 shows the first 10 functions of the POD basis. The POD singular values chosen for each percentage are plotted in Figure 3.6 with blue circles on a grey line that represents the total.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.6: TC1. POD singular values.

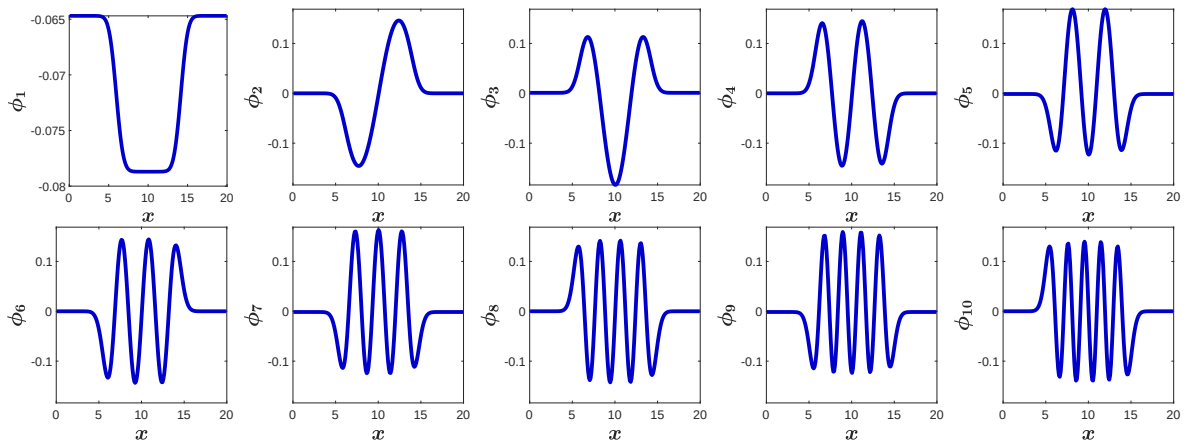


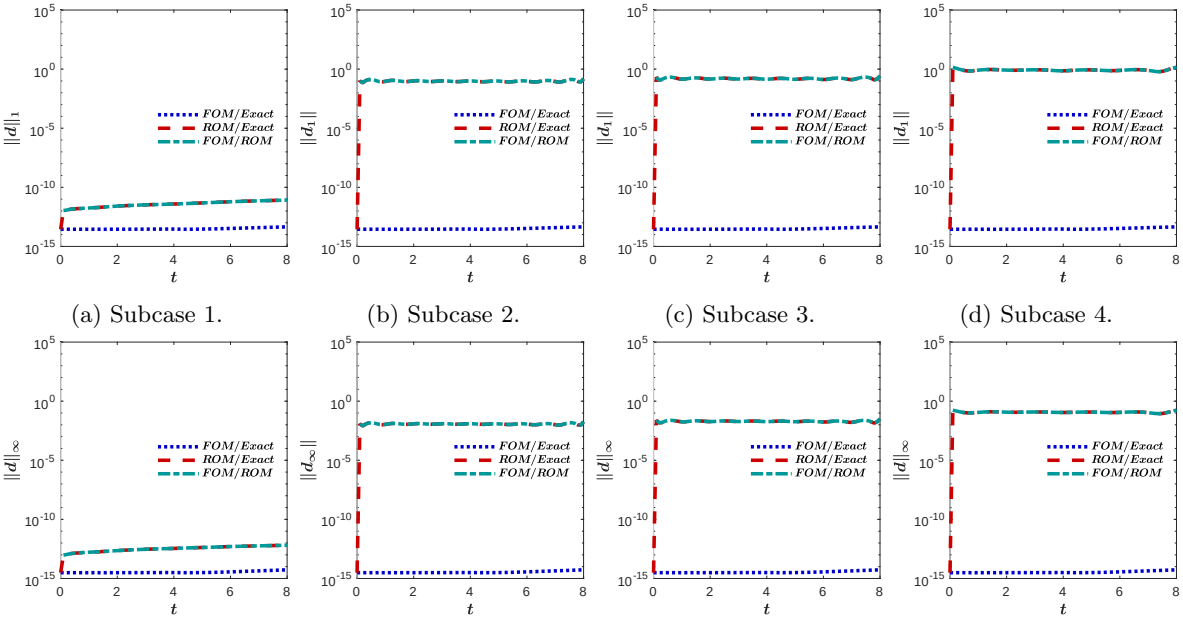

Figure 3.7: TC1. First POD basis functions.

*Resolution of the ROM and efficiency test*

In order to evaluate that the ROM is working correctly, it should first be checked that, by solving the maximum number of POD modes, it is able to recover the solution of the FOM

with machine precision.

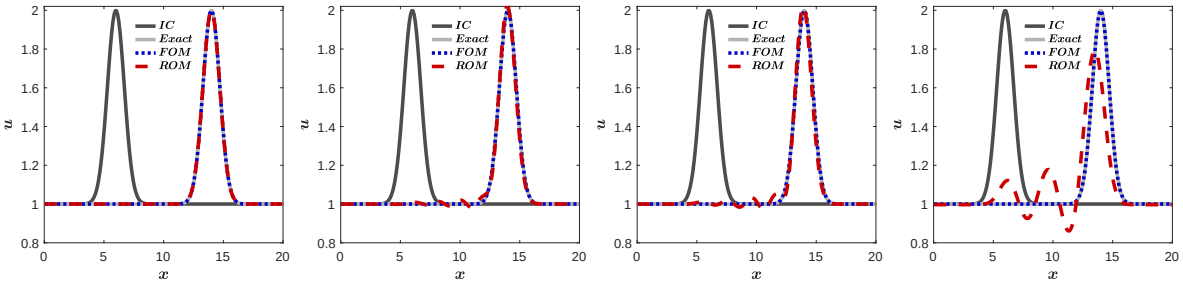
Figure 3.9a shows the solution computed by the ROM using the maximum number of POD modes. The error made by the FOM and the ROM solutions with respect to the exact solution over time are plotted in Figures 3.8a and 3.8e, including the differences between them. Both solutions achieve machine precision in terms of accuracy. The rest of subcases are shown in the remaining plots of Figures 3.9 and 3.8. It can be seen in those figures that error increases as $M_{\text{POD}}$ decreases.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

(e) Subcase 1. (f) Subcase 2. (g) Subcase 3. (h) Subcase 4.

Figure 3.8: TC1. Differences measured with $\|d\|_1$ (top row) and $\|d\|_\infty$ (bottom row).




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.9: TC1. ROM solutions.

Table 3.3 shows the all the errors at the final time of all subcases. These results make clear that there is no relationship between the a priori error understood as the percentage of information represented by the POD modes solved by the ROM and the error measured with respect to the exact solution.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $8.31 \cdot 10^{-12}$ | $1.52 \cdot 10^{-1}$ | $2.66 \cdot 10^{-1}$ | $1.47$ |
| $\|d\|_\infty$ | $6.48 \cdot 10^{-13}$ | $1.83 \cdot 10^{-2}$ | $3.17 \cdot 10^{-2}$ | $1.75 \cdot 10^{-1}$ |

Table 3.3: TC1. Errors made by each subcase.

Once the ROM is proven to perform as the FOM when using the maximum number of POD modes, it is possible to focus on the efficiency of the ROM with respect to the FOM. The CPU time required by the FOM to complete this computation is $t^{\text{CPU}} = 4.5 \cdot 10^{-4}$. Table 3.4 shows that the ROM, when using the maximum number of POD modes, is much slower than the FOM, since the speed-up of subcase 1 is smaller than 1.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| CPU time (s) | $2.2 \cdot 10^{-3}$ | $4.7 \cdot 10^{-5}$ | $3.8 \cdot 10^{-5}$ | $1.4 \cdot 10^{-5}$ |
| Speed-up | $\times 0$ | $\times 9$ | $\times 12$ | $\times 32$ |

Table 3.4: TC1. CPU times required by the ROM for the different subcases.

**Test case 2. Entering of a 1D Gaussian profile from the left domain.**

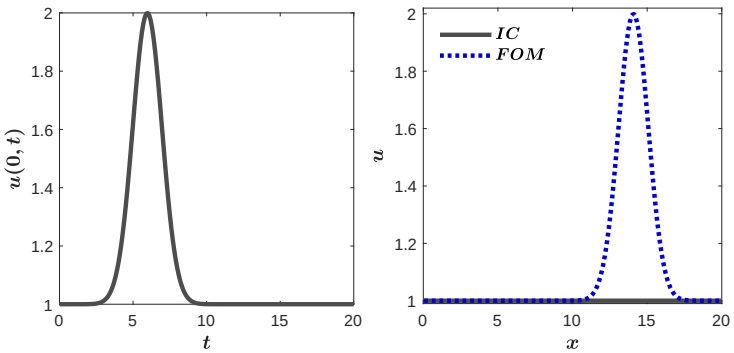
This test case has been designed to study how ROMs solve Dirichlet BCs. The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, 20]$. A 1D Gaussian profile enters the spatial domain from the left boundary

$$u(0, t) = 1 + e^{-0.5(t-6)^2}, \tag{3.13}$$

over a constant level

$$u(x, 0) = 1, \forall x; \tag{3.14}$$

The Gaussian profile is transported along the $x-$axis with constant speed $a = 1$ and no diffusion, $b = 0$. Free boundary condition is considered on the right boundary. The time-dependent BC is shown in Figure 3.10a. All the settings are shown in Table 3.5.




(a) Boundary condition. (b) FOM solution.

Figure 3.10: TC2. Problem settings.

| $L$ | $T$ | $a$ | $\nu$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 20 | 20 | 1 | 0 | Eq. (3.14) | Eq. (3.13) + free | 200 | 1 | 201 | Tab. 3.6 |

Table 3.5: TC2. Problem settings.

*Resolution of the FOM*

No analytical solution has been included in this case. The spatial domain is discretized using $I_x = 200$ volume cells and the time steps are dynamically computed according to the CFL condition (2.5), with CFL = 1. Figure 3.10b shows the solution computed by the FOU-based FOM (2.7) at the final time $T$.

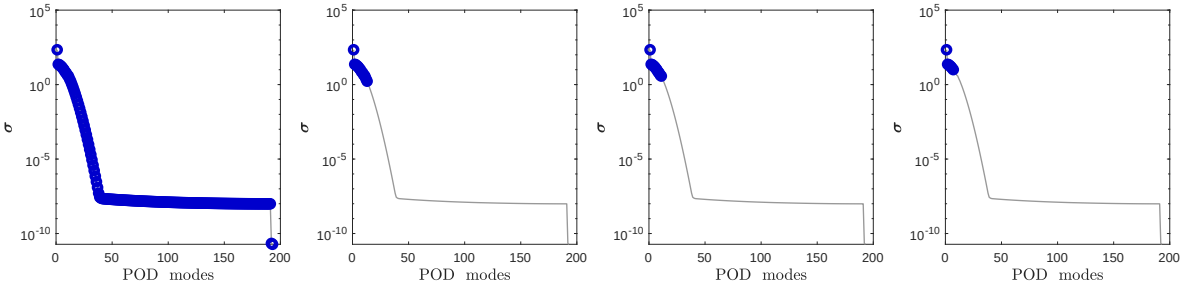
*Definition of the reduced space*

Table 3.6 shows the number of POD modes necessary to recover different percentages of information $P(M_{\text{POD}})$. As in TC1, notice that the 100% of the information is recovered using a non-maximum POD modes.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P(M_{\text{POD}})$ | 100 | 99.9962 | 99.9784 | 99.7295 |
| $M_{\text{POD}}$ | 193 | 13 | 11 | 7 |

Table 3.6: TC2. Number of POD modes necessary to recover different percentages of information.

The POD eigenvalues chosen for each percentage are plotted in Figure 3.11 with blue circles on a grey line that represents the total. Figure 3.12 shows the first 10 functions of the POD basis.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.11: TC2. POD eigenvalues.

*Resolution of the ROM and efficiency test*

Subcase 1 proves that the ROM is working correctly, since the use of the maximum number of POD modes allows it to reach machine accuracy, as shown in Figures 3.13a and 3.13e. The solutions of all subcases are shown in Figure 3.14. Table 3.7 shows the all the errors at the final time of all subcases.

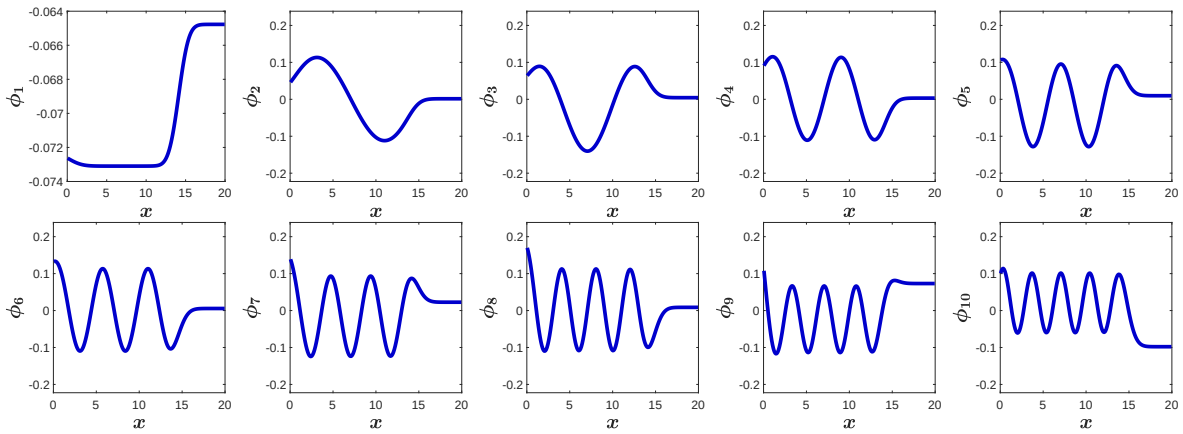
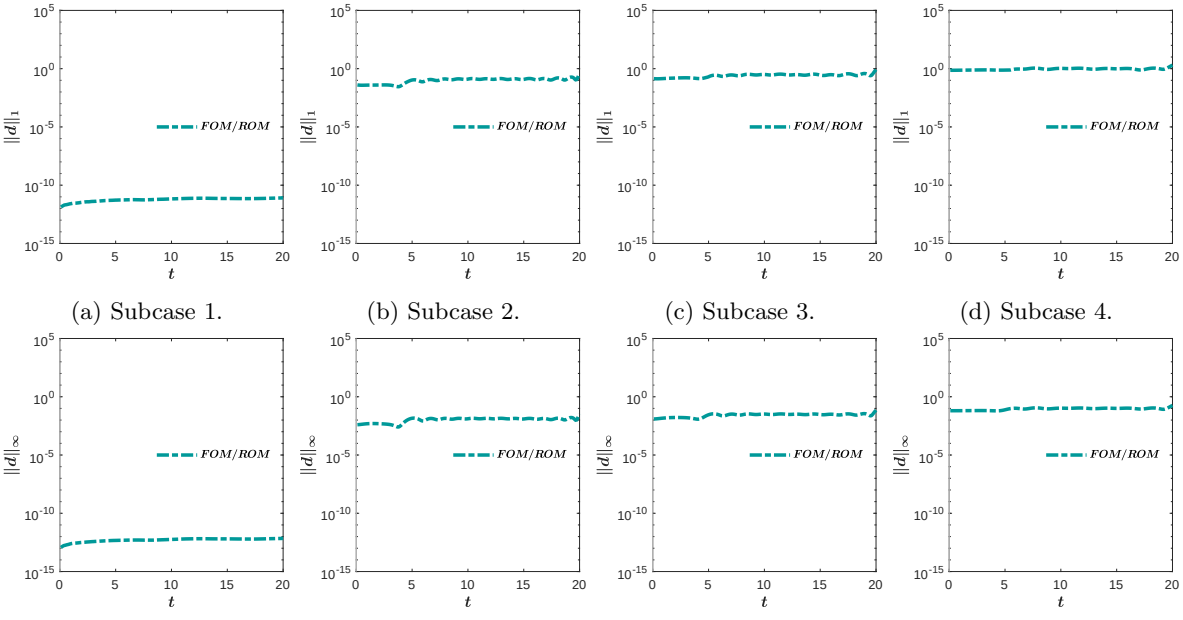



Figure 3.12: TC2. POD eigenvectors.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $8.12 \cdot 10^{-12}$ | $3.72 \cdot 10^{-1}$ | $8.21 \cdot 10^{-1}$ | 1.99 |
| $\|d\|_\infty$ | $6.97 \cdot 10^{-13}$ | $3.21 \cdot 10^{-2}$ | $7.05 \cdot 10^{-2}$ | $1.79 \cdot 10^{-1}$ |

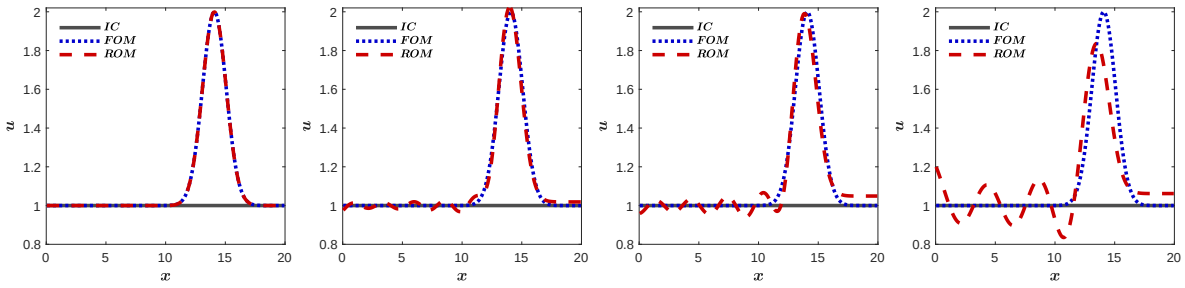
Table 3.7: TC2. Errors made by each subcase.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

(e) Subcase 1. (f) Subcase 2. (g) Subcase 3. (h) Subcase 4.

Figure 3.13: TC2. Differences measured with $\|d\|_1$ (top row) and $\|d\|_\infty$ (bottom row).




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.14: TC1. ROM solutions.

The CPU time required by the FOM to complete this computation is $t^{\text{CPU}} = 1.08 \cdot 10^{-3}$. Table 3.8 shows that Subcase 1, as expected, is much slower than the FOM. In addition to this, the ROM becomes slower when considering Dirichlet BCs, when compared with the speed-ups achieved in Table 3.4. However, 1D advection-based problems are not big enough to assess the improvement in computational efficiency enabled by ROMs. Next test case helps to elucidate this.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| CPU time (s) | $3.34 \cdot 10^{-2}$ | $1.73 \cdot 10^{-4}$ | $1.28 \cdot 10^{-4}$ | $5.7 \cdot 10^{-5}$ |
| Speed-up | $\times 0$ | $\times 6$ | $\times 8$ | $\times 19$ |

Table 3.8: TC2. CPU times required by the ROM for different values of $M_{\text{POD}}$.

**Test case 3. 2D advection of an initial Gaussian profile**

This 2D case has been designed to assess the improvement of computational efficiency achieved by ROMs when solving bigger problems. An initial 2D Gaussian profile is transported with constant speed $\mathbf{a} = (a_x, a_y) = (1, 1)$.

The time-space domain of the case is defined as $(x, y, t) \in [0, 40] \times [0, 40] \times [0, 50]$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-(x-8)^2-(y-8)^2}, \tag{3.15}$$

Free boundary conditions are imposed. The CFL number is set to 0.5. 2D ROMs need smaller values of the CFL number, because they are actually developed from finite differences numerical schemes, which are more unstable.

*Resolution of the FOM*

The spatial domain is discretized using a structured mesh divided into $I_x \times I_y = 100 \times 100$ rectangular volume cells, which can be seen in Figure 3.15. The time evolution of the FOM solutions is shown in Figure 3.16 for different time steps. The initial 2D Gaussian profile advances along the $x$- and $y$-axis and, at $t = 35$ has almost left the domain at the north-east corner.

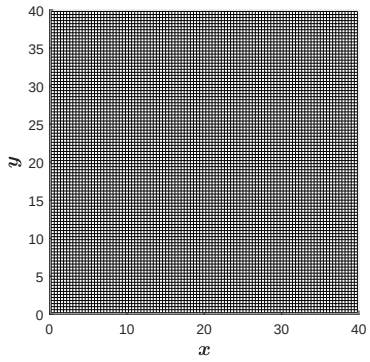
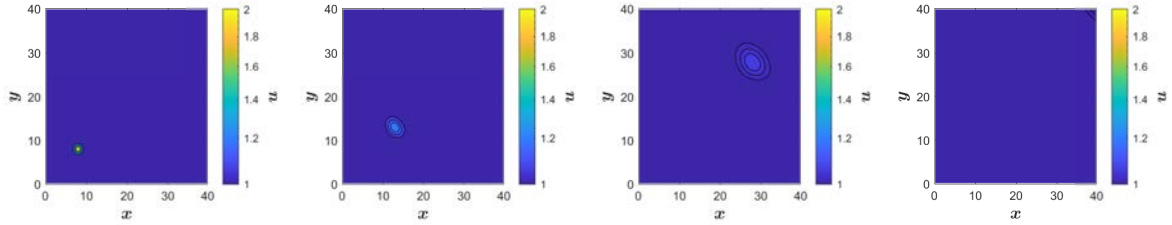



Figure 3.15: TC3. Numerical mesh.




(a) $t = 0$. (b) $t = 4.9$. (c) $t = 20$. (d) $t = 35$.

Figure 3.16: TC3. FOM solutions at different time instants.

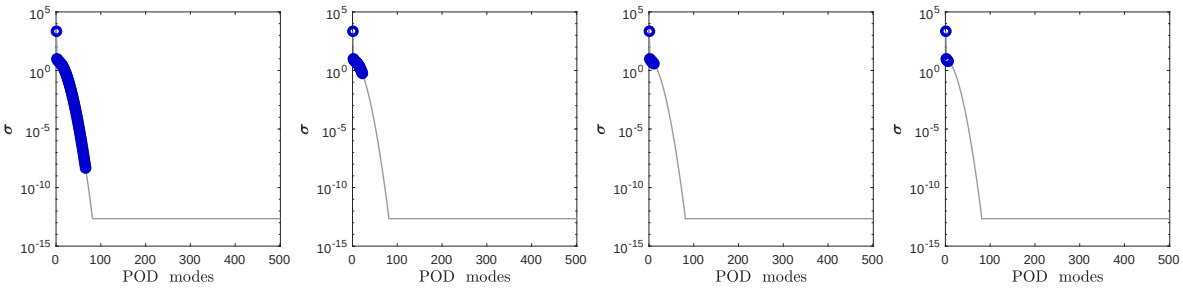
*Definition of the reduced space*

In 2D problems, the number of POD modes determined by Eq. (3.1) decreases faster than in 1D problems, as seen in Figure 3.17. This could mean that the a-posterior error measured with respect to the FOM solutions would worsen, as fewer POD modes are solved. For this reason, in this case higher percentages have been considered, as shown in 3.9. And, again, the 100% of the information is recovered using a non-maximum POD modes.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P(M_{\text{POD}})$ | 100 | 99.999990 | 99.999098 | 99.996850 |
| $M_{\text{POD}}$ | 66 | 22 | 11 | 6 |

Table 3.9: TC3. Number of POD modes necessary to recover different percentages of information.

The POD eigenvalues chosen for each percentage are plotted in Figure 3.17 with blue circles on a grey line that represents the total. The flat region from around mode 100 to the end represents the negligible modes that whose magnitude is smaller than the machine accuracy. The first eight eigenvectors can be seen in Figure 3.18.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.17: TC3. POD eigenvalues.

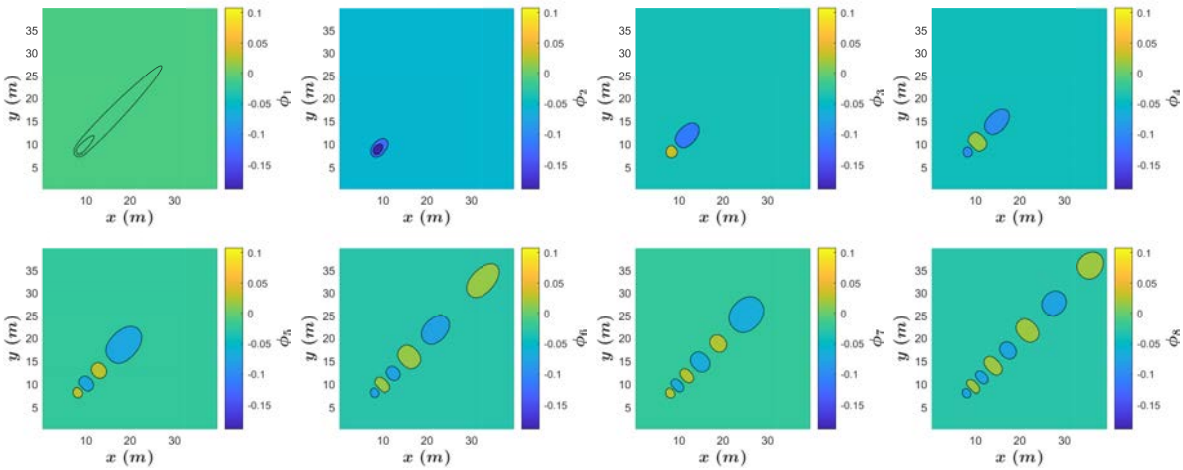



Figure 3.18: TC3. POD eigenvectors.

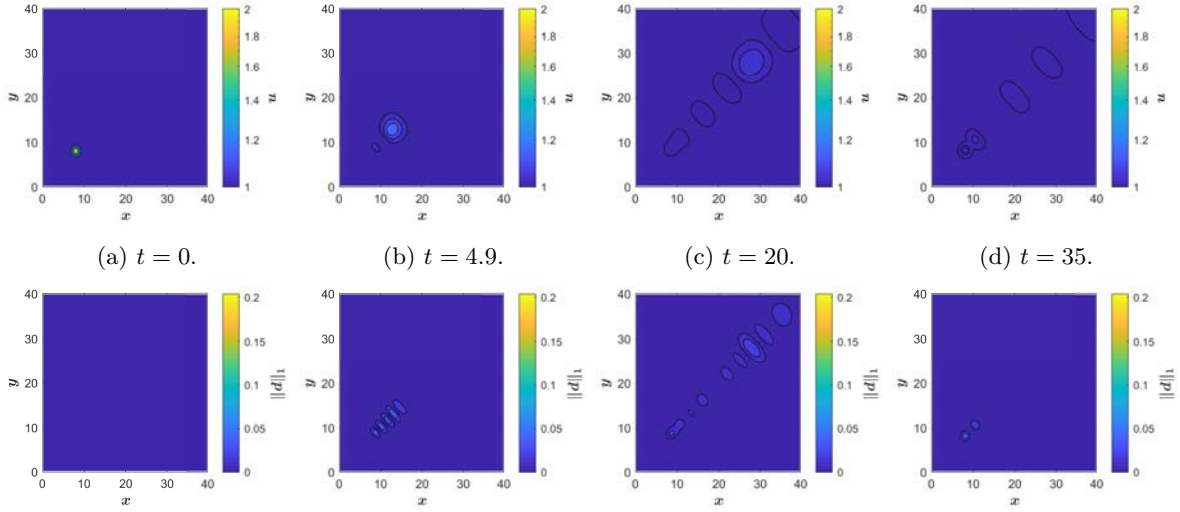
*Resolution of the ROM and efficiency test*

Figure 3.19 shows the solutions computed by the ROM with 11 POD modes (Subcase 3) in the first row and the differences with respect to the FOM in the second row. The error made by Subcase 1 remains around the machine accuracy and those of the rest of the subcases go up to orders -2 and -1, as shown in Figure 3.20.

Table 3.10 shows the all the errors at the final time of all subcases and their CPU times required. The CPU time required by FOM to compute the training solution is $2.28 \cdot 10^{-1}$. In terms of efficiency, even Subcase 1 is faster than the FOM. However, the rest of the subcases achieve 2 and 3 orders of magnitude with respect to the FOM.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $3.20 \cdot 10^{-8}$ | $1.15 \cdot 10^{-1}$ | 2.24 | 2.49 |
| $\|d\|_\infty$ | $2.43 \cdot 10^{-11}$ | $2.28 \cdot 10^{-3}$ | $1.07 \cdot 10^{-2}$ | $1.67 \cdot 10^{-2}$ |
| CPU time (s) | $7.35 \cdot 10^{-3}$ | $9.00 \cdot 10^{-4}$ | $2.00 \cdot 10^{-4}$ | $5.00 \cdot 10^{-5}$ |
| Speed-up | $\times 31$ | $\times 254$ | $\times 1142$ | $\times 4567$ |

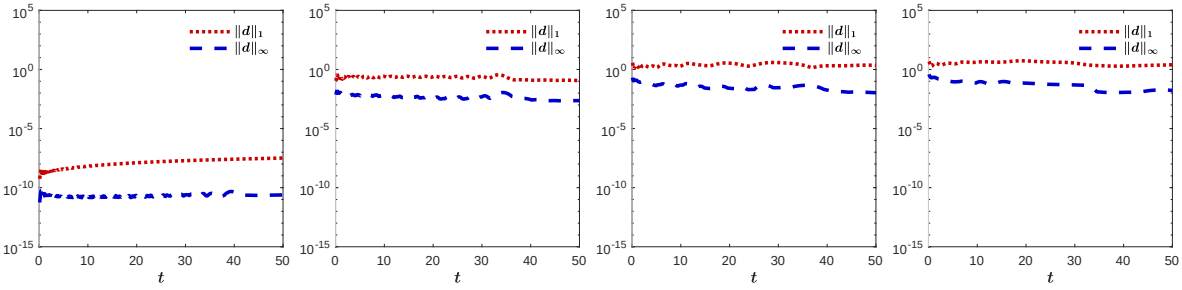
Table 3.10: TC3. CPU times required by the ROM for different values of $M_{\text{POD}}$.



(a) $t = 0$. (b) $t = 4.9$. (c) $t = 20$. (d) $t = 35$.

(e) $t = 0$. (f) $t = 4.9$. (g) $t = 20$. (h) $t = 35$.

Figure 3.19: TC3. ROM solutions and differences of Subcase 3.



(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 3.20: TC3. Differences measured for the different subcases.

## 3.2 ROMs and high-order schemes

High-order schemes increase the accuracy of solutions at the cost of increased computational expense. It is highly interesting to study whether it is possible to train ROMs developed from high-order FOMs with low-order solutions. If in this way the high order in the solutions obtained by the ROM could be maintained, computational costs could be saved, even in training.

To this end, a combination of training and development of low-order and high-order ROMs is proposed in Test case 4. The numerical schemes proposed are the already used first-order FOU, to which is added another first-order scheme based on the Lax-Friedrichs method and a second-order scheme based on the Lax-Wendroff method. These two new schemes are explained below.

### 3.2.1 Lax-Friedrichs-based ROM

The Lax-Friedrichs-based ROM with periodic BCs reads

$$\hat{u}_p^{n+1} = \hat{u}_p^n + \frac{\Delta t}{\Delta x} A_{pk} \hat{u}_k^n + B_{pk} \hat{u}_k^n + \frac{\Delta t}{\Delta x^2} C_{pk} \hat{u}_k^n, \tag{3.16}$$

where matrices are

$$\begin{aligned}
A_{pk} &= \sum_{i \in J^S} a_i \phi_{i,p} - \frac{1}{2} a \sum_{i \in J^I} \left( \phi_{i+1,k} - \phi_{i-1,k} \right) \phi_{i,p}, \\
B_{pk} &= \sum_{i \in J^S} b_i \phi_{i,p} + \frac{1}{2} \xi \sum_{i \in J^I} \left( \phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k} \right) \phi_{i,p}, \\
C_{pk} &= \sum_{i \in J^S} c_i \phi_{i,p} + \nu \sum_{i \in J^I} \left( \phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k} \right) \phi_{i,p}.
\end{aligned}$$

where the coefficients $a_i, b_i, c_i$ are given by the BCs considered, similarly to (3.9).

### 3.2.2 Lax-Wendroff-based ROM

The Lax-Wendroff-based ROM with periodic BCs reads

$$\hat{u}_p^{n+1} = \hat{u}_p^n + \frac{\Delta t}{\Delta x} A_{pk} \hat{u}_k^n + \left( \frac{\Delta t}{\Delta x} \right)^2 B_{pk} \hat{u}_p^n + \frac{\Delta t}{\Delta x^2} C_{pk} \hat{u}_p^n, \tag{3.17}$$

where matrices are

$$A_{pk} = \sum_{i \in J^S} a_i \phi_{i,p} - \frac{1}{2} a \sum_{i \in J^I} \left(\phi_{i+1,k} - \phi_{i-1,k}\right) \phi_{i,p},$$
$$B_{pk} = \sum_{i \in J^S} b_i \phi_{i,p} + \frac{1}{2} a^2 \sum_{i \in J^I} \left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right) \phi_{i,p},$$
$$C_{pk} = \sum_{i \in J^S} c_i \phi_{i,p} + \nu \sum_{i \in J^I} \left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right) \phi_{i,p}.$$

where the coefficients $a_i, b_i, c_i$ are given by the BCs considered, similarly to (3.9).

### 3.2.3 Test case 4. Analysis of the convergence

This test case has been designed to test the order of convergence obtained when the ROM is trained and developed from numerical schemes of different order of accuracy. The 1D advection-diffusion (2.1) is considered.

Nine different subcases have been designed to cover the $3^2 = 9$ combinations between the three numerical schemes considered, i.e., first-order upwind FOU, Lax-Friedrichs (LF), both first-order schemes, and Lax-Wendroff (LW), a second-order scheme. Table 3.11 shows these combinations for all subcases, where the FOM indicates which numerical scheme has been used to compute the training snapshots and the ROM indicates from which FOM the ROM has been developed.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| FOM | FOU | FOU | FOU | LF | LF | LF | LW | LW | LW |
| ROM | FOU | LF | LW | FOU | LF | LW | FOU | LF | LW |

Table 3.11: TC4. Subcases

The time-space domain of the case is defined as $(x, t) \in [0, 2] \times [0, 10]$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-10(x-1)^2}, \tag{3.18}$$

Periodic boundary conditions are considered, so that the initial Gaussian profile exits the domain through the right boundary and enters it again from the left boundary 5 times in a row. This way, the error can be properly measured.

The spatial domain has been discretized using different numbers of volume cells

$$I_x \in \{50, 100, 200, 400, 800, 1600\}. \tag{3.19}$$

The rest of the settings are the same for all subcases, as shown in Table 3.12. The time step is dynamically computed according to (2.5). The ROM solutions have been computed using 10 POD modes.

| $L$ | $T$ | $a$ | $\nu$ | IC | BCs | $I_x$ | CFL | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|
| 2 | 10 | 1 | 0 | Eq. (3.18) | Periodic | Eq. (3.19) | 0.4 | 10 |

Table 3.12: TC4. Problem settings.

Table 3.13 shows the errors and the orders of convergence obtained by means of the double mesh method of the three FOM solutions [128].

The ROM solutions are shown in Figure 3.21. Tables 3.14 and 3.15 show the errors and the orders of convergence obtained of the ROM solutions, respectively. As it can be seen, even though subcases 3 and 6 are trained with first-order solutions, their Lax-Wendroff-based ROM solutions converge to order 2, as highlighted in bold characters in Table 3.15.

| | $I_x$ | $\|d\|_1$ | $\sigma_1$ | $\|d\|_\infty$ | $\sigma_\infty$ |
|---|---|---|---|---|---|
| | 50/100 | 0.1395 | - | 0.1258 | - |
| | 100/200 | 0.1175 | 0.2476 | 0.1317 | -0.0656 |
| FOU | 200/400 | 0.0862 | 0.4470 | 0.1163 | 0.1791 |
| | 400/800 | 0.0563 | 0.6150 | 0.0865 | 0.4277 |
| | 800/1600 | 0.0332 | 0.7592 | 0.0554 | 0.6413 |
| | 50/100 | 0.0612 | - | 0.0481 | - |
| | 100/200 | 0.1278 | -1.0616 | 0.1035 | -1.1058 |
| LF | 200/400 | 0.1372 | -0.1031 | 0.1291 | -0.3188 |
| | 400/800 | 0.1117 | 0.2966 | 0.1304 | -0.0149 |
| | 800/1600 | 0.0801 | 0.4799 | 0.1114 | 0.2281 |
| | 50/100 | 0.1205 | - | 0.1281 | - |
| | 100/200 | 0.0314 | 1.9378 | 0.0458 | 1.4837 |
| LW | 200/400 | 0.0079 | 1.9877 | 0.0126 | 1.8629 |
| | 400/800 | 0.0020 | 1.9982 | 0.0032 | 1.9697 |
| | 800/1600 | 0.0005 | 1.9988 | 0.0008 | 1.9931 |

Table 3.13: TC4. Errors and orders of convergence of FOM using the double mesh method.

| | $I_x$ | 50/100 | 100/200 | 200/400 | 400/800 | 800/1600 |
|---|---|---|---|---|---|---|
| | S1 | $1.39{\cdot}10^{-1}$ | $1.17{\cdot}10^{-1}$ | $8.62{\cdot}10^{-2}$ | $5.63{\cdot}10^{-2}$ | $3.33{\cdot}10^{-2}$ |
| | S2 | $6.12{\cdot}10^{-2}$ | $1.28{\cdot}10^{-1}$ | $1.37{\cdot}10^{-1}$ | $1.12{\cdot}10^{-1}$ | $8.01{\cdot}10^{-2}$ |
| | S3 | $1.20{\cdot}10^{-1}$ | $3.16{\cdot}10^{-2}$ | $8.05{\cdot}10^{-3}$ | $2.02{\cdot}10^{-3}$ | $5.71{\cdot}10^{-4}$ |
| | S4 | $1.39{\cdot}10^{-1}$ | $1.17{\cdot}10^{-1}$ | $8.62{\cdot}10^{-2}$ | $5.63{\cdot}10^{-2}$ | $3.33{\cdot}10^{-2}$ |
| $\|d\|_1$ | S5 | $6.12{\cdot}10^{-2}$ | $1.28{\cdot}10^{-1}$ | $1.37{\cdot}10^{-1}$ | $1.12{\cdot}10^{-1}$ | $8.01{\cdot}10^{-2}$ |
| | S6 | $1.19{\cdot}10^{-1}$ | $3.15{\cdot}10^{-2}$ | $8.18{\cdot}10^{-3}$ | $2.17{\cdot}10^{-3}$ | $5.54{\cdot}10^{-4}$ |
| | S7 | $1.39{\cdot}10^{-1}$ | $1.17{\cdot}10^{-1}$ | $8.62{\cdot}10^{-2}$ | $5.63{\cdot}10^{-2}$ | $3.33{\cdot}10^{-2}$ |
| | S8 | $6.12{\cdot}10^{-2}$ | $1.28{\cdot}10^{-1}$ | $1.37{\cdot}10^{-1}$ | $1.12{\cdot}10^{-1}$ | $8.01{\cdot}10^{-2}$ |
| | S9 | $1.20{\cdot}10^{-1}$ | $3.16{\cdot}10^{-2}$ | $8.02{\cdot}10^{-3}$ | $2.00{\cdot}10^{-3}$ | $4.95{\cdot}10^{-4}$ |
| | S1 | $1.26{\cdot}10^{-1}$ | $1.32{\cdot}10^{-1}$ | $1.16{\cdot}10^{-1}$ | $8.63{\cdot}10^{-2}$ | $5.52{\cdot}10^{-2}$ |
| | S2 | $4.81{\cdot}10^{-2}$ | $1.03{\cdot}10^{-1}$ | $1.29{\cdot}10^{-1}$ | $1.30{\cdot}10^{-1}$ | $1.11{\cdot}10^{-1}$ |
| | S3 | $1.32{\cdot}10^{-1}$ | $4.69{\cdot}10^{-2}$ | $1.24{\cdot}10^{-2}$ | $3.06{\cdot}10^{-3}$ | $7.06{\cdot}10^{-4}$ |
| | S4 | $1.26{\cdot}10^{-1}$ | $1.32{\cdot}10^{-1}$ | $1.16{\cdot}10^{-1}$ | $8.62{\cdot}10^{-2}$ | $5.51{\cdot}10^{-2}$ |
| $\|d\|_\infty$ | S5 | $4.81{\cdot}10^{-2}$ | $1.03{\cdot}10^{-1}$ | $1.29{\cdot}10^{-1}$ | $1.30{\cdot}10^{-1}$ | $1.11{\cdot}10^{-1}$ |
| | S6 | $1.36{\cdot}10^{-1}$ | $4.78{\cdot}10^{-2}$ | $1.22{\cdot}10^{-2}$ | $3.07{\cdot}10^{-3}$ | $8.02{\cdot}10^{-4}$ |
| | S7 | $1.26{\cdot}10^{-1}$ | $1.32{\cdot}10^{-1}$ | $1.16{\cdot}10^{-1}$ | $8.63{\cdot}10^{-2}$ | $5.51{\cdot}10^{-2}$ |
| | S8 | $4.81{\cdot}10^{-2}$ | $1.04{\cdot}10^{-1}$ | $1.29{\cdot}10^{-1}$ | $1.30{\cdot}10^{-1}$ | $1.11{\cdot}10^{-1}$ |
| | S9 | $1.29{\cdot}10^{-1}$ | $4.62{\cdot}10^{-2}$ | $1.25{\cdot}10^{-2}$ | $3.17{\cdot}10^{-3}$ | $8.08{\cdot}10^{-4}$ |

Table 3.14: TC4. Errors of ROM using the double mesh method.

| | $I_x$ | 50/200 | 100/400 | 200/800 | 400/1600 |
|---|---|---|---|---|---|
| | S1 | 0.2475 | 0.4467 | 0.6142 | 0.7587 |
| | S2 | -1.0616 | -0.1032 | 0.2965 | 0.4797 |
| | **S3** | **1.9234** | **1.9749** | **1.9917** | **1.8267** |
| | S4 | 0.2477 | 0.4467 | 0.6136 | 0.7583 |
| $\sigma_1$ | S5 | -1.0616 | -0.1031 | 0.2964 | 0.4795 |
| | **S6** | **1.9230** | **1.9444** | **1.9146** | **1.9701** |
| | S7 | 0.2477 | 0.4468 | 0.6142 | 0.7578 |
| | S8 | -1.0619 | -0.1031 | 0.2966 | 0.4797 |
| | **S9** | **1.9304** | **1.9778** | **2.0006** | **2.0167** |
| | S1 | 0.2475 | 0.4467 | 0.6142 | 0.7587 |
| | S2 | -1.0616 | -0.1032 | 0.2965 | 0.4797 |
| | **S3** | **1.9234** | **1.9749** | **1.9917** | **1.8267** |
| | S4 | 0.2477 | 0.4467 | 0.6136 | 0.7583 |
| $\sigma_\infty$ | S5 | -1.0616 | -0.1031 | 0.2964 | 0.4795 |
| | **S6** | **1.9230** | **1.9444** | **1.9146** | **1.9701** |
| | S7 | 0.2477 | 0.4468 | 0.6142 | 0.7578 |
| | S8 | -1.0619 | -0.1031 | 0.2966 | 0.4797 |
| | **S9** | **1.9304** | **1.9778** | **2.0006** | **2.0167** |

Table 3.15: TC4. Orders of convergence of ROM using the double mesh method.

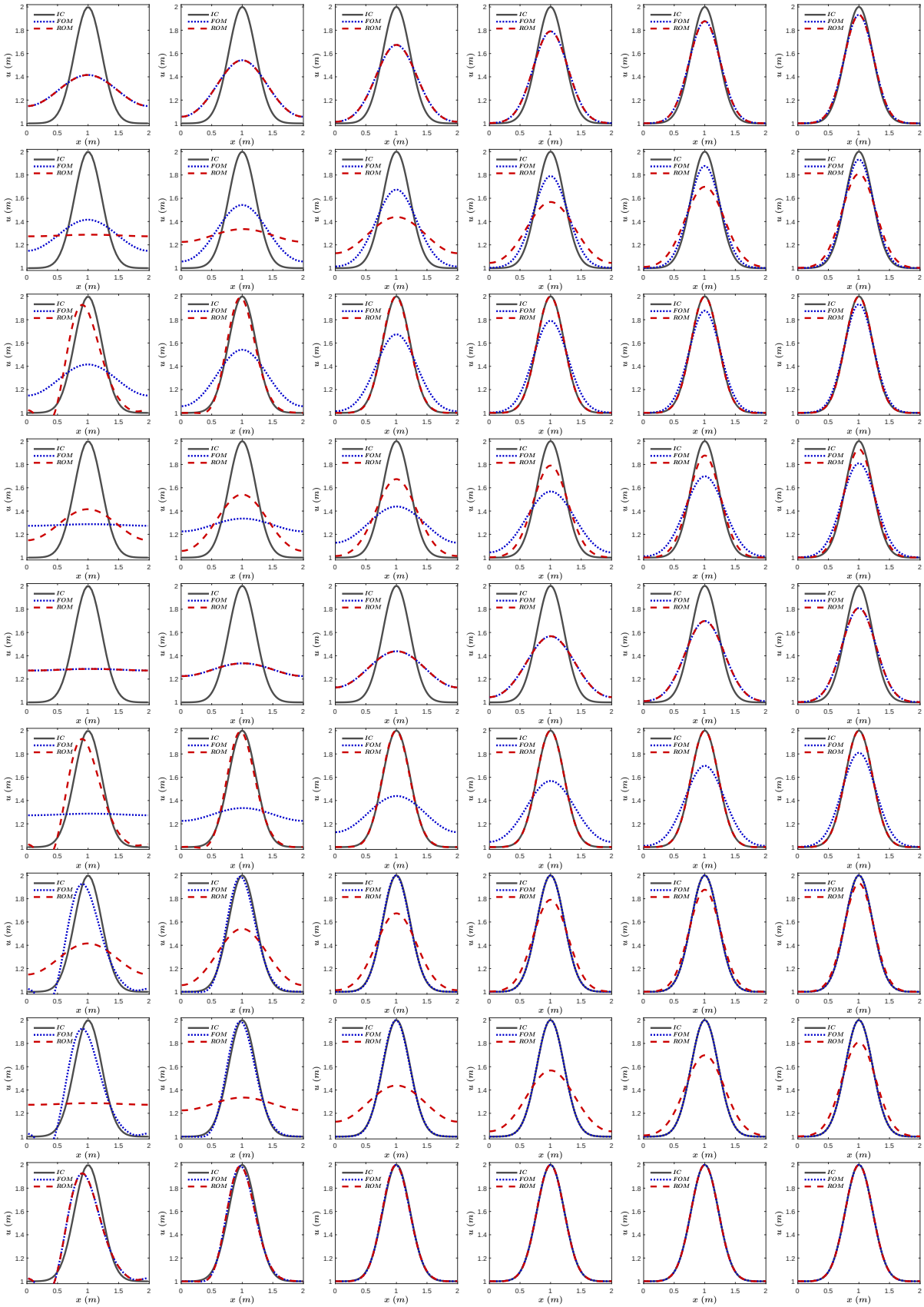



Figure 3.21: TC4. Solutions computed with the ROM and its comparison with the FOM. Each row corresponds to a subcase.

## 3.3 ROMs and parametrized linear problems

All the problems listed in Chapter 2 can be considered parametrized PDEs, because they depend on input parameters [149], for example

i) the constants that define the ICs,

ii) or the BCs,

iii) the advection velocity,

iv) the diffusion coefficient,

v) the Manning coefficient,

vi) the bed topography.

Evaluating the behaviour of these problems for different values of their component parameters can be very expensive using FOMs. Therefore, it is highly interesting to use the computational speed of ROMs to test different values of the input parameters with respect to those of the training. However, the definition of the reduced space from solutions obtained with a specific value of the parameters establishes a strong connection between the two. Thus, the ROM may be confined to these values and not be able to compute valuable solutions for other values.

In previous works [201, 202], it has been studied to what extent ROMs can reproduce the dynamics of new scenarios obtained after varying the original input parameters. In these works, the proposal starts from a single training sample calculated by the FOM for specific values of the input parameters. With this, it is tested how far the ROM can go in solving for different values of the input parameters from those of the training, the target values.

The techineque used in this thesis is based on generating a multiple training sample composed of $M_{\text{train}}$ samples and obtained from random values of the input parameters (training values), as it is done in [14, 13, 62, 177]. Once it has been trained, the ROM is solved for some values of interest of the input parameters (target values). Part of the study focuses on finding out how large the training sample has to be for the ROM to be sufficiently fed; if a minimum number of samples is necessary or even if only one sample is needed; and to check what is the relationship between the ROM configuration parameters and the number of training sample.

The test cases solved below are designed to evaluate the prediction of ROM with all possible input parameters when considering the 1D advection-diffusion equation (2.1). They are the following

- Test case 5 (TC5). The advection velocity and the diffusion coefficient are considered as the input parameters.
- Test case 6 (TC6). The constants that define the initial Gaussian profile are considered the input parameters.

- Test case 7 (TC7). The constants that define the initial discontinuity are considered the input parameters.
- Test case 8 (TC8). The constants that define the Gaussian profile that enters the domain from the left boundary are considered the input parameters.

### 3.3.1 Modification of the standard ROM strategy

The standard ROM strategy explained in 3.1.1 must be modified to cluster all the training samples computed with the FOM.

The input parameters used to train the ROM, called *training parameters*, are denoted throughout the thesis by

$$\boldsymbol{\mu}^m = (\mu_1^m, ..., \mu_P^m)\,,$$

with $m = 1, ..., M_{\text{train}}$, where $M_{\text{train}}$ is the total number of samples computed by the the FOM; and where $P$ is the number of parameters of interest. For example, if the 1D advection-diffusion equation is to be tested, both the advection velocity and the diffusion coefficient can be considered the parameters of interest $\boldsymbol{\mu} = (\mu_1, \mu_2) = (\nu, a)$.

For each solution that is calculated with the FOM for a set of training parameters, also called *training sample*, a snapshot matrix is generated $\mathbf{U}\,(\boldsymbol{\mu}^m)$.

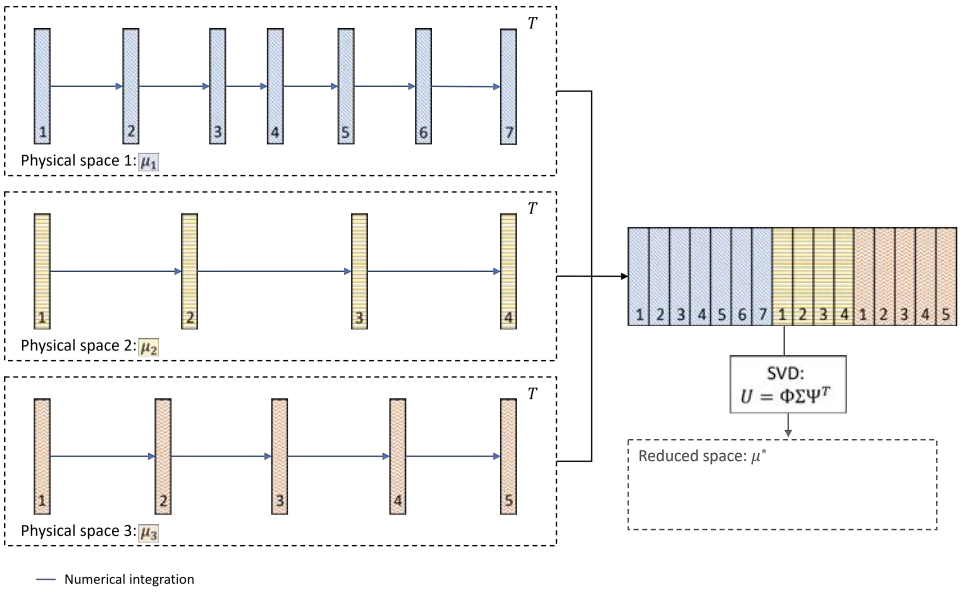



Figure 3.22: ROM strategy when solving parametrized linear problems.

All the training sub-matrices are assembled together into one single snapshot matrix, snapshot sub-matrices into one single snapshot matrix, as indicated in Figure 3.22. This can be done by easily placing one sub-matrix after another

$$\mathbf{U} = \left( \begin{array}{cccc} \mathbf{U}\left(\boldsymbol{\mu}^1\right) & U\left(\boldsymbol{\mu}^2\right) & \ldots & \mathbf{U}\left(\boldsymbol{\mu}^{M_{\text{train}}}\right) \end{array} \right),$$

where $M$ is number of snapshots sub-matrices of the training set. The SVD is applied without special treatment to obtain the reduced basis and obtains the basis functions that define the reduced space for a general value of the parameters

$$\boldsymbol{\mu}^* = (\mu_1^*, ..., \mu_P^*)\,.$$

### 3.3.2 Test case 5. Input parameters: advection velocity and diffusion coefficient

The parameters studied in this case are the advection velocity and the diffusion coefficients

$$\mu_1 = \nu, \quad \mu_2 = a.$$

The value of these parameters determines the ROM's ability to predict more easily, as will be shown in the examples. For this purpose, the first subcases are proposed to show that depending on whether diffusion or advection dominates the ROM needs less or more training sample. The rest of them deal with intermediate scenarios.

The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, T]$, where $T$ depends on the subcase, indicated in Table 3.16 with the rest of the settings. The ICs are defined differently for each of them. Free boundary conditions are considered. The physical domain is discretized using $I_x = 200$ volume cells and CFL = 0.9. The advection velocity and diffusion coefficient included are the target ones.

| S. | $T$ | $\nu^{\text{train}}$ | $a^{\text{train}}$ | $\nu^*$ | $a^*$ | IC | $N_T$ | $M_{\text{POD}}$ | $M_{\text{train}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 0.35 | 0 | 0.88 | 0 | Eq. (3.20) | 7807 | 5 | 1 |
| 2 | 10 | 0 | Fig. 3.24a | 0 | 0.27 | Eq. (3.21) | 85 | 5 | 1 |
| 3 | 10 | 0 | Fig. 3.24b | 0 | 0.27 | Eq. (3.21) | 85 | 5 | 1 |
| 4 | 10 | 0 | Fig. 3.24c | 0 | 0.27 | Eq. (3.21) | 85 | 5 | 2 |
| 5 | 10 | 0 | Fig. 3.24d | 0 | 0.27 | Eq. (3.21) | 85 | 5 | 3 |
| 6 | 10 | Fig. 3.25a | Fig. 3.25a | 0.85 | 0.48 | Eq. (3.21) | 2178 | 5 | 2 |
| 7 | 10 | Fig. 3.25b | Fig. 3.25b | 0.85 | 0.48 | Eq. (3.21) | 2178 | 10 | 4 |
| 8 | 10 | Fig. 3.25c | Fig. 3.25c | 0.85 | 0.48 | Eq. (3.21) | 2178 | 5 | 2 |
| 9 | 10 | Fig. 3.25d | Fig. 3.25d | 0.85 | 0.48 | Eq. (3.21) | 2178 | 10 | 4 |
| $L = 20$, BCs: free, $I_x = 200$, CFL = 0.9 | | | | | | | | | |

Table 3.16: TC5. Problem settings.

#### Diffusion-dominated case

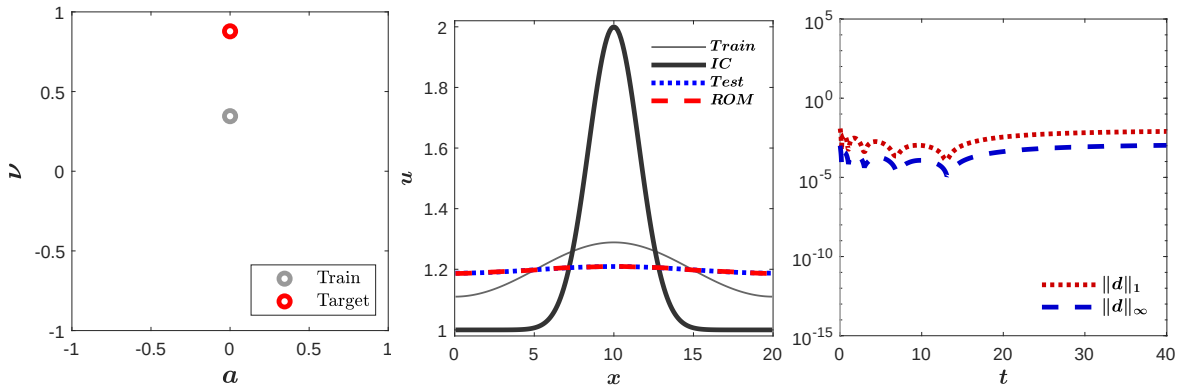
The training sample is computed by the FOM with $\mu_1^{\text{train}} = \nu^{\text{train}} = 0.35$. The target value of the diffusion coefficient is bigger, $\mu_1^* = \nu^* = 0.88$, so that the initial profile is affected by

diffusion. No advection velocity is considered, $\mu_2^{\text{train}} = a^{\text{train}} = \mu_2^* = a^* = 0$. This values are shown in Figure 3.23a. The initial Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-0.2(x-10)^2}. \tag{3.20}$$

The ROM is able to obtain satisfactory results for any value of the diffusion coefficient having been trained on a single sample. This can be seen in Figure 3.23b. In general, parabolic equations allow a low number of training samples.

The ROM solution is computed using only 5 POD modes achieves good accuracy, as it can be seen in Figure 3.23c, and is 37 times faster than the FOM would be if it were to calculate the solution with the same target value of the diffusion coefficient. Its results are contained in Table 3.17, along with those of the rest of the subcases.




(a) S1. Input parameters. (b) S1. ROM solution. (c) S1. Differences.

Figure 3.23: TC5. Results for the diffusion-dominated case.

| Subcase | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\|d\|_1$ | $7.94 \cdot 10^{-3}$ | $1.42 \cdot 10^{-2}$ | 1.38 | 1.45 | $2.45 \cdot 10^{-2}$ |
| $\|d\|_\infty$ | $1.05 \cdot 10^{-3}$ | $1.99 \cdot 10^{-3}$ | $1.84 \cdot 10^{-3}$ | $2.04 \cdot 10^{-3}$ | $4.07 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $4.23 \cdot 10^{-2}$ | $9.28 \cdot 10^{-4}$ | $9.19 \cdot 10^{-4}$ | $9.30 \cdot 10^{-4}$ | $9.26 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $1.13 \cdot 10^{-3}$ | $1.51 \cdot 10^{-5}$ | $1.50 \cdot 10^{-5}$ | $1.54 \cdot 10^{-5}$ | $1.45 \cdot 10^{-5}$ |
| Speed-up | $\times 38$ | $\times 61$ | $\times 61$ | $\times 60$ | $\times 64$ |
| Subcase | 6 | 7 | 8 | 9 | |
| $\|d\|_1$ | $6.63 \cdot 10^{-1}$ | $9.07 \cdot 10^{-2}$ | $3.06 \cdot 10^{-1}$ | $6.13 \cdot 10^{-3}$ | |
| $\|d\|_\infty$ | $7.45 \cdot 10^{-2}$ | $2.02 \cdot 10^{-2}$ | $4.69 \cdot 10^{-2}$ | $1.90 \cdot 10^{-3}$ | |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $1.19 \cdot 10^{-2}$ | $1.21 \cdot 10^{-2}$ | $1.21 \cdot 10^{-2}$ | $1.21 \cdot 10^{-2}$ | |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $3.16 \cdot 10^{-4}$ | $1.12 \cdot 10^{-3}$ | $3.18 \cdot 10^{-4}$ | $1.11 \cdot 10^{-3}$ | |
| Speed-up | $\times 38$ | $\times 11$ | $\times 38$ | $\times 11$ | |

Table 3.17: TC5. Results for all subcases.

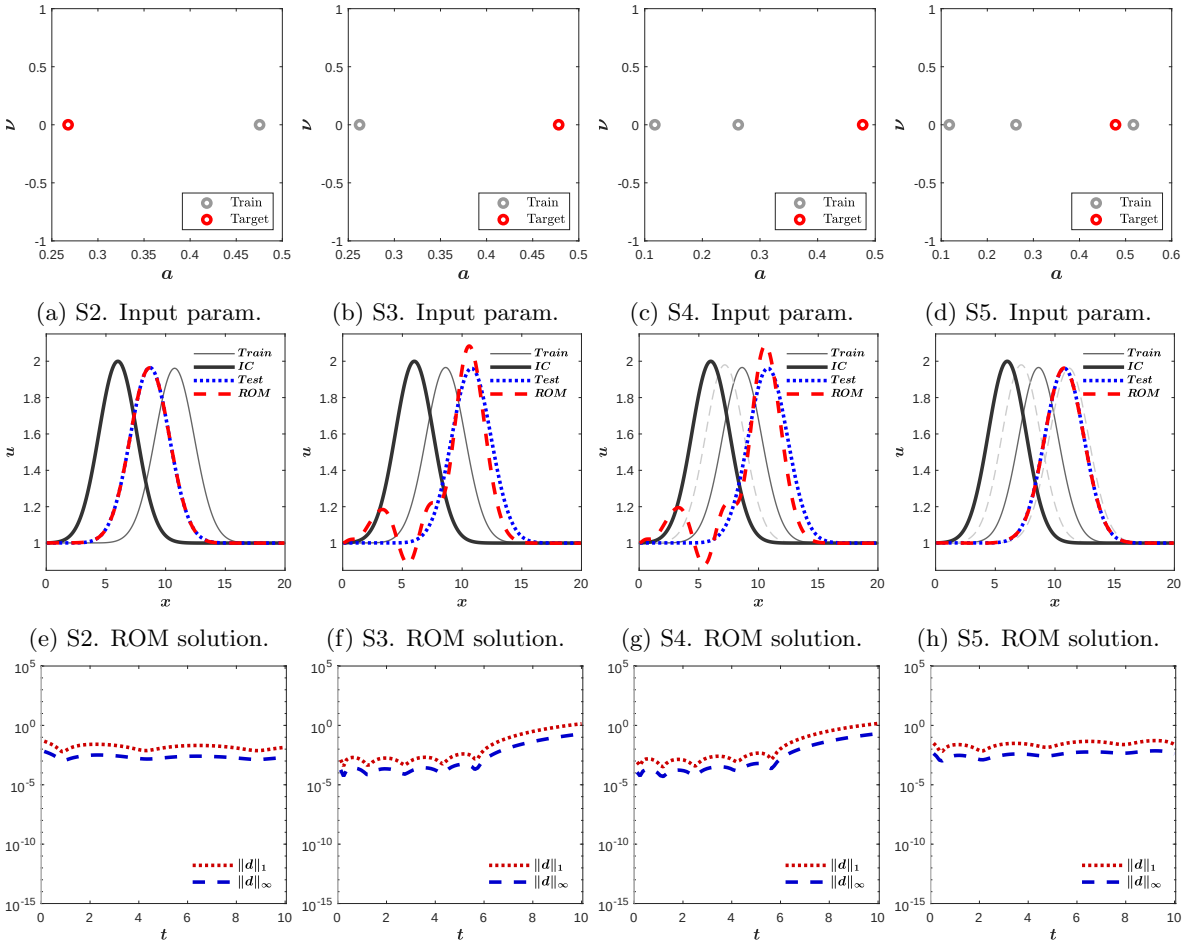
### Advection-dominated cases

Advection-dominated cases need more training. An initial Gaussian profile defined by

$$u(x,0) = 1 + e^{-0.2(x-6)^2}. \tag{3.21}$$

is transported with velocity $a$. Four subcases have been designed in which the advection velocity $a$ is given random values for the training. The number of training samples $M_{\text{train}}$ is progressively increased, keeping the target value of $a$ for ROM resolution fixed. The training and target values of $a$ are plotted in Figures 3.24a-3.24d against the diffusion coefficient ($\nu = 0$).

As shown in Figure 3.24e, if the training velocity is bigger than the target velocity, the ROM is able to properly compute the target solution with just one training sample. But, on the contrary, if the training velocity is smaller than the target velocity, the ROM solution in Figure 3.24f shows a bad performance. If the training set is extended with more samples, the ROM is still unable to perform correctly, as long as the training values of the advection velocity are smaller than the target value, Figure 3.24g. As soon as a new sample is used in the training with a bigger value of the velocity, the ROM shows proper results, as in Figure 3.24h.




(a) S2. Input param. (b) S3. Input param. (c) S4. Input param. (d) S5. Input param.

(e) S2. ROM solution. (f) S3. ROM solution. (g) S4. ROM solution. (h) S5. ROM solution.

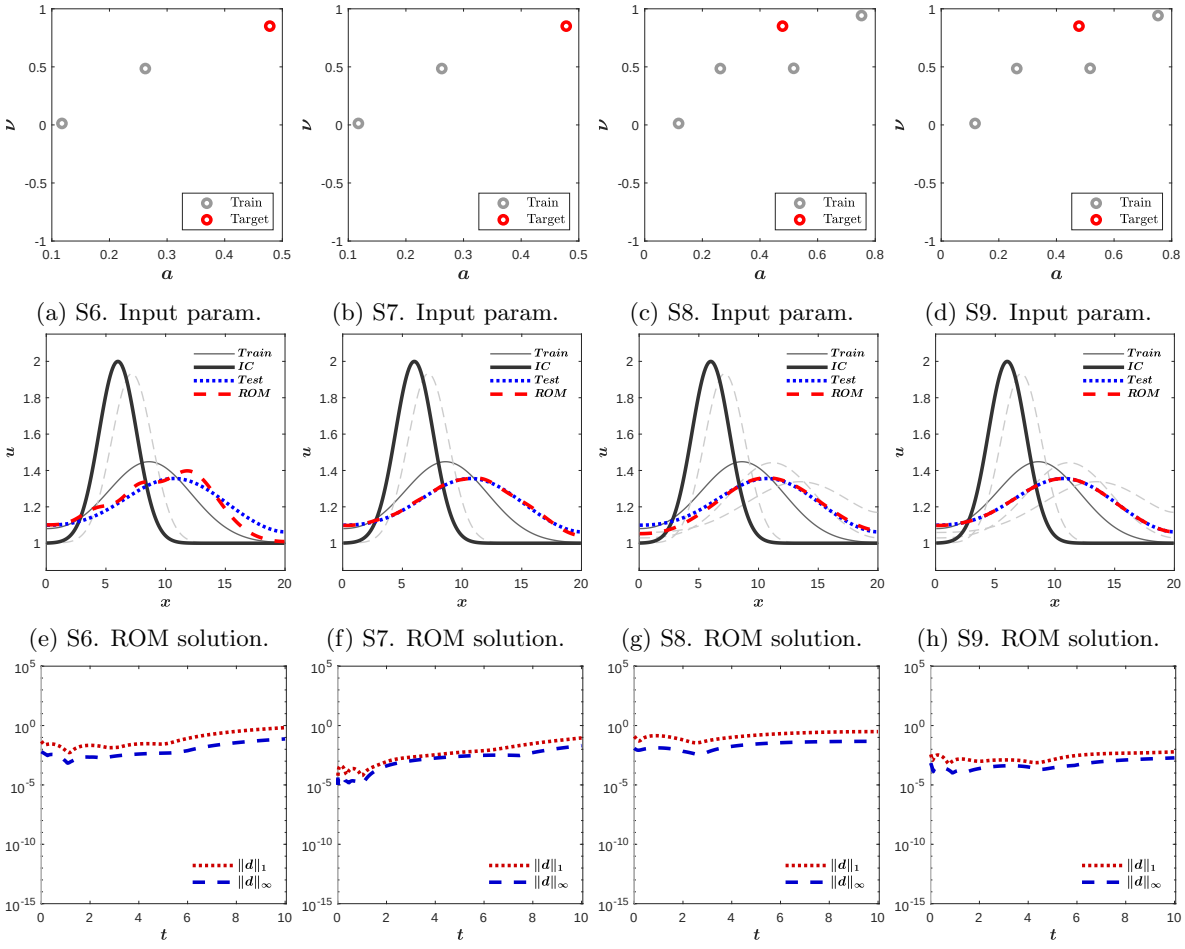
(i) S2. Differences. (j) S3. Differences. (k) S4. Differences. (l) S5. Differences.

Figure 3.24: TC5. Advection-dominated cases. Input paramters, ROM solutions and differences.

### Advection-diffusion cases

Let's solve problems in which both advection and diffusion are considered. These coefficients are given random values for the training phase and the target solution, which can be seen in Figures 3.25a-3.25d. The initial Gaussian profile of the previous case is considered (3.21).

The solutions of the ROM have been computed using 5 (Subcases 6 and 8) and 10 POD modes (Subcases 7 and 9) and with 2 (Subcases 6 and 7) and 4 training samples (Subcases 8 and 9). As it can be seen in Figures 3.25e-3.25h, 2 training samples are not enough to properly train the ROM. With 4 training samples and 10 POD modes, the ROM achieves a good solution in terms of accuracy (Fig. 3.25l), but is only 10 times faster than the FOM, as indicated in Table 3.17. If the number of POD modes is reduced to 5, the solution is slightly different from that of the FOM in the left part of the domain, but the speed-up goes up to $\times 38$.




(a) S6. Input param. (b) S7. Input param. (c) S8. Input param. (d) S9. Input param.

(e) S6. ROM solution. (f) S7. ROM solution. (g) S8. ROM solution. (h) S9. ROM solution.

(i) S6. Differences. (j) S7. Differences. (k) S8. Differences. (l) S9. Differences.

Figure 3.25: TC5. Advection-diffusion cases. ROM solutions and errors.

### 3.3.3 Test case 6. Input parameters: initial Gaussian profile

This test case is designed to study when the coefficients defining the IC act as input parameters. A Gaussian profile is defined as IC

$$u(x,0) = \bar{u} + u_0 e^{-c(x-x_0)^2}, \tag{3.22}$$

where all the coefficients are treated as input parameters

$$\mu_1 = \bar{u}, \quad \mu_2 = u_0, \quad \mu_3 = c, \quad \mu_4 = x_0.$$

This parameters are given random values for the building of the training sample set.

The same target IC has been solved for different 12 different ROM settings. In addition to the number of POD modes $M_{\text{POD}}$, the number of samples of the training set, $M_{\text{train}}$, has been varied too. Figures 3.26a-3.26d show the training ICs of each subcase together with the target IC of the ROM. The rest of the settings of the problem are the same for all subcases and they are contained in Table 3.18.

The time-space domain of the case is defined as $(x,t) \in [0,20]\times[0,10]$. Free boundary conditions are considered. The spatial domain is discretized using $I_x = 200$ volume cells and the time step is dynamically chosen according to (2.5).

| Subcase | IC | $M_{\text{POD}}$ | $M_{\text{train}}$ |
|---|---|---|---|
| 1 | Eq. (3.22) & Fig. 3.26a | 5 | 6 |
| 2 | Eq. (3.22) & Fig. 3.26b | 5 | 8 |
| 3 | Eq. (3.22) & Fig. 3.26c | 10 | 6 |
| 4 | Eq. (3.22) & Fig. 3.26d | 10 | 8 |
| $L = 20$, $T = 10$, $\nu = 0.05$, $a = 0.5$ | | | |
| BCs: free, $I_x = 200$, CFL $= 0.9$, $N_T = 168$ | | | |

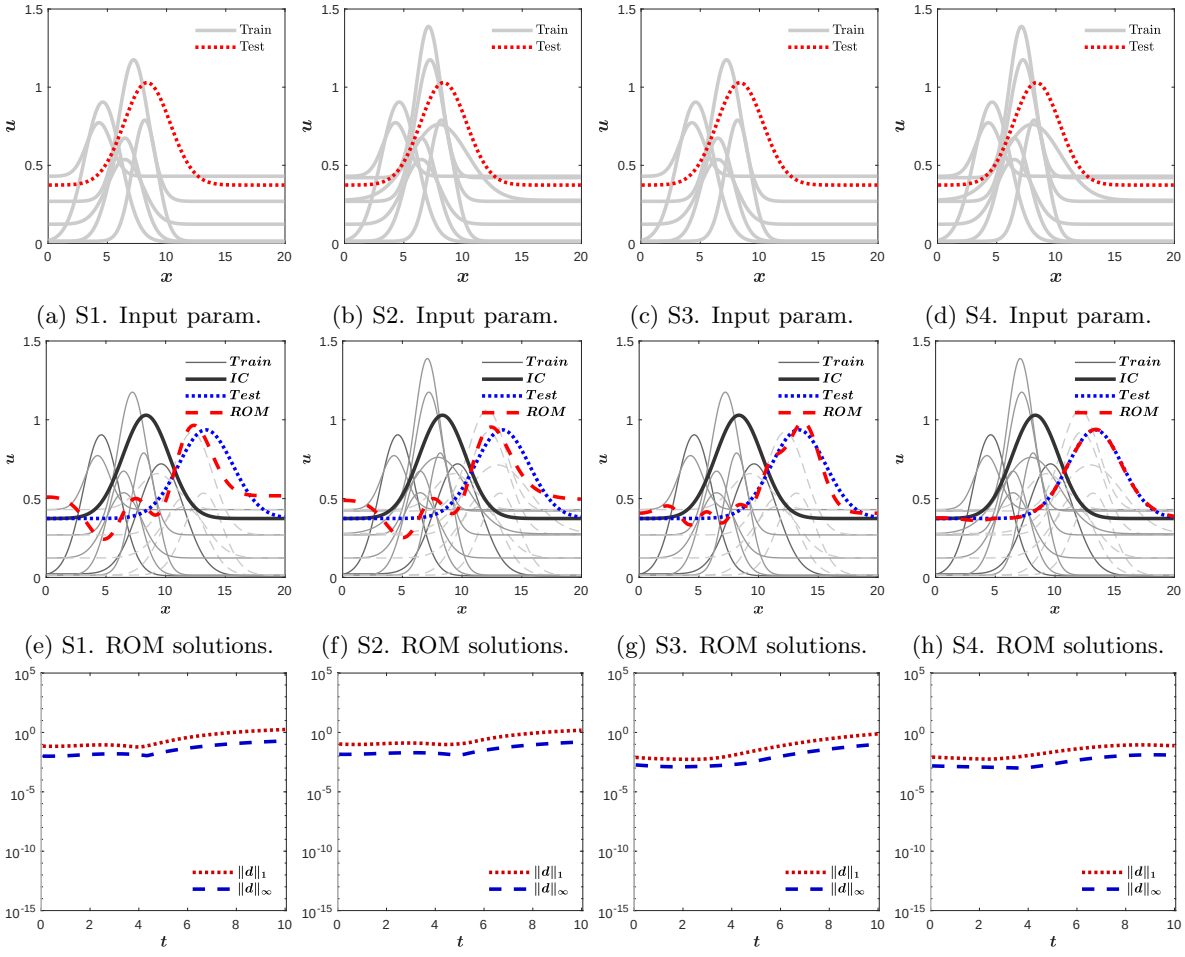
Table 3.18: TC6. Problem settings.

On the one hand, as it can be seen in Figure 3.26h, only Subcase 4 is good enough. In this subcase, 10 POD modes are solved, and 8 training samples are needed, as indicated by the differences in Figures 3.26i-3.26l.

On the other hand, Subcase 4 is just $\times 15$ faster than the FOM, as shown in Table 3.19. In any case, it is worth remembering that this acceleration is the one that compares the effective time of the ROM to calculate the target CI with the FOM if it were to calculate the target CI. So in terms of efficiency, the ROM is good, because it is predicting solutions for new values of the parameters.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | 1.76 | 1.53 | $7.59 \cdot 10^{-1}$ | $7.71 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $1.91 \cdot 10^{-1}$ | $1.56 \cdot 10^{-1}$ | $1.02 \cdot 10^{-1}$ | $1.23 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $1.37 \cdot 10^{-3}$ | $1.37 \cdot 10^{-3}$ | $1.37 \cdot 10^{-3}$ | $1.37 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $2.68 \cdot 10^{-5}$ | $2.76 \cdot 10^{-5}$ | $9.19 \cdot 10^{-5}$ | $9.16 \cdot 10^{-5}$ |
| Speed-up | $\times 51$ | $\times 49$ | $\times 15$ | $\times 15$ |

Table 3.19: TC6. Results for all subcases.




(a) S1. Input param. (b) S2. Input param. (c) S3. Input param. (d) S4. Input param.

(e) S1. ROM solutions. (f) S2. ROM solutions. (g) S3. ROM solutions. (h) S4. ROM solutions.

(i) S1. Differences. (j) S2. Differences. (k) S3. Differences. (l) S4. Differences.

Figure 3.26: TC6. Input parameters (ICs), ROM solutions and differences for all subcases.

### 3.3.4 Test case 7. Input parameters: initial discontinuity

Again, the input parameters considered are the constants that define the IC. In this particular case, a discontinuity is initially set, defined by

$$u(x,0) = \begin{cases} u_L, & \text{if } x \leq x_0, \\ u_R, & \text{if } x > x_0, \end{cases} \tag{3.23}$$

where the position and the left and right levels are the input parameters

$$\mu_1 = x_0, \quad \mu_2 = u_L, \quad \mu_3 = u_R.$$

As shown in Figures 3.27a-3.27d, the levels to the left and right of the discontinuity can be inverted in the training phase.

The time-space domain of the case is defined as $(x,t) \in [0,20] \times [0,20]$. The spatial domain is discretized using $I_x = 200$ volume cells, and the time steps are computed dynamically according to (2.5). Free boundary conditions are considered.

As indicated in Table 3.20, 4 subcases have been computed with different numbers of training samples and POD modes.

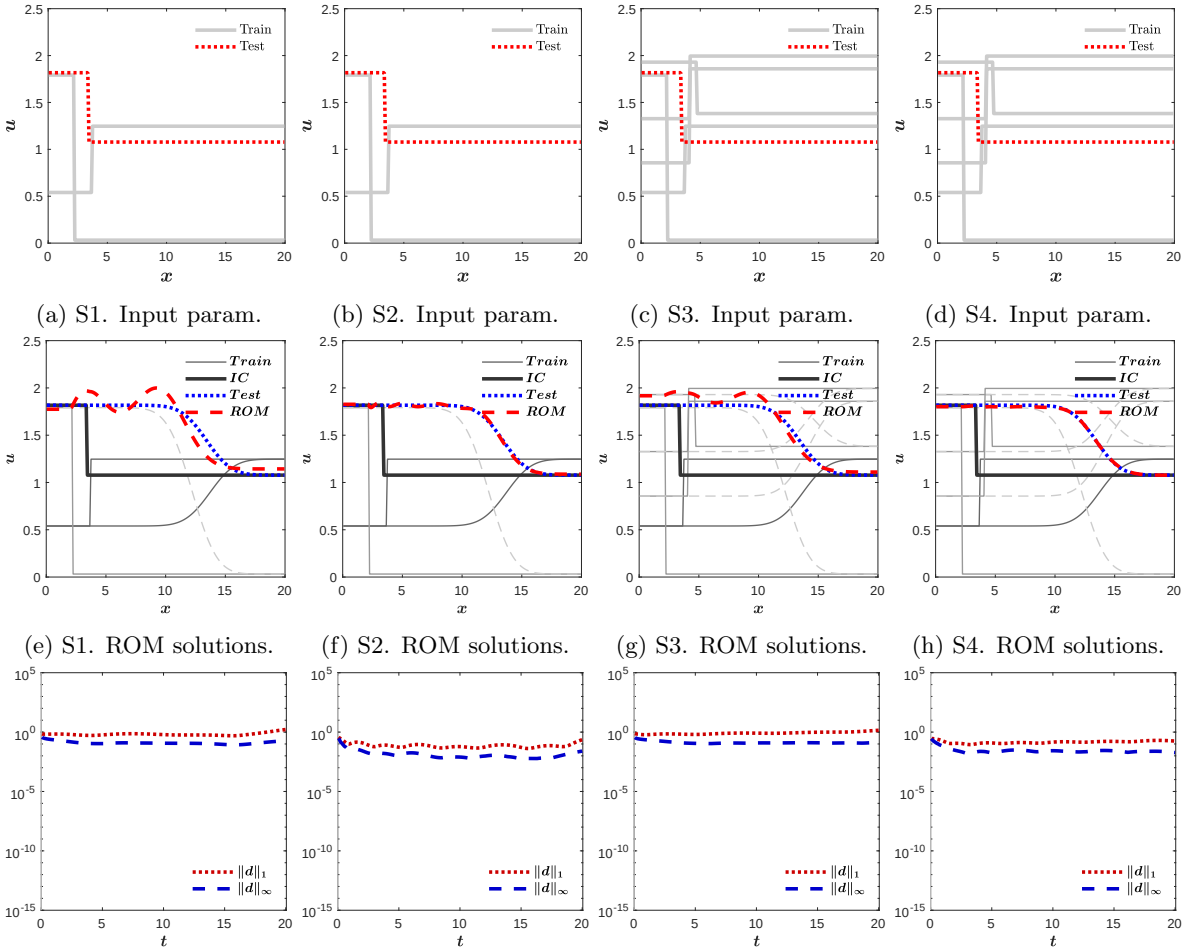
| Subcase | IC | $M_{\text{POD}}$ | $M_{\text{train}}$ |
|---|---|---|---|
| 1 | Eq. (3.23) & Fig. 3.27a | 5 | 2 |
| 2 | Eq. (3.23) & Fig. 3.27b | 10 | 2 |
| 3 | Eq. (3.23) & Fig. 3.27c | 5 | 5 |
| 4 | Eq. (3.23) & Fig. 3.27d | 10 | 5 |
| $L = 20$, $T = 20$, $\nu = 0.05$, $a = 0.5$ | | | |
| BCs: free, $I_x = 200$, CFL $= 0.9$, $N_T = 335$ | | | |

Table 3.20: TC7. Problem settings.

As shown in Figures 3.27, using just 5 POD modes (Subcases 1 and 3) are not enough to obtain good results. The ROM computes accurate solutions with speed-ups of $\times 12$ when using 10 POD modes.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | 16.46 | 2.29 | 14.23 | 1.77 |
| $\|d\|_\infty$ | $1.87 \cdot 10^{-1}$ | $2.58 \cdot 10^{-2}$ | $1.44 \cdot 10^{-1}$ | $1.88 \cdot 10^{-2}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $2.23 \cdot 10^{-3}$ | $2.23 \cdot 10^{-3}$ | $2.23 \cdot 10^{-3}$ | $2.23 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $5.06 \cdot 10^{-5}$ | $1.76 \cdot 10^{-4}$ | $5.08 \cdot 10^{-5}$ | $1.76 \cdot 10^{-4}$ |
| Speed-up | $\times 44$ | $\times 13$ | $\times 44$ | $\times 13$ |

Table 3.21: TC7. Results for all subcases.




(a) S1. Input param. (b) S2. Input param. (c) S3. Input param. (d) S4. Input param.

(e) S1. ROM solutions. (f) S2. ROM solutions. (g) S3. ROM solutions. (h) S4. ROM solutions.

(i) S1. Differences. (j) S2. Differences. (k) S3. Differences. (l) S4. Differences.

Figure 3.27: TC7. Input parameters (ICs), ROM solutions and differences for all subcases.

### 3.3.5 Test case 8. Input parameters: boundary Gaussian profile

Time-dependent Dirichlet BCs can also be considered within the study that is being done in this section. In particular, the constants that define the Gaussian profile that enters the domain initially at rest ($u(x,0) = 1$) from the left boundary

$$u(0,t) = 1 + u_0 e^{-c(t-t_0)^2}, \tag{3.24}$$

where the coefficients are given different random values

$$\mu_1 = u_0, \quad \mu_2 = c, \quad \mu_3 = t_0.$$

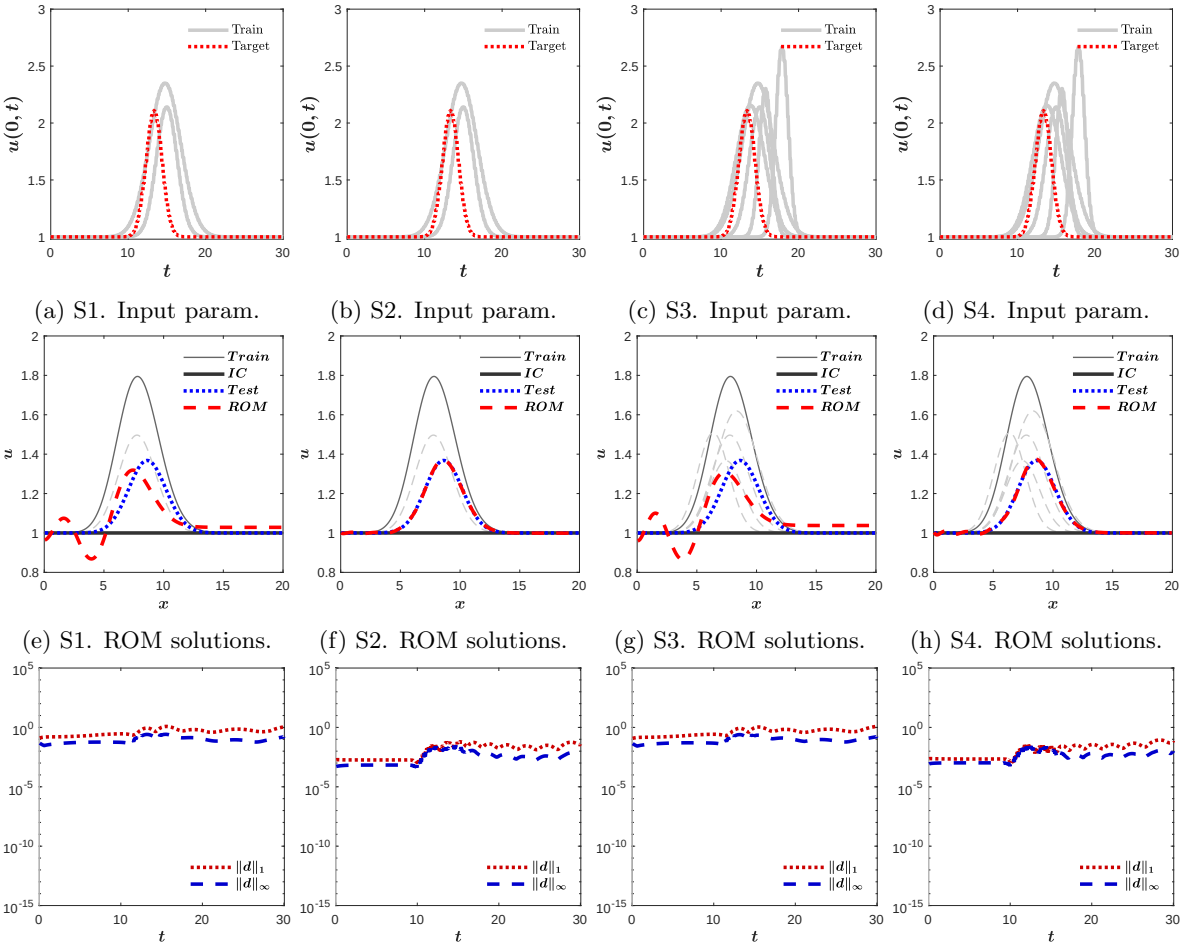
The time-space domain of the case is defined as $(x,t) \in [0,20] \times [0,30]$. The spatial domain is divided into $I_x = 200$ volumes cells, and the time step is dynamically chosen according to (2.5). Free boundary condition is imposed in the right domain.

Four different subcases are considered and are contained in Table 3.22. Figures 3.28a-3.28d show the time evolution of the left BC for the different training samples.

| Subcase | BCs | $M_{\text{POD}}$ | $M_{\text{train}}$ |
|---|---|---|---|
| 1 | Eq. (3.24) & Fig. 3.28a & free | 5 | 2 |
| 2 | Eq. (3.24) & Fig. 3.28b & free | 10 | 2 |
| 3 | Eq. (3.24) & Fig. 3.28c & free | 5 | 5 |
| 4 | Eq. (3.24) & Fig. 3.28d & free | 10 | 5 |
| $L = 20$, $T = 20$, $\nu = 0.05$, $a = 0.5$ | | | |
| IC: 1, $I_x = 200$, CFL $= 0.9$, $N_T = 502$ | | | |

Table 3.22: TC8. Problem settings.

As indicated in the results of Subcases 1 and 3 shown in Figures 3.28e-3.28e, 5 POD modes are not enough, but, with 10 POD modes, the ROM can obtain good results in terms of accuracy (Fig. 3.28j and 3.28l) even with just 2 training samples, as obtained by Subcase 2. Both Subcases 2 and 4 are 12 times faster than the FOM, as shown in Table 3.23.




(a) S1. Input param. (b) S2. Input param. (c) S3. Input param. (d) S4. Input param.

(e) S1. ROM solutions. (f) S2. ROM solutions. (g) S3. ROM solutions. (h) S4. ROM solutions.

(i) S1. Differences. (j) S2. Differences. (k) S3. Differences. (l) S4. Differences.

Figure 3.28: TC8. Input parameters (BCs), ROM solutions and differences for all subcases.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | 1.11 | $3.66 \cdot 10^{-2}$ | 1.24 | $7.24 \cdot 10^{-2}$ |
| $\|d\|_\infty$ | $1.60 \cdot 10^{-1}$ | $4.32 \cdot 10^{-3}$ | $1.64 \cdot 10^{-1}$ | $9.77 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $3.28 \cdot 10^{-3}$ | $3.28 \cdot 10^{-3}$ | $3.28 \cdot 10^{-3}$ | $3.28 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $7.63 \cdot 10^{-5}$ | $2.63 \cdot 10^{-4}$ | $7.57 \cdot 10^{-5}$ | $2.64 \cdot 10^{-4}$ |
| Speed-up | $\times 43$ | $\times 12$ | $\times 43$ | $\times 12$ |

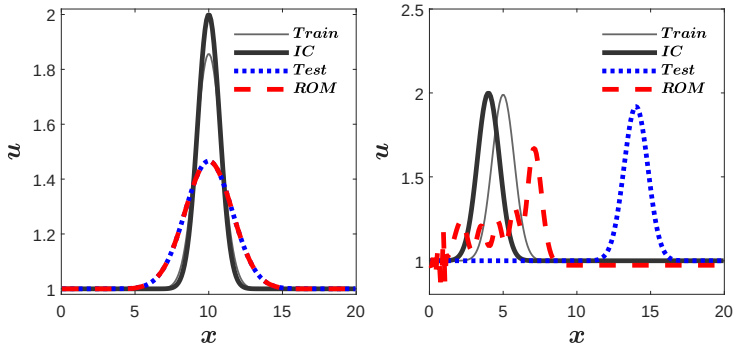
Table 3.23: TC8. Results for all subcases.

## 3.4 Beyond the training time (I)

The computation of solutions beyond the training time is not always possible and represents one of the major limitations of the POD method (and, in general, SVD-based methods, such as DMD) when dealing with advection-dominated equations [3, 97, 110], this being a challenging problem of recent interest.

The following example will help to better appreciate this. Consider the 1D advection-diffusion equation (2.1). If diffusion dominates, as shown in Figure 3.29a, the ROM is able to obtain solutions in times much larger than the training time. In this case, the ROM is trained with the time evolution of an initial Gaussian profile with $\nu = 0.05$ and no advection velocity. The training time is $T_{\text{train}} = 2$ and the final time is $T = 20$.

But if advection dominates, as in Figure 3.29b, the ROM fails to solve the same training and final times, with velocity $a = 0.5$. Thefore, it is necessary to modify the standard ROM strategy in order to be able to calculate solutions beyond the training time.




(a) Diffusion-dominated case. (b) Advection-dominated case.

Figure 3.29: Example to illustrate the inability of standard ROMs to predict solutions beyond the training time.

Computing predicted solutions with a ROM for times longer than the training time would suppose a major step in the field of computational hydraulics. Different works have been carried out in the field of temporal prediction, such as the application to the diffusion equation [109], the application to the dominant convection equation based on a Lagrangian formulation of the ROM [110] and the application to hyperbolic equations based on auto-encoder type

algorithms [120].

In the present thesis, a ROM strategy based on the coordinate transformation proposed in [69] and called CTROM is proposed with the aim of predicting solutions beyond the training time in the Eulerian framework. Following this method, the computational grid evolves in time following a main characteristic line. Other works propose different mapping strategies, such as the Lagragian framework approach in [110, 119, 120].

The CTROM herein introduced is a genuinely 1D method. An extension of this strategy to 2D problems by means of the Radon transform is also presented [151]. This extension is based on the intertwining property of the Radon transform, which allows to express the 2D problem as a collection of 1D problems, all of them written in terms of a univariate derivative [40, 154, 156]. Then, the CTROM strategy can be applied to each of those 1D problems, and the solution in the 2D physical domain is computed by means of a back-projection, i.e., the inverse Radon transform. This approach proves useful for the application of the CTROM method to hyperbolic partial differential equations in 2D.

First, the theoretical framework of the CTROM method will be presented, which includes the transformation of the equation of interest, the FOM derived from it and, finally, the ROM.

Then, the CTROM methodology is applied to:

- the linear 1D advection-diffusion equation,
- to a variation of the previous equation in which the transport velocity depends on time,
- to a system of coupled 1D equations
- and to the 2D advective equation will be presented.

In all these problems, the application of the CTROM methodology will be introduced first. And all of them are followed by their corresponding numerical results.

This section entitled *Beyond the training time* is continued in the next two chapters, where the CTROM methodology is extended to non-linear problems and SWE.

### 3.4.1 Coordinate transform-based ROM strategy

A modified ROM is presented based on an appropriate coordinate transformation to allow the computation of solutions beyond the training time. For that purpose, an interior point $d^0 \in (0, L)$ must be identified in the initial condition, such as, for example, the peak of a Gaussian function or a discontinuity. The solution is approximated in a new coordinate system which is aligned with the characteristic curve emanating from point $d^0$.

Consider the characteristic curve $d(t)$ defined by

$$d'(t) = a(t),\ 0 < t \leq T, \quad d(0) = d^0.$$

It is given by

$$d(t) = \int_{s=0}^{t} a(s)\, ds + d(0),$$

and it is assumed that $d(T) < L$. This characteristic curve is used to define the following mapping [69]

$$\tilde{x}(t) = \begin{cases} \dfrac{d(0)}{d(t)} x, & \text{if } 0 \leq x \leq d(t), \\ L - \dfrac{L - d(0)}{L - d(t)}(L - x), & \text{if } d(t) < x \leq L. \end{cases} \tag{3.25}$$

Note that the sub-domains $\{(x,t),\, 0 \leq x \leq d(t),\, 0 \leq t \leq T\}$ and $\{(x,t),\, d(t) \leq x \leq L,\, 0 \leq t \leq T\}$ are transformed into the rectangular sub-domains $[0, d(0)] \times [0, T]$ and $[d(0), L] \times [0, T]$, respectively. In addition, note that $\tilde{x} = x$ at $t = 0$. The transformation proposed maps the physical space into a new one, where the IC is defined an evolved in time by the FOM to obtain the snapshot matrix, as outlined in Figure 3.30.

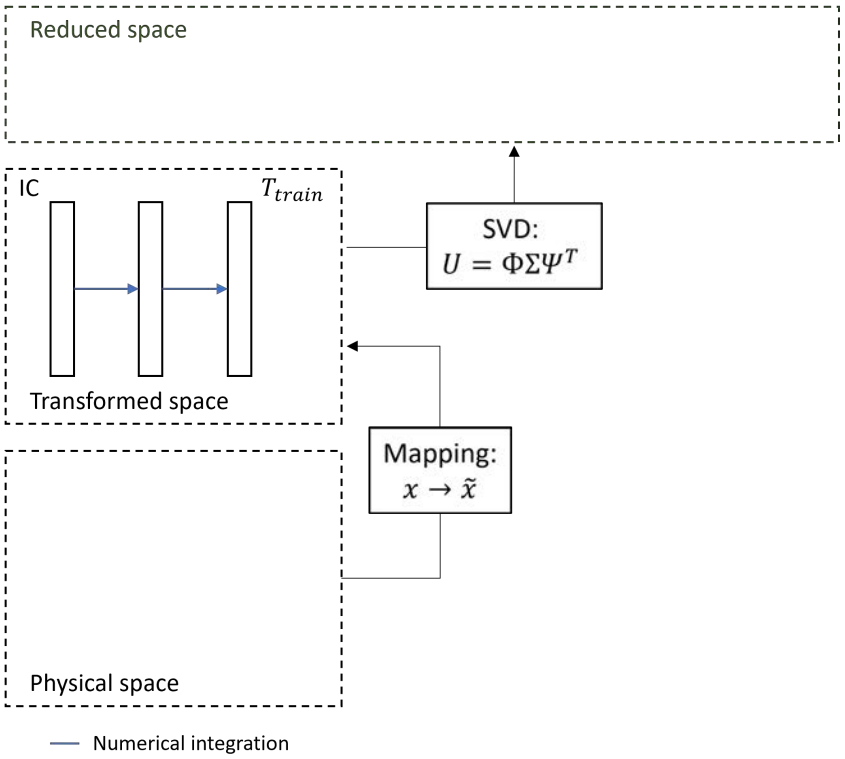



Figure 3.30: ROM training.

Once the reduced space is defined, the IC is solved up to the final time $T$. The final solution must be projected back to the transformed space and then transformed back to the physical space, as indicated in Figure 3.31.

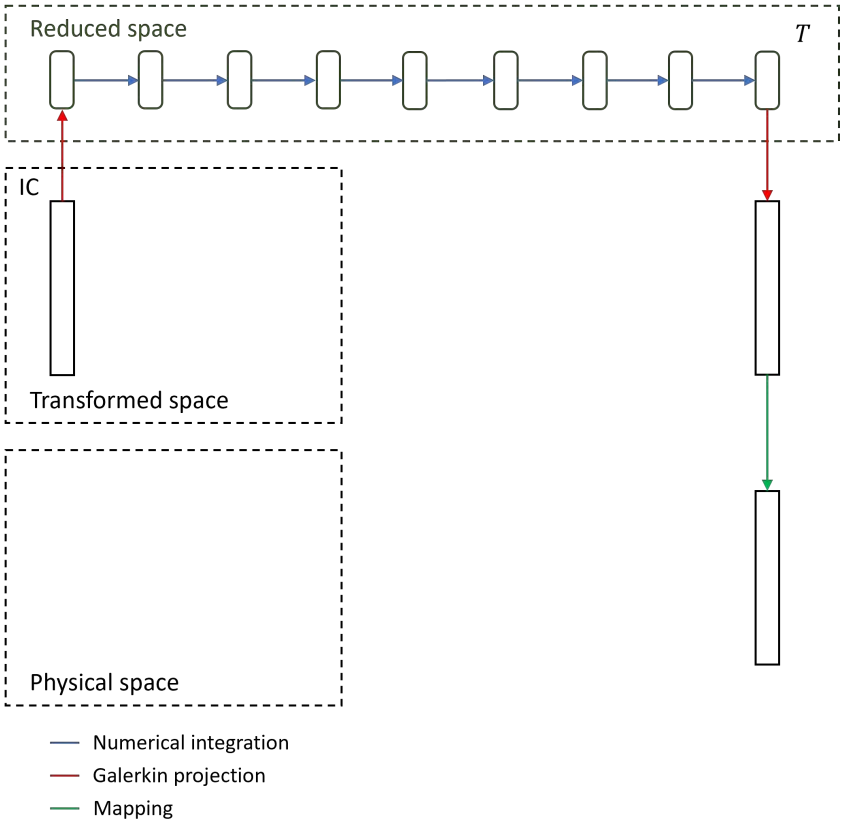



Figure 3.31: ROM solving beyond the training time.

### 3.4.2 1D advection-diffusion equation

Using the chain rule, the transformed version of the 1D advection-diffusion equation (2.1) can be obtained.

$$\frac{\partial u}{\partial x} = \begin{cases} \dfrac{d(0)}{d(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } 0 < x < d(t), \\ \dfrac{L-d(0)}{L-d(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } d(t) < x < L, \end{cases}$$

and

$$\frac{\partial u}{\partial t} = \begin{cases} \dfrac{\partial \tilde{u}}{\partial t} - a(t)\dfrac{\tilde{x}}{d(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } 0 < x < d(t), \\ \dfrac{\partial \tilde{u}}{\partial t} - a(t)\dfrac{L-\tilde{x}}{L-d(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}, & \text{if } d(t) < x < L. \end{cases}$$

Thus, the following problem is obtained when the mapping (3.25) is applied to (2.1)

$$\begin{cases} \dfrac{\partial \tilde{u}}{\partial t} + \left[a(t)\dfrac{d(0)}{d(t)} - a(t)\dfrac{\tilde{x}}{d(t)}\right]\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = \nu\left(\dfrac{d(0)}{d(t)}\right)^2 \dfrac{\partial^2 \tilde{u}}{\partial \tilde{x}^2}, & \text{if } 0 < \tilde{x} < d(0), \\ \dfrac{\partial \tilde{u}}{\partial t} + \left[a(t)\dfrac{L-d(0)}{L-d(t)} - a(t)\dfrac{L-\tilde{x}}{L-d(t)}\right]\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = \nu\left(\dfrac{L-d(0)}{L-d(t)}\right)^2 \dfrac{\partial^2 \tilde{u}}{\partial \tilde{x}^2}, & \text{if } d(0) < \tilde{x} < L, \end{cases} \tag{3.26}$$

where $\tilde{u}(\tilde{x},t) = u(x,t)$.

**Coordinate transform-based full-order model**

In the transformed variables $(\tilde{x},t)$, the computational mesh is rectangular, but in the physical variables $(x,t)$, it is a time dependent mesh which is aligned with the characteristic curve $d(t)$. The spatial mesh in the transformed domain is uniform in the subintervals $[0,d(0)]$ and $[d(0),L]$ with $\tilde{x}_{I_d+1/2} = d(0)$. The coordinate transformed-based FOM (CTFOM) is defined on this mesh and it is given by

$$\begin{cases} \tilde{u}_i^{n+1} &= \tilde{u}_i^n - \dfrac{\Delta t}{\Delta x}\left[\dfrac{d^0}{d^n}\left(\tilde{f}_{i+1/2}^{n,-,*} - \tilde{f}_{i-1/2}^{n,+,*}\right) - \dfrac{1}{d^n}\left(\tilde{f}_{i+1/2}^{n,-,**} - \tilde{f}_{i-1/2}^{n,+,**}\right)\right] \\ &\quad + \nu\dfrac{\Delta t}{\Delta \tilde{x}^2}\left(\dfrac{d^0}{d^n}\right)^2 \left(\tilde{u}_{i+1}^n - 2\tilde{u}_i^n + \tilde{u}_{i-1}^n\right), \text{with } i = 2, ..., I_d, \\ \tilde{u}_i^{n+1} &= \tilde{u}_i^n - \dfrac{\Delta t}{\Delta x}\left[\dfrac{L-d^0}{L-d^n}\left(\tilde{f}_{i+1/2}^{n,-,*} - \tilde{f}_{i-1/2}^{n,+,*}\right) - \dfrac{1}{L-d^n}\left(\tilde{f}_{i+1/2}^{n,-,**} - \tilde{f}_{i-1/2}^{n,+,**}\right)\right] \\ &\quad + \nu\dfrac{\Delta t}{\Delta \tilde{x}^2}\left(\dfrac{L-d^0}{L-d^n}\right)^2 \left(\tilde{u}_{i+1}^n - 2\tilde{u}_i^n + \tilde{u}_{i-1}^n\right), \text{with } i = I_d+1, ..., I_x - 1, \end{cases} \tag{3.27}$$

where $a^n = a(t^n)$, $d^n = d(t^n)$ and the numerical fluxes are

$$\tilde{f}_{i+1/2}^{n,\pm,*} = f(\tilde{u}_i^n) \mp \left(\bar{\lambda}^{\pm}\right)_{i+1/2}^n \delta\tilde{u}_{i+1/2}^n,$$

$$\tilde{f}_{i+1/2}^{n,\pm,**} = \begin{cases} f(\tilde{u}_i^n) \mp \tilde{x}_{i+1/2}\left(\bar{\lambda}^{\pm}\right)_{i+1/2}^n \delta\tilde{u}_{i+1/2}^n, & \text{if } i \le I_d, \\ f(\tilde{u}_i^n) \mp \left(L - \tilde{x}_{i+1/2}\right)\left(\bar{\lambda}^{\pm}\right)_{i+1/2}^n \delta\tilde{u}_{i+1/2}^n, & \text{if } i > I_d, \end{cases}$$

with $\delta\tilde{u}_{i+1/2}^n = \tilde{u}_{i+1}^n - \tilde{u}_i^n$, and

$$\left(\bar{\lambda}^{\pm}\right)_{i+1/2}^n = \frac{1}{2}\left(\bar{\lambda} \pm |\bar{\lambda}|\right)_{i+1/2}^n = \frac{1}{2}\left(a^n \pm |a^n|\right),$$

The time step is computed to satisfy the following stability condition

$$\Delta t = CFL\frac{\Delta \tilde{x}^2}{\Delta \tilde{x}\max\{\tilde{a}_1, \tilde{a}_2\} + 2\max\{\tilde{\nu}_1, \tilde{\nu}_2\}}, \tag{3.28}$$

where the modified velocities and modified viscosities are

$$
\begin{aligned}
\tilde{a}_1 &= \max_{0\le\tilde{x}\le d(0)} \left| a\frac{d(0)}{d(t)} - a\frac{\tilde{x}}{d(t)} \right| = |a|\frac{d(0)}{d(t)}, \\
\tilde{a}_2 &= \max_{d(0)\le\tilde{x}\le L} \left| a\frac{L-d(0)}{L-d(t)} - a\frac{L-\tilde{x}}{L-d(t)} \right| = |a|\frac{L-d(0)}{L-d(t)}, \\
\tilde{\nu}_1 &= \nu\left(\frac{d(0)}{d(t)}\right)^2, \quad \tilde{\nu}_2 = \nu\left(\frac{L-d(0)}{L-d(t)}\right)^2.
\end{aligned}
$$

**Coordinate transform-based reduced-order model**

The explicit updating equation of the coordinate transform-based reduced order model (CTROM) is obtained following the same three steps aforementioned

$$
\begin{aligned}
\hat{u}_p^{n+1} = \hat{u}_p^n &+ \frac{\Delta t}{\Delta\tilde{x}}\frac{1}{d^n}\sum_{k=1}^{M_{\mathrm{POD}}} A_{pk}^L \hat{u}_k^n + \frac{\Delta t}{\Delta\tilde{x}}\frac{1}{L-d^n}\sum_{k=1}^{M_{\mathrm{POD}}} A_{pk}^R \hat{u}_k^n \\
&+ \frac{\Delta t}{\Delta\tilde{x}^2}\frac{1}{(d^n)^2}\sum_{k=1}^{M_{\mathrm{POD}}} B_{pk}^L \hat{u}_k^n + \frac{\Delta t}{\Delta\tilde{x}^2}\frac{1}{(L-d^n)^2}\sum_{k=1}^{M_{\mathrm{POD}}} B_{pk}^R \hat{u}_k^n,
\end{aligned}
\tag{3.29}
$$

where the coefficients are

$$
\begin{aligned}
A_{pk}^L = &-d_0\frac{1}{2}a\left[(\phi_{2,k}-\phi_{1,k})\,\phi_{1,p} + \sum_{i=2}^{I_d}(\phi_{i+1,k}-\phi_{i-1,k})\,\phi_{i,p}\right] \\
&+ d_0\frac{1}{2}|a|\left[(\phi_{2,k}-\phi_{1,k})\,\phi_{1,p} + \sum_{i=2}^{I_d}(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k})\,\phi_{i,p}\right] \\
&+ \frac{1}{2}(a-|a|)\sum_{i=1}^{I_d}\tilde{x}_{i+1/2}(\phi_{i+1,k}-\phi_{i,k})\,\phi_{i,p} \\
&+ \frac{1}{2}(a+|a|)\sum_{i=2}^{I_d}\tilde{x}_{i-1/2}(\phi_{i,k}-\phi_{i-1,k})\,\phi_{i,p}, \\
A_{pk}^R = &-(L-d_0)\frac{1}{2}a\left[\sum_{i=I_d+1}^{I_x-1}(\phi_{i+1,k}-\phi_{i-1,k})\,\phi_{i,p} + (\phi_{I_x,k}-\phi_{I_x-1,k})\,\phi_{I_x,p}\right] \\
&+ (L-d_0)\frac{1}{2}|a|\left[\sum_{i=I_d+1}^{I_x-1}(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k})\,\phi_{j,p} - (\phi_{I_x,k}-\phi_{I_x-1,k})\,\phi_{I_x,p}\right] \\
&+ \frac{1}{2}(a-|a|)\sum_{i=I_d+1}^{I_x-1}\left(L-\tilde{x}_{i+1/2}\right)(\phi_{i+1,k}-\phi_{i,k})\,\phi_{i,p} \\
&+ \frac{1}{2}(a+|a|)\sum_{i=I_d+1}^{I_x}\left(L-\tilde{x}_{i-1/2}\right)(\phi_{i,k}-\phi_{i-1,k})\,\phi_{i,p}, \\
B_{pk}^L = &\,(d_0)^2\,\nu\left[(\phi_{2,k}-\phi_{1,k})\,\phi_{1,p} + \sum_{i=2}^{I_d}(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k})\,\phi_{i,p}\right], \\
B_{pk}^R = &\,(L-d_0)^2\,\nu\left[\sum_{i=I_d+1}^{I_x-1}(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k})\,\phi_{i,p} - (\phi_{I_x,k}-\phi_{I_x-1,k})\,\phi_{I_x,p}\right],
\end{aligned}
$$

where $I_d$ is the position of the adjacent cell to $\tilde{x}_{I_d+1/2} = d(0)$.

**Test case 9. Limits of time prediction**

Let's consider the 1D advection-diffusion equation (2.1), with constant velocity $a$ so that the characteristic curve evolves in time as

$$d(t) = d(0) + at. \tag{3.30}$$

In this case, the limits of time prediction are studied. For this purpose, four subcases are defined in which different training times, $T_{\text{train}} = (1, 2, 5, 7)$, are used to solve the same final time, $T = 15$. All these settings are shown in Table 3.24.

The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, 15]$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-0.6(x-4)^2}, \tag{3.31}$$

Free BCs are imposed in both boundaries. The advection velocity is $a = 0.5$ and the diffusion coefficient, $\nu = 0.01$.

The starting point of the characteristic curve $d(t)$ is placed at the location of the maximum of the initial Gaussian profile, i.e., $d(0) = 4$. It can easily be seen how the mapping (3.25) works: the mesh in the physical domain evolves in time following the characteristic curve (3.30) as shown in Figures 3.32. In all subcases the ROM uses the maximum number of POD modes.

| Subcase | $T_{\text{train}}$ | $N_{T_{\text{train}}}$ | $M_{\text{POD}}$ |
|---|---|---|---|
| 1 | 1 | 13 | 13 |
| 2 | 2 | 25 | 25 |
| 3 | 5 | 60 | 60 |
| 4 | 7 | 84 | 84 |
| $L = 20$, $T = 8$, $a = 0.5$, $\nu = 0.01$, IC: Eq. (3.31) | | | |
| BCs: free, $d(0) = 4$, $I_x = 200$, CFL $= 0.9$, $N_T = 177$ | | | |

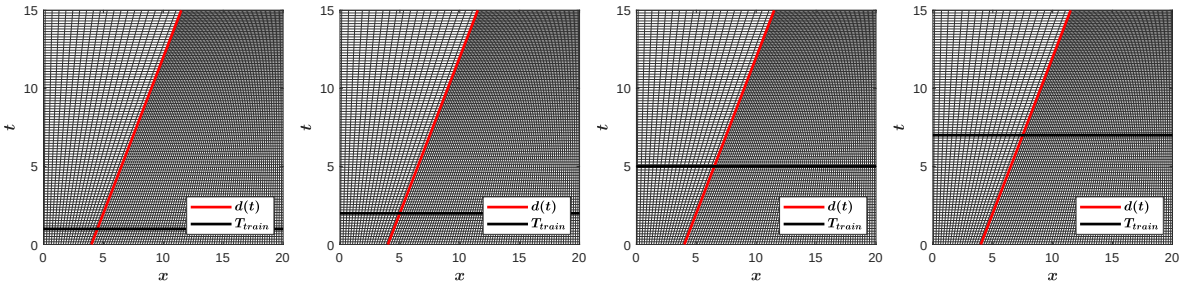
Table 3.24: TC9. Problem settings.

The solutions obtained by the FOM in the transformed domain $\tilde{u}$ are shown in Figures 3.33a-3.33d, where it can be seen that the Gaussian profile remains in the same position while is slightly modified due to the diffusion. But, when taken back to the physical space, the Gaussian profile is advected in space, Figures 3.33e-3.33h.

A similar behaviour can be noted in the ROM solutions in Figure 3.34. Subcases 1 and 2 show errors that are not present in subcase 3 and 4. The training time and the final time cannot be separated by an arbitrary distance, i.e. there is a maximum time that the ROM is able to predict. This is also proven by Figures 3.35, where the solutions of the FOM and the ROM in the physical domain are compared, together with their differences.

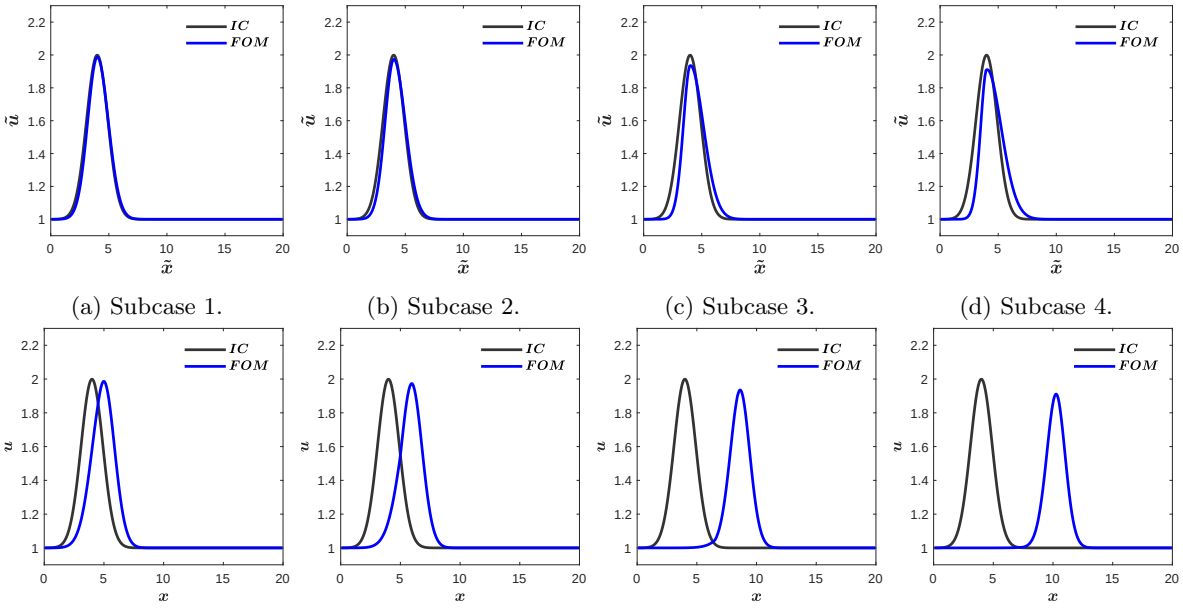
| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $5.19 \cdot 10^{-1}$ | $1.16 \cdot 10^{-1}$ | $7.70 \cdot 10^{-2}$ | $5.39 \cdot 10^{-4}$ |
| $\|d\|_\infty$ | $1.10 \cdot 10^{-1}$ | $3.51 \cdot 10^{-2}$ | $1.73 \cdot 10^{-3}$ | $1.34 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{train}}$ | $1.06 \cdot 10^{-4}$ | $2.10 \cdot 10^{-4}$ | $5.15 \cdot 10^{-4}$ | $7.08 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $1.51 \cdot 10^{-3}$ | $1.51 \cdot 10^{-3}$ | $1.51 \cdot 10^{-3}$ | $1.51 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $2.00 \cdot 10^{-4}$ | $6.55 \cdot 10^{-4}$ | $3.46 \cdot 10^{-3}$ | $6.81 \cdot 10^{-3}$ |
| Speed-up (train) | $\times 1$ | $\times 0$ | $\times 0$ | $\times 0$ |
| Speed-up (test) | $\times 8$ | $\times 2$ | $\times 0$ | $\times 0$ |

Table 3.25: TC9. Results for all subcases.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

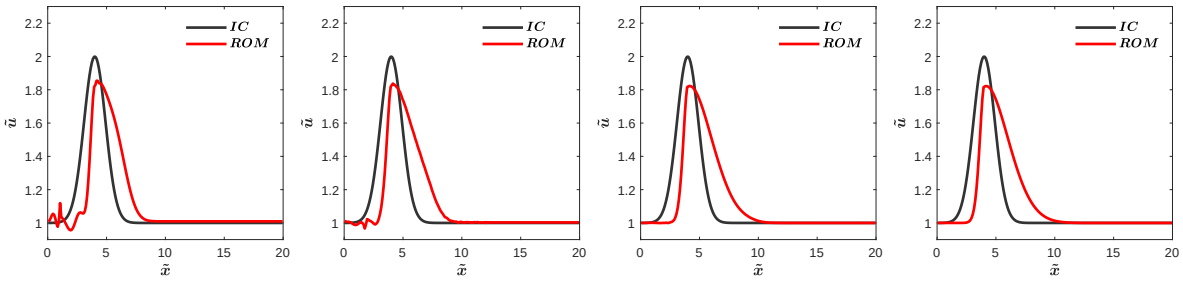
Figure 3.32: TC9. ROM meshes with different training times.



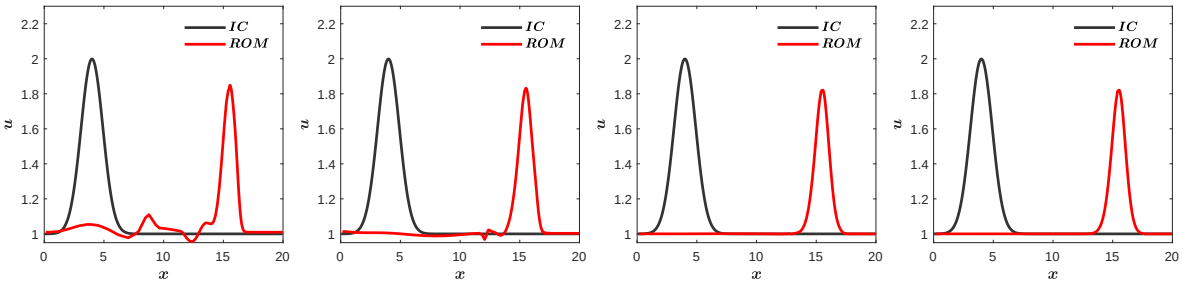

(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

(e) Subcase 1. (f) Subcase 2. (g) Subcase 3. (h) Subcase 4.

Figure 3.33: TC9. Training solutions.

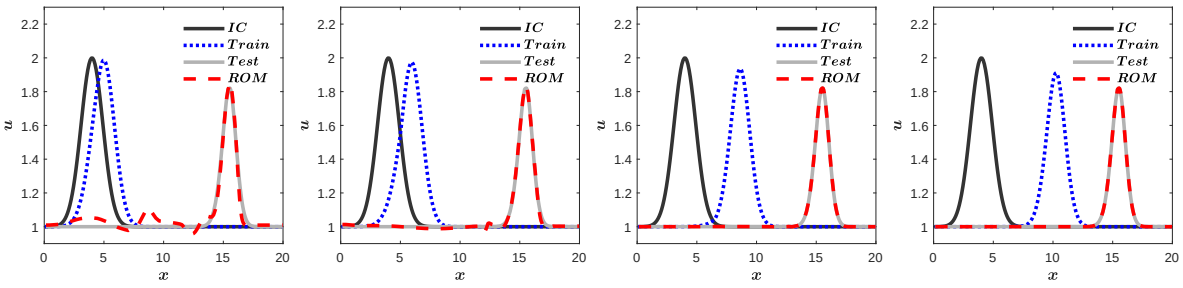



(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

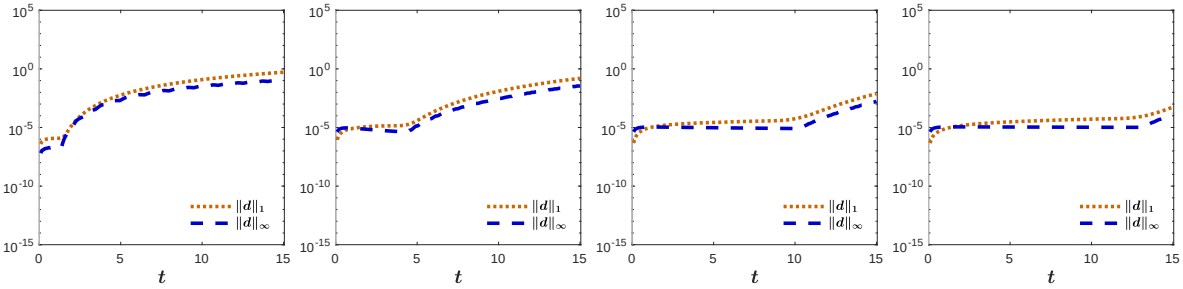



(e) Subcase 1. (f) Subcase 2. (g) Subcase 3. (h) Subcase 4.

Figure 3.34: TC9. ROM solutions.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.




(e) Subcase 1. (f) Subcase 2. (g) Subcase 3. (h) Subcase 4.

Figure 3.35: TC9. Solutions and errors.

### Test case 10. Changing the advection velocity

Taking into account the time prediction limit obtained in the previos test case, the same problem has been solved changing the value of the advection velocity and fixing the diffusion coefficient. The Péclet number is used to consider a range of advection-diffusion problems. It

is defined to be the ratio of the advection to the diffusion transport

$$Pe = \frac{a\Delta x}{\nu},$$

and, depending on the value of this number, the problem is advection or diffusion dominated. In the numerical experiments, the value of the diffusion coefficient is fixed with $\nu = 0.01$ and the advection coefficient $a$ takes the values shown in Table 3.26.

The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, 15]$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-0.6(x-4)^2}, \tag{3.32}$$

Free BCs are considered. The space domain is discretized by means of 200 volume cells and the time steps is dynamically computed according to (2.5).

The number of POD modes used to solve the ROM has been selected to cover 98% of the information according to (3.1). These results evidenced that the ROM needs more POD modes when solving advection-dominated problems, as the 98% of the information implies more POD modes as the Péclet number grows, as shown in Table 3.26.

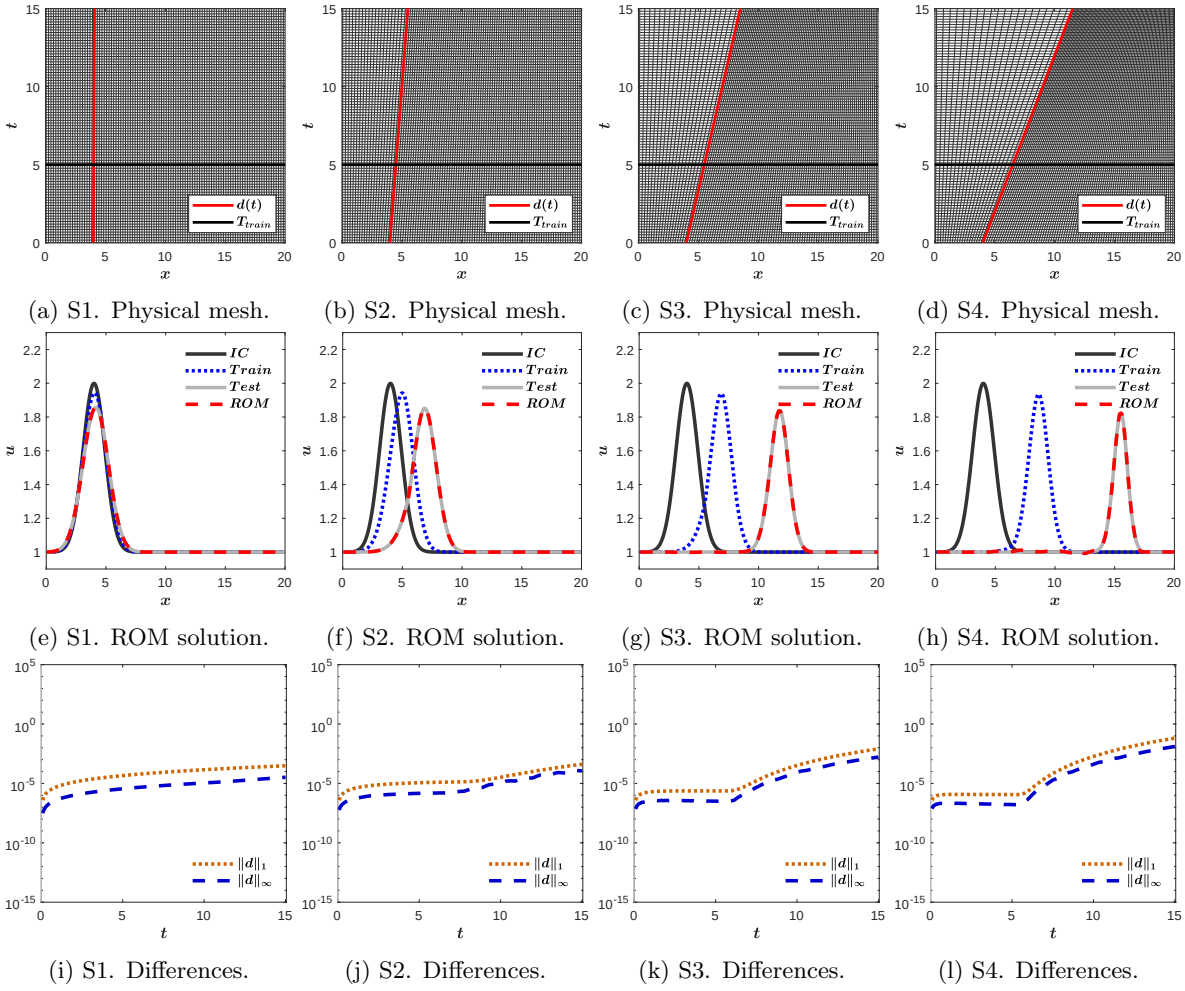
The physical mesh evolves in time according to (3.30) with $d(0) = 4$ and it can be seen in Figures 3.36a-3.36d. The CTROM is able to predict solutions in time in all subcases, as shown in Figures 3.36e-3.36h. The differences show very high accuracies, Figures 3.36i-3.36l. The speed-ups computed with respect to the test solutions are bigger when diffusion dominates, as shown in Table 3.27.

| Subcase | $a$ | Pe | $M_{\text{POD}}$ |
|---|---|---|---|
| 1 | 0.005 | 0.05 | 9 |
| 2 | 0.1 | 1 | 10 |
| 3 | 0.3 | 3 | 14 |
| 4 | 0.5 | 5 | 17 |
| $L = 20$, $T_{\text{train}} = 5$, $T = 15$, $\nu = 0.01$, IC: Eq. (3.32), BCs: free | | | |
| $d(0) = 4$, $I_x = 200$, CFL $= 0.9$, $N_{T_{\text{train}}} = 60$, $N_T = 177$, $M_{\text{train}} = 60$ | | | |

Table 3.26: TC10. Problem settings.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $3.05 \cdot 10^{-4}$ | $4.09 \cdot 10^{-4}$ | $8.13 \cdot 10^{-3}$ | $6.50 \cdot 10^{-2}$ |
| $\|d\|_\infty$ | $3.29 \cdot 10^{-5}$ | $1.3 \cdot 10^{-4}$ | $1.66 \cdot 10^{-3}$ | $1.29 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{train}}$ | $5.10 \cdot 10^{-4}$ | $5.12 \cdot 10^{-4}$ | $5.03 \cdot 10^{-4}$ | $5.19 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $1.49 \cdot 10^{-3}$ | $1.50 \cdot 10^{-3}$ | $1.48 \cdot 10^{-3}$ | $1.51 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $1.13 \cdot 10^{-4}$ | $1.26 \cdot 10^{-4}$ | $2.25 \cdot 10^{-4}$ | $3.18 \cdot 10^{-4}$ |
| Speed-up (train) | $\times 5$ | $\times 4$ | $\times 2$ | $\times 2$ |
| Speed-up (test) | $\times 13$ | $\times 12$ | $\times 7$ | $\times 5$ |

Table 3.27: TC10. Results for all subcases.




(a) S1. Physical mesh. (b) S2. Physical mesh. (c) S3. Physical mesh. (d) S4. Physical mesh.

(e) S1. ROM solution. (f) S2. ROM solution. (g) S3. ROM solution. (h) S4. ROM solution.

(i) S1. Differences. (j) S2. Differences. (k) S3. Differences. (l) S4. Differences.

Figure 3.36: TC10. Physical meshes, ROM solutions and differences.

### 3.4.3 1D advection equation with time-varying velocity

If the advection velocity varies in time as $a = 1 - t$, the characteristic line has an analytical solution, such as

$$d(t) = \int_0^t a dt' = t - \frac{t^2}{2} + d_0. \tag{3.33}$$

Since the velocity depends on time, the ROM has to be modified to get it out of the matrices

$$\begin{aligned}
\hat{u}_p^{n+1} = \hat{u}_p^n &+ \frac{\Delta t}{\Delta \tilde{x}} \sum_{k=1}^{M_{\text{POD}}} a^n \left[ \frac{d^0}{d^n} A_{pk}^L + \frac{L-d^0}{L-d^n} A_{pk}^R \right] \hat{u}_k^n + \frac{\Delta t}{\Delta \tilde{x}} \sum_{k=1}^{M_{\text{POD}}} |a^n| \left[ \frac{d^0}{d^n} B_{pk}^L + \frac{L-d^0}{L-d^n} B_{pk}^R \right] \hat{u}_k^n \\
&+ \frac{\Delta t}{\Delta \tilde{x}} \sum_{k=1}^{M_{\text{POD}}} (a^n - |a^n|) \left[ \frac{C_{pk}^L}{d^n} + \frac{C_{pk}^R}{L-d^n} \right] \hat{u}_k^n + \frac{\Delta t}{\Delta \tilde{x}} \sum_{k=1}^{M_{\text{POD}}} (a^n + |a^n|) \left[ \frac{D_{pk}^L}{d^n} + \frac{D_{pk}^R}{L-d^n} \right] \hat{u}_k^n \\
&+ \frac{\Delta t}{\Delta \tilde{x}^2} \sum_{k=1}^{M_{\text{POD}}} \left[ \left( \frac{d^0}{d^n} \right)^2 E_{pk}^L + \left( \frac{L-d^0}{L-d^n} \right)^2 E_{pk}^R \right] \hat{u}_k^n,
\end{aligned}$$

where the elements of these matrices are

$$
\begin{aligned}
A^L_{pk} &= -\frac{1}{2}\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_d}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p}\right],\\
A^R_{pk} &= -\frac{1}{2}\left[\sum_{i=I_d+1}^{I_x-1}\left(\phi_{i+1,k}-\phi_{i-1,k}\right)\phi_{i,p}+\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right],\\
B^L_{pk} &= \frac{1}{2}\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_d}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}\right],\\
B^R_{pk} &= \frac{1}{2}\left[\sum_{i=I_d+1}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{j,p}-\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right],\\
C^L_{pk} &= \frac{1}{2}\sum_{i=1}^{I_d}\tilde{x}_{i+1/2}\left(\phi_{i+1,k}-\phi_{i,k}\right)\phi_{i,p},\\
C^R_{pk} &= \frac{1}{2}\sum_{i=I_d+1}^{I_x-1}\left(L-\tilde{x}_{i+1/2}\right)\left(\phi_{i+1,k}-\phi_{i,k}\right)\phi_{i,p},\\
D^L_{pk} &= \frac{1}{2}\sum_{i=2}^{I_d}\tilde{x}_{i-1/2}\left(\phi_{i,k}-\phi_{i-1,k}\right)\phi_{i,p},\\
D^R_{pk} &= \frac{1}{2}\sum_{i=I_d+1}^{I_x}\left(L-\tilde{x}_{i-1/2}\right)\left(\phi_{i,k}-\phi_{i-1,k}\right)\phi_{i,p},\\
E^L_{pk} &= \nu\left[\left(\phi_{2,k}-\phi_{1,k}\right)\phi_{1,p}+\sum_{i=2}^{I_d}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}\right],\\
E^R_{pk} &= \nu\left[\sum_{i=I_d+1}^{I_x-1}\left(\phi_{i+1,k}-2\phi_{i,k}+\phi_{i-1,k}\right)\phi_{i,p}-\left(\phi_{I_x,k}-\phi_{I_x-1,k}\right)\phi_{I_x,p}\right],
\end{aligned}
$$

where $I_d$ is the position of the adjacent cell to $\tilde{x}_{I_d+1/2} = d(0)$. This version cannot be compacted.

**Test case 11. Time dependent advection velocity**

Let's consider now a test case in which the advection velocity depends on time, as indicated in (3.33).

The time-space domain of the case is defined as $(x,t) \in [0,2]\times[0,2]$. Initially, the Gaussian profile is defined as

$$u(x,0) = 1 + e^{-200(x-1)^2}, \tag{3.34}$$

Free BCs are considered.

| $L$ | $T_{\text{train}}$ | $T$ | $a$ | $\nu$ | IC | BCs | $d(0)$ | $I_x$ | CFL | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.1 | 2 | (3.33) | 0 | (3.34) | Free | 1 | 200 | 0.9 | 23 | 154 | 10 |

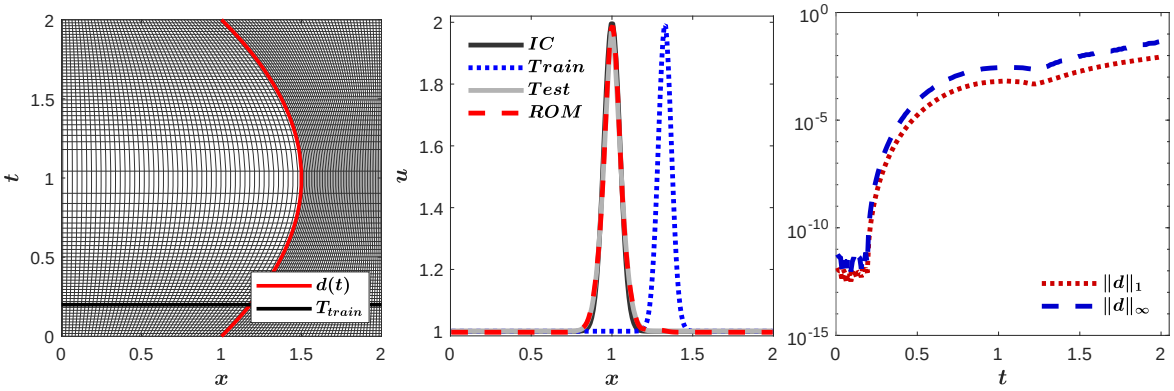
Table 3.28: TC11. Problem settings.

The spatial domain is divided into 200 volume cells and the time step is dynamically calculated according to the CFL condition (2.5), with CFL=0.9. Figure 3.37a show the physical spatial mesh as it evolves in time following the characteristic curve defined by (3.33).

The initial Gaussian profile moves to the right until $t = 1$, when $a = 0$, and, from that moment on, it moves to the left. Finally, at $T = 2$, the solution arrives at the initial position. As shown in Figures 3.37b and 3.38, the CTROM is able to reproduce the change of direction in the movement of the solution with a training time much shorter than the time in which the velocity changes sign, i.e., $T_{\text{train}} = 0.1 < 1$. The CTROM solution at the final time $T = 2$ reproduces accurately the reference solution computed with the CTFOM. The differences stay very low while $t < T_{\text{train}}$, as shown in Figure 3.37c, and then grow, but they remain low enough to maintain accuracy.

| $\|d\|_1$ | $\|d\|_\infty$ | $\tau_{\text{CPU}}^{\text{train}}$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | S.up (train) | S.up (test) |
|---|---|---|---|---|---|---|
| $8.68 \cdot 10^{-3}$ | $4.57 \cdot 10^{-2}$ | $1.00 \cdot 10^{-5}$ | $1.67 \cdot 10^{-3}$ | $3.40 \cdot 10^{-4}$ | $\times 0$ | $\times 5$ |

Table 3.29: TC11. Results.




(a) Phsyical mesh. (b) ROM solution. (c) Differences.

Figure 3.37: TC11. Physical mesh, ROM solution and differences.

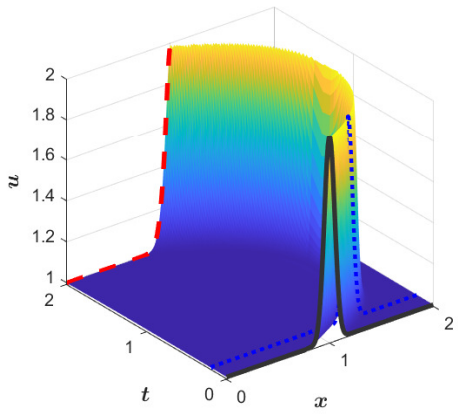



Figure 3.38: TC11. ROM solution in 3D view: training solution (blue) and ROM solution (red).

### 3.4.4 1D reactive transport of two coupled solutes

Let the system of equations of a reactive transport of two coupled solutes be

$$\begin{aligned}\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}&=-ku,\\ \frac{\partial v}{\partial t}+a\frac{\partial v}{\partial x}&=ku,\end{aligned}$$

where $u=u(x,t)$ and $v=v(x,t)$ are the two coupled solutes; and $k$ is the reaction coefficient.

The application of the coordinate transformation (3.25) to each variable yields to the following modified equation

$$\begin{cases}\dfrac{\partial \tilde{u}}{\partial t}+a\dfrac{d_u(0)-\tilde{x}}{d_u(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}=-k\tilde{u}, & \text{if } \tilde{x}\le d_u(0),\\ \dfrac{\partial \tilde{u}}{\partial t}+a\dfrac{\tilde{x}-d_u(0)}{L-d_u(t)}\dfrac{\partial \tilde{u}}{\partial \tilde{x}}=-k\tilde{u}, & \text{if } \tilde{x}> d_u(0),\end{cases}$$

$$\begin{cases}\dfrac{\partial \tilde{v}}{\partial t}+a\dfrac{d_v(0)-\tilde{x}}{d_v(t)}\dfrac{\partial \tilde{v}}{\partial \tilde{x}}=+k\tilde{u}, & \text{if } \tilde{x}\le d_v(0),\\ \dfrac{\partial \tilde{v}}{\partial t}+a\dfrac{\tilde{x}-d_v(0)}{L-d_v(t)}\dfrac{\partial \tilde{v}}{\partial \tilde{x}}=+k\tilde{u}, & \text{if } \tilde{x}> d_v(0),\end{cases}$$

where $d_u(t)$ and $d_v(t)$ are the characteristic lines for each equation

$$d_u(t)=d_u(0)+at,\quad d_v(t)=d_v(0)+at.$$

The CTFOM reads

$$\begin{cases}\tilde{u}_i^{n+1}=\tilde{u}_i^n-\dfrac{\Delta t}{\Delta\tilde{x}}\left[\dfrac{d_u^0}{d_u^n}\left(\tilde{f}_{i+1/2}^{n,-,*}-\tilde{f}_{i-1/2}^{n,+,*}\right)-\dfrac{1}{d_u^n}\left(\tilde{f}_{i+1/2}^{n,-,**}-\tilde{f}_{i-1/2}^{n,+,**}\right)\right]-k\tilde{u}_i^n, \text{ if } \tilde{x}_i\le d_u^0,\\ \tilde{u}_i^{n+1}=\tilde{u}_i^n-\dfrac{\Delta t}{\Delta\tilde{x}}\left[\dfrac{L-d_u^0}{L-d_u^n}\left(\tilde{f}_{i+1/2}^{n,-,*}-\tilde{f}_{i-1/2}^{n,+,*}\right)-\dfrac{1}{L-d_u^n}\left(\tilde{f}_{i+1/2}^{n,-,**}-\tilde{f}_{i-1/2}^{n,+,**}\right)\right]-k\tilde{u}_i^n, \text{ if } \tilde{x}_i> d_u^0,\end{cases}$$

$$\begin{cases}\tilde{v}_i^{n+1}=\tilde{v}_i^n-\dfrac{\Delta t}{\Delta\tilde{x}}\left[\dfrac{d_v^0}{d_v^n}\left(\tilde{g}_{i+1/2}^{n,-,*}-\tilde{g}_{i-1/2}^{n,+,*}\right)-\dfrac{1}{d_v^n}\left(\tilde{g}_{i+1/2}^{n,-,**}-\tilde{g}_{i-1/2}^{n,+,**}\right)\right]+k\tilde{u}_i^n, \text{ if } \tilde{x}_i\le d_v^0,\\ \tilde{v}_i^{n+1}=\tilde{v}_i^n-\dfrac{\Delta t}{\Delta\tilde{x}}\left[\dfrac{L-d_v^0}{L-d_v^n}\left(\tilde{g}_{i+1/2}^{n,-,*}-\tilde{g}_{i-1/2}^{n,+,*}\right)-\dfrac{1}{L-d_v^n}\left(\tilde{g}_{i+1/2}^{n,-,**}-\tilde{g}_{i-1/2}^{n,+,**}\right)\right]+k\tilde{u}_i^n, \text{ if } \tilde{x}_i> d_v^0,\end{cases}$$

The Galerkin method is applied here to both variables

$$\tilde{u}_i^n=\sum_{k=1}^{M_{\text{POD}}}\hat{u}_k\phi_{i,k},\quad \tilde{v}_i^n=\sum_{k=1}^{M_{\text{POD}}}\hat{v}_k\varphi_{i,k};\tag{3.35}$$

so that, the CTROM of this system of equations is

$$\hat{u}_p^{n+1} = \hat{u}_p^n + \frac{\Delta t}{\Delta \tilde{x}} \frac{1}{d_u^n} \sum_{k=1}^{M_{\text{POD}}} \left(A_u^L\right)_{pk} \hat{u}_k^n + \frac{\Delta t}{\Delta \tilde{x}} \frac{1}{L - d_u^n} \sum_{k=1}^{M_{\text{POD}}} \left(A_u^R\right)_{pk} \hat{u}_k^n - k\hat{u}_p^n,$$
$$\hat{v}_p^{n+1} = \hat{v}_p^n + \frac{\Delta t}{\Delta \tilde{x}} \frac{1}{d_v^n} \sum_{k=1}^{M_{\text{POD}}} \left(A_v^L\right)_{pk} \hat{v}_k^n + \frac{\Delta t}{\Delta \tilde{x}} \frac{1}{L - d_v^n} \sum_{k=1}^{M_{\text{POD}}} \left(A_v^R\right)_{pk} \hat{v}_k^n + k \sum_{k=1}^{M_{\text{POD}}} B_{kp} \hat{u}_p^n,$$

where the coefficients are

$$\begin{aligned}
\left(A_u^L\right)_{pk} = & -d_u^0 \frac{1}{2} a^n \left[ \left(\phi_{2,k} - \phi_{1,k}\right) \phi_{1,p} + \sum_{i=2}^{I_d} \left(\phi_{i+1,k} - \phi_{i-1,k}\right) \phi_{i,p} \right] \\
& + d_u^0 \frac{1}{2} \left|a^n\right| \left[ \left(\phi_{2,k} - \phi_{1,k}\right) \phi_{1,p} + \sum_{i=2}^{I_d} \left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right) \phi_{i,p} \right] \\
& + \frac{1}{2} \left(a - |a|\right) \sum_{i=1}^{I_d} \tilde{x}_{i+1/2} \left(\phi_{i+1,k} - \phi_{i,k}\right) \phi_{i,p} \\
& + \frac{1}{2} \left(a + |a|\right) \sum_{i=2}^{I_d} \tilde{x}_{i-1/2} \left(\phi_{i,k} - \phi_{i-1,k}\right) \phi_{i,p}, \\
\left(A_u^R\right)_{pk} = & -\left(L - d_u^0\right) \frac{1}{2} a^n \left[ \sum_{i=I_d+1}^{I_x-1} \left(\phi_{i+1,k} - \phi_{i-1,k}\right) \phi_{i,p} + \left(\phi_{I_x,k} - \phi_{I_x-1,k}\right) \phi_{I_x,p} \right] \\
& + \left(L - d_u^0\right) \frac{1}{2} \left|a^n\right| \left[ \sum_{i=I_d+1}^{I_x-1} \left(\phi_{i+1,k} - 2\phi_{i,k} + \phi_{i-1,k}\right) \phi_{j,p} - \left(\phi_{I_x,k} - \phi_{I_x-1,k}\right) \phi_{I_x,p} \right] \\
& + \frac{1}{2} \left(a - |a|\right) \sum_{i=I_d+1}^{I_x-1} \left(L - \tilde{x}_{i+1/2}\right) \left(\phi_{i+1,k} - \phi_{i,k}\right) \phi_{i,p} \\
& + \frac{1}{2} \left(a + |a|\right) \sum_{i=I_d+1}^{I_x} \left(L - \tilde{x}_{i-1/2}\right) \left(\phi_{i,k} - \phi_{i-1,k}\right) \phi_{i,p}, \\
\left(A_v^L\right)_{pk} = & -d_v^0 \frac{1}{2} a^n \left[ \left(\varphi_{2,k} - \varphi_{1,k}\right) \varphi_{1,p} + \sum_{i=2}^{I_d} \left(\varphi_{i+1,k} - \varphi_{i-1,k}\right) \varphi_{i,p} \right] \\
& + d_v^0 \frac{1}{2} \left|a^n\right| \left[ \left(\varphi_{2,k} - \varphi_{1,k}\right) \varphi_{1,p} + \sum_{i=2}^{I_d} \left(\varphi_{i+1,k} - 2\varphi_{i,k} + \varphi_{i-1,k}\right) \varphi_{i,p} \right] \\
& + \frac{1}{2} \left(a - |a|\right) \sum_{i=1}^{I_d} \tilde{x}_{i+1/2} \left(\varphi_{i+1,k} - \varphi_{i,k}\right) \varphi_{i,p} \\
& + \frac{1}{2} \left(a + |a|\right) \sum_{i=2}^{I_d} \tilde{x}_{i-1/2} \left(\varphi_{i,k} - \varphi_{i-1,k}\right) \varphi_{i,p},
\end{aligned}$$

$$
\begin{aligned}
\left(A_v^R\right)_{pk} = &- \left(L - d_v^0\right) \frac{1}{2} a^n \left[ \sum_{i=I_d+1}^{I_x-1} \left(\varphi_{i+1,k} - \varphi_{i-1,k}\right) \varphi_{i,p} + \left(\varphi_{I_x,k} - \varphi_{I_x-1,k}\right) \varphi_{I_x,p} \right] \\
&+ \left(L - d_v^0\right) \frac{1}{2} \left|a^n\right| \left[ \sum_{i=I_d+1}^{I_x-1} \left(\varphi_{i+1,k} - 2\varphi_{i,k} + \varphi_{i-1,k}\right) \varphi_{j,p} - \left(\varphi_{I_x,k} - \varphi_{I_x-1,k}\right) \varphi_{I_x,p} \right] \\
&+ \frac{1}{2} \left(a - |a|\right) \sum_{i=I_d+1}^{I_x-1} \left(L - \tilde{x}_{i+1/2}\right) \left(\varphi_{i+1,k} - \varphi_{i,k}\right) \varphi_{i,p} \\
&+ \frac{1}{2} \left(a + |a|\right) \sum_{i=I_d+1}^{I_x} \left(L - \tilde{x}_{i-1/2}\right) \left(\varphi_{i,k} - \varphi_{i-1,k}\right) \varphi_{i,p}, \\
B_{pk} = &\sum_{i=1}^{I_x} \phi_{i,k} \varphi_{i,p},
\end{aligned}
$$

where $I_d$ is the position of the adjacent cell to $\tilde{x}_{I_d+1/2} = d(0)$.

**Test case 12. Reactive transport of sinusoidal ICs**

The time-space domain of the case is defined as $(x,t) \in [0,10] \times [0,10]$. The initial sinusoidal profiles are defined as

$$
\begin{aligned}
u(x,0) &= \begin{cases} 0, & \text{if } 0 < x \leq 0.3, \\ \sin\left(\dfrac{2\pi}{L}\left(x - 0.3\right)\right), & \text{if } 0.3 < x < 5.3, \\ 0, & \text{if } 5.3 \leq x < 10, \end{cases} \\
v(x,0) &= \begin{cases} 1, & \text{if } 0 < x \leq 0.3, \\ 1 - \sin\left(\dfrac{2\pi}{L}\left(x - 0.3\right)\right), & \text{if } 0.3 < x < 5.3, \\ 1, & \text{if } 5.3 \leq x < 10, \end{cases}
\end{aligned}
\tag{3.36}
$$

Free BCs are imposed on both domains.

In this case, $d_u(0) = d_v(0) = 3$. The time evolution of both characteristic curves can be seen in Figure 3.39.

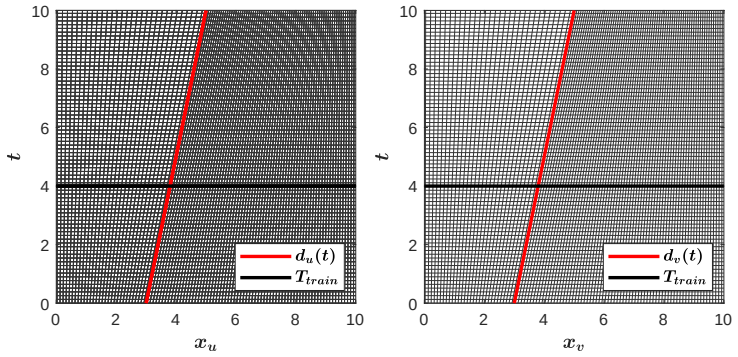
The spatial domain is divided into $I_x = 200$ volume cells and the time step is dynamically calculated according to (2.5), with CFL = 0.9. The training time is $T_{\text{train}} = 4$ and 20 POD modes are used, as shown in Table 3.30.

| $L$ | $T_{\text{train}}$ | $T$ | $a$ | IC | BCs | $d_u(0)$ | $d_v(0)$ | $I_x$ | CFL | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 3 | 10 | 0.2 | (3.36) | Free | 3 | 3 | 200 | 0.9 | 64 | 157 | 20 |

Table 3.30: TC12. Problem settings.

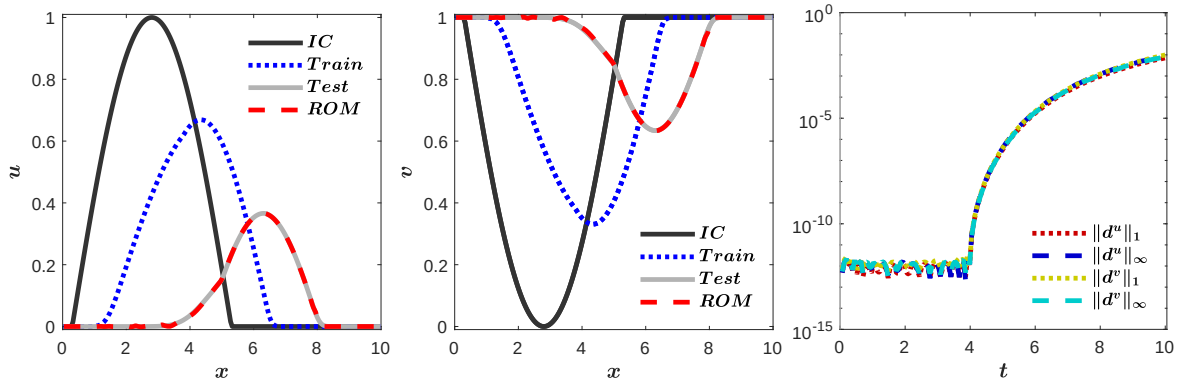
Figure 3.40 shows the IC, the results of the CTROM at the final time $T = 10$ and the result of the CTFOM at $T_{\text{train}} = 4$. A separately calculated CTFOM solution at $T = 10$ is also included for comparison with the CTROM solution. The CTROM accurately predicts the

location and the shape of the solution at the final time $T = 10$, as it is also shown in the error plot 3.40c. And, as indicated in Table 3.31, the speed-up achieved with respect to the test solution computed with the FOM is $\times 17$.



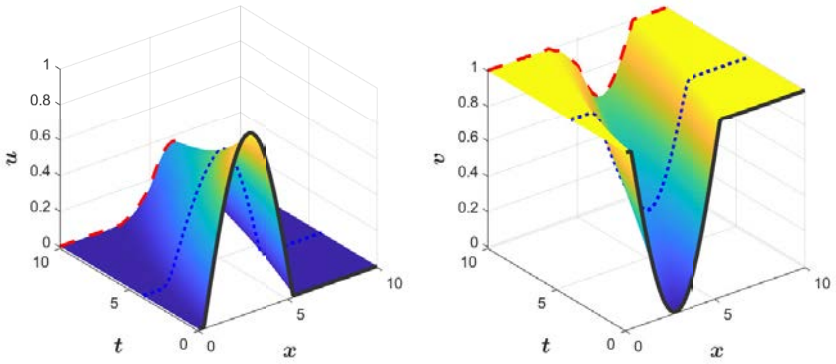

(a) Physical mesh of $u$. (b) Physical mesh of $v$.

Figure 3.39: TC12. Meshes of $u$ and $v$.




(a) Solution of $u$. (b) Solution of $v$. (c) Errors.

Figure 3.40: TC12. Solutions of $u$ and $v$ computed with the CTFOM and the CTROM and errors.




(a) Solution of $u$. (b) Solution of $v$.

Figure 3.41: TC12. 3D solutions of $u$ and $v$: training solution (blue) and ROM solution (red).

| $\|d^u\|_1$ | $\|d^v\|_1$ | $\|d^u\|_\infty$ | $\|d^v\|_\infty$ | |
|---|---|---|---|---|
| $7.21 \cdot 10^{-3}$ | $9.97 \cdot 10^{-3}$ | $8.82 \cdot 10^{-3}$ | $8.24 \cdot 10^{-3}$ | |
| $\tau_{\text{CPU}}^{\text{train}}$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up (train) | Speed-up (test) |
| $2.50 \cdot 10^{-2}$ | $6.05 \cdot 10^{-1}$ | $3.60 \cdot 10^{-2}$ | $\times 1$ | $\times 17$ |

Table 3.31: TC12. Results.

### 3.4.5 2D extension of the CTROM strategy using Radon transform

The CTROM strategy introduced in this work is a genuine 1D method and cannot be directly applied to 2D problems. In this section, the Radon transform is used to extend that strategy to the 2D advection equation (2.13).

This transform was introduced in 1917 by Johann Radon [151] and is an essential tool in medical imaging techniques, such as X-ray computed tomography. It allows the construction of detailed three-dimensional representations of the human body by reconstructing cross-sectional images. It is a very versatile mathematical tool and has many other applications, among others, in astronomy, crystallography, electron microscopy, geophysics, optics, and material science.

Here, the Radon transform is used to reduce (2.13) to a set of one-dimensional problems and each of them is approximated with the CTROM strategy. Similarly to the computed tomography, the obtained solutions of the 1D problems are used to reconstruct the solution of the 2D problem.

The Radon transform is based on the parametrization of the points along any straight line $L$ with respect to the arc length $z$ as

$$\begin{aligned} x(z) &= s\cos\alpha - z\sin\alpha \\ y(z) &= s\sin\alpha + z\cos\alpha, \end{aligned}$$

where $s$ is the distance from $L$ to the origin and $\alpha$ is the angle between $L$ and the $y$-axis [40, Section 2.2].

The Radon transform of a function $f(x, y)$ is given by the integral of $f$ along the line $L$

$$\mathcal{R}f(\alpha, s) = \int_{-\infty}^{+\infty} f(x(z), y(z))dz,$$

as shown in Figure 3.42.

The Radon transform is also called *sinogram*, which defined as a collection of projections $s$ at different angles $\alpha$.

The intertwining property of the Radon transform is of particular interest for the objective of this section. The Radon transform allows to intertwine a partial derivative with a univariate derivative as follows [40, Section 3.6]

$$\mathcal{R}\left\{\frac{\partial f}{\partial x}\right\} = \cos\alpha\,\frac{\partial \mathcal{R}f}{\partial s}, \quad \mathcal{R}\left\{\frac{\partial f}{\partial y}\right\} = \sin\alpha\,\frac{\partial \mathcal{R}f}{\partial s}. \tag{3.37}$$

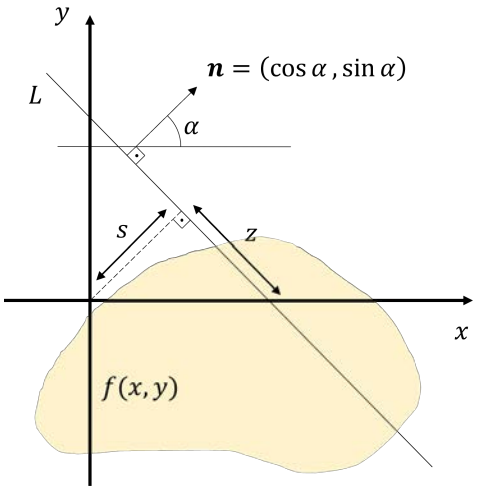



Figure 3.42: Radon transformation. Application $f$ in the $(x, y)$ domain to $f$ in the $(\alpha, s)$ domain.

Consider now equation (2.13) in $(x, y, t) \in (-L, L) \times (-L, L) \times (0, T]$. By applying the intertwining property (3.37) of the Radon transform to (2.13), the following set of one-dimensional problems is obtained

$$\frac{\partial \mathcal{R}u}{\partial t} + \hat{\lambda}\frac{\partial \mathcal{R}u}{\partial s} = 0, \quad (s, t) \in (-L, L) \times (0, T]\,, \tag{3.38}$$

where $\hat{\lambda} = a_x \cos\alpha + a_y \sin\alpha$, for $\alpha \in (0, \pi)$ [156].

The 1D CTROM (3.29) strategy can therefore be used to predict the time evolution of the 2D hyperbolic problem (2.13). To do this, first, the IC is transformed from the physical space into the Radon domain, i.e., the $(s, \alpha)$ domain. Then, the CTROM strategy described in previous sections is applied to (3.38) for a discrete collection of values of $\alpha \in (0, \pi)$. Finally, the solution in the $(s, \alpha)$ domain is transformed into the physical space using a filtered back-projection inversion formula for the Radon transform [154, 156].

**Test case 13. 2D advection of an initial Gaussian profile**

Consider the problem (2.13) with $a_x = a_y = 1$. The time-space domain of the case is defined as $(x, y, t) \in [0, 10] \times [0, 10] \times [0, 2]$. The initial Gaussian profile is defined as

$$u(x, y, 0) = e^{-\frac{x^2+y^2}{2}}, \tag{3.39}$$

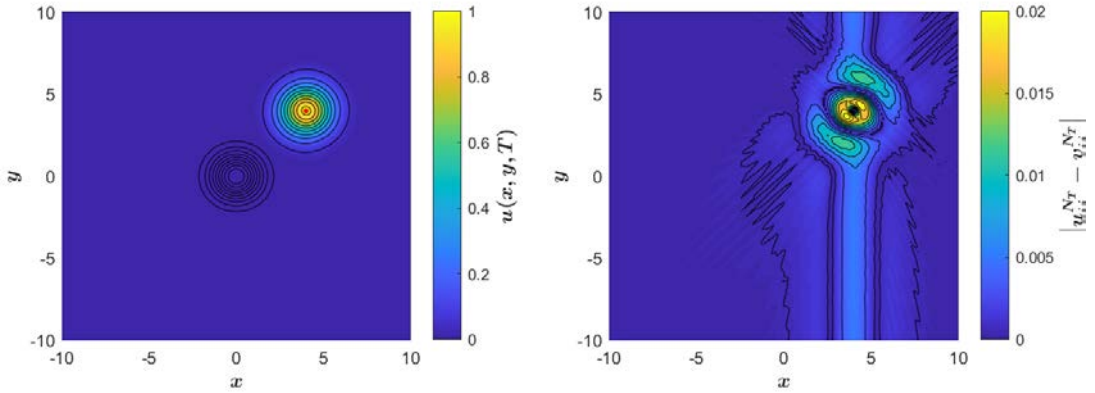
Free BCs are imposed in all the boundary. The 1D intervals $[-10, 10]$ and $[0, 180]$ obtained by applying the Radon transform (3.37) are uniformly divided into 200 subintervals. In this case a uniform time step size $\Delta t = 0.05$ is used to approximate the set of problems (3.38).

Training solutions are computed with the 1D CTFOM (3.27) up to $T_{\text{train}} = 0.5$. With these data, new solutions are computed using the 1D CTROM (3.29) up to $T = 2$ and $M_{\text{POD}} = 6$.

| $L_x \times L_y$ | $T_{\text{train}}$ | $T$ | $a_x$ | $a_y$ | IC | BCs | $d_x(0)$ | $d_y(0)$ | $I_x$ | $\Delta t$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $10 \times 10$ | 0.5 | 2 | 1 | 1 | Eq. (3.39) | Free | 0 | 0 | 200 | 0.05 | 6 |

Table 3.32: TC13. Problem settings.

The numerical solution provided by the CTROM at $T = 2$ is shown in Figure 3.43a. The position of the center of the CTFOM solution is marked by a red dot and the IC is also shown. The differences shown in Figure 3.43b have a maximum value of 0.0217, i.e. two orders of magnitude smaller than the maximum amplitude of the solution. Figure 3.44 shows the sinogram of the numerical solution provided by the CTROM, i.e., the solution in the $(s, \alpha)$ plane. The contour line corresponding to the maximum value of the sinogram of the CTFOM solution is also depicted using a red line, showing a good agreement between both solutions.




(a) Solution of the CT-ROM. (b) Differences.

Figure 3.43: TC13. CTROM solution, its differences with the CTFOM at $T = 2$.

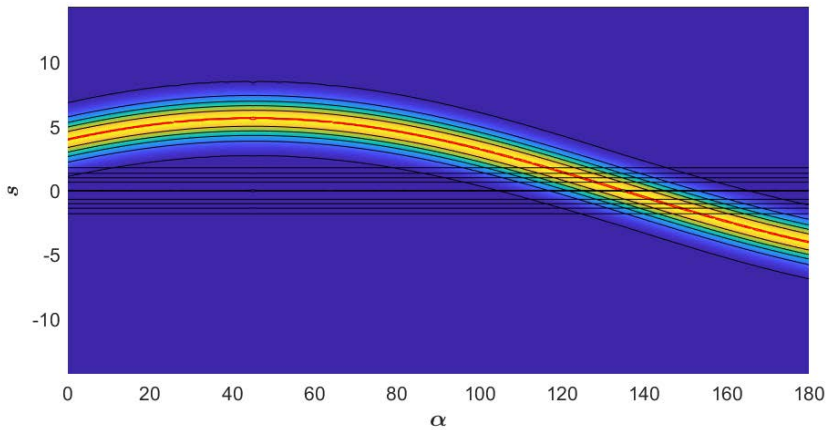



Figure 3.44: TC13. Sinogram at $T = 2$.

## 3.5 Concluding remarks

In this the first chapter the strategy for solving intrusive POD-based ROMs method applied to the 1D and 2D linear advection-diffusion equations has been introduced. The basic aspects of ROMs have been analysed by means of examples leading to the following conclusions. They have shown satisfactory efficiency results, measured as the agreement between the precision of the solution and the acceleration in the calculation times achieved with respect to that of the training solution, being capable of obtaining high-precision solutions with calculation accelerations of an order of magnitude with respect to the FOM. Additionally, ROMs are able to solve time dependent Dirichlet-type BCs satisfactorily. Furthermore, the CPU time accelerations achieved by the ROM have increased to 3 orders of magnitude in the 2D test case, and it can be concluded that ROMs are more efficient the larger the problem of interest. Finally, the development of a high-order ROM based on the Lax-Wendroff method and a first-order ROM based on the Lax-Friedrichs method has been introduced. It has been proved that a ROM developed from a high order scheme can calculate solutions that maintain this high order even if it has been trained with lower order solutions.

A methodology has been proposed to solve with ROMs parametrized problems with other parameter values than those with which they have been trained. It is based on generating training samples calculated with different values of the input parameters. With this, ROMs are capable of predicting different values of the advection velocity and the diffusion coefficient and achieving solutions with high accuracy and even speeding up the computations required by the FOM by an order of magnitude. Other input parameters accurately solved are the coefficients defining the initial Gaussian profile and discontinuity and different time-dependent BCs.

And a methodology has been proposed to overcome the limitation of the standard ROM strategy that prevents them from solving problems beyond the training time. This is possible thanks to a coordinate transformation that allows equations to be solved in a transformed domain that is aligned with an initially defined characteristic curve. In this way, the physical domain evolves over time and the ROM can obtain high-precision solutions beyond the training time and even save computational time. It has been applied to different the linear equations and it can be concluded that the CTROM is very useful in predicting solutions beyond the calculation time when solving linear equations with good accuracy. It has also even been shown to speed up the times required by the FOM to calculate training solutions.

# Chapter 4

# ROMs and non-linear problems

Having shown the accuracy and the efficiency of ROMs in their application to linear equations, it is now necessary to address their extension to nonlinear problems. In order to develop ROMs based on the SWE, the Burgers equation (2.15) is first considered, since it condenses the main features of the nonlinearity into a single equation.

In Section 4.1, the theoretical basis of the PID methodology based on splitting the time variable in disjoint subintervals (called *time windows*) is presented together with the linearization of ROMs, along with a series of test cases that serve to justify it in terms of efficiency.

The definition of time windows makes it impossible to use the a priori criterion (3.1) to choose the number of POD modes. In addition to this, the number of time windows has to be chosen. Because of all this, an a posteriori selection criterion of POD modes and time windows is proposed in this thesis based on the efficiency achieved by the ROM.

Sections 4.2 and 4.3 are the continuation of the sections presented in the previous chapter. They extend the techniques of time and parameter prediction presented earlier to non-linear problems. It should be noted that, time prediction does not need the use of the PID, since just one time window is enough to obtain satisfactory results. Therefore, the test cases presented in this particular section do not make use of time windows.

## 4.1 Why and how to approximate non-linearity with linearized ROMs

The resolution of the main features of non-linearity, which can be summarised as the generation of shock waves and rarefaction, is not trivial when using ROMs. As will be seen below, it is possible to develop ROMs from nonlinear FOMs that are also nonlinear. However, the increase in complexity also means an increase in the computational cost of solving them.

It is therefore interesting to look for alternatives to speed up the resolution of ROMs based on non-linear models. A strategy for this purpose is presented in the literature: the PID [202], according to which time windows are defined and the snapshot matrix is accordingly divided into sub-matrices. This involves defining a new reduced space for each time window and, by renewing the basis functions that define the reduced spaces, the nonlinearity can be approximated successfully.

The Burgers equation (2.15) can be used as a simpler model of many SWE features. At the same time, it provides a simple understanding of the reasons for using linearized ROMs of non-linear FOMs instead of non-linear ROMs (if these are possible).

This first section justifies why it is necessary to linearize ROMs of nonlinear problems. There are two main reasons why this is necessary. The first one, as shown by means of a couple of test cases, is the gain in efficiency that this entails, since the linearized ROM is faster due to its lower complexity, and without loss of accuracy. And the second reason, which is related to the development of the ROM itself, is that it requires an approximation to overcome some of the impediments presented by non-linear FOMs (if they can be overcome, which is not the case for some FOMs applied to the SWEs). Both reasons are well developed in what follows.

Likewise, the methodology used to carry out the linearization of the ROM is also explained, based on the construction of time windows from the snapshot matrix, in accordance with the PID method [86, 164, 202]. This involves a modification of the standard ROM strategy presented in Chapter 3, which will be duly detailed, and which is also used in Chapter 5 for the SWEs.

### 4.1.1 Development of a fully non-linear ROM

In the Burgers equation (2.15), the transport velocity is the conserved variable itself. Therefore, when applying the Galerkin method (3.2) to the FOM of the Burgers equation (2.17), it is necessary to take into account that absolute values of this variable are present. Thus, using the triangular inequality, the following is obtained

$$|u_i^n| = |\sum_{k=1}^{M_{\text{POD}}} \hat{u}_k^n \phi_{i,k}| \leq \sum_{k=1}^{M_{\text{POD}}} |\hat{u}_k^n \phi_{i,k}| = \sum_{k=1}^{M_{\text{POD}}} |\hat{u}_k^n||\phi_{i,k}|; \tag{4.1}$$

and in order to develop the fully non-linear ROM, the last term of (4.1) is used in this approach, even though it would lead to unavoidable errors in the solutions. Following the steps indicated in Section 3.1.2, the non-linear ROM reads

$$\begin{aligned} \hat{u}_p^{n+1} = &\sum_{i \in J^D} (u_0)_i^n \, \phi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} \hat{u}_k^n A_{pk} + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} \hat{u}_k^n \sum_{q=1}^{M_{\text{POD}}} \hat{u}_q^n B_{pkq} \\ &+ \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} |\hat{u}_k^n| \sum_{q=1}^{M_{\text{POD}}} \hat{u}_q^n C_{pkq} + \frac{\Delta t}{\Delta x} \sum_{q=1}^{M_{\text{POD}}} \hat{u}_q^n D_{pk} + \frac{\Delta t}{\Delta x} \sum_{q=1}^{M_{\text{POD}}} |\hat{u}_q^n| E_{pk}, \end{aligned} \tag{4.2}$$

where the coefficients are

$$A_{pk} = \sum_{i \in J^I \cup J^S} \phi_{i,q} \phi_{i,p},$$

$$B_{pkq} = \sum_{i \in J^S} b_i \phi_{i,p} - \frac{1}{4} \sum_{i \in J^I} \left[ (\phi_{i+1,k} + \phi_{i,k}) (\phi_{i+1,q} - \phi_{i,q}) + (\phi_{i,k} + \phi_{i-1,k}) (\phi_{i,q} - \phi_{i-1,q}) \right] \phi_{i,p},$$

$$C_{pkq} = \sum_{i \in J^S} c_i \phi_{i,p} + \frac{1}{4} \sum_{i \in J^I} \left[ |\phi_{i+1,k} + \phi_{i,k}| \left( \phi_{i+1,q} - \phi_{i,q} \right) - |\phi_{i,k} + \phi_{i-1,k}| \left( \phi_{i,q} - \phi_{i-1,q} \right) \right] \phi_{i,p},$$

$$D_{pk} = \sum_{i \in J^S} d_i \phi_{i,p} - \frac{1}{4} \sum_{i \in J^I} \left[ \left( \phi_{i+1,q} + \phi_{i,q} \right) \left( z_{i+1} - z_i \right) + \left( \phi_{i,q} + \phi_{i-1,q} \right) \left( z_i - z_{i-1} \right) \right] \phi_{i,p},$$

$$E_{pk} = \sum_{i \in J^S} e_i \phi_{i,p} + \frac{1}{4} \sum_{i \in J^I} \left[ |\phi_{i+1,q} + \phi_{i,q}| \left( z_{i+1} - z_i \right) - |\phi_{i,q} + \phi_{i-1,q}| \left( z_i - z_{i-1} \right) \right] \phi_{i,p}.$$

The coefficients $b, c, d, e$ depend on the type of BCs imposed and can be easily calculated as shown in Section 3.1.2 in Equation (3.9). It should be noted that, in this case, there are coefficients with three subindices, $B_{pkq}$ and $C_{pkq}$, which is due to the presence of the products of the variable conserved by itself, i.e. due to the non-linearity of the Burgers equation.

**Test case 14. Non-linear transport of an initial Gaussian profile**

This test case serves to illustrate the weak efficiency obtained by the non-linear ROM (4.2) applied to the 1D Burgers equation (2.15). For this purpose, the evolution in space and time of an initial Gaussian profile is solved, so that, from a smooth function, a shock wave and a rarefaction wave are generated, with their consequent discontinuity.

The time-space domain of the case is defined as $(x, t) \in [0, 20] \times [0, 5]$. Initially, the Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-(x-6)^2}. \tag{4.3}$$

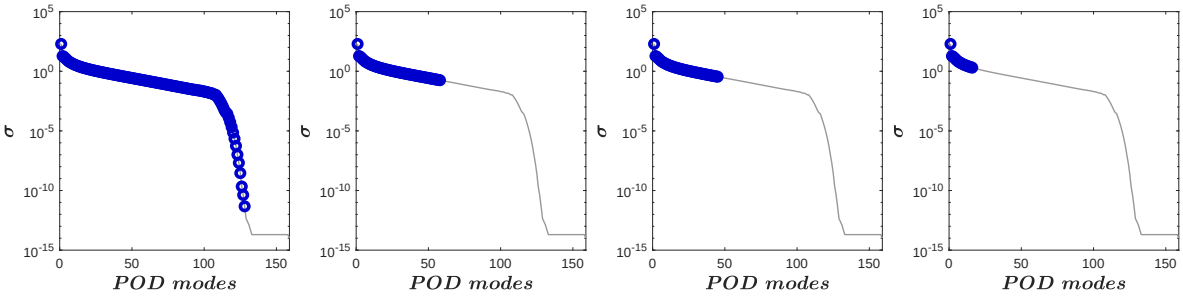
No bed elevation is set, $z(x) = 0$, $0 \leq x \leq 20$ and free boundary conditions are considered.

The spatial domain is discretized in $I_x = 200$ volume cells and the CFL number is set to 0.9. Table 4.1 shows the different subcases considered by varying the number of POD modes, with the rest of the settings. The POD eigenvalues chosen for each percentage are plotted in Figure 4.1 with blue circles on a grey line that represents the total.

| Subcase | $M_{\text{POD}}$ | $P(M_{\text{POD}})$ |
|---|---|---|
| 1 | 128 | 100 |
| 2 | 58 | 99 |
| 3 | 45 | 98 |
| 4 | 16 | 90 |
| $L = 20$, $T = 5$, IC: Eq. (4.3), BCs: free | | |
| $I_x = 200$, CFL $= 0.9$, $N_T = 159$ | | |

Table 4.1: TC14. Problem settings.

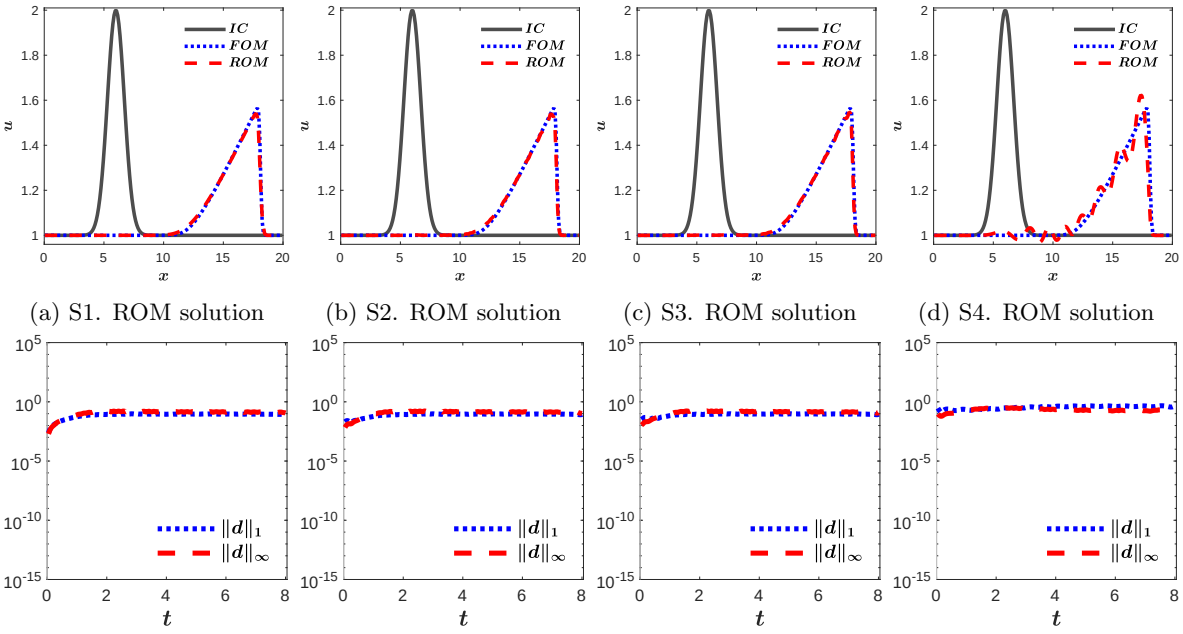
Figure 4.2a shows the solutions computed by the non-linear ROM in the different subcases. Although the ROM solves for the optimal number of modes which is 100% of the information according to the POD, it is not able to recover the training solution with machine accuracy, as shown in Table 4.2. and in Figure 4.2e, where the differences are plotted.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3. (d) Subcase 4.

Figure 4.1: TC14. POD eigenvalues.

As the number of POD modes decreases, as could be expected, the accuracy gets worse, as small oscillations around the solution profile appear (see Fig. 4.2d).




(a) S1. ROM solution (b) S2. ROM solution (c) S3. ROM solution (d) S4. ROM solution

(e) S1. Differences (f) S2. Differences (g) S3. Differences (h) S4. Differences

Figure 4.2: TC14. ROM solutions and differences.

The CPU time required by the FOM to complete this computation is much smaller than those of the ROM for all subcases, as shown in Table 4.2. That is why the speed-up is always zero. This is due to the fact that the ROM is also non-linear and it needs a big computational effort to compute the solution with the 3D coefficients from (4.2). The speed-ups achieved by the ROM suggest the need to tackle the construction of ROMs based on non-linear equations by following a different strategy to increase their efficiency.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d\|_1$ | $8.81 \cdot 10^{-2}$ | $8.81 \cdot 10^{-2}$ | $9.11 \cdot 10^{-2}$ | $4.48 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $1.14 \cdot 10^{-1}$ | $1.03 \cdot 10^{-1}$ | $1.122 \cdot 10^{-1}$ | $2.69 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $5.45 \cdot 10^{-4}$ | $5.45 \cdot 10^{-4}$ | $5.45 \cdot 10^{-4}$ | $5.45 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | 2.47 | $2.12 \cdot 10^{-1}$ | $9.73 \cdot 10^{-2}$ | $4.70 \cdot 10^{-3}$ |
| Speed-up | $\times 0$ | $\times 0$ | $\times 0$ | $\times 0$ |

Table 4.2: TC14. Efficiency results based on differences and CPU times.

### 4.1.2 PID method

As proposed in [86, 164, 202], PID faces and overcomes the development and resolution of ROMs based on non-linear problems. This modification of the standard ROM strategy allows the linearization of the ROM to speed-up computational times at the same time that enables the generation of shocks and rarefaction waves in the solutions computed by the ROM.

In this strategy, the total simulation time $T$ is divided into $M_W$ non-overlapping time windows

$$\left[0, t^{N_{w_1}}\right] \cup \left[t^{N_{w_1}+1}, t^{N_{w_2}}\right] \cup \cdots \cup \left[t^{N_{M_W-1}+1}, t^{N_{M_W}}\right],$$

where $0 < t^{N_{w_1}} < t^{N_{w_2}} < \cdots < t^{N_{M_W}} = t^{N_T} = T$.

The training solutions computed with the FOM are stored in $M_W$ different snapshot submatrices according to the definition of the time windows

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \cdots & \mathbf{U}_{M_W} \end{pmatrix},$$

where

$$\mathbf{U}_1 = \begin{pmatrix} u_1^1 & \cdots & u_1^{N_{w_1}} \\ \vdots & \ddots & \vdots \\ u_{I_x}^1 & \cdots & u_{I_x}^{N_{w_1}} \end{pmatrix}, \quad \cdots, \quad \mathbf{U}_{M_W} = \begin{pmatrix} u_1^{N_{M_W-1}+1} & \cdots & u_1^{N_{M_W}} \\ \vdots & \ddots & \vdots \\ u_{I_x}^{N_{M_W-1}+1} & \cdots & u_{I_x}^{N_{M_W}} \end{pmatrix}.$$

The SVD is applied to each snapshot submatrix

$$\mathbf{U}^w = \mathbf{\Phi}^w \mathbf{\Sigma}^w \left(\mathbf{\Psi}^w\right)^T, \ w = 1, ..., M_W;$$

so that, $M_W$ different POD basis are obtained, where the $w$-th basis is

$$\mathbf{\Phi}_{M_{\text{POD}}}^w = \begin{pmatrix} \phi_{1,1}^w & \phi_{1,2}^w & \cdots & \phi_{1,M_{\text{POD}}}^w \\ \phi_{2,1}^w & \phi_{2,2}^w & \cdots & \phi_{2,M_{\text{POD}}}^w \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{I_x,1}^w & \phi_{I_x,2}^w & \cdots & \phi_{I_x,M_{\text{POD}}}^w \end{pmatrix}$$

The basis of each window defines a different reduced space. The PID strategy states that, as shown in Figure 4.3, once the solution exceeds the limit of one reduced space, it is projected back

to the physical space and the projected within the following reduced space. All these projections that are needed to change from one time window to another, slow down the computation of the ROM, as the solution needs to be returned to the physical space, where it returns to its largest size.

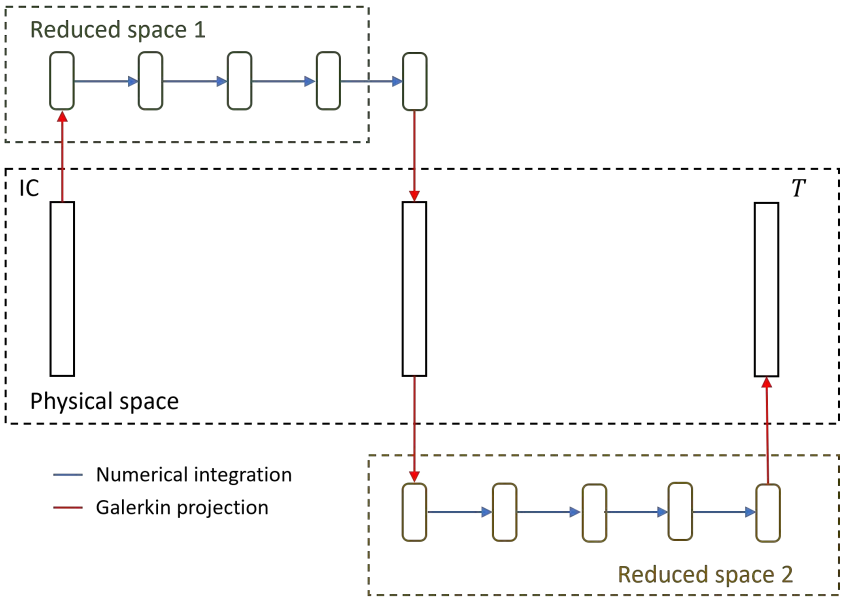



Figure 4.3: PID-based ROM strategy.

The number of POD modes remains constant throughout all time windows. However, there is the possibility of defining the number of POD modes variable according to the evolution of the solution, as proposed in [7].

### 4.1.3 Development of a linearized ROM

The linearisation of the ROM can be qualitatively understood as an approximation of the resolution of the linear equation in which the advection velocity is constant in each time window. Thus, if the time windows are sufficiently small, the velocity varies in time allowing to emulate the non-linear transport of the Burgers equation.

In this way, the application of the PID to develop a linearized version of the ROM of the Burgers equation consists of the averaging of the velocity within the $w$-th time window $[t^{N_{w_1}+1}, t^{N_{w+1}}]$

$$\bar{u}_i^w = \frac{1}{N_{w+1} - N_w} \sum_{n=M_W+1}^{N_{w+1}} u_i^n,$$

where $N^{w+1} - N^w$ is the number of time steps contained within the $w$-th time window, and $N^w + 1$ and $N^{w+1}$ are the initial and final snapshots of the $w$-th time window. The number of time steps per time window is considered constant in most of the test cases in this thesis. In a couple of occasions included in the final results, non-uniform windows are defined in order

not to overtrain the ROM at times when it is not necessary (for example, when the solution reaches a steady state).

The linearized FOU-based ROM of the 1D Burgers equation is

$$\hat{u}_p^{n+1} = \sum_{i\in J^D} (u_0)_i^n \, \phi_{i,p} + \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n A_{pk} + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\mathrm{POD}}} \hat{u}_k^n B_{pk} + \frac{\Delta t}{\Delta x} C_p, \tag{4.4}$$

where the coefficients are

$$\begin{aligned}
A_{pk} &= \sum_{i\in J^I \cup J^S} \phi_{i,q}\phi_{i,p}, \\
B_{pk} &= \sum_{i\in J^S} b_i \phi_{i,p} - \frac{1}{2}\sum_{i\in J^I} \left[\tilde{u}^w_{i+1/2}\left(\phi_{i+1,k}-\phi_{i,k}\right)\phi_{i,p} + \tilde{u}^w_{i-1/2}\left(\phi_{i,k}-\phi_{i-1,k}\right)\phi_{i,p}\right] \\
&\quad + \frac{1}{2}\sum_{i\in J^I} \left[|\tilde{u}^w_{i+1/2}|\left(\phi_{i+1,k}-\phi_{i,k}\right)\phi_{i,p} - |\tilde{u}^w_{i-1/2}|\left(\phi_{i,k}-\phi_{i-1,k}\right)\phi_{i,p}\right], \\
C_p &= \sum_{i\in J^S} c_i \phi_{i,p} - \frac{1}{2}\sum_{i\in J^I} \left[\tilde{u}^w_{i+1/2}\left(z_{i+1}-z_i\right)\phi_{i,p} + \tilde{u}^w_{i-1/2}\left(z_i-z_{i-1}\right)\phi_{i,p}\right] \\
&\quad + \frac{1}{2}\sum_{i\in J^I} \left[|\tilde{u}^w_{i+1/2}|\left(z_{i+1}-z_i\right)\phi_{i,p} - |\tilde{u}^w_{i-1/2}|\left(z_i-z_{i-1}\right)\phi_{i,p}\right],
\end{aligned}$$

with $\tilde{u}^w_{i+1/2} = \left(u^w_{i+1} + u^w_i\right)/2$. It is worth noting that the conserved variable only appears within absolute values in the ROM coefficients, which are calculated during the off-line phase. Thanks to that, the triangular inequality is fulfilled, because the Galerkin method is not applied on the variable in the absolute value. Coefficients $b, c$ are given depending on the BCs, similar to (3.9).

**Test case 15. Linearized transport of an initial Gaussian profile**

This test case reproduces the previous one using the linearized version of the Burgers-based ROM (4.4). It will help to explore the most fundamental consequences of the PID method. For this purpose, three different subcases are defined, all of them solving the linearized ROM (4.4). The rest of the settings of the problem are the same, as it can be seen in Table 4.3.

| Subcase | $M_{\mathrm{POD}}$ | $M_W$ |
|---|---|---|
| 1 | 128 | 1 |
| 2 | 128 | 159 |
| 3 | 200 | 159 |
| $L = 20$, $T = 5$, IC: Eq. (4.3), BCs: free | | |
| $I_x = 200$, CFL $= 0.9$, $N_T = 159$ | | |

Table 4.3: TC15. Problem settings.

It is convenient to make a detailed analysis of the results obtained in each subcase:

- Subcase 1. Just one time window is set and the number of POD modes is the maximum,

according to (3.1).

When using only on time window, the velocity is constant from the begining of the computation until the final time. This implies that the initial Gaussian profile is transported along the $x$-axis with constant velocity and suffers no modification, as can be seen in Figure 4.4a. No shock nor rarefaction can be generated using only one time window. The differences between the FOM and the ROM solutions are big from the begining of the computation, as shown in Figure 4.4d.

- Subcase 2. The numbers of time windows and POD modes are set to be the maximum in both cases, the latter according to (3.1) within each time window.

  If the maximum number of time windows is set $M_W = N_T$ (i.e., the Galerkin projection is used in each time step to change the POD basis), the linearized ROM is able to generate shocks and rarefaction, as shown in Figure 4.4b. However, even though the number of POD modes has been computed according to (3.1) to cover the 100% of the information, the solution does not perfectly match the FOM solution. The error is bigger than machine precision, as shown in Figure 4.4e. The gray vertical lines in the background represent the time levels in which the projections are used. In this subcase, these are all the time levels.

- Subcase 3. The number of time windows is set maximum and the number of POD modes matches the number of volume cells, so it is bigger than the maximum provided by (3.1).

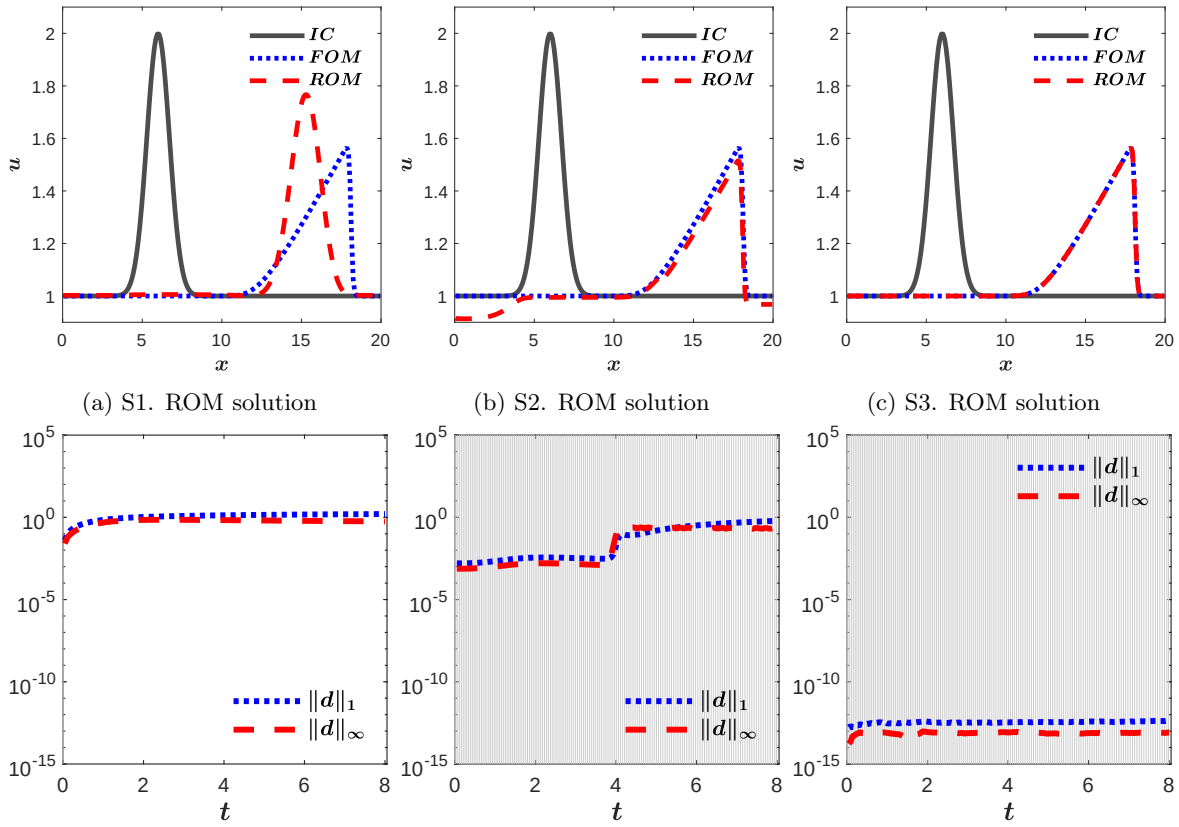
  In this subcase, the linearized ROM (4.4) is able to recover the training solution with machine accuracy, as shown in Figures 4.4c and 4.4f. That is, the actual number of POD modes that allows to accurately recover the training solution is bigger than the maximum proposed by (3.1) within each time window.

Taking all this into account, the percentage criterion (3.1) is no longer useful. It is necessary to select the number of POD modes to solve and the number of time windows based on an a posteriori error criterion. Next test case deals with this issue.

Apart from that, the CPU times required by the linearized ROM in these three subcases, contained in Table 4.4, are higher than those of the FOM, as it is expected, since the number of POD modes is very large. The real speed-up of the linearized ROM will be studied in the next section.

| Subcase | 1 | 2 | 3 |
|---|---|---|---|
| $\|d\|_1$ | 1.58 | $6.13 \cdot 10^{-1}$ | $4.21 \cdot 10^{-13}$ |
| $\|d\|_\infty$ | $5.60 \cdot 10^{-1}$ | $1.17 \cdot 10^{-1}$ | $8.40 \cdot 10^{-13}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $5.45 \cdot 10^{-4}$ | $5.45 \cdot 10^{-4}$ | $5.45 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $8.84 \cdot 10^{-3}$ | $3.44 \cdot 10^{-3}$ | $6.37 \cdot 10^{-3}$ |
| Speed-up | $\times 0$ | $\times 0$ | $\times 0$ |

Table 4.4: TC15. Efficiency results based on differences and CPU times.




(a) S1. ROM solution (b) S2. ROM solution (c) S3. ROM solution

(d) S1. Differences (e) S2. Differences (f) S3. Differences

Figure 4.4: TC15. ROM solutions and differences.

### 4.1.4 A posteriori error/efficiency criterion

As proved in TC15, the a priori criterion (3.1) is not useful as long as the PID method is applied. Furthermore, in addition to the number of POD modes, $M_{\text{POD}}$, it is necessary to determine the number of time windows, $M_W$, with which to solve the ROM. As there is no other a priori criterion in the literature that includes the PID method, an a posteriori one has been proposed.

The following test cases are designed to study how many POD modes and time windows are necessary to achieve good results in terms of efficiency, i.e. a good compromise between the error of the ROM solution relative to that of the FOM and the speed-up achieved by the ROM.

**Test case 16. Transport of an initial discontinuity**

The a posteriori criterion is based on mapping the ROM efficiency for different values of $M_{\text{POD}}$ and $M_W$ and combinations of them. For this purpose, the following values have been used

$$M_{\text{POD}} \in \{1, 5, 10, 15, 20, 25, 30\},$$
$$M_W \in \{1, 2, 5, 10, 21, 42, 82\}.$$

All these combinations are labelled as $7 \times 7 = 49$ different subcases as listed in Table 4.5.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 1 | 5 | 10 | 15 | 20 | 25 | 30 | 1 | 5 | 10 | 15 | 20 | 25 | 30 |
| $M_W$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Subcase | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| $M_{\text{POD}}$ | 1 | 5 | 10 | 15 | 20 | 25 | 30 | 1 | 5 | 10 | 15 | 20 | 25 | 30 |
| $M_W$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 |
| $M_{\text{POD}}$ | 1 | 5 | 10 | 15 | 20 | 25 | 30 | 1 | 5 | 10 | 15 | 20 | 25 | 30 |
| $M_W$ | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 42 | 42 | 42 | 42 | 42 | 42 | 42 |
| Subcase | 43 | 44 | 45 | 46 | 47 | 48 | 49 | | | | | | | |
| $M_{\text{POD}}$ | 1 | 5 | 10 | 15 | 20 | 25 | 30 | | | | | | | |
| $M_W$ | 84 | 84 | 84 | 84 | 84 | 84 | 84 | | | | | | | |

Table 4.5: TC16. Values of $M_{\text{POD}}$ and $M_W$.

The time-space domain of the case is defined as $(x,t) \in [0,40] \times [0,15]$. The spatial domain is discretized using $I_x = 200$ volume cells, and the time steps are computed dynamically according to (2.5), with CFL $= 0.9$. Free boundary conditions are considered. A discontinuity is initially set, defined by

$$u(x,0) = \begin{cases} 2, & \text{if } x \leq 14, \\ 1, & \text{if } x > 14. \end{cases} \tag{4.5}$$

All the settings of this problem are the same for all subcases, as indicated in Table 4.6, except for the values of $M_{\text{POD}}$ and $M_W$.

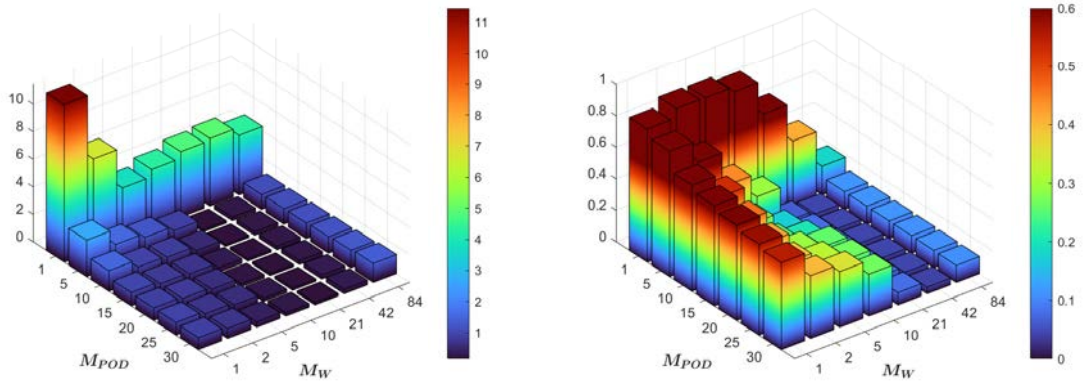
| $L$ | $T$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|
| 40 | 15 | Eq. (4.5) | Free | 200 | 0.9 | 168 | Tab. 4.5 | Tab. 4.5 |

Table 4.6: TC6. Problem settings.

Figures 4.5a and 4.5b contain the differences $\|d\|_1$ and $\|d\|_\infty$ of the ROM solutions for each subcase, having the latter a smaller scale range. It can be seen that in both cases the largest differences are obtained when the number of POD modes is smaller. In fact, for $M_{\text{POD}} = 1$, all values of $M_W$ have similar differences. However, from 5 POD modes onwards, the error decreases considerably and $M_W$ presents notable differences between its values. Moreover, $\|d\|_\infty$ reports that a single time window has very high errors. Both differences decrease with $M_W$, but 84 time windows presents bigger differences than 42 time windows. This is because the Galerkin projection (3.2) when using a non-maximum number of modes also introduces a certain error which, if many time windows are arranged, can considerably worsen the solution.

In addition to that, Figure 4.5c shows the speed-ups achieved by the ROM in each subcase. The biggest speed-ups are achieved when using the smallest number of POD modes, but this means bigger errors. For high values of $M_{\text{POD}}$, the speed-ups have the same order of magnitude

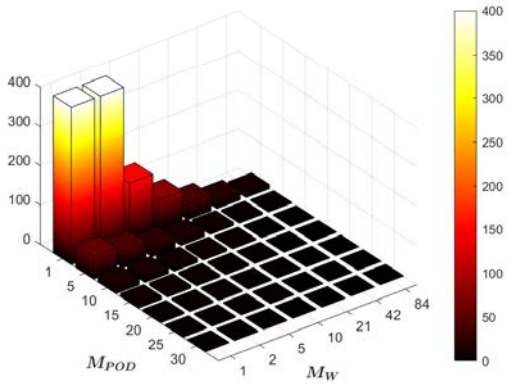
for the different $M_W$ values.




(a) $\|d\|_1$.




(b) $\|d\|_\infty$.




(c) Speed-ups.

Figure 4.5: TC16. ROM solutions differences and speed-ups achieved.

It is important to find the pair $(M_{\text{POD}}, M_W)$ that minimize differences while maximising speed-up. In this way, Figure 4.6 helps to find them by checking the effectiveness of the ROM in a simple way. In this figure, the speed-up is represented in the left $y$-axis against the $\|d\|_1$ differences in the right $y$-axis. The subcases have been ordered according to the magnitude of the differences $\|d\|_1$, in decreasing order.

The 6 blue circles indicate the candidate subcases that have been proposed on the basis of their speedup being the best in their neighbourhood. These subcases range from order $O(1)$ to $O(10^1)$. On the other hand, their differences vary from order $O(10^{-1})$ to order $O(10^1)$, as shown in Table 4.7. In general, it has been found that the optimal number of POD modes lies between 5 and 10, as fewer means loss of accuracy and more slows down calculations. And the number of time windows lies between 2 and 21. As will be seen below, if too few windows are used, it is necessary to increase $M_{\text{POD}}$ and too many windows can slow down the ROM.

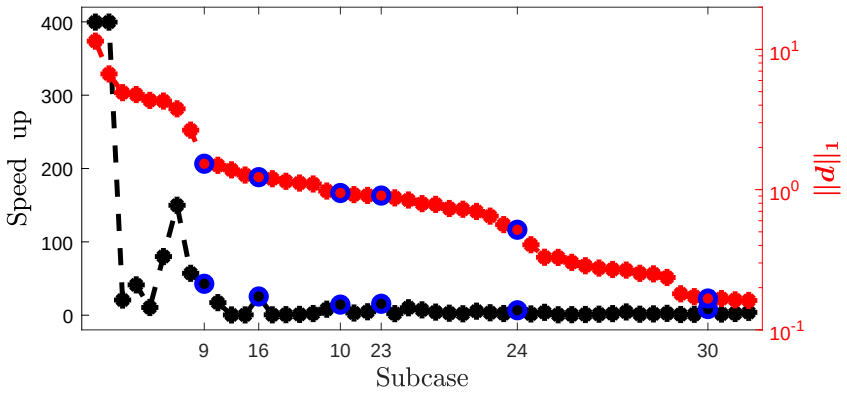
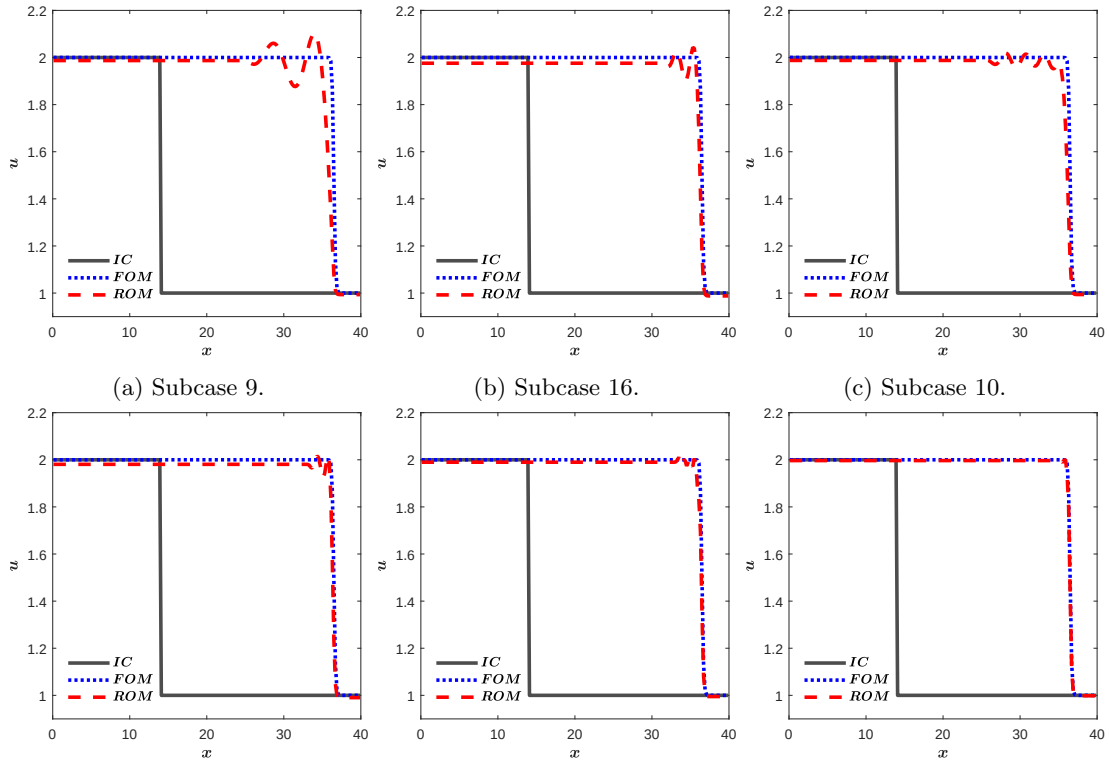



Figure 4.6: TC16. Possible optima values of $M_{\text{POD}}$ and $M_W$.

In order to be able to choose which of them is better, it is also necessary to take into account the behaviour of their solutions. Figure 4.7 shows the solutions of the ROM computed in the candidate subcases. The solution of Subcase 9, despite having the best speed-up, cannot be taken into account, due to the severe oscillations that appear next to the discontinuity. Subcases 16, 10, 23 and 24 show damped oscillations. Subcase 9 would be the worst of them, as its oscillations are larger. And Subcase 24 would be the best of them, were it not for the fact that its speedup is one order of magnitude smaller than the rest and even smaller than Subcase 30, which has no oscillations, but neither of them two are competitive. It could therefore be concluded that, despite the small oscillations, Subcases 10 and 23 are the most efficient.




(a) Subcase 9. (b) Subcase 16. (c) Subcase 10.

(d) Subcase 23. (e) Subcase 24. (f) Subcase 30.

Figure 4.7: TC16. ROM solutions of optima candidates.

| Subcase | 9 | 16 | 10 | 23 | 24 | 30 |
|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 5 | 10 | 5 | 10 | 5 |
| $M_W$ | 2 | 5 | 2 | 10 | 10 | 21 |
| $\|d\|_1$ | 1.53 | 1.23 | $9.46 \cdot 10^{-1}$ | $9.07 \cdot 10^{-1}$ | $5.18 \cdot 10^{-1}$ | $1.67 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $6.59 \cdot 10^{-1}$ | $4.41 \cdot 10^{-1}$ | $5.31 \cdot 10^{-1}$ | $2.95 \cdot 10^{-1}$ | $1.73 \cdot 10^{-1}$ | $6.53 \cdot 10^{-2}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $1.20 \cdot 10^{-3}$ | $1.20 \cdot 10^{-3}$ | $1.20 \cdot 10^{-3}$ | $1.20 \cdot 10^{-3}$ | $1.20 \cdot 10^{-3}$ | $1.20 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $2.80 \cdot 10^{-5}$ | $4.70 \cdot 10^{-5}$ | $8.30 \cdot 10^{-5}$ | $7.70 \cdot 10^{-5}$ | $1.78 \cdot 10^{-4}$ | $1.43 \cdot 10^{-4}$ |
| Speed-up | $\times 43$ | $\times 26$ | $\times 14$ | $\times 16$ | $\times 7$ | $\times 8$ |

Table 4.7: TC16. Efficiency results of candidate subcases.

**Test case 17. Transport of an initial Gaussian profile**

This case presents a problem in which an initial Gaussian profile is transported. The search values are bounded to the most optimal range obtained in the previous test case, namely

$$M_{\text{POD}} \in \{1, 5, 10\}\,,$$
$$M_W \in \{1, 2, 3, 6, 13, 26\}\,.$$

All these combinations are labelled as $3 \times 6 = 18$ different subcases as listed in Table 4.8.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 |
| $M_W$ | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 |
| $M_W$ | 6 | 6 | 6 | 13 | 13 | 13 | 26 | 26 | 26 |

Table 4.8: TC17. Values of $M_{\text{POD}}$ and $M_W$.

The time-space domain of the case is defined as $(x, t) \in [0, 40] \times [0, 15]$. The spatial domain is discretized using $I_x = 200$ volume cells, and the time steps are computed dynamically according to (2.5), with CFL = 0.9. Free BCs are considered. The initial Gaussian profile is defined as

$$u(x, 0) = 1 + e^{-(x-6)^2}. \tag{4.6}$$

All the settings of this problem are the same for all subcases, as indicated in Table 4.9, except for the values of $M_{\text{POD}}$ and $M_W$.

| $L$ | $T$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|
| 20 | 5 | Eq. (4.6) | Free | 200 | 0.9 | 105 | Tab. 4.8 | Tab. 4.8 |

Table 4.9: TC17. Problem settings.

In this case, taking into account what is shown in Figure 4.8, the candidate subcases are 5, 8,

11 and 14. Among these results, Subcases 8 and 11 have obtained very satisfactory solutions with speed-ups of 28 and 16, respectively, as shown in Table 4.10. Subcase 5 is considerably worst than the rest, as its solution does not faithfully reflect the training solution at the peak, as shown in Figure 4.9a. And Subcase 14 obtains a really good precision, but it is slower than the rest. All things considered, Subcase 11 presents a very good solution (Fig. 4.9c) and its efficiency is the best.

While the number of time windows $M_W$ can be modified without substantial changes in ROM efficiency, there is an optimal number of POD modes to obtain the best results, which is 5.

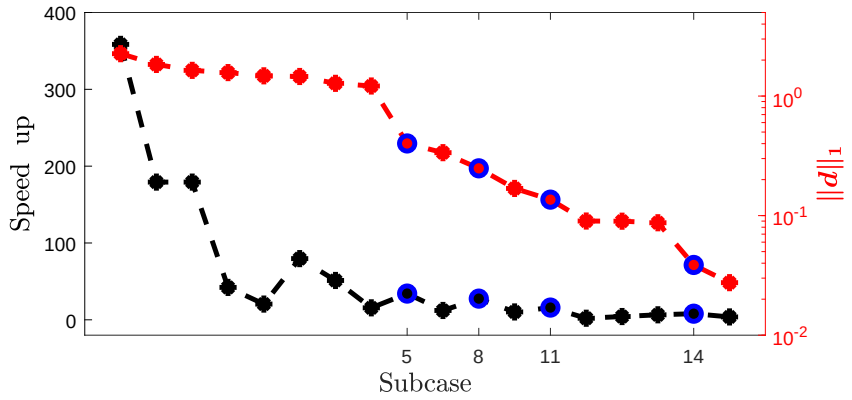
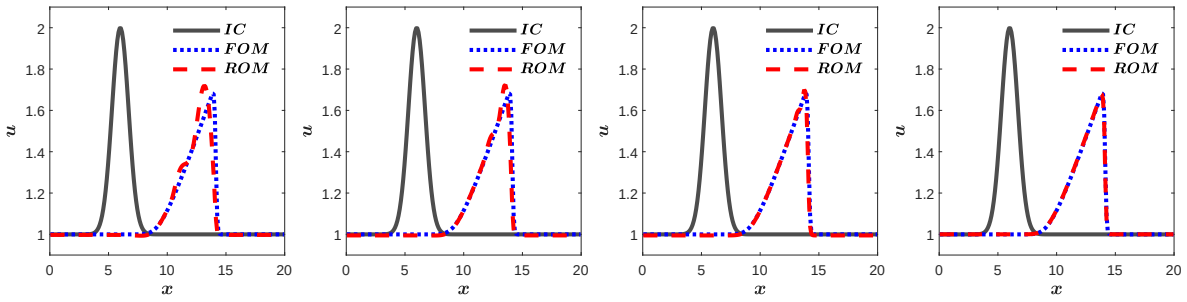



Figure 4.8: TC17. Possible optima values of $M_{\text{POD}}$ and $M_W$.




(a) Subcase 5. (b) Subcase 8. (c) Subcase 11. (d) Subcase 14.

Figure 4.9: TC17. ROM solutions of optima candidates.

| Subcase | 5 | 8 | 11 | 14 |
|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 5 | 5 | 5 |
| $M_W$ | 2 | 3 | 6 | 13 |
| $\|d\|_1$ | $4.01 \cdot 10^{-1}$ | $2.49 \cdot 10^{-1}$ | $1.36 \cdot 10^{-1}$ | $3.87 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $4.09 \cdot 10^{-1}$ | $3.15 \cdot 10^{-1}$ | $1.75 \cdot 10^{-1}$ | $5.50 \cdot 10^{-2}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $7.17 \cdot 10^{-4}$ | $7.17 \cdot 10^{-4}$ | $7.17 \cdot 10^{-4}$ | $7.17 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $2.10 \cdot 10^{-5}$ | $2.60 \cdot 10^{-5}$ | $4.50 \cdot 10^{-5}$ | $9.00 \cdot 10^{-5}$ |
| Speed-up | $\times 34$ | $\times 28$ | $\times 16$ | $\times 8$ |

Table 4.10: TC17. Efficiency results of candidate subcases.

## 4.2 ROMs and the parametrized Burgers equation

In this section, the methodology presented in Section 3.3 is extended to non-linear problems with application to the 1D Burgers equation (2.15). In the training phase of the ROM, the strategy needs to be slightly modified to include both the PID method and the combination of several training samples, as explained in the next section. Two test cases are proposed that illustrate the ability of the ROM to predict solutions of parametrized non-linear problems, as well as a new drawback that needs to be solved.

### 4.2.1 Combination of the standard ROM strategy & PID

In order to extend the ROM strategy to parametrized non-linear problems, it is necessary to combine the method appplied to linear problems (Fig. 3.22) with the PID (Fig. 4.3). For this purpose, it is necessary to group all the snapshots generated in the different training samples according to the defined time windows, as shown in the Figure 4.10. The SVD is applied on each grouping of samples to obtain the different reduced spaces.

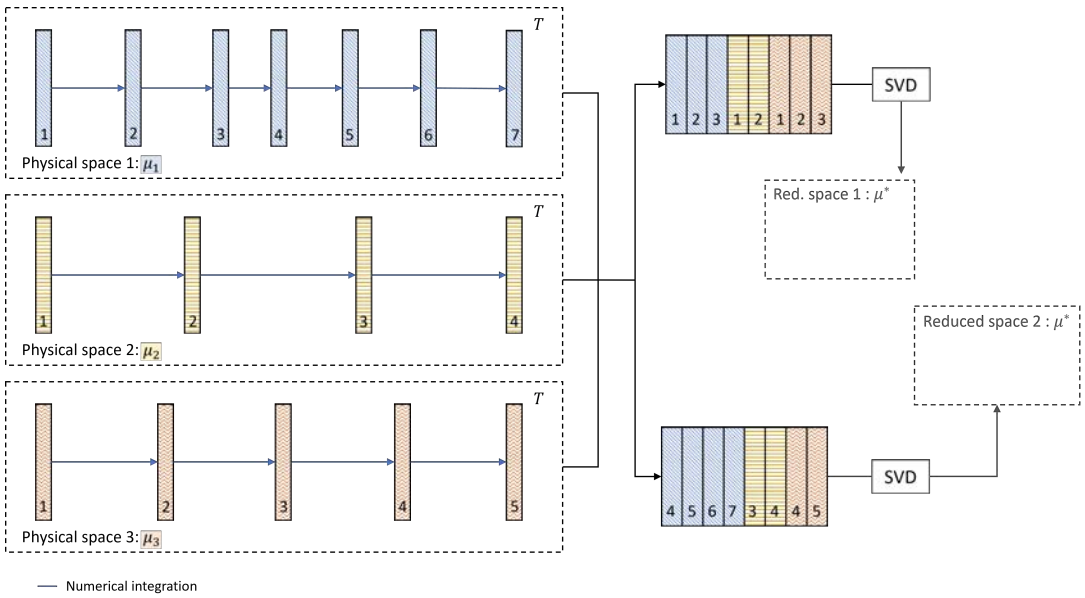



Figure 4.10: ROM strategy when solving parametrized non-linear problems.

### 4.2.2 Test case 18. Predicting the levels of a discontinuous IC

To the a posteriori criterion parameters presented in Section 4.1.4, the number of training samples is now added. This also has to be found in order to optimise the efficiency of the ROM. For this purpose, the values proposed to carry out this search are as follows

$$M_{\text{POD}} \in \{5, 10, 20\}\,,$$
$$M_W \in \{13, 27, 55\}\,,$$
$$M_{\text{train}} \in \{4, 8, 12\}\,.$$

All these combinations are labelled as $3 \times 3 \times 3 = 27$ different subcases as listed in Table 4.11.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 13 | 13 | 13 | 27 | 27 | 27 | 55 | 55 | 55 |
| $M_{\text{train}}$ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 13 | 13 | 13 | 27 | 27 | 27 | 55 | 55 | 55 |
| $M_{\text{train}}$ | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 13 | 13 | 13 | 27 | 27 | 27 | 55 | 55 | 55 |
| $M_{\text{train}}$ | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

Table 4.11: TC18. Values of $M_{\text{POD}}$ and $M_W$.

The time-space domain of the case is defined as $(x, t) \in [0, 40] \times [0, 10]$. The spatial domain is discretized using $I_x = 200$ volume cells, and the time steps are computed dynamically according to (2.5), with CFL = 0.9. Free boundary conditions are considered. The initial discontinuity is defined by

$$u(x, 0) = \begin{cases} u_L, & \text{if } x \leq 14, \\ u_R, & \text{if } x > 14, \end{cases} \tag{4.7}$$

where the left and right levels are the input parameters

$$\mu_1 = u_L, \quad \mu_2 = u_R.$$

The position of the discontinuity needs to be more carefully handled in order to include it as an input parameter, as shown in the next Test case 18. Figure 4.11 shows the training initial conditions in gray as well as the target initial condition in red.

All the settings of this problem are the same for all subcases, as indicated in Table 4.12, except for the values of $M_{\text{POD}}$, $M_W$ and $M_{\text{train}}$, and the initial conditions.

| $L$ | $T$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 40 | 15 | Eq. (4.7) | Free | 200 | 0.9 | 111 | Tab. 4.11 | Tab. 4.11 | Tab. 4.11 |

Table 4.12: TC18. Problem settings.

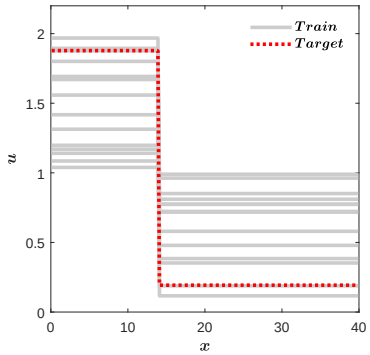



Figure 4.11: TC18. Training set of ICs and target IC.

Following the a posteriori criterion, Subcases 10, 2, 20 and 11, highlighted with the blue circles in Figure 4.12, have been chosen to be the optimal candidates. In this case, the number of POD modes must be increased to 10. The optimal number of time windows is 13, as shown in Table 4.13.

However, the results of Subcases 10 and 2 show some discrepancies with the training solution, as shown in Figures 4.13a and 4.13b. Subcase 11 is the best one in terms of similarity to the training solution, and it has the smaller differences and the same speed-up, 5, as the others, as shown in Table 4.13.

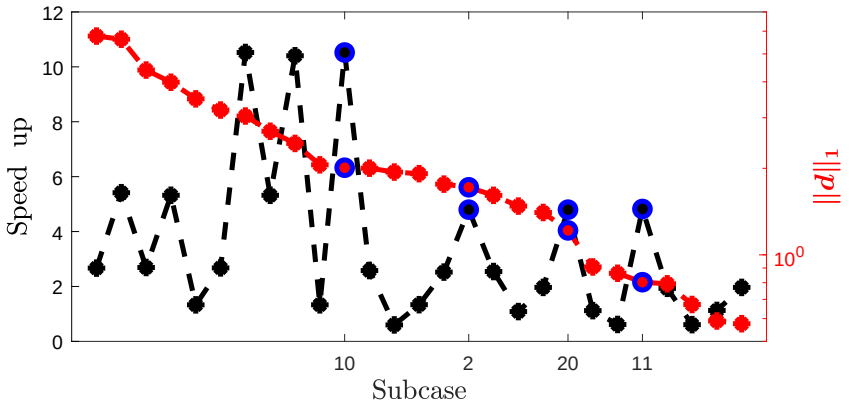
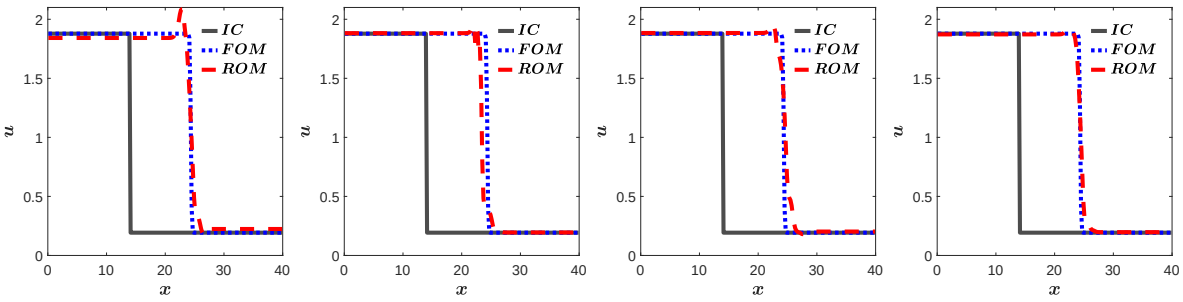



Figure 4.12: TC18. Possible optima values of $M_{\text{POD}}$ and $M_W$.




(a) Subcase 10. (b) Subcase 2. (c) Subcase 20. (d) Subcase 11.

Figure 4.13: TC18. ROM solutions of optima candidates.

| Subcase | 10 | 2 | 20 | 11 |
|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 10 | 10 | 10 |
| $M_W$ | 13 | 13 | 13 | 13 |
| $M_{\text{train}}$ | 8 | 4 | 12 | 8 |
| $\|d\|_1$ | 2.01 | 1.72 | 1.22 | $8.03 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $5.71 \cdot 10^{-1}$ | 1.48 | $7.91 \cdot 10^{-1}$ | $5.85 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $9.26 \cdot 10^{-4}$ | $9.26 \cdot 10^{-4}$ | $9.26 \cdot 10^{-4}$ | $9.26 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $8.80 \cdot 10^{-5}$ | $1.93 \cdot 10^{-4}$ | $1.93 \cdot 10^{-4}$ | $1.92 \cdot 10^{-4}$ |
| Speed-up | $\times 11$ | $\times 5$ | $\times 5$ | $\times 5$ |

Table 4.13: TC18. Efficiency results of candidate subcases.

### 4.2.3 Test case 19. Predicting the position of a discontinuous IC

This test case addresses the question mentioned in the previous test case about considering the position of the discontinuity as an input parameter. To do so, the following initial condition is solved

$$u(x,0) = \begin{cases} 2, & \text{if } x \leq x_0, \\ 1, & \text{if } x > x_0, \end{cases} \tag{4.8}$$

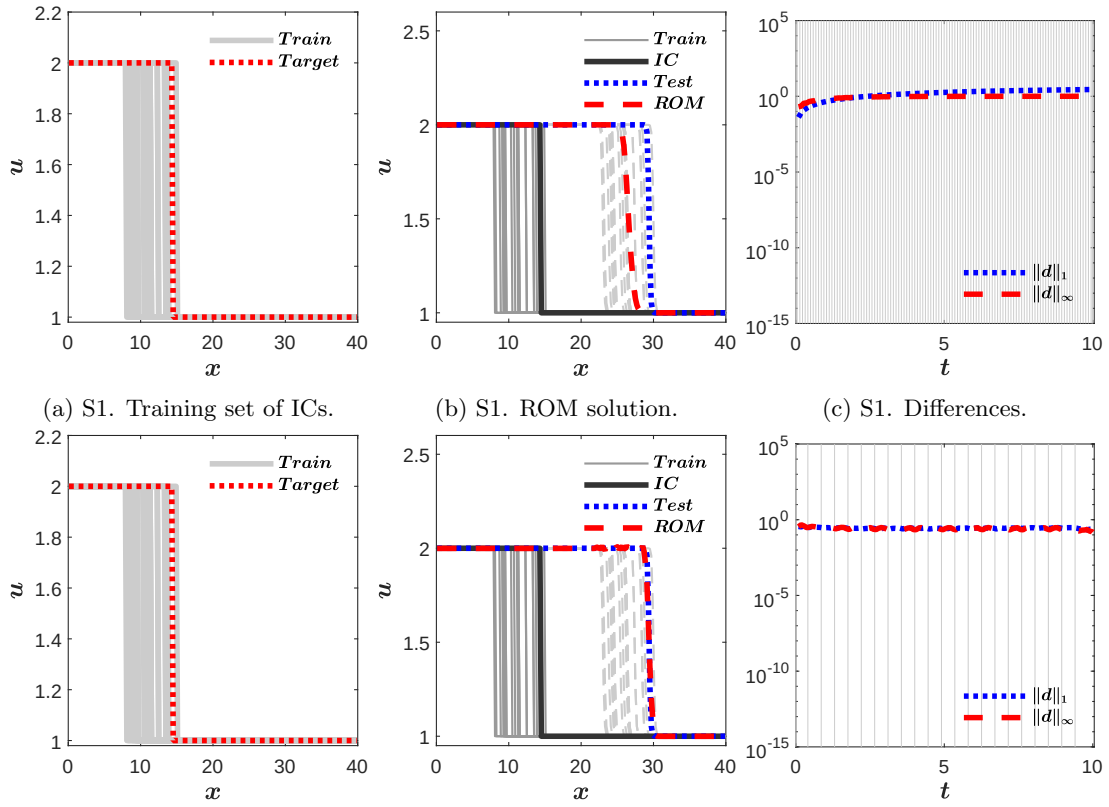
with the position $x_0$ as the only input parameter $\mu = x_0$. Table 4.14 shows the rest of the settings.

| Subcase | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|
| 1 | 200 | 113 |
| 2 | 25 | 22 |
| $L = 40$, $T = 10$, IC: Eq. (4.8), BCs: free | | |
| $I_x = 200$, CFL $= 0.9$, $N_T = 113$, $M_{\text{train}} = 20$ | | |

Table 4.14: TC19. Problem settings.

The linearized ROM (4.4) fails to predict the position of the discontinuity when the target value is different than those of the training, even if it uses the maximum number of POD modes, the maximum number of time windows and a high number of training samples, as Subcase 1 does. The final position of the discontinuity is underestimated, as shown in Figure 4.14b.

To achieve good prediction it is necessary to use ROMs that do not require linearization. Thus, if the non-linear ROM (4.2) solves the same case, it is capable of adequately predicting the target position of the discontinuity, as can be seen in the Figure 4.14e. Despite this, since it is a non-linear ROM, it does not manage to speed up the calculation times required by the test solution calculated by the FOM, as can be observed in the Table 4.15.




(a) S1. Training set of ICs. (b) S1. ROM solution. (c) S1. Differences.

(d) S2. Training set of ICs. (e) S2. ROM solution. (f) S2. Differences.

Figure 4.14: TC19. Treating $x_0$ as a varying parameters.

| Subcase | $\|d\|_1$ | $\|d\|_\infty$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
|---|---|---|---|---|---|
| 1 | 2.80 | $9.99 \cdot 10^{-1}$ | $4.49 \cdot 10^{-3}$ | $6.66 \cdot 10^{-2}$ | 0 |
| 2 | $2.07 \cdot 10^{-1}$ | $1.52 \cdot 10^{-1}$ | $4.68 \cdot 10^{-3}$ | $1.34 \cdot 10^{-2}$ | 0 |

Table 4.15: TC19. Efficiency results.

### 4.2.4 Test case 20. Predicting the shape of a Gaussian profile as IC

The values proposed to carry out this search are as follows

$$M_{\text{POD}} \in \{10, 25, 50\}\,,$$
$$M_W \in \{8, 17, 69\}\,,$$
$$M_{\text{train}} \in \{5, 10, 20\}\,.$$

All these combinations are labelled as $3 \times 3 \times 3 = 27$ different subcases as listed in Table 4.16.

The time-space domain of the case is defined as $(x, t) \in [0, 40] \times [0, 10]$. The spatial domain is discretized using $I_x = 200$ volume cells, and the time steps are computed dynamically according to (2.5), with CFL $= 0.9$. Free boundary conditions are considered. The initial condition is defined by

$$u(x,0) = 1 + u_0 e^{-c(x-14)^2}, \tag{4.9}$$

where the coefficients are treated as input parameters $\mu_1 = u_0$, $\mu_2 = c$.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 10 | 25 | 50 | 10 | 25 | 50 | 10 | 25 | 50 |
| $M_W$ | 8 | 8 | 8 | 17 | 17 | 17 | 69 | 69 | 69 |
| $M_{\text{train}}$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 10 | 25 | 50 | 10 | 25 | 50 | 10 | 25 | 50 |
| $M_W$ | 8 | 8 | 8 | 17 | 17 | 17 | 69 | 69 | 69 |
| $M_{\text{train}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 10 | 25 | 50 | 10 | 25 | 50 | 10 | 25 | 50 |
| $M_W$ | 8 | 8 | 8 | 17 | 17 | 17 | 69 | 69 | 69 |
| $M_{\text{train}}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Table 4.16: TC20. Values of $M_{\text{POD}}$ and $M_W$.

Again, the position of the Gaussian profile is studied in the next case. Figure 4.15 shows the training initial conditions in gray as well as the target initial condition in red. The settings of the problem are found in Table 4.17.

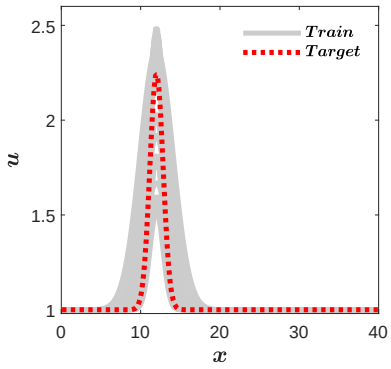



Figure 4.15: TC20. Training set of ICs and target IC.

| $L$ | $T$ | IC | BCs | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 40 | 15 | Eq. (4.9) | Free | 200 | 0.9 | 139 | Tab. 4.16 | Tab. 4.16 | Tab. 4.16 |

Table 4.17: TC20. Problem settings.

Subcases 1, 19, 13 and 10, highlighted with blue circles in Figure 4.16, have been chosen to be the optimal candidates. Again, $M_{\text{POD}}$ is common to all candidate subcases, while $M_W$ and $M_{\text{train}}$ are the parameters that can vary, as shown in Table 4.18.

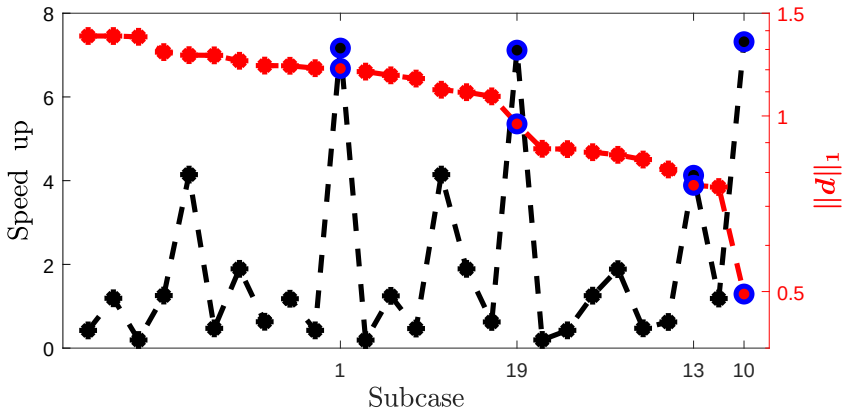
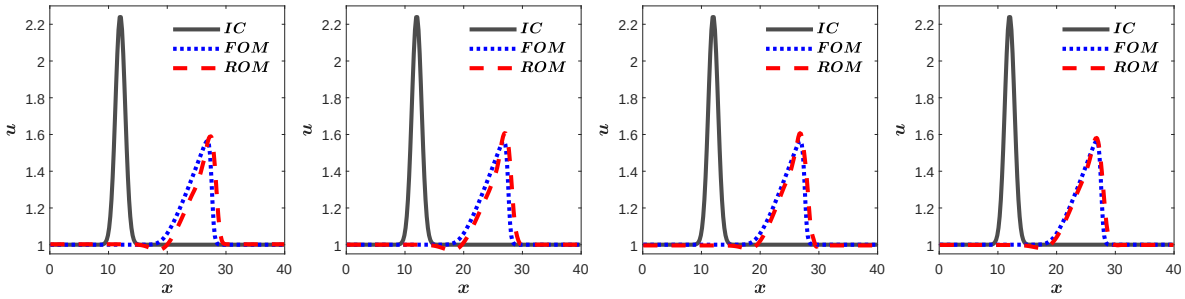



Figure 4.16: TC20. Possible optima values of $M_{\text{POD}}$ and $M_W$.

Looking at the Figure 4.17, it is immediately clear that Subcase 10 is the best, as it preserves the Gaussian profile until the final time, which corresponds to a low error, as shown in the Table 4.18. Moreover, its speed-up, although not very large, is among the highest in the candidate set. Apart from that, Subcase 19, despite having twice as many training samples as Subcase 10, presents a considerably poorer solution. This is due to the over-training of the ROM.




(a) Subcase 1. (b) Subcase 19. (c) Subcase 13. (d) Subcase 10.

Figure 4.17: TC20. ROM solutions of optima candidates.

| Subcase | 1 | 19 | 13 | 10 |
|---|---|---|---|---|
| $M_{\text{POD}}$ | 10 | 10 | 10 | 10 |
| $M_W$ | 8 | 8 | 17 | 8 |
| $M_{\text{train}}$ | 5 | 20 | 10 | 10 |
| $\|d\|_1$ | 1.21 | $9.69 \cdot 10^{-1}$ | $7.61 \cdot 10^{-1}$ | $4.95 \cdot 10^{-1}$ |
| $\|d\|_\infty$ | $3.89 \cdot 10^{-1}$ | $3.28 \cdot 10^{-1}$ | $2.60 \cdot 10^{-1}$ | $2.06 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $1.03 \cdot 10^{-3}$ | $1.03 \cdot 10^{-3}$ | $1.03 \cdot 10^{-3}$ | $1.03 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $1.44 \cdot 10^{-4}$ | $1.45 \cdot 10^{-4}$ | $2.50 \cdot 10^{-4}$ | $1.41 \cdot 10^{-4}$ |
| Speed-up | $\times 7$ | $\times 7$ | $\times 4$ | $\times 7$ |

Table 4.18: TC20. Efficiency results of candidate subcases.

### 4.2.5 Test case 21. Predicting the position of a Gaussian profile

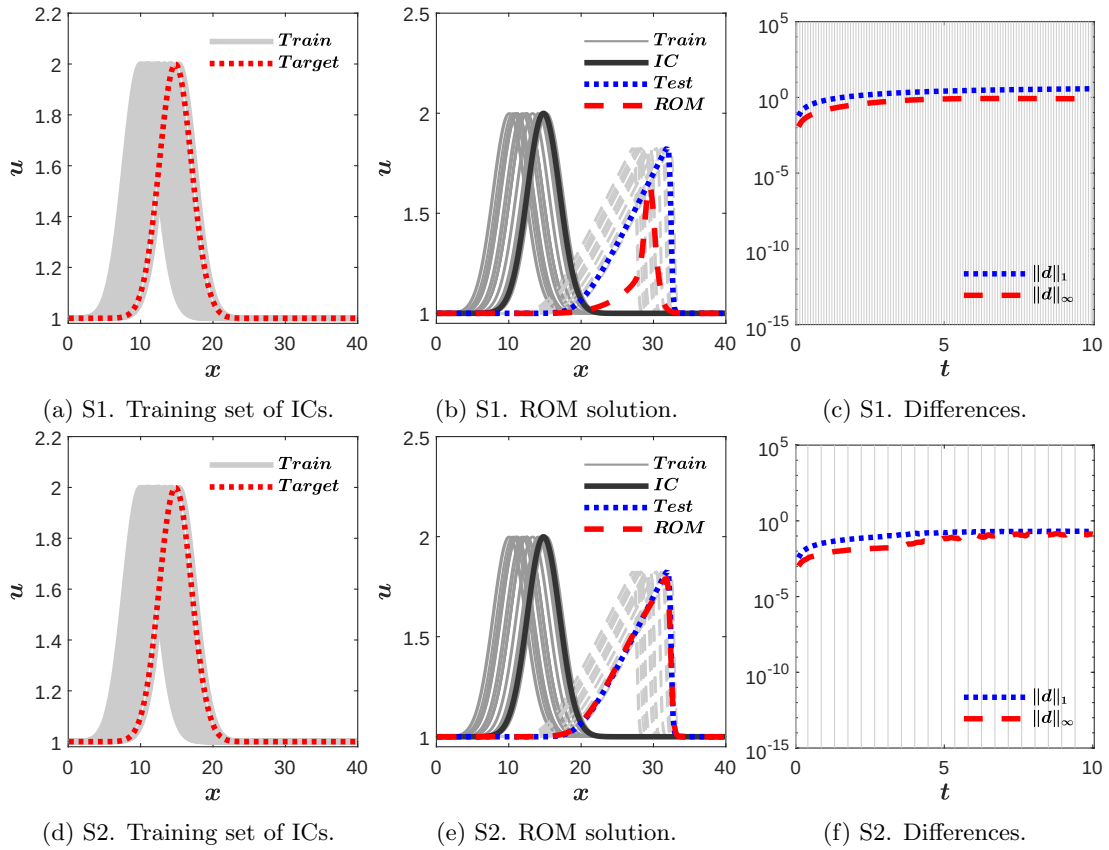
If the position of the Gaussian is proposed as an input parameter, this test case yields the same conclusions as the discontinuity case. The following Gaussian function condition is proposed as the IC

$$u(x,0) = 1 + e^{-0.1(x-x_0)^2}, \tag{4.10}$$

with the position as the only input parameter $\mu = x_0$. Table 4.19 shows the rest of the settings.

| Subcase | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|
| 1 | 200 | 113 |
| 2 | 25 | 22 |
| $L = 40$, $T = 10$, IC: Eq. (4.10), BCs: free | | |
| $I_x = 200$, CFL $= 0.9$, $N_T = 113$, $M_{\text{train}} = 20$ | | |

Table 4.19: TC21. Problem settings.




(a) S1. Training set of ICs. (b) S1. ROM solution. (c) S1. Differences.

(d) S2. Training set of ICs. (e) S2. ROM solution. (f) S2. Differences.

Figure 4.18: TC21. Treating $x_0$ as a varying parameters.

The linearized ROM (4.4) fails to predict the position of the discontinuity, whereas the non-linear ROM (4.2) accurately predicts it, as shown in Figures 4.18b and 4.18e, respectively, both

with poor speed-ups (see Table 4.20).

| Subcase | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ | $\|d\|_1$ | $\|d\|_\infty$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
|---|---|---|---|---|---|---|---|---|
| 1 | 200 | 113 | 20 | 3.74 | $8.05 \cdot 10^{-1}$ | $4.67 \cdot 10^{-3}$ | $7.08 \cdot 10^{-2}$ | 0 |
| 2 | 25 | 22 | 20 | $2.04 \cdot 10^{-1}$ | $1.31 \cdot 10^{-1}$ | $4.60 \cdot 10^{-3}$ | $1.33 \cdot 10^{-2}$ | 0 |

Table 4.20: TC21. Efficiency results.

## 4.3 Beyond the training time (II)

The extension of the coordinate transformation (3.25) to non-linear problems such as the Burgers equation (2.15) is not directly applicable. The shock and rarefaction waves that are generated as a result of non-linearity clash with the mapping, for different reasons in each case. In shocks, several characteristic lines intrinsic to the problem converge into one, so it is necessary to follow this reduction in the number of curves; and in rarefactions, fans of several characteristic lines appear, so there is an increase in the number of characteristic curves.

This means that the numerical mesh can have a variable number of characteristic curves and, therefore, when applying the coordinate transformation (3.25), the equations will be divided into as many pieces as there are characteristic curves. This definition of the problem according to its characteristics considerably increases its complexity in non-linear problems, where multiple lines may appear. For both shocks and rarefactions, it is necessary to closely follow the evolution of the problem in the physical space, which complicates the resolution of the ROM enormously.

In this thesis, an extension method of the coordinate transformation is proposed that allows the ROM to predict solutions beyond the training time in non-linear problems. It is illustrated and evaluated by means of a couple of test cases that present in a simplified way how to deal with the resolution of shock and rarefaction waves separately. It should be noted that the PID method is not applicable when considering non-linear problems and time prediction, so no time windows are used in these test cases.

### 4.3.1 Development of the CTROM of the Burgers equation

Let there be two characteristic curves such that

$$d_i'(t) = u(d_i(t), t), \quad d_i(0) \text{ given}, \, i = 1, 2, \quad 0 \leq t \leq t_c,$$

where $t_c \leq T$ is the critical value such that the solution is single-valued and $d_1(t) \leq d_2(t)$ is assumed for $0 \leq t \leq t_c$. If the two characteristic curves intersect, $d_1(t_c) = d_2(t_c)$, then a shock wave is generated at $t = t_c$ and a similar transformation to (3.25) is used for $t > t_c$. If a rarefaction wave is produced by the Burgers equation, three sub-domains are considered for $0 \leq t \leq T$.

When the spatial domain is divided into three sub-domains, the coordinate transformation for the characteristic curves $d_1(t)$ and $d_2(t)$, reads as follows

$$\tilde{x}(t) = \begin{cases} \dfrac{d_1(0)}{d_1(t)}x, & \text{if } 0 \le x < d_1(t), \\ d_1(0) + \dfrac{d_2(0) - d_1(0)}{d_2(t) - d_1(t)}\left(x - d_1(t)\right), & \text{if } d_1(t) \le x < d_2(t), \\ L - \dfrac{L - d_2(0)}{L - d_2(t)}(L - x), & \text{if } d_2(t) \le x \le L. \end{cases} \tag{4.11}$$

The 1D Burgers equation (2.15) in the transformed domain is

$$\begin{cases} \dfrac{\partial \tilde{u}}{\partial t} + \left(\tilde{u}(\tilde{x},t)\dfrac{d_1(0)}{d_1(t)} - \tilde{u}(d_1(0),t)\dfrac{\tilde{x}}{d_1(t)}\right)\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } 0 \le \tilde{x} < d_1(0), \\ \dfrac{\partial \tilde{u}}{\partial t} = 0, & \text{if } d_1(0) \le \tilde{x} < d_2(0), \\ \dfrac{\partial \tilde{u}}{\partial t} + \left(\tilde{u}(\tilde{x},t)\dfrac{L - d_2(0)}{L - d_2(t)} - \tilde{u}(d_2(0),t)\dfrac{L - \tilde{x}}{L - d_2(t)}\right)\dfrac{\partial \tilde{u}}{\partial \tilde{x}} = 0, & \text{if } d_2(0) \le \tilde{x} \le L, \end{cases}$$

where $\tilde{u}(\tilde{x},t) = u(x,t)$.

The CTFOM of the 1D inviscid Burgers equation is obtained by means of the FV method

$$\begin{aligned} \tilde{u}_i^{n+1} &= \tilde{u}_i^n - \frac{\Delta t}{\Delta \tilde{x}}\left[\frac{d_1^0}{d_1^n}\left(\tilde{f}_{i+1/2}^{n,-,*} - \tilde{f}_{i-1/2}^{n,+,*}\right) - \frac{1}{d_1^n}\left(\tilde{f}_{i+1/2}^{n,-,**} - \tilde{f}_{i-1/2}^{n,+,**}\right)\right], \text{ if } 0 \le \tilde{x}_i < d_1^0, \\ \tilde{u}_i^{n+1} &= \tilde{u}_i^n, \text{ if } d_1^0 \le \tilde{x}_i < d_2^0, \\ \tilde{u}_i^{n+1} &= \tilde{u}_i^n - \frac{\Delta t}{\Delta \tilde{x}}\left[\frac{L - d_2^0}{L - d_2^n}\left(\tilde{f}_{i+1/2}^{n,-,*} - \tilde{f}_{i-1/2}^{n,+,*}\right) - \frac{1}{L - d_2^n}\left(\tilde{f}_{i+1/2}^{n,-,**} - \tilde{f}_{i-1/2}^{n,+,**}\right)\right], \\ &\qquad\qquad \text{if } d_2^0 \le \tilde{x}_i \le L, \end{aligned} \tag{4.12}$$

where the numerical fluxes are

$$\tilde{f}_{i+1/2}^{n,\pm,*} = f(\tilde{u}_i^n) \mp \left(\bar{\lambda}^{\pm}\right)_{i+1/2}^n \delta\tilde{u}_{i+1/2}^n,$$

$$\tilde{f}_{i+1/2}^{n,\pm,**} = \begin{cases} f(\tilde{u}_i^n) \mp \tilde{x}_{i+1/2}\left(\bar{\lambda}^{\pm}\right)_{I_{d_1}+1/2}^n \delta\tilde{u}_{i+1/2}^n, & \text{if } i \le I_{d_1} \\ f(\tilde{u}_i^n), & \text{if } I_{d_1} < i \le I_{d_2}, \\ f(\tilde{u}_i^n) \mp \left(L - \tilde{x}_{i+1/2}\right)\left(\bar{\lambda}^{\pm}\right)_{I_{d_2}+1/2}^n \delta\tilde{u}_{i+1/2}^n, & \text{if } I_{d_2} < i \end{cases},$$

with $\delta\tilde{u}_{i+1/2}^n = \tilde{u}_{i+1}^n - \tilde{u}_i^n$, and

$$\left(\bar{\lambda}^{\pm}\right)_{I_{d_1}+1/2}^n = \frac{1}{2}\left(\bar{\lambda} \pm |\bar{\lambda}|\right)_{I_{d_1}+1/2}^n,$$
$$\left(\bar{\lambda}^{\pm}\right)_{I_{d_2}+1/2}^n = \frac{1}{2}\left(\bar{\lambda} \pm |\bar{\lambda}|\right)_{I_{d_2}+1/2}^n,$$

where $\bar{\lambda}_{I_{d_1}+1/2}^n = \dfrac{1}{2}\left(u_{I_{d_1}+1}^n + u_{I_{d_1}}^n\right)$; and $\bar{\lambda}_{I_{d_2}+1/2}^n = \dfrac{1}{2}\left(u_{I_{d_2}+1}^n + u_{I_{d_2}}^n\right)$.

In this case, the characteristic curves are approximated with the explicit Euler method

$$\frac{d_k^{n+1}-d_k^n}{\Delta t}=(\tilde{u}_i)^n_{I_{ik}+1/2},\ k=1,2.$$

The intrusive CTROM is obtained from the CTFOM (4.12) as has been done in previous cases

$$\begin{aligned}\hat{u}_p^{n+1}=\hat{u}_p^n+\frac{\Delta t}{\Delta\tilde{x}}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{u}_k^n\sum_{q=1}^{M_{\mathrm{POD}}}\hat{u}_q^n\left[\frac{1}{d_1^n}A^L_{pkq}+\frac{1}{L-d_2^n}A^R_{pkq}\right]\\+\frac{\Delta t}{\Delta\tilde{x}}\sum_{k=1}^{M_{\mathrm{POD}}}|\hat{u}_k^n|\sum_{q=1}^{M_{\mathrm{POD}}}\hat{u}_q^n\left[\frac{1}{d_1^n}B^L_{pkq}+\frac{1}{L-d_2^n}B^R_{pkq}\right],\end{aligned}\tag{4.13}$$

where the coeficcients are

$$\begin{aligned}A^L_{pkq}&=a_1\phi_{1,p}+d_1^0\frac{1}{2}\sum_{i=2}^{I_{d_1}}\left[\tilde{\phi}_{i+1/2,k}\delta\phi_{i+1/2,q}+\tilde{\phi}_{i-1/2,k}\delta\phi_{i-1/2,q}\right]\phi_{i,p}\\&\quad+\frac{1}{2}\sum_{i=2}^{I_{d_1}}\left[\tilde{x}_{i+1/2}\tilde{\phi}_{I_{d_1}+1/2,k}\delta\phi_{i+1/2,q}+\tilde{x}_{i-1/2}\tilde{\phi}_{I_{d_1}-1/2,k}\delta\phi_{i-1/2,q}\right]\phi_{i,p},\\A^R_{pkq}&=a_{I_x}\phi_{I_x,p}+\left(L_0-d_2^0\right)\frac{1}{2}\sum_{i=I_{d_2}+1}^{I_x-1}\left[\tilde{\phi}_{i+1/2,k}\delta\phi_{i+1/2,q}+\tilde{\phi}_{i-1/2,k}\delta\phi_{i-1/2,q}\right]\phi_{i,p}\\&\quad+\frac{1}{2}\sum_{i=I_{d_2}+1}^{I_x-1}\left[\left(L-\tilde{x}_{i+1/2}\right)\tilde{\phi}_{I_{d_2}+1/2,k}\delta\phi_{i+1/2,q}+\left(L-\tilde{x}_{i-1/2}\right)\tilde{\phi}_{I_{d_2}-1/2,k}\delta\phi_{i-1/2,q}\right]\phi_{i,p},\\B^L_{pkq}&=b_1\phi_{1,p}+d_1^0\frac{1}{2}\sum_{i=2}^{I_{d_1}}\left[|\tilde{\phi}_{i+1/2,k}|\delta\phi_{i+1/2,q}-|\tilde{\phi}_{i-1/2,k}|\delta\phi_{i-1/2,q}\right]\phi_{i,p}\\&\quad+\frac{1}{2}\frac{1}{d_1^n}\sum_{i=2}^{I_{d_1}}\left[\tilde{x}_{i+1/2}|\tilde{\phi}_{I_{d_1}+1/2,k}|\delta\phi_{i+1/2,q}-\tilde{x}_{i-1/2}|\tilde{\phi}_{I_{d_1}-1/2,k}|\delta\phi_{i-1/2,q}\right]\phi_{i,p},\\B^R_{pkq}&=b_{I_x}\phi_{I_x,p}+\left(L-d_2^0\right)\frac{1}{2}\sum_{i=I_{d_2}+1}^{I_x-1}\left[|\tilde{\phi}_{i+1/2,k}|\delta\phi_{i+1/2,q}-|\tilde{\phi}_{i-1/2,k}|\delta\phi_{i-1/2,q}\right]\phi_{i,p}\\&\quad+\frac{1}{2}\sum_{i=I_{d_2}+1}^{I_x-1}\left[\left(L-\tilde{x}_{i+1/2}\right)|\tilde{\phi}_{I_{d_2}+1/2,k}|\delta\phi_{i+1/2,q}-\left(L-\tilde{x}_{i-1/2}\right)|\tilde{\phi}_{I_{d_2}-1/2,k}|\delta\phi_{i-1/2,q}\right]\phi_{i,p}.\end{aligned}$$

where $\delta\phi_{i+1/2,q}=\phi_{i+1,q}-\phi_{i,q}$, $\tilde{\phi}_{i+1/2,k}=\left(\phi_{i+1,k}+\phi_{i,k}\right)/2$ and $\tilde{\phi}_{I_{d_m}+1/2,k}=\left(\phi_{I_{d_m}+1,k}+\phi_{I_{d_m},k}\right)/2$, with $m=1,2$. Coefficients $a_1$, $a_{I_x}$, $b_1$, $b_{I_x}$ are determined by the boundary conditions imposed similar to (3.9).

### 4.3.2 Test case 22. 1D shock generation

In this case, the generation of a shock wave is considered. The IC of this problem is

$$u(x,0) = \begin{cases} 3, & \text{if } 0 < x \leq d_1(0), \\ 3 - 2\dfrac{x - d_1(0)}{d_2(0) - d_1(0)}, & \text{if } d_1(0) < x < d_2(0), \\ 1, & \text{if } d_2(0) \leq x \leq L, \end{cases} \tag{4.14}$$

where $L = 2$ and the starting points of the characteristic curves are $d_1(0) = 0.25$ and $d_2(0) = 0.55$. The final time is $T = 0.65$ and the training time is $T_{\text{train}} = 0.35$. Free boundary condition are imposed.

The linear slope of the ramp in the IC (4.14) will steepen until a shock is generated. At this time $t_c$, the two characteristic curves $x = d_1(t)$ and $x = d_2(t)$, with $t < t_c$, converge into a single characteristic curve $x = d_3(t)$, with $t \geq t_c$. All the points of the physical mesh in the central sub-domain are eventually mapped in the shock front and are no longer useful. Thus, the central sub-domain is suppressed to return to a two sub-domain problem. Numerically, at the critical time $t_c$, the problem is redefined by re-meshing, maintaining the original number of cells. The characteristic curves in the spatial domain are shown in Figure 4.19, where the time evolution of the physical mesh is represented for both the CTFOM and the CTROM.

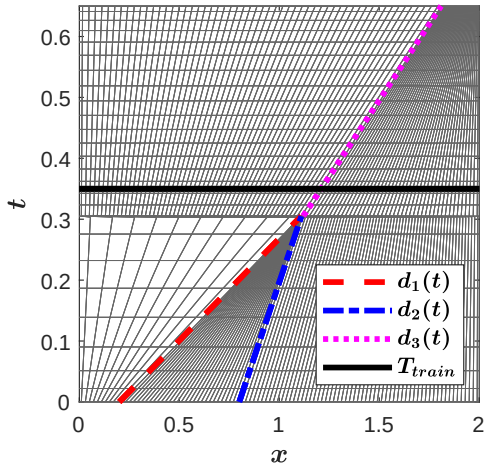
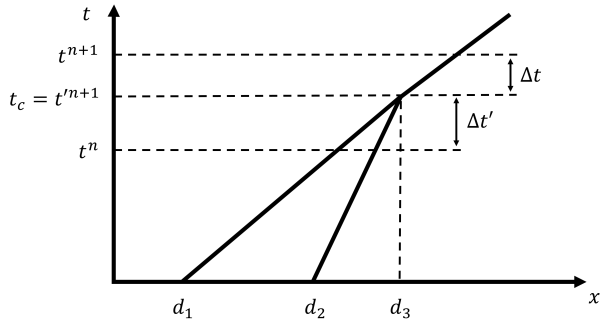



Figure 4.19: TC22: Time evolution of the physical mesh for a shock wave generation.

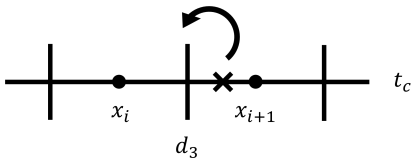
The training time $T_{\text{train}}$ is also shown in Figure 4.19 and it is observed that $T_{\text{train}} > t_c$ in this case. It should be noted that $T_{\text{train}}$ could be shorter than $t_c$, but, in that case, it would be necessary to train the reduced order model with the CTFOM before and after the shock wave is generated.

Before commenting on the results obtained with the CTROM for this case, it is necessary to take into account a couple of numerical considerations for solving a problem with three sub-domains. On the one hand, the confluence must be carefully solved, fitting the time step $\Delta t$ to the critical time $t_c$ satisfying that $d_1(t_c) = d_2(t_c)$, as depicted in Figure 4.20a. On the other

hand, the starting point $(d_3(t_c), t_c)$ of the new characteristic curve has to be moved to the nearest wall so that $\tilde{x}_{I+1/2} = d_3(t_c)$, in order to keep the stability, as depicted in Figure 4.20b. The coordinate transform method is very sensitive to this point.




(a) Confluence of $x = d_1(t)$ and $x = d_2(t)$ into $x = d_3(t)$.




(b) Moving $(d_3(t_c), t_c)$ to the nearest interface.

Figure 4.20: TC22: Numerical considerations for solving a problem with three sub-domains.

Regarding the data of the numerical problem, the spatial domain $[0, L = 2]$ is divided into $I_x = 100$ volume cells, so that $\Delta\tilde{x} = 0.02$. The CFL number considered is 0.9 and the time step is computed to satisfy the following stability condition

$$\Delta t = CFL \frac{\Delta\tilde{x}}{\max\{\tilde{a}_1, \tilde{a}_2\}},$$

where the modified velocities are

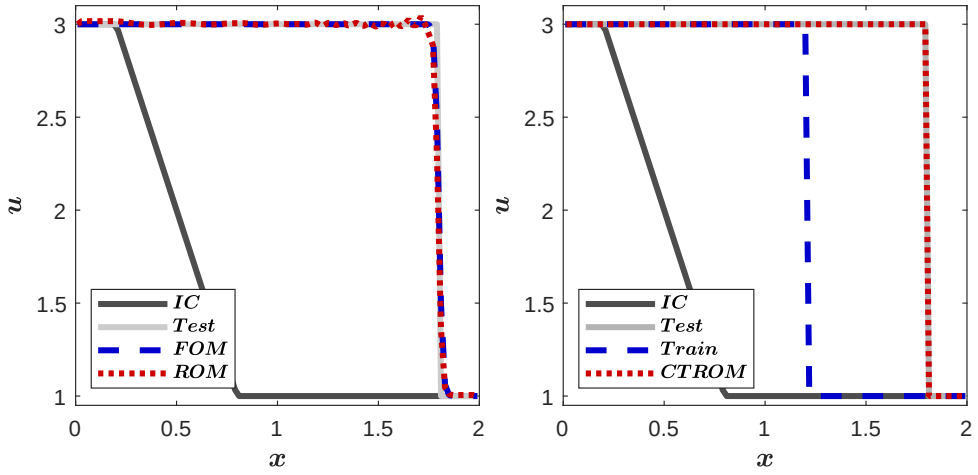
$$\tilde{a}_1 = \max_{0<\tilde{x}<d_1(0)} \left| \tilde{u}(\tilde{x}, t) \frac{d_1(0)}{d_1(t)} - \tilde{u}(d_1(0), t) \frac{\tilde{x}}{d_1(t)} \right|,$$
$$\tilde{a}_2 = \max_{d_2(0)<\tilde{x}<L} \left| \tilde{u}(\tilde{x}, t) \frac{L - d_2(0)}{L - d_2(t)} - \tilde{u}(d_2(0), t) \frac{L - \tilde{x}}{L - d_2(t)} \right|.$$

The number of time steps used to train the CTROM is $M_{\text{train}} = 33$ and $M_{\text{POD}} = 10$. All these settings are contained in Table 4.21.

| $L$ | $T_{\text{train}}$ | $T$ | IC | BCs | $d_1(0)$ | $d_2(0)$ | $I_x$ | CFL | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.35 | 0.65 | Eq. (4.14) | Fixed | 0.25 | 0.55 | 100 | 0.9 | 32 | 75 | 10 |

Table 4.21: TC22. Problem settings.

Figure 4.21 shows the solutions computed with the FOM (2.17) and the ROM (4.2); and the CTFOM (4.12) and the CTROM (4.13). From this figure, the following conclusions can be drawn: i) a proper prediction in time is computed with the CTROM; and ii) the solution computed with the CTROM does not exhibit spurious oscillations, as is the case with the ROM.




(a) FOM/ROM. (b) CTFOM/CTROM.

Figure 4.21: TC22: Solutions computed with the FOM/ROM (left) and with the CTFOM/CTROM (right).

### 4.3.3 Test case 23. 1D rarefaction generation

This case considers the Burgers equation (2.15) with the following IC

$$u(x,0) = \begin{cases} 1, & \text{if } 0 < x \leq d_1(0), \\ 1 + 2\dfrac{x - d_1(0)}{d_2(0) - d_1(0)}, & \text{if } d_1(0) < x < d_2(0), \\ 3, & \text{if } d_2(0) \leq x \leq L, \end{cases} \tag{4.15}$$

where the spatial domain $[0, L = 2]$ is divided into $I_x = 100$ volume cells, the starting points of the characteristic curves are $d_1(0) = 0.2$ and $d_2(0) = 0.22$ and a free BCs are imposed.

In this case, the characteristic curves $x = d_1$ and $x = d_2$ do not intersect. However, the starting points of the characteristic curves are so close to each other that a uniform mesh would contain very few points between them, and it could even contain only one point if it is coarse enough. Then, a finer mesh is set in the middle section, between $d_1(0)$ and $d_2(0)$, to properly reproduce the non-linear character of the Burgers equation. Taking the latter into account, the spatial domain $[0, L]$ is divided into

$$[0, L] = [0, d_1(0)] \cup [d_1(0), d_2(0)] \cup [d_2(0), L],$$

and a piecewise uniform mesh is constructed with mesh widths $\Delta\tilde{x} = 0.02$, $\Delta\tilde{x} = 0.001$ and $\Delta\tilde{x} = 0.02$, respectively. The CFL number considered is 0.9, the training time is $T_{\text{train}} = 0.2$, which corresponds to $M_{\text{train}} = 41$, and the final time is $T = 0.5$. $M_{\text{POD}} = 11$ are used. All problem settings are shown in Table 4.22. The time evolution of the physical mesh for both CTFOM and CTROM is shown in Figure 4.22.

| $L$ | $T_{\text{train}}$ | $T$ | IC | BCs | $d_1(0)$ | $d_2(0)$ | $I_x$ | CFL | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.2 | 0.5 | Eq. (4.15) | Fixed | 0.25 | 0.55 | 100 | 0.9 | 40 | 212 | 10 |

Table 4.22: TC23. Problem settings.

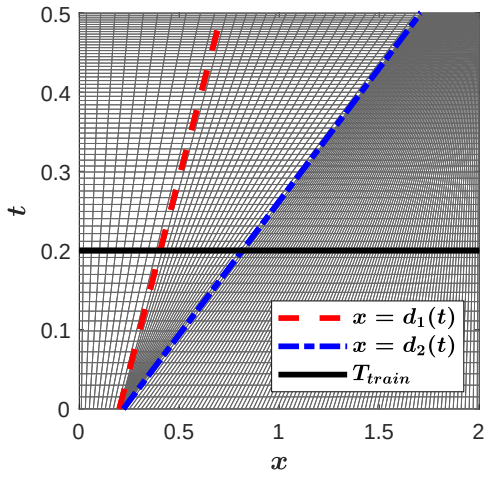
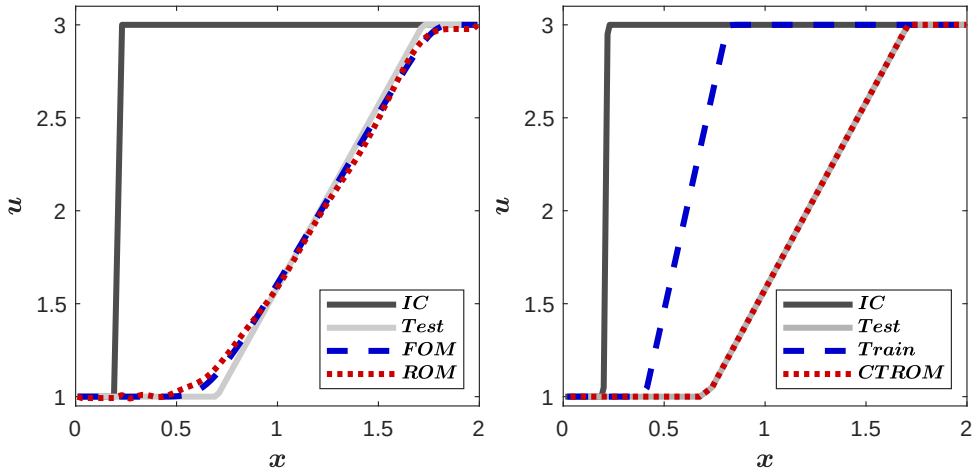



Figure 4.22: TC23. Time evolution of the physical mesh for a rarefaction wave generation.

Figure 4.23 shows solutions computed using the FOM/ROM (left) and CTFOM/CTROM (right). Although the ROM is trained until the final time $T$, the CTROM gives a better approximation to the solution of TC22. In this rarefaction case, the same conclusions can be drawn as in TC22 above and the prediction in time is only possible with the CTROM.




(a) FOM/ROM. (b) CTFOM/CTROM.

Figure 4.23: TC23: Solutions computed with the ROM (left) and with the CTROM (right).

Finally, in the next case, the IC of the non-linear problem is not a piecewise linear function is considered. The problem is defined in the domain $[0, 2] \times [0, 0.4]$ and the IC is

$$u(x,0) = \begin{cases} 0, & \text{if } 0 < x \leq d_1(0), \\ \dfrac{(x - d_1(0))x^2}{(d_2(0) - d_1(0))d_2^2(0)}, & \text{if } d_1(0) < x < d_2(0), \\ 1, & \text{if } d_2(0) \leq x \leq 2, \end{cases}$$

where the starting points of the characteristic curves are $d_1(0) = 0.25$ and $d_2(0) = 0.5$ and free boundary conditions are considered.

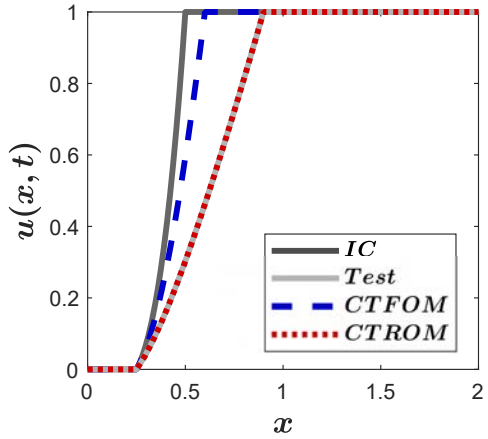



Figure 4.24: TC23: Solutions computed with the CTFOM/CTROM.

The solution of this case has a raferaction wave and it is approximated with the CTFOM and the CTROM. The spatial domain is divided into $I_x = 128$ volume cells, $CFL = 0.9$, the training time is $T_{\text{train}} = 0.1$ and the number of modes is $M_{\text{POD}} = 10$. The IC and the computed solutions with both models at the final time $T = 0.4$ are shown in Figure 4.24. It can be concluded that both solutions for this case are very similar even though the number of modes of the CTROM is very small.

## 4.4 Concluding remarks

In this chapter, reduced-order modelling has been applied to the 1D Burgers equation (2.15).

The advantages in terms of efficiency of using linearized versus fully non-linear ROMs when solving non-linear problems have been explained. The development of both types of ROMs has been detailed, making use of the PID method for the former. According to this method, the snapshot matrix is divided following the time windows, each of which defines a different reduced space. This allows the non-linearity of the solutions to be approximated by a linearized ROM. Numerical results have been shown to encourage the use of the linearised version due to its faster computational times.

An a posteriori criterion based on the efficiency of the solutions calculated by the linearised ROM has been proposed to choose the optimal pair of values for the number of POD modes and the number of time windows used by the ROM. The results yielded by this criterion establish

that the most efficient number of POD modes is between 5 and 10, while $M_W$ may vary more depending on the problem considered.

Regarding the application of ROMs to the parametrized Burgers equation, it has been shown that there is a limitation when using linearized ROMs in predicting positions of the initial conditions or the geometric term beyond the training sample set. This can be solved by using ROMs that do not require linearization.

The extension of the time prediction method to nonlinear problems requires a careful and complex treatment of the defining characteristics of the nonlinearity, i.e. the shock and rarefaction waves. This precludes the use of time windows. A solution to this problem has been proposed for both particular cases.

# Chapter 5

# ROMs applied to the SWE

In this chapter, a step-by-step analysis of the application of ROMs to the SWE is carried out before solving realistic cases.

Firstly, the ROM developments of the 1D and 2D SWE based on the augmented Roe method are included. ROMs based on the Lax-Friedrichs method are also given, which will serve to contrast the well-balanced property in the test cases. As indicated in Chapter 2, some high-performance numerical methods used to solve the SWE require numerical corrections in order to avoid non-physical solutions. The fact that FOMs require such corrections does not mean that ROMs do as well. In this section a particular analysis of each of the proposed numerical corrections is presented in order to discern the need to include them or not in the ROM. A procedure is also proposed for their application to ROMs, namely the entropy fix and the wet/dry treatment.

In the framework of realistic SWE applications, it is of great interest to study the possibility to simulate beyond the training values of the parameters. Parameters such as the Manning coefficient that serves to model the bed roughness, the boundary conditions, or the bed geometry are studied in this chapter.

The time prediction presented in Sections 3.4 and 4.3 is applied to a linearized version of the SWE, since its extension to non-linear problems is of very high complexity.

Finally, the efficiency of the ROM is shown by solving three 2D problems. In these cases, all the knowledge presented so far in the thesis is applied. Two of these cases are validated by means of experimental data provided in the literature.

It should be mentioned that all the physical variables that appear in this chapter are in units of the International System. They have been omitted in all cases for the sake of simplicity.

## 5.1 Development of ROMs

This section presents the ROMs of the 1D SWE (2.18) based on the augmented Roe scheme (2.25) and the Lax-Friedrichs schemes (2.26) and (2.27).

### 5.1.1 ARoe-based ROMs

In the 1D version, the Galerkin method [56] is applied to the water depth and the water discharge

$$h_i^n = \sum_{k=1}^{M_{\text{POD}}} \hat{h}_k^n \phi_{i,k}, \quad q_i^n = \sum_{k=1}^{M_{\text{POD}}} \hat{q}_k^n \varphi_{i,k}, \tag{5.1}$$

where $\phi_{i,k}$ and $\varphi_{i,k}$ are basis the functions of each variable. Since the mass and momemtum equations (2.18) are coupled, the dimension $M_{\text{POD}}$ of both reductions is the same. The water velocity, $u = q/h$, is averaged within the time windows.

The **1D ARoe-based ROM** is obtained by applying the Galerkin decomposition (5.1) to the 1D ARoe-based FOM (2.25)

$$\begin{aligned}
\hat{h}_p^{n+1} &= \sum_{i\in J^D} (h_0)_i^n \, \phi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} A_{pk}^h \hat{h}_k^n + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} B_{pk}^h \hat{h}_k^n + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} C_{pk}^h \hat{q}_k^n, \\
\hat{q}_p^{n+1} &= \sum_{i\in J^D} (q_0)_i^n \, \varphi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} A_{pk}^q \hat{q}_k^n + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} B_{pk}^q \hat{h}_k^n + \frac{\Delta t}{\Delta x} \sum_{k=1}^{M_{\text{POD}}} C_{pk}^q \hat{q}_k^n,
\end{aligned} \tag{5.2}$$

where $(h_0)_i^n$ and $(q_0)_i^n$ are the time-dependent Dirichlet BCs imposed, if any, and the coefficients are

$$A_{pk}^h = \sum_{i\in J^I\cup J^S} \phi_{i,k}\phi_{i,p},$$

$$\begin{aligned}
B_{pk}^h = &\sum_{i\in J^S} b_i^h \phi_{i,p} - \frac{1}{2}\sum_{i\in J^I}\left[\delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)_{i+1/2}^w + \delta\phi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)_{i-1/2}^w\right]\phi_{i,p} \\
&+ \frac{g}{2}\sum_{i\in J^I}\left[\tilde{\phi}_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2}\right)_{i+1/2}^w \frac{\delta z_{i+1/2}}{\tilde{c}_{i+1/2}^w} + \tilde{\phi}_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^+}{\tilde{\lambda}_2}\right)_{i-1/2}^w \frac{\delta z_{i-1/2}}{\tilde{c}_{i-1/2}^w}\right]\phi_{i,p} \\
&+ \Delta x \frac{g n_b^2}{2}\sum_{i\in J^I}\left[\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2}\right)_{i+1/2}^w \left(\frac{\tilde{u}|\tilde{u}|}{\tilde{c}}\right)_{i+1/2}^w \frac{\tilde{\phi}_{i+1/2,k}}{\max\left(h_{i+1}^w, h_i^w\right)}\right. \\
&\qquad \left. + \left(\frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^+}{\tilde{\lambda}_2}\right)_{i-1/2}^w \left(\frac{\tilde{u}|\tilde{u}|}{\tilde{c}}\right)_{i-1/2}^w \frac{\tilde{\phi}_{i-1/2,k}}{\max\left(h_i^w, h_{i-1}^w\right)^{4/3}}\right]\phi_{i,p},
\end{aligned}$$

$$C_{pk}^h = \sum_{i\in J^S} c_i^h \phi_{i,p} + \frac{1}{2}\sum_{i\in J^I}\left[\delta\varphi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_2^-}{\tilde{c}}\right)_{i+1/2}^w + \delta\varphi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+ - \tilde{\lambda}_2^+}{\tilde{c}}\right)_{i-1/2}^w\right]\phi_{i,p},$$

$$A_{pk}^q = \sum_{i\in J^I\cup J^S} \varphi_{i,q}\varphi_{i,p},$$

$$
\begin{aligned}
B^q_{qk} = &\sum_{i\in J^S} b^q_i \varphi_{i,p} - \frac{1}{2}\sum_{i\in J^I}\left[\delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}^-_1-\tilde{\lambda}^-_2}{\tilde{c}}\tilde{\lambda}_1\tilde{\lambda}_2\right)^w_{i+1/2} + \delta\phi_{i-1/2,k}\left(\frac{\tilde{\lambda}^+_1-\tilde{\lambda}^+_2}{\tilde{c}}\tilde{\lambda}_1\tilde{\lambda}_2\right)^w_{i-1/2}\right]\varphi_{i,p}\\
&+\frac{g}{2}\sum_{i\in J^I}\left[\tilde{\phi}_{i+1/2,k}\left(\tilde{\lambda}^-_1-\tilde{\lambda}^-_2\right)^w_{i+1/2}\frac{\delta z_{i+1/2}}{\tilde{c}^w_{i+1/2}} + \tilde{\phi}_{i-1/2,k}\left(\tilde{\lambda}^+_1-\tilde{\lambda}^+_2\right)^w_{i-1/2}\frac{\delta z_{i-1/2}}{\tilde{c}^w_{i-1/2}}\right]\varphi_{i,p}\\
&+\Delta x\frac{g n_b^2}{2}\sum_{i\in J^I}\left[\left(\tilde{\lambda}^-_1-\tilde{\lambda}^-_2\right)^w_{i+1/2}\left(\frac{\tilde{u}|\tilde{u}|}{\tilde{c}}\right)^w_{i+1/2}\frac{\tilde{\phi}_{i+1/2,k}}{\max\left(h^w_{i+1},h^w_i\right)}\right.\\
&\qquad\left.+\left(\tilde{\lambda}^+_1-\tilde{\lambda}^+_2\right)^w_{i-1/2}\left(\frac{\tilde{u}|\tilde{u}|}{\tilde{c}}\right)^w_{i-1/2}\frac{\tilde{\phi}_{i-1/2,k}}{\max\left(h^w_i,h^w_{i-1}\right)^{4/3}}\right]\varphi_{i,p},\\
C^q_{qk} = &\sum_{i\in J^S} c^q_i \varphi_{i,p} + \frac{1}{2}\sum_{i\in J^I}\left[\delta\varphi_{i+1/2,k}\left(\frac{\tilde{\lambda}^-_1\tilde{\lambda}_1-\tilde{\lambda}^-_2\tilde{\lambda}_2}{\tilde{c}}\right)^w_{i+1/2} + \delta\varphi_{i-1/2,k}\left(\frac{\tilde{\lambda}^+_1\tilde{\lambda}_1-\tilde{\lambda}^+_2\tilde{\lambda}_2}{\tilde{c}}\right)^w_{i-1/2}\right]\varphi_{i,p}.
\end{aligned}
$$

where $\delta\phi_{i+1/2,k} = \phi_{i+1,k} - \phi_{i,k}$, $\delta\varphi_{i+1/2,k} = \varphi_{i+1,k} - \varphi_{i,k}$ and $\tilde{\phi}_{i+1/2,k} = \left(\phi_{i+1,k} + \phi_{i,k}\right)/2$. Coefficients $b^h, c^h, b^q, c^q$ are given depending on the BCs, similar to (3.9).

In the 2D version, the Galerkin method decomposes the following variables

$$
h^n_i = \sum_{k=1}^{M_{\mathrm{POD}}} \hat{h}^n_k \phi_{i,k}, \quad (q_x)^n_i = \sum_{k=1}^{M_{\mathrm{POD}}} (\hat{q}_x)^n_k \varphi_{i,k}, \quad (q_y)^n_i = \sum_{k=1}^{M_{\mathrm{POD}}} (\hat{q}_y)^n_k \psi_{i,k}, \tag{5.3}
$$

where $\phi_{i,k}$, $\varphi_{i,k}$ and $\psi_{i,k}$ are the functions of the basis of each variable in its reduced space.

Conversely, the velocities $(v_x)^n_i$ and $(v_y)^n_i$ are not reduced, so that computational time can be saved by not going back to the physical space to update its values from those of the conserved variables. Instead, they are time-averaged following the PID method.

The **2D ARoe-based ROM** is obtained by applying the Galerkin decomposition (5.3) to the 2D ARoe-based FOM (2.42)

$$
\begin{aligned}
\hat{h}^{n+1}_p = &\sum_{i\in J^D}(h_0)^n_i\,\phi_{i,p} + \sum_{k=1}^{M_{\mathrm{POD}}} A^h_{pk}\hat{h}^n_k + \Delta t\sum_{k=1}^{M_{\mathrm{POD}}} B^h_{pk}\hat{h}^n_k\\
&+\Delta t\sum_{k=1}^{M_{\mathrm{POD}}} C^h_{pk}(\hat{q}_x)^n_k + \Delta t\sum_{k=1}^{M_{\mathrm{POD}}} D^h_{pk}(\hat{q}_y)^n_k,\\
(\hat{q}_x)^{n+1}_p = &\sum_{i\in J^D}(q_{x,0})^n_i\,\varphi_{i,p} + \sum_{k=1}^{M_{\mathrm{POD}}} A^{q_x}_{pk}(\hat{q}_x)^n_k + \Delta t\sum_{k=1}^{M_{\mathrm{POD}}} B^{q_x}_{pk}\hat{h}^n_k\\
&+\Delta t\sum_{k=1}^{M_{\mathrm{POD}}} C^{q_x}_{pk}(\hat{q}_x)^n_k + \Delta t\sum_{k=1}^{M_{\mathrm{POD}}} D^{q_x}_{pk}(\hat{q}_y)^n_k,
\end{aligned}
\tag{5.4}
$$

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

where the coefficients are

$$A_{pk}^h = \sum_{i\in J^I\cup J^S} \phi_{i,k}\phi_{i,p},$$

$$\begin{aligned} B_{pk}^h = {} & \sum_{i\in J^S} b_i^h \phi_{i,p} - \frac{1}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\phi_{e,k}\left[\tilde{\lambda}_1^-\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)+\tilde{\lambda}_3^-\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^w l_e\phi_{i,p} \\ & + \frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_e^w \frac{\delta z_e}{\tilde{c}_e^w} l_e\phi_{i,p} \\ & + \frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\frac{\tilde{\mathbf{u}}_e^w\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^w|}{\tilde{c}_e^w\max\left(h_i^w,h_j^w\right)^{4/3}}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\right)_e^w \left(\tilde{n}_b^2 d_n\right)_e l_e\phi_{i,p}, \end{aligned}$$

$$C_{pk}^h = \sum_{i\in J^S} c_i^h\phi_{i,p} + \frac{1}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\varphi_{e,k}\left(\frac{\tilde{\lambda}^{-1}-\tilde{\lambda}^{-3}}{\tilde{c}}\right)_e^w (n_x)_e l_e\phi_{i,p},$$

$$D_{pk}^h = \sum_{i\in J^S} d_i^h\phi_{i,p} + \frac{1}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\psi_{e,k}\left(\frac{\tilde{\lambda}^{-1}-\tilde{\lambda}^{-3}}{\tilde{c}}\right)_e^w (n_y)_e l_e\phi_{i,p},$$

$$A_{pk}^{q_x} = \sum_{i\in J^I\cup J^S} \varphi_{i,k}\varphi_{i,p},$$

$$\begin{aligned} B_{pk}^{q_x} = {} & \sum_{i\in J^S} b_i^{q_x}\varphi_{i,p} - \sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\phi_{e,k}\left[\tilde{\lambda}_1^-\frac{\tilde{u}_x-\tilde{c}n_x}{2}\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right. \\ & \left. + \tilde{\lambda}_2^-\left(\tilde{u}_y n_x-\tilde{u}_x n_y\right)n_y + \tilde{\lambda}_3^-\frac{\tilde{u}_x+\tilde{c}n_x}{2}\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^w l_e\varphi_{i,p} \\ & + \frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\frac{\delta z_e}{\tilde{c}_e^n}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_x-\tilde{c}n_x\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_x+\tilde{c}n_x\right)\right]_e^w l_e\varphi_{i,p} \\ & + \frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\frac{\tilde{\mathbf{u}}_e^w\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^w|\left(\tilde{n}_b^2 d_n\right)_e}{\tilde{c}_e^w\max\left(h_i^w,h_j^w\right)^{4/3}}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_x-\tilde{c}n_x\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_x+\tilde{c}n_x\right)\right]_e^w l_e\varphi_{i,p}, \end{aligned}$$

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

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

$$
\begin{aligned}
A_{pk}^{q_y} &= \sum_{i\in J^I\cup J^S} \psi_{i,k}\psi_{i,p},\\
B_{pk}^{q_y} &= \sum_{i\in J^S} b_i^{q_y}\psi_{i,p} - \sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\phi_{e,k}\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2}\left(1+\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right.\\
&\quad\left. -\tilde{\lambda}_2^-\left(\tilde{u}_y n_x-\tilde{u}_x n_y\right)n_x+\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2}\left(1-\frac{\tilde{\mathbf{u}}\cdot\mathbf{n}}{\tilde{c}}\right)\right]_e^w l_e\psi_{i,p}\\
&\quad+\frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\frac{\delta z_e}{\tilde{c}_e^w}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_y-\tilde{c}n_y\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_y+\tilde{c}n_y\right)\right]_e^w l_e\psi_{i,p}\\
&\quad+\frac{g}{2}\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\tilde{\phi}_{e,k}\frac{\tilde{\mathbf{u}}_e^w\cdot\mathbf{n}_e|\tilde{\mathbf{u}}_e^w|\left(\tilde{n}_b^2 d_n\right)_e}{\tilde{c}_e^w\max\left(h_i^w,h_j^w\right)^{4/3}}\left[\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\left(\tilde{u}_y-\tilde{c}n_y\right)-\frac{\tilde{\lambda}_3^-}{\tilde{\lambda}_3}\left(\tilde{u}_y+\tilde{c}n_y\right)\right]_e^w l_e\psi_{i,p},\\
C_{pk}^{q_y} &= \sum_{i\in J^S} c_i^{q_y}\psi_{i,p}+\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\varphi_{e,k}\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2\tilde{c}}n_x+\tilde{\lambda}_2^- n_x n_y-\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2\tilde{c}}n_x\right]_e^w l_e\psi_{i,p},\\
D_{pk}^{q_y} &= \sum_{i\in J^S} d_i^{q_y}\psi_{i,p}+\sum_{i\in J^I}\frac{1}{S_i}\sum_{e=1}^{I_f}\delta\psi_{e,k}\left[\tilde{\lambda}_1^-\frac{\tilde{u}_y-\tilde{c}n_y}{2\tilde{c}}n_y-\tilde{\lambda}_2^- n_x n_x-\tilde{\lambda}_3^-\frac{\tilde{u}_y+\tilde{c}n_y}{2\tilde{c}}n_y\right]_e^w l_e\psi_{i,p}.
\end{aligned}
$$

Coefficients $b^h, c^h, d^h, b^{q_x}, c^{q_x}, d^{q_x}, b^{q_y}, c^{q_y}, d^{q_y}$ depend on the BCs imposed and are determined similar to (3.9).

### 5.1.2 1D Lax-Friedrichs-based ROM

The 1D Lax-Friedrichs-based FOM (2.26) allows the direct application of the Galerkin method to $h$, $q$ and $u$, so that no linearization is needed, except for the friction source term. The Galerkin method reads

$$
h_i^n=\sum_{k=1}^{M_{\mathrm{POD}}}\hat{h}_k^n\phi_{i,k},\quad q_i^n=\sum_{k=1}^{M_{\mathrm{POD}}}\hat{q}_k^n\varphi_{i,k},\quad u_i^n=\sum_{k=1}^{M_{\mathrm{POD}}}\hat{u}_k^n\psi_{i,k},\tag{5.5}
$$

where $\phi_{i,k}$, $\varphi_{i,k}$ and $\psi_{i,k}$ are the functions of the basis of the reduced space for the water depth, water discharge and water velocity, respectively.

The Lax-Friedrichs-based ROM is

$$
\begin{aligned}
\hat{h}_p^{n+1} &= \sum_{i\in J^D}\left(h_0\right)_i^n\phi_{i,p}+\sum_{k=1}^{M_{\mathrm{POD}}}A_{pk}^h\hat{h}_k^n+\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}}B_{pk}^h\hat{q}_p^n+C_p^h,\\
\hat{q}_p^{n+1} &= \sum_{i\in J^D}\left(q_0\right)_i^n\varphi_{i,p}+\sum_{k=1}^{M_{\mathrm{POD}}}\hat{q}_k^n A_{pk}^q+\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{q}_k^n\sum_{q=1}^{M_{\mathrm{POD}}}\hat{u}_q^n B_{pkq}^q\\
&\quad+\frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\mathrm{POD}}}\hat{h}_k^n\sum_{q=1}^{M_{\mathrm{POD}}}\hat{h}_q^n C_{pkq}^q+\Delta t\sum_{k=1}^{M_{\mathrm{POD}}}D_{pk}^q\hat{h}_k^n,
\end{aligned}\tag{5.6}
$$

where the coefficients are

$$
\begin{aligned}
A^h_{pk} &= \sum_{i\in J^I\cup J^S} \phi_{i,k}\phi_{i,p} + \sum_{i\in J^S} a^h_i \phi_{i,k} + \frac{1}{2}\xi \sum_{i\in J^I} \left(\phi_{k,i+1} - 2\phi_{k,i} + \phi_{k,i-1}\right)\phi_{p,i},\\
B^h_{pk} &= \sum_{i\in J^S} b^h_i \phi_{i,k} - \frac{1}{2}\sum_{i\in J^I} \left(\varphi_{i+1,k} - \varphi_{i-1,k}\right)\phi_{i,p},\\
C^h_{p} &= \sum_{i\in J^S} c^h_i \phi_{i,p} + \frac{1}{2}\xi \sum_{i\in J^I} \left(z_{i+1} - 2z_i + z_{i-1}\right)\phi_{i,p},\\
A^q_{pk} &= \sum_{i\in J^I\cup J^S} \varphi_{i,k}\varphi_{i,p} + \sum_{i\in J^S} a^q_i \varphi_{i,p} + \frac{1}{2}\xi \sum_{i\in J^I} \left(\varphi_{i+1,k} - 2\varphi_{i,k} + \varphi_{i-1,k}\right)\varphi_{i,p},\\
B^q_{pkq} &= \sum_{i\in J^S} b^q_i \varphi_{i,p} - \frac{1}{2}\sum_{i\in J^I} \left(\varphi_{i+1,k}\psi_{i+1,q} - \varphi_{i-1,k}\psi_{i-1,q}\right)\varphi_{i,p},\\
C^q_{pkq} &= \sum_{i\in J^S} c^q_i \varphi_{i,p} - \frac{1}{4} g \sum_{i\in J^I} \left(\phi_{i+1,k}\phi_{i+1,q} - \phi_{i-1,k}\phi_{i-1,q}\right)\varphi_{i,p},\\
D^q_{pk} &= \sum_{i\in J^S} d^q_i \varphi_{i,p} - \frac{1}{\Delta x}\frac{1}{2} g \sum_{i\in J^I} \left(\tilde{\phi}_{i+1/2,k}\delta z_{i+1/2} + \tilde{\phi}_{i-1/2,k}\delta z_{i-1/2}\right)\varphi_{i,p}\\
&\quad - \frac{g n_b^2}{2}\sum_{i\in J^I}\left[\tilde{\phi}_{i+1/2,k}\frac{u^w_{i+1/2}|u^w_{i+1/2}|}{\max\left(h^w_{i+1}, h^w_i\right)^{4/3}} + \tilde{\phi}_{i-1/2,k}\frac{u^w_{i-1/2}|u^w_{i-1/2}|}{\max\left(h^w_i, h^w_{i-1}\right)^{4/3}}\right]\varphi_{i,p}.
\end{aligned}
$$

Coefficients $a^h, b^h, c^h, a^q, b^q, c^q, d^q$ are given depending on the BCs, similar to (3.9).

### 5.1.3 1D well-balanced Lax-Friedrichs-based ROM

As shown in Chapter 2, the optimized version of the Lax-Friedrichs numerical scheme (2.26) is not well-balanced and the additional term (2.27) needs to be added. The formulation of the well-balanced Lax-Friedrichs-based ROM is very similar to (5.6), but with the new term in the mass equation due to the well-balancing correction that needs time averaging

$$
\begin{aligned}
\hat{h}^{n+1}_p &= \sum_{i\in J^D} \left(h_0\right)^n_i \phi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} A^h_{pk}\hat{h}^n_k + \frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\text{POD}}} B^h_{pk}\hat{q}^n_p + C^h_p, \qquad (5.7)\\
\hat{q}^{n+1}_p &= \sum_{i\in J^D} \left(q_0\right)^n_i \varphi_{i,p} + \sum_{k=1}^{M_{\text{POD}}} \hat{q}^n_k A^q_{pk} + \frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\text{POD}}}\hat{q}^n_k \sum_{q=1}^{M_{\text{POD}}} \hat{u}^n_q B^q_{pkq}\\
&\quad + \frac{\Delta t}{\Delta x}\sum_{k=1}^{M_{\text{POD}}}\hat{h}^n_k \sum_{q=1}^{M_{\text{POD}}}\hat{h}^n_q C^q_{pkq} + \Delta t \sum_{k=1}^{M_{\text{POD}}} D^q_{pk}\hat{h}^n_k,
\end{aligned}
$$

where the coefficients are

$$
\begin{aligned}
A^h_{pk} &= \sum_{i\in J^I\cup J^S} \phi_{i,k}\phi_{i,p} + \sum_{i\in J^S} a^h_i \phi_{i,k} + \frac{1}{2}\xi \sum_{i\in J^I} \left(\phi_{k,i+1} - 2\phi_{k,i} + \phi_{k,i-1}\right)\phi_{p,i}\\
&\quad + \frac{\nu}{g}\sum_{i\in J^I}\left(\frac{\phi_{i+1,k}\left(u^w_{i+1}\right)^2 - \phi_{i,k}\left(u^w_i\right)^2}{\bar{h}^w_{i+1} + \bar{h}^w_i} - \frac{\phi_{i,k}\left(u^w_i\right)^2 - \phi_{i-1,k}\left(u^w_{i-1}\right)^2}{\bar{h}^w_i + \bar{h}^w_{i-1}}\right)\phi_{i,p},
\end{aligned}
$$

$$
\begin{aligned}
B^h_{pk} &= \sum_{i\in J^S} b^h_i \phi_{i,k} - \frac{1}{2}\sum_{i\in J^I} \left(\varphi_{i+1,k} - \varphi_{i-1,k}\right)\phi_{i,p},\\
C^h_p &= \sum_{i\in J^S} c^h_i \phi_{i,p} \frac{1}{2}\xi \sum_{i\in J^I} \left(z_{i+1} - 2z_i + z_{i-1}\right)\phi_{i,p},\\
A^q_{pk} &= \sum_{i\in J^I\cup J^S} \varphi_{i,k}\varphi_{i,p} + \sum_{i\in J^S} a^q_i \varphi_{i,p} + \frac{1}{2}\xi \sum_{i\in J^I} \left(\varphi_{i+1,k} - 2\varphi_{i,k} + \varphi_{i-1,k}\right)\varphi_{i,p},\\
B^q_{pkq} &= \sum_{i\in J^S} b^q_i \varphi_{i,p} - \frac{1}{2}\sum_{i\in J^I} \left(\varphi_{i+1,k}\psi_{i+1,q} - \varphi_{i-1,k}\psi_{i-1,q}\right)\varphi_{i,p},\\
C^q_{pkq} &= \sum_{i\in J^S} c^q_i \varphi_{i,p} - \frac{1}{4} g \sum_{i\in J^I} \left(\phi_{i+1,k}\phi_{i+1,q} - \phi_{i-1,k}\phi_{i-1,q}\right)\varphi_{i,p},\\
D^q_{pk} &= \sum_{i\in J^S} d^q_i \varphi_{i,p} - \frac{1}{\Delta x}\frac{1}{2} g \sum_{i\in J^I} \left(\tilde{\phi}_{i+1/2,k}\delta z_{i+1/2} + \tilde{\phi}_{i-1/2,k}\delta z_{i-1/2}\right)\varphi_{i,p}\\
&\quad - \frac{g n_b^2}{2}\sum_{i\in J^I}\left[\tilde{\phi}_{i+1/2,k}\frac{u^w_{i+1/2}|u^w_{i+1/2}|}{\max\left(h^w_{i+1}, h^w_i\right)^{4/3}} + \tilde{\phi}_{i-1/2,k}\frac{u^w_{i-1/2}|u^w_{i-1/2}|}{\max\left(h^w_i, h^w_{i-1}\right)^{4/3}}\right]\varphi_{i,p}.
\end{aligned}
$$

Coefficients $a^h, b^h, c^h, a^q, b^q, c^q, d^q$ are given depending on the BCs, similar to (3.9).

## 5.2 Numerical corrections

In this section, it is checked whether the ARoe-based ROM needs numerical corrections or it is able to inherit them when trained with corrected solutions. For this purpose, different test cases are considered and it is examined if the training solution is recovered with machine accuracy using the ROM with and without the corresponding numerical correction. It has been found that there are some corrections that are required in the ROM and others that do not. Therefore, the procedure for their inclusion in the ROM is also presented.

Likewise, well-balancing has also been considered in this section, because, although the ARoe-based FOM (2.25) is already well-balanced, there are other numerical schemes that are not, such as the Lax-Friedrichs-based FOM (2.26) that needs a correction in order to calculate well-balanced solutions (2.27).

### 5.2.1 Well-balancing

The well-balanced property preserves the quiescent equilibrium, as explained in Chapter 2. In this case, it is studied whether ROMs must also be well-balanced or whether it is sufficient for them to be trained with solutions calculated from well-balanced FOMs. For this purpose, the following two test cases are proposed.

**Test case 24. 1D equilibrium Riemann problem**

The ARoe-based FOM (2.25) used is well-balanced [127]. On the contrary, the LF-based FOM (2.26) proposed in [26] is not well-balanced. Therefore, a correction term must be added to reproduce properly equilibrium problems, as indicated in Sections 2.4.3 and 5.1.3, where the WLF-based FOM is introduced (2.27).

The performance of ROMs is analyzed by means of the 9 subcases shown in Table 5.1. According to [88], Subcases 1, 5 and 9 are consistent and the remaining subcases are inconsistent.

| Subcase | FOM | ROM |
|---|---|---|
| 1 | LF | LF |
| 2 | LF | WLF |
| 3 | LF | ARoe |
| 4 | WLF | LF |
| 5 | WLF | WLF |
| 6 | WLF | ARoe |
| 7 | ARoe | LF |
| 8 | ARoe | WLF |
| 9 | ARoe | ARoe |

Table 5.1: TC24. Subcases considered.

The equilibrium Riemann problem RP2 proposed in [132] is here considered. The time-space domain of the case is defined as $(x,t) \in [0, 0.5] \times [0, 0.01]$. The solution is discontinuous

$$h(x,0) = \begin{cases} 1, & \text{if } x < 0.25, \\ 0.6245627691, & \text{if } x \geq 0.25, \end{cases} \quad q(x,0) = 1. \tag{5.8}$$

and it satisfies the Rankine-Hugoniot condition [127]. The bed slope is defined as

$$z(x) = \begin{cases} 0, & \text{if } x < 0.25, \\ 0.3, & \text{if } x \geq 0.25, \end{cases} \tag{5.9}$$

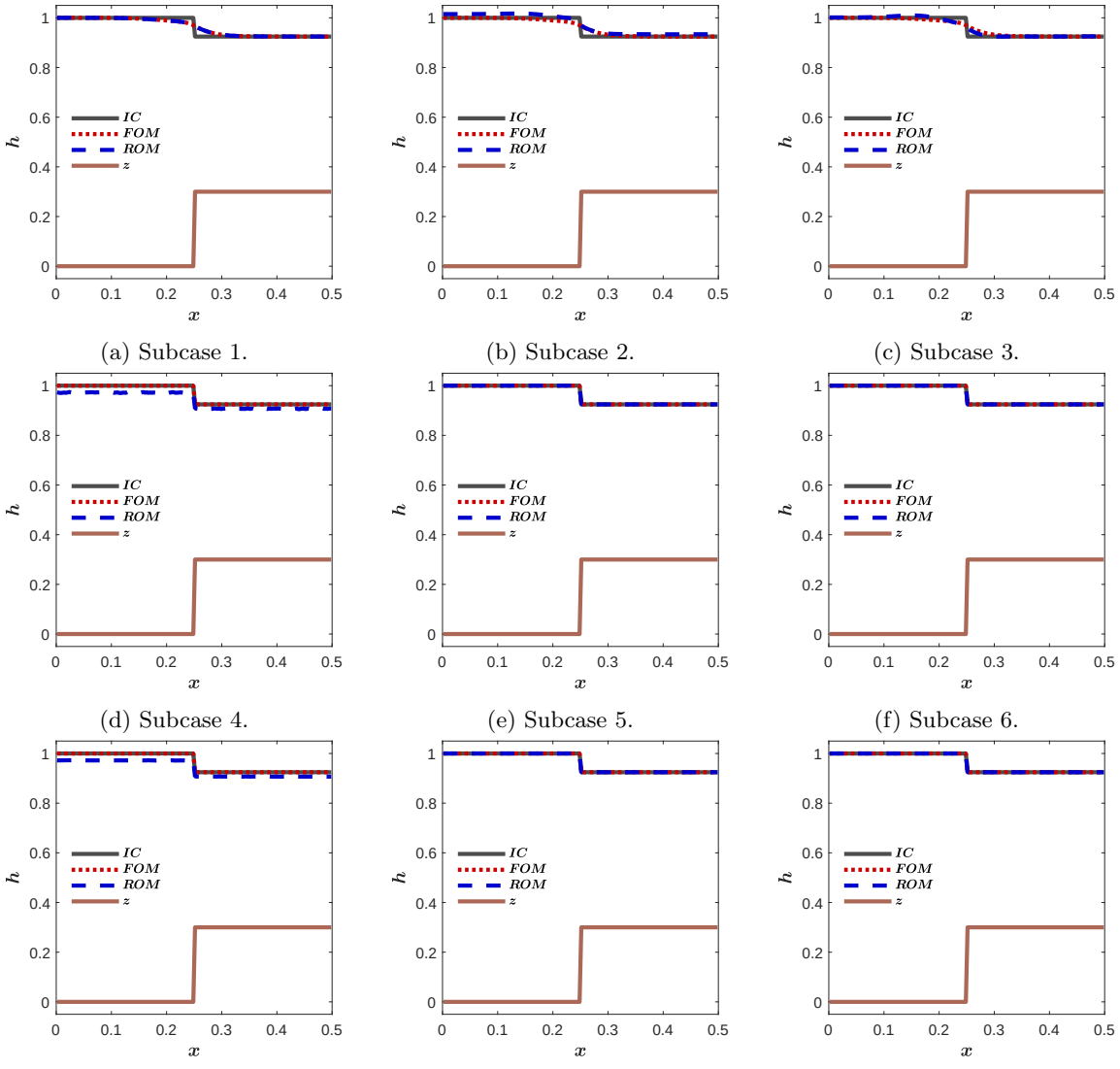
and free boundary conditions are imposed.

The spatial domain is discretized using $I_x = 160$ cells, and the time steps are computed dynamically, with CFL = 0.1. The ROMs are solved using 3 POD modes and 89 time windows. In all subcases the settings are those shown in Table 5.2.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | $I_x$ | CFL | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.01 | Eq. (5.8) | Free | Eq. (5.9) | 0 | 160 | 0.1 | 267 | 3 | 89 |

Table 5.2: TC24. Problem settings.

In essence, ROMs reproduce properly the solutions of the FOM in Subcases 1, 5, 6, 8 and 9, as shown in Figures 5.1 and 5.2, but the solution of Subcase 1 is not well-balanced and therefore does not make physical sense and does not have to be taken into account. Similarly, Subcases 2 and 3 are trained by the Lax-Friedrichs-based FOM and are not well-balanced. In the remaining Subcases 4 and 7 in which the Lax-Friedrichs-based ROM is trained by the well-balanced Lax-Friedrichs-based FOM and the ARoe-based FOM, respectively, satisfactory results are found. In the end, it can be concluded that ROMs must have been developed and trained from well-balanced FOMs.

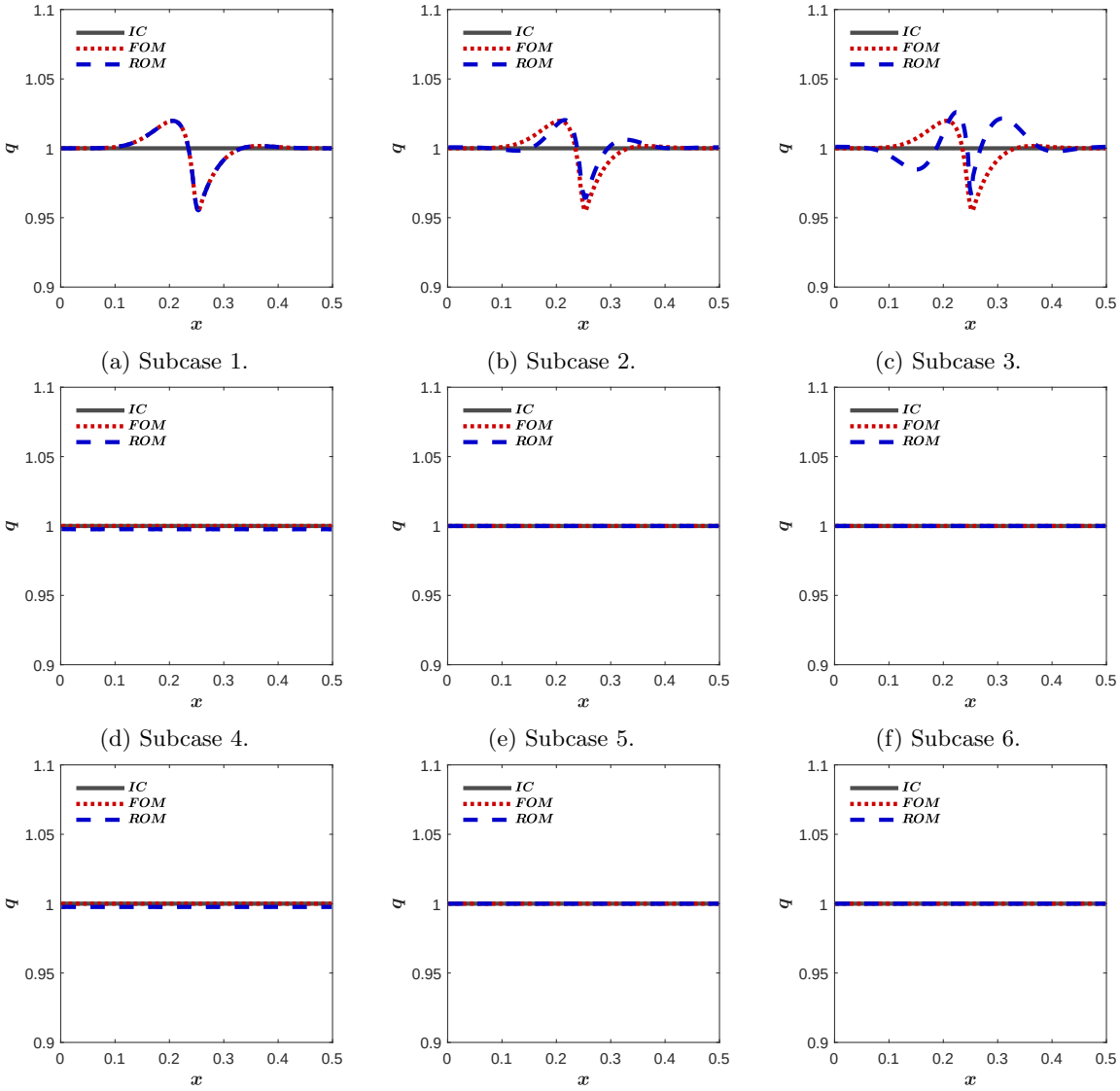



(a) Subcase 1. (b) Subcase 2. (c) Subcase 3.

(d) Subcase 4. (e) Subcase 5. (f) Subcase 6.

(g) Subcase 7. (h) Subcase 8. (i) Subcase 9.

Figure 5.1: TC24. Solutions $h$ for all subcases considered.




(a) Subcase 1. (b) Subcase 2. (c) Subcase 3.

(d) Subcase 4. (e) Subcase 5. (f) Subcase 6.

(g) Subcase 7. (h) Subcase 8. (i) Subcase 9.

Figure 5.2: TC24. Solutions $q$ for all subcases considered.

### Test case 25. 1D transient RP

This case takes a further step in the study of relationship between the well-balancing property and ROMs. To this end, the transient RP1 proposed in [132] is considered. The time-space domain of the case is defined as $(x,t) \in [0,0.5] \times [0,0.02]$. The initial condition is defined as

$$h(x,0) = \begin{cases} 1, & \text{if } x < 0.25, \\ 0.1614067989, & \text{if } x \geq 0.25, \end{cases} \quad q(x,0) = 0. \tag{5.10}$$

The bed slope is defined as

$$z(x) = \begin{cases} 0, & \text{if } x < 0.25, \\ 0.05, & \text{if } x \geq 0.25, \end{cases} \tag{5.11}$$

and free boundary conditions are imposed.

Again, combinations of different FOMs and ROMs are studied, but, since the Lax-Friedrichs method is not well-balanced as seen in the previous test case TC24, it is omitted in the current test case and only the combinations listed in the Table 5.3 are considered. Subcases 1 and 4 are consistent, because the training FOM is the same as the developing FOM; and, therefore, Subcases 2 and 3 are inconsistent.

| Subcase | FOM | ROM |
|---|---|---|
| 1 | WLF | WLF |
| 2 | WLF | ARoe |
| 3 | ARoe | WLF |
| 4 | ARoe | ARoe |

Table 5.3: TC25. Subcases considered.

The spatial domain is discretized using $I_x = 320$ volume cells, and the time steps are computed dynamically according to (2.5). The CFL number is varied to test the performance of the ROMs. The ROM is solved using 5 POD modes. In all subcases the settings are those shown in Table 5.4; and the numerical time steps are indicated in Table 5.5.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | $I_x$ | CFL | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.02 | Eq. (5.10) | Free | Eq. (5.11) | 0 | 320 | 0.1/0.5/0.9 | 5 | 29 |

Table 5.4: TC25. Problem settings.

| | CFL | | |
|---|---|---|---|
| Subcase | 0.1 | 0.5 | 0.9 |
| 1 | 545 | 102 | 58 |
| 2 | 545 | 102 | 58 |
| 3 | 514 | 104 | 58 |
| 4 | 514 | 104 | 58 |

Table 5.5: TC25. Numerical time steps.

Figures 5.3 and 5.4 show the solutions of $h$ and $q$ respectively computed with the different subcases and for the different values of the CFL number: 0.1, 0.5 and 0.9.

The most basic aspect that is worth mentioning in relation to these results is that the FOMs show different dependencies on the CFL number: while the well-balanced Lax-Friedrichs-based FOM is highly dissipative for low CFL numbers, as shown in Subcases 1 and 2 in Figures

5.4a and 5.4b, the ARoe-based FOM shows hardly any numerical dissipation for different CFL values.

Regarding the water depth, all subcases obtain proper results for high CFL numbers, as shown in Figure 5.3, even in the inconsistent Subcases 2 and 3. As the CFL number decreases, the results of Subcases 2 and 3 get worse, leading to solutions with no physical meaning with CFL = 0.1. These solutions underestimate the value of the water depth to the left of the rarefaction wave, where no perturbation is expected.

In terms of water discharge, the performance of the ROMs is slightly worse as can be seen in Figure 5.4. This difference with the water depth results may be due to the time averaging approach. The inconsistent Subcases 2 and 3 are not able to ensure a proper calculation of the water discharge, as they under- or overestimate it for all values of the CFL number (except Subcase 2 with CFL = 0.9, which incidentally is correct). Finally, Subcase 1 shows good agreement between the ROM and FOM solutions, but both differ from the exact solution when the CFL is reduced.

The differences of the solutions of all subcases between the FOM and the ROM are included in Table 5.6. All subcases obtain accurate solutions, but, as expected, the differences in the consistent Subcases 1 and 4 are smaller than in the inconsistent Subcases 2 and 3. Although Subcase 1 obtains better differences than Subcase 4, it must be recalled that these solutions are very different from the exact solution due to the high numerical dissipation. Subcase 4 shows an increasing tendency of $\|d^q\|_1$ when the CFL number increases. This may be related to the time averaging approach, because the larger the CFL number, the larger the time step and therefore the wider the time windows, which may imply a slight loss of accuracy in the solution. In spite of this, Subcase 4, with approximately constant differences, is the most suitable method to compute proper solutions to this problem, as seen in Figures 5.3 and 5.4.

| | | CFL | | |
|---|---|---|---|---|
| | Subcase | 0.1 | 0.5 | 0.9 |
| $\|d^h\|_1$ | 1 | $8.22 \cdot 10^{-5}$ | $5.25 \cdot 10^{-5}$ | $2.22 \cdot 10^{-4}$ |
| | 2 | $3.35 \cdot 10^{-3}$ | $1.01 \cdot 10^{-3}$ | $8.76 \cdot 10^{-4}$ |
| | 3 | $1.31 \cdot 10^{-2}$ | $2.77 \cdot 10^{-3}$ | $1.92 \cdot 10^{-3}$ |
| | 4 | $1.89 \cdot 10^{-4}$ | $4.52 \cdot 10^{-4}$ | $3.22 \cdot 10^{-4}$ |
| $\|d^q\|_1$ | 1 | $5.51 \cdot 10^{-5}$ | $5.74 \cdot 10^{-4}$ | $3.28 \cdot 10^{-3}$ |
| | 2 | $5.74 \cdot 10^{-2}$ | $9.78 \cdot 10^{-3}$ | $2.14 \cdot 10^{-3}$ |
| | 3 | $5.10 \cdot 10^{-2}$ | $1.54 \cdot 10^{-2}$ | $1.19 \cdot 10^{-2}$ |
| | 4 | $6.73 \cdot 10^{-4}$ | $8.13 \cdot 10^{-4}$ | $4.00 \cdot 10^{-3}$ |

Table 5.6: TC25. $\|d^h\|_1$ and $\|d^q\|_1$ computed with the ROM measured with respect to the FOM solution.

Tables 5.7 and 5.8 contain all the CPU times required by the FOMs and all subcases, respectively, to compute their solutions. The ROMs, in all subcases, are faster than their corresponding FOMs.

| | CFL | | |
|---|---|---|---|
| FOM | 0.1 | 0.5 | 0.9 |
| WLF | $5.75 \cdot 10^{-2}$ | $1.91 \cdot 10^{-2}$ | $1.33 \cdot 10^{-2}$ |
| ARoe | $7.41 \cdot 10^{-2}$ | $1.94 \cdot 10^{-2}$ | $1.52 \cdot 10^{-2}$ |

Table 5.7: TC25. CPU times of the WLF and the ARoe.

| | CFL | | |
|---|---|---|---|
| Subcase | 0.1 | 0.5 | 0.9 |
| 1 | $5.03 \cdot 10^{-3}$ | $1.54 \cdot 10^{-3}$ | $8.84 \cdot 10^{-4}$ |
| 2 | $4.74 \cdot 10^{-3}$ | $1.46 \cdot 10^{-3}$ | $8.30 \cdot 10^{-4}$ |
| 3 | $7.72 \cdot 10^{-3}$ | $1.57 \cdot 10^{-3}$ | $8.78 \cdot 10^{-4}$ |
| 4 | $7.32 \cdot 10^{-3}$ | $1.52 \cdot 10^{-3}$ | $8.41 \cdot 10^{-4}$ |

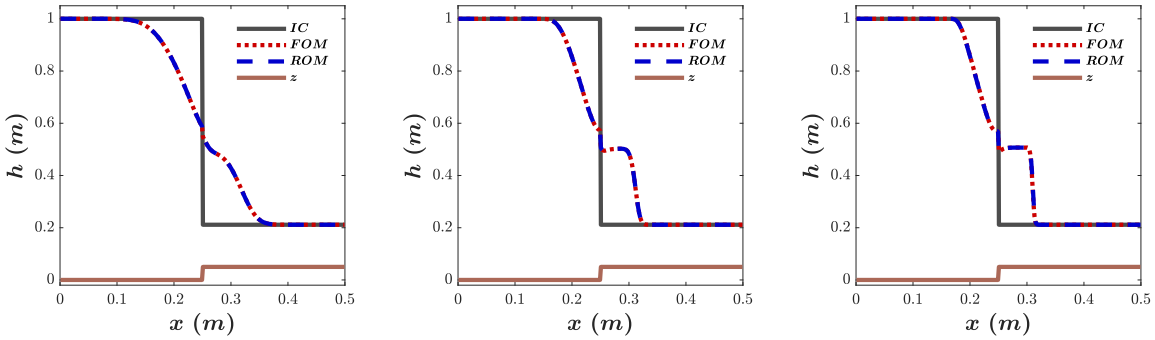
Table 5.8: TC25. CPU times of all subcases.

In terms of speed-up, all subcases reach at least one order of magnitude with respect to their corresponding FOMs, as can be seen in Table 5.9.

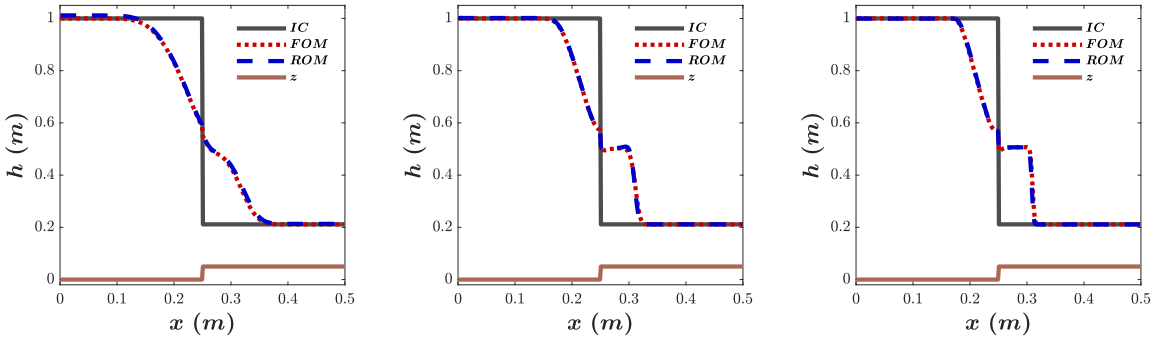
| | CFL | | |
|---|---|---|---|
| Subcase | 0.1 | 0.5 | 0.9 |
| 1 | $\times 11$ | $\times 12$ | $\times 15$ |
| 2 | $\times 12$ | $\times 13$ | $\times 16$ |
| 3 | $\times 10$ | $\times 12$ | $\times 17$ |
| 4 | $\times 10$ | $\times 13$ | $\times 18$ |

Table 5.9: TC25 Speed-ups of all subcases.

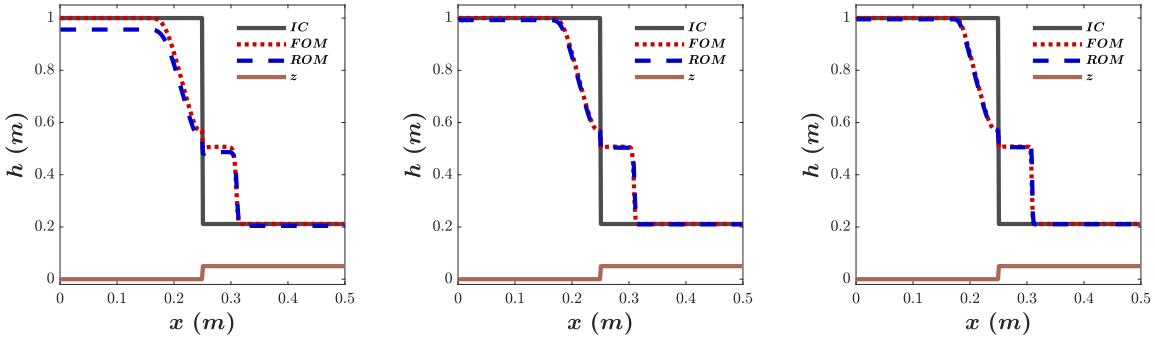
Well-balanced FOMs such as the well-balanced Lax-Friedrichs-based FOM and the ARoe-based FOM can be used to train their corresponding well-balanced ROMs to solve transient problems with high accuracy. However, the well-balanced Lax-Friedrichs-based FOM is too dissipative, so that its solutions do not match up to what they should be. The numerical results of this test case show that the ARoe-based ROM obtains highly efficient results when trained with the solutions of the ARoe-based FOM independently of the CFL number. If trained with WLF solutions computed by the well-balanced Lax-Friedrichs-based FOM, non-physical results are obtained for low CFL numbers.



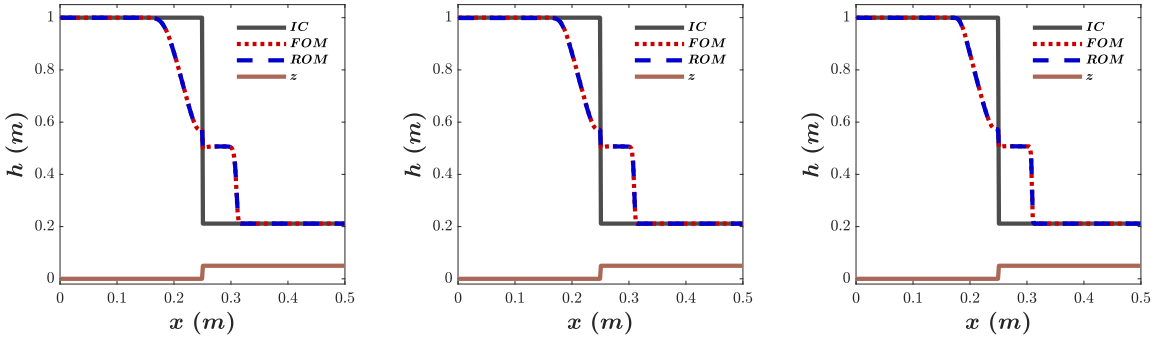

(a) Subcase 1: CFL = 0.1. (b) Subcase 1: CFL = 0.5. (c) Subcase 1: CFL = 0.9.




(d) Subcase 2: CFL = 0.1. (e) Subcase 2: CFL = 0.5. (f) Subcase 2: CFL = 0.9.

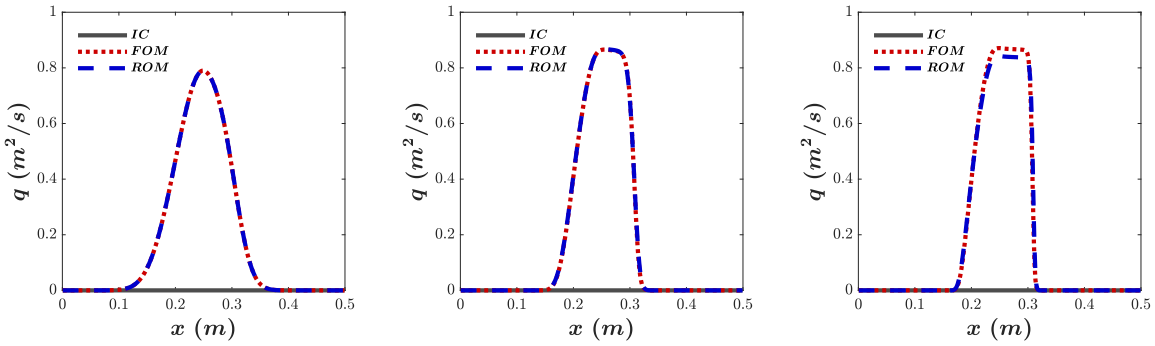



(g) Subcase 3: CFL = 0.1. (h) Subcase 3: CFL = 0.5. (i) Subcase 3: CFL = 0.9.

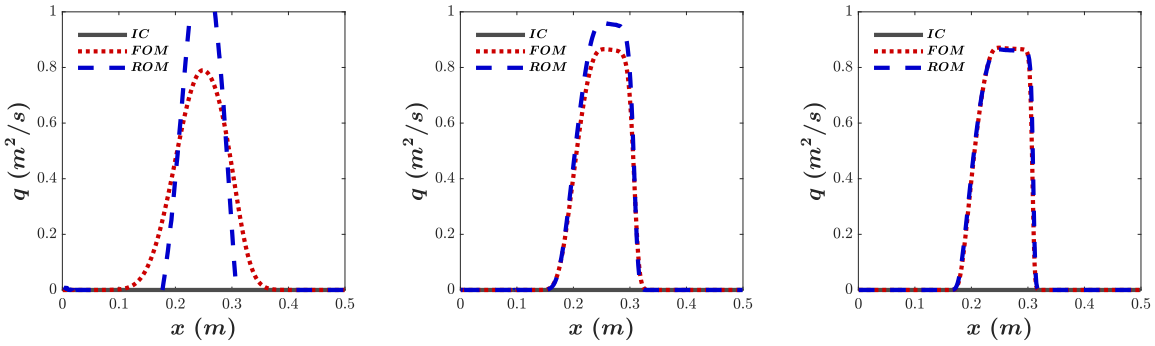



(j) Subcase 4: CFL = 0.1. (k) Subcase 4: CFL = 0.5. (l) Subcase 4: CFL = 0.9.

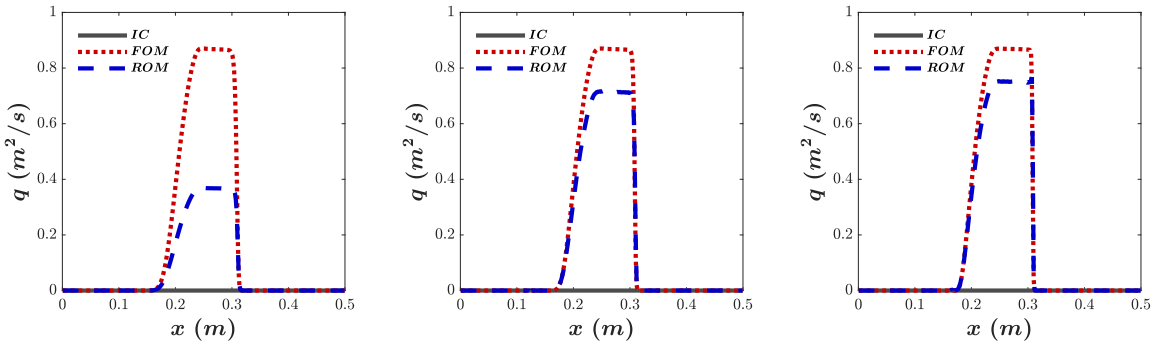
Figure 5.3: TC25. Results of $h$ computed by different subcases.



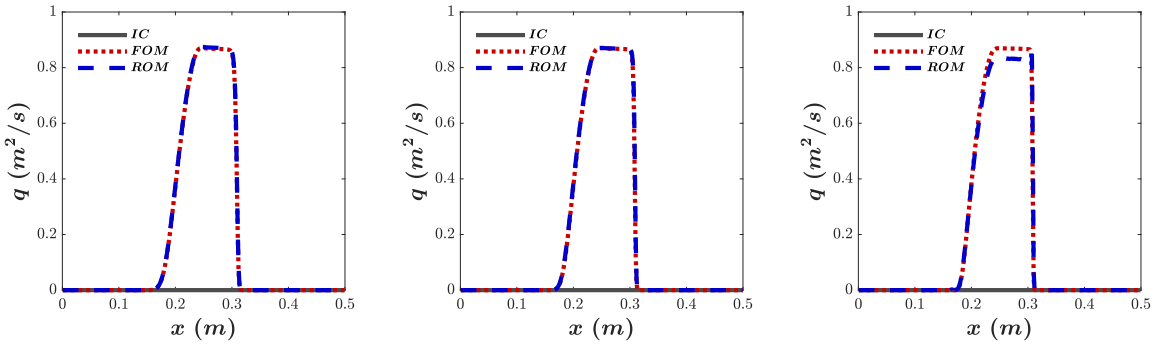

(a) Subcase 1: CFL = 0.1. (b) Subcase 1: CFL = 0.5. (c) Subcase 1: CFL = 0.9.




(d) Subcase 2: CFL = 0.1. (e) Subcase 2: CFL = 0.5. (f) Subcase 2: CFL = 0.9.




(g) Subcase 3: CFL = 0.1. (h) Subcase 3: CFL = 0.5. (i) Subcase 3: CFL = 0.9.




(j) Subcase 4: CFL = 0.1. (k) Subcase 4: CFL = 0.5. (l) Subcase 4: CFL = 0.9.

Figure 5.4: TC25. Results of $q$ computed by different subcases.

### 5.2.2 Source term correction to ensure positive water depth

As indicated in Section 2.4.4, when using the ARoe-based FOM (2.25), it is necessary to avoid the water depth becoming negative. However, the ROM does not need any additional correction when solved. It just needs the training snapshots to be properly corrected, as shown in the following test case.

**Test case 26. 1D dam break problem**

A dam-break problem is to be solved to test whether this correction is necessary or not. The time-space domain is $(x,t) \in [0,6] \times [0,4]$ and the initial conditions are defined as follows

$$h(x,0) = \begin{cases} 2, & x \leq 3, \\ 1, & 3 < x, \end{cases} \quad q(x,0) = 0. \tag{5.12}$$

The bed is given by

$$z(x) = \begin{cases} 0.1, & x \leq 3, \\ 0.4, & 3 < x, \end{cases} \tag{5.13}$$

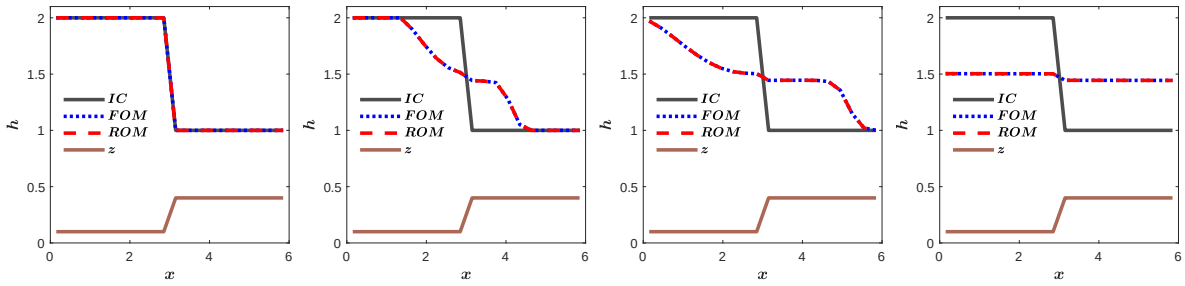
and the Manning coefficient is $n_b = 0.01$. Free boundary conditions are considered. The spatial domain is divided into $I_x = 20$ cells; and the rest of the settings are shown in Table 5.10.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 4 | Eq. (5.12) | Free | Eq. (5.13) | 0.01 | 0.9 | 20 | 74 | 20 | 74 |

Table 5.10: TC26: problem settings.

The training solutions have been computed by the ARoe-based FOM (2.25). The ARoe-based ROM (5.2) has used the maximum number of POD modes and with one snapshot per time window to check whether or not it requires the inclusion of this correction. A time window is provided for each time step so that the time averages of the variables do not introduce error and the training solution can be recovered with machine accuracy. In this way, it can be checked whether the error made by the ROM is due to the absence of the numerical correction. Figure 5.5 shows the high concordance between the solutions of the FOM (in blue) and the ROM (in red) at different time. This is verified by the differences between them measured using $\|d\|_1$ and $\|d\|_\infty$ (3.11), shown in Figure 5.6 and in Table 5.11. The ARoe-based ROM is able to recover the training solution with machine accuracy. So, it can be concluded that this correction is not necessary when solving the ROM, which is faster than the FOM even when using the maximum number of POD modes.

This example has also proven the ability of the ROM to reconstruct solutions with machine accuracy in cases with bed and friction source terms.




(a) $h : t = 0.$ (b) $h : t = 0.29.$ (c) $h : t = 0.57.$ (d) $h : t = 3.86.$

(e) $t = 0.$ (f) $q : t = 0.29.$ (g) $q : t = 0.57.$ (h) $q : t = 3.86.$

Figure 5.5: TC26. ROM solutions at different time.

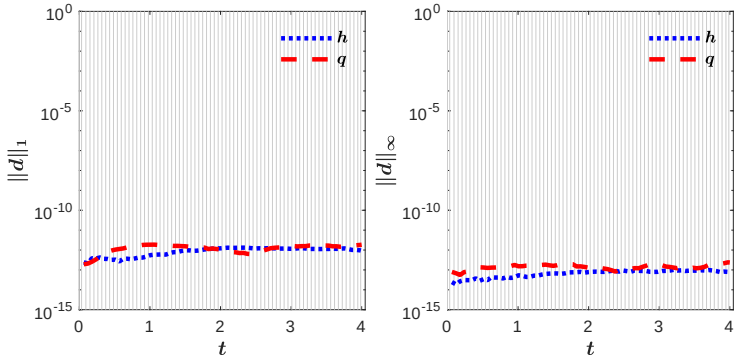



Figure 5.6: TC26. differents between the FOM and the ROM solutions using $\|d\|_1$ and $\|d\|_\infty$.

| $\|d^h\|_1$ | $\|d^q\|_1$ | $\|d^h\|_\infty$ | $\|d^q\|_\infty$ | $\tau_{\text{CPU}}^{\text{FOM}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
|---|---|---|---|---|---|---|
| $9.20 \cdot 10^{-13}$ | $1.85 \cdot 10^{-12}$ | $7.34 \cdot 10^{-14}$ | $2.35 \cdot 10^{-13}$ | $5.50 \cdot 10^{-2}$ | $5.00 \cdot 10^{-3}$ | $\times 11$ |

Table 5.11: TC26. Efficiency results.

### 5.2.3 Friction correction to avoid reverse flow

There are some problems where the friction source term, due to the magnitude of the Manning coefficient, can cause the flow to change direction without physical sense. In these cases it is necessary to take into account the correction on the friction term, according to which, if the intermediate states given by (2.30) have opposite sign, it is necessary to redefine the friction term as indicated in (2.31).

The reduced version of this correction requires special treatment which is included in the ofline part of the ROM. This is explained below with an example.

The water depth is updated in time according to (2.25) as follows

$$h_i^{n+1} = h_i^n + ... + \frac{\Delta t}{\Delta x}\left[\left(\tilde{\beta}_f\right)^n_{i+1/2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\right)^n_{i+1/2} + \left(\tilde{\beta}_f\right)^n_{i-1/2}\left(\frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1}\right)^n_{i-1/2}\right].$$

Suppose that at the $i+/2$-th wall the condition indicated in (2.31) is fulfilled, so that the friction term $\left(\tilde{\beta}_f\right)^n_{i+1/2}$ is replaced by the corresponding intermediate state $(q^{**})^n_{i+1/2}$ defined in (2.30), such that

$$h_i^{n+1} = h_i^n + ... + \frac{\Delta t}{\Delta x}\left[(q^{**})^n_{i+1/2}\left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1}\right)^n_{i+1/2} + \left(\tilde{\beta}_f\right)^n_{i-1/2}\left(\frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_1^+}{\tilde{\lambda}_1}\right)^n_{i-1/2}\right].$$

While the friction source term is linearized in the ROM by using $\hat{h}$, the reduced version of the intermediate state $q^{**}$ depends on both $\hat{h}$ and $\hat{q}$. This makes it necessary for its contribution to be split into different coefficients of the ROM (5.2); the contribution of $\left(\tilde{\beta}_f\right)^n_{i+1/2}$ has to be removed from the corresponding ROM coefficient as follows

$$\begin{aligned}
B^h_{pk} &= B^h_{pk} + \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2}\right)^w_{i+1/2}\left(\hat{\beta}^{\mathrm{RF}}_h\right)^w_{i+1/2}\phi_{i,p} - \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2}\right)^w_{i+1/2}\left(\hat{\beta}_f\right)^w_{i+1/2}\phi_{i,p},\\
C^h_{pk} &= C^h_{pk} + \left(\frac{\tilde{\lambda}_1^-}{\tilde{\lambda}_1} - \frac{\tilde{\lambda}_2^-}{\tilde{\lambda}_2}\right)^w_{i+1/2}\left(\hat{\beta}^{\mathrm{RF}}_q\right)^w_{i+1/2}\phi_{i,p},\\
B^q_{pk} &= B^q_{pk} + \left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)^w_{i+1/2}\left(\hat{\beta}^{\mathrm{RF}}_h\right)^w_{i+1/2}\phi_{i,p} - \left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)^w_{i+1/2}\left(\hat{\beta}_f\right)^w_{i+1/2}\phi_{i,p},\\
C^q_{pk} &= C^q_{pk} + \left(\tilde{\lambda}_1^- - \tilde{\lambda}_2^-\right)^w_{i+1/2}\left(\hat{\beta}^{\mathrm{RF}}_q\right)^w_{i+1/2}\phi_{i,p},
\end{aligned}$$

with

$$\begin{aligned}
\left(\hat{\beta}^{\mathrm{RF}}_{\hat{h}}\right)^w_{i+1/2} &= \left(\frac{\tilde{\lambda}_1\tilde{\lambda}_2}{\tilde{c}}\right)^w_{i+1/2}\delta\phi_{i+1/2,k} - \frac{g\delta z_{i+1/2}}{2\tilde{c}^w_{i+1/2}}\tilde{\phi}_{i+1/2,k},\\
\left(\hat{\beta}^{\mathrm{RF}}_{\hat{q}}\right)^w_{i+1/2} &= \varphi_{i,k} - \delta\varphi_{i+1/2,k}\frac{1}{2}\left(\frac{\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2},\\
\left(\hat{\beta}_f\right)^w_{i+1/2} &= \Delta x\frac{gn^{2_b}}{2}\left(\frac{\tilde{u}|\tilde{u}|}{\tilde{c}}\right)^w_{i+1/2}\frac{\tilde{\phi_{i+1/2,k}}}{\max\left(h^w_i, h^w_{i+1}\right)^{4/3}}.
\end{aligned}$$

The extension of this correction to the 2D version of the ROM is straightforward to implement.

**Test case 27. 1D dam break problem**

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

$$h(x,y,0) = \begin{cases} 0.05, & \text{if } x+y \leq 20, \\ 0, & \text{if } 20 < x+y, \end{cases} \quad q(x,0) = 0. \tag{5.14}$$

There is no bed slope (i.e., $z \equiv 0$) and the Manning coefficient is $n_b = 0.03$. The time-space domain $(x,y,t) \in [0,20] \times [0,20] \times [0,20]$ is divided into $I_c = 8$ cells, as shown in Figure 5.7, where the initial water depth is represented in greyscale according to (5.14).

Figure 5.7: TC27. 2D mesh with initial water depth.

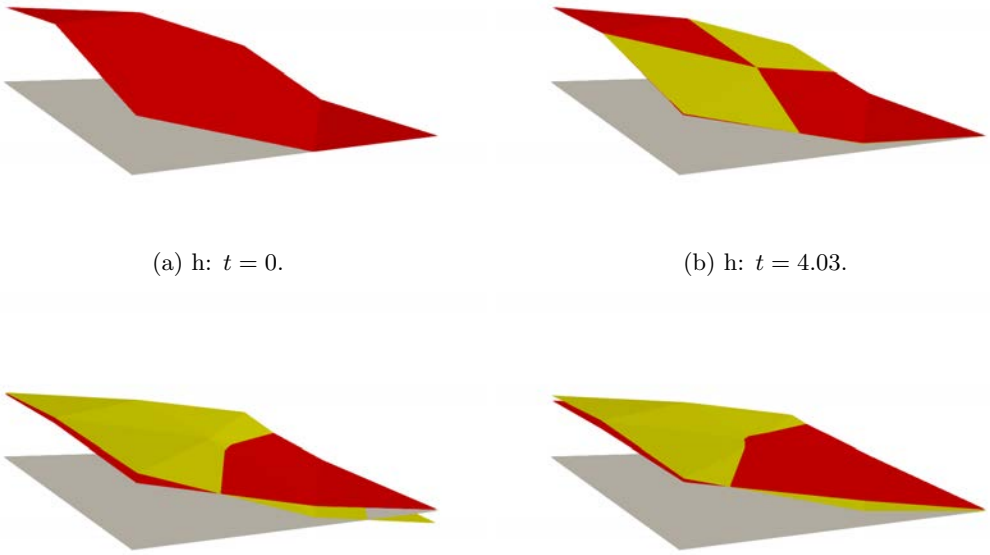
The ROM makes use of the maximum number of POD modes and is trained with just one snapshot per time window, as indicated in Table 5.12.

| $L_x \times L_y$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $20 \times 20$ | 20 | Eq. (5.14) | Wall | 0 | 0.03 | 0.4 | 8 | 11 | 8 | 11 |

Table 5.12: TC26: problem settings.

The subcases presented in this test case are used to illustrate the need to include the correction of the friction term in the ROM training phase. While Subcase 1 solves the ROM without including such a correction, Subcase 2 does.

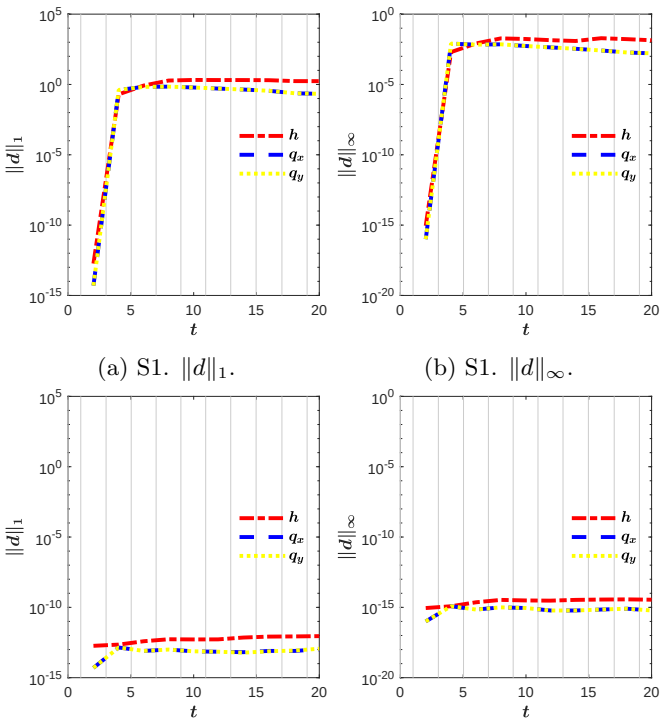
Errors introduced by not including the friction correction in the ROM can lead to negative water depths, as can be seen in the ROM solutions of Subcase 1, represented by the yellow color in the Figure 5.8. The ROM solution computed by Subcase 2, on the contrary, matches with machine precision the FOM solution, as indicated by the differences shown in Figure 5.9. This is the reason why in Figure 5.8 the ROM solution of Subcase 2 (red) and the FOM solution (blue) are overlapping and the latter is not visible.



(a) h: $t = 0$. (b) h: $t = 4.03$.

(c) h: $t = 13.95$. (d) h: $t = 20$.

Figure 5.8: TC27. Time evolution of the water depth $h$ computed by the FOM in blue, the ROM for Subcase 1 (without friction correction) in yellow and for Subcase 2 (with friction correction) in red.




(a) S1. $\|d\|_1$. (b) S1. $\|d\|_\infty$.

(c) S2. $\|d\|_1$. (d) S2. $\|d\|_\infty$.

Figure 5.9: TC27. differents between the FOM and the ROM solutions using $\|d\|_1$ and $\|d\|_\infty$.

### 5.2.4 Entropy fix

The entropy fix, as explained in Section 2.4.4, is necessary to avoid non-physical solutions, when solving the ARoe-based FOM (2.25). Unfortunately, the ARoe-based ROM (5.2) does not inherit such correction by simply being trained with corrected snapshots. It is necessary to apply the entropy correction also to its numerical resolution. This leads to a conflict, as the entropy fix is evaluated on the physical space and, on the contrary, the ROM is solved in the reduced space. One possible solution is to apply it in the training phase. That is, in each time window, it is evaluated whether or not it is necessary to correct the entropy on the time-averaged variables. And once this condition has been evaluated according to what is indicated in Section 2.4.4.4, the matrices are constructed within that time window with the modified eigenvalues and source terms, following (2.35), (2.36) and (2.37).

The following example shows how a solution requiring this correction can be recovered with machine accuracy by evaluating the variables in the training phase, thus allowing the ROM to compute proper solutions.

#### Test case 28. 1D dam break problem

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

$$h(x,0) = \begin{cases} 2, & \text{if } x \leq 3, \\ 0.2, & \text{if } 3 < x, \end{cases} \quad q(x,0) = 0. \tag{5.15}$$

There is no bed slope and no friction (i.e., $z \equiv 0$ and $n_b \equiv 0$). The time-space domain $(x,t) \in [0,6] \times [0,0.8]$ is divided in two different mesh refinements, as indicated in Table 5.13.

To better illustrate this study, a very coarse mesh has been provided, the ROM makes use of the maximum number of POD modes and is trained with just one snapshot per time window. This allows the solution to be recovered with machine accuracy where it is well resolved. In Subcases 2 and 4 the entropy fix is applied to the training phase of the ROM, but it is not in Subcases 1 and 3.

| Subcase | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | Entropy fix |
|---|---|---|---|---|---|
| 1 | 20 | 19 | 20 | 19 | No |
| 2 | 20 | 19 | 20 | 19 | Yes |
| 3 | 200 | 181 | 5 | 36 | No |
| 4 | 200 | 181 | 5 | 36 | Yes |
| $L = 6$, $T = 0.8$, IC: Eq. (5.15) | | | | | |
| BCs: free, $z = 0$, $n_b = 0$, CFL $= 0.9$ | | | | | |

Table 5.13: TC28. Problem settings.

Figures 5.11 and 5.12 show the solutions of the ROM for $h$ and $q$ computed in the four subcases at differents time steps. For all subcases, the differences between the solutions of the FOM

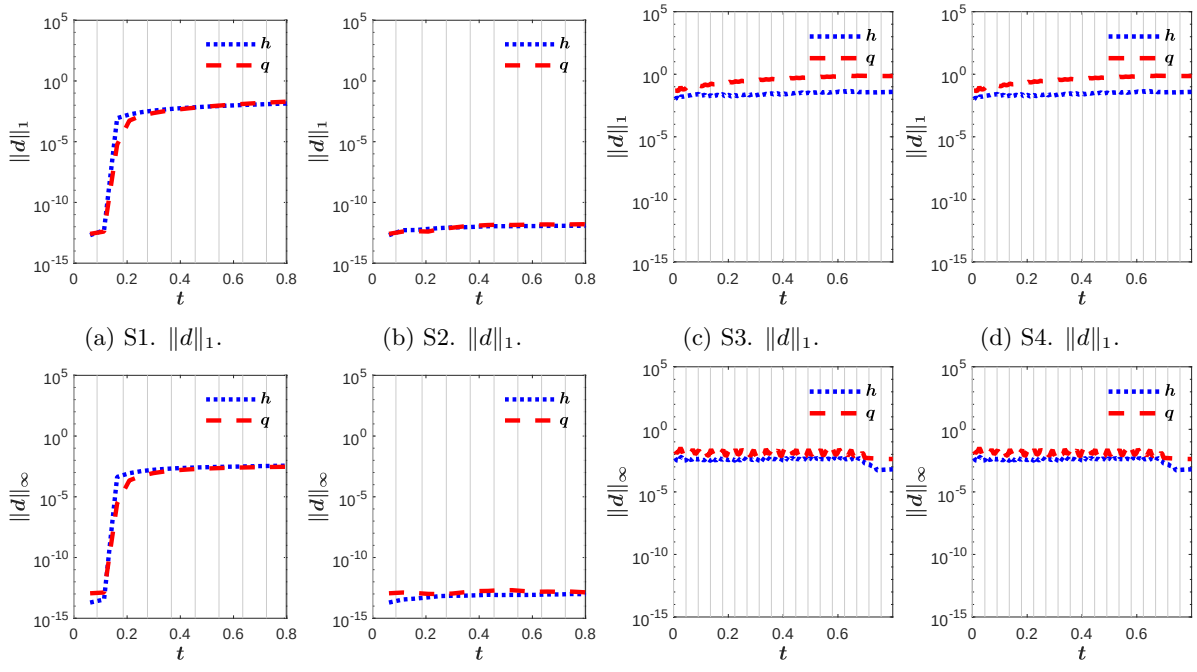
and the ROM for $h$ are not visible. On the contrary, regarding the water discharge $q$, the differences are visible for Subcase 3 and 4. The Froude number at the same time instants is shown in Figure 5.13, and it can be seen how the regime changes from sub- to supercritical.

Figure 5.10 shows the time evolution of the differences of the ROM solutions with respect to the FOM solutions for all subcases. Thanks to these, it is easy to see that it is indeed necessary to include the entropy fix in the ROM in order to recover the solution with machine precision, as shown in Figures 5.10b and 5.10f for Subcase 2. If not corrected (Subcase 1), a large error is introduced from the first instant, as shown in Figures 5.10a and 5.10e.

When the number of POD modes used to solve the ROM is reduced, the error introduced by not correcting for entropy is smaller in magnitude and is shielded by the error of using a non-maximum number of POD modes. This is the reason why in Figures 5.10c-5.10g and 5.10d-5.10h similar errors are obtained for Subcases 3 and 4.

In other words, it seems that it is not absolutely necessary to include the entropy fix when using the ROM. However, since the entropy fix is carried out in the ROM training phase and, thus, does not involve extra computational cost, it is better to take it into account.

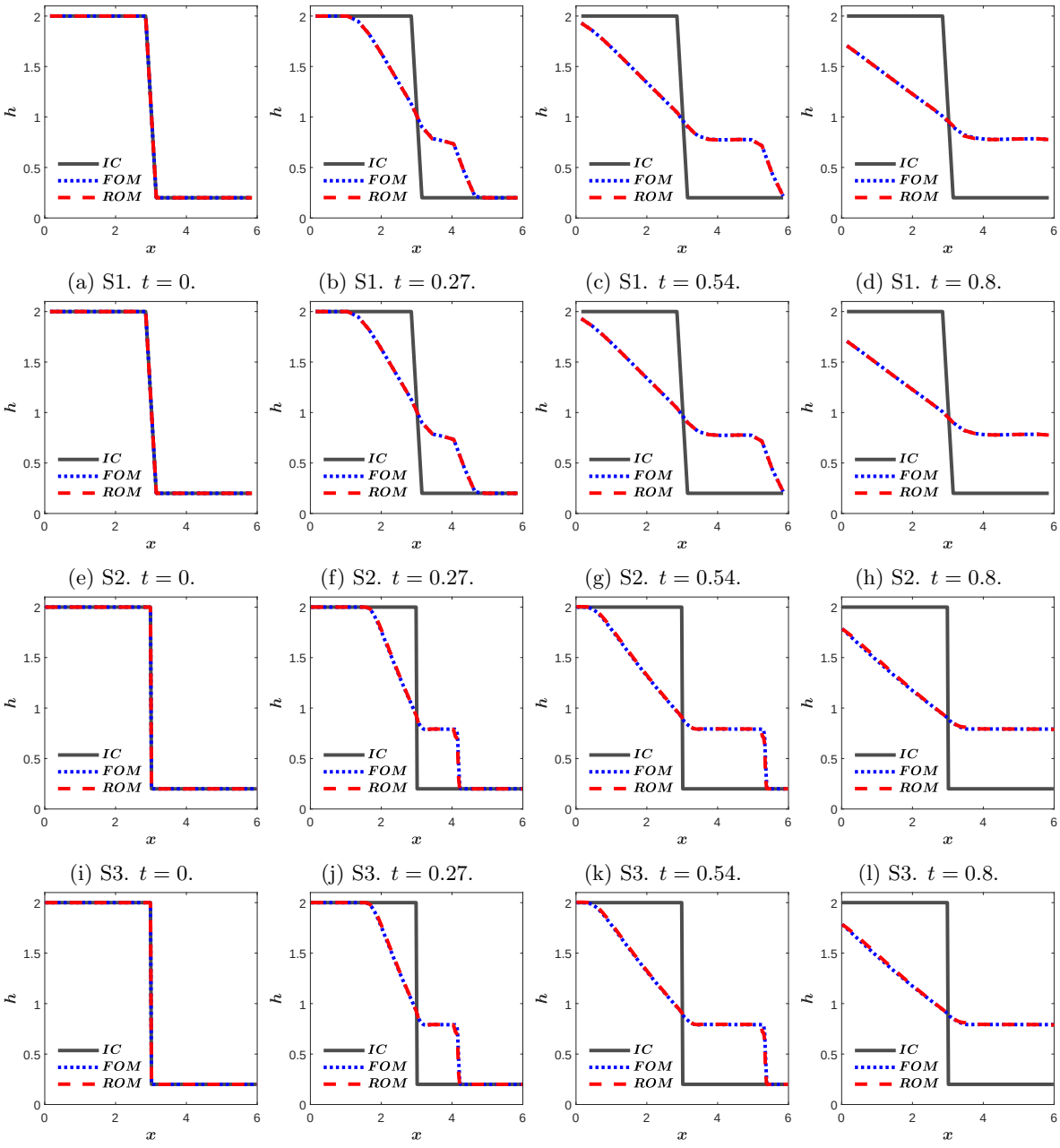
Figures 5.13 have also been included, plotting the time evolution of the Froude number and showing the change from subcritical to supercritical regime, as explained in Section 2.4.4.4.




(a) S1. $\|d\|_1$. (b) S2. $\|d\|_1$. (c) S3. $\|d\|_1$. (d) S4. $\|d\|_1$.

(e) S1. $\|d\|_\infty$. (f) S2. $\|d\|_\infty$. (g) S3. $\|d\|_\infty$. (h) S4. $\|d\|_\infty$.

Figure 5.10: TC28. differents between the FOM and the ROM solutions using $\|d\|_1$ and $\|d\|_\infty$.



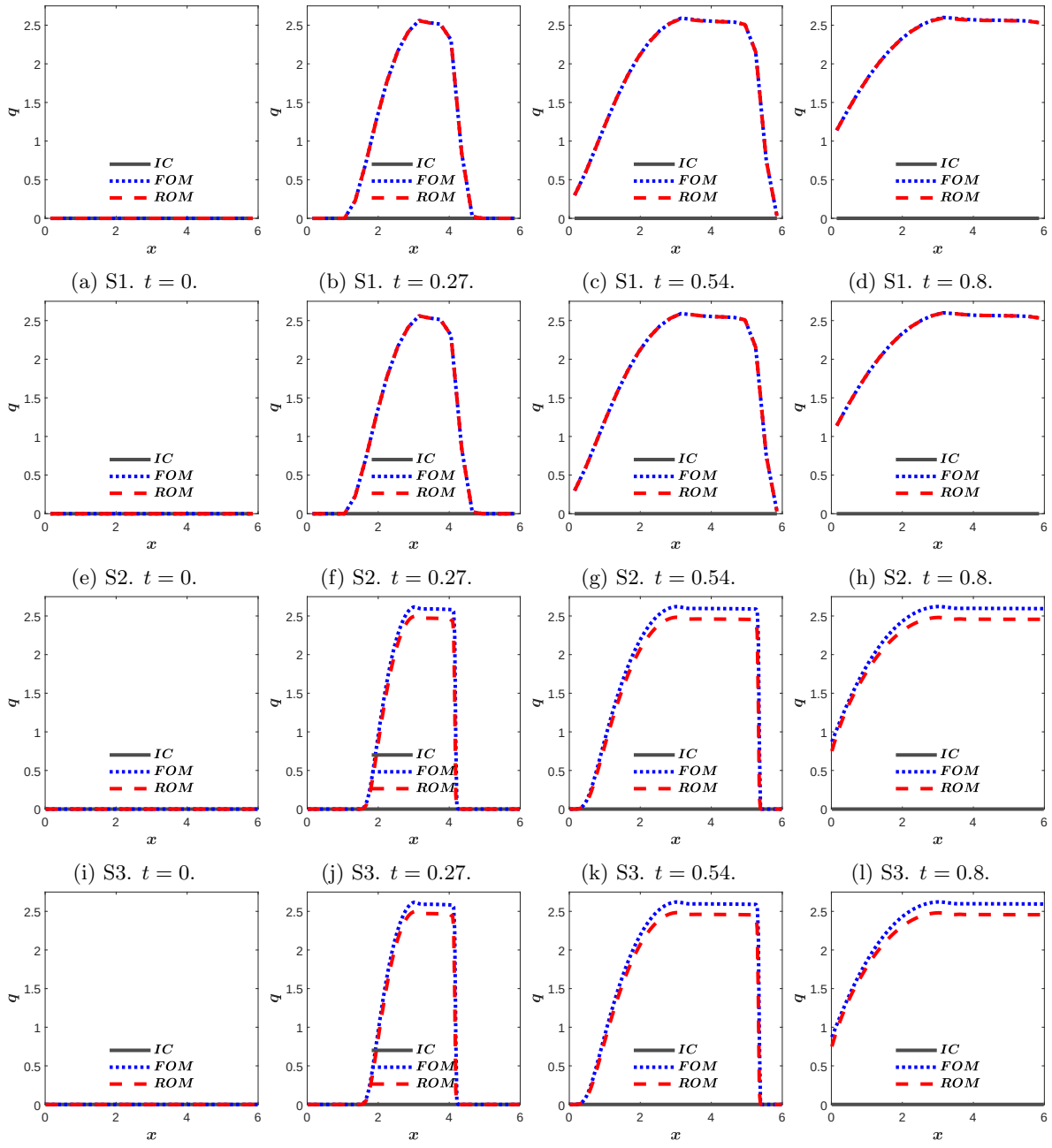

(a) S1. $t = 0$. (b) S1. $t = 0.27$. (c) S1. $t = 0.54$. (d) S1. $t = 0.8$.

(e) S2. $t = 0$. (f) S2. $t = 0.27$. (g) S2. $t = 0.54$. (h) S2. $t = 0.8$.

(i) S3. $t = 0$. (j) S3. $t = 0.27$. (k) S3. $t = 0.54$. (l) S3. $t = 0.8$.

(m) S4. $t = 0$. (n) S4. $t = 0.27$. (o) S4. $t = 0.54$. (p) S4. $t = 0.8$.

Figure 5.11: TC28. ROM solutions for $h$ at different time. Each subcase is in a different row.



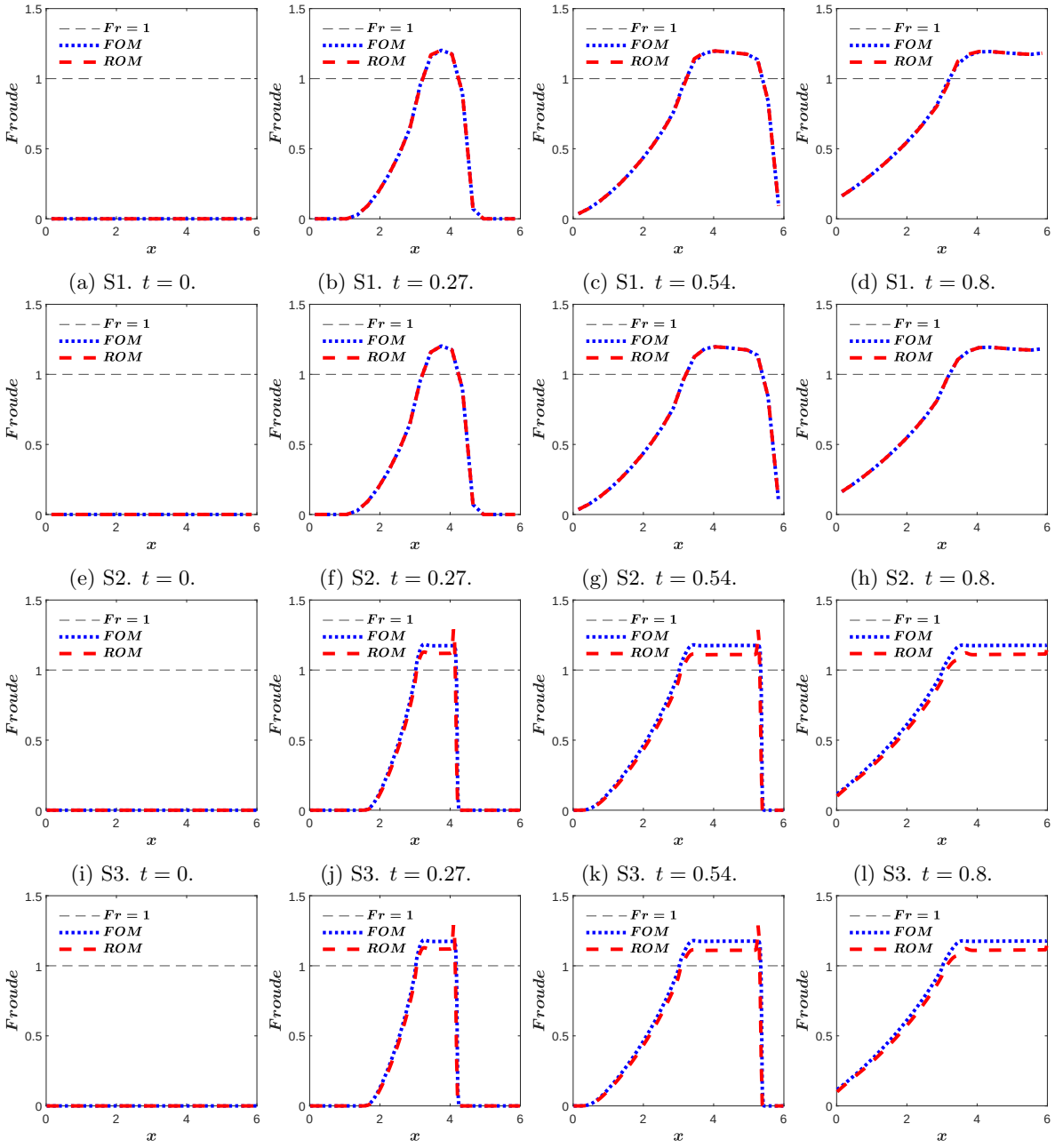

(a) S1. $t = 0$. (b) S1. $t = 0.27$. (c) S1. $t = 0.54$. (d) S1. $t = 0.8$.

(e) S2. $t = 0$. (f) S2. $t = 0.27$. (g) S2. $t = 0.54$. (h) S2. $t = 0.8$.

(i) S3. $t = 0$. (j) S3. $t = 0.27$. (k) S3. $t = 0.54$. (l) S3. $t = 0.8$.

(m) S4. $t = 0$. (n) S4. $t = 0.27$. (o) S4. $t = 0.54$. (p) S4. $t = 0.8$.

Figure 5.12: TC28. ROM solutions for $q$ at different time. Each subcase is in a different row.




(a) S1. $t = 0$. (b) S1. $t = 0.27$. (c) S1. $t = 0.54$. (d) S1. $t = 0.8$.

(e) S2. $t = 0$. (f) S2. $t = 0.27$. (g) S2. $t = 0.54$. (h) S2. $t = 0.8$.

(i) S3. $t = 0$. (j) S3. $t = 0.27$. (k) S3. $t = 0.54$. (l) S3. $t = 0.8$.

(m) S4. $t = 0$. (n) S4. $t = 0.27$. (o) S4. $t = 0.54$. (p) S4. $t = 0.8$.

Figure 5.13: TC28. ROM solutions for the Froude number (2.34) at different time. Each subcase is in a different row.

### 5.2.5 Wet/dry treatment

The wet/dry treatment, as explained in Section 2.4.4, is applied to the ARoe-based FOM (2.25) to avoid unphysical solutions in which volume cells that should stay dry get wet. Similar to the entropy correction, wet/dry is corrected by evaluating space-dependent variables.

Of the two steps that constitute the wet/dry processing, the first one (2.32) has to be applied

in the off-line phase, as it is an assessment of space-dependent variables. To illustrate how this treatment works, a 1D example will be used (the extension to 2D is straightforward).

Let consider the first term of coefficient $B^h_{pk}$ of the ARoe-based ROM (5.2)

$$B^h_{pk} = -\frac{1}{2}\sum_{i\in J^I}\left[\delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2} + \delta\phi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i-1/2}\right]\phi_{i,p}$$

Assume that at the $(i+1/2)$-th wall the wet/dry condition is met, i.e. $h_i + z_i < z_{i+1}$, as shown in Figure 2.5. In such a case, inside the summation to all $J^I$ interior volume cells, originally the following would be obtained

$$\begin{aligned} B^h_{pk} = -\ldots &- \frac{1}{2}\delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2}\phi_{i,p} + \delta\phi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i-1/2}\phi_{i,p} \\ &- \frac{1}{2}\delta\phi_{i+3/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+3/2}\phi_{i+1,p} + \delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2}\phi_{i+1,p} - \ldots \end{aligned}$$

However, when applying the wet/dry treatment, the bounce of information to the wet cell is solved as follows

$$\begin{aligned} B^h_{pk} = -\ldots &- \frac{1}{2}\delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2}\phi_{i,p} + \delta\phi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i-1/2}\phi_{i,p} \\ &- \frac{1}{2}\delta\phi_{i+3/2,k}\left(\frac{\tilde{\lambda}_1^-\tilde{\lambda}_2 - \tilde{\lambda}_2^-\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+3/2}\phi_{i+1,p} + \delta\phi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^+\tilde{\lambda}_2 - \tilde{\lambda}_2^+\tilde{\lambda}_1}{\tilde{c}}\right)^w_{i+1/2}\phi_{i,p} - \ldots \end{aligned}$$

where the last term is multiplied by $\phi_{i,p}$ instead of by $\phi_{i+1,p}$, as indicated in red.

This is evaluated during the off-line phase and is extended to the rest of the coefficients of the ARoe-based FOM to complete the first wet/dry step.

The second step, as already mentioned in Chapter 2, is treated very differently in the 1D and 2D cases, following (2.33) and (2.43), respectively. In a similar way, it is the same for the ROM.

In 1D problems, the water velocity is cancelled (and water discharge) in the two volume cells adjacent to the wall on which the wet/dry treatment is applied. This has a very simple translation to the ROM domain: in the construction of the ARoe-based ROM coefficients (5.4) during the off-line phase, the water discharge contribution in the cells involved is cancelled out. An example is given below to illustrate this, making use of the ARoe-based ROM $C^h_{pk}$ coefficient

$$C^h_{pk} = \frac{1}{2}\sum_{i\in J^I}\left[\delta\varphi_{i+1/2,k}\left(\frac{\tilde{\lambda}_1^- - \tilde{\lambda}_2^-}{\tilde{c}}\right)^w_{i+1/2} + \delta\varphi_{i-1/2,k}\left(\frac{\tilde{\lambda}_1^+ - \tilde{\lambda}_2^+}{\tilde{c}}\right)^n_{i-1/2}\right]\phi_{i,p}.$$

Again, assume that at the $(i+1/2)$-th wall the wet/dry condition is met. In that case, the summation would not extend to all the cells inside the domain, but to a subgroup of the domain that does not include the cells on both sides of the $(i+1/2)$-th wall, i.e. $J^I \setminus \{i, i+1\}$

$$C^h_{pk} = \frac{1}{2} \sum_{i \in J^I \setminus \{i,i+1\}} \left[ \delta\varphi_{i+1/2,k} \left( \frac{\tilde{\lambda}_1^- - \tilde{\lambda}_2^-}{\tilde{c}} \right)^w_{i+1/2} + \delta\varphi_{i-1/2,k} \left( \frac{\tilde{\lambda}_1^+ - \tilde{\lambda}_2^+}{\tilde{c}} \right)^n_{i-1/2} \right] \phi_{i,p},$$

where the modification is indicated in red. This extends to all elements of the ARoe-based ROM (5.2) coefficients.

But in 2D cases it is not so simple. As explained in Section 2.5, once the update in time of the numerical scheme has been completed, it is necessary to correct both components of the water flow in the $x$ and $y$ directions to cancel the component perpendicular to the wall involved in the wet/dry. Since this is an update at each time instant, it is necessary to obtain a reduced version of it

$$\begin{aligned} (\hat{q}_x)_p^{n+1} &= \sum_{k=1}^{M_{\text{POD}}} \hat{a}_{pk}^{WD} \, (\hat{q}_x)_p^{n+1} + \sum_{k=1}^{M_{\text{POD}}} \hat{b}_{pk}^{WD} \, (\hat{q}_y)_p^{n+1}, \\ (\hat{q}_y)_p^{n+1} &= \sum_{k=1}^{M_{\text{POD}}} \hat{c}_{pk}^{WD} \, (\hat{q}_x)_p^{n+1} + \sum_{k=1}^{M_{\text{POD}}} \hat{d}_{pk}^{WD} \, (\hat{q}_y)_p^{n+1}. \end{aligned} \tag{5.16}$$

The reduced coefficients only depend on space and are constructed in different ways depending on whether the numerical mesh consists of triangular or rectangular cells. In the triangular case, they are defined as follows

$$\begin{aligned} \hat{a}_{pk}^{WD} &= \sum_{i \in J^I \cup J^S}^{I_c} \left[ a_{i,3}^{WD} \left( a_{i,2}^{WD} a_{i,1}^{WD} + b_{i,2}^{WD} c_{i,1}^{WD} \right) + b_{i,3}^{WD} \left( c_{i,2}^{WD} a_{i,1}^{WD} + d_{i,2}^{WD} c_{i,1}^{WD} \right) \right] \varphi_{i,k} \varphi_{i,p}, \\ \hat{b}_{pk}^{WD} &= \sum_{i \in J^I \cup J^S}^{I_c} \left[ a_{i,3}^{WD} \left( a_{i,2}^{WD} b_{i,1}^{WD} + b_{i,2}^{WD} d_{i,1}^{WD} \right) + b_{i,3}^{WD} \left( c_{i,2}^{WD} b_{i,1}^{WD} + d_{i,2}^{WD} d_{i,1}^{WD} \right) \right] \psi_{i,k} \varphi_{i,p}, \\ \hat{c}_{pk}^{WD} &= \sum_{i \in J^I \cup J^S}^{I_c} \left[ c_{i,3}^{WD} \left( a_{i,2}^{WD} a_{i,1}^{WD} + b_{i,2}^{WD} c_{i,1}^{WD} \right) + d_{i,3}^{WD} \left( c_{i,2}^{WD} a_{i,1}^{WD} + d_{i,2}^{WD} c_{i,1}^{WD} \right) \right] \varphi_{i,k} \psi_{i,p}, \\ \hat{d}_{pk}^{WD} &= \sum_{i \in J^I \cup J^S}^{I_c} \left[ c_{i,3}^{WD} \left( a_{i,2}^{WD} b_{i,1}^{WD} + b_{i,2}^{WD} d_{i,1}^{WD} \right) + d_{i,3}^{WD} \left( c_{i,2}^{WD} b_{i,1}^{WD} + d_{i,2}^{WD} d_{i,1}^{WD} \right) \right] \psi_{i,k} \psi_{i,p}, \end{aligned} \tag{5.17}$$

And in the case of rectangular cells, the reduced coefficients are defined as follows

$$\begin{aligned} \hat{a}_{pk}^{WD} = \sum_{i \in J^I \cup J^S}^{I_c} \Bigg\{ & a_{i,4}^{WD} \left[ a_{i,3}^{WD} \left( a_{i,2}^{WD} a_{i,1}^{WD} + b_{i,2}^{WD} c_{i,1}^{WD} \right) + b_{i,3}^{WD} \left( c_{i,2}^{WD} a_{i,1}^{WD} + d_{i,2}^{WD} c_{i,1}^{WD} \right) \right] \\ & + b_{i,4}^{WD} \left[ a_{i,3}^{WD} \left( a_{i,2}^{WD} a_{i,1}^{WD} + b_{i,2}^{WD} c_{i,1}^{WD} \right) + b_{i,3}^{WD} \left( c_{i,2}^{WD} a_{i,1}^{WD} + d_{i,2}^{WD} c_{i,1}^{WD} \right) \right] \Bigg\} \varphi_{i,k} \varphi_{i,p}, \end{aligned} \tag{5.18}$$

$$\hat{b}_{pk}^{WD} = \sum_{i\in J^I\cup J^S}^{I_c} \Bigg\{ a_{i,4}^{WD}\Bigg[a_{i,3}^{WD}\left(a_{i,2}^{WD}b_{i,1}^{WD}+b_{i,2}^{WD}d_{i,1}^{WD}\right)+b_{i,3}^{WD}\left(c_{i,2}^{WD}b_{i,1}^{WD}+d_{i,2}^{WD}d_{i,1}^{WD}\right)\Bigg] + b_{i,4}^{WD}\Bigg[a_{i,3}^{WD}\left(a_{i,2}^{WD}b_{i,1}^{WD}+b_{i,2}^{WD}d_{i,1}^{WD}\right)+b_{i,3}^{WD}\left(c_{i,2}^{WD}b_{i,1}^{WD}+d_{i,2}^{WD}d_{i,1}^{WD}\right)\Bigg]\Bigg\}\psi_{i,k}\varphi_{i,p},$$

$$\hat{c}_{pk}^{WD} = \sum_{i\in J^I\cup J^S}^{I_c} \Bigg\{ c_{i,4}^{WD}\Bigg[c_{i,3}^{WD}\left(a_{i,2}^{WD}a_{i,1}^{WD}+b_{i,2}^{WD}c_{i,1}^{WD}\right)+d_{i,3}^{WD}\left(c_{i,2}^{WD}a_{i,1}^{WD}+d_{i,2}^{WD}c_{i,1}^{WD}\right)\Bigg] + d_{i,4}^{WD}\Bigg[c_{i,3}^{WD}\left(a_{i,2}^{WD}a_{i,1}^{WD}+b_{i,2}^{WD}c_{i,1}^{WD}\right)+d_{i,3}^{WD}\left(c_{i,2}^{WD}a_{i,1}^{WD}+d_{i,2}^{WD}c_{i,1}^{WD}\right)\Bigg]\Bigg\}\varphi_{i,k}\psi_{i,p},$$

$$\hat{d}_{pk}^{WD} = \sum_{i\in J^I\cup J^S}^{I_c} \Bigg\{ c_{i,4}^{WD}\Bigg[c_{i,3}^{WD}\left(a_{i,2}^{WD}b_{i,1}^{WD}+b_{i,2}^{WD}d_{i,1}^{WD}\right)+d_{i,3}^{WD}\left(c_{i,2}^{WD}b_{i,1}^{WD}+d_{i,2}^{WD}d_{i,1}^{WD}\right)\Bigg] + d_{i,4}^{WD}\Bigg[c_{i,3}^{WD}\left(a_{i,2}^{WD}b_{i,1}^{WD}+b_{i,2}^{WD}d_{i,1}^{WD}\right)+d_{i,3}^{WD}\left(c_{i,2}^{WD}b_{i,1}^{WD}+d_{i,2}^{WD}d_{i,1}^{WD}\right)\Bigg]\Bigg\}\psi_{i,k}\psi_{i,p},$$

where the coefficients $a_{i,m}^{WD}$, $b_{i,m}^{WD}$, $c_{i,m}^{WD}$ and $d_{i,m}^{WD}$ can be found in (2.44).

**Test case 29. 1D overtopping of a wall.**

A 1D case is proposed to study how the inclusion of the wet/dry treatment affects the solution of the ARoe-based ROM. The time-space domain is defined as $(x,t)\in[0,6]\times[0,10]$; and the spatial domain is divided in two. In this way, the initial condition is defined as

$$h(x,0)=\begin{cases}1, & \text{if } x\leq 3,\\ 0, & \text{if } 3<x,\end{cases} \tag{5.19}$$

with the following bed level

$$z(x)=\begin{cases}0, & \text{if } x\leq 3,\\ 2, & \text{if } 3<x.\end{cases} \tag{5.20}$$

Free boundary conditions are imposed on the right boundary and, from the left boundary, a constant water discharge enters the domain

$$q(0,t)=4,\ t>0. \tag{5.21}$$

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

The spatial domain has been discretized using two different mesh refinements, as indicated in Table 5.14. In Subcases 2 and 4 the wet/dry treatment is applied to the ROM, but it is not in Subcases 1 and 3.

| Subcase | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | Wet/dry treatment |
|---|---|---|---|---|---|
| 1 | 20 | 255 | 20 | 255 | No |
| 2 | 20 | 255 | 20 | 255 | Yes |
| 3 | 200 | 2566 | 5 | 256 | No |
| 4 | 200 | 2566 | 5 | 256 | Yes |
| $L = 6$, $T = 10$, IC: Eq. (5.19) | | | | | |
| BCs: Eq. (5.21), $z$: Eq. (5.20), $n_b = 0$, CFL $= 0.9$ | | | | | |

Table 5.14: TC29. Problem settings.

The four subcases shown in the Table 5.14 have been proposed to test whether or not it is also necessary to consider this correction when solving the ROM, taking into account that the training snapshots are wet/dry treated.

The first two subcases help to better illustrate this study. For this purpose, a very coarse mesh has been provided, the ROM makes use of the maximum number of POD modes and is trained with just one snapshot per time window. This will allow the solution to be recovered with machine accuracy where it is well resolved. In Subcase 1, wet/dry treatment has not been included, while in Subcase 2, it has been included.

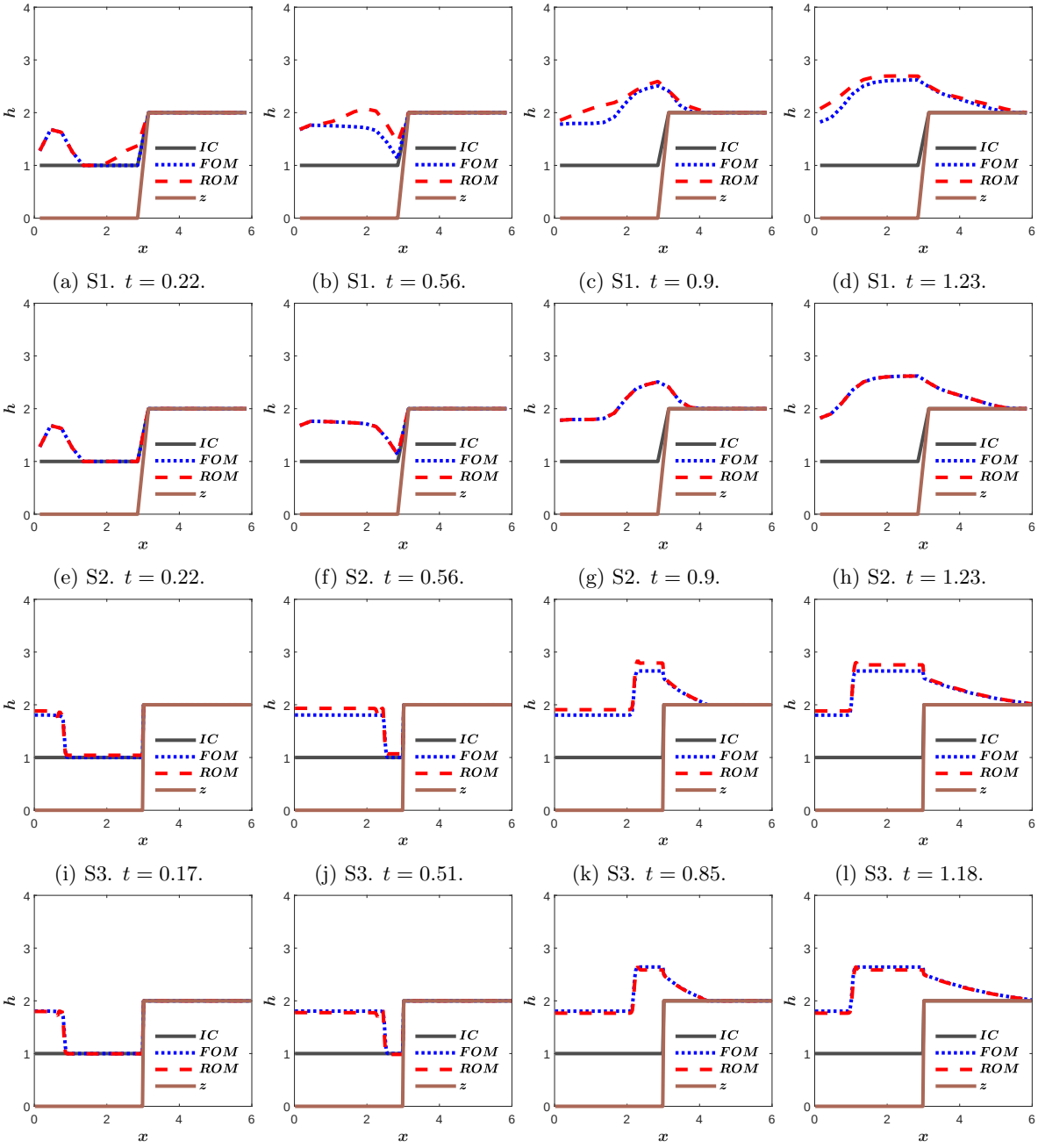
Figures 5.14 and 5.15 show the time evolution of the solutions computed by the FOM and the ROM for $h$ and $q$, respectively. It is clear from these figures that the wet/dry treatment allows the ROM to compute solutions with physical significance, as the water depth remains constant in areas where it should not rise. This detail can be seen in Figures 5.14a and 5.14e. In the first one, the ROM wets cells next to the wall that should not be wet. Whereas in the second one, where the ROM is treated, its solution perfectly matches the FOM solution. The consequence of this is that the error of Subcase 1 increases from the first instant, as shown in Figures 5.17a and 5.17e. On the contrary, Subcase 2 is able to compute exactly the same solution as the FOM with machine accuracy, as shown in Figures 5.17b and 5.17f. In other words, the wet/dry treatment is necessary to reproduce physically satisfactory solutions.

However, the error introduced by the absence this correction is screened by the error of the POD method, just like in the case of the entropy fix. This can be seen in the results of Subcases 3 and 4, where the same problem is solved but with more cells, without and with the treatment, respectively. The differences shown for both subcases lay in the same order of magnitude, as shown in Figures 5.17c-5.17g and 5.17d and 5.17h.

In addition to all this, there is a regime change from sub- to supercritical from $t = 0.9$, as can be seen in Figure 5.16, so the entropy fix is also involved in this test case.

It is worth assessing in 2D problems whether this treatment entails a significant loss of computational cost, as the second step of the treatment is performed in the on-line phase of the

ROM.



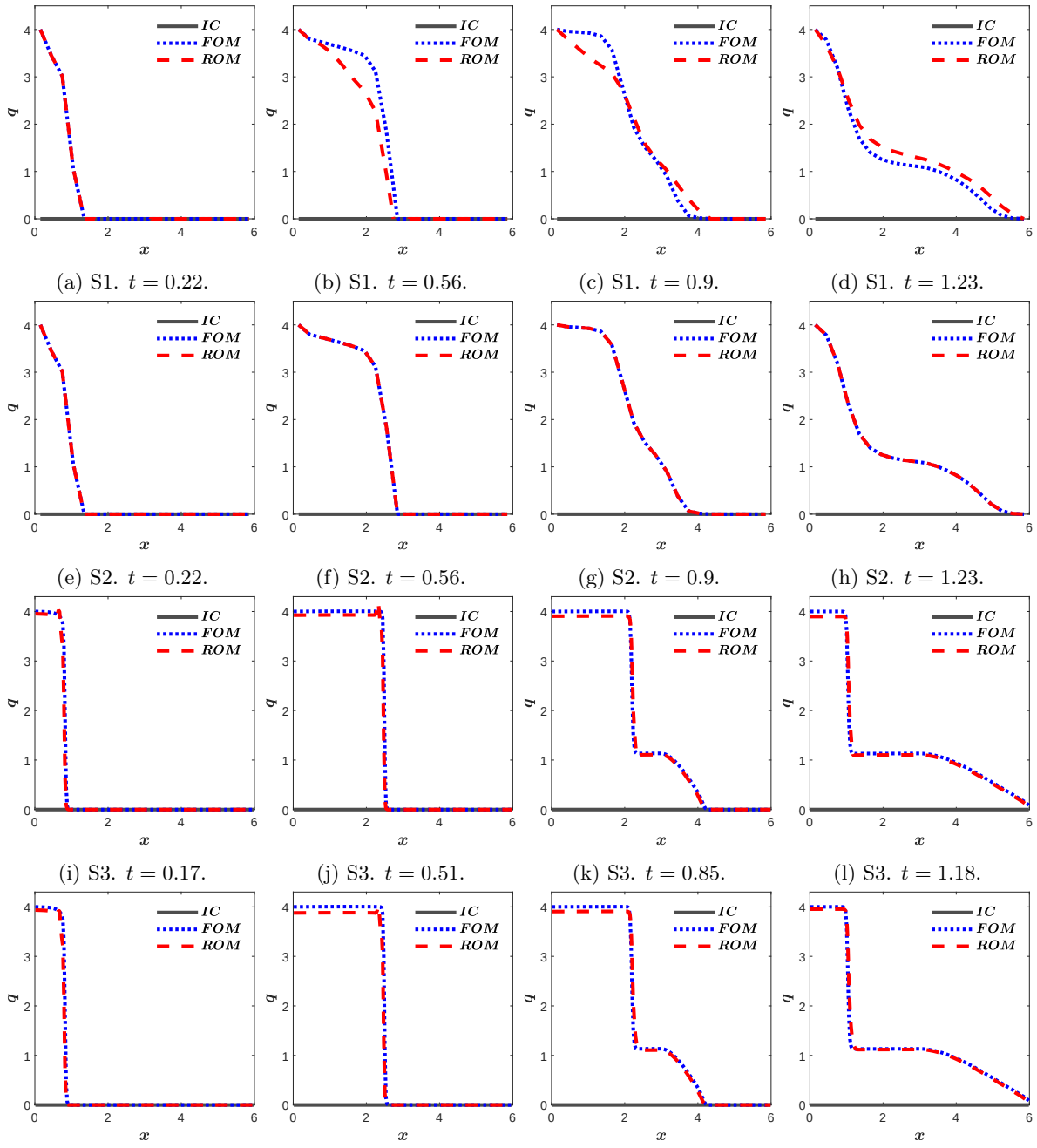

(a) S1. $t = 0.22$. (b) S1. $t = 0.56$. (c) S1. $t = 0.9$. (d) S1. $t = 1.23$.

(e) S2. $t = 0.22$. (f) S2. $t = 0.56$. (g) S2. $t = 0.9$. (h) S2. $t = 1.23$.

(i) S3. $t = 0.17$. (j) S3. $t = 0.51$. (k) S3. $t = 0.85$. (l) S3. $t = 1.18$.

(m) S4. $t = 0.17$. (n) S4. $t = 0.51$. (o) S4. $t = 0.85$. (p) S4. $t = 1.18$.

Figure 5.14: TC29. ROM solutions for $h$ at different time. Each subcase is in a different row.



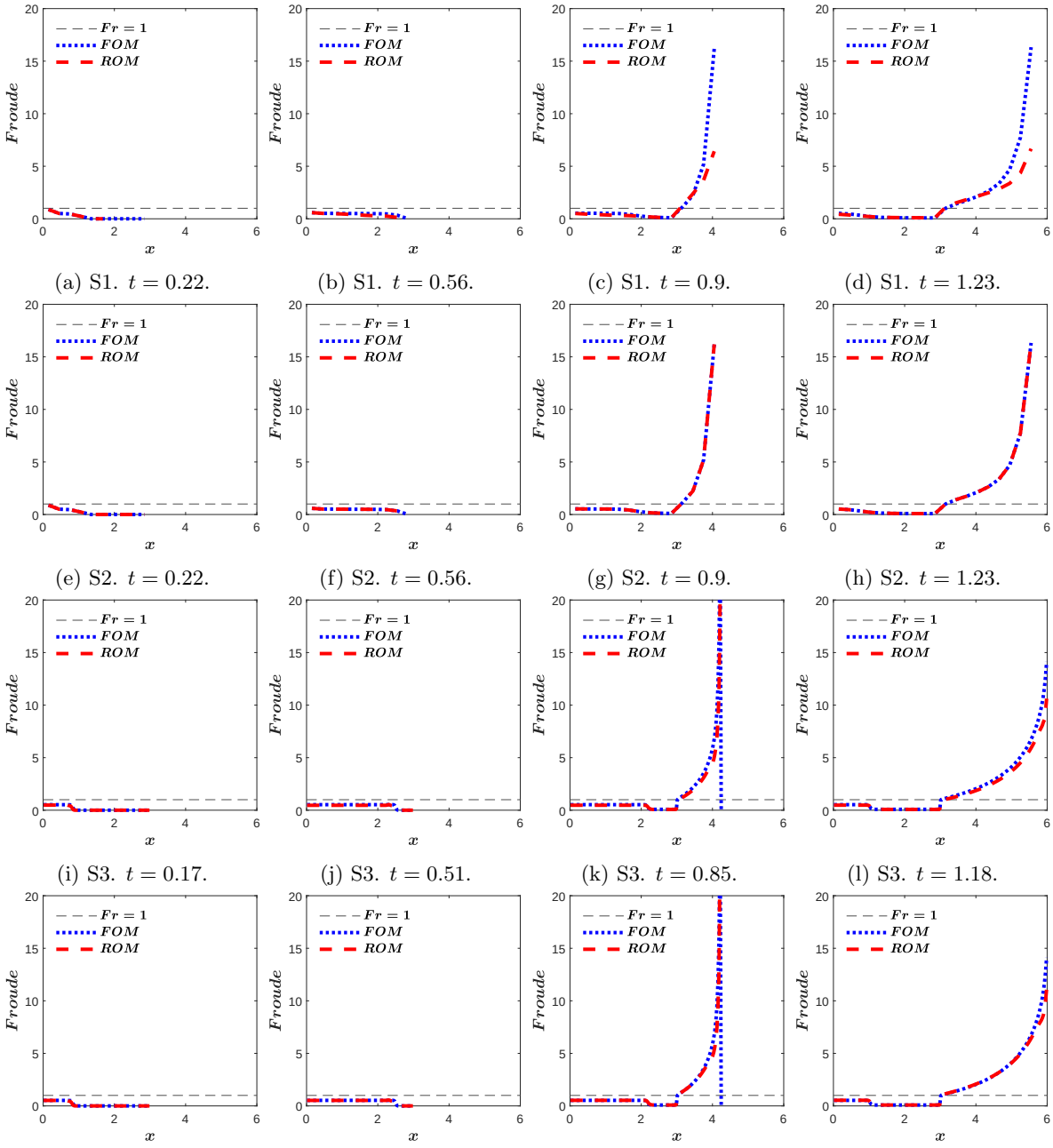

(a) S1. $t = 0.22$. (b) S1. $t = 0.56$. (c) S1. $t = 0.9$. (d) S1. $t = 1.23$.

(e) S2. $t = 0.22$. (f) S2. $t = 0.56$. (g) S2. $t = 0.9$. (h) S2. $t = 1.23$.

(i) S3. $t = 0.17$. (j) S3. $t = 0.51$. (k) S3. $t = 0.85$. (l) S3. $t = 1.18$.

(m) S4. $t = 0.17$. (n) S4. $t = 0.51$. (o) S4. $t = 0.85$. (p) S4. $t = 1.18$.

Figure 5.15: TC29. ROM solutions for $q$ at different time. Each subcase is in a different row.



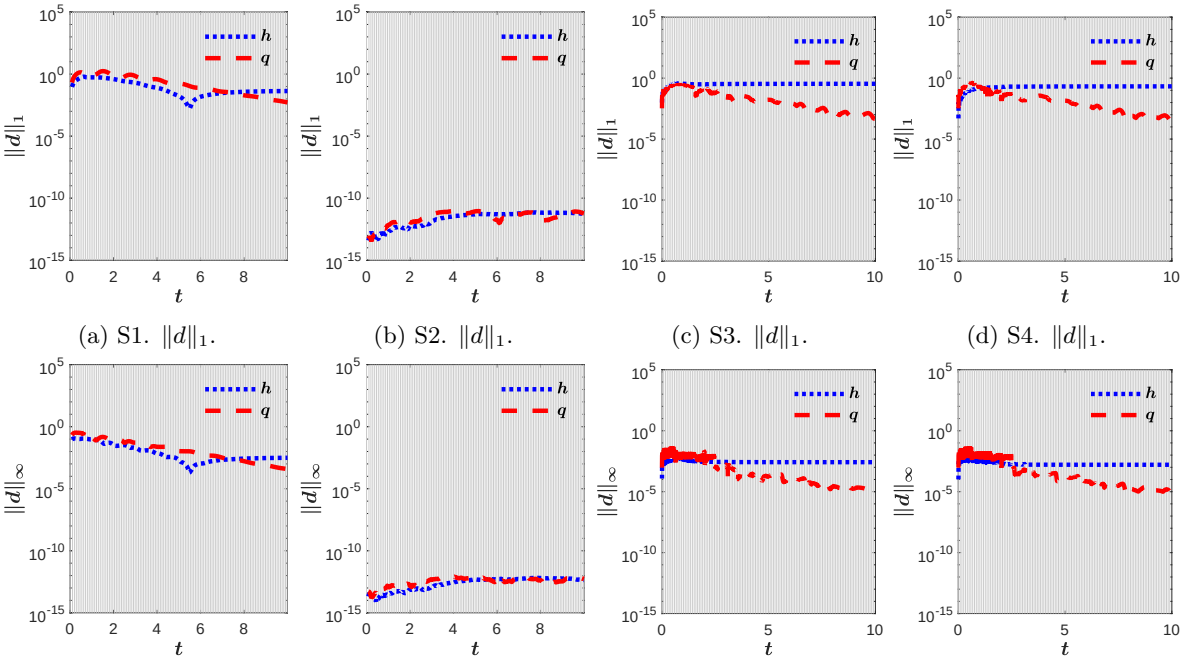

(a) S1. $t = 0.22$. (b) S1. $t = 0.56$. (c) S1. $t = 0.9$. (d) S1. $t = 1.23$.

(e) S2. $t = 0.22$. (f) S2. $t = 0.56$. (g) S2. $t = 0.9$. (h) S2. $t = 1.23$.

(i) S3. $t = 0.17$. (j) S3. $t = 0.51$. (k) S3. $t = 0.85$. (l) S3. $t = 1.18$.

(m) S4. $t = 0.17$. (n) S4. $t = 0.51$. (o) S4. $t = 0.85$. (p) S4. $t = 1.18$.

Figure 5.16: TC29. ROM solutions for the Froude number (2.34) at different time. Each subcase is in a different row.




(a) S1. $\|d\|_1$. (b) S2. $\|d\|_1$. (c) S3. $\|d\|_1$. (d) S4. $\|d\|_1$.

(e) S1. $\|d\|_\infty$. (f) S2. $\|d\|_\infty$. (g) S3. $\|d\|_\infty$. (h) S4. $\|d\|_\infty$.

Figure 5.17: TC29. Differences between ROM and FOM solutions.

**Test case 30. 2D overtopping of a wall.**

This test case reproduces the same problem as the previous one, but in its 2D version, with the aim of checking whether or not it is better to dispense with the wet/dry treatment for efficiency reasons.

The time-space domain is defined as $(x, y, t) \in [0, 6] \times [0, 6] \times [0, 10]$. The initial condition is defined as

$$h(x, y, 0) = \begin{cases} 1, & \text{if } x \leq 3, \\ 0, & \text{if } 3 < x, \end{cases} \tag{5.22}$$

with the following bed level

$$z(x, y) = \begin{cases} 0, & \text{if } x \leq 3, \\ 2, & \text{if } 3 < x. \end{cases} \tag{5.23}$$

Free boundary conditions are imposed on the right boundary and a constant water discharge along the entire west boundary

$$q(0, y, t) = 4, \tag{5.24}$$

with $0 < y < 6$ and $t > 0$.

The spatial domain has been discretized using two different mesh refinements, as indicated in Table 5.15. A rectangular structured mesh has been used, as shown in the Figure 5.18. Regarding the wet/dry treatment, in Subcases 2 and 4 it is applied to the ROM, but it is not in Subcases 1 and 3. All the settings of the problem are shown in Table 5.15. The black volume cells indicate the obstacle that water must overtop.

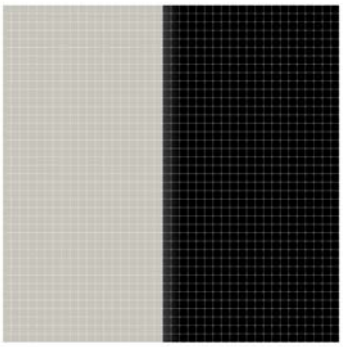


Figure 5.18: TC30. 2D mesh with $I_c = 2500$ and the two subregions of the bed elevation.

| Subcase | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | Wet/dry treatment |
|---|---|---|---|---|---|
| 1 | 36 | 171 | 36 | 171 | No |
| 2 | 36 | 171 | 36 | 171 | Yes |
| 3 | 2500 | 1152 | 5 | 23 | No |
| 4 | 2500 | 1152 | 5 | 23 | Yes |
| $L_x \times L_y = 6 \times 6$, $T = 10$, IC: Eq. (5.22) | | | | | |
| BCs: Eq. (5.24), $z$: Eq. (5.23), $n_b = 0$, CFL $= 0.4$ | | | | | |

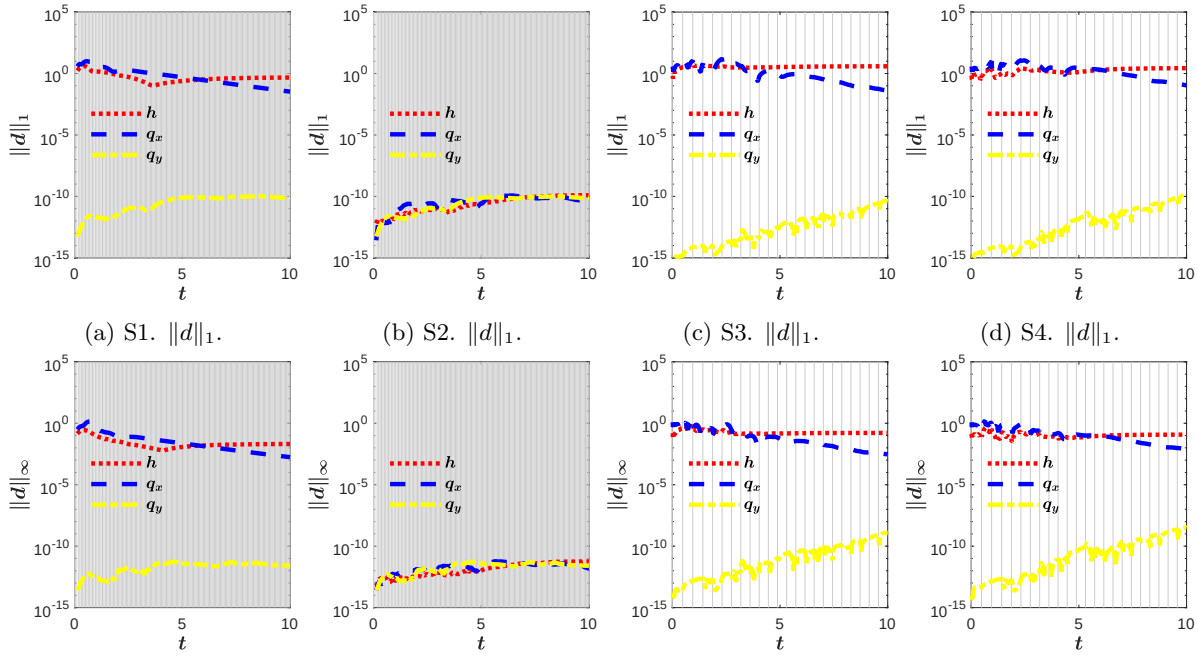
Table 5.15: TC30. Problem subcases and settings.

As indicated by the differences shown in Figure 5.19 for Subcases 1 and 2, the ARoe-based ROM (5.4) in Subcase 2 is able to recover the training solution with machine precision when applying the wet/dry treatment.

Figure 5.20 shows the time evolution of the water depth $h$ computed by the FOM in blue, the ROM for Subcase 3 (without wet/dry treatment) in yellow and the ROM for Subcase 4 (with wet/dry treatment) in red. It is clear that the solution for Subcase 3 does not follow the training solution as well as Subcase 4 does, since it wets areas that should stay dry and overestimates the water depth throughout the whole calculation. This is however not visible when looking at the differences shown by Figures 5.19c-5.19g and 5.19d-5.19h, where both sub-cases are in the same order of magnitude. Nevertheless, as shown in Figure 5.20, it can be concluded that Subcase 4 is better, as it obtains physically satisfactory results. In addition, the wet/dry treatment, far from what it might seem, does not add a significant overhead to the on-line ROM calculation. Table 5.16 shows that the speed-up achieved by both cases is very similar, three orders of magnitude, but it is very slightly lower in Subcase 4.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\|d^h\|_1$ | $4.70 \cdot 10^{-1}$ | $1.27 \cdot 10^{-10}$ | 3.87 | 2.73 |
| $\|d^{q_x}\|_1$ | $3.33 \cdot 10^{-2}$ | $3.64 \cdot 10^{-11}$ | $4.28 \cdot 10^{-2}$ | $1.09 \cdot 10^{-1}$ |
| $\|d^{q_y}\|_1$ | $7.74 \cdot 10^{-11}$ | $7.57 \cdot 10^{-11}$ | $4.06 \cdot 10^{-11}$ | $1.09 \cdot 10^{-10}$ |
| $\|d^h\|_\infty$ | $2.06 \cdot 10^{-2}$ | $5.67 \cdot 10^{-12}$ | $1.64 \cdot 10^{-1}$ | $1.19 \cdot 10^{-1}$ |
| $\|d^{q_x}\|_\infty$ | $1.75 \cdot 10^{-3}$ | $1.59 \cdot 10^{-12}$ | $3.05 \cdot 10^{-3}$ | $7.50 \cdot 10^{-3}$ |
| $\|d^{q_y}\|_\infty$ | $2.53 \cdot 10^{-12}$ | $2.49 \cdot 10^{-12}$ | $1.18 \cdot 10^{-9}$ | $3.18 \cdot 10^{-9}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $2.50 \cdot 10^{-2}$ | $2.50 \cdot 10^{-2}$ | 7.14 | 7.14 |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $6.20 \cdot 10^{-3}$ | $7.47 \cdot 10^{-3}$ | $3.30 \cdot 10^{-3}$ | $3.30 \cdot 10^{-3}$ |
| Speed-up | $\times 4$ | $\times 3$ | $\times 2163$ | $\times 2107$ |

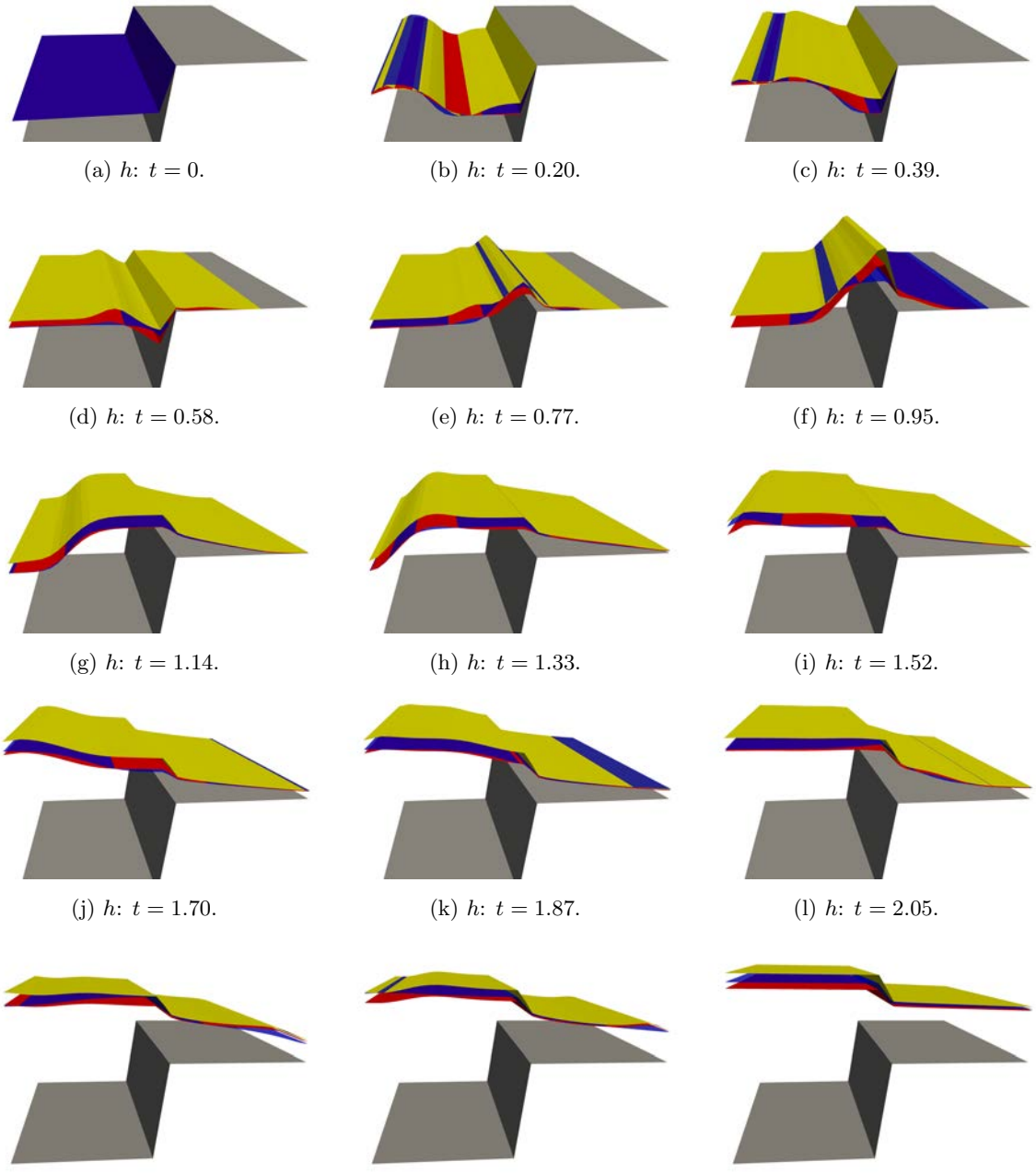
Table 5.16: TC30. Results of efficiency.




(a) S1. $\|d\|_1$. (b) S2. $\|d\|_1$. (c) S3. $\|d\|_1$. (d) S4. $\|d\|_1$.

(e) S1. $\|d\|_\infty$. (f) S2. $\|d\|_\infty$. (g) S3. $\|d\|_\infty$. (h) S4. $\|d\|_\infty$.

Figure 5.19: TC30. differents between the FOM and the ROM solutions using $\|d\|_1$ and $\|d\|_\infty$.



(a) $h$: $t = 0$. (b) $h$: $t = 0.20$. (c) $h$: $t = 0.39$.

(d) $h$: $t = 0.58$. (e) $h$: $t = 0.77$. (f) $h$: $t = 0.95$.

(g) $h$: $t = 1.14$. (h) $h$: $t = 1.33$. (i) $h$: $t = 1.52$.

(j) $h$: $t = 1.70$. (k) $h$: $t = 1.87$. (l) $h$: $t = 2.05$.

(m) $h$: $t = 2.40$. (n) $h$: $t = 2.75$. (o) $h$: $t = 9.74$.

Figure 5.20: TC30. Time evolution of the water depth $h$ computed by the FOM in blue, the ROM for Subcase 3 (without wet/dry treatment) in yellow and the ROM for Subcase 4 (with wet/dry treatment) in red.

**Test case 31. 2D overtopping of a diagonal wall.**

This test case takes the analysis of the application of wet/dry treatment to ROM in 2D problems, but the mesh is structured triangular and has a diagonal obstacle, as shown in Figure 5.21, modelled as a jump in $z$. The black volume cells indicate the obstacle that water must overtop.

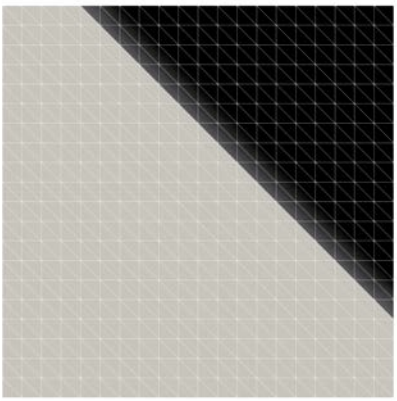


Figure 5.21: TC31. 2D mesh with bed elevation.

The time-space domain is defined as $(x, y, t) \in [0, 6] \times [0, 6] \times [0, 5]$. The initial condition is defined as

$$h(x, y, 0) = \begin{cases} 1, & \text{if } x + y < 7.6, \\ 0, & \text{otherwise,} \end{cases} \tag{5.25}$$

and the obstacle is defined as

$$z(x, y) = \begin{cases} 0, & \text{if } x + y < 7.6, \\ 2, & \text{otherwise.} \end{cases} \tag{5.26}$$

Free boundary conditions are imposed on all the boundaries, except for the west boundary at $t = 0$, where a constant water discharge is imposed

$$q(0, y, t) = 4, \tag{5.27}$$

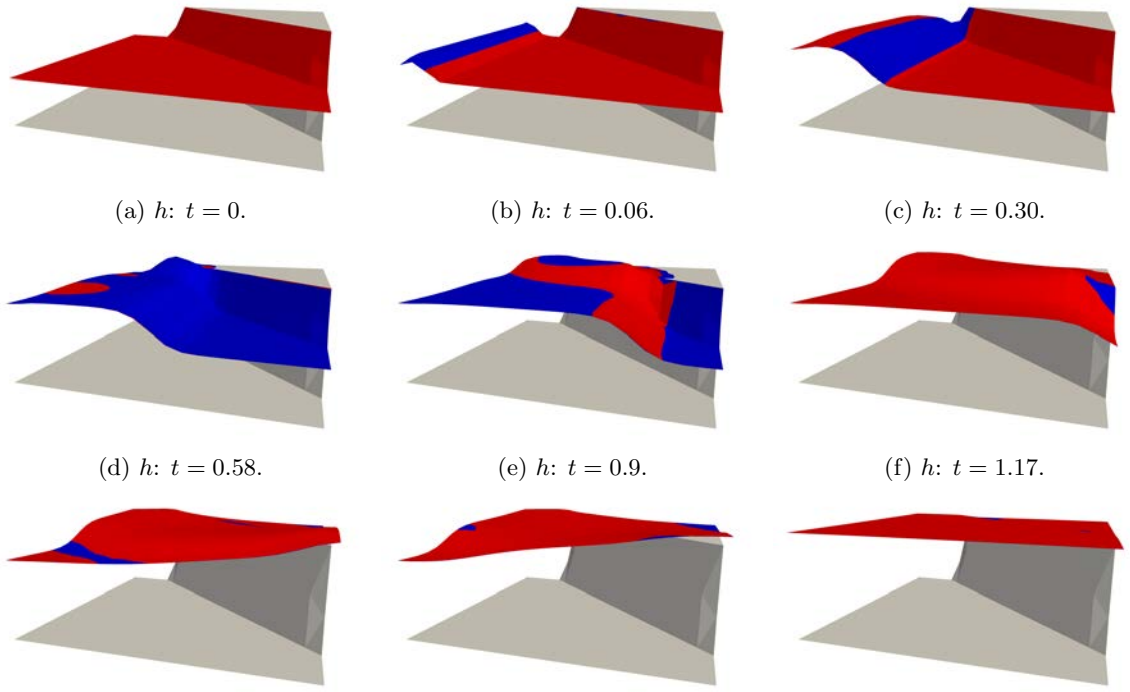
with $0 < y < 6$ and $t > 0$.

The spatial domain is discretized using 800 cells and the wet/dry treatmente is applied. All the settings of the problem are shown in Table 5.17.

| $L_x \times L_y$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | Wet/dry treatment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $6 \times 6$ | 5 | (5.25) | (5.27) | (5.26) | 0 | 0.4 | 800 | 775 | 5 | 77 | Yes |

Table 5.17: TC31. Problem settings.

The time evolution of the water depth $h$ computed by the FOM in blue and the ROM in red is shown in Figure 5.22. At first glance, a good match between the two solutions can be observed. This is confirmed by the time evolution of the error, shown in Figure 5.23 and Table 5.18. The speed-up achieved, contained in Table 5.18, reaches two orders of magnitude, so the ROM is so much faster than the FOM in 2D problems, even when it includes the wet/dry treatment in its online phase.



(a) $h$: $t = 0$. (b) $h$: $t = 0.06$. (c) $h$: $t = 0.30$.

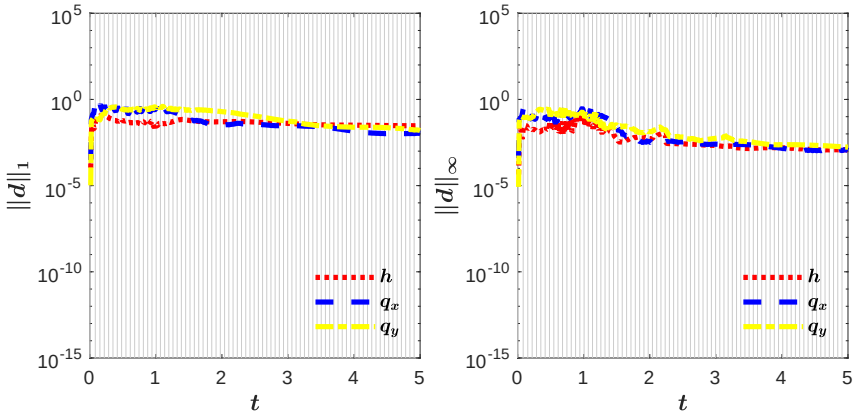
(d) $h$: $t = 0.58$. (e) $h$: $t = 0.9$. (f) $h$: $t = 1.17$.

(g) $h$: $t = 1.43$. (h) $h$: $t = 1.62$. (i) $h$: $t = 5$.

Figure 5.22: TC31. Time evolution of the water depth $h$ computed by the FOM in blue and the ROM in red.

| $\|d^h\|_1$ | $\|d^{q_x}\|_1$ | $\|d^{q_y}\|_1$ |
|---|---|---|
| $2.96 \cdot 10^{-2}$ | $1.56 \cdot 10^{-2}$ | $1.06 \cdot 10^{-2}$ |
| $\|d^h\|_\infty$ | $\|d^{q_x}\|_\infty$ | $\|d^{q_y}\|_\infty$ |
| $2.96 \cdot 10^{-2}$ | $1.08 \cdot 10^{-3}$ | $1.72 \cdot 10^{-3}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
| 1.71 | $4.20 \cdot 10^{-3}$ | $\times 407$ |

Table 5.18: TC31. Differences between FOM and ROM solutions and CPU times required.




(a) $\|d\|_1$. (b) $\|d\|_\infty$.

Figure 5.23: TC31. Time evolution of the differences between FOM and ROM solutions.

## 5.3 ROMs and parametrized SWEs

The results of applying ROMs to the parametrized SWEs are presented in this section. In the following numerical results, Manning's coefficient, bed shape, initial and boundary conditions are considered as input parameters.

The results reflect the same limitation as in Section 4.2 with respect to including position of the IC as an input parameter when using linearized ROMs. To deal with this drawback, use is made of the ROM based on the Lax-Friedrichs scheme (5.6).

An efficiency analysis is proposed in these cases to find the best trio of ROM configuration parameters: $(M_{\text{POD}}, M_W, M_{\text{train}})$. For this purpose, the a posteriori criterion presented in Section 4.1.4 is used.

### 5.3.1 Test case 32. Input parameter: Manning coefficient

In this first case, a training set of 20 samples has been computed with the ARoe-based FOM for 20 random values of the Manning coefficient. These training values and the target value are shown in Figure 5.24. Note that only sample 14 presents a larger value of $n_b$ than the target value.

The time-space domain is defined as $(x,t) \in [0,40] \times [0,5]$. The initial condition is defined as a 1D dambreak

$$h(x,0) = \begin{cases} 2, & \text{if } x \leq 8, \\ 1, & \text{if } x > 8, \end{cases} \quad q(x,0) = 0, \tag{5.28}$$

and no bed slope is set, $z(x) = 0, \forall x$. Free boundary conditions are imposed on both sides.

The spatial domain is discretized by means of 200 cells. The time step is dynamically computed according to (2.23) with CFL = 0.9.

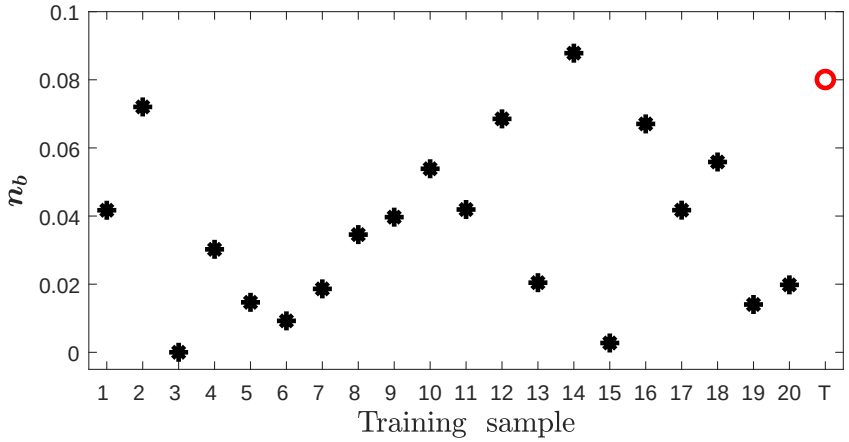



Figure 5.24: TC32. Friction training values in black and target value in red.

The a posteriori criterion is used to find the optimal values of the ROM parameters. With this purpose, $3 \times 3 \times 3 = 27$ subcases have been defined, as indicated in Table 5.20, as result of the combinations of the following values

$$\begin{aligned} M_{\text{POD}} &\in \{2, 5, 10\}\,, \\ M_W &\in \{154, 308, 616\}\,, \\ M_{\text{train}} &\in \{5, 10, 20\}\,. \end{aligned} \tag{5.29}$$

The rest of the settings are shown in Table 5.19.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 5 | Eq. (5.28) | Free | 0 | Fig. 5.24 | 0.9 | 200 | 1232 | Eq. (5.29) | | |

Table 5.19: TC32. Problem settings.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 2 | 5 | 10 | 2 | 5 | 10 | 2 | 5 | 10 |
| $M_W$ | 154 | 154 | 154 | 308 | 308 | 308 | 616 | 616 | 616 |
| $M_{\text{train}}$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 2 | 5 | 10 | 2 | 5 | 10 | 2 | 5 | 10 |
| $M_W$ | 154 | 154 | 154 | 308 | 308 | 308 | 616 | 616 | 616 |
| $M_{\text{train}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 2 | 5 | 10 | 2 | 5 | 10 | 2 | 5 | 10 |
| $M_W$ | 154 | 154 | 154 | 308 | 308 | 308 | 616 | 616 | 616 |
| $M_{\text{train}}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Table 5.20: TC32. Problem subcases.

As shown in Figure 5.25, the candidate subcases to be the optimal ones are 1, 19 and 21. The three of them are solved using low numbers of POD modes, as indicated in Table 5.20, but Subcases 19 and 21 need 20 training samples and Subcase 1 just 5. However, Subcases 1 and 19 present very high speed-up in the on-line phase and all of them show similar differences (see Table 5.21). For all subcases, 154 time windows are the optimal value of $M_W$. So it can be concluded that friction is a physical quantity that can be easily malleable when predicting solutions, as has been shown in this test case.

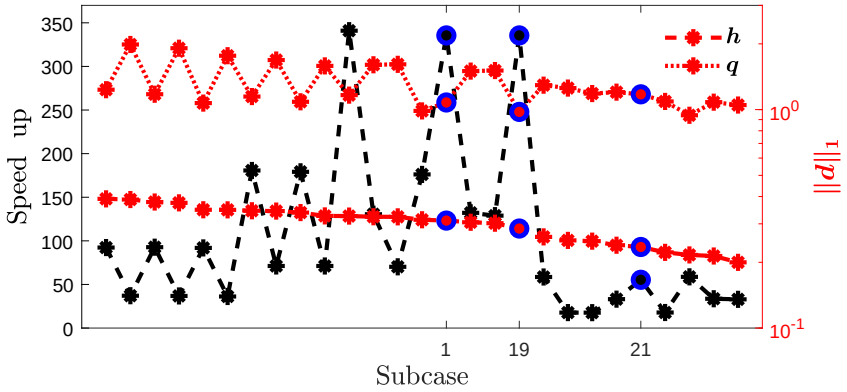
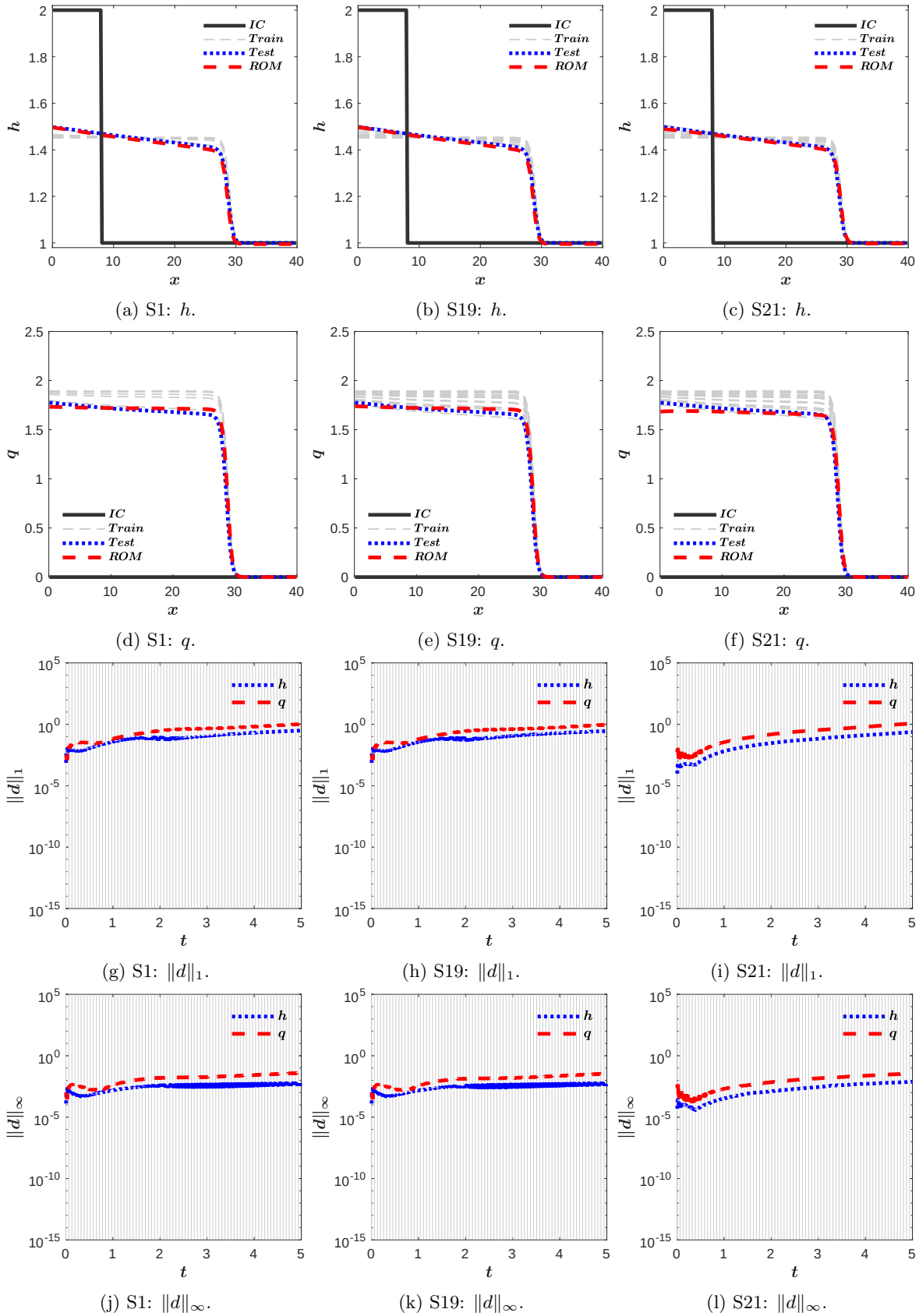



Figure 5.25: TC32. Candidate sucbases.

| Subcase | 1 | 19 | 21 |
|---|---|---|---|
| $M_{\text{POD}}$ | 2 | 2 | 10 |
| $M_W$ | 154 | 154 | 154 |
| $M_{\text{train}}$ | 5 | 20 | 20 |
| $\|d^h\|_1$ | $3.10 \cdot 10^{-1}$ | $2.85 \cdot 10^{-1}$ | $2.35 \cdot 10^{-1}$ |
| $\|d^q\|_1$ | 1.08 | $9.76 \cdot 10^{-1}$ | 1.17 |
| $\|d^h\|_\infty$ | $3.20 \cdot 10^{-2}$ | $3.16 \cdot 10^{-2}$ | $3.84 \cdot 10^{-2}$ |
| $\|d^q\|_\infty$ | $1.92 \cdot 10^{-1}$ | $1.68 \cdot 10^{-1}$ | $1.78 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $2.11 \cdot 10^{-1}$ | $2.11 \cdot 10^{-1}$ | $2.11 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $6.30 \cdot 10^{-4}$ | $6.30 \cdot 10^{-4}$ | $3.82 \cdot 10^{-3}$ |
| Speed-up | $\times 336$ | $\times 336$ | $\times 55$ |

Table 5.21: TC32. Efficiency results.




(a) S1: $h$. (b) S19: $h$. (c) S21: $h$.

(d) S1: $q$. (e) S19: $q$. (f) S21: $q$.

(g) S1: $\|d\|_1$. (h) S19: $\|d\|_1$. (i) S21: $\|d\|_1$.

(j) S1: $\|d\|_\infty$. (k) S19: $\|d\|_\infty$. (l) S21: $\|d\|_\infty$.

Figure 5.26: TC32. ROM solutions of optimal candidates for $h$ and $q$ and their differences with respect to FOM solutions.

### 5.3.2 Test case 33. Input parameter: shape of the bed slope

As concluded in Section 4.1, linearized ROMs are not able to predict solutions if the position of a discontinuity in the bed is modified. This is shown in this test case.

The time-space domain is defined as $(x,t) \in [0,40] \times [0,5]$. A 1D dambreak has been set as follows

$$h(x,0) = \begin{cases} 2, & \text{if } x \leq 8, \\ 1, & \text{if } x > 8, \end{cases} \tag{5.30}$$

over a discontinuity in the bed, given by

$$z(x) = \begin{cases} z_L, & \text{if } x \leq 8, \\ z_R, & \text{if } x > 8, \end{cases} \tag{5.31}$$

where the levels on both sides of the discontinuity act as input parameters $\mu_1 = z_L$, $\mu_2 = z_R$. Free boundary conditions are considered. The spatial domain is discretized by means of 200 cells. The time step is dynamically computed according to (2.23) with CFL $= 0.9$.

The training set has been computed with 20 random values of the bed levels, which act here as input parameters and can be seen in Figure 5.27. Apart from that, $3 \times 3 \times 3 = 27$ subcases have been defined as result of the combinations between the following values of the parameters of the ROM

$$\begin{aligned} M_{\text{POD}} &\in \{5, 15, 30\}, \\ M_W &\in \{154, 308, 616\}, \\ M_{\text{train}} &\in \{5, 10, 20\}. \end{aligned} \tag{5.32}$$

Table 5.22 shows the configuration of the parameters of the ROM for each subcase. All the settings of the problem are shown in Table 5.23.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 153 | 153 | 153 | 307 | 307 | 307 | 615 | 615 | 615 |
| $M_{\text{train}}$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 153 | 153 | 153 | 307 | 307 | 307 | 615 | 615 | 615 |
| $M_{\text{train}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $M_W$ | 153 | 153 | 153 | 307 | 307 | 307 | 615 | 615 | 615 |
| $M_{\text{train}}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Table 5.22: TC33. Problem subcases.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ |
|---|---|---|---|---|---|
| 40 | 5 | Eq. (5.30) | Free | Eq. (5.31) & Fig. 5.27 | 0 |
| CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
| 0.9 | 200 | 1230 | Eq. (5.32) | Eq. (5.32) | Eq. (5.32) |

Table 5.23: TC33. Problem settings.

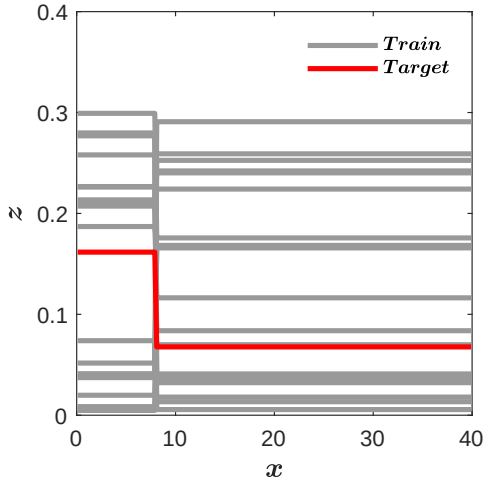



Figure 5.27: TC33. Boundary conditions training samples.

Following the a-posterior criterion, as shown in Figure 5.28, Subcases 23, 2 and 5 are the canditates to the optimal values of $(M_{\text{POD}}, M_W, M_{\text{train}})$. Figure 5.29 shows the solutions for these candidate subcases. Even though their speed-ups reach just one order of magnitude (Table 5.24), the ARoe-based ROM is able to predict solutions with different bed slopes from those of the training with high accuracy, as shown in Figures 5.30. In the following test case it is shown what happens if the position of a discontinuity in the bed is considered as an input parameter.

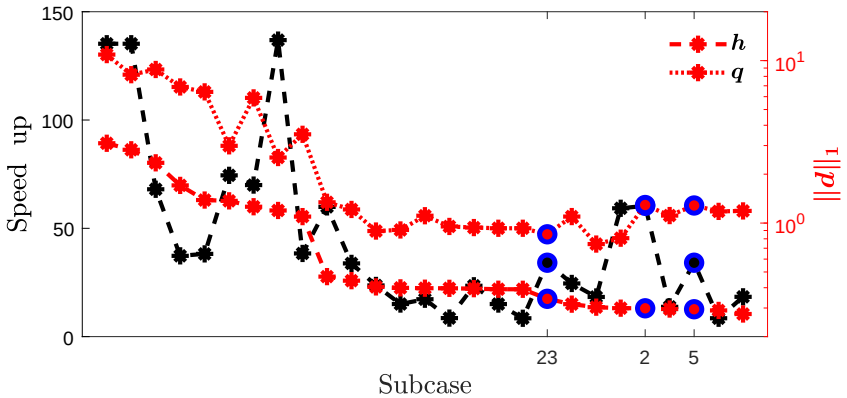
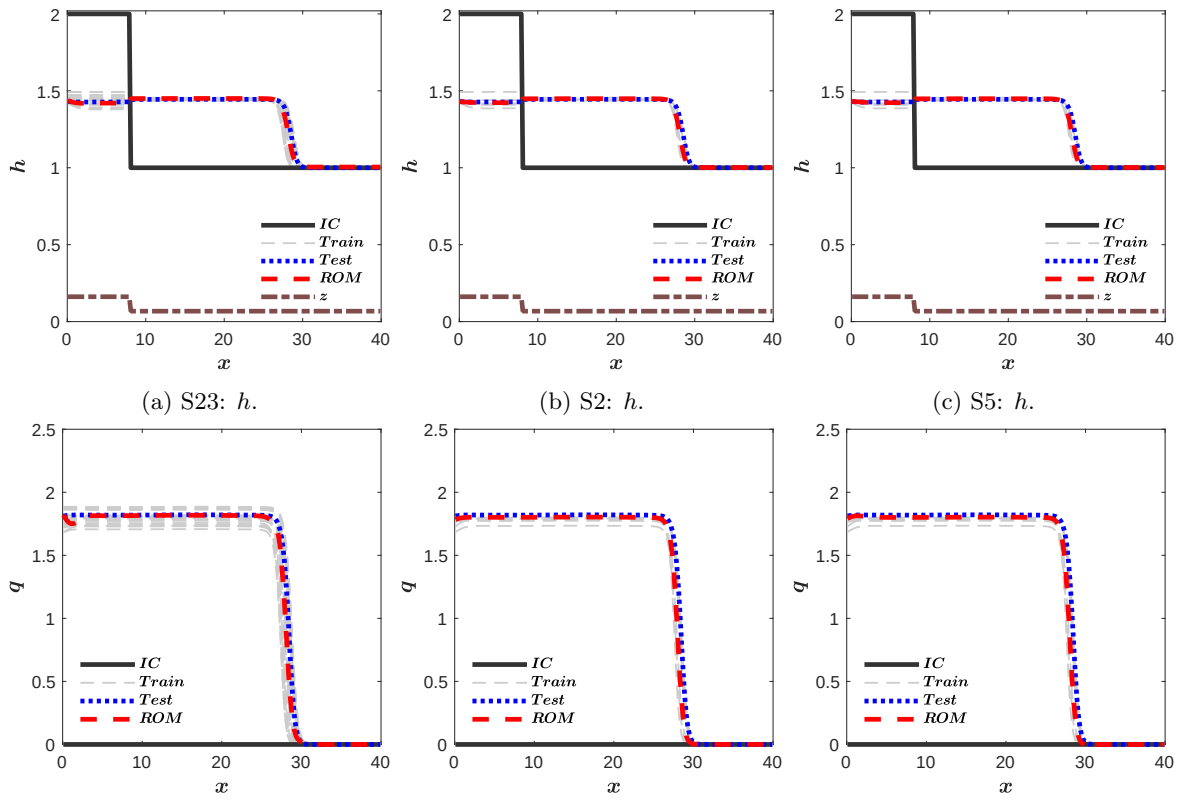



Figure 5.28: TC33. Candidate subcases.



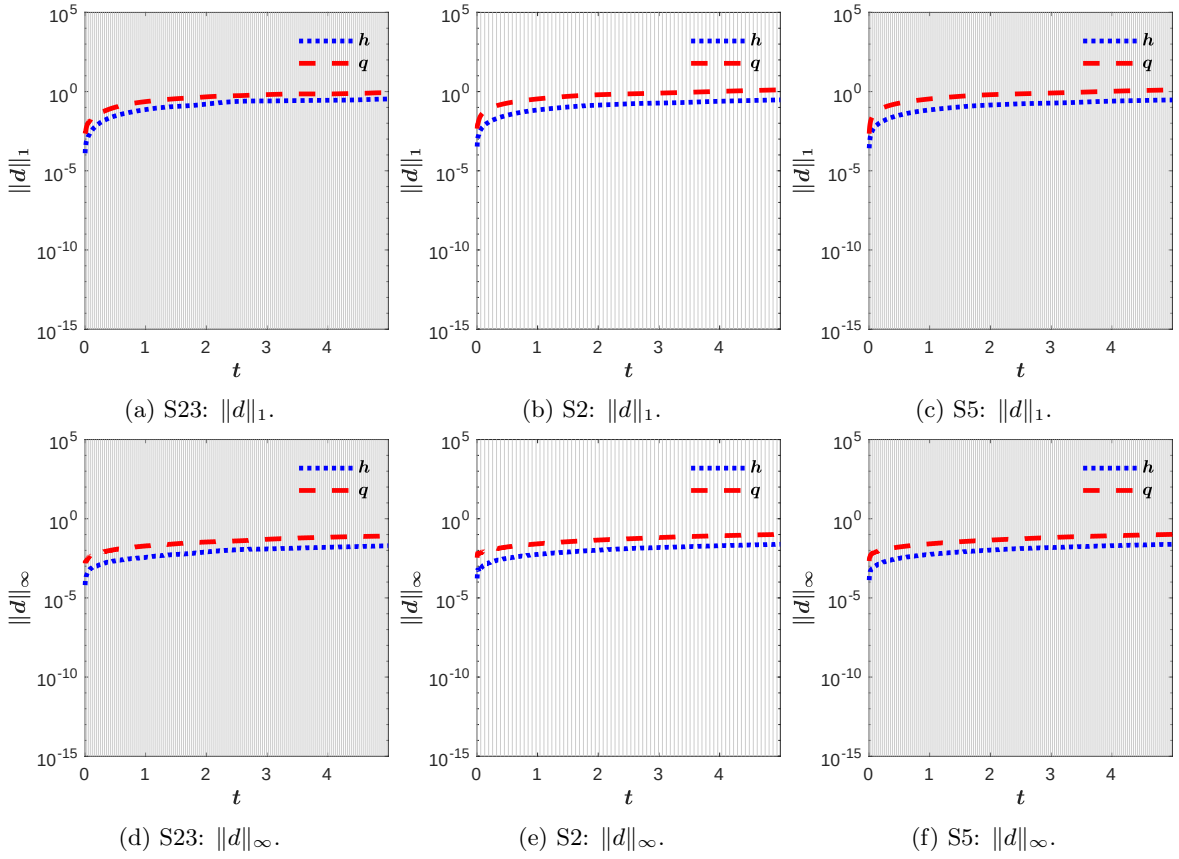

(a) S23: $h$. (b) S2: $h$. (c) S5: $h$.

(d) S23: $q$. (e) S2: $q$. (f) S5: $q$.

Figure 5.29: TC33. ROM solutions of optimal candidates for $h$ and $q$.

| Subcase | 23 | 2 | 5 |
|---|---|---|---|
| $M_{\text{POD}}$ | 10 | 10 | 10 |
| $M_W$ | 307 | 153 | 307 |
| $M_{\text{train}}$ | 20 | 5 | 5 |
| $\|d^h\|_1$ | $3.42 \cdot 10^{-1}$ | $2.99 \cdot 10^{-1}$ | $2.95 \cdot 10^{-1}$ |
| $\|d^q\|_1$ | $8.53 \cdot 10^{-1}$ | 1.29 | 1.28 |
| $\|d^h\|_\infty$ | $9.78 \cdot 10^{-2}$ | $1.21 \cdot 10^{-1}$ | $1.21 \cdot 10^{-1}$ |
| $\|d^q\|_\infty$ | $4.05 \cdot 10^{-1}$ | $5.13 \cdot 10^{-1}$ | $5.15 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $2.20 \cdot 10^{-1}$ | $2.20 \cdot 10^{-1}$ | $2.20 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $6.46 \cdot 10^{-3}$ | $3.65 \cdot 10^{-3}$ | $6.46 \cdot 10^{-3}$ |
| Speed-up | $\times 34$ | $\times 60$ | $\times 34$ |

Table 5.24: TC33. Efficiency results.




(a) S23: $\|d\|_1$. (b) S2: $\|d\|_1$. (c) S5: $\|d\|_1$.

(d) S23: $\|d\|_\infty$. (e) S2: $\|d\|_\infty$. (f) S5: $\|d\|_\infty$.

Figure 5.30: TC33. Differences between FOM and ROM solutions.

### 5.3.3 Test case 34. Input parameter: defining position of the bed slope

As it was shown in Section 4.2, linearized ROMs are not able to predict solutions when the position of discontinuity is considered as an input parameter. This can also be extended to the position of a discontinuity in the bed. Two sub-cases are compared below, in which the same problem is solved using two different numerical schemes: the ARoe-based ROM (5.2) (linearized) and the LF-based ROM (5.6) (non-linear). With this purpose, the same problem is considered as in the previous case, with the following IC

$$h(x,0) = \begin{cases} 2, & \text{if } x \leq 8, \\ 1, & \text{if } x > 8. \end{cases} \tag{5.33}$$

The bed slope is defined as

$$z(x) = \begin{cases} z_L, & \text{if } x \leq x_0, \\ z_R, & \text{if } x > x_0, \end{cases} \tag{5.34}$$

where the levels on both sides of the discontinuity and its position act as input parameters

$$\mu_1 = z_L, \quad \mu_2 = z_R, \quad \mu_3 = x_0.$$

The training and target values are shown in Figure 5.31. Free boundary conditions are considered.

In Subcase 1, the ARoe-based ROM (5.2) is solved using 20 POD modes, 986 time windows and 2 training samples; and, in Subcase 2, the LF-based ROM (5.6), just 1 time window, as indicated in Table 5.25.

| Subcase | $M_W$ | FOM/ROM |
|---|---|---|
| 1 | 986 | ARoe |
| 2 | 1 | LF |
| $L = 40$, $T = 5$, IC: Eq. (5.33), BCs: free, $z$: Eq. 5.31, $n_b = 0$ | | |
| CFL $= 0.9$, $I_x = 200$, $N_T = 986$, $M_{\text{POD}} = 20$, $M_{\text{train}} = 2$ | | |

Table 5.25: TC34. Problem settings.

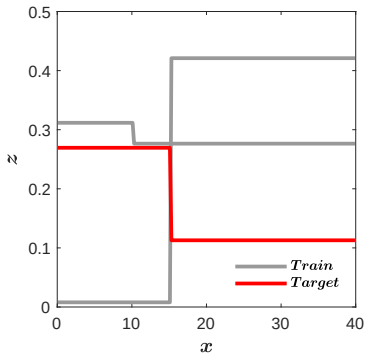
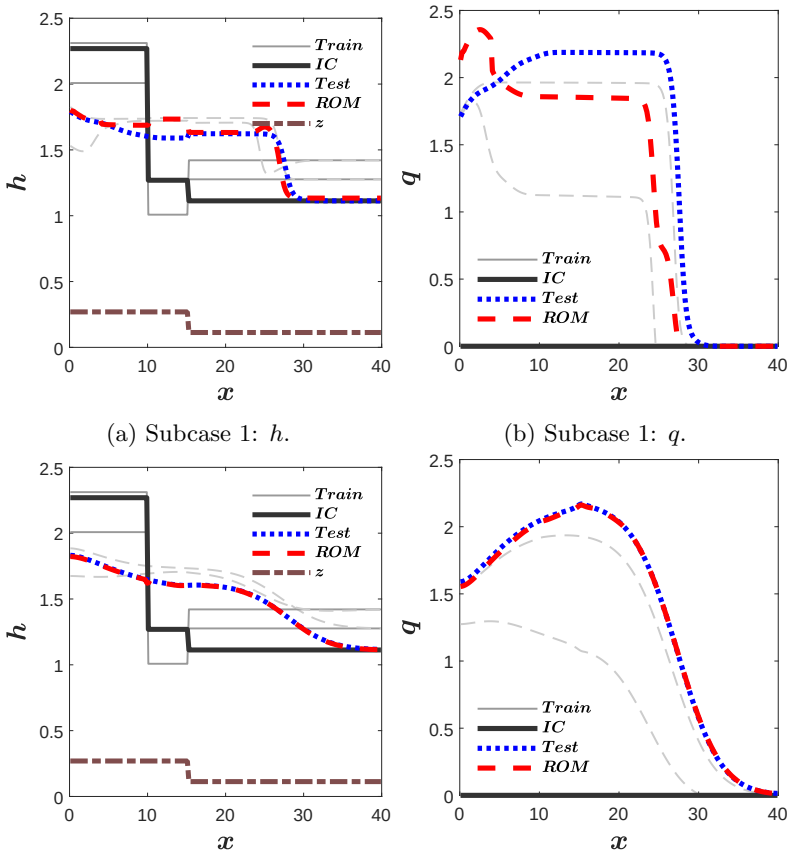



Figure 5.31: TC34. Training and target bed slopes.

The ARoe-based ROM (5.2) is not able to accurately predict the target position of the discontinuity, as it can be seen in Figures 5.32a and 5.32b. Moreover, since the number of time windows is maximum, the speed-up is just $\times 5$. On the contrary, the LF-based ROM (5.6) satisfactory predicts the solution. However, it is just as fast as the LF-based FOM (2.26), since the speed-up is $\times 1$, which is due to its non-linearity, as seen in Figures 5.32c and 5.32d and Table 5.26.

| S | $\|d^h\|_1$ | $\|d^q\|_1$ | $\|d^h\|_\infty$ | $\|d^q\|_\infty$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
|---|---|---|---|---|---|---|---|
| 1 | 1.16 | $1.08 \cdot 10^{1}$ | $3.61 \cdot 10^{-2}$ | $1.15 \cdot 10^{-1}$ | $1.90 \cdot 10^{-1}$ | $3.86 \cdot 10^{-2}$ | $\times 5$ |
| 2 | $3.08 \cdot 10^{-1}$ | $7.09 \cdot 10^{-1}$ | $6.00 \cdot 10^{-3}$ | $9.73 \cdot 10^{-3}$ | $1.06 \cdot 10^{-1}$ | $7.37 \cdot 10^{-2}$ | $\times 1$ |

Table 5.26: TC34. Efficiency solutions.




(a) Subcase 1: $h$.

(b) Subcase 1: $q$.

(c) Subcase 2: $h$.

(d) Subcase 2: $q$.

Figure 5.32: TC34. ROM solutions.

### 5.3.4 Test case 35. Input parameters: initial discontinuity

The time-space domain is defined as $(x,t) \in [0,40] \times [0,5]$. A 1D dambreak has been set

$$h(x,0) = \begin{cases} h_L, & \text{if } x \leq 8, \\ h_R, & \text{if } x > 8, \end{cases} \tag{5.35}$$

over a flat bed, $z(x) = 0, \forall x$, where the water levels act as input parameters

$$\mu_1 = h_L, \quad \mu_2 = h_R.$$

Free boundary conditions are considered. The spatial domain is discretized by means of 200 volume cells. The time step is dynamically computed according to (2.23) with CFL $= 0.9$.

The training set has been computed with 20 random values of $h_L$ and $h_R$, which act here as input parameters and can be seen in Figure 5.33. Apart from that, $3 \times 3 \times 3 = 27$ subcases have been defined as result of the combinations between the following values of the parameters

$$
\begin{aligned}
M_{\text{POD}} &\in \{5, 15, 30\}\,, \\
M_W &\in \{75, 150, 301\}\,, \\
M_{\text{train}} &\in \{2, 10, 20\}\,,
\end{aligned}
\tag{5.36}
$$

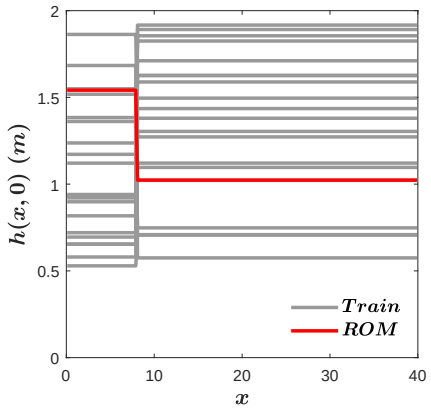



Figure 5.33: TC35. Initial conditions training samples.

Table 5.27 shows the configuration of the parameters of the ROM for each subcase. All the settings of the problem are shown in Table 5.28.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 75 | 75 | 75 | 150 | 150 | 150 | 301 | 301 | 301 |
| $M_{\text{train}}$ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 75 | 75 | 75 | 150 | 150 | 150 | 301 | 301 | 301 |
| $M_{\text{train}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 75 | 75 | 75 | 150 | 150 | 150 | 301 | 301 | 301 |
| $M_{\text{train}}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Table 5.27: TC35. Problem subcases.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ |
|---|---|---|---|---|---|
| 40 | 5 | Eq. (5.35) + Fig. 5.33 | Free | 0 | 0 |
| CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
| 0.9 | 200 | 1206 | Eq. (5.36) | Eq. (5.36) | Eq. (5.36) |

Table 5.28: TC35. Problem settings.

Subcases 22, 5 and 11 are proposed as candidates for the optimal values of $(M_{\text{POD}}, M_W, M_{\text{train}})$. As shown in Figures 5.35c and 5.35f, Subcase 11 presents big discrepancies with respect to the test solution for both $h$ and $q$, whereas Subcases 22 and 5 obtain better results, as shown in Table 5.29. In terms of speed-up, Subcase 5 is slower than Subcase 22, but it only needs 2 training samples, as shown in Table 5.29.

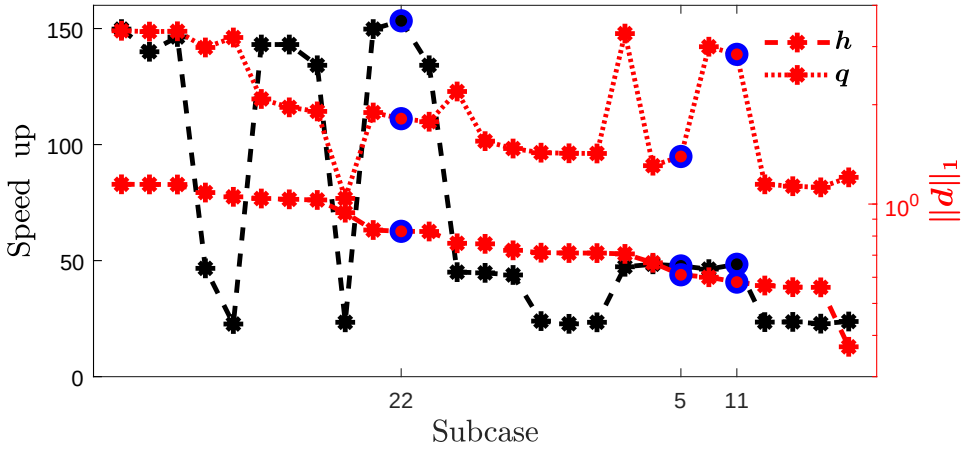
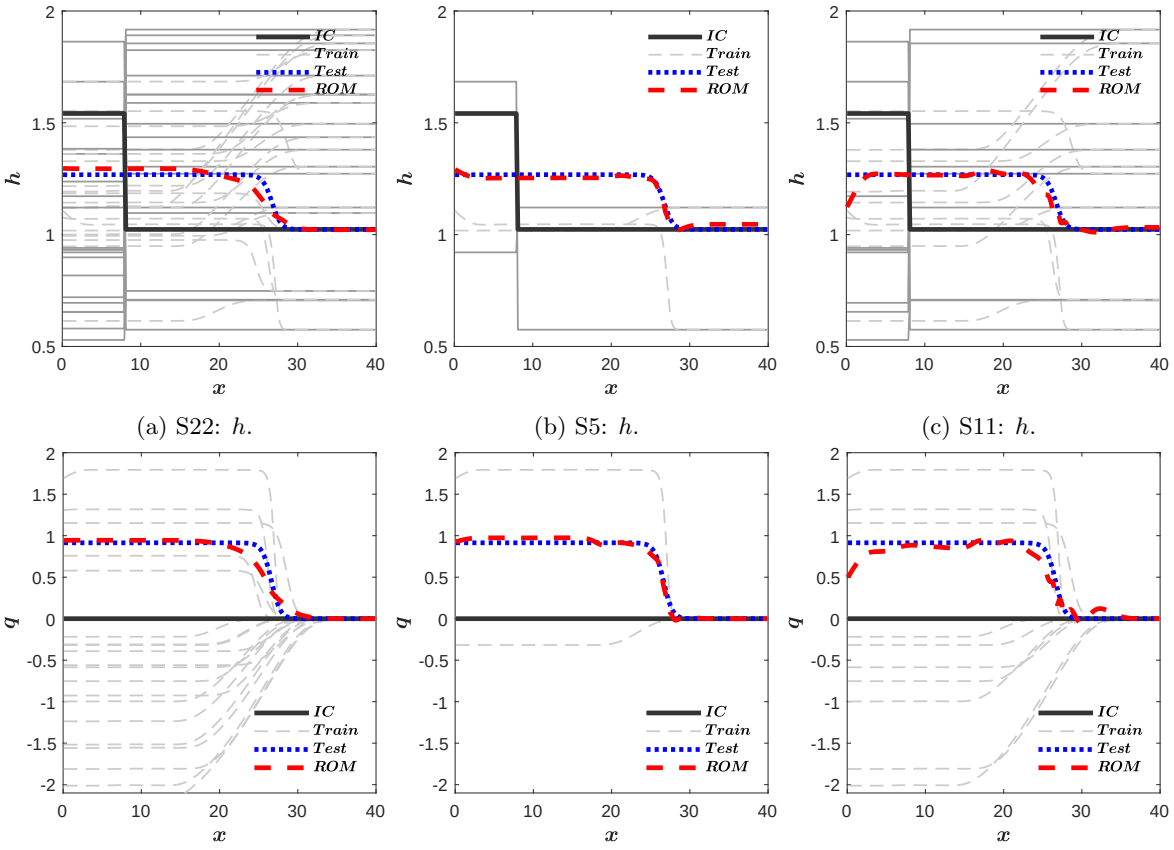



Figure 5.34: TC35. Candidate subcases.

| Subcase | 22 | 5 | 11 |
|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 15 | 15 |
| $M_W$ | 150 | 150 | 75 |
| $M_{\text{train}}$ | 20 | 2 | 10 |
| $\|d^h\|_1$ | $8.27 \cdot 10^{-1}$ | $6.11 \cdot 10^{-1}$ | $5.80 \cdot 10^{-1}$ |
| $\|d^q\|_1$ | 1.81 | 1.39 | 2.84 |
| $\|d^h\|_\infty$ | $1.69 \cdot 10^{-2}$ | $5.67 \cdot 10^{-3}$ | $2.88 \cdot 10^{-2}$ |
| $\|d^q\|_\infty$ | $5.13 \cdot 10^{-2}$ | $1.81 \cdot 10^{-2}$ | $8.29 \cdot 10^{-2}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $6.44 \cdot 10^{-4}$ | $6.44 \cdot 10^{-4}$ | $6.44 \cdot 10^{-4}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $4.20 \cdot 10^{-6}$ | $1.35 \cdot 10^{-5}$ | $1.33 \cdot 10^{-5}$ |
| Speed-up | $\times 153$ | $\times 48$ | $\times 48$ |

Table 5.29: TC35. Efficiency results.




(a) S22: $h$. (b) S5: $h$. (c) S11: $h$.

(d) S22: $q$. (e) S5: $q$. (f) S11: $q$.

Figure 5.35: TC35. ROM solutions of optimal candidates.

### 5.3.5 Test case 36. Input parameters: position of the initial discontinuity

Linearized ROMs are not able to predict solutions when the position of the discontinuity is considered as an input parameter. The ARoe-based ROM (5.2) (linearized) and the LF-based ROM (5.6) (non-linear) are used to solve this test case.

With this purpose, the same problem is considered as in the previous TC35, with the following initial condition

$$h(x,0) = \begin{cases} h_L, & \text{if } x \leq x_0, \\ h_R, & \text{if } x > x_0, \end{cases} \tag{5.37}$$

where the levels on both sides of the discontinuity and its position act as input parameters

$$\mu_1 = h_L, \quad \mu_2 = h_R, \quad \mu_3 = x_0.$$

The training and target values are shown in Figure 5.36. Free boundary conditions are considered. In Subcase 1, the ARoe-based ROM (5.2) is solved using 20 POD modes, 986 time windows and 2 training samples; and, in Subcase 2, the LF-based ROM (5.6), with just 1 time window, as indicated in Table 5.30.

| Subcase | $M_W$ | FOM/ROM |
|---|---|---|
| 1 | 986 | ARoe |
| 2 | 1 | LF |
| $L = 40$, $T = 5$, IC: Eq. (5.37), BCs: free, $z = 0$, $n_b = 0$ | | |
| CFL $= 0.9$, $I_x = 200$, $N_T = 933$, $M_{\text{POD}} = 20$, $M_{\text{train}} = 2$ | | |

Table 5.30: TC36. Problem settings.

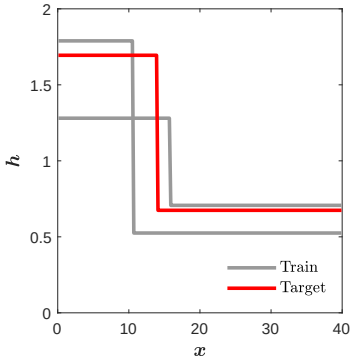
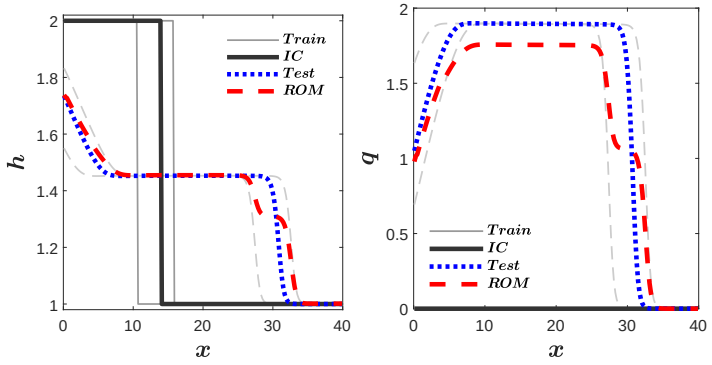



Figure 5.36: TC36. Training and target initial conditions.

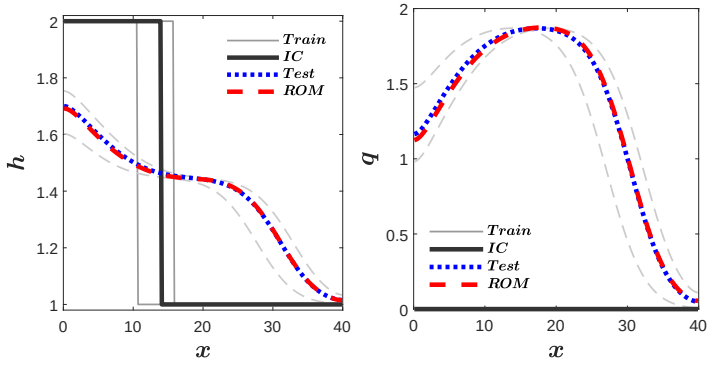
The ARoe-based ROM fails at predicting the final solution, as it can be seen in Figures 5.37a and 5.37b. On the contrary, the LF-based ROM (5.6) satisfactory predicts the solution, but it is only 2 times faster than the LF-based FOM, as seen in Figures 5.37c and 5.37d and Table 5.31.




(a) Subcase 1: $h$.




(b) Subcase 1: $q$.




(c) Subcase 2: $h$.




(d) Subcase 2: $q$.

Figure 5.37: TC36. ROM solutions.

| S | $\|d^h\|_1$ | $\|d^q\|_1$ | $\|d^h\|_\infty$ | $\|d^q\|_\infty$ | $\tau_{\text{CPU}}^{\text{test}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
|---|---|---|---|---|---|---|---|
| 1 | 2.22 | $1.35 \cdot 10^{1}$ | $3.87 \cdot 10^{-2}$ | $2.46 \cdot 10^{-1}$ | $1.81 \cdot 10^{-1}$ | $3.58 \cdot 10^{-2}$ | $\times 5$ |
| 2 | $8.41 \cdot 10^{-2}$ | $2.85 \cdot 10^{-1}$ | $8.42 \cdot 10^{-4}$ | $2.77 \cdot 10^{-3}$ | $1.10 \cdot 10^{-1}$ | $6.86 \cdot 10^{-2}$ | $\times 2$ |

Table 5.31: TC36. Efficiency solutions.

### 5.3.6 Test case 37. Input parameter: boundary condition

This test case is an example of parametric prediction applied to boundary conditions. The time-space domain is defined as $(x,t) \in [0,40] \times [0,5]$ The following piecewise constant function is imposed $x = 0$ to the water discharge

$$q(0,t) = \begin{cases} q_0, & \text{if } 0.1 \leq x \leq 2, \\ 0, & \text{otherwise,} \end{cases} \tag{5.38}$$

where the level of the water discharge act as the input parameter

$$\mu = q_0.$$

A free boundary condition is imposed on the right boundary. The initial condition is defined as

$$h(x,0) = 1, \quad q(x,0) = 0. \tag{5.39}$$

The time-space domain $(x,t) \in [0,40] \times [0,5]$ is discretized by means of 200 volume cells. The time step is dynamically computed according to (2.23) with CFL $= 0.9$.

The training set has been computed with 20 random values of the level of the water discharge $q_0$, which act here as input parameters and can be seen in Figure 5.38. $3 \times 3 \times 3 = 27$ subcases have been defined as a result of the combinations between the following values of the parameters of the ROM

$$\begin{aligned} M_{\text{POD}} &\in \{5, 15, 30\}, \\ M_W &\in \{109, 218, 436\}, \\ M_{\text{train}} &\in \{5, 10, 20\}, \end{aligned} \tag{5.40}$$

Table 5.32 shows the configuration of the parameters of the ROM for each subcase. All the settings of the problem are shown in Table 5.33.

| Subcase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 109 | 109 | 109 | 218 | 218 | 218 | 436 | 436 | 436 |
| $M_{\text{train}}$ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Subcase | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 109 | 109 | 109 | 218 | 218 | 218 | 436 | 436 | 436 |
| $M_{\text{train}}$ | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Subcase | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $M_{\text{POD}}$ | 5 | 15 | 30 | 5 | 15 | 30 | 5 | 15 | 30 |
| $M_W$ | 109 | 109 | 109 | 218 | 218 | 218 | 436 | 436 | 436 |
| $M_{\text{train}}$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Table 5.32: TC37. Problem subcases.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ | $M_{\text{train}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 5 | Eq. (5.39) | Eq. (5.38) | 0 | 0 | 0.9 | 200 | 872 | Eq. (5.40) | | |

Table 5.33: TC37. Problem settings.

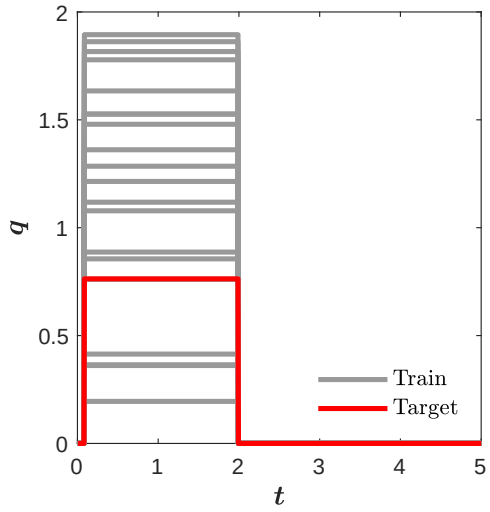



Figure 5.38: TC37. BCs training values.

Subcases 11, 1 and 2 are the proposed candidates, as shown in Figure 5.39. Figure 5.40 shows the time evolution of the their solutions for $h$ and $q$. The number of time windows is the same in all three subcases, but the number of POD modes used is smaller in Subcase 1, so that the speed-up achieved is $\times 143$, as shown in Table 5.34. Only 5 training samples are enough to obtain efficient solutions in terms of accuracy and CPU time, as measured in the differences shown in Table 5.34.

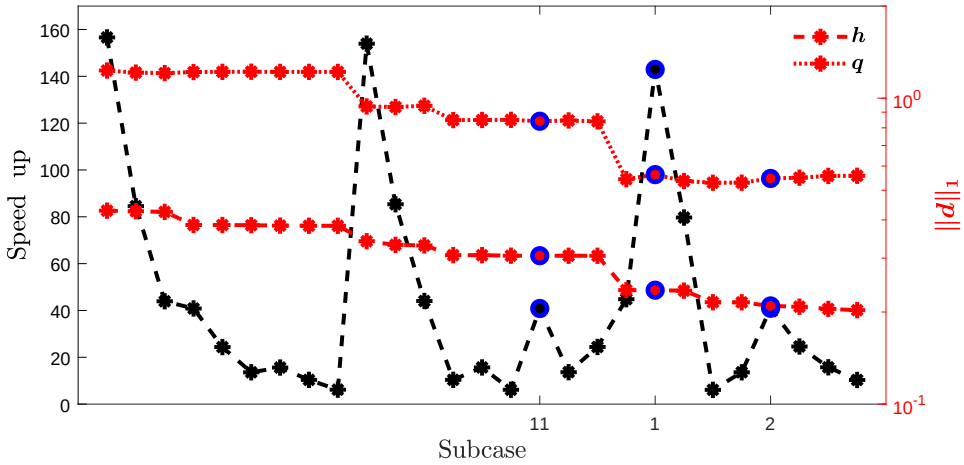
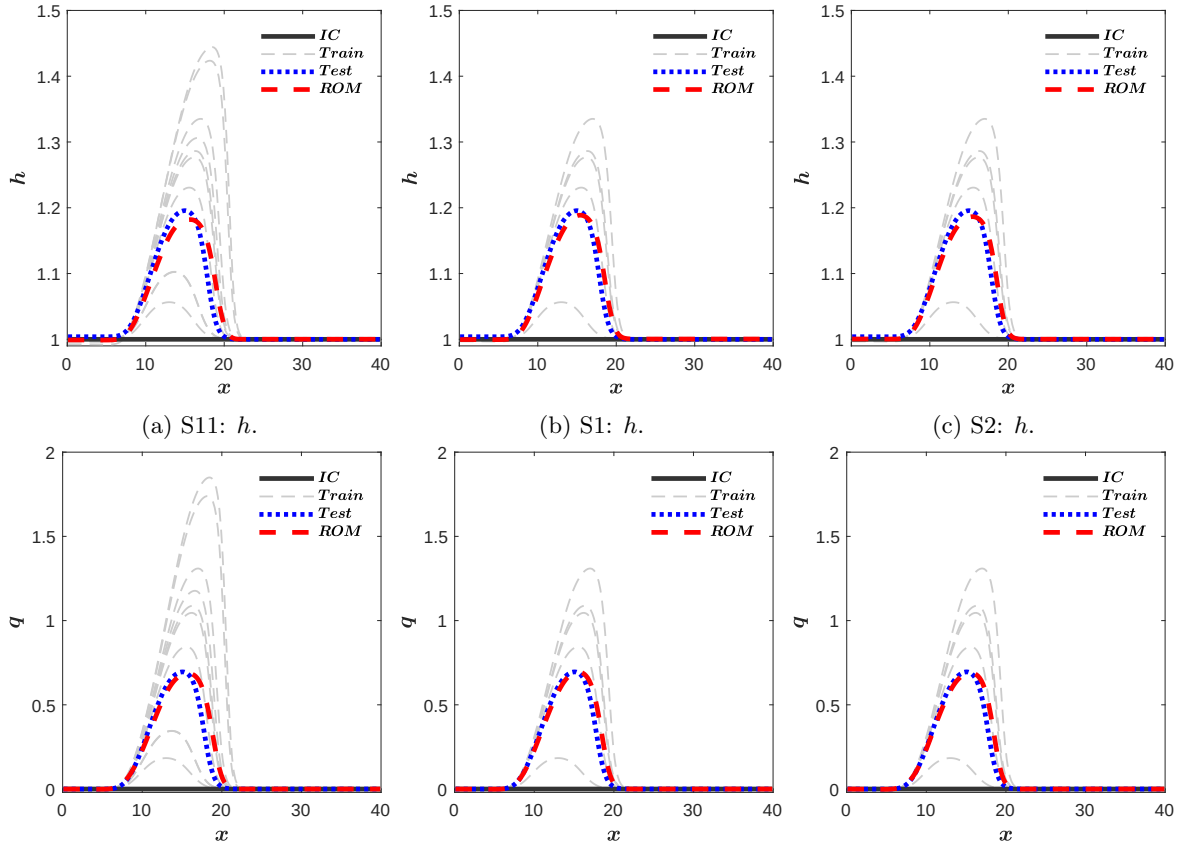



Figure 5.39: TC37. Candidate subcases.




(a) S11: $h$. (b) S1: $h$. (c) S2: $h$.

(d) S11: $q$. (e) S1: $q$. (f) S2: $q$.

Figure 5.40: TC37. ROM solutions of optimal candidates.

| Subcase | 11 | 1 | 2 |
|---|---|---|---|
| $M_{\text{POD}}$ | 15 | 5 | 15 |
| $M_W$ | 109 | 109 | 109 |
| $M_{\text{train}}$ | 10 | 5 | 5 |
| $\|d^h\|_1$ | $3.06 \cdot 10^{-1}$ | $2.36 \cdot 10^{-1}$ | $2.09 \cdot 10^{-1}$ |
| $\|d^q\|_1$ | $8.41 \cdot 10^{-1}$ | $5.62 \cdot 10^{-1}$ | $5.47 \cdot 10^{-1}$ |
| $\|d^h\|_\infty$ | $6.58 \cdot 10^{-2}$ | $4.80 \cdot 10^{-2}$ | $4.76 \cdot 10^{-2}$ |
| $\|d^q\|_\infty$ | $2.67 \cdot 10^{-1}$ | $1.82 \cdot 10^{-1}$ | $1.89 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $1.80 \cdot 10^{-1}$ | $1.80 \cdot 10^{-1}$ | $1.80 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $4.41 \cdot 10^{-3}$ | $1.26 \cdot 10^{-3}$ | $4.40 \cdot 10^{-3}$ |
| Speed-up | $\times 41$ | $\times 143$ | $\times 41$ |

Table 5.34: TC37. Efficiency results.

## 5.4 Beyond the training time (III)

The coordinate transformation methodology is applied to a linearized version of the SWE with the aim to predict solutions beyond the training time. Two test cases are shown to check this application to 1D and 2D problems.

### 5.4.1 1D linearized shallow water equations

The 1D linearized SWE are

$$\begin{aligned} \frac{\partial h}{\partial t} + h_0 \frac{\partial u}{\partial x} &= 0, \\ \frac{\partial u}{\partial t} + g \frac{\partial h}{\partial x} &= 0, \end{aligned} \tag{5.41}$$

where $h = h(x,t)$ is the water depth and $u = u(x,t)$ is the depth-averaged water velocity in the $x$-direction, $h_0$ is the undisturbed water depth at $t = 0$.

In order to approximate problem (5.41) in a new coordinate system using the mapping (3.25), it is necessary to decouple the system of equations (5.41). The procedure is explained below. First, problem (5.41) is written in vector form

$$\frac{\partial}{\partial t}\mathbf{U} + \mathbf{J}\frac{\partial}{\partial x}\mathbf{U} = 0, \tag{5.42}$$

where $\mathbf{U} = (h, u)^T$ is the conserved variables vector and

$$\mathbf{J} = \begin{pmatrix} 0 & h_0 \\ g & 0 \end{pmatrix}$$

is a diagonalizable Jacobian matrix with $\mathbf{J} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$ and

$$\mathbf{\Lambda} = \begin{pmatrix} c & 0 \\ 0 & -c \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 1 & 1 \\ c/h_0 & -c/h_0 \end{pmatrix}, \quad c = \sqrt{gh_0}.$$

Second, the conserved variables are decoupled by multiplying (5.42) by $\mathbf{P}^{-1}$. Then,

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{\Lambda} \frac{\partial \mathbf{W}}{\partial x} = 0, \tag{5.43}$$

where $\mathbf{W} = \mathbf{P}^{-1}\mathbf{U} = (\omega_1, \omega_2)^T$ are the characteristic variables

$$\omega_1 = (ch + h_0 u)/2c, \quad \omega_2 = (ch - h_0 u)/2c. \tag{5.44}$$

Finally, the mapping (3.25) is applied to problem (5.43) and the characteristic variables $\tilde{w}_i$ in the new coordinate system are separately approximated with a CTFOM and a CTROM.

**Test case 38. Prediction of an initial 1D Gaussian profile.**

In this case, the undisturbed water depth is given the following value $h_0 = 1$. The time-space domain is defined as $(x, t) \in [0, 2] \times [0, 0.15]$. The initial condition is given by

$$h(x, 0) = 1 + e^{-200(x-1)^2}, \quad u(x, 0) = 0, \quad 0 \leq x \leq 2, \tag{5.45}$$

Free boundary conditions are considered.

From (5.44), observe that the initial conditions and boundary conditions of the characteristic variables are

$$w_i(x, 0) = h/2,\ 0 \leq x \leq 2, \quad w_i(0, t) = w_i(2, t),\ 0 < t \leq T, \quad i = 1, 2.$$

The characteristic curves for each decoupled equation are given by

$$d_1(t) = d_1(0) + ct, \quad d_2(t) = d_2(0) - ct, \quad 0 < t \leq 0.15,$$

with $d_1(0) = d_2(0) = 1$. Their time evolution is shown in Figure 5.41. Note that the functions $w_i(x, 0)$ reach the maximum value at $d(0) = 1$.

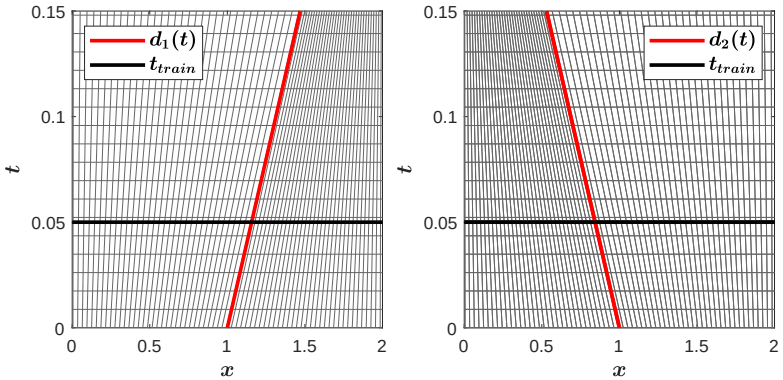



Figure 5.41: TC38: Time evolution of the physical meshes of reduced variables, $w_1$ and $w_2$.

The spatial domain is discretized by means of 200 volume cells. The time step is dynamically computed according to (2.23) with CFL = 0.9, and the training time is $T_{\text{train}} = 0.05$. The CTROM is solved using 14 POD modes. All these settings are contained in Table 5.35.

| $L$ | $T_{\text{train}}$ | $T$ | IC | BC | $h_0$ | CFL | $I_x$ | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.05 | 0.15 | Eq. (5.45) | Free | 1 | 0.9 | 200 | 19 | 53 | 14 |

Table 5.35: TC38. Problem settings.

Figure 5.42 shows the initial condition, the results of the CTFOM at $T_{\text{train}} = 0.05$ and the results of the CTROM at $T = 0.15$. A separately calculated CTFOM solution at $T = 0.15$ is also included for comparison with the CTROM solution. The CTROM is able to predict the position and the amplitude of the solution at the final time $T = 0.15$.

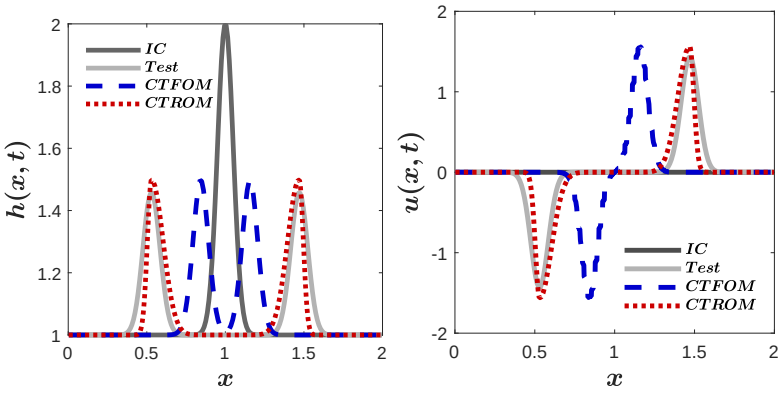



Figure 5.42: TC38: Solutions of physical variables, $h$ and $u$.

The CPU times required are contained in Table 5.36, where $I_x = 100$, 200, 500, 1000, 2000 and 3000. The CPU times of the CTROM at $T = 0.15$ are lower than those of CT-FOM at the training time $T_{\text{train}} = 0.05$ and therefore even lower than those at the final time.

| $I_x$ | 100 | 200 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| $\tau_{\text{CPU}}^{\text{train}}$ | $3.01 \cdot 10^{-3}$ | $4.07 \cdot 10^{-3}$ | $1.03 \cdot 10^{-2}$ | $2.96 \cdot 10^{-2}$ | $1.04 \cdot 10^{-1}$ | $2.26 \cdot 10^{-1}$ |
| $\tau_{\text{CPU}}^{\text{test}}$ | $4.57 \cdot 10^{-3}$ | $9.80 \cdot 10^{-3}$ | $4.23 \cdot 10^{-2}$ | $1.64 \cdot 10^{-1}$ | $6.18 \cdot 10^{-1}$ | 1.38 |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $9.18 \cdot 10^{-4}$ | $2.37 \cdot 10^{-3}$ | $9.12 \cdot 10^{-3}$ | $2.61 \cdot 10^{-2}$ | $8.11 \cdot 10^{-2}$ | $1.38 \cdot 10^{-1}$ |
| Speed-up (train) | $\times 3$ | $\times 2$ | $\times 1$ | $\times 1$ | $\times 1$ | $\times 2$ |
| Speed-up (test) | $\times 5$ | $\times 4$ | $\times 5$ | $\times 6$ | $\times 8$ | $\times 10$ |

Table 5.36: TC38. Efficiency results for all subcases.

### 5.4.2 2D linearized shallow water equations

Consider now the 2D version of the linearized SWE system (5.41) defined on the domain $(x, y, t) \in [0, L_x] \times [0, L_y] \times [0, T]$

$$\begin{aligned}
&\frac{\partial h}{\partial t} + h_0 \frac{\partial u}{\partial x} + h_0 \frac{\partial v}{\partial y} = 0,\\
&\frac{\partial u}{\partial t} + g \frac{\partial h}{\partial x} + g \frac{\partial h}{\partial y} = 0,\\
&\frac{\partial v}{\partial t} + g \frac{\partial h}{\partial x} + g \frac{\partial h}{\partial y} = 0,
\end{aligned} \tag{5.46}$$

where $h = h(x, y, t)$ is the water depth and $u = u(x, y, t)$ and $v = v(x, y, t)$ are the depth-averaged water velocities in the $x$- and $y$-direction, $h_0$ is the undisturbed water depth at $t = 0$ and $g$ is the gravitational acceleration. Following the Radon transform (3.37), problem (5.46) is transformed into a 1D system of equations.

**Test case 39. Prediction of an initial 2D Gaussian profile**

The time-space domain is defined as $(x, y, t) \in [0, 10] \times [0, 10] \times [0, 5]$ and the spatial domain is uniformly divided into $201 \times 201$ volume cells. The ICs are defined as a dam-break with Gaussian profile

$$\begin{aligned}
&h(x, y, 0) = e^{-\frac{(x)^2+(y)^2}{2}},\\
&u(x, y, 0) = 0,\\
&v(x, y, 0) = 0;
\end{aligned} \tag{5.47}$$

and the boundary conditions are

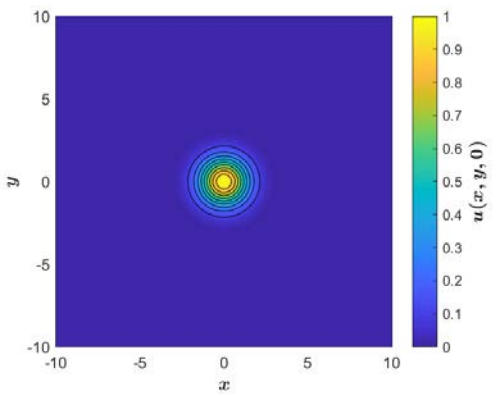
$$\begin{aligned}
&h(0, y, t) = h(10, y, t), \quad h(x, 0, t) = h(x, 10, t),\\
&u(0, y, t) = u(10, y, t), \quad u(x, 0, t) = u(x, 10, t),\\
&v(0, y, t) = v(10, y, t), \quad v(x, 0, t) = v(x, 10, t).
\end{aligned} \tag{5.48}$$

The CFL number considered in this case is 0.9 and the time step is computed to satisfy the stability condition (3.28) in the transformed coordinates. Solutions are computed with the CTFOM up to $T_{\text{train}} = 0.5$ and, with these data, new solutions are computed using the CTROM up to $T = 5$. In this case, the number of POD modes solved with the CTROM is $M_{\text{POD}} = 14$. All these settings are contained in Table 5.37.

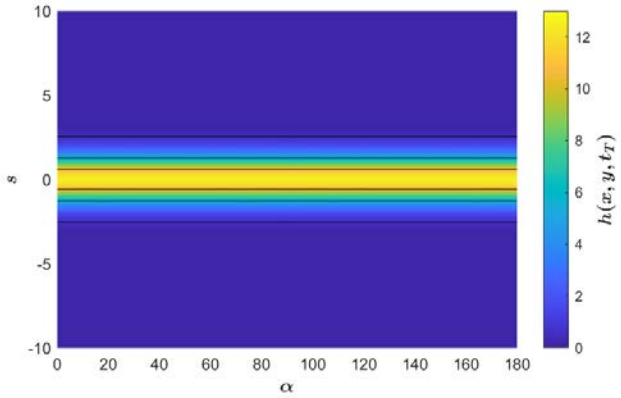
| $L_x \times L_y$ | $T_{\text{train}}$ | $T$ | IC | BC | $h_0$ | CFL | $I_x$ | $N_{T_{\text{train}}}$ | $N_T$ | $M_{\text{POD}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $10 \times 10$ | 0.5 | 5 | (5.47) | (5.48) | 1 | 0.9 | $201 \times 201$ | 5 | 32 | 14 |

Table 5.37: TC39. Problem settings.

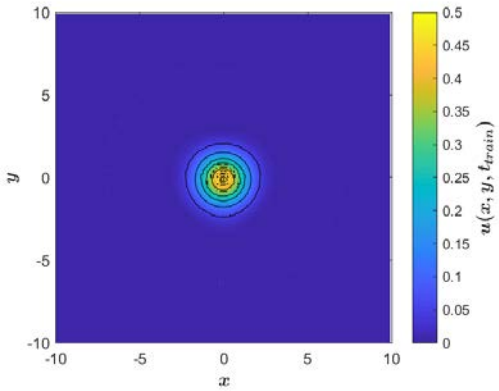
Figure 5.43 shows the solutions of the CTFOM at $t = 0$ and $T_{\text{train}} = 0.5$ and the solution of the CTROM at $T = 5$, together with the their respective sinograms. Note that the scales of all the figures are different, so the maximum values reached are also different, decreasing as time advances. Figure 5.44 shows the pointwise differences between the CTFOM and the CTROM solutions, whose maximum value is an order of magnitude smaller than the amplitude of the solution at $T = 5$.



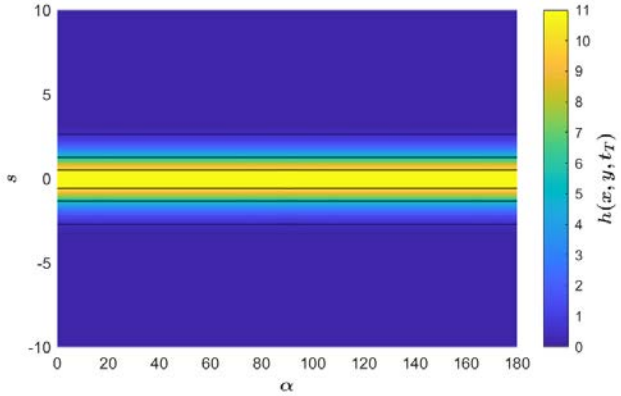

(a) Solution at $t = 0$.



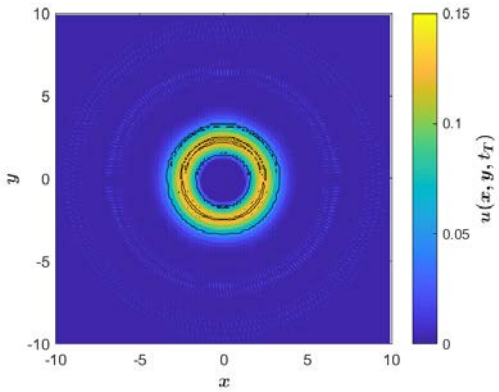

(b) Sinogram at $t = 0$.



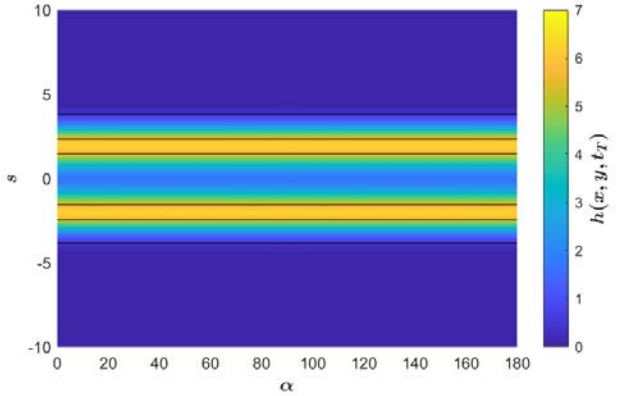

(c) Solution of CTFOM at $T_{\text{train}} = 0.5$.




(d) Sinogram of CTFOM at $t_{train} = 0.5$.




(e) Solution of CTROM at $T = 5$.




(f) Sinogram of CTROM at $T = 5$.

Figure 5.43: TC39: Solution computed with the CTROM and its comparison with the CTFOM.

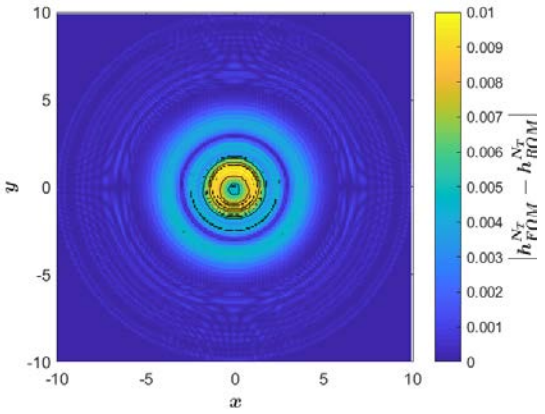



Figure 5.44: TC39: CTFOM vs. CTROM.

These results evidence that, although the CTROM strategy herein proposed is a genuine method for 1D time dependent problems, it can be extended to higher spatial dimensions using the Radon transform.

## 5.5 Final numerical cases

In this last section, the ROM methodology will be applied to 2D problems of high dimension and complexity, in which all the corrections mentioned above, as well as Dirichlet boundary conditions, are applied.

### 5.5.1 Test case 40. 1D/2D dam-break over a triangular obstacle

The last case is focused on studying the performance of the ARoe-based ROM on a realistic problem for which there are experimental data. This case serves to test the relationship of the ARoe-based ROM with the numerical corrections detailed in Section 5.2. All these corrections need to be used in this problem. In order to do so, the ARoe-based ROM will be trained with results computed by the ARoe-based FOM in which these corrections have been taken into account.

The problem used for this purpose has been proposed in [24] and [125, Section 8.2]. Experimental data was obtained from the Recherches Hydrauliques Lab. Châtelet together with the University of Bruxelles (Belgium) under the supervision of J.M. Hiver. The test case deals with the evolution of a dam-break wave over a triangular obstacle. The channel geometry, shown in Figure 5.45, is given by

$$z(x) = \begin{cases} 0.4 + \frac{0.4}{3}\left(x - 28.5\right), & \text{if } 25.5 \leq x < 28.5, \\ -\frac{0.4}{3}\left(x - 31.5\right), & \text{if } 28.5 \leq x < 31.5, \\ 0, & \text{otherwise.} \end{cases} \tag{5.49}$$

Glauker-Manning formula is used to compute friction losses with $n_b = 0.0125$. The time-space domain is defined as $(x,t) \in [0,38] \times [0,40]$. The ICs are defined as a water reservoir

$$h(x,0) = \begin{cases} 0.75, & \text{if } 0 \leq x < 15.5, \\ 0, & \text{if } 15.5 \leq x \leq 38, \end{cases} \quad q(x,0) = 0, \quad 0 \leq x \leq 38, \tag{5.50}$$

and solid wall and free outlet BCs are considered upstream and downstream, respectively.

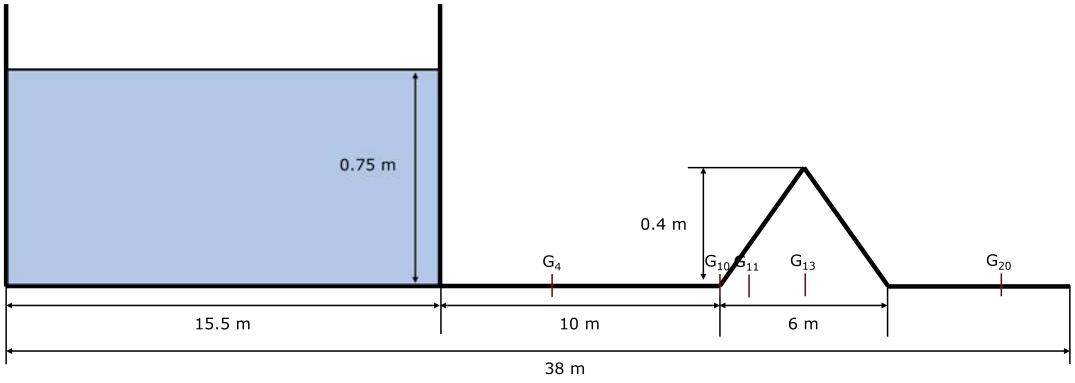



Figure 5.45: TC40: Geometry of the channel.

Experimental water depth data were measured at five gauging points along the channel. The positions of the gauging points $G_4$, $G_{10}$, $G_{11}$, $G_{13}$ and $G_{20}$ are shown in Figure 5.45 and contained in Table 5.38.

| Point | $G_4$ | $G_{10}$ | $G_{11}$ | $G_{13}$ | $G_{20}$ |
|---|---|---|---|---|---|
| Position | 19.5 | 25.5 | 26.5 | 28.5 | 35.5 |

Table 5.38: TC40: Positions of the gauging points.

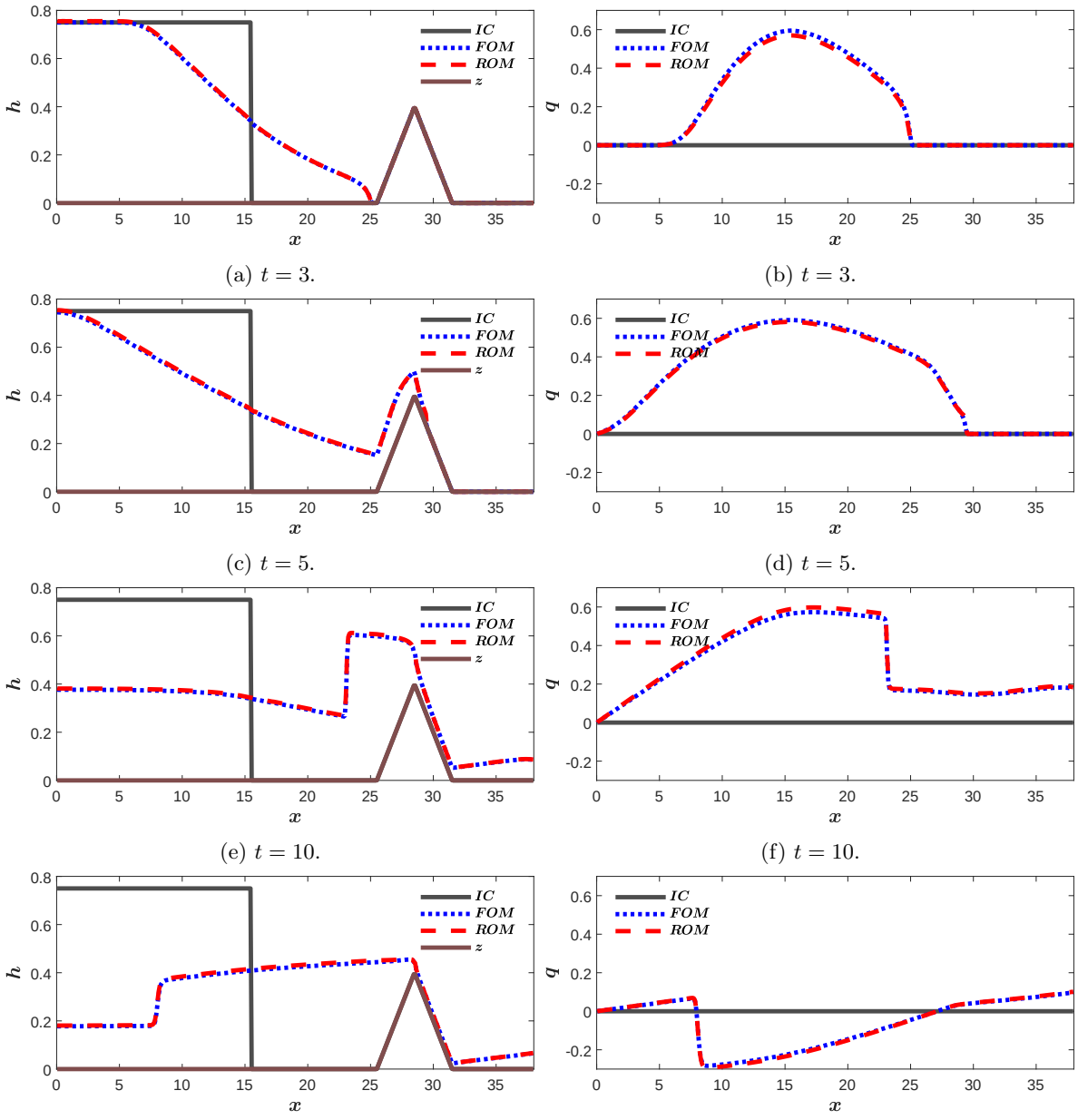
The ARoe-based FOM (2.25) have been solved using 400 cells and CFL = 0.1; and the ARoe-based ROM (5.4) has been solved using 5 POD modes and 3283 time windows. All these settings are contained in Table 5.39.

| $L$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_x$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 38 | 40 | Eq. (5.50) | Free | Fig. 5.45 | 0.0125 | 0.1 | 400 | 13132 | 5 | 3283 |

Table 5.39: TC40. Problem settings.

Figure 5.46 shows the results of $h$ and $q$ computed by the ARoe-based FOM and ROM along the channel at different times, namely $t = 3$, 5, 10 and 20.

Figure 5.47 shows the time evolution of $h$ (left) and $q$ (right) at the positions of the five gauging points $G_4$, $G_{10}$, $G_{11}$, $G_{13}$ and $G_{20}$.



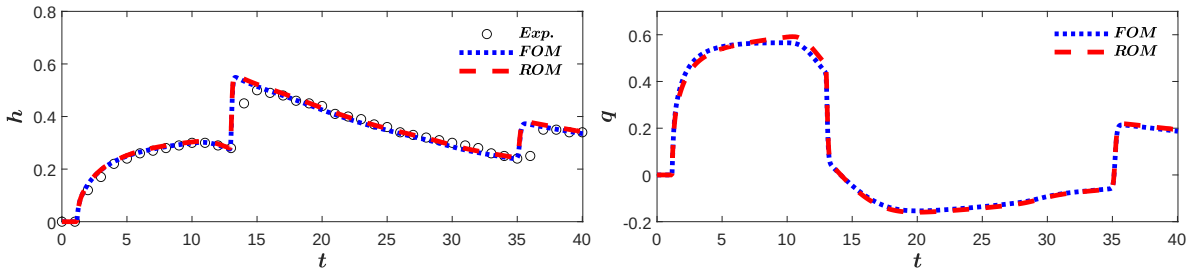

(a) $t = 3$. (b) $t = 3$.

(c) $t = 5$. (d) $t = 5$.

(e) $t = 10$. (f) $t = 10$.

(g) $t = 20$. (h) $t = 20$.

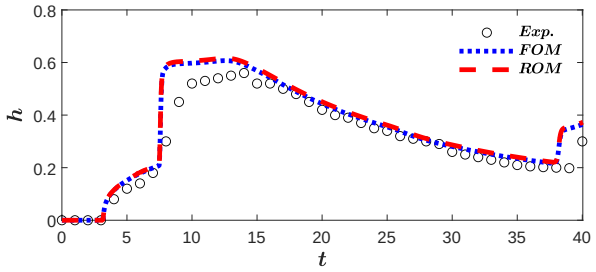
Figure 5.46: TC40: Results of $h$ (left) and $q$ (right) along the channel at different times.



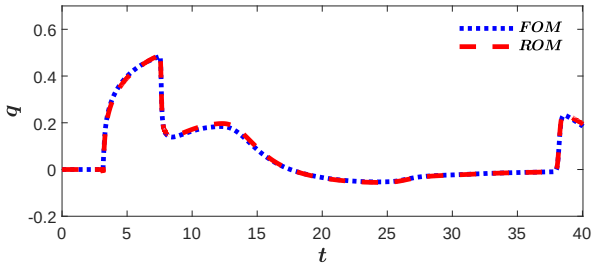

(a) G4 ($x = 19.5$).



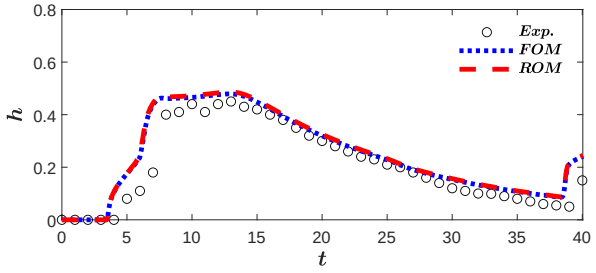

(b) G4 ($x = 19.5$).



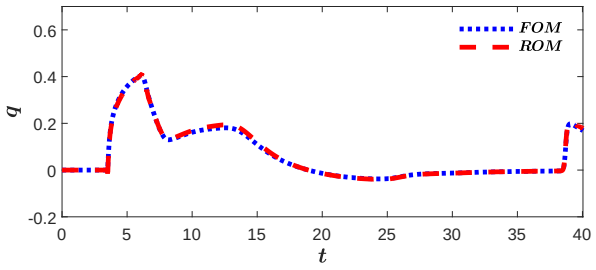

(c) G10 ($x = 25.5$).



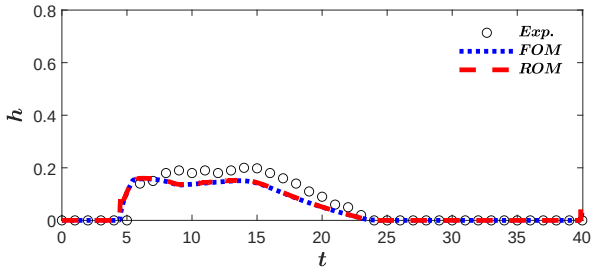

(d) G10 ($x = 25.5$).



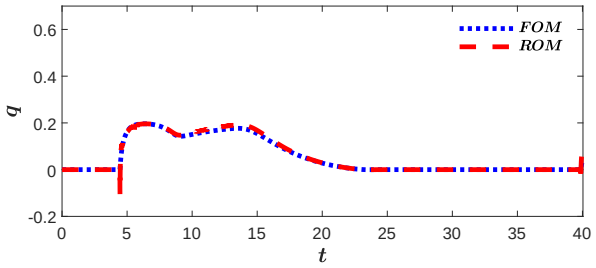

(e) G11 ($x = 26.5$).



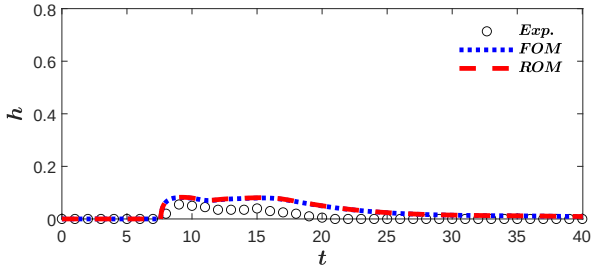

(f) G11 ($x = 26.5$).



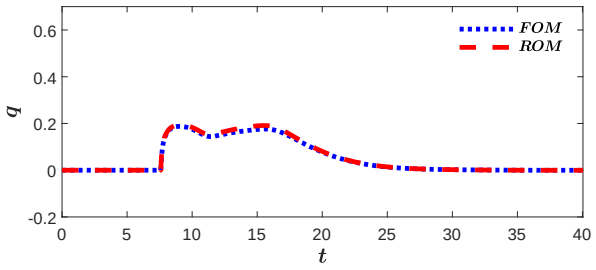

(g) G13 ($x = 28.5$).




(h) G13 ($x = 28.5$).




(i) G20 ($x = 32.5$).




(j) G20 ($x = 32.5$).

Figure 5.47: TC40: Results of $h$ (compared with the experimental data, left) and $q$ (right) in the different gauging points.

Table 5.40 contains the CPU times required by the FOM and the ROM to obtain their results for different mesh refinements. It also shows that the speed-ups of the ROM with respect to the FOM achieve one order of magnitude. These results can also be seen in Figure 5.48. The differences between the FOM and the ROM solutions can be seen in Figure 5.49 for the different mesh refinements.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $I_x$ | 100 | 200 | 400 | 600 |
| $N_T$ | 3212 | 6512 | 13132 | 19748 |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $1.35 \cdot 10^{-1}$ | $4.53 \cdot 10^{-1}$ | 1.54 | 3.37 |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $6.05 \cdot 10^{-3}$ | $2.65 \cdot 10^{-2}$ | $8.71 \cdot 10^{-2}$ | $1.92 \cdot 10^{-1}$ |
| Speed-up | $\times 22$ | $\times 17$ | $\times 18$ | $\times 18$ |

Table 5.40: TC40: CPU times required by the ARoe-based FOM and the ARoe-based ROM for different mesh refinements and the corresponding CPU times per time step and the corresponding speed-ups.

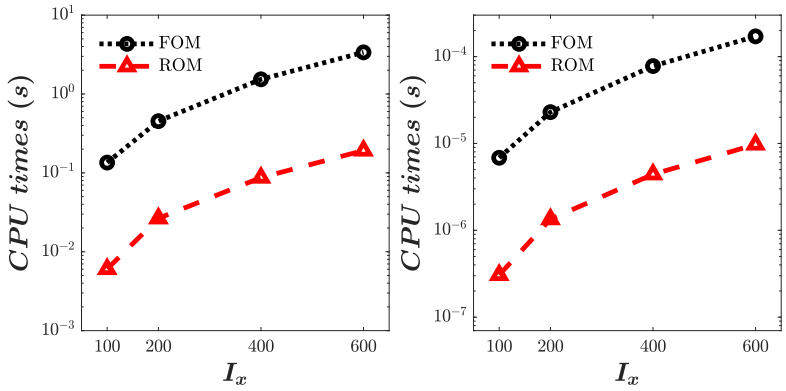



Figure 5.48: TC40: CPU times (left) and CPU times by time step (right) required by the FOM and the ROM.

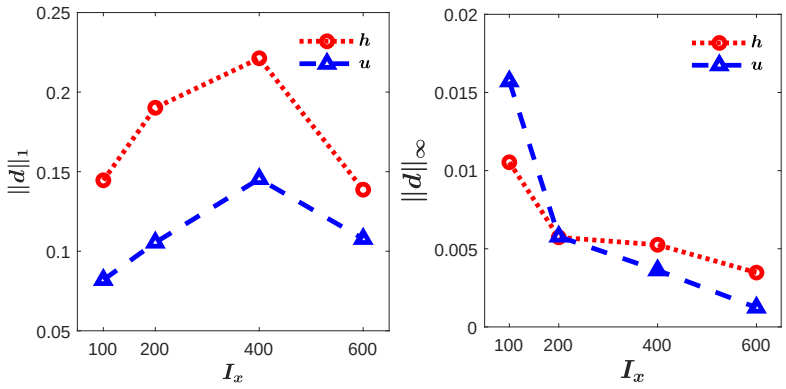



Figure 5.49: TC40: Differences between FOM and ROM solutions using the $\|d\|_1$ and $\|d\|_\infty$ norms.

Finally, an extension of this problem to the 2D case is carried out following [127] to show more clearly the computational advantages of ROMs. The time-space domain is defined as $(x, y, t) \in [0, 38] \times [0, 2] \times [0, 40]$. The IC of the problem is similar to the 1D version

$$\begin{aligned} h(x, y, 0) &= \begin{cases} 0.75, & \text{if } 0 \leq x < 15.5,\ 0 \leq y \leq 2, \\ 0, & \text{if } 15.5 \leq x \leq 38,\ 0 \leq y \leq 2, \end{cases} \\ q_x(x, y, 0) &= q_y(x, y, 0) = 0,\ \forall x, y, \end{aligned} \tag{5.51}$$

where $q_x$ and $q_y$ are the water discharges in the $x$- and $y$-direction, respectively, and solid wall and free outlet boundary conditions are considered upstream and downstream, respectively. The 2D domain is discretized using 1600 rectangular cells, as shown in Figure 5.50.

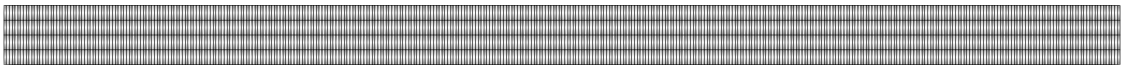


Figure 5.50: TC40: 2D mesh.

The ARoe-based FOM (2.40) have been solved with CFL = 0.5; and the ARoe-based ROM (5.4) has used 5 POD modes and 822 time windows. The settings are contained in Table 5.41.

| $L_x \times L_y$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 38×2 | 40 | Eq. (5.51) | Free | Fig. 5.45 | 0.0125 | 0.1 | 1600 | 1645 | 5 | 822 |

Table 5.41: TC40. Problem settings.

Figure 5.51 shows the results of $h$ computed with the ARoe-based ROM along the channel at different time.

(a) $t = 0$.

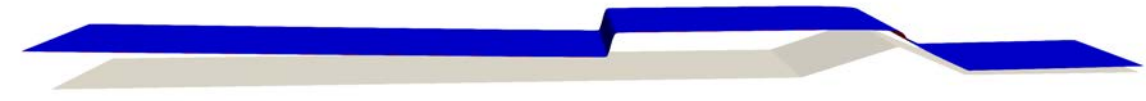
(b) $t = 1.75$.

(c) $t = 3.60$.

(d) $t = 4.55$.

(e) $t = 6.47$.

(f) $t = 6.47$.



(g) $t = 19.37$.

Figure 5.51: TC40: 2D SWE results of $h$ along the channel at different times computed with the ARoe-based FOM in blue and the ARoe-based ROM in red.

Table 5.42 contains the CPU times required by the ARoe-based FOM and the ARoe-based ROM to obtain their results for different mesh refinements for the 2D case. It also shows the corresponding speed-ups of the ARoe-based ROM with respect to the FOM. The speed-ups of one order of magnitude are faster than those of the 1D case.

| Subcase | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $I_c$ | 400 | 800 | 1600 | 2400 |
| $N_T$ | 411 | 821 | 1645 | 2469 |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $5.91 \cdot 10^{-1}$ | 2.35 | 9.26 | $2.08 \cdot 10^{1}$ |
| $\tau_{\text{CPU}}^{\text{ROM}}$ | $5.00 \cdot 10^{-3}$ | $2.00 \cdot 10^{-2}$ | $7.90 \cdot 10^{-2}$ | $1.75 \cdot 10^{-1}$ |
| Speed-up | $\times 118$ | $\times 117$ | $\times 117$ | $\times 119$ |

Table 5.42: TC40: CPU times required by the 2D ARoe-based FOM and ROM for different mesh refinements and the corresponding speed-ups.

Figure 5.52 shows the CPU times required by the 1D/2D FOMs and ROMs. The differences between FOM and ROM solutions using the $\|d\|_1$ and $\|d\|_\infty$ norms are shown in Figure 5.53.

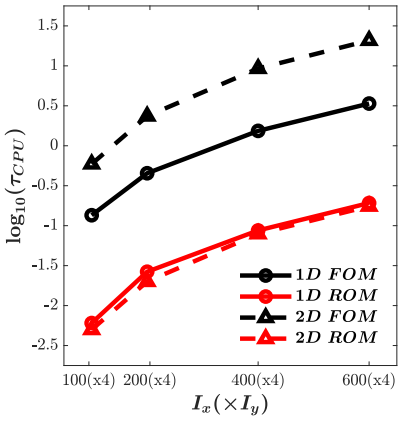



Figure 5.52: TC40: CPU times required by FOMs and ROMs in both 1D and 2D versions.

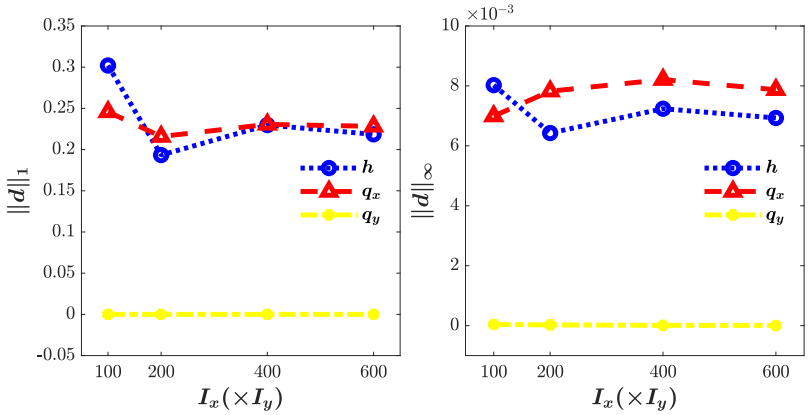



Figure 5.53: TC40: Differences between FOM and ROM solutions using the $\|d\|_1$ and $\|d\|_\infty$ norms.

### 5.5.2 Test case 41. 2D water depth source.

This test case is designed to prove that ROMs are very useful to reproduce solutions with injections at inner-domain points, such as point sources of water depth. The time-space domain of the case is defined as $(x, y, t) \in [0, 6] \times [0, 6] \times [0, 3]$. Initially, all the domain is dry

$$h(x, y, 0) = 0.0, \tag{5.52}$$

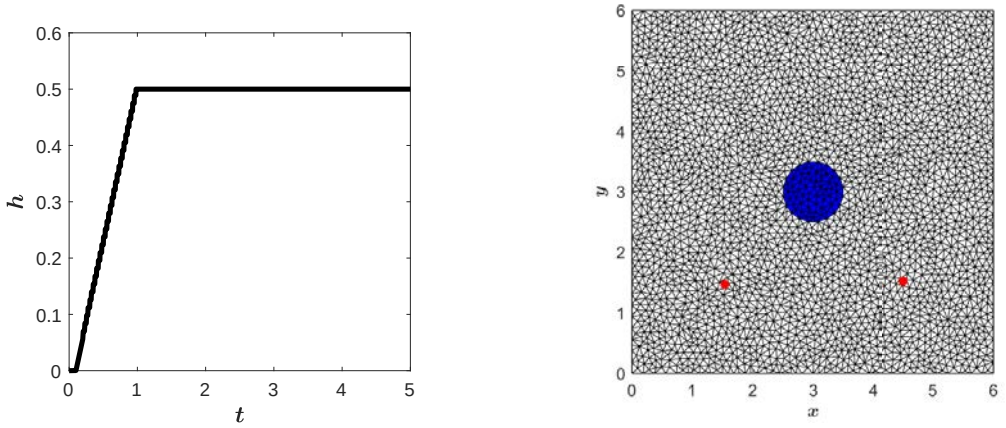
with $0 \leq x \leq 6$, $0 \leq y \leq 6$; and no bed elevation

$$z(x, y) = 0, \tag{5.53}$$

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

$$h(x, y) = \begin{cases} 0.5 - 0.56(1 - t), & \text{if } 0.1 \leq t < 1 \text{ and } (x, y) \in \Omega^J, \\ 0.5, & \text{if } 1 \geq t \text{ and } (x, y) \in \Omega^J. \end{cases} \tag{5.54}$$

with $\Omega^J = \{(x, y),\ (x-3)^2 + (y-3)^2 \leq 4\}$, as shown in Figure 5.54a. Closed walls are considered in all boundaries and the Manning coefficient is $n_b = 0.01$.




(a) Injection $h(x, y, t)$ with $(x, y) \in \Omega^J$.

(b) 2D numerical mesh.

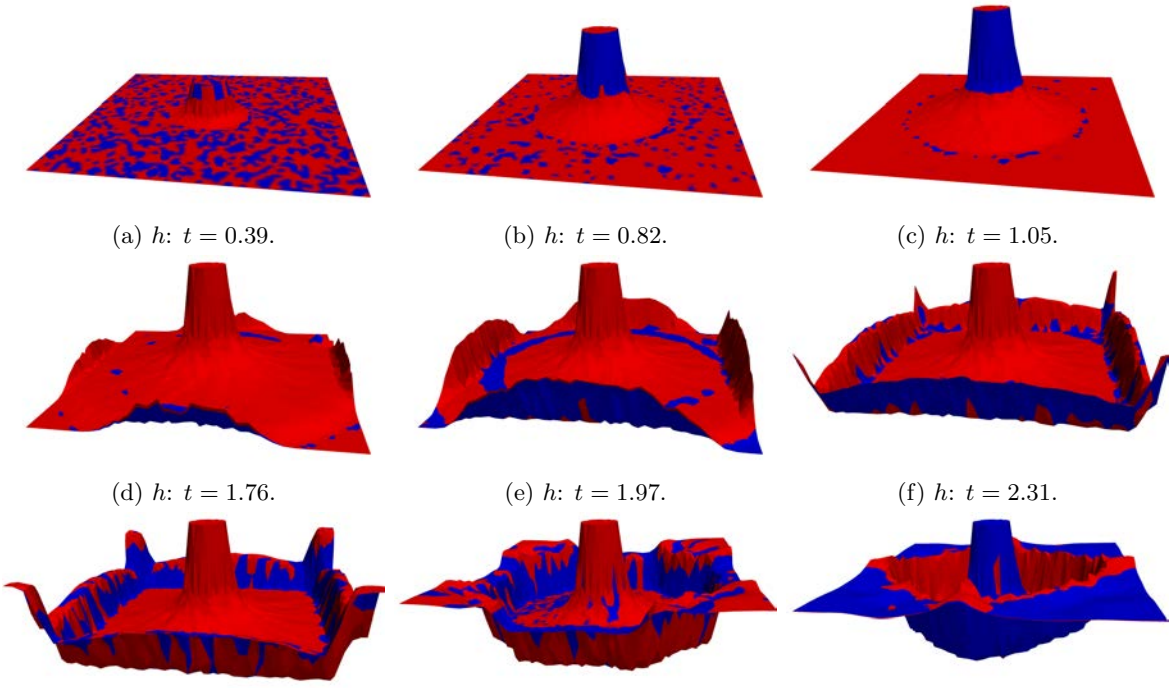
Figure 5.54: TC42. Water depth injected in the center of the domain (left) and 2D numerical mesh (right) with injection volume cells indicated in blue. Red dots represent the position of the probes.

The spatial domain is discretized using $I_c = 5603$ unstructured elements, as shown in Fig. 5.54b, where the injection points in which the water depth is imposed are plotted in blue.

| $L_x \times L_y$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $6 \times 6$ | 5 | (5.52) | (5.54) | (5.53) | 0.01 | 0.5 | 5603 | 3280 | 5 | 66 |

Table 5.43: TC42. Problem settings.

The numerical results obtained with the ROM can be seen in Figure 5.55 for different times. The solutions computed with the FOM are represented in blue and the ones of the ROM, in red. The quasi-stationary regime at the beginning of the simulation is solved by the ARoe-based ROM (5.4) with a single very wide time window and then 66 time windows with a fixed number of 50 snapshots per window. There are no significant differences between its solutions and those of the FOM. Furthermore, the differences between the solutions of the FOM and the ROM have been computed to test the accuracy of the latter. Figure 5.56 shows the time evolution of the differences, alongside the limits of time windows depicted by the vertical gray lines.



(a) $h$: $t = 0.39$. (b) $h$: $t = 0.82$. (c) $h$: $t = 1.05$.

(d) $h$: $t = 1.76$. (e) $h$: $t = 1.97$. (f) $h$: $t = 2.31$.

(g) $h$: $t = 2.72$. (h) $h$: $t = 3.88$. (i) $h$: $t = 4.83$.

Figure 5.55: TC42. ROM solutions of $h$ at different time steps.

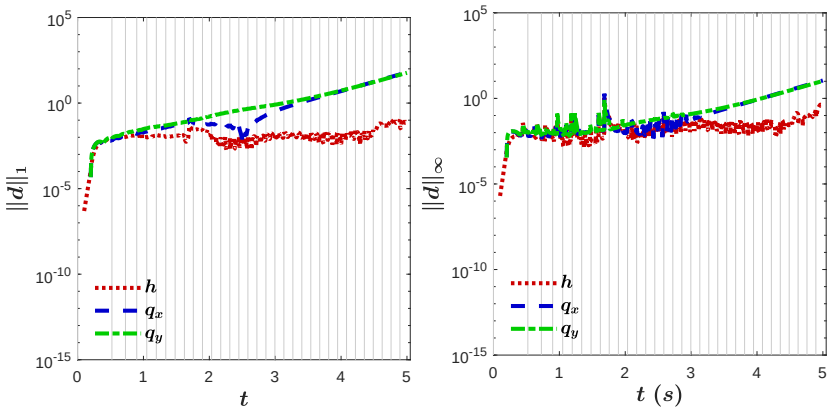



Figure 5.56: TC42. Differences between FOM and ROM solutions.

Two probes have been set

$$P_1 = (1.5,\ 1.5),$$
$$P_2 = (4.5,\ 1.5),$$

to study how the ROM performs the symmetry of the problem. The position of these probes is indicated with red dots within the spatial domain in Figure 5.54b. Figure 5.57 shows strong agreement between the solutions of the FOM and the ROM in both probes.

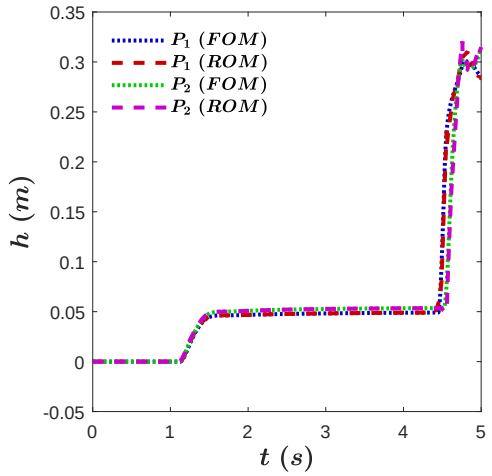



Figure 5.57: TC42. Time evolution of water depth in probes $P_1$ and $P_2$.

Finally, the CPU times required by the FOM and the ROM are shown in the Table 5.44, as well as the speed-up achieved in this case, which reaches 2 orders of magnitude. That table also shows the differences computed at the final time.

| $\|d^h\|_1$ | $\|d^{q_x}\|_1$ | $\|d^{q_y}\|_1$ |
|---|---|---|
| $7.45 \cdot 10^{-2}$ | $6.18 \cdot 10^{1}$ | $5.87 \cdot 10^{1}$ |
| $\|d^h\|_\infty$ | $\|d^{q_x}\|_\infty$ | $\|d^{q_y}\|_\infty$ |
| $3.40 \cdot 10^{-1}$ | $1.11 \cdot 10^{1}$ | $1.05 \cdot 10^{1}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
| $1.71 \cdot 10^{1}$ | $3.03 \cdot 10^{-2}$ | $\times 563$ |

Table 5.44: TC42. Differences between FOM and ROM solutions and CPU times required.

### 5.5.3 Test case 43. 2D tsunami test case.

In this final case, the method presented in this work is tested in a 1/400 scale laboratory tsunami test case done by [106], and included in many other works [27, 52, 123]. The time-space domain of the case is defined as $(x, y, t) \in [0, 5.488] \times [0, 3.388] \times [0, 22.5]$.

The bed elevation of the case is shown in Figure 5.59. The boundary conditions are defined as walls in the north, south and east boundaries and as the water depth hydrograph shown in

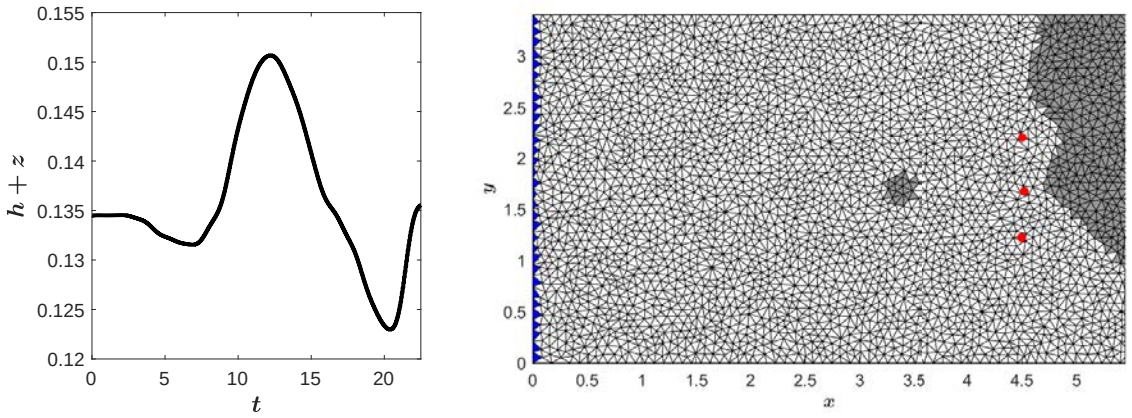
Figure 5.58a. The ICs are set as water at rest

$$h(x, y, 0) + z(x, y) = 0,\ 0 \leq x \leq 5.488,\ 0 \leq y \leq 3.388, \tag{5.55}$$

as shown in Figure 5.60, and the Manning coefficient is $n_b = 0.01$.

The spatial domain is discretized using an unstructured mesh that consists of $I_c = 5863$ volume cells, as shown in Figure 5.58b, where the boundary points in which the water depth is imposed are plotted in blue. The volume cells represented in grey are those in which the bed level is positive, i.e., $z > 0$ m; and the red dots indicate the position of the three gauging points where the experimental data were measured, given by these coordinates

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




(a) Injection $h(x', y', t)$.

(b) 2D unstructured numerical mesh.

Figure 5.58: TC43. Water depth injected in the western boundary (left) and 2D numerical mesh (right) with injection volume cells indicated in blue. Red dots represent the position of the probes.

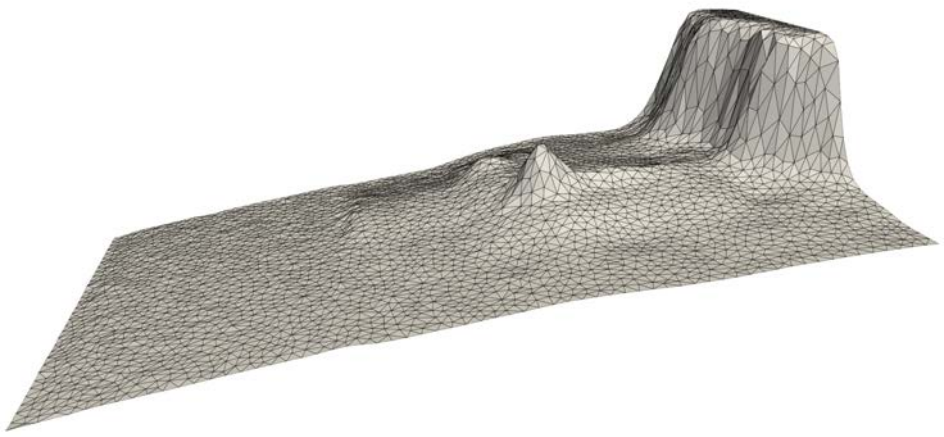


Figure 5.59: TC43. Bed elevation with numerical mesh.

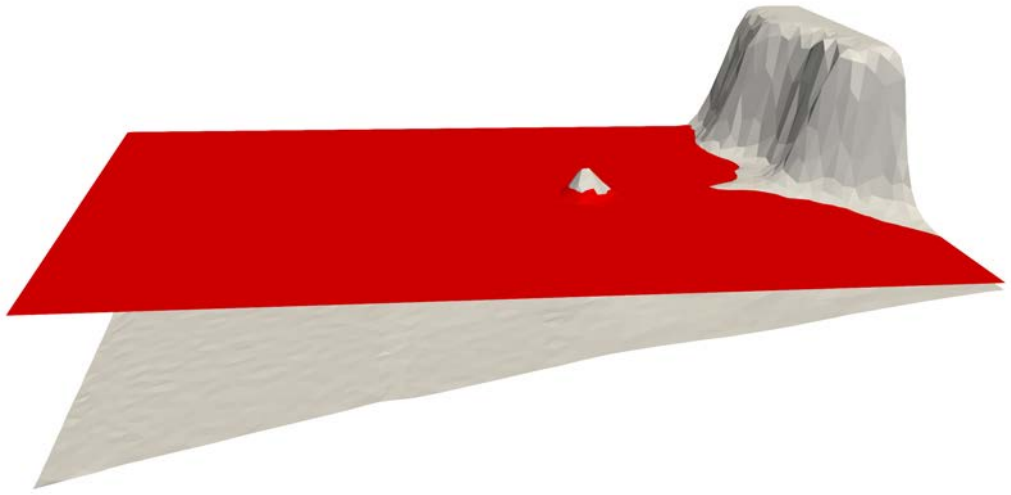


Figure 5.60: TC43. Initial condition.

The ARoe-based ROM has been solved using 3 POD modes and 70 time windows, as shown in Table 5.45 with the rest of the settings of the problem.

| $L_x \times L_y$ | $T$ | IC | BC | $z$ | $n_b$ | CFL | $I_c$ | $N_T$ | $M_{\text{POD}}$ | $M_W$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $5.488 \times 3.388$ | 22.5 | (5.55) | 5.58a | 5.59 | 0.01 | 0.5 | 5863 | 5447 | 3 | 70 |

Table 5.45: TC43. POD setting, ROM CPU time and speed-ups achieved of each subcase. *Non-homogeneous time windows.

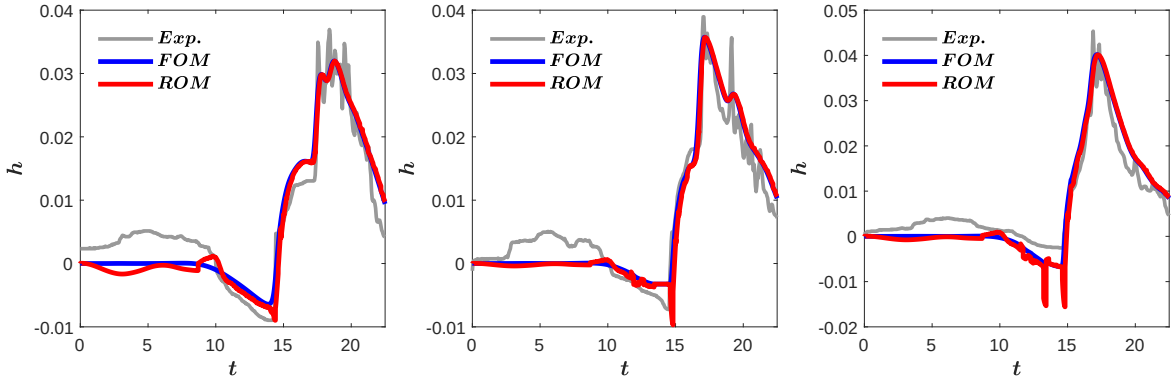
The time evolution of the numerical solutions computed by the ARoe-based ROM (5.4) for the water depth is shown in Figure 5.63, represented red, and the ARoe-bsaed FOM (2.40) solution in blue. From time instant $t = 10.73$ s (Fig. 5.63b) to $t = 14.98$ s (Fig. 5.63e), the water wave enters the left domain and advances towards the coast. At $t = 15.83$ s (Fig. 5.63f) it can be seen that the water rises along the dry coastal land. And then, once the wave has bounced, it returns to the inner sea (Fig. 5.63j). In all these figures it can be seen how closely the FOM and ROM solutions agree.

Figure 5.61 shows the good agreement between the FOM and the ROM solutions and the experimental data measured in the three gauging points. Mass is conserved as shown in Figure 5.62a, where some discrepancies between the solutions of the FOM and the ROM are observed. Using a non-maximum number of POD modes when solving the ROM can imply losses or gains in mass that have no physical significance. However, when comparing both numerical solutions with experimental data, it can be seen that the general trend is well achieved.

The time windows have been homogeneously defined starting from the instant at which the hydrograph of the water depth varies. Until then, as can be seen in Figures 5.62b and 5.62c, a single very wide window is defined that covers all the time in which the state of the problem does not change significantly. The error, as can be seen here, is kept at the usual when a

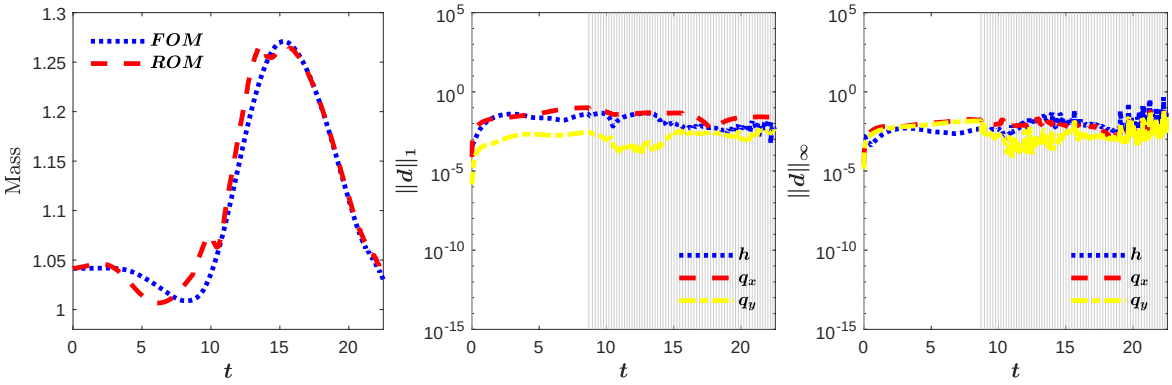
non-maximum number of POD modes is used.

Finally, as indicated in Table 5.46, the ARoe-based ROM is 819 times faster than the ARoe-based FOM; and its errors at the final time are acceptable.




(a) Probe 1. (b) Probe 2. (c) Probe 3.

Figure 5.61: TC43. Time evolution of water depth at 3 different probes.

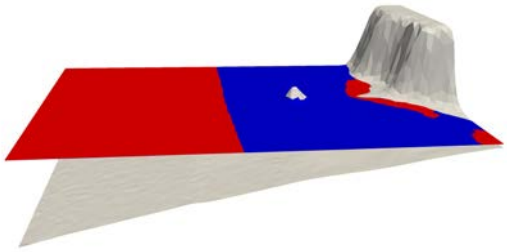



(a) TC43. Mass conservation. (b) $\|d\|_1$. (c) $\|d\|_\infty$.

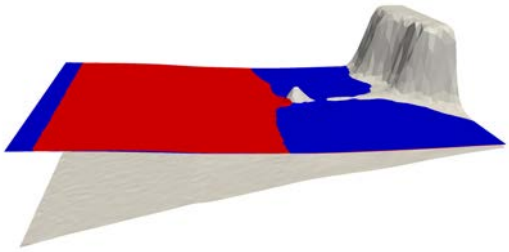
Figure 5.62: TC43. Time evolution of mass conservation and solution differences.

| $\|d^h\|_1$ | $\|d^{q_x}\|_1$ | $\|d^{q_y}\|_1$ |
|---|---|---|
| $5.77 \cdot 10^{-3}$ | $2.68 \cdot 10^{-2}$ | $4.77 \cdot 10^{-3}$ |
| $\|d^h\|_\infty$ | $\|d^{q_x}\|_\infty$ | $\|d^{q_y}\|_\infty$ |
| $3.38 \cdot 10^{-2}$ | $5.51 \cdot 10^{-3}$ | $1.38 \cdot 10^{-2}$ |
| $\tau_{\text{CPU}}^{\text{FOM}}$ | $\tau_{\text{CPU}}^{\text{ROM}}$ | Speed-up |
| $1.46 \cdot 10^{1}$ | $1.78 \cdot 10^{-2}$ | $\times 819$ |

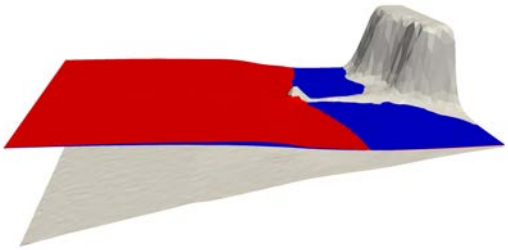
Table 5.46: TC43. Differences between FOM and ROM solutions and CPU times required.



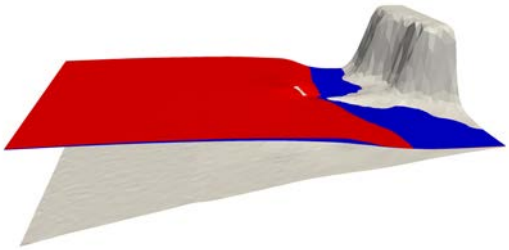
(a) S4. $t = 0.44$.



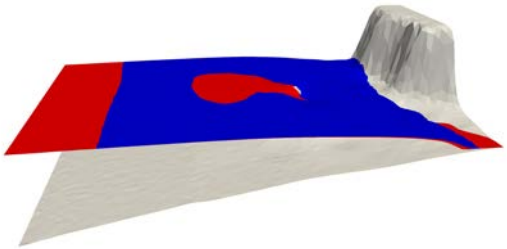
(b) S4. $t = 10.73$.



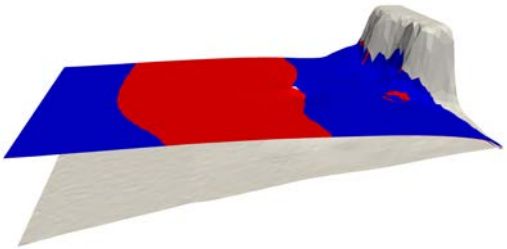
(c) S4. $t = 12.24$.



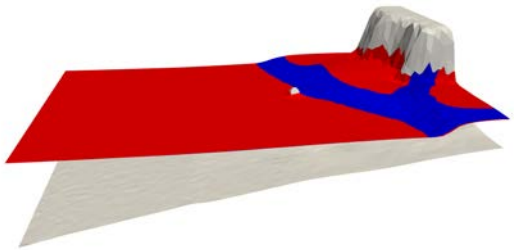
(d) S4. $t = 13.38$.



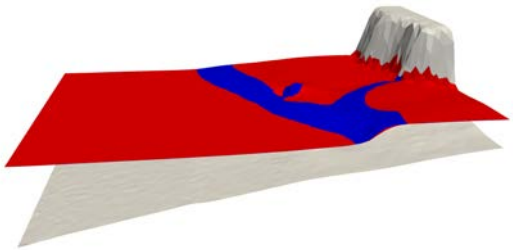
(e) S4. $t = 14.98$.



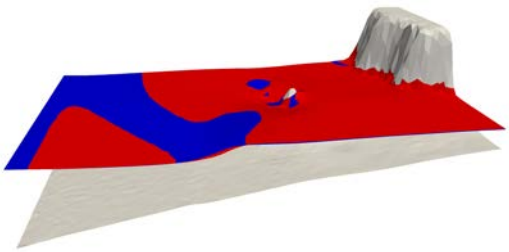
(f) S4. $t = 15.83$.



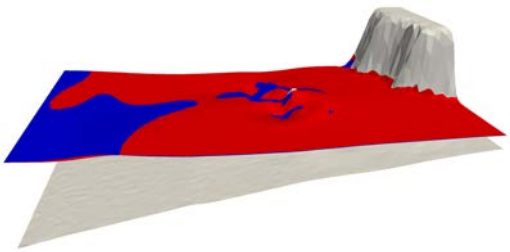
(g) S4. $t = 17.91$.



(h) S4. $t = 19.14$.



(i) S4. $t = 20.79$.



(j) S4. $t = 21.96$.

Figure 5.63: TC43. Time evolution of water depth.

## 5.6 Concluding remarks

In this final chapter, the intrusive POD-based ROM is applied to the 1D and 2D SWEs. The relationship between ROMs and the numerical corrections required by FOMs has been analysed in detail. From this part it can be concluded that it is necessary that ROMs: i) are well-balanced; ii) include the friction source term correction, the entropy fix and the wet/dry treatment; and iii) are trained by snapshots that take all corrections into account. Apart from that, ROMs do not need to include the source term corrections to ensure positive water depths.

Two methodologies have been proposed to include the entropy fix and the wet/dry treatment in the on-line phase of the ROM. Both of them operate only in the off-line phase when solving 1D problems, so they do not contribute to the ROM efficiency computation. The second, in 2D problems, involves adding a subsequent step of updating the flow discharges at each time step during the on-line phase. The results obtained show large speed-ups achieved by the ROM in 2D problems. Also, the results analysed for each of these corrections isolated show that their errors are shielded by the error introduced by solving a low number of POD modes. That is, their presence does not alleviate the efficiency of the ROM, but it does ensures physics fulfilment.

In addition to this, ROM prediction beyond the parametric training set has been extended to SWEs for different input parameters, including initial and boundary conditions and friction and bed source terms. The same limitation has been found for predicting the defining positions of the bed slope or the initial condition other than the training ones, for which the use of non-linear ROMs has been proposed as an alternative. The prediction beyond the training time has been applied to a linearized version of both 1D and 2D SWE, the latter thanks to the Radon transform.

Finally, the last three test cases have been used to show the application of ROMs to more complex problems governed by the SWEs. All the tools proposed throughout the thesis have been used in these cases, including the PID method, the numerical corrections, the a posteriori criterion, etc. Their results have shown speed-ups of $\times 500$ and $\times 800$, with highly accurate results.

# Chapter 6

# Concluding remarks

## 6.1 Final conclusions

In this thesis, the application of intrusive POD-based ROMs to linear and nonlinear, parabolic and hyperbolic equations, and, above all, to SWEs, has been carried out. A large number of ROMs have been developed to serve as alternative numerical schemes to FOMs. These have been properly validated by means of the test cases presented.

The main difference with previous works in the literature is the detailed and exhaustive study of the application of the ROMs to shallow water problems, with special emphasis on the needs of each of them, such as the well-balancing and the numerical corrections. This is due to the intrusive nature of the ROMs used, which allows them to be developed carefully, consciously and directly focused on the target problem.

Starting with the simplest problems, ROMs of the linear diffusion-advection equation, in their 1D and 2D versions, have been developed and applied with satisfactory results in terms of efficiency. In addition, the inclusion of Dirichlet-type time-dependent boundary conditions in the ROMs has been successfully solved, proving that machine accuracy is recovered. It was found that if the ROM is developed from a high-order FOM (i.e., the ROM is of high order), even if it is trained by lower-order solutions, its results converge to its own order. This can be highly useful, as high-order ROMs are much less expensive to solve than higher-order FOMs and their solutions retain their high accuracy.

Non-linear ROMs have been developed from non-linear equations, as in the case of Burgers, which allow high-precision solutions to be obtained. However, there are some cases in which it is not possible to develop non-linear ROMs, as in the case of the augmented Roe scheme applied to SWEs due to the presence of conserved variables in denominators and square roots. This makes the use of linearized ROMs mandatory, which, in addition, allow for improved efficiency with respect to non-linear ones. A series of linearized ROMs applied to the Burgers equation and SWEs have been proposed. The use of the PID method and, therefore, of time windows, negates the possibility of using the a priori criterion to select the number of POD modes with which to solve the ROM. To solve this impossibility, an a posteriori criterion based on the efficiency of the solutions has been proposed to find the optimal pair of values of the number of POD modes and time windows.

A detailed analysis of the consistency of ROMs in their application to SWEs has been carried

out, from which the following conclusions can be drawn: i) ROMs have to be developed from well-balanced FOMs (i.e. ROMs have to be well-balanced) and have to be trained by well-balanced solutions in order to obtain well-balanced solutions. This allows development and training to be performed by different FOMs while maintaining high solution accuracy and the well-balanced property; ii) ROMs have to take into account the friction source term correction and the entropy fix in their off-line phase and have to be trained by corrected solutions to obtain physically meaningful solutions; iii) In the same way, ROMs require wet/dry treatment in their off-line phase (and part of the on-line phase in 2D cases, as explained in Section 5.2.1), in addition to treated snapshots; and iv) on the contrary, the numerical correction on the source terms to ensure positive water depths does not need to be taken into account when solving the ROM. It is only necessary that the training snapshots are suitably corrected.

Results with satisfactory efficiencies have been obtained using ROMs in realsitic 2D shallow water problems, achieving several orders of magnitude of acceleration without losing significant accuracy in the solutions.

The ability of ROMs to compute solutions for values of the problem parameters other than the training ones has been tested. This has been achieved thanks to the modification of the POD method presented in the thesis, according to which all the training samples are grouped in the same snapshot matrix (with the desired number of time windows). This has made it possible to solve the ROM for different values of the initial and boundary conditions, as well as for other parameters specific to the problem, such as the advection velocity or the diffusion coefficient in the linear diffusion-advection equation, the geometric source term of the Burgers equation or the friction and slope source terms of the SWE. Also, the limitation of linearized ROMs to solve for different positions of the initial conditions, the geometric term or the slope term from those of the training has been verified. This can be overcome, as shown, by using non-linear ROMs.

By means of an appropriate coordinate transformation, CTROMs are able to obtain solutions beyond the training time. This technique is easily applicable to linear problems and allows high accuracy solutions to be calculated for times much longer than the training time. When working with non-linear problems it is necessary to take into account a modification of the strategy based on the addition of characteristic curves. This allows to predict solutions in time in the presence of shock and rarefaction waves. By means of the Radon transform, 2D problems can be transformed to 1D and the CTROM methodology can be applied to them.

## 6.2 Future work

The study carried out in this thesis has opened the door to extending the use of intrusive POD-based ROMs in several directions.

There is the possibility of combining ROMs with other numerical schemes. It would be very interesting to study the use of HLL-type numerical schemes in order to compare them with all the knowledge gained with the Roe and Lax-Friedrichs type schemes. It would also be valuable to continue the application of ROMs to high-order schemes. Runge-Kutta type schemes and

WENO/TENO type reconstructions could be useful for this purpose. It would be desirable not to remain only in the field of FV. There is already work in the literature applying ROMs to Galerkin-based schemes, such as the Discontinuous Galerkin method [169]. It would be worthwhile to apply this to the SWEs and it would also serve to extend the study of the relationship between high order and ROMs.

The ROM strategy could be used to speed up the computational calculations of other methods. For example, in solving the adjoint method, ROMs could be included to solve the forward and backward calculations in time that are necessary to find the target value.

The definition of the parameters defining the ROM can also be improved to optimise its resolution. Firstly, the number of POD modes could be defined dynamically in each time window, since it is possible that in more transient moments of the simulation a very high number is required and when stationary is reached, a few modes are sufficient. Secondly, following the same idea, the number of windows could be adapted to be refined when necessary. In both cases it would be necessary to carry out a post-processing of the training snapshots to analyse at which moments the parameters can be lowered and at which moments it is necessary to increase their value.

Within the framework of prediction in time, it would be interesting to combine the Radon transform with the transformations needed to resolve shock and rarefaction waves as presented in the Section 4.3 to solve 2D non-linear problems, such as the 2D SWE.

## 6.3 Main contributions

The following is a summary of the main contributions generated by this thesis. It begins with the most general aspects, followed by articles and other publications directly related to the content of the thesis, and ends with talks and other communications at conferences and workshops.

### 6.3.1 General contributions

- Production of numerical calculation tools based on intrusive POD-based ROMs for the following equations
  - 1D and 2D linear advection-diffusion equation,
  - 1D Burgers equation with geometrical source term
  - 1D and 2D shallow water equations,

  and based on the following numerical schemes
  - first-order upwind scheme,
  - optimized Lax-Friedrichs scheme,
  - Lax-Wendroff scheme,

  - augmented Roe scheme.

- Development of a version of the Lax-Friedrichs scheme applied to SWE in which the well-balanced property is corrected by means of a new source term.
- Development of a novel technique to achieve the inclusion of Dirichlet type boundary conditions in intrusive POD-based ROMs with machine accuracy.
- Development of novel technique to predict solutions using ROMs beyond the training time.
- Development of modified PID method to predict solutions on parametrized problems using ROMs, i.e. going beyond the training values of the parameters that define the problem (IC, BC, Manning coefficient, bed slope, etc.).

### 6.3.2 Articles and other publications

- Article published (01/09/2020) in Journal of Hydroinformatics [132] with title: *Discontinuous Galerkin well-balanced schemes using augmented Riemann solvers with application to the shallow water equations.* Authors: **A. Navas-Montilla, P. Solán-Fustero, J. Murillo, and P. García-Navarro**.
- Article published (15/06/2021) in Journal of Computational Physics [175] with title: *Application of approximate dispersion-diffusion analyses to under-resolved Burgers turbulence using high resolution WENO and UWC schemes.* Authors: **P. Solán-Fustero and A. Navas-Montilla and E. Ferrer and J. Manzanero and P. García-Navarro**.
- Article published (15/12/2022) in Journal of Computational Physics [172] with title: *A POD-based ROM strategy for the prediction in time of advection-dominated problems.* Authors: **P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro**.
- Article published (15/05/2023) in Computer Methods in Applied Mechanics and Engineering [176] with title: *Development of POD-based Reduced Order Models applied to shallow water equations using augmented Riemann solvers.* Authors: **P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro**.
- Article published (29/06/2024) in Computers & Fluids [133] with title: *Exploring the potential of TENO and WENO schemes for simulating under-resolved turbulent flows in the atmosphere using Euler equations.* Authors: **A. Navas-Montilla, J. Guallart, P. Solán-Fustero, and P. García-Navarro**.
- Article published as conference proceedings by International Association for Hydro-Environment Engineering and Research (IAHR) [174] with title: *A 1D Shallow Water Reduced-Order Model Based on POD.* Authors: **P. Solán-Fustero, A. Navas-Montilla, J.L. Gracia, and P. García-Navarro**.

- Article published as conference proceedings in IOP Conf. Series: Earth and Environmental Science [173] with title: *A POD-based reduced order model applied to 1D shallow water equations.* Authors: **P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro**.

### 6.3.3 Talks and other communications

- Poster presented (18/11/2020) in 1st IAHR Young Professionals Congress, with title: *A well-balanced DG scheme for the resolution of the shallow water equations.*
- Talk given (28/07/2021) in 7th International Conference on Numerical Methods for Hyperbolic Problems, NumHyp 2021, with title: *Application of approximate dispersion-diffusion analyses to under-resolved Burgers turbulence using high resolution WENO and UWC schemes.*
- Poster presented (03/09/2021) in XIX Jacques-Louis Lions Spanish-French School on Numerical Simulation in Physics and Engineering, Madrid, Spain, with title: *POD method applied to Hyperbolic Equations.*
- Talk given (21/10/2021) in II Jornada del I3A y X Jornada de Jóvenes Investigadores del I3A, Zaragoza, Spain, with title: *A New POD Method for Transport Equations.* Award for the best contribution from the ICT and Industrial Technologies Divisions.
- Poster presented (01/12/2021) in 2nd IAHR Young Professionals Congress with title: *A priori estimation of the performance of WENO and UWC schemes as iLES methods.*
- Poster presented (16/06/2022) in XI Jornada de Jovenes Investigadores del I3A, Zaragoza, Spain, with title: *A reduced-order model applied to the 2D unsteady shallow water equations.*
- Talk given (22/06/2022) in 39th IAHR World Congress, Granada, Spain, with title: *A 1D shallow water reduced-order model based on POD.*
- Talk given (06/07/2022) in 14th International Conference on Hydroinformatics - HIC 2022, Bucharest, Romania. with title: *A POD-Based Reduced Order Model Applied to 1D Shallow Water Equations.*
- Poster presented (18/07/2022) in XXVII Congress of Differential Equations and Applications XVII Congress of Applied Mathematics, Zaragoza, Spain, with title: *Predictive ROM based on a coordinate transform technique applied to 1D and 2D transport problems.*
- Talk given (08/09/2022) in Conference Zaragoza-Pau on Mathematics and its Applications, Jaca, Spain, with title: *POD-based ROM modified to predict solutions in time.*
- Talk given (09/02/2023) in VI Congreso de Jóvenes Investigadores de la Real Sociedad Matemática Española, León, Spain, with title: *Reduced-order models applied to hyperbolic transport equations.*

- Poster presented (09/03/2023) in ARIA 1st Workshop, Bordeaux, France, with title: *Coordinate-transform ROM to extrapolate solutions in time with advection-dominated problems.*
- Talk given (04/08/2023) in Bi-annual congress of the Italian Society of Applied and Industrial Mathematics (SIMAI), University of Basilicata, Matera, Italia, with title: *Study on the consistency of POD-based ROM applied to SWE solved with augmented Riemann solvers.*
- Talk given (09/01/2024) in Jornada IUI Modelización matemática: teoría y aplicaciones, Universidad de Zaragoza, Zaragoza, Spain, with title: *Métodos numéricos para flujos hiperbólicos y aplicaciones geofísicas y biomédicas.*
- Poster presented (18/01/2024) in XI International Conference BIFI 2024, Universidad de Zaragoza, Zaragoza, Spain, with title: *Study on the combination of POD-based ROMs and to augmented Riemann solvers applied to 1D SWE.*
- Talk given (01/07/2024) in 9th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2024, Lisbon, Portugal, with title *A POD-based reduced order finite volume model for solving shallow water equations with parameterized source terms.*
- Talk given (27/07/2024) in XXVIII Congress of Differential Equations and Applications / XVIII Congress of Applied Mathematics (CEDYA/CMA), Bilbao, Spain, with title *POD-based reduced-order modelling applied to parametrized hyperbolic problems.*

# Conclusiones

## Conclusiones finales

En esta tesis se ha llevado a cabo la aplicación de ROMs intrusivos basados en el método POD a ecuaciones lineales y no lineales, parabólicas e hiperbólicas, y, sobre todo, a las SWEs. Se han desarrollado un gran número de ROMs que actúan como esquemas numéricos alternativos a los FOMs clásicos. Éstos han sido debidamente validados mediante los casos de prueba presentados.

La principal diferencia con trabajos previos en la literatura es el estudio detallado y exhaustivo de la aplicación de los ROMs a problemas de aguas poco profundas, con especial énfasis en las necesidades de cada uno de ellos, como la propiedad *well-balanced* y las correcciones numéricas. Esto se debe a la naturaleza intrusiva de los ROMs utilizados, que permite desarrollarlos de forma cuidadosa, consciente y enfocada directamente al problema objetivo.

Partiendo de los problemas más sencillos, se han desarrollado y aplicado ROMs de la ecuación lineal de difusión-advección, en sus versiones 1D y 2D, con resultados satisfactorios en términos de eficiencia. Además, se ha resuelto con éxito la inclusión de condiciones de contorno dependientes del tiempo de tipo Dirichlet en los ROMs, demostrando que se recupera la solución con precisión de la máquina. Se ha comprobado que si el ROM se desarrolla a partir de un FOM de alto orden (es decir, el ROM es de alto orden), aunque se entrene mediante soluciones de orden inferior, sus resultados convergen a su propio orden. Esto puede ser muy útil, ya que los ROMs de alto orden son mucho menos costosos de resolver que los FOMs de orden superior y sus soluciones conservan su alta precisión.

Se han desarrollado ROMs no lineales a partir de ecuaciones no lineales, como en el caso de Burgers, que permiten obtener soluciones de alta precisión. Sin embargo, existen algunos casos en los que no es posible desarrollar ROMs no lineales, como en el caso del esquema de Roe aumentado aplicado a las SWEs debido a la presencia de variables conservadas en denominadores y raíces cuadradas. Esto hace obligatorio el uso de ROMs linealizados que, además, permiten mejorar la eficiencia respecto a los no lineales. Se han propuesto una serie de ROMs linealizadas aplicadas a la ecuación de Burgers y a las SWEs. La utilización del método PID y, por tanto, de ventanas temporales, anula la posibilidad de utilizar el criterio a-priori para seleccionar el número de modos POD con los que resolver el ROM. Para solventar esta imposibilidad, se ha propuesto un criterio a-posteriori basado en la eficiencia de las soluciones para encontrar la pareja óptima de valores del número de modos POD y de ventanas temporales.

Se ha llevado a cabo un análisis detallado de la propiedad de coherencia de los ROMs en su aplicación a las SWE, del que se pueden extraer las siguientes conclusiones: i) los ROMs

tienen que desarrollarse a partir de FOMs well-balanced (es decir, los ROMs tienen que ser well-balance) y tienen que entrenarse mediante soluciones well-balance para obtener soluciones well-balance. Esto permite que el desarrollo y el entrenamiento se lleven a cabo por diferentes FOMs mientras se mantiene una alta precisión de la solución y la propiedad *well-balanced*; ii) los ROMs tienen que tener en cuenta la corrección de entropía en su fase off-line y tienen que ser entrenados por soluciones corregidas para obtener soluciones físicamente significativas; iii) de la misma manera, los ROMs requieren un tratamiento del seco/mojado en su fase off-line (y parte de la fase on-line en casos 2D, como se explica en la Sección 5.2.1), además de requerir *snapshots* corregidos; y iv) por el contrario, el resto de correcciones numéricas sobre los términos fuente para asegurar ajustes positivos y evitar flujos revertidos no necesitan ser tenidas en cuenta a la hora de resolver el ROM. Sólo es necesario que los *snapshots* de entrenamiento estén convenientemente corregidos.

Se han obtenido resultados con eficiencias satisfactorias utilizando ROMs en problemas 2D reales de aguas poco profundas, logrando varios órdenes de magnitud de aceleración sin perder precisión significativa en las soluciones.

Se ha probado la capacidad de los ROMs para calcular soluciones para valores de los parámetros del problema distintos de los de entrenamiento. Esto se ha conseguido gracias a la modificación del método POD presentada en la tesis, según la cual todas las muestras de entrenamiento se agrupan en una misma matriz de *snapshots* (con el número de ventanas temporales deseado). Esto ha permitido resolver el ROM para diferentes valores de las condiciones iniciales y de contorno, así como para otros parámetros específicos del problema, como la velocidad de advección o el coeficiente de difusión en la ecuación de difusión-advección lineal, el término fuente geométrico de la ecuación de Burgers o los términos fuente de fricción y lecho de la SWE. También se ha comprobado la limitación de los ROMs linealizados para resolver para diferentes posiciones de las condiciones iniciales, el término geométrico o el término de lecho respecto a las del entrenamiento. Esto puede superarse, como se ha demostrado, utilizando ROMs no lineales.

Mediante una transformación de coordenadas adecuada, los CTROM son capaces de obtener soluciones más allá del tiempo de entrenamiento. Esta técnica es fácilmente aplicable a problemas lineales y permite calcular soluciones de alta precisión para tiempos muy superiores al tiempo de entrenamiento. Cuando se trabaja con problemas no lineales es necesario tener en cuenta una modificación de la estrategia basada en la adición de curvas características. Esto permite predecir soluciones en el tiempo en presencia de ondas de choque y rarefacción. Mediante la transformada de Radon se pueden transformar problemas 2D a 1D y aplicarles la metodología CTROM.

## Trabajo futuro

El estudio realizado en esta tesis ha abierto la puerta a ampliar el uso de ROMs intrusivos basados en POD en varias direcciones.

Existe la posibilidad de combinar ROMs con otros esquemas numéricos. Sería muy interesante

estudiar el uso de esquemas numéricos de tipo HLL para compararlos con todos los conocimientos adquiridos con los esquemas de tipo Roe y Lax-Friedrichs. También sería valioso continuar con la aplicación de ROMs a esquemas de alto orden. Los esquemas de tipo Runge-Kutta y las reconstrucciones de tipo WENO/TENO podrían ser útiles para este fin. Sería deseable no quedarse sólo en el campo de los métodos FV. Ya hay trabajos en la literatura aplicando ROMs a esquemas basados en Galerkin, como el método *Discontinuous Galerkin* [169]. Merecería la pena aplicarlo a las SWE y además serviría para ampliar el estudio de la relación entre alto orden y ROMs.

La estrategia ROM podría utilizarse para acelerar los cálculos computacionales de otros métodos. Por ejemplo, en la resolución del método adjunto, podrían incluirse ROMs para resolver los cálculos hacia delante y hacia atrás en el tiempo que son necesarios para encontrar el valor objetivo.

La definición de los parámetros que definen el ROM también puede mejorarse para optimizar su resolución. En primer lugar, el número de modos POD podría definirse dinámicamente en cada ventana temporal, ya que es posible que en los momentos más transitorios de la simulación se requiera un número muy elevado y cuando se alcanza el régimen estacionario basten unos pocos modos. En segundo lugar, siguiendo la misma idea, se podría adaptar el número de ventanas para refinarlas cuando fuera necesario. En ambos casos sería necesario realizar un post-procesado de los *snapshots* de entrenamiento para analizar en qué momentos se pueden bajar los parámetros y en qué momentos es necesario aumentar su valor.

En el marco de la predicción en el tiempo, sería interesante combinar la transformada de Radon con las transformaciones necesarias para resolver las ondas de choque y rarefacción tal y como se presentan en la Sección 4.3 para resolver problemas no lineales 2D, como el SWE 2D.

# Bibliography


[1] R. Abadía-Heredia, M. López-Martín, B. Carro, J.I. Arribas, J.M. Pérez, and S. Le Clainche. A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures. *Exp. Sys. App.*, 187:115910, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.115910. URL https://www.sciencedirect.com/science/article/pii/S0957417421012653.

[2] L.I. Abreu, A.V.G. Cavalieri, P. Schlatter, R. Vinuesa, and D.S. Henningson. Spectral proper orthogonal decomposition and resolvent analysis of near-wall coherent structures in turbulent pipe flows. *J. Fluid Mec.*, 900:A11, 2020. doi: 10.1017/jfm.2020.445.

[3] M. Ahmed and O. San. Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks. *Comput. Appl. Math.*, 37: 6870–6902, 2018.

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

[5] I. Akhtar, Z. Wang, J. Borggaard, and T. Iliescu. A New Closure Strategy for Proper Orthogonal Decomposition Reduced-Order Models. *J. Comput. Nonlinear Dyn.*, 7(3), 04 2012. ISSN 1555-1415. 034503.

[6] F. Alcrudo and P. Garcia-Navarro. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *Int. J. for Numer. Methods Fluids*, 16(6):489–505, 1993. doi: https://doi.org/10.1002/fld.1650160604. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650160604.

[7] A. Alla and M. Falcone. A Time-Adaptive POD Method for Optimal Control Problems. *IFAC Proc. Vol.*, 46(26):245–250, 2013. ISSN 1474-6670. doi: https://doi.org/10.3182/20130925-3-FR-4043.00042. URL https://www.sciencedirect.com/science/article/pii/S1474667015378307. 1st IFAC Workshop on Control of Systems Governed by Partial Differential Equations.

[8] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16(5):525–528, 1978. doi: 10.2514/3.7539. URL https://doi.org/10.2514/3.7539.

[9] D. Amsallem and C. Farhat. An Online Method for Interpolating Linear Parametric Reduced-Order Models. *SIAM J. Sci. Comp.*, 33(5):2169–2198, 2011. doi: 10.1137/100813051. URL https://doi.org/10.1137/100813051.

[10] D. Amsallem, M.J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Num. Met. Eng.*, 92(10):891–916, 2012. doi: https://doi.org/10.1002/nme.4371. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4371.

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

[12] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering, 2004.

[13] F. Ballarin and G. Rozza. POD–Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems. *Int. J. Num. Met. Flu.*, 82(12):1010–1034, 2016. doi: https://doi.org/10.1002/fld.4252. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4252.

[14] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Int. J. Num. Met. Eng.*, 102(5):1136–1161, 2015. doi: https://doi.org/10.1002/nme.4772. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4772.

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

[16] F. Behzad, B.T. Helenbrook, and G. Ahmadi. On the sensitivity and accuracy of proper-orthogonal-decomposition-based reduced order models for burgers equation. *Comp. Flu.*, 106:19–32, 2015. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2014.09.041. URL https://www.sciencedirect.com/science/article/pii/S0045793014003867.

[17] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *J. Comp. Phys.*, 227(16):7813–7840, 2008. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2008.04.034. URL https://www.sciencedirect.com/science/article/pii/S0021999108002659.

[18] A. Bermudez and M.E. Vázquez-Cendón. Upwind methods for hyperbolic conservation laws with source terms. *Comput. Fluids*, 23(8):1049–1071, 1994. ISSN 0045-7930. doi: https://doi.org/10.1016/0045-7930(94)90004-3. URL https://www.sciencedirect.com/science/article/pii/0045793094900043.

[19] G Bertaglia, C. Lu, L. Pareschi, and X. Zhu. Asymptotic-Preserving Neural Networks for multiscale hyperbolic models of epidemic spread. *Mat. Mod. Met. App. Sci.*, 32(10):

1949–1985, 2022. doi: 10.1142/S0218202522500452. URL https://doi.org/10.1142/S0218202522500452.

[20] J. Borggaard, A. Hay, and D. Pelletier. Interval-based reduced-order models for unsteady fluid flow. *Int. J. Numer. Anal. Model.*, 4:353–367, 2007.

[21] J. Borggaard, Z. Wang, and L. Zietsman. A goal-oriented reduced-order modeling approach for nonlinear systems. *Comp. Mat. App.*, 71(11):2155–2169, 2016. ISSN 0898-1221. doi: https://doi.org/10.1016/j.camwa.2016.01.031. URL https://www.sciencedirect.com/science/article/pii/S089812211630027X. Proceedings of the conference on Advances in Scientific Computing and Applied Mathematics. A special issue in honor of Max Gunzburger's 70th birthday.

[22] E. Botia-Vera, A. Souto-Iglesias, G. Bulian, and L. Lobovský. Three SPH Novel Benchmark Test Cases for free surface flows. In *Proceedings of the 5th ERCOFTAC SPHERIC workshop on SPH applications, 2010*, Manchester, UK, 2010. University of Manchester. URL http://www.mace.manchester.ac.uk/5thspheric/.

[23] N. Botta, R. Klein, S. Langenberg, and S. Lützenkirchen. Well balanced finite volume methods for nearly hydrostatic flows. *J. Comp. Phys.*, 196(2):539–565, 2004. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2003.11.008. URL https://www.sciencedirect.com/science/article/pii/S0021999103006077.

[24] P. Brufau, M. E. Vázquez-Cendón, and P. García-Navarro. A numerical model for the flooding and drying of irregular domains. *Int. J. Numer. Methods Fluids*, 39(3):247–275, 2002. doi: https://doi.org/10.1002/fld.285. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.285.

[25] J.M. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. App. Math.*, 1:171–199, 1948. ISSN 0065-2156. doi: https://doi.org/10.1016/S0065-2156(08)70100-5. URL https://www.sciencedirect.com/science/article/pii/S0065215608701005.

[26] J. Burguete and P. García-Navarro. Improving simple explicit methods for unsteady open channel and river flow. *Int. J. Numer. Methods Fluids*, 45(2):125–156, 2004. doi: https://doi.org/10.1002/fld.619. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.619.

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

[28] S. Busto Ulloa, M. Dumbser, C. Escalante Sánchez, N. Favrie, and S. Gavrilyuk. On High Order ADER Discontinuous Galerkin Schemes for First Order Hyperbolic Reformulations of Nonlinear Dispersive Systems. *J. Sci. Comp.*, 87, 05 2021. doi: 10.1007/s10915-021-01429-8.

[29] A. Bērziņš, J. Helmig, F. Key, and Stefanie N. E. Standardized Non-Intrusive Reduced Order Modeling Using Different Regression Models With Application to Complex Flow Problems. *ArXiv*, page 28 Seiten, 2020. URL https://publications.rwth-aachen.de/record/794141.

[30] O. Castro-Orgaz and W. Hager. *Non-Hydrostatic Free Surface Flows*. Springer, 01 2017. ISBN 978-3-319-47969-9. doi: 10.1007/978-3-319-47971-2.

[31] T. Chacón Rebollo, S. Rubino, M. Oulghelou, and C. Allery. Error analysis of a residual-based stabilization-motivated POD-ROM for incompressible flows. *Com. Met. App. Mec. Eng.*, 401:115627, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115627. URL https://www.sciencedirect.com/science/article/pii/S0045782522005825.

[32] Chapelle, D., Gariah, A., and Sainte-Marie, . Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESAIM: M2AN*, 46(4):731–757, 2012. doi: 10.1051/m2an/2011053. URL https://doi.org/10.1051/m2an/2011053.

[33] S. Chaturantabut and D.C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 4316–4321, 2009. doi: 10.1109/CDC.2009.5400045.

[34] E.A. Christensen, M. Brøns, and J.N. Sørensen. Evaluation of Proper Orthogonal Decomposition–Based Decomposition Techniques Applied to Parameter-Dependent Nonturbulent Flows. *SIAM J. Sci. Comput.*, 21(4):1419–1434, 1999. doi: 10.1137/S1064827598333181. URL https://doi.org/10.1137/S1064827598333181.

[35] B. Cockburn, S.Y. Lin, and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.*, 84(1):90–113, 1989. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(89)90183-6. URL https://www.sciencedirect.com/science/article/pii/0021999189901836.

[36] B. Cockburn, S. Hou, and C.W. Shu. The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. IV: The Multidimensional Case. *Math. Comput.*, 54(190):545–581, 1990. ISSN 00255718, 10886842. URL http://www.jstor.org/stable/2008501.

[37] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49(1):1 – 23, 1943.

[38] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Math. Ann.*, 100(1):32–74, 1928. ISSN 0025-5831. doi: 10.1007/BF01448839.

[39] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck. Model order reduction assisted by deep neural networks (ROM-net). *Ad. Mod. Sim. Eng. Sci.*, 7(1), December 2020. doi: 10.1186/s40323-020-00153-6. URL https://hal.science/hal-02539647.

[40] S.R. Deans. *The Radon transform and some of its applications*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1983. ISBN 0-471-89804-X.

[41] M. Dehghan and M. Abbaszadeh. The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations. *J. Comput. Phys.*, 351:478–510, 2017. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2017.09.007. URL https://www.sciencedirect.com/science/article/pii/S0021999117306575.

[42] J. Du, F. Fang, C.C. Pain, I.M. Navon, J. Zhu, and D.A. Ham. POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow. *Comp. Math. App.*, 65(3): 362–379, 2013. ISSN 0898-1221. doi: https://doi.org/10.1016/j.camwa.2012.06.009. URL https://www.sciencedirect.com/science/article/pii/S0898122112004397. Efficient Numerical Methods for Scientific Applications.

[43] I. Echeverribar, M. Morales-Hernández, P. Brufau, and P. García-Navarro. Numerical simulation of 2D real large scale floods on GPU: the Ebro river. *E3S Web Conf.*, 40:06007, 2018. doi: 10.1051/e3sconf/20184006007. URL https://doi.org/10.1051/e3sconf/20184006007.

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

[45] I. Echeverribar, M. Morales-Hernández, P. Brufau, and P. García-Navarro. Analysis of the performance of a hybrid CPU/GPU 1D2D coupled model for real flood cases. *J. Hydro.*, 22(5):1198–1216, 07 2020. ISSN 1464-7141. doi: 10.2166/hydro.2020.032. URL https://doi.org/10.2166/hydro.2020.032.

[46] I. Echeverribar, S. Martínez-Aranda, J. Fernández-Pato, and P. García-Navarro. A GPU-based 2D viscous flow model with variable density and heat exchange. *Adv. Eng. Soft.*, 175:103340, 2023.

[47] J. Eggers. Nonlinear dynamics and breakup of free-surface flows. *Rev. Mod. Phys.*, 69: 865–930, Jul 1997. doi: 10.1103/RevModPhys.69.865. URL https://link.aps.org/doi/10.1103/RevModPhys.69.865.

[48] B. Einfeldt. On Godunov-Type Methods for Gas Dynamics. *SIAM J. Num. Anal.*, 25 (2):294–318, 1988. doi: 10.1137/0725021. URL https://doi.org/10.1137/0725021.

[49] M. Fahl. *Trust-region Methods for Flow Control based on Reduced Order Modelling*. doctoralthesis, Universität Trier, 2004.

[50] A. Falcó, L. Hilario, N. Montés, and M.C. Mora. Numerical strategies for the Galerkin proper generalized decomposition method. *Math. Comp. Mod.*, 57(7):1694–1702, 2013. ISSN 0895-7177. doi: https://doi.org/10.1016/j.mcm.2011.11.012. URL https://www.sciencedirect.com/science/article/pii/S0895717711006972. Public Key Services and Infrastructures EUROPKI-2010-Mathematical Modelling in Engineering & Human Behaviour 2011.

[51] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceedings of the Conference on European Design Automation*, EURO-DAC '94, page 170–175, Washington, DC, USA, 1994. IEEE Computer Society Press. ISBN 0897916859.

[52] J. Fernández-Pato, M. Morales-Hernández, and P. García-Navarro. Implicit finite volume simulation of 2D shallow water flows in flexible meshes. *Comp. Met. App. Mec. Eng.*, 328:1–25, 2018. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2017.08.050. URL https://www.sciencedirect.com/science/article/pii/S0045782517302724.

[53] E. Ferrer, G. Rubio, G. Ntoukas, W. Laskowski, O.A. Mariño, S. Colombo, A. Mateo-Gabín, H. Marbona, F. Manrique de Lara, D. Huergo, J. Manzanero, A.M. Rueda-Ramírez, D.A. Kopriva, and E. Valero. Image 1: A high-order discontinuous galerkin solver for flow simulations and multi-physics applications. *Computer Physics Communications*, 287:108700, 2023. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2023.108700. URL https://www.sciencedirect.com/science/article/pii/S0010465523000450.

[54] L. Fick, Y. Maday, A.T. Patera, and T. Taddei. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *J. Comp. Phys.*, 371:214–243, 2018. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.05.027. URL https://www.sciencedirect.com/science/article/pii/S0021999118303279.

[55] S. Fresca and A. Manzoni. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Comp. Met. App. Mec. Eng.*, 388:114181, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2021.114181. URL https://www.sciencedirect.com/science/article/pii/S0045782521005120.

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

[57] T. Gallouët, J.M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow-water equations with topography. *Compu. & Flu.*, 32(4):479–513, 2003. ISSN 0045-7930. doi: https://doi.org/10.1016/S0045-7930(02)00011-7. URL https://www.sciencedirect.com/science/article/pii/S0045793002000117.

[58] S. Gangrade, G. R. Ghimire, S.C. Kao, M. Morales-Hernández, A.A. Tavakoly, J.L. Gutenson, K.H. Sparrow, G.K. Darkwah, A.J. Kalyanapu, and M.L. Follum. Unraveling the 2021 Central Tennessee flood event using a hierarchical multi-model inundation

modeling framework. *J. Hydro*, 625:130157, 2023. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2023.130157. URL https://www.sciencedirect.com/science/article/pii/S0022169423010995.

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

[60] B. García-Archilla, J. Novo, and S. Rubino. Error analysis of proper orthogonal decomposition data assimilation schemes with grad–div stabilization for the Navier–Stokes equations. *J. Comp. App. Math.*, 411:114246, 2022. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2022.114246. URL https://www.sciencedirect.com/science/article/pii/S0377042722000954.

[61] B. García-Archilla, J. Novo, and S. Rubino. On the influence of the nonlinear term in the numerical approximation of Incompressible Flows by means of proper orthogonal decomposition methods. *Comp. Met. App. Mec. Eng.*, 405:115866, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115866. URL https://www.sciencedirect.com/science/article/pii/S0045782522008222.

[62] S. Georgaka, G. Stabile, G. Rozza, and M. Bluck. Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems, 08 2018.

[63] M. Girfoglio, A. Quaini, and G. Rozza. A POD-Galerkin reduced order model for a LES filtering approach. *J. Comp. Phys.*, 436:110260, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110260. URL https://www.sciencedirect.com/science/article/pii/S0021999121001558.

[64] S.K. Godunov and I. Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij Sbornik*, 47(89)(3):271–306, 1959. URL https://hal.archives-ouvertes.fr/hal-01620642.

[65] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. ISBN 9781421407944.

[66] G. Gordillo, M. Morales-Hernández, and P. García-Navarro. A gradient-descent adjoint method for the reconstruction of boundary conditions in a river flow nitrification model. *Environ. Sci.: Processes Impacts*, 22:381–397, 2020. doi: 10.1039/C9EM00500E. URL http://dx.doi.org/10.1039/C9EM00500E.

[67] G. Gordillo, M. Morales-Hernández, and P. García-Navarro. Solute Transport Control at Channel Junctions Using Adjoint Sensitivity. *Mathematics*, 10(1), 2022. ISSN 2227-7390. doi: 10.3390/math10010093. URL https://www.mdpi.com/2227-7390/10/1/93.

[68] M. Gosses, W. Nowak, and T. Wöhling. Explicit treatment for Dirichlet, Neumann and Cauchy boundary conditions in POD-based reduction of groundwater models. *Adv. Wat. Res.*, 115:160–171, 2018. ISSN 0309-1708. doi: https://doi.org/10.1016/

j.advwatres.2018.03.011. URL https://www.sciencedirect.com/science/article/pii/S0309170817307467.

[69] J.L. Gracia and E. O'Riordan. A singularly perturbed convection-diffusion problem with a moving interior layer. *Int. J. Numer. Anal. Model.*, 9(4):823–843, 2012. ISSN 1705-5105.

[70] J. M. Greenberg and A. Y. Leroux. A Well-Balanced Scheme for the Numerical Processing of Source Terms in Hyperbolic Equations. *SIAM J. Numer. Anal.*, 33(1):1–16, feb 1996. ISSN 0036-1429. doi: 10.1137/0733001. URL https://doi.org/10.1137/0733001.

[71] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM Math. Model. Numer. Anal.*, 39:157–181, 2005.

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

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

[74] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mod. math. ana. num.*, 42(2):277–302, 2008. doi: 10.1051/m2an:2008001. URL http://www.numdam.org/articles/10.1051/m2an:2008001/.

[75] D.A. Haleem, G. Kesserwani, and D Caviedes-Voullième. Haar wavelet-based adaptive finite volume shallow water solver. *J. Hydroinformatics*, 17(6):857–873, 07 2015. ISSN 1464-7141. doi: 10.2166/hydro.2015.039. URL https://doi.org/10.2166/hydro.2015.039.

[76] A Harten and J M Hyman. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comp. Phys.*, 50:2, 5 1983. doi: 10.1016/0021-9991(83)90066-9. URL https://www.osti.gov/biblio/5730902.

[77] A. Harten, P. D. Lax, and B. Leer. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Rev.*, 25(1):35–61, 1983. doi: 10.1137/1025002. URL https://doi.org/10.1137/1025002.

[78] C. Hirsch. *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier, 2007.

[79] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 1996. doi: 10.1017/CBO9780511622700.

[80] D.S. Hoskin, R.L. Van Heyningen, N.C. Nguyen, J. Vila-Pérez, W.L. Harris, and J. Peraire. Discontinuous Galerkin methods for hypersonic flows. *Prog. Aero. Sci.*, 146:100999, 2024. ISSN 0376-0421. doi: https://doi.org/10.1016/j.paerosci.2024.100999. URL https://www.sciencedirect.com/science/article/pii/S0376042124000253.

[81] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Edu. Psy.*, 24:498–520, 1933. URL https://api.semanticscholar.org/CorpusID:144828484.

[82] P. Houston, M.E. Hubbard, T.J. Radley, O.J. Sutton, and R.S.J. Widdowson. Efficient High-Order Space-Angle-Energy Polytopic Discontinuous Galerkin Finite Element Methods for Linear Boltzmann Transport. *J. Sci. Comp.*, 100(2), July 2024. ISSN 1573-7691. doi: 10.1007/s10915-024-02569-3. URL http://dx.doi.org/10.1007/s10915-024-02569-3.

[83] M.E. Hubbard. Multidimensional Slope Limiters for MUSCL-Type Finite Volume Schemes on Unstructured Grids. *J. Comp. Phys.*, 155(1):54–74, 1999. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1999.6329. URL https://www.sciencedirect.com/science/article/pii/S0021999199963295.

[84] M.E. Hubbard and N. Dodd. A 2D numerical model of wave run-up and overtopping. *Coast. Eng.*, 47(1):1–26, 2002. ISSN 0378-3839. doi: https://doi.org/10.1016/S0378-3839(02)00094-7. URL https://www.sciencedirect.com/science/article/pii/S0378383902000947.

[85] M.E. Hubbard and P. Garcia-Navarro. Flux Difference Splitting and the Balancing of Source Terms and Flux Gradients. *J. Comp. Phys.*, 165(1):89–125, 2000. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.2000.6603. URL https://www.sciencedirect.com/science/article/pii/S0021999100966038.

[86] W.L. IJzerman. *Signal Representation and Modeling of Spatial Structures in Fluids*. PhD thesis, University of Twente, April 2000.

[87] T. Iliescu, H. Liu, and X. Xie. Regularized Reduced Order Models for a Stochastic Burgers Equation. *Int. J. Num. Ana. Mod.*, 15(4-5):594–607, 2018. ISSN 2617-8710. doi: https://doi.org/. URL http://global-sci.org/intro/article_detail/ijnam/12533.html.

[88] S. Ingimarson, L.G. Rebholz, and T. Iliescu. Full and reduced order model consistency of the nonlinearity discretization in incompressible flows. *Comp. Met. App. Mec. Eng.*, 401:115620, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115620. URL https://www.sciencedirect.com/science/article/pii/S0045782522005758.

[89] G.S. Jiang and C.W. Shu. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys.*, 126(1):202–228, 1996. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1996.0130. URL https://www.sciencedirect.com/science/article/pii/S0021999196901308.

[90] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002. ISBN 9780387954424. URL https://books.google.es/books?id=_olByCrhjwIC.

[91] B. Karasözen, S. Yıldız, and M. Uzunca. Intrusive and data-driven reduced order modelling of the rotating thermal shallow water equation. *App. Math. Comp.*, 421: 126924, 2022. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2022.126924. URL https://www.sciencedirect.com/science/article/pii/S0096300322000108.

[92] K. Karhunen. Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, 37:1–79, 1947.

[93] B. Koc, T. Chacón-Rebollo, and S. Rubino. Uniform Bounds with Difference Quotients for Proper Orthogonal Decomposition Reduced Order Models of the Burgers Equation. *J. Sci. Comput.*, 95(2), mar 2023. ISSN 0885-7474. doi: 10.1007/s10915-023-02160-2. URL https://doi.org/10.1007/s10915-023-02160-2.

[94] J. Kou, S. Le Clainche, and W. Zhang. A reduced-order model for compressible flows with buffeting condition using higher order dynamic mode decomposition with a mode selection criterion. *Physics of Fluids*, 30(1):016103, 01 2018. ISSN 1070-6631. doi: 10.1063/1.4999699. URL https://doi.org/10.1063/1.4999699.

[95] P.K. Kundu and I.M. Cohen. *Fluid Mechanics*. Elsevier Science, 2010. ISBN 9780123814005. URL https://books.google.es/books?id=d9B5NElxUKwC.

[96] K. Kunisch and S. Volkwein. Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition. *J. Opt. The. App.*, 102(2):345–371, August 1999. doi: 10.1023/A:1021732508059. URL https://ideas.repec.org/a/spr/joptap/v102y1999i2d10.1023_a1021732508059.html.

[97] J. Kutz, S. Brunton, B. Brunton, and J. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM-Society for Industrial and Applied Mathematics, 11 2016. ISBN 978-1-611974-49-2.

[98] S. Le Clainche and E. Ferrer. A Reduced Order Model to Predict Transient Flows around Straight Bladed Vertical Axis Wind Turbines. *Energies*, 11(3), 2018. ISSN 1996-1073. doi: 10.3390/en11030566. URL https://www.mdpi.com/1996-1073/11/3/566.

[99] S. Le Clainche and J.M. Vega. Higher Order Dynamic Mode Decomposition. *SIAM J. App. Dyn. Sys.*, 16(2):882–925, 2017. doi: 10.1137/15M1054924. URL https://doi.org/10.1137/15M1054924.

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

[101] R.J. LeVeque. Wave Propagation Algorithms for Multidimensional Hyperbolic Systems. *Journal of Computational Physics*, 131(2):327–353, 1997. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1996.5603. URL https://www.sciencedirect.com/science/article/pii/S002199919695603X.

[102] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. doi: 10.1017/CBO9780511791253.

[103] R.J. LeVeque. A well-balanced path-integral f-wave method for hyperbolic problems with source terms. *J. Sci. Comput.*, 58:209–226, 2011.

[104] X. Li, D. Fu, J. Nielsen-Gammon, S. Gangrade, S.C. Kao, P. Chang, Z. Zhang, N. Voisin, M. Morales-Hernández, and H. Gao. Impacts of climate change on future hurricane induced rainfall and flooding in a coastal watershed: a case study on hurricane harvey. *J. Hydro.*, 616:128774, 2023. URL https://www.sciencedirect.com/science/article/pii/S0022169422013440.

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

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

[107] X.D. Liu, S. Osher, and T. Chan. Weighted Essentially Non-oscillatory Schemes. *J. Comput. Phys.*, 115(1):200–212, 1994. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1994.1187. URL https://www.sciencedirect.com/science/article/pii/S0021999184711879.

[108] M. Loève. *Probability theory*. The University Series in Higher Mathematics. D. Van Nostrand Co., Inc., Princeton, N. J.-Toronto-New York-London, 1960. 2nd ed.

[109] H. Lu and D.M. Tartakovsky. Predictive Accuracy of Dynamic Mode Decomposition, 2019. URL https://arxiv.org/abs/1905.01587.

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

[111] J.L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–176, 1967.

[112] Z. Luo, H. Li, Y. Zhou, and Z. Xie. A reduced finite element formulation based on POD method for two-dimensional solute transport problems. *J. Math. Anal. Appl.*, 385(1):371–383, 2012. ISSN 0022-247X. doi: 10.1016/j.jmaa.2011.06.051. URL https://doi.org/10.1016/j.jmaa.2011.06.051.

[113] W. Maccormack and A.J. Paullay. *Computational efficiency achieved by time splitting of finite difference operators*. AIAA, 1972. doi: 10.2514/6.1972-154. URL https://arc.aiaa.org/doi/abs/10.2514/6.1972-154.

[114] J. Mairal, J. Murillo, and P. García-Navarro. The entropy fix in augmented Riemann solvers in presence of source terms: Application to the Shallow Water Equations. *Comp. Met. App. Mec. Eng.*, 417:116411, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2023.116411. URL https://www.sciencedirect.com/science/article/pii/S0045782523005352.

[115] S. Martínez-Aranda, A. Ramos-Pérez, and P. García-Navarro. A 1D shallow-flow model for two-layer flows based on FORCE scheme with wet"–"dry treatment. *J. Hydroinformatics*, 22(5):1015–1037, 06 2020. ISSN 1464-7141. doi: 10.2166/hydro.2020.002. URL https://doi.org/10.2166/hydro.2020.002.

[116] P. W. McDonald. The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades. volume ASME 1971 International Gas Turbine Conference and Products Show of *Turbo Expo: Power for Land, Sea, and Air*, page V001T01A089, 03 1971. doi: 10.1115/71-GT-89. URL https://doi.org/10.1115/71-GT-89.

[117] G. Mengaldo and R. Maulik. PySPOD: A Python package for Spectral Proper Orthogonal Decomposition (SPOD). *J. Open Sou. Sof.*, 6(60):2862, 2021. doi: 10.21105/joss.02862. URL https://doi.org/10.21105/joss.02862.

[118] M. Milano and P. Koumoutsakos. Neural Network Modeling for Near Wall Turbulent Flow. *J. Comput. Phys.*, 182(1):1–26, 2002. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.2002.7146. URL https://www.sciencedirect.com/science/article/pii/S0021999102971469.

[119] R. Mojgani and M. Balajewicz. Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows, 2017. URL https://arxiv.org/abs/1701.04343.

[120] R. Mojgani and M. Balajewicz. Low-Rank Registration Based Manifolds for Convection-Dominated PDEs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (1):399–407, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16116.

[121] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Tra. Auto. Cont.*, 26(1):17–32, 1981. doi: 10.1109/TAC.1981.1102568.

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

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

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

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

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

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

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

[129] D.A. Nagy. Modal representation of geometrically nonlinear behavior by the finite element method. *Comput. Struct.*, 10(4):683–688, 1979. ISSN 0045-7949. doi: https://doi.org/10.1016/0045-7949(79)90012-9. URL https://www.sciencedirect.com/science/article/pii/0045794979900129.

[130] A. Navas-Montilla and J. Murillo. 2D well-balanced augmented ADER schemes for the shallow water equations with bed elevation and extension to the rotating frame. *J. Comput. Phys.*, 372:316–348, 2018. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.06.039. URL https://www.sciencedirect.com/science/article/pii/S0021999118304170.

[131] A. Navas-Montilla, C. Juez, M.J. Franca, and J. Murillo. Depth-averaged unsteady RANS simulation of resonant shallow flows in lateral cavities using augmented WENO-ADER schemes. *J. Com. Phy.*, 395:511–536, 2019. ISSN 0021-9991. doi: https://doi.org/

10.1016/j.jcp.2019.06.037. URL https://www.sciencedirect.com/science/article/pii/S0021999119304450.

[132] A. Navas-Montilla, P. Solán-Fustero, J. Murillo, and P. García-Navarro. Discontinuous Galerkin well-balanced schemes using augmented Riemann solvers with application to the shallow water equations. *J. Hydroinformatics*, 22(5):1038–1058, 04 2020. ISSN 1464-7141. doi: 10.2166/hydro.2020.206. URL https://doi.org/10.2166/hydro.2020.206.

[133] A. Navas-Montilla, J. Guallart, P. Solán-Fustero, and P. García-Navarro. Exploring the potential of TENO and WENO schemes for simulating under-resolved turbulent flows in the atmosphere using Euler equations. *Comp. Flu.*, 280:106349, 2024. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2024.106349. URL https://www.sciencedirect.com/science/article/pii/S0045793024001816.

[134] N.C. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46:157–185, 2009.

[135] S. Noelle, N. Pankratz, G. Puppo, and J.R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *J. Comp. Phys.*, 213(2):474–499, 2006. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2005.08.019. URL https://www.sciencedirect.com/science/article/pii/S002199910500389X.

[136] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455–462, 1980. doi: 10.2514/3.50778. URL https://doi.org/10.2514/3.50778.

[137] J. Novo and S. Rubino. Error Analysis of Proper Orthogonal Decomposition Stabilized Methods for Incompressible Flows. *SIAM J. Num. Ana.*, 59(1):334–369, 2021. doi: 10.1137/20M1341866. URL https://doi.org/10.1137/20M1341866.

[138] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathematique*, 351(23): 901–906, 2013. ISSN 1631-073X. doi: https://doi.org/10.1016/j.crma.2013.10.028. URL https://www.sciencedirect.com/science/article/pii/S1631073X13002847.

[139] M. Ohlberger and S. Rave. Reduced Basis Methods: Success, Limitations and Future Challenges. *Proceedings of the Conference Algoritmy*, pages 1–12, 2016. URL http://www.iam.fmph.uniba.sk/amuc/ojs/index.php/algoritmy/article/view/389.

[140] G. Padula, M. Girfoglio, and G. Rozza. A brief review of Reduced Order Models using intrusive and non-intrusive techniques, 2024. URL https://arxiv.org/abs/2406.00559.

[141] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric Model Order Reduction by Matrix Interpolation. *at - Automatisierungstechnik*, 58(8):475–484, 2010. doi: doi: 10.1524/auto.2010.0863. URL https://doi.org/10.1524/auto.2010.0863.

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

[143] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720.

[144] N. Peters, A. Wissink, and J. Ekaterinaris. Machine learning-based surrogate modeling approaches for fixed-wing store separation. *Aeros. Sci. Tech.*, 133:108150, 2023. ISSN 1270-9638. doi: https://doi.org/10.1016/j.ast.2023.108150. URL https://www.sciencedirect.com/science/article/pii/S1270963823000470.

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

[146] D.C. Pham, G. Mercère, R. Ouvrard, and T. Poinot. Heat equation parameter estimation based on the POD-Galerkin approach. *IFAC-PapersOnLine*, 51(15):245–250, 2018. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2018.09.142. URL https://www.sciencedirect.com/science/article/pii/S2405896318318032. 18th IFAC Symposium on System Identification SYSID 2018.

[147] R.H. Pletcher, J.C. Tannehill, and D. Anderson. *Computational Fluid Mechanics and Heat Transfer, Second Edition*. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis, 1997. ISBN 9781560320463. URL https://books.google.es/books?id=ZJPbtHeilCgC.

[148] C. Prud'Homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Eng.*, 124(1):70–80, November 2001. doi: 10.1115/1.1448332. URL https://hal.archives-ouvertes.fr/hal-00798326.

[149] A. Quarteroni, G. Rozza, L. Dedè, and A. Quaini. Numerical approximation of a control problem for advection-diffusion processes. In F. Ceragioli, A. Dontchev, H. Futura, K. Marti, and L. Pandolfi, editors, *System Modeling and Optimization*, pages 261–273, Boston, MA, 2006. Springer US. ISBN 978-0-387-33006-8.

[150] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: An introduction*, volume 92 of *Unitext*. Springer, Cham, 2016. ISBN 978-3-319-15430-5; 978-3-319-15431-2. doi: 10.1007/978-3-319-15431-2. URL https://doi.org/10.1007/978-3-319-15431-2.

[151] J. Radon. On the determination of functions from their integral values along certain manifolds. *IEEE Trans. Med. Imaging*, 5(4):170–176, 1986. doi: 10.1109/TMI.1986.4307775.

[152] W. H. Reed and T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1973. URL https://www.osti.gov/servlets/purl/4491151.

[153] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann. The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena. *SIAM J. Sci. Comp.*, 40, 12 2015. doi: 10.1137/17M1140571.

[154] D. Rim. Dimensional splitting of hyperbolic partial differential equations using the Radon transform. *SIAM J. Sci. Comput.*, 40(6):A4184–A4207, 2018. doi: 10.1137/17M1135633. URL https://doi.org/10.1137/17M1135633.

[155] D. Rim, R. Baraldi, C.M. Liu, R.J. LeVeque, and K. Terada. Tsunami Early Warning From Global Navigation Satellite System Data Using Convolutional Neural Networks. *Geophysical Research Letters*, 49(20):e2022GL099511, 2022. doi: https://doi.org/10.1029/2022GL099511. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022GL099511. e2022GL099511 2022GL099511.

[156] D. Rim, B. Peherstorfer, and K.T. Mandli. Manifold Approximations via Transported Subspaces: Model Reduction for Transport-Dominated Problems. *SIAM J. Sci. Comp.*, 45(1):A170–A199, 2023. doi: 10.1137/20M1316998. URL https://doi.org/10.1137/20M1316998.

[157] A. W. Rizzi and M. Inouye. Time-Split Finite-Volume Method for Three-Dimensional Blunt-Body Flow. *AIAA Journal*, 11(11):1478–1485, November 1973. doi: 10.2514/3.50614.

[158] R. Rodríguez-Sánchez, M. Buchschmid, and G. Müller. Model order reduction in design of parameterized structures under different load configurations. *Pro. Eng.*, 199:378–383, 2017. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2017.09.049. URL https://www.sciencedirect.com/science/article/pii/S1877705817334793. X International Conference on Structural Dynamics, EURODYN 2017.

[159] D. V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA J. Numer. Anal.*, 26(3):423–445, 2006. doi: 10.1093/imanum/dri044.

[160] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bif. Chaos*, 15(03):997–1013, 2005. doi: 10.1142/S0218127405012429. URL https://doi.org/10.1142/S0218127405012429.

[161] S. Rubino. A streamline derivative POD-ROM for advection-diffusion-reaction equations. *ESAIM: ProcS*, 64:121–136, 2018. doi: 10.1051/proc/201864121. URL https://doi.org/10.1051/proc/201864121.

[162] S. Rubino. Numerical Analysis of a Projection-Based Stabilized POD-ROM for Incompressible Flows. *SIAM J. Numer. Anal.*, 58(4):2019–2058, jan 2020. ISSN 0036-1429. doi: 10.1137/19M1276686. URL https://doi.org/10.1137/19M1276686.

[163] F. Sabetghadam and A. Jafarpour. $\alpha$ regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *App Math. Comp.*, 218(10):6012–6026, 2012. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2011.11.083. URL https://www.sciencedirect.com/science/article/pii/S0096300311014299.

[164] O. San and J. Borggaard. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. *Int. J. Num. Met. Flu.*, 78(1):37–62, 2015. doi: https://doi.org/10.1002/fld.4006. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4006.

[165] O. San and T. Iliescu. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. *ArXiv*, 1308.3276, 2013.

[166] A. Sanfilippo, I. Moore, F. Ballarin, and T. Iliescu. Approximate deconvolution Leray reduced order model for convection-dominated flows. *FE Ana. Des.*, 226:104021, 2023. ISSN 0168-874X. doi: https://doi.org/10.1016/j.finel.2023.104021. URL https://www.sciencedirect.com/science/article/pii/S0168874X23001142.

[167] P.J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010. doi: 10.1017/S0022112010001217.

[168] G. Serre, P. Lafon, X. Gloerfelt, and C. Bailly. Reliable reduced-order models for time-dependent linearized Euler equations. *J. Comp. Phys.*, 231(15):5176–5194, 2012. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2012.04.019. URL https://www.sciencedirect.com/science/article/pii/S0021999112002008.

[169] N.V. Shah, M.W. Hess, and G. Rozza. Discontinuous Galerkin Model Order Reduction of Geometrically Parametrized Stokes Equation. In Fred J. Vermolen and Cornelis Vuik, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2019*, pages 551–561, Cham, 2021. Springer International Publishing. ISBN 978-3-030-55874-1.

[170] A. Shmyrov, A. Mizev, V. Demin, M. Petukhov, and D. Bratsun. On the extent of surface stagnation produced jointly by insoluble surfactant and thermocapillary flow. *Adv. Col. Int. Sci.*, 255:10–17, 2018. ISSN 0001-8686. doi: https://doi.org/10.1016/j.cis.2017.08.010. URL https://www.sciencedirect.com/science/article/pii/S0001868617300155. Multiscale Applications of Surface Tension and Capillary Forces.

[171] L. Sirovich. Turbulence and the dynamics of coherent structures. I - Coherent structures. II - Symmetries and transformations. III - Dynamics and scaling. *Q. Appl. Math.*, 45: 561–571,573–582, 583–590, 1987. doi: 10.1090/qam/910463.

[172] P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro. A POD-based ROM strategy for the prediction in time of advection-dominated problems. *J. Comput. Phys.*, 471:111672, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111672. URL https://www.sciencedirect.com/science/article/pii/S0021999122007355.

[173] P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro. A POD-based reduced order model applied to 1D shallow water equations. In *Proceedings of the 14th International Conference on Hydroinformatics - HIC 2022 (Bucharest, 2022)*, Bucharest, Romania, 2022. IOP Conference Series: Earth and Environmental Science.

[174] P. Solán-Fustero, A. Navas-Montilla, J.L. Gracia, and P. García-Navarro. A 1D shallow water reduced-order model based on POD. In *Proceedings of the 39th IAHR World Congress (Granada, 2022)*, Granada, Spain, 2022. International Association for Hydro-Environment Engineering and Research (IAHR).

[175] P. Solán-Fustero, A. Navas-Montilla, E. Ferrer, J. Manzanero, and P. García-Navarro. Application of approximate dispersion-diffusion analyses to under-resolved Burgers turbulence using high resolution WENO and UWC schemes. *J. Comp. Phys.*, 435: 110246, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110246. URL https://www.sciencedirect.com/science/article/pii/S0021999121001418.

[176] P. Solán-Fustero, J.L. Gracia, A. Navas-Montilla, and P. García-Navarro. Development of POD-based Reduced Order Models applied to shallow water equations using augmented Riemann solvers. *Comp. Met. App. Mec. Eng.*, 410:116038, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2023.116038. URL https://www.sciencedirect.com/science/article/pii/S0045782523001627.

[177] G. Stabile and G. Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids*, 173: 273–284, 2018. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2018.01.035. URL https://www.sciencedirect.com/science/article/pii/S0045793018300422.

[178] R. Ştefănescu and I.M. Navon. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. *J. of Comp. Phys.*, 237:95–114, 2013. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2012.11.035. URL https://www.sciencedirect.com/science/article/pii/S0021999112007152.

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

[180] T. Taddei. A Registration Method for Model Order Reduction: Data Compression and Geometry Reduction. *SIAM J. Sci. Comp.*, 42(2):A997–A1027, 2020. doi: 10.1137/19M1271270. URL https://doi.org/10.1137/19M1271270.

[181] T. Taddei and L. Zhang. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs. *ESAIM: M2AN*, 55(1):99–130, 2021. doi: 10.1051/m2an/2020073. URL https://doi.org/10.1051/m2an/2020073.

[182] T. Taddei, S. Perotto, and A. Quarteroni. Reduced basis techniques for nonlinear conservation laws. *ESAIM Math. Model. Numer. Anal.*, 49(3):787–814, 2015. ISSN 0764-583X. doi: 10.1051/m2an/2014054. URL https://doi.org/10.1051/m2an/2014054.

[183] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. 2006. URL https://api.semanticscholar.org/CorpusID:14893043.

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

[185] E.F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Sho. Wav.*, 4:25–34, 1994.

[186] A. Towne, O. T. Schmidt, and T. Colonius. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J. Fluid Mech.*, 847:821–867, 2018. doi: 10.1017/jfm.2018.283.

[187] W. Tsai and D.K.P. Yue. Computation of Nonlinear Free-Surface Flows. *Annual Review of Fluid Mechanics*, 28(Volume 28, 1996):249–278, 1996. ISSN 1545-4479. doi: https://doi.org/10.1146/annurev.fl.28.010196.001341. URL https://www.annualreviews.org/content/journals/10.1146/annurev.fl.28.010196.001341.

[188] P. Vallés, I. Echeverribar, J. Mairal, S. Martínez-Aranda, J. Fernández-Pato, and P. García-Navarro. 2D numerical simulation of floods in Ebro River and analysis of strategies to model the Mequinenza Reservoir. *Geohazards*, 1:1–23, 2023. URL https://www.mdpi.com/2624-795X/4/2/9.

[189] J.M. Vega and S. Le Clainche. *Higher Order Dynamic Mode Decomposition and its Applications*. Academic, 2020. ISBN 0128197439 9780128197431. doi: 10.1016/C2019-0-00038-6. URL https://www.sciencedirect.com/book/9780128197431/higher-order-dynamic-mode-decomposition-and-its-applications.

[190] D. Violeau and R. Issa. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. *Int. J. Num. Met. Flu.*, 53(2):277–304, 2007. doi: https://doi.org/10.1002/fld.1292. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1292.

[191] J. Weiss. *A Tutorial on the Proper Orthogonal Decomposition*. doi: 10.2514/6.2019-3333. URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-3333.

[192] K. Willcox and J. Peraire. Balanced Model Reduction via the Proper Orthogonal Decomposition. *AIAA Journal*, 40(11):2323–2330, 2002. doi: 10.2514/2.1570. URL https://doi.org/10.2514/2.1570.

[193] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, and A. Muggeridge. Non-intrusive reduced order modelling of the Navier–Stokes equations. *Comp. Met. App. Mec. Eng.*, 293:522 – 541, 2015. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2015.05.015. URL http://www.sciencedirect.com/science/article/pii/S0045782515001887.

[194] D. Xiao, F. Fang, C.C. Pain, and I.M. Navon. Towards non-intrusive reduced order 3D free surface flow modelling. *Ocean Engineering*, 140:155–168, 2017. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2017.05.020. URL https://www.sciencedirect.com/science/article/pii/S0029801817302731.

[195] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Numerical analysis of the Leray reduced order model. *J. Comp. App. Math.*, 328:12–29, 2018. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2017.06.026. URL https://www.sciencedirect.com/science/article/pii/S0377042717303266.

[196] Y. Xing. Chapter 13 - Numerical Methods for the Nonlinear Shallow Water Equations. In Rémi Abgrall and Chi-Wang Shu, editors, *Handbook of Numerical Methods for Hyperbolic Problems*, volume 18 of *Handbook of Numerical Analysis*, pages 361–384. Elsevier, 2017. doi: https://doi.org/10.1016/bs.hna.2016.09.003. URL https://www.sciencedirect.com/science/article/pii/S1570865916300126.

[197] E.J. Yoo. *Parametric model order reduction for structural analysis and control*. PhD thesis, Technische Universität München, 2010. URL https://mediatum.ub.tum.de/997277.

[198] J. Yu, L. Lu, X. Meng, and G.E. Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Comp. Met. App. Mec. Eng.*, 393:114823, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114823. URL https://www.sciencedirect.com/science/article/pii/S0045782522001438.

[199] J.M. Zokagoa and A. Soulaïmani. Modeling of wetting–drying transitions in free surface flows over complex topographies. *Comp. Meth. App. Mech. Eng.*, 199(33):2281–2304, 2010. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2010.03.023. URL https://www.sciencedirect.com/science/article/pii/S0045782510001003.

[200] J.M. Zokagoa and A. Soulaïmani. A POD-based reduced-order model for free surface shallow water flows over real bathymetries for Monte-Carlo-type applications. *Comp. Met. App. Mec. Eng.*, 221-222:1–23, 2012. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2011.11.012. URL https://www.sciencedirect.com/science/article/pii/S0045782511003586.

[201] J.M. Zokagoa and A. Soulaïmani. Low-order modelling of shallow water equations for sensitivity analysis using proper orthogonal decomposition. *Int. J. Comp. Flu. Dyn.*, 26(5):275–295, 2012. doi: 10.1080/10618562.2012.715153. URL https://doi.org/10.1080/10618562.2012.715153.

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