Calibration-Based ALE Model Order Reduction for Hyperbolic Problems with Self-Similar Travelling Discontinuities

Monica Nonino[1] · Davide Torlo[2]

Abstract

We propose a novel Model Order Reduction framework that is able to handle solutions of hyperbolic problems characterized by multiple travelling discontinuities. By means of an optimization based approach, we introduce suitable calibration maps that allow us to transform the original solution manifold into a lower dimensional one. The novelty of the methodology is represented by the fact that the optimization process does not require the knowledge of the discontinuities location. The optimization can be carried out simply by choosing some reference control points, thus avoiding the use of some implicit shock tracking techniques, which would translate into an increased computational effort during the offline phase. In the online phase, we rely on a non-intrusive approach, where the coefficients of the projection of the reduced order solution onto the reduced space are recovered by means of an Artificial Neural Network. To validate the methodology, we present numerical results for the 1D Sod shock tube problem, for the 2D double Mach reflection problem, also in the parametric case, and for the triple point problem.

List of Symbols

Ω	The physical domain
$\mathcal{R}$	The reference domain
d	The dimension of the physical domain
t	The time variable

M. Nonino, D. Torlo: These authors contributed equally to this work.

✉ Monica Nonino
monica.nonino@univie.ac.at

Davide Torlo
davide.torlo@uniroma1.it

1   Department of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, Vienna 1090, Austria

2   Department of Mathematics "Guido Castelnuovo", University of Rome La Sapienza, Piazzale Aldo Moro 5, Roma 00185, Italy

μ	The physical parameter(s)
$\boldsymbol{\mu}$	$\boldsymbol{\mu} = (t, \mu)$ (parametric case) or $\boldsymbol{\mu} = t$ (non-parametric case)
$\mathcal{P}_{\text{phys}}$	The domain of μ the physical parameter
$\mathcal{P}$	The domain of $\boldsymbol{\mu}$ the time and physical parameters
M	The number of control points
$\boldsymbol{w}(\boldsymbol{\mu})$	The vector of M control points in Ω
$\overline{\boldsymbol{w}}(\boldsymbol{\mu})$	The vector M reference control points in $\mathcal{R}$
$\boldsymbol{\alpha}$	The multiindex $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$
$w_{\boldsymbol{\alpha}}^k$	The k-th coordinate of the control point $\boldsymbol{w}(\boldsymbol{\mu})_{\boldsymbol{\alpha}}$
$\boldsymbol{\theta}(\boldsymbol{\mu})$	The vector of the free coordinates of the control point $\boldsymbol{w}(\boldsymbol{\mu})$
$\overrightarrow{\theta}$	The matrix of the free coordinates. $\overrightarrow{\theta} \in \mathbb{R}^{N_{few}} \times Q$, and $\overrightarrow{\theta}[i,:] = \boldsymbol{\theta}(\tilde{\boldsymbol{\mu}}_i)$, $i = 1, \ldots, N_{few}$.
$T[\cdot]$	The family of geometrical transformation maps. $T[\cdot] \in \mathcal{C}^1(\Omega^M, \mathcal{C}^1(\mathcal{R}, \Omega))$.
$\overline{\rho}(\cdot)$	The reference solution for the calibration in the self-similar setting
$\hat{\rho}(\cdot)$	The calibrated snapshot
$N_{\text{few}}^{\text{POD}}$	The dimension of the linear space V_{POD} used for the calibration in the quasi-self-similar setting
V_{POD}	The linear space considered in the residual function to perform the projection error for the minimization in the quasi-self-similar case, instead of the use of the reference solution
$\mathcal{M}_\rho$	The original solution manifold (for the density ρ)
$\hat{\mathcal{M}}_\rho$	The calibrated solution manifold (for the density ρ)
$\{\hat{\Phi}_i\}_{i=1}^N$	The set of N reduced basis functions obtained by POD-compression of $\hat{\mathcal{M}}_\rho$
$\underline{\hat{\rho}}_N(\boldsymbol{\mu})$	The vector of the L^2-projection coefficients of the calibrated snapshot $\hat{\rho}(\boldsymbol{\mu})$ onto the linear space spanned by $\{\hat{\Phi}_i\}_{i=1}^N$

1 Introduction

The goal of MOR techniques [1–3], which are particularly suited for the real-time computations and many-query context, is to obtain efficient and reliable approximations of solutions of high dimensional systems of partial differential equations (PDEs). Let us consider the approximate solution $u(t; \mu) \in L^2(\Omega)$ of a parametrized PDE, with $\Omega \subset \mathbb{R}^d$, with the parameter $\mu \in \mathcal{P}_{\text{phys}} \subset \mathbb{R}^s$ and with time $t \in [0, t_f]$: for the spatial discretization one can consider, for instance, the Finite Volume (FV) discretization. We introduce the *solution manifold* related to this parametric PDE: $\mathcal{M} = \{u(t; \mu) \in V_{\mathcal{N}}, \mu \in \mathcal{P}_{\text{phys}}, t \in [0, t_f]\}$, where $V_{\mathcal{N}}$ is a suitable functional space defined by the chosen spatial discretization. The key idea behind MOR is to represent $\mathcal{M}$ with a finite dimensional linear space V_N, such that $N \ll \mathcal{N}$, where $N = \dim V_N$ and $\mathcal{N} = \dim V_{\mathcal{N}}$. To find the lower dimensional space V_N, one can use the well known Proper Orthogonal Decomposition (POD) strategy that, given in input a set of discrete solutions (obtained, for example, with the FV method), is able to extract a set of small cardinality N, which contains the so-called reduced basis functions that best approximate the manifold. A pivotal aspect for the efficiency of the MOR is the ability of the POD of compressing the discrete solution manifold $\mathcal{M}$: this concept is strictly related to the definition of Kolmogorov N-width of $\mathcal{M}$ and, ultimately, to the reducibility of the problem of interest.

The *Kolmogorov N-width d_N* of $\mathcal{M}$ is defined as

$$d_N(\mathcal{M}, V_{\mathcal{N}}) = \inf_{\substack{V_N \subset V_{\mathcal{N}} \\ \dim V_N = N}} \sup_{f \in \mathcal{M}} \inf_{g \in V_N} ||f - g||, \tag{1}$$

where $||\cdot||$ is a suitable norm in $V_{\mathcal{N}}$. Definition (1) describes in a rigorous mathematical setting the capability of finite dimensional linear subspaces $V_N \subseteq V$ of reproducing any element in $\mathcal{M}$, that is, any discrete solution of the problem of interest. Therefore, the faster d_N decays the more efficient a linear MOR will be for such a problem, as N grows. Some rigorous bounds for d_N, for particular classes of problems, are available in literature [4, 5]; as an alternative, one can look at the rate of decay of the eigenvalues λ_k returned by the POD on $\mathcal{M}$.

Despite the capability of standard MOR techniques to handle a vast number of applications, problems advecting local structures still represent a challenge for the MOR community. Indeed, for such problems the decay of the Kolmogorov N-width is slow, see for example [6]. As a result, standard MOR struggles to suitably reproduce steep features, such as solutions with (multiple) travelling shocks. For this reason, in the last decade a great number of works appeared in the literature, offering numerous approaches to deal with advection dominated problems. We mention the method of freezing [7], the shifted POD [8] (also in combination with the use of neural networks [9]), the generation of advection modes by means of an optimal mass transport problem [10–13], L^1 minimization [14], the calibration method introduced in [15, 16], Lagrangian based MOR techniques [17, 18], the preprocessing of the snapshots used in [19, 20], the registration method [21–23], adaptive basis methods [24], implicit feature tracking [25] and displacement interpolation [26, 27]. Next to these more classical techniques, some nonlinear approaches have been lately studied starting from convolutional autoencoders neural network based approaches for learning the solution manifold [28, 29], passing through graph neural networks autoencoders [30] to graph neural network to perform the limit to vanishing viscosity [31] and entropy-stable rational quadratic manifolds [32]. Motivated by the interest that MOR for transport dominated problems sparks in the applied mathematics community, the goal of this work is to propose a calibration based reduced order algorithm that can be used to gain significant speedup in the simulations of solutions of hyperbolic PDEs. The proposed methodology will avoid the use of implicit feature tracking [25], with the goal of not increasing the computational costs of the offline phase that is performed explicitly. The framework is very similar to the calibration/registration approach proposed in [15, 21, 22] with novelties in the calibration process and in the range of applicability of the method. Indeed, our methodology has a more physical intuition as it is based on the interpolation of some calibration points rather than on more abstract interpolation functions. Moreover, the proposed applications range on a novel paradigm, where parametric time-dependent solutions widely vary, with large deformations of the solutions structures. To the knowledge of the authors, this is the first time that solutions with such a varying behavior were successfully tackled with a model order reduction technique.

In this work, we will focus on time-dependent hyperbolic problems, whose solutions are (quasi) self-similar: the formal definition of (quasi) self-similar solution will be given in Sect. 1.1. In particular, we want to study problems where multiple structures travel along the domain with different speeds. This is typical for hyperbolic problems, where shocks, rarefactions and other discontinuities are generated and travel along the domain. The novelties of this work, in comparison to the state of the art [22], lie in two key aspects. First, our optimization process operates independently of the solution structure, eliminating the need for shock detectors or similar tools. Secondly, our method demonstrates a broader range of

applicability, encompassing problems featuring multiple shocks whose positions undergo significant variation, sometimes nearly colliding with each other.

The rest of the manuscript is structured as follows. In Sect. 1.1, we introduce the problems of interest and the definition of self-similarity. In Sect. 2, we define the calibration procedure and the optimization algorithms. In Sect. 3, we present some geometrical transformations that interpolate the calibrated points and allow to define the original problem onto a reference domain where all the structures do not move. In Sect. 4, we describe the combination of classical MOR techniques with the calibration process. In Sect. 5, we show the good performances of the proposed reduced order model (ROM) onto one and two dimensional parametric time–dependent problems and, in Sect. 6, we draw some conclusions. To improve the readability of the manuscript, a table with the list of all the mathematical symbols used is provided at the end of the paper.

1.1 Motivation

We begin by introducing the problem of interest that we will tackle in this manuscript: we focus here on hyperbolic time–dependent conservation laws. As an example, we turn our attention to Euler equations, but the same framework can be applied to other conservation and balance laws. Let $\Omega \subset \mathbb{R}^d$, $d \geqslant 1$, be our physical domain. We restrict ourselves to rectangular domains of the type $\Omega = [a^1, b^1] \times \cdots \times [a^d, b^d]$, with $a^i, b^i \in \mathbb{R}$ for $i = 1, \ldots, d$. The generalization for more complex domains can be performed as in [33]. Let $[0, t_f] \subset \mathbb{R}$ be the time span of the problem and let $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{s+1}$, $s \geqslant 0$, be the collection of all parameters (including time). From now on, we will assume that $s = 0$ in the non parametric regime, i.e., $\boldsymbol{\mu} = t$ and $\mathcal{P} = [0, t_f]$, or $s \geqslant 1$ in the parametric regime, i.e., $\boldsymbol{\mu} = (\mu, t)$ and $\mathcal{P} = \mathcal{P}_{\text{phys}} \times [0, t_f]$. The parameteric Euler equations of gas dynamics, in conservative form, read as follows: find the density $\rho \colon \mathcal{P} \times \Omega \mapsto \mathbb{R}$, the momentum $\boldsymbol{m} \colon \mathcal{P} \times \Omega \mapsto \mathbb{R}^d$ and the total energy $E \colon \mathcal{P} \times \Omega \mapsto \mathbb{R}$ such that

$$\begin{cases} \partial_t \rho + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{m} = 0 & \text{in } \mathcal{P} \times \Omega, \\ \partial_t \boldsymbol{m} + \nabla_{\boldsymbol{x}} \cdot \left(\frac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho} + p\, I \right) = 0 & \text{in } \mathcal{P} \times \Omega, \\ \partial_t E + \nabla_{\boldsymbol{x}} \cdot \left(\frac{\boldsymbol{m}}{\rho}(E + p) \right) = 0 & \text{in } \mathcal{P} \times \Omega, \end{cases} \tag{2}$$

where $\nabla_{\boldsymbol{x}} \cdot$ is the divergence with respect to $\boldsymbol{x} \in \Omega$, $I \in \mathbb{R}^{d \times d}$ is the identity matrix and the pressure p is defined through the following equation of state $p = (\gamma - 1)(E - 0.5|\boldsymbol{m}|^2/\rho)$, with $\gamma = 1.4$ being the adiabatic constant. System (2) is then completed by some proper initial conditions (IC) and boundary conditions (BC). We will consider as IC some Riemann problems both in one and two dimensional problems: the Sod shock tube problem [34], the 2D double Mach reflection problem [35] and the triple point problem. In these examples, the solution of (2) turns out to be self-similar, with features as shocks, contact discontinuities and rarefaction waves traveling in the physical domain.

Definition 1 Let $\mathcal{R} \subset \mathbb{R}^d$ be a reference domain, which is time (and parameter) independent. We call **self–similar** a solution manifold $\mathcal{M}$ for which there exists a reference solution $\bar{u} \colon \mathcal{R} \to \mathbb{R}^d$ and a transformation $T^{-1}[\boldsymbol{\mu}](\cdot) \colon \Omega \to \mathcal{R}$ such that we have $u(\boldsymbol{\mu})(T^{-1}[\boldsymbol{\mu}](\boldsymbol{x})) \approx \bar{u}(\boldsymbol{x})$ for all $\boldsymbol{\mu} \in \mathcal{P}$, $\forall \boldsymbol{x} \in \mathcal{R}$. When this condition is not satisfied, but still all solutions in $\mathcal{M} = \{u(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{P}}$ present the same features, with different values of the solution in between these features, we will call such manifold **quasi–self–similar**. More precisely, $\mathcal{M}$ is **quasi–self–similar** if there exists a transformation T such that the transformed solution manifold $\hat{\mathcal{M}} := \{u(\boldsymbol{\mu})(T^{-1}[\boldsymbol{\mu}](\cdot))\}$ has a fast decay of the Kolmogorov N-width.
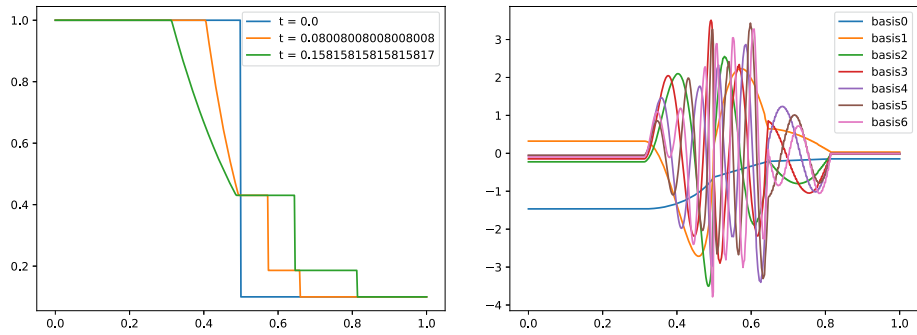
Fig. 1 The density ρ, solution of the Sod shock tube problem at different timesteps, and the corresponding POD modes (right)

We start by considering a simple 1D Sod shock tube problem, in the non–parametric regime. Here, $\Omega = [0, 1]$, and the following initial data is considered:

$$\begin{bmatrix} \rho \ u \ p \end{bmatrix} (t = 0) = \begin{cases} \begin{bmatrix} 1 \ 0 \ 1 \end{bmatrix}^T & \text{if } x < 0.5, \\ \begin{bmatrix} 0.1 \ 0 \ 0.125 \end{bmatrix}^T & \text{if } x > 0.5, \end{cases}$$

where $u \colon \Omega \mapsto \mathbb{R}^d$ is the velocity. The initial conditions in conservative variables m and E can be derived using $m = \rho u$ and $E = p/(\gamma - 1) + 1/2|m|^2/\rho$. Figure 1 shows the density $\rho(\boldsymbol{\mu})$ for the Sod shock tube problem (left), and the corresponding modes (right), obtained by running a POD on the solution manifold $\mathcal{M}_\rho$: the solution of the Sod shock tube problem is exact, and its analytical expression has been taken from [34]. The density presents a shock, a contact discontinuity and a rarefaction wave that travel in the domain: as a consequence, the POD modes exhibit an highly oscillatory behavior, struggling to correctly capture the position of the moving features. Still, the solutions are **self-similar** as we just need to transport each feature onto reference positions to make the solutions essentially linearly dependent. Indeed, as observed in [11], the optimal transport for the density of the Sod 1D problem would lead to the exact solution, without the need for further ROM techniques. Nevertheless, we are interested also in higher dimensional problems with more complicated discontinuity structures. Hence, we will not proceed in the optimal transport direction.

2 Calibration of the Snapshots

We now present the calibration technique that we use to align the different features of our snapshots, to obtain a solution manifold with a faster decaying Kolmogorov N-width. The key of the proposed calibration is that it can be used to align different travelling features (shocks, contact discontinuities, rarefaction waves), without the need to know explicitly the exact location of these features, as opposed to, for example, what is assumed in [20, 22]. Moreover, we assume to calibrate the density ρ of the Euler system (2), other scalar quantities depending on the system unknowns, e.g., the entropy, can be equivalently used.

Let $\mathcal{R}$ be the *reference domain*, similarly to what is used in the Arbitrary Lagrangian Eulerian (ALE) formalism [36], and let Ω be the *physical domain*. For every $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{s+1}$, we introduce a grid of $M = \prod_{i=1}^d M_i$ *control points* that are collected in the vector $\boldsymbol{w}(\boldsymbol{\mu}) \in \Omega^M$ that we use for the calibration. These control points should lead the transformation to align the different features at different $\boldsymbol{\mu}$. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ be a multi-index with
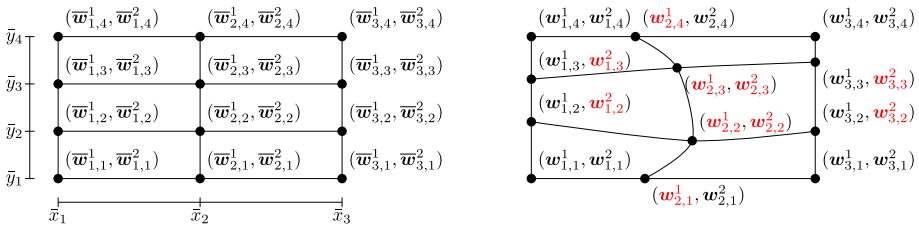
Fig. 2 Example of a control point grid in 2D with $M_1 = 3$ and $M_2 = 4$ on the reference domain (left) and on the physical domain (right). Note that the coordinates of the reference control points are the tensor product of unidirectional control points. We highlight in red the free coordinates of these control points

$\alpha_i = 1, \ldots, M_i$. We consider the control point $\boldsymbol{w}_\alpha$ to be tensor product of points in the interval $[a^i, b^i]$ for every dimension $i = 1, \ldots, d$, for example in 2D: $\boldsymbol{w}_{\alpha=(\alpha_1, \alpha_2)} = (\boldsymbol{w}^1_{\alpha_1, \alpha_2}, \boldsymbol{w}^2_{\alpha_1, \alpha_2})$, see Fig. 2. Each control point $\boldsymbol{w}(\boldsymbol{\mu})_\alpha$ can either belong to the physical domain, i.e., $\boldsymbol{w}(\boldsymbol{\mu})_\alpha \in \mathring{\Omega}$, or to the boundary of the domain, namely $\boldsymbol{w}(\boldsymbol{\mu})_\alpha \in \partial\Omega$. If $\boldsymbol{w}(\boldsymbol{\mu})_\alpha \in \partial\Omega$, then this point is constrained for all $\boldsymbol{\mu}$ to the boundary hyperplanes where it belongs: this, in turn, sets a constraint on one (or both) of the coordinates $(\boldsymbol{w}^1_{\alpha_1, \alpha_2}, \boldsymbol{w}^2_{\alpha_1, \alpha_2})$. The constrained coordinates are, hence, not to be optimized and we can discard them in the set of coordinates that we will optimize. Motivated by this, we introduce $\boldsymbol{\theta}(\boldsymbol{w}(\boldsymbol{\mu})) \in \mathbb{R}^Q$, that is the row vector of the *Q free coordinates* of the control points, with $Q \leqslant d \times M$. This vector represents all the coordinates of the control points that are free to move during the calibration. In Fig. 2, the free coordinates of the control points are highlighted in red. We remark that there is a bijection between $\boldsymbol{w}$ and $\boldsymbol{\theta}$, by definition.

In order to align different features of our set of snapshots, we look for a geometrical transformation map $T \colon \Omega^M \times \mathcal{R} \mapsto \Omega$, such that the following properties hold true:

- $T[\cdot] \in \mathcal{C}^1(\Omega^M, \mathcal{C}^1(\mathcal{R}, \Omega))$;
- $\forall \boldsymbol{\mu} \in \mathcal{P}$ and $\forall \boldsymbol{w}(\boldsymbol{\mu}) \in \Omega^M, \exists T^{-1}[\boldsymbol{w}(\boldsymbol{\mu})] \colon \Omega \mapsto \mathcal{R}$ such that

$$T^{-1}[\boldsymbol{w}(\boldsymbol{\mu})](T[\boldsymbol{w}(\boldsymbol{\mu})](\hat{\boldsymbol{x}})) = \hat{\boldsymbol{x}} \quad \forall \hat{\boldsymbol{x}} \in \mathcal{R},$$

$$T[\boldsymbol{w}(\boldsymbol{\mu})](T^{-1}[\boldsymbol{w}(\boldsymbol{\mu})](\boldsymbol{x})) = \boldsymbol{x}, \text{ for all } \boldsymbol{x} \in \Omega;$$

- $T^{-1}[\cdot] \in \mathcal{C}^1(\Omega^M, \mathcal{C}^1(\Omega; \mathcal{R}))$.

The properties are imposed to setup an ALE formulation [21, 36]. Some possibilities for the geometrical transformation T have been presented in the literature over the past years: among the others we mention here translation maps, dilatation maps, polynomials and Gordon-Hall maps, see [15, 21, 36]. We will use some transformations based on Piecewise Cubic Hermite Interpolation Polynomials (PCHIPs) carefully described in Sect. 3. In order to use the transformation map T, one needs to find the calibration map $\boldsymbol{w} \colon \mathcal{P} \mapsto \Omega^M$, such that:

- $\boldsymbol{w}(\cdot) \in \mathcal{C}^1(\mathcal{P}; \Omega^M)$;
- $\rho(\boldsymbol{\mu})(T[\boldsymbol{w}(\boldsymbol{\mu})](\hat{\boldsymbol{x}})) \approx \overline{\rho}(\hat{\boldsymbol{x}})$, for all $\hat{\boldsymbol{x}} \in \mathcal{R}$, $t \in [0, t_f]$ and for all $\boldsymbol{\mu} \in \mathcal{P}$, where $\overline{\rho}(\cdot)$ is a reference solution of choice.

We are now ready to present a general calibration technique: the ultimate goal is to transform the solution manifold $\mathcal{M}_\rho$ so that the POD, applied to the transformed manifold, is more efficient. In order to achieve this goal, we need to perform the following steps. First of all, we introduce the *reference* density $\overline{\rho}$, namely a solution of problem (2) for some $\overline{\boldsymbol{\mu}} \in \mathcal{P}$. Once $\overline{\rho}$ has been chosen, we select M control points $\overline{\boldsymbol{w}} \in \mathcal{R}^M$. Now, for any $\boldsymbol{\mu} \in \mathcal{P}$ and

$w(\mu) \in \Omega^M$, we can define a geometrical transformation map $T[w(\mu)]\colon \mathcal{R} \mapsto \Omega$ that maps the reference control points in the reference domain onto the (parametric) control points in the physical domain, i.e.,

$$T[w(\mu)](\overline{w}_{\alpha}) = w_{\alpha}(\mu) \quad \forall \alpha.$$

Once $T[w(\mu)]$ has been defined, we can introduce the *calibrated snapshot*, which is the pullback of a solution $\rho(\mu)$ to the reference domain, i.e.,

$$\hat{\rho}(\mu; w(\mu))(\cdot) := \rho(\mu)(T[w(\mu)](\cdot)) : \mathcal{R} \to \mathbb{R}.$$

We want to stress that, numerically, we rely on two meshes, one on the physical and one on the reference domain. In our simulations, these two domains will coincide; this does not mean that each transformation map leads to a one-to-one correspondence between the degrees of freedom on the two domains. Hence, we will perform an interpolation of ρ to evaluate $\hat{\rho}$ at its degree of freedom. In the numerical simulations, this procedure will bring an error of the first order of accuracy (as we will use FV approximations).

The map $T[w(\mu)]$ is identified by the control points $w(\mu)$, which are sought in order to minimize the following residual function:

$$\begin{aligned}
R(\rho(\mu), \theta(w(\mu)), \overline{\rho}) =&||\hat{\rho}(\mu; w(\mu)) - \overline{\rho}||^2_{L^2(\mathcal{R})} + \frac{\delta}{2} \left\| \partial_{\mu} w(\mu) \right\|^2_{\ell^2(\mathcal{P})} \\
&+ \frac{\alpha}{2} \max_{x \in \Omega} \left(\max \left\{ \left\| \nabla T[w(\mu)](\hat{x}) \right\|, \left\| \nabla T^{-1}[w(\mu)](x) \right\| \right\} \right),
\end{aligned} \quad (3)$$

where α and δ are two penalty parameters user defined, and $\partial_{\mu} w(\mu)$ will be defined more in detail in the algorithmic section. The first term is the one we aim at minimizing, while the other two are regularization terms that penalizes discontinuities in time of the calibration points and that regularize the geometrical transformations. We are now ready to present the calibration technique in two different cases: the self-similar setting, and the quasi self-similar one.

2.1 Calibration in the Self–Similar Setting

To keep the presentation as general as possible, we consider here the parametric case, hence $\mu = (\mu, t) \in \mathcal{P} = \mathcal{P}_{\text{phys}} \times [0, t_f]$. We select a training set of physical parameters $\{\mu_1, \ldots, \mu_{N_{\text{train}}}\} = \mathcal{P}_{\text{phys}}^{\text{train}} \subset \mathcal{P}_{\text{phys}}$ and a training set of times $\{t_1, \ldots, t_{N_{\mu}}\}$ for each $\mu \in \mathcal{P}_{\text{phys}}^{\text{train}}$. We then denote $\mathcal{P}^{\text{train}} := \{\mu = (\mu, t) : \mu \in \mathcal{P}_{\text{phys}}^{\text{train}} \text{ and } t \in [t_1, \ldots, t_{N_{\mu}}]\} \subset \mathcal{P}$. For each $\mu \in \mathcal{P}^{\text{train}}$, we compute the full order model solution $\rho(\mu)$. We then choose as reference solution $\overline{\rho} = \rho(\overline{\mu})$: in our numerical tests we will choose $\overline{\mu} = (\mu_{N_{\text{train}}}, t_f)$. In addition to this, we also fix the M reference control points $\overline{w} \in \mathcal{R}^M$ as specified above. Before introducing the constrained minimization problem, let us recall that there exist a bijection between $\theta(w(\mu))$ and $w(\mu)$. For this reason, in order to keep the notation light, from now on we will denote with $\theta(\mu)$ the free coordinates $\theta(w(\mu))$ of the control points $w(\mu)$.

Now, for all μ in the training set, we solve the following constrained minimization problem:

$$\theta^{\text{opt}}(\mu) := \min_{\theta \in \mathbb{R}^Q} R(\rho(\mu), \theta; \overline{\rho}), \quad (4)$$

subject to the following constraints:

- all the control points are within our physical domain: $w_{\alpha}(\mu) \in \Omega$ for all α and for all training parameters $\mu \in \mathcal{P}^{\text{train}}$;

- det $J[\boldsymbol{w}(\boldsymbol{\mu})] > 0$, where $J[\boldsymbol{w}(\boldsymbol{\mu})]$ is the Jacobian of the map $T[\boldsymbol{w}(\boldsymbol{\mu})]$. This constraint must be checked on a quite fine grid of the physical domain Ω, we have used the mesh grid;
- for $i = 1, \ldots, d$: if $\alpha_j = \beta_j$ for all $j \neq i$ and $\alpha_i < \beta_i$, then $\boldsymbol{w}_{\boldsymbol{\alpha}}^i < \boldsymbol{w}_{\boldsymbol{\beta}}^i$ (see Fig. 2), i.e., we never switch the order of the control points on each grid line.

We approximate $\partial_{\boldsymbol{\mu}}$ in (3) with the discrete derivative $D_{\boldsymbol{\mu}}\boldsymbol{w}(\boldsymbol{\mu})$ defined as:

$$
D_{\boldsymbol{\mu}}\boldsymbol{w}_{\boldsymbol{\alpha}}^i(\boldsymbol{\mu}) = \begin{cases} 0 & \text{if } \boldsymbol{w}_{\boldsymbol{\alpha}}^i(\boldsymbol{\mu}) \text{ is not a free coordinate of } \boldsymbol{w}(\boldsymbol{\mu}); \\ \frac{\boldsymbol{\theta}_q(\boldsymbol{\mu}) - \boldsymbol{\theta}_q^{\text{opt}}(\boldsymbol{\mu}_{\text{neigh}}(\boldsymbol{\mu}))}{\boldsymbol{\mu} - \boldsymbol{\mu}_{\text{neigh}}(\boldsymbol{\mu})} & \text{with } q \text{ s.t. } \boldsymbol{w}_{\boldsymbol{\alpha}}^i(\boldsymbol{\mu}) = \boldsymbol{\theta}_q(\boldsymbol{\mu}). \end{cases}
$$

In the previous equation, the *neighboring parameter* $\boldsymbol{\mu}_{\text{neigh}}$ is defined as follows:

$$
\boldsymbol{\mu}_{\text{neigh}}(\boldsymbol{\mu}, \mathcal{S}) = \arg\min_{\boldsymbol{v} \in \mathcal{S}}||\boldsymbol{\mu} - \boldsymbol{v}||_{\ell^2(\mathbb{R}^{p+1})},
$$

where $\mathcal{S} \subset \mathcal{P}^{\text{train}}$ will contain the parameters for which we have already computed the optimal $\boldsymbol{\theta}$. If the minimum is not unique, we take one of the minimizers. The definition of the discrete spatial gradient ∇T in (3) will be specified in Sect. 3 for the specific transformation map we use.

Problem (4) is solved with the Sequential Least SQuares Programming (SLSQP) method that is available within the `scipy.optimize.minimize` library. We solve Problem (4) for the physical parameter $\mu \in \mathcal{P}_{\text{phys}}^{\text{train}}$ for which the solution has more developed structures (the last in our tests): we start from the final timestep t_{N_μ} and we proceed backwards in time. We then move onto solving Problem (4) for the neighboring parameters. In both physical and temporal parameters, the rationale is the following: we proceed form the solutions where the structures that we want calibrate are more developed and we proceed with nearest parameters until we solve the problem for all the training set $\mathcal{P}_{\text{phys}}^{\text{train}}$. The initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu}_{(\ell,j)})$ for $\boldsymbol{\mu}_{(\ell,j)} = (\mu_\ell, t_j) \in \mathcal{P}^{\text{train}}$ is the optimal output of the minimization for the closest parameter already performed. In our tests it will be defined as:

$$
\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu}_{(\ell,j)}) = \begin{cases} \boldsymbol{\theta}^{\text{opt}}(\boldsymbol{\mu}_{(\ell,j+1)}) \text{ if } j = 1, \ldots, N_{\mu_\ell} - 1, \\ \boldsymbol{\theta}^{\text{opt}}(\boldsymbol{\mu}_{(\ell+1,j)}) \text{ if } j = N_{\mu_\ell}. \end{cases} \tag{5}
$$

Algorithm 1 shows the details of the procedure.

Algorithm 1 Calibration for Self–Similar Solution Manifold

1: **Input:** the reference solution $\overline{\rho}$, the control points $\overline{\boldsymbol{w}} \in \mathcal{R}^M$ and the training solution manifold on the physical domain $\{\rho(\mu_\ell, t_j) \text{ for } \ell = 1, \ldots, N_{\text{train}}, j = 1, \ldots, N_{\mu_\ell}\}$.
2: **for** $\ell = N_{\text{train}}, \ldots, 1$ **do**
3:     **for** $j = N_{\mu_\ell}, \ldots, 1$ **do**
4:         Set the initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu}_{(\ell,j)})$ as in (5).
5:         Solve Problem (4).
6:     **end for**
7: **end for**
8: **Output:** the optimal control points $\boldsymbol{w}^{\text{opt}}(\boldsymbol{\mu}_{(\ell,j)})$.

The chosen strategy is not trivially favorable with respect to other ones and it is quite arbitrary. We have performed different trials in the tests that we have carried out, and we noticed differences in the sensitivity to this choice. Overall, taking as initial guess the optimal

Fig. 3 The density $\rho(t = 0)$ for three different values of the parameter $\mu = [0.8943, 0.1075, 1.0906, 0.0572]$ (blue), $\mu = [1.0286, 0.1031, 0.7358, 0.0706]$ (orange), and $\mu = [1.2253, 0.1157, 1.1172, 0.1094]$ (green)
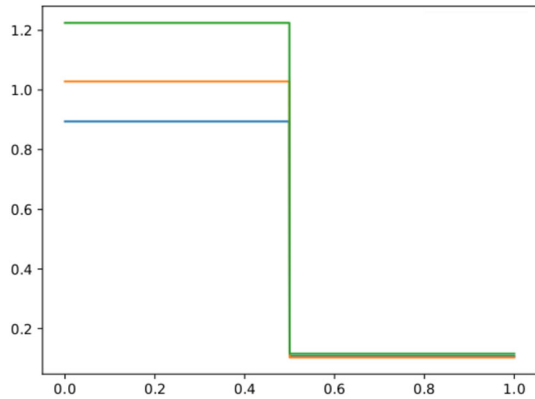
solution of a parameter close to the one we want to optimize is always a good strategy, but open questions remain on which reference configuration $\overline{\rho}$ to choose and on how spread can the parameter of the training set be.

For the 1D test cases, the change in the choice of the initial guess was less relevant than for the 2D cases. Indeed, we could use initial guesses that are quite far away from the exact one and still obtain very accurate results. For the 2D cases, the choice of the initial guess, and how spread the training space can be is more relevant. For example, the timesteps used for the calibration and, hence, the distance between two consecutive parameters and the initial guess, had a strong impact. We therefore had to calibrate on timesteps that are not too far away one from the other, in order to obtain smooth and continuous results.

For the 1D test, we perform an analytical study of this choice where the exact solution is available: the outcome of this study is presented in Sect. 5.1.1.

2.2 Calibration in the Quasi–Self–Similar Setting

To motivate the need of a different algorithm for quasi–self–similar solutions, we focus now on the parametric version of the 1D Sod shock tube problem (2). In this example, the physical parameter $\mu = (\mu^0, \dots, \mu^3) \in \mathcal{P}_{\text{phys}} \subset \mathbb{R}^4$ represents the IC for the Euler problem:

$$\begin{bmatrix} \rho & u & p \end{bmatrix}^T (t = 0; \mu) = \begin{cases} \begin{bmatrix} \mu^0 & 0 & \mu^1 \end{bmatrix}^T & \text{if } x < 0.5, \\ \begin{bmatrix} \mu^2 & 0 & \mu^3 \end{bmatrix}^T & \text{if } x > 0.5. \end{cases}$$

In Fig. 3, we show some analytical solutions for the parametric 1D Sod shock tube, for different values of μ. Looking at Fig. 3 it is clear there is no straightforward choice for the reference solution $\overline{\rho}$ that would lead to reasonable minimization problems, similar to the one presented in Algorithm 1. Indeed, we would minimize the L^2 error between two solutions ($\overline{\rho}$ and $\hat{\rho}(\boldsymbol{\mu})$) that may have very different heights at the boundaries. When different boundaries behaviors are present, the calibration procedure needs to be reformulated in a different way.

Instead of fixing a reference solution $\overline{\rho}$, we consider a *suitable* linear reduced space V_{POD}, thus introducing a new residual for the minimization problem:

$$
\min_{\boldsymbol{\theta}(\boldsymbol{\mu})\in\mathbb{R}^Q}\|\hat{\rho}(\boldsymbol{\mu};\boldsymbol{w}(\boldsymbol{\mu}))-\Pi_{V_{\text{POD}}}\hat{\rho}(\boldsymbol{\mu};\boldsymbol{w}(\boldsymbol{\mu}))\|_{L^2(\mathcal{R})}+\frac{\delta}{2}\|\partial_{\boldsymbol{\mu}}\boldsymbol{w}(\boldsymbol{\mu})\|^2_{\ell^2(\mathcal{P})}
$$
$$
+\frac{\alpha}{2}\max_{\boldsymbol{x}\in\Omega}\left(\max\left\{\left\|\nabla T[\boldsymbol{w}(\boldsymbol{\mu})](\hat{\boldsymbol{x}})\right\|,\left\|\nabla T^{-1}[\boldsymbol{w}(\boldsymbol{\mu})](\boldsymbol{x})\right\|\right\}\right),
\tag{6}
$$

where we recall that $\boldsymbol{\theta}(\boldsymbol{\mu})=\boldsymbol{\theta}(\boldsymbol{w}(\boldsymbol{\mu}))\in\mathbb{R}^Q$ is the vector of the free coordinates of the control points, and $Q\leqslant d\times M$ is the total number of free coordinates in the calibration step. In Problem (6), $\boldsymbol{\theta}$ is constrained as in the previous section and $\Pi_{V_{\text{POD}}}$ is the orthogonal projection onto a linear space V_{POD} obtained through a preliminary procedure that we will describe now shortly.

This minimization allows to overcome the issue of quasi–self–similar solutions thanks to the projection onto a reduced space. To carry out this projection, we first need to build *a priori* a suitable linear space V_{POD}, which needs to capture the *minimal* amount of information on the solution manifold: we want to be able to approximate the solution manifold with a linear space of very low dimension.

We therefore introduce the matrix of the free coordinates $\overrightarrow{\theta}\in\mathbb{R}^{N_{\text{few}}\times Q}$:

$$
\overrightarrow{\theta}[i,:]=\boldsymbol{\theta}(\tilde{\boldsymbol{\mu}}_i),\quad i=1,\dots,N_{\text{few}},
$$

where we recall once again that $\boldsymbol{\theta}(\tilde{\boldsymbol{\mu}}_i)$ is the vector of the free coordinates of the control points $\boldsymbol{w}(\tilde{\boldsymbol{\mu}}_i)$ associated to the parameter $\tilde{\boldsymbol{\mu}}_i$. The free coordinates can be selected through another optimization process carried out in the same spirit of (6) on the whole matrix $\overrightarrow{\theta}$, minimizing the projection error over all the parameters $\tilde{\boldsymbol{\mu}}_1,\dots,\tilde{\boldsymbol{\mu}}_{N_{\text{few}}}$, while updating the space $V_{\text{POD}}(\overrightarrow{\theta})$ obtained by compressing the solutions on the reference domain for these parameters transformed using the calibration points given by $\overrightarrow{\theta}$. Therefore, we solve the following constrained minimization problem:

$$
\min_{\overrightarrow{\theta}\in\mathbb{R}^{N_{\text{few}}\times Q}}\sum_{i=1}^{N_{\text{few}}}\|\hat{\rho}(\tilde{\boldsymbol{\mu}}_i;\boldsymbol{w}(\tilde{\boldsymbol{\mu}}_i))-\Pi_{V_{\text{POD}}(\overrightarrow{\theta})}\hat{\rho}(\tilde{\boldsymbol{\mu}}_i;\boldsymbol{w}(\tilde{\boldsymbol{\mu}}_i))\|_{L^2(\mathcal{R})},
\tag{7}
$$

with $V_{\text{POD}}:=\text{POD}(\{\hat{\rho}(\tilde{\boldsymbol{\mu}}_i)\}_{i=1}^{N_{\text{few}}},N_{\text{few}}^{\text{POD}})$. This optimization process is of larger dimension with respect to the previous ones and it requires, for each residual evaluation, the computation of a POD over the N_{few} calibrated snapshots $\hat{\rho}(\tilde{\boldsymbol{\mu}}_i;\boldsymbol{w}(\tilde{\boldsymbol{\mu}}_i))$, with a user defined number of modes $N_{\text{few}}^{\text{POD}}$. The choice of $N_{\text{few}}^{\text{POD}}$ is, in general, not simple. In our numerical tests, we adopted the following heuristics: we want to take into account all the different travelling features. For this reason, in the Sod test case, we have chosen $N_{\text{few}}^{\text{POD}}=3$ to keep into account the different values that there might be between the rarefaction and the contact discontinuity and between the contact and the shock. $N_{\text{few}}^{\text{POD}}$ should be kept as low as possible not to overload the minimization process. In the Sod test case, we performed a few tests with increasing $N_{\text{few}}^{\text{POD}}$ and 3 was the first value where the minimization process was giving successful results. While this is an heuristic argument, it still provides a valid starting point for the numerical simulations. A more in depth analysis on the role of $N_{\text{few}}^{\text{POD}}$ and on how to choose it in an automated way is envisioned as a future research direction. Again, we solve problem (7) with SLSQP using the same constraints defined in the previous section. This extra optimization step can be skipped when other techniques to detect interesting features can be used [22]. One possibility is the use of classical shock detection procedures to find

the steepest points of the solutions and calibrating them [25]. We summarize the steps of the whole procedure in Algorithm (2).

Algorithm 2 Calibration for quasi–self–similar solution manifold

---

Input: the training solution manifold on the physical domain $\{\rho(\mu_\ell, t_j)$ for $\ell = 1, \ldots, N_{\text{train}}, j = 1, \ldots, N_{\mu_\ell}\}$ and the number of modes $N_{\text{few}}^{\text{POD}}$.

Select *few* parameters $\tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\boldsymbol{\mu}}_{N_{\text{few}}}$.

Find optimal $\overrightarrow{\theta} \in \mathbb{R}^{N_{\text{few}} \times Q}$ by (7) for the *few* parameters $\tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\boldsymbol{\mu}}_{N_{\text{few}}}$.

for $\ell = N_{\text{train}}, \ldots, 1$ **do**
    for $j = N_{\mu_\ell}, \ldots, 1$ **do**
        Set the initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu}_{(\ell,j)})$ as in (5).
        Solve Problem (4).
    end for
end for
Output: the optimal control points $\boldsymbol{w}^{\text{opt}}(\boldsymbol{\mu}_{(\ell,j)})$.

As an alternative to the whole projection onto V_{POD} we would also like to remark that, whenever shock detection techniques can be used, the quasi-self-similar setting can be efficiently handled by the convex displacement interpolation (CDI) [37].

3 The Geometrical Transformation Map

We now present the geometrical transformation map $T[\boldsymbol{w}^{\text{opt}}(\boldsymbol{\mu})]\colon \mathcal{R} \mapsto \Omega$ used to define the calibrated snapshots. We start with the simpler case, namely the 1D setting, and later we consider the 2D case.

3.1 The 1D Setting

Let $\boldsymbol{w}^{\text{opt}}(\boldsymbol{\mu})$ be the control points whose free coordinates are the solution to Problem (4): to interpolate the values $\{(\overline{\boldsymbol{w}}_{\boldsymbol{\alpha}}(\boldsymbol{\mu}), \boldsymbol{w}_{\boldsymbol{\alpha}}^{\text{opt}}(\boldsymbol{\mu}))\}_{\boldsymbol{\alpha}}$, we use monotone cubic C^1 splines. These are the so-called PCHIPs (Piecewise Cubic Hermite Interpolating Polynomials) interpolators, available in the `scipy` Python library under the `interpolate` classes and the built-in function is called `PchipInterpolator`. By employing monotone cubic splines, we obtain a transformation function that preserves the monotonicity of the calibration map $\boldsymbol{w}(\cdot)$, guaranteeing its bijectivity and C^1 smoothness if the calibration points are in the "right order", as prescribed in Sect. 2. In Fig. 4, we can see an example of a PCHIP transformation applied to one of the snapshot calibrated on the detected features. On the right of the figure, the transformation is depicted and we can observe that it interpolates the points, it is $\mathcal{C}^1$ and it is very close to the identity on the boundaries, because we introduce two extra interpolation points outside the boundaries of the domain. This helps to keep the regularity of the transformation in the ALE formulation [21].

In this work, we do not focus on the ALE formulation on $\mathcal{R}$, nevertheless the PCHIPs allow to easily compute all the necessary ingredients. Indeed, they are polynomials and their derivatives and the inverse of their derivative is easy to compute. Moreover, the inverse of the transformation exists and it is unique in each point, hence, with a simple Newton method, we can easily recast the inverse function.
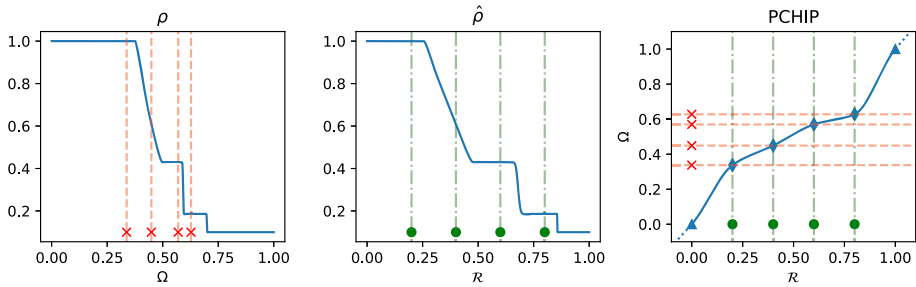
Fig. 4 Example of PCHIP calibration: solution on physical domain with calibration control points (left), solution on reference domain with reference control points (center) and PCHIP transformation with all control points (right)

3.2 The 2D Setting

In the 2D setting we use tensor product of one dimensional PCHIPs, in order to exploit their properties for Cartesian geometries. We refer again to Fig. 2 to better understand the transformation map. We need to compute $T[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})] : \mathcal{R} \mapsto \Omega$, such that

$$T[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\overline{\boldsymbol{w}}^1_{\alpha_1,\alpha_2}, \overline{\boldsymbol{w}}^2_{\alpha_1,\alpha_2}) = (\boldsymbol{w}^{1,\mathrm{opt}}_{\alpha_1,\alpha_2}(\boldsymbol{\mu}), \boldsymbol{w}^{2,\mathrm{opt}}_{\alpha_1,\alpha_2}(\boldsymbol{\mu})) \quad \text{for } \alpha_i = 1, \ldots, M_i, \ i = 1, 2.$$

Let $\hat{\boldsymbol{x}} \in \mathcal{R}$ be a point with coordinates $\hat{\boldsymbol{x}} = (\hat{x}, \hat{y})$. We define the map:

$$T[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y}) := (T^x[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y}), T^y[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y})), \text{ with}$$

$$T^x[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y}) := \sum_{\ell=1}^{M_2} \gamma_\ell^y(\hat{y}) P_\ell^x(\hat{x}), \qquad T^y[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y}) := \sum_{k=1}^{M_1} \gamma_k^x(\hat{x}) P_k^y(\hat{y}).$$

In the previous equations, we made use of the following quantities:

1. P_ℓ^x is a PCHIP interpolating the points $\{\overline{\boldsymbol{w}}^1_{\alpha_1,\ell}, \boldsymbol{w}^{1,\mathrm{opt}}_{\alpha_1,\ell}(\boldsymbol{\mu})\}_{\alpha_1=1}^{M_1}$, where the control points $\overline{\boldsymbol{w}}^1_{\alpha_1,\ell}$ for $\alpha_1 = 1, \ldots, M_1$ are on horizontal lines $\hat{y} = \bar{y}_\ell$ in the reference domain, see Fig. 2, namely for all $\alpha_1 = 1, \ldots, M_1$ we have $P_\ell^x(\overline{\boldsymbol{w}}^1_{\alpha_1,\ell}) = \boldsymbol{w}^{1,\mathrm{opt}}_{\alpha_1,\ell}(\boldsymbol{\mu})$;

2. P_k^y is a PCHIP interpolating the points $\{\overline{\boldsymbol{w}}^2_{k,\alpha_2}, \boldsymbol{w}^{2,\mathrm{opt}}_{k,\alpha_2}(\boldsymbol{\mu})\}_{\alpha_2=1}^{M_2}$, where the control points $\overline{\boldsymbol{w}}^2_{k,\alpha_2}$ for $\alpha_1 = 1, \ldots, M_1$ are on vertical lines $\hat{x} = \bar{x}_k$ in the reference domain, see Fig. 2, namely for all $\alpha_2 = 1, \ldots, M_2$ we have $P_k^y(\overline{\boldsymbol{w}}^2_{k,\alpha_2}(\boldsymbol{\mu})) = \boldsymbol{w}^{2,\mathrm{opt}}_{k,\alpha_2}(\boldsymbol{\mu})$;

3. $\gamma_\ell^y(\cdot)$ is a PCHIP interpolating the points $\{\overline{\boldsymbol{w}}^2_{\alpha_1,\alpha_2}, \delta_{\alpha_2,\ell}\}_{\alpha_2=1}^{M_2}$, being $\delta_{\alpha_2,\ell}$ the Kronecker delta. By doing so, we obtain that $T^x[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})]$ is a convex combination of the $\{P_\ell^x(\cdot)\}_{\ell=1}^{M_2}$ such that $T^x[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x}, \hat{y} = \overline{\boldsymbol{w}}^2_{\alpha_1,\alpha_2}) = P_{\alpha_2}^x(\hat{x})$;

4. $\gamma_k^x(\cdot)$ is a PCHIP interpolating the points $\{\overline{\boldsymbol{w}}^1_{\alpha_1,\alpha_2}, \delta_{\alpha_1,k}\}_{\alpha_1=1}^{M_1}$, as before, leading to the property $T^y[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\hat{x} = \overline{\boldsymbol{w}}^1_{\alpha_1,\alpha_2}, \hat{y}) = P_{\alpha_1}^y(\hat{y})$.

Notice that, ultimately, it holds that

$$T[\boldsymbol{w}^{\mathrm{opt}}(\boldsymbol{\mu})](\overline{\boldsymbol{w}}^1_{\alpha_1,\alpha_2}, \overline{\boldsymbol{w}}^2_{\alpha_1,\alpha_2}) = (P_{\alpha_2}^x(\overline{\boldsymbol{w}}^1_{\alpha_1,\alpha_2}), P_{\alpha_1}^y(\overline{\boldsymbol{w}}^2_{\alpha_1,\alpha_2})) = \boldsymbol{w}^{\mathrm{opt}}_{\alpha_1,\alpha_2}(\boldsymbol{\mu}).$$

Also in the 2D case, the Jacobian of the transformation, which is needed in the ALE formulation, is easily accessible, since all the terms are polynomials. Similarly to the 1D case, we can compute the Jacobian of the inverse of the transformation, using the inverse of the Jacobian of the transformation, provided that we can invert the map T.
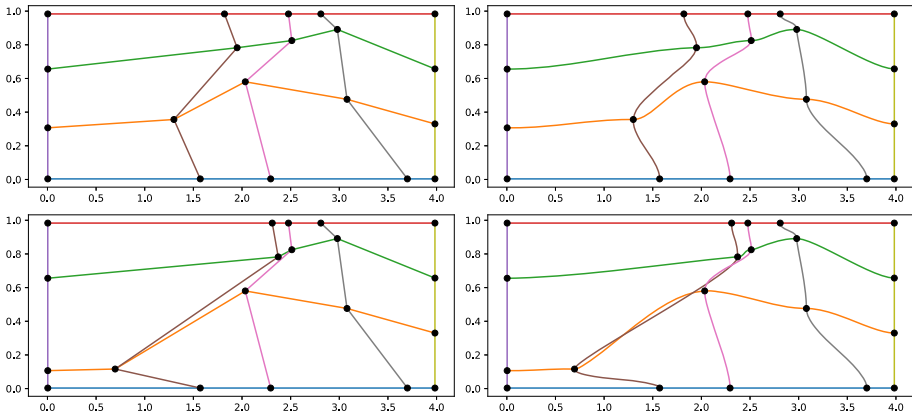
Fig. 5 Calibration points ($M_1 = 5$, $M_2 = 4$) on the physical domain $\Omega = [0, 4] \times [0, 1]$ (left) and the transformation of horizontal and vertical lines on the physical domain (right). On the top calibration points that lead to an invertible map, on the bottom calibration points that respect the monotonicity in each line (left), but whose transformed horizontal and vertical lines cross multiple times (right) and, hence, whose transformation is not invertible

3.2.1 Invertibilty of T

The computation of the inverse of T is fundamental in many aspect of the algorithm: to display the function on the physical domain, to compute quantities and errors on the physical domain and to compute the inverse of the Jacobian for the ALE formulation.

Unfortunately, the so defined map T cannot be proven to be invertible, as it might happen that some of the vertical and horizontal lines cross each other multiple times in the physical domain, see Fig. 5 second line. To avoid this, we impose, in the optimization procedure for the calibration, to have positive determinant of the Jacobian of the transformation on the meshpoints of the reference domain, see the constraints in Sect. 2.1. This typically guarantees invertibility. We recall that, being PCHIPs polynomials, the computation of the Jacobian can be performed explicitly in each point. To compute the inverse of the transformation, we perform the following steps. We first apply the transformation map T to the elements of the Cartesian meshgrid of $\mathcal{R}$: these will be mapped to quadrilateral elements in Ω. Now, given a point $x \in \Omega$, we can easily find to which of these quadrilaterals it belongs and with a Projective transformation (see the python module `transform` of the package `skimage`) we pull it back onto $\mathcal{R}$. Then, we use the found point as initial guess to find the solution of $T[w^{\mathrm{opt}}(\mu)](\hat{x}) = x$ through a Newton type nonlinear solver (`scipy.root`).

4 Model Order Reduction with Calibration

We are now ready to perform the model order reduction step. In what follows the procedure is similar for the non parametric and the parametric setting: we will therefore present it for the latter case, for the sake of generality.
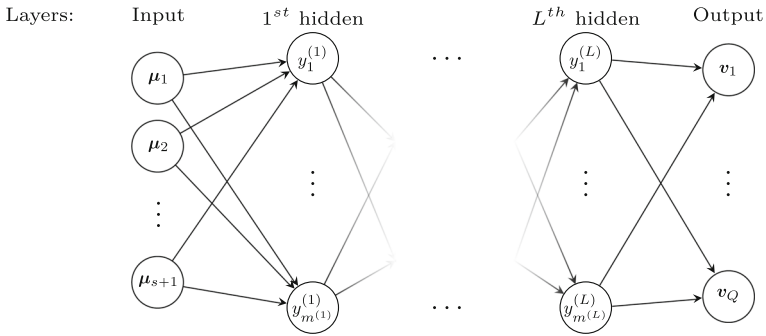
Fig. 6 Example of the architecture of an ANN

4.1 Learning the Calibration Map

Once the calibration procedure presented in Sect. 2 has been carried out, the map $\boldsymbol{w} \colon \mathcal{P} \mapsto \Theta$ is known only through the sample values $\boldsymbol{w}^{\text{opt}}(\boldsymbol{\mu})$, for $\boldsymbol{\mu} \in \mathcal{P}^{\text{train}}$. To learn the calibration map $\boldsymbol{w}(\cdot)$ for any parameter $\boldsymbol{\mu}$, we employ an Artificial Neural Network (ANN) composed by several layers: an input layer $\boldsymbol{\mu} \in \mathcal{Y}^{(0)} = \mathcal{P}$ where we pass the parameters of the problem of interest, L hidden layers $\mathcal{Y}^{(j)}$, $j = 1, \dots, L$, and an output layer $\mathcal{Y}^{(L+1)}$, see Fig. 6. As output layer, we would like to obtain $\boldsymbol{\theta} \in \mathbb{R}^Q$, from which we can extract the calibration points $\boldsymbol{w}$. Keeping in mind the monotonicity constraints applied to the control points in the constrained minimization problems (4) and (6), we try to enforce this constraint in the ANN. It is not easy to strongly enforce such constraints, but we can force the output to be positive, using as final activation function a Softplus. Hence, we take as output layer of our ANN not directly $\boldsymbol{\theta}$, but the vector $\boldsymbol{v} \in \mathbb{R}^Q$ of the differences of the free coordinates with the *previous* ones (in 2D it is referred to the same line). Doing so, the positivity of $\boldsymbol{v}$ is equivalent to the monotonicity in each line of points. In one dimension, it is defined as $v_i = w_i - w_{i-1}$ for $i = 2, \dots, M_1 - 1$, while in two dimensions this operation is done in each horizontal or vertical line.

Each layer $\mathcal{Y}^{(j)}$, $j = 1, \dots, L$ is connected to the next and to the previous ones through affine maps $\delta^{(j)} \colon \mathcal{Y}^{(j)} \mapsto \mathcal{Y}^{(j+1)}$ and at every node a nonlinear activation function $\zeta^{(j)} \colon \mathbb{R} \mapsto \mathbb{R}$ is applied component–wise. We used the hyperbolic tangent in all the ANN except in the output layer where the Softplus function is used for the positivity of the outputs. On a training set, the learning process changes the weights $\delta^{(k)}$ minimizing the error between the output and the optimal calibration points.

To build and train the ANN, we used a Python based library EZyRB [38] that uses Adam method [39] to perform the minimization. More details on the architecture of the ANN for the different test cases will be provided in the numerical section.

4.2 POD-NN

In Sects. 2.1 and 2.2, we presented the calibration algorithm in the self-similar and in the quasi self-similar setting. Thanks to the calibration of the snapshots, we obtain the *calibrated manifold* $\hat{\mathcal{M}}_\rho = \{\hat{\rho}(\boldsymbol{\mu})(\cdot), \quad \boldsymbol{\mu} \in \mathcal{P}_{\text{train}}\}$, where $\hat{\rho}(\boldsymbol{\mu})(\cdot)$ is the *calibrated snapshot* defined in Sect. 2. We now proceed with the compression of $\hat{\mathcal{M}}_\rho$ by means of the Proper Orthogonal Decomposition (POD): we refer to [2] for more details. The dimension of the reduced basis

can be chosen either setting a maximum number of basis $N_{\text{POD}}^{\max}$ or choosing the most relevant modes such that the discarded energy is smaller than a certain tolerance τ_{POD}. Once the POD has been carried out, we obtain a linear space V_N^ρ spanned by the N orthonormal reduced basis functions $\{\hat{\Phi}_i\}_{i=1}^N$ on the reference domain $\mathcal{R}$. V_N^ρ should now represent with a good accuracy any element of the calibrated solution manifold $\hat{\mathcal{M}}_\rho$.

4.3 Online Solution by Means of ANN

In this work, we mainly focus on the calibration procedure and the offline phase. Hence, we will use a non–intrusive approach for the reconstruction of the online solutions. Let $\boldsymbol{\mu} \in \mathcal{P}$ be a parameter of choice: the goal is to construct a linear approximation $\rho_N(\boldsymbol{\mu})$ of the snapshot $\rho(\boldsymbol{\mu})$. It should be clear by now that, since we are dealing with advection dominated problems, this is not a simple task within the standard MOR setting. However, in Sects. 2 and 3, we presented a calibration technique that allows us to obtain a linear space V_N^ρ that approximates with good accuracy the calibrated manifold $\hat{\mathcal{M}}_\rho$: we are therefore able to construct a linear approximation of the calibrated solution of interest $\hat{\rho}(\boldsymbol{\mu})$ *in the reference domain* $\mathcal{R}$. This means that we can approximate $\hat{\rho}(\boldsymbol{\mu})$ with

$$\hat{\rho}(\boldsymbol{\mu}) \approx \hat{\rho}_N(\boldsymbol{\mu}) := \sum_{i=1}^N \underline{\hat{\rho}}_N^i(\boldsymbol{\mu}) \hat{\Phi}_i,$$

where $\underline{\hat{\rho}}_N^i(\boldsymbol{\mu}) = (\hat{\rho}(\boldsymbol{\mu}), \hat{\Phi}_i)_{L^2(\mathcal{R})}$ for $i = 1, \ldots, N$, being $(\cdot, \cdot)_{L^2(\mathcal{R})}$ the L^2 scalar product on the reference domain. In order to find the vector $\underline{\hat{\rho}}_N(\boldsymbol{\mu})$ one can adopt two alternative ways: an intrusive approach, by means of a Galerkin projection of the high order algebraic system onto the reduced space, or a non-intrusive approach by means of an ANN. If the standard Galerkin projection setting is adopted, the online system and the reconstruction of the online solution is carried out within an ALE formalism [21, 36, 40]: the original problem of interest, formulated over Ω, has to be re-written into a problem formulated over the reference domain $\mathcal{R}$. In this approach, a hyper-reduction procedure [41, 42] will be necessary to tackle the nonlinearities of the problem and of the transformation map. This approach is currently under investigation, and it will be part of a future extensions.

An alternative to the intrusive approach is represented by the use of ANN, in the spirit of Sect. 4.1. We consider the map $\Pi_N : \mathcal{P} \mapsto \mathbb{R}^N$

$$\Pi_N(\boldsymbol{\mu}) := \underline{\hat{\rho}}_N(\boldsymbol{\mu}) = [\underline{\hat{\rho}}_N^1(\boldsymbol{\mu}), \ldots, \underline{\hat{\rho}}_N^N(\boldsymbol{\mu})]^T.$$

We make use of an ANN to learn the L^2 projection map Π_N: to train the map, we employ the set of input samples $\boldsymbol{\mu}$ for $\boldsymbol{\mu} \in \mathcal{P}^{\text{train}}$, and the set of output samples $\{\underline{\hat{\rho}}_N(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathcal{P}^{\text{train}}\}$. In this algorithm, we do not use the optimal calibrated points obtained with the optimization process, but we use the ones predicted by the ANN of Sect. 4.1. Doing so, any systematic error in the online calibration should be already taken into account while performing the L^2 projection and automatically corrected by this approach. Moreover, it is also possible to use different training sets for the calibration optimization procedure and the model order reduction ones. More details on the architecture of the ANN employed will be provided in the numerical section.

Table 1 Test details for the 1D Sod shock tube problem, non parametric (Nonparam) and parametric (Param) setting

Quantity	Nonparam	Param	Quantity	Nonparam	Param
ρ_L	1	[0.7, 1.3]	ρ_R	0.1	[0.1, 0.15]
u_L	0	0	u_R	0	0
p_L	1	[0.7, 1.3]	p_R	0.125	[0.05, 0.15]
t_f	0.2s	0.2s	N_h	1500	1500

4.4 Offline–Online Splitting

We summarize in this section the operations that should be performed in the offline part (i.e. only once for every new parametric problem one wishes to consider) and the online operations that should be performed to predict the solution for a new parameter.

Offline phase: FOM simulations for all parameters in the training set, choice of the reference calibration points, optimization procedure to find the calibration points for a training set, learning of the regression map for the calibration points for all parameters, transformation of the FOM solutions onto the reference domain, computation of the RB, learning of the regression for the coefficients-to-(reduced coefficient) map.

Online phase: given a new parameter evaluate the regression of the calibration points to know the geometrical transformation, evaluate the regression for the RB coefficients to get the solution on the reference domain, combine the two maps to get the solution on the physical domain.

5 Numerical Results

We now present some numerical results to validate the proposed methodology. We will consider different time dependent test cases: the Sod shock tube problem in 1D, in the non parametric and in the parametric setting, already introduced in Sects. 2.1 and 2.2, respectively. To further test the performance of our methodology, we subsequently consider a 2D problem, namely the double Mach reflection (DMR) problem, again in the non parametric and in the parametric setting. To conclude, some numerical results for the non parametric triple point problem will be shown. For all presented tests, the FOM consists of high order solutions obtained with an explicit Finite Volume discretization with WENO reconstruction of order 5, with explicit time discretization given by the optimal SSPRK54, with CFL coefficient 0.8 and Rusanov numerical flux.

5.1 Non-Parametric Sod Shock Tube Problem in 1D

We consider Problem (2) introduced in Sect. 1.1, where the physical domain is $\Omega = \mathcal{R} = [0, 1]$. The number of spatial degrees of freedom is 1500 and this leads to computational costs of around 2 min using a Fortran code [43] on a Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz. In both cases, we use Dirichlet BC as the waves do not exit the domain before the final time. The details of all the relevant quantities are presented in Table 1.
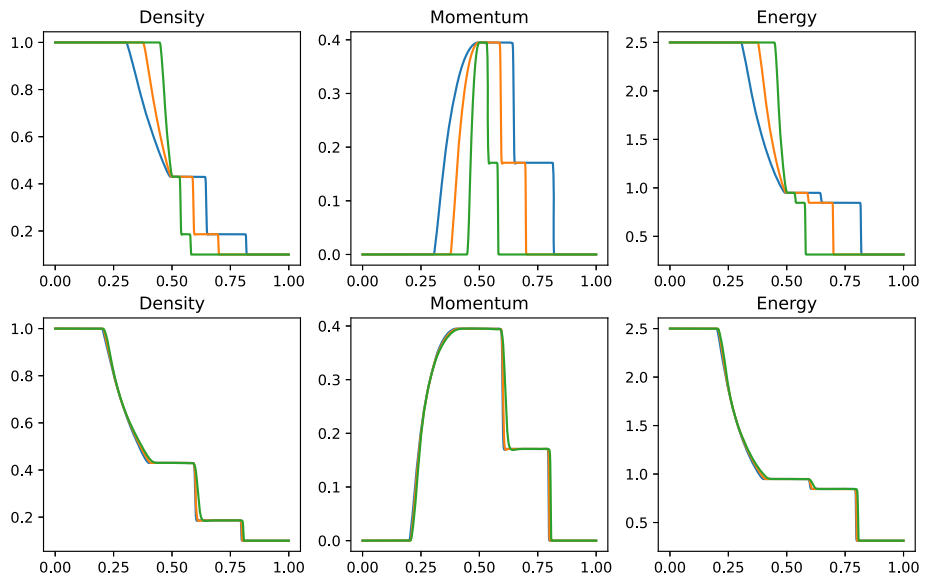
Fig. 7 FOM simulation of Sod 1D problem non parameteric at times 0.04 (green), 0.1 (orange) and 0.16 (blue), on the original domain (top) and calibrated on the reference domain (bottom)

Table 2 Calibration of the 1D Sod shock tube problem, non parametric and parametric setting

Quantity	Nonparam	Param	Quantity	Nonparam	Param
δ	10^{-6}	10^{-6}	α	0.	0.
$\overline{\rho}$	$\rho(t=0.16)$	–	M_1	6	6
Max. iter	100	100	minim. alg	SLSPQ	SLSPQ
N_{train}	1	16	N_μ	25	25
N_{few}	–	10	$N_{\text{few}}^{\text{POD}}$	–	3

Figure 7 shows some snapshots for the density ρ, the moment m and the energy E at three different times of the simulation: it is clear that the structures of all three the components of the solution present discontinuities that travel in the domain at the same locations.

We carry out the calibration technique proposed in Algorithm 1: we choose as reference solution $\overline{\rho}$ the density $\rho(t=0.16)$. We then choose $M=4$ control points $\overline{w}_1 = 0.2$, $\overline{w}_2 = 0.4$, $\overline{w}_3 = 0.6$, $\overline{w}_4 = 0.8$ equispaced in the reference domain $\mathcal{R} = [0,1]$. The calibrated solutions (using the ANN to forecast the calibration points) are shown in Fig. 7: the main features of the solutions, namely the shock, the contact discontinuity and the rarefaction wave are well aligned with the reference solution. The details of the quantities required to carry out the calibration step are listed in Table 2. We point out that the calibration step and the training of the ANN have been carried out on the training set $\mathcal{P}^{\text{train}} \subset [0.01, 0.16] \subset [0, 0.2]$ sampled with 25 equispaced parameters. The reason for excluding the first timesteps from the training is that the minimization is tricky during the first timesteps: indeed we have a transition phase, during which all the features are in the same point, leading to non invertible maps. An alternative way to overcome this difficulty could be to restore to local reduced basis
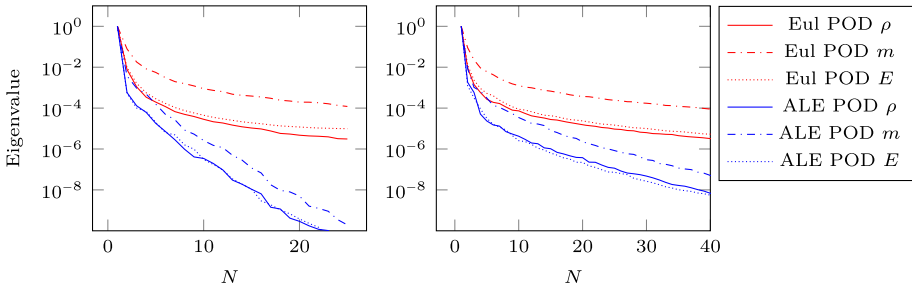
Fig. 8 Sod 1D: Eigenvalue decay of the PODs (normalized to have $\lambda_1 = 1$) non parametric (left) and parametric (right)
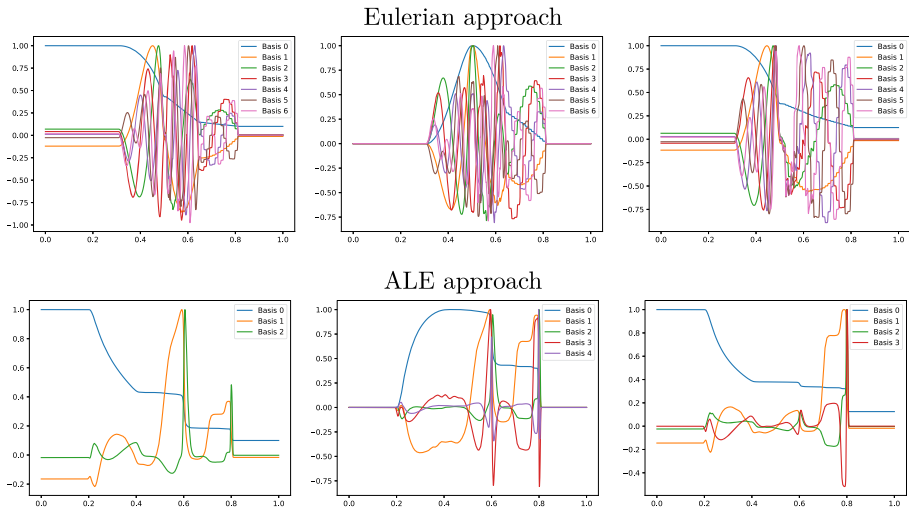
Fig. 9 Sod 1D non parametric: The first modes obtained by compressing the original manifolds (Eulerian on the top) and the calibrated manifolds (ALE on the bottom) for ρ (left), m (center) and E (right), with POD with $\tau_{\text{POD}} = 10^{-4}$ and $N_{\text{POD}}^{\max} = 7$

spaces [44], or to use a FOM approach for the first timesteps. The final times are excluded to test the extrapolatory performances of the ROM.

Figure 8 (left) shows the eigenvalue decay of the POD for both calibrated (ALE, in blue) and original (Eulerian, in red) approaches. We can see that, differently from the Eulerian approach, for the calibrated approach the first eigenvalue is much more relevant than all the others and the Kolmogorov n–width decay is much faster. Figure 9 shows the behavior of the first modes obtained by compressing with a POD the non-calibrated manifolds, for the three conservative variables ρ, m and E. We remark that also here we consider the FOM solutions for $t \in \mathcal{P}^{\text{train}}$, thus excluding the initial and the final times from the compression. The modes are highly oscillatory, because they struggle to correctly represent the positions of the three discontinuities in the domain. Figure 9 also shows the POD modes obtained by running a POD on the calibrated manifolds: after the calibration the oscillations in the modes are much milder and focused on the discontinuity locations. Figure 10 shows some POD-NN results with $N = 7$, without calibration, for the density ρ at different times t (including the extrapolatory regime at $t = 0.2$). The comparison is made between the FOM
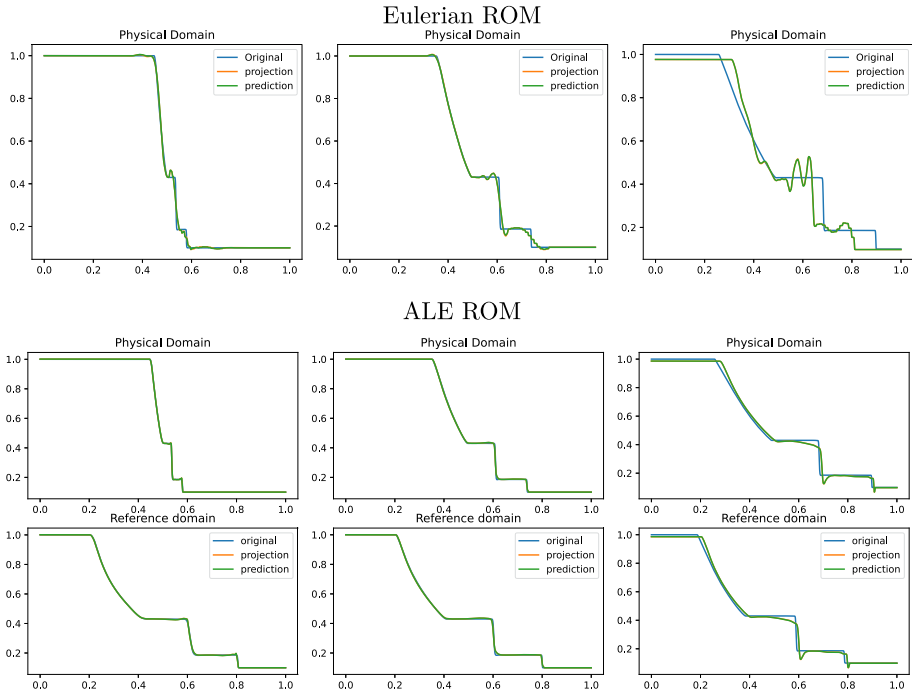
Fig. 10 Sod 1D non parametric: Online approximation of the density ρ at times 0.04, 0.12 and 0.2 (left to right). Top row: Eulerian ROM simulations with $N = 7$ modes on Ω. Central row: ALE ROM simulations on Ω with $N = 3$ modes. Bottom row: ALE ROM simulations on $\mathcal{R}$ with $N = 3$ modes

solution ρ, its L^2 projection onto the reduced space generated by the first $N = 7$ modes of the non-calibrated manifold (Eulerian modes) and the online reconstruction obtained using a linear combination of said modes, with coefficients that are predicted by an ANN. We can see that the Eulerian approach fails to correctly reproduce the calibrated FOM solutions: in particular, the standard MOR struggles to capture the correct position of the discontinuities, and it shows some oscillations in the online approximation that are most likely due the highly oscillating nature of the Eulerian modes themselves. Figure 10 also shows some POD-NN simulations obtained after the calibration procedure (computational cost of prediction of both calibration points and ROM coefficients below 0.001s); here we use $N = 3$ modes as the POD algorithm stops earlier for the imposed tolerance. We can see that we obtain a very good approximation of the calibrated snapshots in the reference configuration $\mathcal{R}$, i.e., $\hat{\rho}$, and in the physical domain Ω, i.e. ρ, with the main features correctly reproduced by the online solution. We stress the fact that $t = 0.2$ is *outside* the training interval $\mathcal{P}^{\mathrm{train}}$: in this case, the positions of the shock, the contact and the rarefaction wave have been slightly misplaced by the online model, hence, the approximation is not as great as in the interpolatory regime. All the details on the architecture of the ANN used to learn the calibration map and to predict the online solution are summarized in Table 3.

5.1.1 Validation of the Calibration Strategy

In this section, we focus on the two following aspects: the initial guess considered in the calibration, for each parameter, and the order of the parameters for the calibration. To validate

Table 3 Architecture and details of the calibration-NN and the POD-NN in both non parametric and parametric setting

Parameter	Non parametric case		Parametric case	
	Calibration-NN	POD-NN	Calibration-NN	POD-NN
L	4	4	4	4
Neurons per layer	16	16	16	16
Max. epochs	20000	10000	20000	10000
Loss fun. tol	10^{-6}	10^{-5}	10^{-6}	10^{-5}
$\tilde{\zeta}$	Tanh/Softplus	Tanh	Tanh/Softplus	Tanh
τ_{POD}	–	10^{-4}	10^{-4}	10^{-4}
N_{POD}^{max}	–	7	3	7

our methodology, we perform two types of tests, described in the following paragraphs. We recall that, since we are in the non-parametric case here, the parameter $\boldsymbol{\mu}$ represents only the time t. Again, we also remark that we focus here on the density ρ.

Reference solution choice in the calibration
We are interested in performing the calibration for every parameter, by changing the reference calibration points $\bar{\boldsymbol{w}}$ through the free coordinates $\bar{\boldsymbol{\theta}}$, the reference solution $\bar{\rho} = \rho(\bar{\boldsymbol{\mu}})$ and the initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu})$ in the minimization problem (4). We then evaluate the calibration error: the chosen measures for estimating this error will be clarified later on in this paragraph.

Exact characteristics calibration reference points. For this test, we know the exact solution of the problem (2), and therefore the exact location of the rarefaction and the discontinuities of the density function. To perform the calibration, we choose a reference parameter $\bar{\boldsymbol{\mu}} \in \mathcal{P}$, and we consider $\bar{\rho} = \rho(\bar{\boldsymbol{\mu}})$ and $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^{ex}(\bar{\boldsymbol{\mu}})$, where $\boldsymbol{\theta}^{ex}(\bar{\boldsymbol{\mu}}) \in \mathbb{R}^4$ is the vector of the exact locations of the beginning and end of the rarefaction wave, the contact and the shock discontinuities for $\rho(\bar{\boldsymbol{\mu}})$. We then carry on the calibration, for every $\boldsymbol{\mu} \in \mathcal{P}^{train}$: the initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu})$ will be $\boldsymbol{\theta}^{ex}(\bar{\boldsymbol{\mu}})$. We study what happens if we change $\bar{\boldsymbol{\mu}}$: the measure that we consider to estimate the calibration error in this case is the 1-norm of the difference between $\boldsymbol{\theta}^{ex}(\boldsymbol{\mu})$ (the vector of exact locations of beginning and end of the rarefaction wave, the contact and the shock discontinuities for the $\rho(\boldsymbol{\mu})$) and $\boldsymbol{\theta}^{opt}(\boldsymbol{\mu})$ (the output of the calibration for the parameter $\boldsymbol{\mu}$), namely

$$||\boldsymbol{\theta}^{opt}(\boldsymbol{\mu}) - \boldsymbol{\theta}^{ex}(\boldsymbol{\mu})||_1. \tag{8}$$

This choice allows us to measure how far our output is from the actual location of the discontinuities and rarefaction of the solution. The result is reported in Fig. 11 (left). Using the initial guess close to the final time results in the largest errors when calibrating early time solutions, while the opposite seem less problematic and optimally one could choose a reference solution close to the initial time to have the least amount of error for all the considered optimized parameters. It is also clear from this test that the best choice would be to choose the initial guesses close to the parameter that we are calibrating minimizes the error, i.e. staying close to the diagonal. And this is what we are actually doing in presented strategy in Sect. 2.1 in (5).

Equispaced calibration reference points. For this second test, we assume we do not know the exact solution, nor the wave locations. In this case, we choose as reference calibration points $\bar{\boldsymbol{\theta}} = \{0.2, 0.4, 0.6, 0.8\}$ equispaced points in $[0, 1]$: these points will be fixed for all the reference parameters $\bar{\boldsymbol{\mu}}$ we will use and they will also be used as initial guess of the
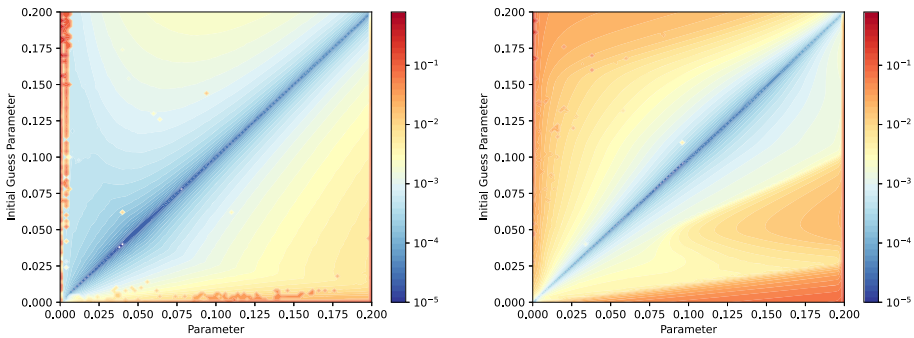
Fig. 11 Calibration error for Sod 1D at different times (parameter μ) using different initial guess times (parameter $\bar{\mu}$), for the FV solutions. Left: $\overline{\rho} = \rho(\bar{\mu})$, $w^{(0)}(\mu) = w^{ex}(\bar{\mu})$, error measure (8). Right: $\overline{\rho} = \rho(\bar{\mu})$ and $\bar{w} = w^{(0)}(\mu) = \{0.2, 0.4, 0.6, 0.8\}$ for all μ, error measure (9)

Table 4 Details for the order calibration test, where we analyze the calibration output according to the order (in time) in which we perform it

Strategy	Reference sol	Order	Initial guess
T2B	Final time	From last time to first	Last parameter computed
T2B10	Final time	From last time to first, every 10 steps	Last parameter computed
B2T	Initial time	From first time to last	Last parameter computed
B2T10	Initial time	From first time to last, every 10 steps	Last parameter computed
Fixed	Final time	From first time to last	Final time

calibration process. We then choose $\overline{\rho} = \rho(\bar{\mu})$, and we analyse the results of the calibration when we change $\bar{\mu}$. The measure chosen to study the error of the calibration (since we do not know $\theta^{ex}(\mu)$) is the following:

$$||\hat{\rho}(\mu, w(\theta^{opt}(\mu))) - \Pi_{\overline{\rho}}\hat{\rho}(\mu, w(\theta^{opt}(\mu)))||_2^2, \tag{9}$$

where $\Pi_{\overline{\rho}}$ is the projection onto the linear space spanned by the chosen reference solution. The results are reported in Fig. 11: again, we can see that choosing a reference solution close to the parameter of interest leads to smaller errors and that using one reference solution the optimal choice would be something close to $\bar{\mu} \approx 0.1$.

Order in which we perform the calibration
Motivated by the results shown in Fig. 11, we now use as initial guess the calibration points found for the closest parameter already computed. We will compare the different orders described in Table 4. In particular, we will try to start from the final time or the initial one and we will use either all 100 timesteps we have in the training set or only one every 10 of these. Finally, we also compare it with a fixed initial guess strategy. As we can observe in Fig. 12, starting with the final solution as reference solution, we obtain considerably lower errors all along the time domain, almost independently on how many parameters we sample (this will not be true for more complicated tests in 2D), with respect to using the initial solution as a reference one. The authors are anyway surprised by how the error using the initial solution as reference is not too large. Also the fixed initial guess at the final time produces interesting results.
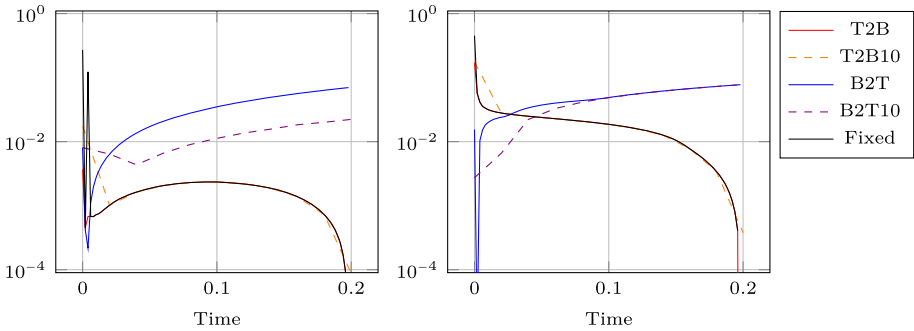
Fig. 12 Calibration error for Sod 1D at different times (parameters) using different calibration strategies, for the FV solutions. Left: exact waves calibration as initial guess and right for equispaced initial guess calibration points. Error measure: (8) (left) and (9) (right)

In 2D tests, we cannot perform similar comparison, as no exact calibration could be performed. We can undoubtedly say that in 2D being close to the initial guess is of more importance than in 1D, hence, in general, it is better to have a large enough training set.

5.2 Parametric Sod Shock Tube Problem in 1D

We now consider the parametric version of the Sod problem, already introduced in Sect. 2.2. We recall that we consider $\mu = (\rho_L, \rho_R, p_L, p_R) \in \mathcal{P}_{\text{phys}} \subset \mathbb{R}^4$. All the details for the numerical simulations are provided in Table 1. Also in this case, the FOM solutions have been obtained employing the same FV discretization. In the parametric setting, we generate the training space $\mathcal{P}_{\text{phys}}^{\text{train}}$ using $N_{\text{train}} = 16$ randomly selected parameters μ from $\mathcal{P}_{\text{phys}}$. Again, we consider the training time interval $[0.01, 0.16]$ discretized with around 45 times for each physical parameter.

Figure 13 shows the first modes for the three components ρ, m and E, without calibration (Eulerian approach) and with calibration (ALE approach), respectively. Also in the parametric case, we can notice that the Eulerian modes are highly oscillating, similarly to the non-parametric test case: the calibration helps to significantly mitigate this phenomenon. To further validate this, we show in Fig. 8 (right) a comparison between the rate of decay of the eigenvalues obtained with a POD on the non-calibrated (red) and calibrated (blue) manifolds. The calibration results in an improvement in the rate of decay and we clearly observe that, in comparison to the non parametric case, the decay is slower and we need more basis functions to represent our solution manifold. All the details for the numerical implementation of the calibration procedure are summarized in Table 2.

Figures 14 and 15 represent the FOM, the L^2 projection on the reduced space and the POD–NN online approximation for ρ, for two parameters in the test set. We plot both the Eulerian ROM and the ALE one, for the latter both in the physical Ω and in the reference domain $\mathcal{R}$. The online approximations are obtained with $N = 4$ modes. As we can see, in both cases the Eulerian ROM is struggling to correctly capture the positions of the discontinuities, and it provides an approximated solution that exhibits some non-negligible oscillations, most likely due to the oscillating nature of the Eulerian modes themselves. The results provided with the calibration are much more accurate, since the MOR is now able to correctly represent the positions of the discontinuities, and it does not present any oscillations in the approximations.
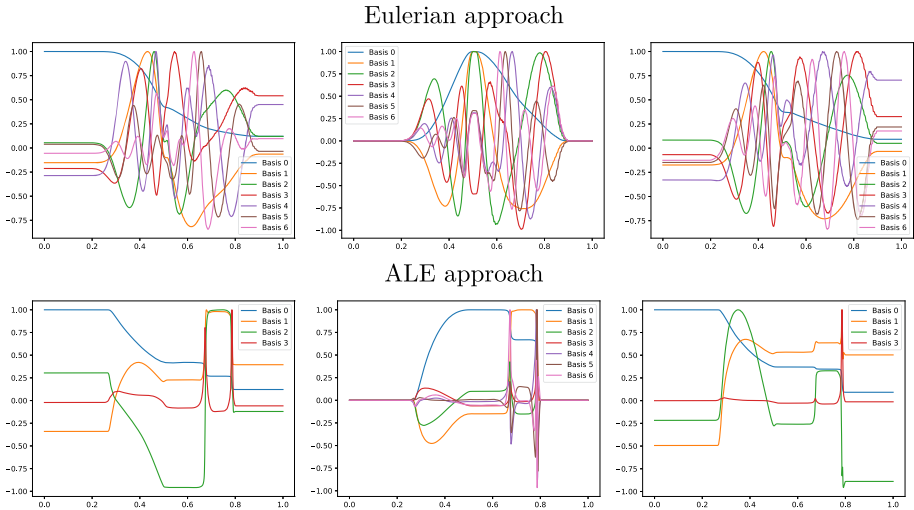
Fig. 13 Sod 1D parametric: The first modes obtained by compressing the original manifolds (Eulerian on the top) and the calibrated manifolds (ALE on the bottom) for ρ (left), m (center) and E (right), with POD with $\tau_{\text{POD}} = 10^{-4}$ and $N_{\text{POD}}^{\max} = 7$
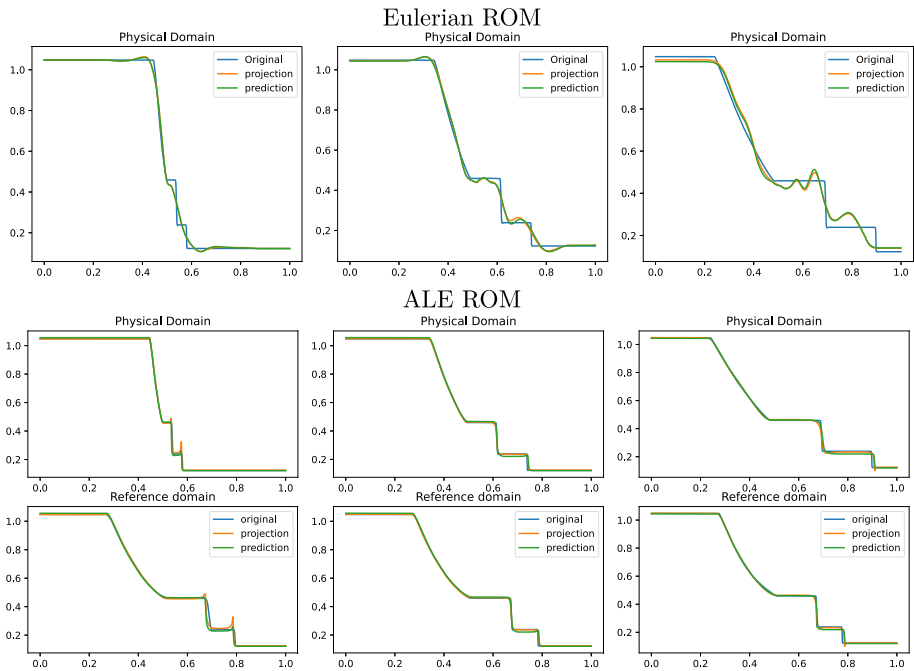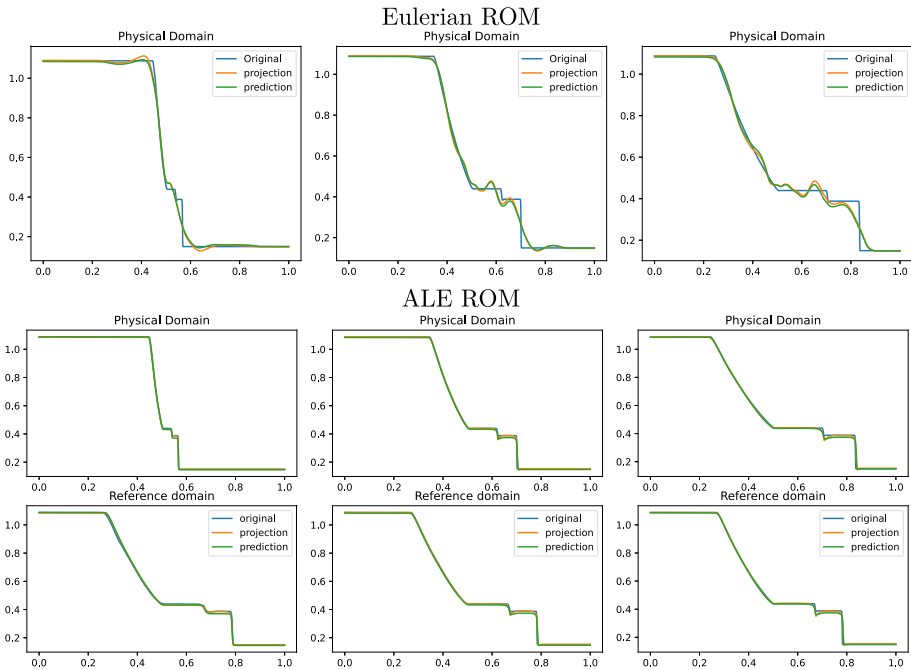

Fig. 14 Sod 1D parametric: Online approximation of the density $\overline{\rho}$ at times 0.04, 0.12 and 0.2 (left to right), for $\rho_L = 1.047937$, $\rho_R = 0.122810$, $p_L = 1.203980$, $p_R = 0.144468$. Top row: Eulerian ROM simulations on Ω with $N = 7$ modes. Central row: ALE ROM simulations on Ω with $N = 4$ modes. Bottom row: ALE ROM simulations on $\mathcal{R}$ with $N = 4$ modes

Fig. 15 Sod 1D parametric: Online approximation of the density $\overline{\rho}$ at times 0.04, 0.12 and 0.2 (left to right), for $\rho_L = 1.08827$, $\rho_R = 0.149654$, $p_L = 1.193154$, $p_R = 0.078459$. Top row: Eulerian ROM simulations on Ω with $N = 7$ modes. Central row: ALE ROM simulations on Ω with $N = 4$ modes. Bottom row: ALE ROM simulations on $\mathcal{R}$ with $N = 4$ modes

There are minor flaws in the extrapolatory regime and in the early times, still keeping the quality of the reduced solution very high.

All the details on the architecture of the ANN used to learn the calibration map and to predict the online solution are summarized in Table 3.

5.3 Non-Parametric DMR Problem in 2D

We now consider a 2D test case, namely the Double Mach Reflection (DMR) problem [35]. Let $\Omega = [0, 4] \times [0, 1]$: we consider the Euler Eq. (2), in the time interval $[0, 0.25]$, with the following IC

$$\begin{cases} (\rho_L, u_L, v_L, p_L) = (8, 8.25\cos(\beta), -8.25\sin(\beta), 116.5) & x \in \Omega_L(\beta, t = 0) \\ (\rho_R, u_R, v_R, p_R) = (1.4, 0, 0, 1) & x \in \Omega_R(\beta, t = 0) \end{cases} \tag{10}$$

$$\Omega_L(\beta, t) = \left\{ x \in \Omega : x < \frac{1}{6} + \tan(\beta)y + \frac{10}{\cos(\beta)}t \right\}, \qquad \Omega_R(\beta, t) = \Omega \setminus \Omega_L(\beta, t), \tag{11}$$
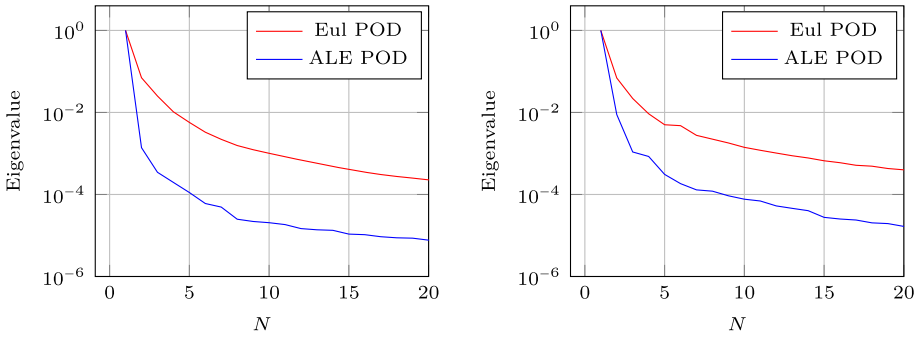
Fig. 16 DMR: Eigenvalue decay of the PODs (normalized to have $\lambda_1 = 1$), non parametric on the left, parametric on the right
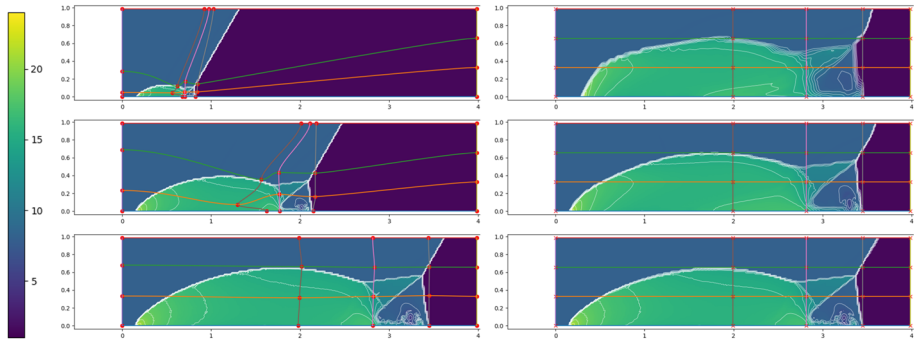
Fig. 17 FOM solution for ρ for DMR non parametric at times 0.05 (top), 0.15 (center) and 0.25 (bottom) in the physical domain Ω (left) and, after calibration, in the reference configuration $\mathcal{R}$ (right). We mark on the plots the control points and the Cartesian grid that links them in the reference domain and its image through T on the physical one. We plot in white 20 contour lines at equispaced values between 1 and 25

with $\beta = \frac{\pi}{6}$. The BCs are assigned through ghost cells as

$$(\rho, u, v, p) = \begin{cases} (\rho_L, u_L, v_L, p_L), & \text{if } x = 0 \text{ or } (y = 1 \text{ and } \boldsymbol{x} \in \Omega_L(\beta, t)), \\ (\rho_R, u_R, v_R, p_R), & \text{if } y = 1 \text{ and } \boldsymbol{x} \in \Omega_R(\beta, t), \\ (\rho_{in}, u_{in}, -v_{in}, p_{in}), & \text{if } y = 0 \text{ (wall BC)}, \\ (\rho_{in}, u_{in}, v_{in}, p_{in}), & \text{if } x = 4 \text{ (outflow)}, \end{cases} \tag{12}$$

where $\cdot_{in}$ denotes the value inside the domain at the corresponding boundary cell.

Figure 17 (left column) represents the FOM snapshots for the density ρ at three different times of the numerical simulation; here, the same FV scheme has been employed at the FOM level on a mesh of 2400×600 cells (computational time of 5 days), then downsampled to a mesh of 240×60 cells to perform the offline phase (including calibration) in reasonable computational times. We retain 500 time samples in $[0, 0.25]$ of which we include 100 in the training set of the calibration procedure (every $\Delta t = 0.0025$) and 45 in the training set of the reduced algorithms all in $[0.02, 0.2]$.

Figure 16 (left) shows the rate of decay of the eigenvalues returned by the POD on $\mathcal{M}_\rho$ (red): also for this test case we have a solution manifold with a slowly decaying Kolmogorov n-width, due to the fact that the shock moves inside Ω. We therefore perform a calibration

Table 5 Calibration of the 2D DMR problem, non parametric and parametric setting

Quantity	Nonparam	Param	Quantity	Nonparam	Param
$\overline{\rho}$	$\rho(t = 0.2)$	–	α	10^{-4}	10^{-4}
M_1	7	7	δ	10^{-2}	10^{-1}
M_2	6	6	minim. alg	SLSPQ	SLSPQ
N_{train}	1	16	max. iter	100	100
N_μ	180	23			
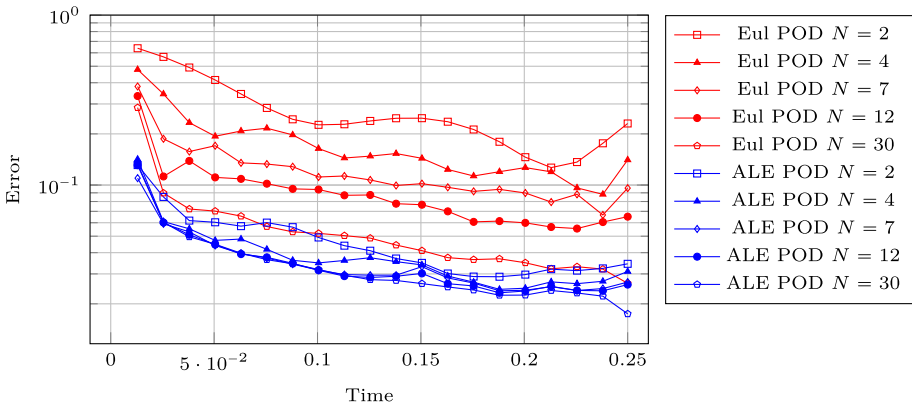
Fig. 18 DMR non parametric: Error in time of reduced methods with different number N of modes

procedure, using Algorithm 1 and the 2D geometrical transformation map T introduced in Sect. 3.2: all the details for the calibration step are summarized in Table 5. Figure 17 (right column) shows the outcome of the calibration for the density ρ, at different times: here the snapshots are represented in the reference configuration $\mathcal{R}$ (computational time for forecasting the calibration points around 0.05s). With the calibration procedure, we obtain an improvement in the rate of decay of the eigenvalues, as it is shown in Fig. 16 (blue line). To conclude, in Fig. 18 we show the behavior in time of the approximation error, between the FOM solution and the online solution (computational time to evaluate the NN for the ROM coefficients 0.001s), with or without calibration, according to the number N of modes used. Both errors have been computed in the physical domain Ω and are defined as:

$$\frac{||\rho(t) - \rho_N(t)||_{L^2(\Omega)}}{||\rho(t)||_{L^2(\Omega)}} \quad \text{and} \quad \frac{||\rho(t) - \hat{\rho}_N(t) \circ T^{-1}[\boldsymbol{w}^{\text{opt}}(t)]||_{L^2(\Omega)}}{||\rho(t)||_{L^2(\Omega)}}. \tag{13}$$

As we see in Fig. 18, the error of the online approximation (with calibration) does not go below a certain lower bound, even increasing the number of bases. We recall that, in the calibrated setting, we are interpolating the solutions to perform the transformations: for this reason, we believe that, after a certain number N of modes, the interpolation error dominates the global error, and this leads to the plateau that one can observe in the figure. We notice that with around 30 basis for the POD in the Eulerian framework we achieve errors that are comparable with the ALE solutions.

Nevertheless, the qualitatively comparison of the Eulerian and ALE approach at the ROM level with $N = 2$ for the ALE approach and $N = 30$ for the Eulerian approach depicted in
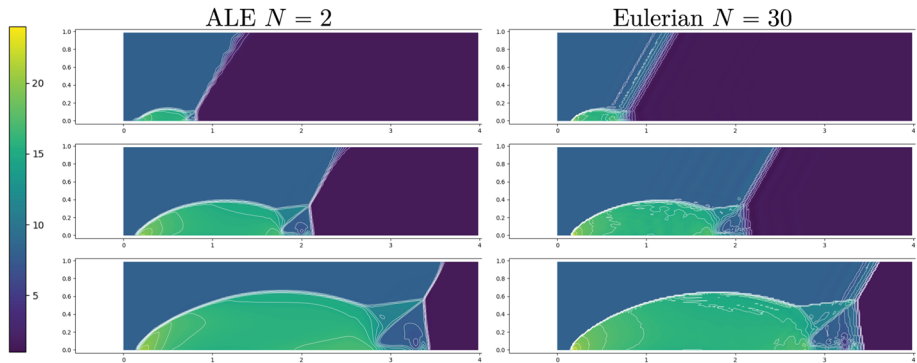
Fig. 19 DMR non parametric: ROM solutions for ρ in Ω at times 0.05 (top), 0.15 (center) and 0.25 (bottom). Left column: ALE ROM solution with $N = 2$. Right column: Eulerian ROM solution with $N = 30$. We plot in white 20 contour lines at equispaced values between 1 and 25

Fig. 19 is still in favor of the ALE approach. Indeed, the Eulerian ROM shows an oscillatory behavior that deteriorates the shape of the solution, the shock position and the flat areas, which are not anymore flat. On the contrary, the ALE ROM solutions are very similar to the FOM ones and they preserve all the original features even with a much smaller reduced basis. So, even if the L^2 errors of the two approaches are comparable, the quality of the two solutions is very different.

5.4 Parametric DMR Problem

In this section, we consider the parametric version of the 2D DMR problem: the physical parameter is the angle β introduced in Sect. 5.3; the physical parameter interval is $\mathcal{P}_{\text{phys}} = [0.1, 0.675]$, and the time interval is $[0, 0.2]$. Also in this case, the FOM snapshots have been obtained with the same FV scheme, on a mesh 600×150 (computational time of 2 h each) then downsampled to 200×50 for reduction of computational time of the offline phase. In the training set, we include for each physical parameter 51 snapshots every $\Delta t = 0.004$, and we use all the snapshots for the calibration.

Figure 20 shows some snapshots, for two different values of β and at different times, before and after the calibration: all the details for the calibration procedure are summarized in Table 5. We also depicted the control points grid on the reference domain and its transformation onto the physical one, showing how the tracking of the interesting point is done and how much distortion we can get with such transformations. As we can see from Fig. 16, also in the parametric case the calibration procedure improves significantly the rate of the decay of the eigenvalues returned by the POD and hence, ultimately, the Kolmogorov n-width of the problem under consideration. In Fig. 21, we plot the behavior of the relative error on the physical domain, as explained in Sect. 5.3 in (13), varying time and for different number of modes used in the reduced spaces. On the left, we plot the error between the FOM solution and the L^2 projection onto the reduced space; on the right, we have the error obtained using the POD-NN to predict the online solution. We see that both the Eulerian and the ALE projection errors improve as we increment the number of POD modes, with the Eulerian being always much larger. In the POD-NN error, on the other side, the decay of the error is slower and it seems to stagnate at some bottleneck values, in particular for the ALE case. That is why we aim at extending this work in the future with a hyper-reduced Galerkin projection approach,
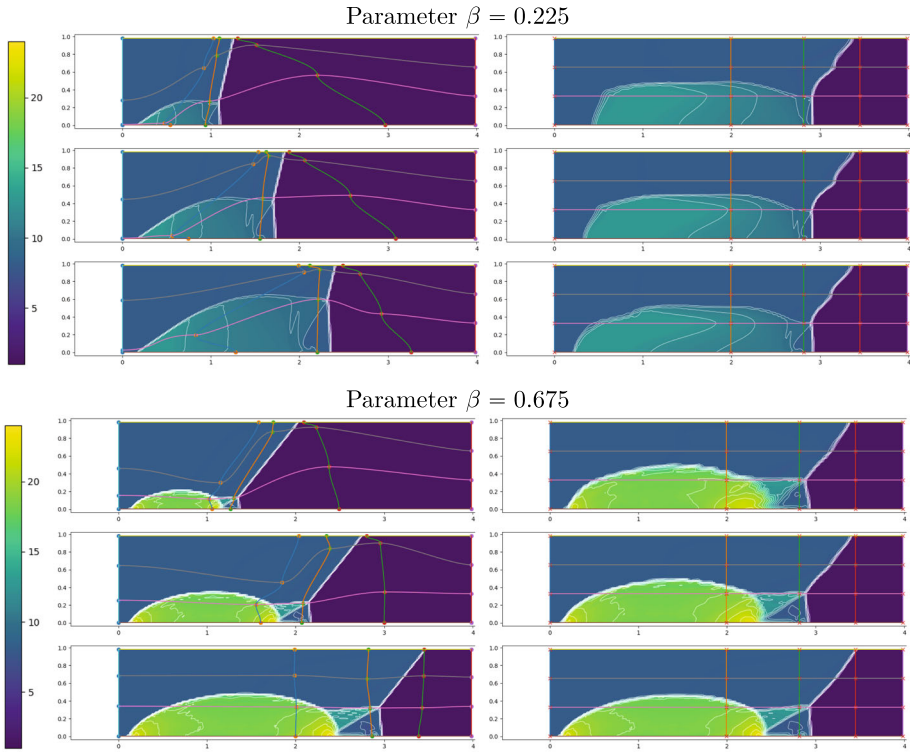
Parameter $\beta = 0.225$

Parameter $\beta = 0.675$

Fig. 20 DMR parametric FOM solution for ρ at times $t = 0.096$, $t = 0.148$ and $t = 0.2$ (top to bottom) in the physical domain Ω (left) and, after calibration, in the reference configuration $\mathcal{R}$ (right). We mark on the plots the control points and the Cartesian grid that links them. Parameter values $\beta = 0.225$ (top) and $\beta = 0.675$ (bottom). We plot in white 20 contour lines at equispaced values between 1 and 25
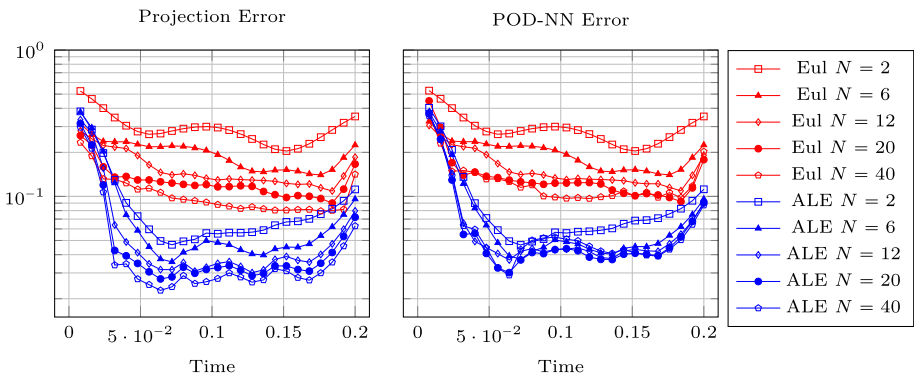
Fig. 21 DMR parametric: Error in time of reduced methods with different number N of modes. Parameter in test set $\beta = 0.675$
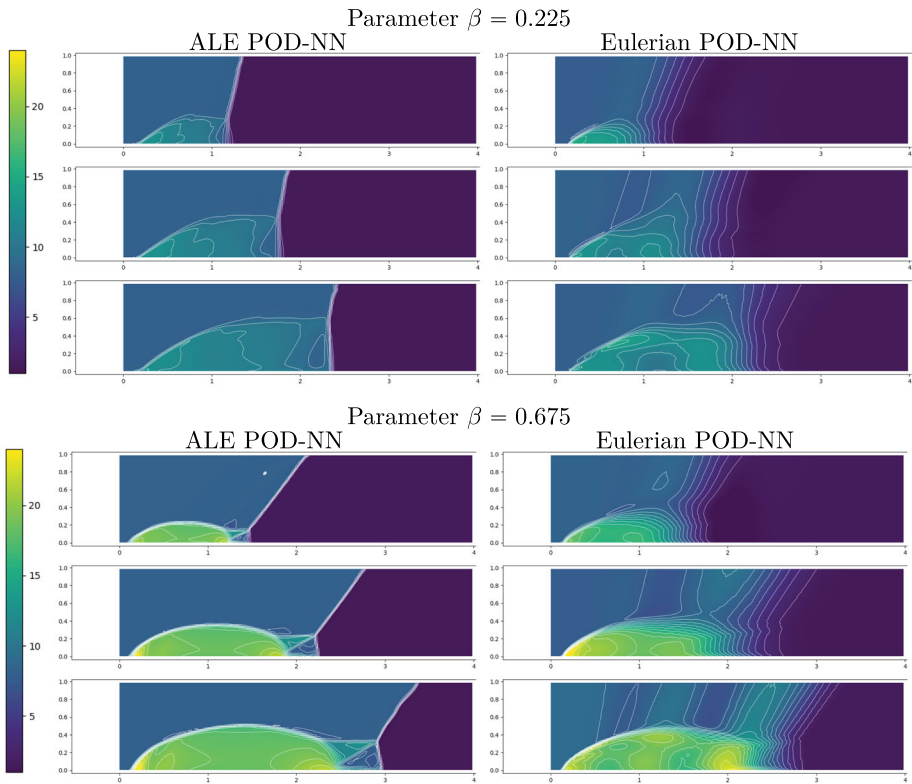
Fig. 22 POD-NN solutions with $N = 6$ of ρ for DMR parametric at times $t = 0.096$, $t = 0.148$ and $t = 0.2$ (top to bottom) in the physical domain Ω. Left column: with the calibration of the manifold. Right column: without calibration. Parameter values $\beta = 0.225$ (top) and $\beta = 0.675$ (bottom). We plot in white 20 contour lines at equispaced values between 1 and 25

to reintroduce some mathematical rigorousness hoping to decrease the online error. Finally, in Fig. 22 we represent the online solutions for $\beta = 0.225$ and $\beta = 0.675$, both with the Eulerian and the ALE approach with $N = 6$. Similarly to what happens in the non parametric test case, the Eulerian approach struggles to reproduce the FOM solution, providing an approximation that sometimes even loses the main features (the shape of the solution, the shocks, the flat areas). On the contrary, with the ALE approach, the online approximation preserves all these features. The two parameters shown validate the ability of this ROM approach to work in strongly nonlinear parametric context, where the parameters changes the solution's feature geometry, the values of the solution and vaguely the structure of the features. On the other hand, we remark that this approach works only for quasi-self-similar solutions, where we can recognize a similar structure along the parameter domain.

5.5 Triple Point Non Parametric

The applications of the proposed algorithm are various, we have showed some self-similar solutions, but this class includes a huge amount of test problems: for example, many hypersonic problems where a shock or multiple shocks are present in the solution as airfoil
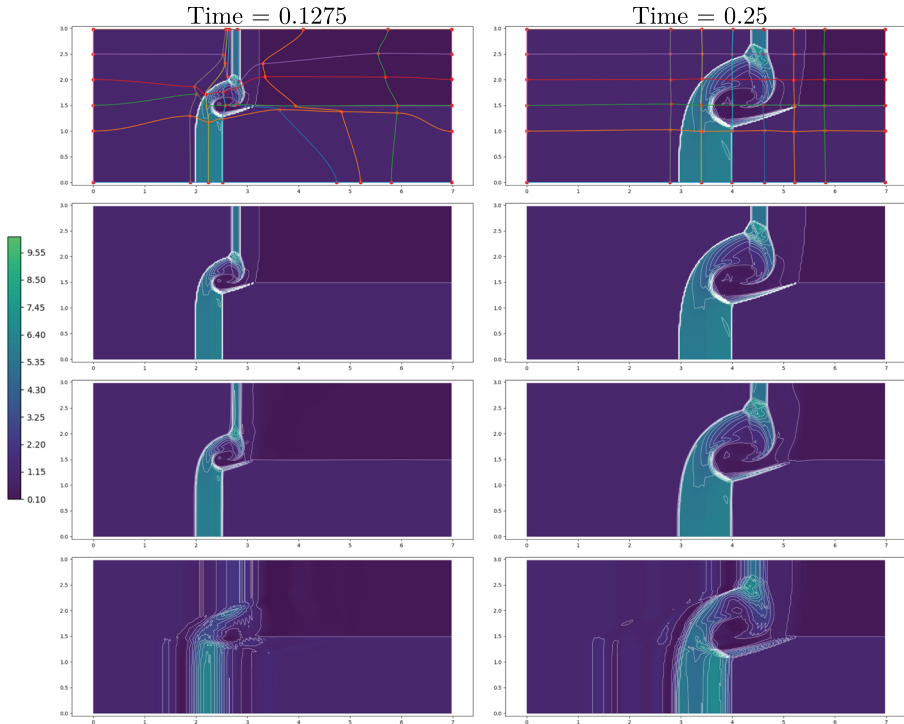
Fig. 23 FOM (first and second line), ALE POD-NN (third line) and Eulerian POD-NN (bottom) solutions with $N = 7$ of ρ at times $t = 0.1275$ (left) and $t = 0.25$ (right) in the physical domain Ω. The first line shows also the calibration points. We plot in white 20 contour lines at equispaced values between 1 and 15

simulations, water waves propagation or acoustic waves. We want to solve another test here that involves a more complicated solution structure, which is a triple point shock interaction test. We consider a physical domain $\Omega = [0, 7] \times [0, 3]$. The initial conditions are:

$$\begin{cases} (\rho_W, u_W, v_W, p_W) = (1, 20, 0, 1) & x \in [0, 1] \times [0, 3], \\ (\rho_{NE}, u_{NE}, v_{NE}, p_{NE}) = (0.125, 0, 0, 0.1) & x \in [1, 7] \times [1.5, 3], \\ (\rho_{SE}, u_{SE}, v_{SE}, p_{SE}) = (1, 0, 0, 0.1) & x \in [1, 7] \times [0, 1.5]. \end{cases} \quad (14)$$

Boundary conditions are transmissive on the right, Dirichlet with state (ρ_W, u_W, v_W, p_W) on the left and reflective at the top and bottom of the domain. Final time is set to $t_f = 0.25$. We solve the problem on a grid 2800×1200 and then we downsample it to 280×120 for the reduction of computational time in the offline phase. We have used in the training set of both calibration and reduced algorithms $N_\mu = 100$ snapshots at $\Delta t = 0.0025$.

In Fig. 23, we show the solutions of the calibration points, FOM, ALE ROM and Eulerian ROM at times $t = 0.1275$ and $t = 0.25$. The ROM solutions are obtained with a reduced basis of dimension $N = 7$. In the first FOM plots, we also plot the $M_1 \cdot M_2 = 8 \cdot 6$ optimal calibration points (the one at the final time are essentially the reference one) that vaguely surround the most dynamical area. Again, the ALE ROM performs much better than the Eulerian ROM in catching the right position of the waves and to sharply represent them. On the other side, there is a slight mistake in the calibration in catching the most right shock that is represented by a vertical contour line, which is not perfectly represented in the ALE ROM.
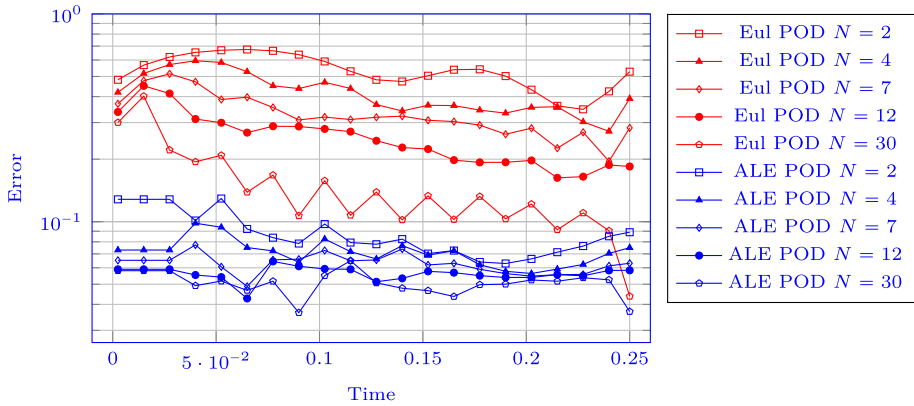
Fig. 24 Triple point non parametric: Error in time of reduced methods with different number N of modes

In Fig. 24, we observe that also here the error of the ALE approach is much smaller than the Eulerian, but that the interpolation error is probably a lower bound for the ALE ROM that keeps it from decreasing with the increasing of the reduced basis dimension. On the other side, for the Eulerian ROM the error decays increasing the reduced basis dimensions, even if with irregularity due to the spurious oscillations. All the parameters not specified are as in the DMR non-parametric case.

6 Conclusions

We presented a novel, optimization-based calibration technique suited for hyperbolic conservation laws with (quasi) self-similar solutions that present multiple travelling structures, such as discontinuities. We combined the calibration technique with an ANN based Model Order Reduction, in order to obtain a non intrusive approximation setting that is able to provide satisfying results both in the non parametric and in the parametric framework, without the use of implicit shock tracking techniques, which additionally translates into a much more limited computational effort during the offline phase. To test the proposed methodology and to show its broad range of applicability, we considered three time-dependent problems of interest: the 1D Sod shock tube problem (non parametric and parametric), the 2D DMR problem (non parametric and parametric) and the non parametric triple point problem. In all of our tests, we have shown the benefits of using the proposed calibration based MOR: this is confirmed not only by the comparison on the rate of decay of the eigenvalues returned by the PODs, but also by the behavior in time of the relative L^2-errors (in the physical domain Ω) obtained with the two approaches. To conclude, a qualitative comparison on the FOM solutions and the ROM solutions (with and without the calibration approach) is provided, in order to highlight that, by using a smaller number of modes, our proposed methodology is able to correctly capture all the important features of the full order solutions. Indeed, classical ROMs produce oscillations, smear the shocks and cannot preserve flat areas, while the presented calibrated version does, even in the context of multiple intersecting shocks and waves (such as in the triple point test).

We also showed the robustness of the calibration algorithm with respect to the choice of the reference solution $\overline{\rho}$, the initial guess $\boldsymbol{\theta}^{(0)}(\boldsymbol{\mu})$ and the order with which we perform the

calibration: all the tests have been performed for the $1D$ problem, for which an analytical solution is available. The results show that the calibration algorithm provides good results, almost independently on the choice of the reference control points. Nevertheless, we are aware that a more in depth study has to be carried out on the number of control points to choose: we envision this as a future development of the proposed work. The replacement of the Neural Networks with a purely ALE approach for the online system is a work in progress and a future extension of this present work. At the time being, the strongest limitations of our method to get more physically complicated solutions are two: the fact that the domain Ω needs to be rectangular, and the fact that the configuration of the features should not vary too much, in a topological sense. We intend to address both points in the future, introducing techniques that can map non-polyhedral shapes into rectangles [33], and using local ROMs for different parameter/time zones.

The proposed approximation setting is based on the use of piecewise cubic Hermite interpolating polynomials (or on some tensorial product of them), and works well with rectangular domains and Cartesian meshes: the extension of this approach to more complex geometries and other kinds of meshes (i.e. triangular ones) is envisioned as another future direction of this work. We also remark that, so far, we only worked with FV approximations of the full order solution. We expect to generalize the whole methodology to other discretization techniques.

Data Availibility Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Benner, P., Cohen, A., Ohlberger, M., Willcox, K.: Model reduction and approximation: theory and algorithms. Computational Science and Engineering. Society for Industrial and Applied Mathematics, Philadelphia (2017)
2. Hesthaven, J.S., Rozza, G., Stamm, B.: Certified reduced basis methods for parametrized partial differential equations. SpringerBriefs in Mathematics. Springer, Cham (2016)
3. Haasdonk, B., Ohlberger, M.: Reduced basis method for finite volume approximations of parametrized linear evolution equations. ESAIM Math. Modelling Num. Anal. **42**(2), 277–302 (2008). https://doi.org/10.1051/m2an:2008001
4. Cohen, A., DeVore, R.: Kolmogorov widths under holomorfic mappings. IMA J. Numer. Anal. **36**, 1–12 (2015). https://doi.org/10.1093/imanum/dru066
5. Melenk, J.M.: On n-widths for elliptic problems. J. Math. Anal. Appl. **247**(1), 272–289 (2000). https://doi.org/10.1006/jmaa.2000.6862
6. Greif, C., Urban, K.: Decay of the Kolmogorov N-width for wave problems. Appl. Math. Lett. **96**, 216–222 (2019). https://doi.org/10.1016/j.aml.2019.05.013
7. Ohlberger, M., Rave, S.: Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. C.R. Math. **351**(23), 901–906 (2013). https://doi.org/10.1016/j.crma.2013.10.028
8. Reiss, J., Schulze, P., Sesterhenn, J., Mehrmann, V.: The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. SIAM J. Sci. Comput. **40**(3), 1322–1344 (2018). https://doi.org/10.1137/17M1140571
9. Zorawski, B., Burela, S., Krah, P., Marmin, A., Schneider, K.: Automated transport separation using the neural shifted proper orthogonal decomposition. arXiv preprint arXiv:2407.17539 (2024)
10. Iollo, A., Lombardi, D.: Advection modes by optimal mass transfer. Phys. Rev. E **89**, 022923 (2014). https://doi.org/10.1103/PhysRevE.89.022923
11. Iollo, A., Taddei, T.: Mapping of coherent structures in parameterized flows by learning optimal transportation with Gaussian models. J. Comput. Phys. **471**, 111671 (2022). https://doi.org/10.1016/j.jcp.2022.111671
12. Khamlich, M., Pichi, F., Rozza, G.: Optimal Transport-inspired Deep Learning Framework for Slow-Decaying Problems: Exploiting Sinkhorn Loss and Wasserstein Kernel. arXiv preprint arXiv:2308.13840 (2023)
13. Battisti, B., Blickhan, T., Enchery, G., Ehrlacher, V., Lombardi, D., Mula, O.: Wasserstein model reduction approach for parametrized flow problems in porous media. ESAIM Proc. Surveys **73**, 28 (2023). https://doi.org/10.1051/proc/202373028
14. Abgrall, R., Amsallem, D., Crisovan, R.: Robust model reduction by L^1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems. Adv. Modeling Simulation Eng. Sci. **3**(1), 1 (2016). https://doi.org/10.1186/s40323-015-0055-3
15. Cagniart, N., Maday, Y., Stamm, B.: Model order reduction for problems with large convection effects. In: Chetverushkin, B.N., Fitzgibbon, W., Kuznetsov, Y.A., Neittaanmäki, P., Periaux, J., Pironneau, O. (eds.) Contributions to partial differential equations and applications, pp. 131–150. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-78325-3_10
16. Cagniart, N., Crisovan, R., Maday, Y., Abgrall, R.: Model Order Reduction for Hyperbolic Problems: a new framework. hal-01583224 (2017)
17. Mojgani, R., Balajewicz, M., Hassanzadeh, P.: Kolmogorov n-width and Lagrangian physics-informed neural networks: A causality-conforming manifold for convection-dominated PDEs. Comput. Methods Appl. Mech. Eng. **404**, 115810 (2023). https://doi.org/10.1016/j.cma.2022.115810
18. Mojgani, R., Balajewicz, M.: Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows. arXiv preprint arXiv:1701.04343 (2017)
19. Karatzas, E.N., Ballarin, F., Rozza, G.: Projection-based reduced order models for a cut finite element method in parametrized domains. Comput. Math. Appl. **79**(3), 833–851 (2020). https://doi.org/10.1016/j.camwa.2019.08.003
20. Nonino, M., Ballarin, F., Rozza, G., Maday, Y.: A reduced basis method by means of transport maps for a fluid-structure interaction problem with slowly decaying Kolmogorov n-width. Adv. Comput. Sci. Eng. **1**(1), 36–58 (2023). https://doi.org/10.3934/acse.2023002
21. Taddei, T.: A registration method for model order reduction: Data compression and geometry reduction. SIAM J. Sci. Comput. **42**(2), 997–1027 (2020). https://doi.org/10.1137/19M1271270
22. Ferrero, A., Taddei, T., Zhang, L.: Registration-based model reduction of parameterized two-dimensional conservation laws. J. Comput. Phys. **457**, 111068 (2022). https://doi.org/10.1016/j.jcp.2022.111068

23. Barral, N., Taddei, T., Tifouti, I.: Registration-based model reduction of parameterized PDEs with spatio-parameter adaptivity. J. Comput. Phys. **499**, 112727 (2024). https://doi.org/10.1016/j.jcp.2023.112727
24. Peherstorfer, B.: Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. SIAM J. Sci. Comput. **42**(5), 2803–2836 (2020). https://doi.org/10.1137/19M1257275
25. Mirhoseini, M.A., Zahr, M.J.: Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. J. Comput. Phys. **473**, 111739 (2023). https://doi.org/10.1016/j.jcp.2022.111739
26. Rim, D., Moe, S., LeVeque, R.: Transport reversal for model reduction of hyperbolic partial differential equations. SIAM/ASA J. Uncertainty Quantif. **6**(1), 118–150 (2018). https://doi.org/10.1137/17M1113679
27. Rim, D., Mandli, K.T.: Displacement interpolation using monotone rearrangement. SIAM/ASA J. Uncertainty Quantif. **6**(4), 1503–1531 (2018). https://doi.org/10.1137/18M1168315
28. Lee, K., Carlberg, K.T.: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J. Comput. Phys. **404**, 108973 (2020). https://doi.org/10.1016/j.jcp.2019.108973
29. Fresca, S., Dedé, L., Manzoni, A.: A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. J. Sci. Comput. (2021). https://doi.org/10.1007/s10915-021-01462-7
30. Pichi, F., Moya, B., Hesthaven, J.S.: A graph convolutional autoencoder approach to model order reduction for parametrized PDEs. J. Comput. Phys. **501**, 112762 (2024). https://doi.org/10.1016/j.jcp.2024.112762
31. Romor, F., Torlo, D., Rozza, G.: Friedrichs' systems discretized with the Discontinuous Galerkin method: domain decomposable model order reduction and Graph Neural Networks approximating vanishing viscosity solutions. arXiv preprint arXiv:2308.03378 (2023)
32. Klein, R., Sanderse, B., Costa, P., Pecnik, R., Henkes, R.: Entropy-stable model reduction of one-dimensional hyperbolic systems using rational quadratic manifolds. arXiv preprint arXiv:2407.12627 (2024)
33. Taddei, T.: Compositional maps for registration in complex geometries. arXiv preprint arXiv:2308.15307 (2023)
34. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer, Berlin Heidelberg (2013)
35. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**(1), 115–173 (1984). https://doi.org/10.1016/0021-9991(84)90142-6
36. Torlo, D.: Model reduction for advection dominated hyperbolic problems in an ALE framework: Offline and online phases. arXiv preprint arXiv:2003.13735 (2020)
37. Cucchiara, S., Iollo, A., Taddei, T., Telib, H.: Model order reduction by convex displacement interpolation. J. Comput. Phys. (2024). https://doi.org/10.1016/j.jcp.2024.113230
38. Demo, N., Tezzele, M., Rozza, G.: EZyRB. https://mathlab.github.io/EZyRB/index.html
39. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
40. Richter, T.: Fluid-structure Interactions: Models, Analysis and Finite Elements. Lecture Notes in Computational Science and Engineering. Springer, Cham (2017)
41. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C.R. Math. **339**(9), 667–672 (2004). https://doi.org/10.1016/j.crma.2004.08.006
42. Yano, M.: Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. Adv. Comput. Math. **45**, 2287–2320 (2019). https://doi.org/10.1007/s10444-019-09710-z
43. Núñez-de la Rosa, J.: High-order finite volume solver for the magnetohydrodynamics equations. https://github.com/jbnunezd/fv-solver-mhd (2020)
44. Amsallem, D., Zahr, M.J., Farhat, C.: Nonlinear model order reduction based on local reduced-order bases. Int. J. Numer. Meth. Eng. **92**(10), 891–916 (2012). https://doi.org/10.1002/nme.4371

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.