MODEL REDUCTION OF TIME-DEPENDENT HYPERBOLIC EQUATIONS USING COLLOCATED RESIDUAL MINIMISATION AND SHIFTED SNAPSHOTS

NEERAJ SARNA * AND SARA GRUNDEL †

Abstract. We develop a non-linear approximation for solution manifolds of parametrised time-dependent hyperbolic PDEs. Our non-linear approximation space is a span of snapshots evaluated on a transformed spatial domain. We compute a solution in the non-linear approximation space using residual minimisation. We reduce the cost of residual minimisation by minimising and evaluating the residual on a set of collocation points. We decompose the collocation points computation into an offline and an online phase. The offline phase computes the collocation points for a set of training parameters by minimising a bound on the L^2-error of the reduced-order model. Moreover, the online phase transports the collocation points computed offline. Our hyper-reduction is general in the sense that it does not assume a specific form of the spatial transform. As a particular instance of the non-linear approximation space, we consider a span of shifted snapshots. We consider shifts that are local in the time-parameter domain and propose an efficient algorithm to compute the same. Our shift computation is data-driven in the sense that it only requires the snapshots and not the underlying characteristic speeds of the hyperbolic equation. Several benchmark examples involving single and multi-mode transport demonstrate the effectiveness and the limitations of our method.

1. Introduction

We consider the evolution equation

(1.1)
$$\partial_t u(x,t,\mu) = L(u(x,t,\mu),\mu) \text{ for } (x,t,\mu) \in \Omega \times [0,T] \times \mathcal{P}, \quad u(x,0,\mu) = u_0(x,\mu) \text{ for } (x,\mu) \in \Omega \times \mathcal{P},$$
$$\mathcal{G}(u(x,t,\mu),\mu) = 0 \text{ for } (x,t,\mu) \in \partial\Omega \times [0,T] \times \mathcal{P},$$

where $\mathcal{P} \subset \mathbb{R}$ is a bounded parameter domain, T is the final time, $u_0(\cdot,\mu)$ is the initial data, and $\Omega \subset \mathbb{R}^d$ is a bounded and open spatial domain. Moreover, the operator $\mathcal{G}(\cdot,\mu)$ prescribes some boundary conditions. We denote the spatial regularity by $u(\cdot,t,\mu) \in \mathcal{X}$, and we consider scalar equations i.e., $u(x,t,\mu) \in \mathbb{R}$. For hyperbolic problems, the evolution operator $L(\cdot,\mu) : \mathbb{R} \to \mathbb{R}$ is of the form

$$L(\cdot,\mu) = -\nabla \cdot f(\cdot,\mu), \tag{1.2}$$

where $f(\cdot,\mu) : \mathbb{R} \to \mathbb{R}^d$ is the flux function and ∇ represents a spatial gradient. To discretize the evolution equation we consider discrete time steps $0 = t_1 < t_2 < \cdots < t_K = T$ where at each time instance, we approximate $u(\cdot,t_k,\mu)$ by $u^N(\cdot,t_k,\mu) \in \mathcal{X}^N$. Here, $\mathcal{X}^N \subset \mathcal{X}$ is N-dimensional and is (usually) of the finite-element or finite-volume type. For any $k \in \{1,\ldots,K\}$, we refer to $u^N(\cdot,t_k,\mu)$ as the full-order model (FOM).

The space $\mathcal{X}^N$ is high-dimensional resulting in an expensive computation for applications where the evolution equation must be solved for different parameter instances. This motivates one to look for a reduced-order model (ROM) where one approximates the FOM by $u^n(\cdot,t_k,\mu) \in \mathcal{X}^n$ with $\dim(\mathcal{X}^n) = n \ll N$. The low-dimensionality of $\mathcal{X}^n$ allows for a possibility of fast computations. We emphasize that the availability of a low-dimensional approximation space is not sufficient to ensure the computational efficiency of a ROM. Further reduction might be needed to make the ROM efficient [2, 5, 6, 9]. Indeed, this will also be the case for the approximation space we propose.

In most ROMs, the solution manifold (i.e. the set $\{u(\cdot,t,\mu) : (t,\mu) \in [0,T] \times \mathcal{P}\}$) of the evolution equation (1.1) is approximated in a linear space $\mathcal{X}^n$. See, for example, [3, 14, 24, 26]. The approximability of a solution manifold in a linear space is quantified by its Kolmogorov n-width. The faster the decay in the Kolmogorov width with n, better is the approximability of a solution manifold in a n-dimensional linear space. For parameterised elliptic and parabolic problems, the Kolmogorov width decays exponentially, allowing for (acceptably) low-dimensional and accurate linear approximations [7]. However, for hyperbolic equations, the Kolmogorov n-width decay slowly. One example is the 1D wave equation where the Kolmogorov width is bounded from below by $\mathcal{O}(1/\sqrt{n})$; see [12] for a proof. A slow Kolmogorov width decay makes a linear approximation unappealing because it forces one to choose a large value of n for acceptable accuracy.

Poor accuracy of a linear approximation motivates the search for a non-linear approximation. To construct a non-linear approximation space, one can introduce (μ,t)-locality in the linear space $\mathcal{X}^n$.

*Corresponding author, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr 1, 39106, Magdeburg, Germany,sarna@mpi-magdeburg.mpg.de

†Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr 1, 39106, Magdeburg, Germany,grundel@mpi-magdeburg.mpg.de

We denote such a space by $\mathcal{X}^n_{\mu,t}$. Such locality can allow one to capture the characteristic wave speeds of a hyperbolic equation that are local in the (μ, t)-domain. The use of a non-linear approximation space (for hyperbolic problems) first appeared in [25] where authors considered a shifted KL-expansion with the shift accounting for the transport in the solution. Similar idea was explored in [19] where authors consider the decomposition $\mathcal{X}^n_{\mu,t} = \{g(\mu, t) \cdot v(\mu, t)\}$, with $g(\mu, \cdot)$ being an element of a Lie-group and $v(\mu, \cdot)$ being a template function. Replacing $g(\mu, t) \cdot v(\mu, t)$ into the evolution equation (1.1), the resulting equations for $g(\mu, \cdot)$ and $v(\mu, \cdot)$ are closed with phase conditions and then reduced using empirical operator interpolation. Authors in [4] consider a different approach and construct $\mathcal{X}^n_{\mu,t}$ by (i) reducing the Kolmogorov width with a (μ, t)-dependent spatial transform; (ii) performing POD on the transformed spatial domain; (iii) transforming the POD modes back to the original spatial domain; and (iv) performing residual minimisation with the transformed POD modes. Similarly in [18], for steady-state problems, $\mathcal{X}^n_\mu$ is the span of snapshots evaluated on a transformed spatial domain with the transformation being a polynomial in μ and a Fourier series expansion in x. Some works develop the approximation space online (thus skipping the offline phase altogether) and this, by construction, can lead to a non-linear approximation [11, 20]. Works in [22, 30] consider a non-linear interpolation for solutions to hyperbolic problems where the snapshots computed offline are non-linearly interpolated (with Lagrange polynomials for example) online. This results in the evolution equation being used only in the offline phase, which leads to computational efficiency.

Following the work in [18, 25, 30], we consider a $\mathcal{X}^n_{\mu,t}$ that is a span of snapshots evaluated on a transformed spatial domain. We find a solution in $\mathcal{X}^n_{\mu,t}$ using residual minimisation [1, 2, 5, 18]. The residual computation loops over all the mesh elements (or all the degrees of freedom of $\mathcal{X}^N$), which increases the online computation cost of a ROM. To reduce this cost, we evaluate and minimise the residual on a subset of mesh points, or the so-called collocation points. We develop an L^2-error bound for our ROM and, offline, we use it to compute the collocation points for a set of training parameters. We then transport these collocation points during the online phase. Our algorithm for the computation of the collocation points does not assume a specific form of the spatial transform and thus, can be applied to various other spatial transforms developed in [18, 22, 30].

As a particular instance of the non-linear space $\mathcal{X}^n_{\mu,t}$, we consider the span of shifted snapshots. With the shifts in the snapshots, we capture the dominant "transport" mode in the solution, and with a linear combination of the shifted snapshots, we capture the "shape" change in the solution. We consider spatial shifts that are local in the (μ, t)-domain. We compute the shifts using residual minimisation, where, we minimise the residual over shifts that align shocks and local minima/maxima in the snapshots. While doing so, we do not use the characteristic speeds of the hyperbolic equation (1.1). This results in a data-driven algorithm that computes the shifts using solely the solution snapshots. Such an algorithm is desirable for problems with a spatial dimension greater than one where using the characteristic speeds are difficult to compute [30].

The article is structured as follows. The second and the third section present the FOM and the ROM, respectively. The fourth section presents our algorithm for hyper-reduction. The fifth section discusses the computation of the spatial transform. The sixth section discusses the relation to the relevant previous works, and the seventh section presents our numerical results.

2. Full-Order Model (FOM)

For simplicity of exposition, in the following sections, we assume a one-dimensional domain i.e., $d = 1$ in (1.1). The coming discussion clarifies that an extension to multi-dimensions is straightforward. Later, we also consider numerical example involving a multi-dimensional spatial domain. Let $\Omega = [x_{\min}, x_{\max}]$. Let $\{\mathcal{I}^x_i\}_{i=1,\dots,n_x}$ be a discretization of Ω with $\mathcal{I}^x_i = [x_{\min} + (i-1)\Delta x, x_{\min} + i \times \Delta x]$ and $n_x \Delta x = x_{\max} - x_{\min}$. For the spatial and the temporal discretization of the evolution equation (1.1), we consider a cell-centred finite-volume (FV) scheme and an explicit-Euler scheme, respectively. For a FV scheme, $\mathcal{X}^N$ is a span of scaled characteristic functions of the set $\mathcal{I}^x_i$, and reads

$$L^2(\Omega) \supset \mathcal{X}^N = \text{span}\{\phi_i \ : \ \phi_i = \frac{1}{\sqrt{\Delta x}} \mathbb{1}_{\mathcal{I}^x_i}, \ i \in \{1, \dots, N\}\}. \tag{2.1}$$

Above, $\mathbb{1}_A$ represents a characteristic function of the set $A \subset \mathbb{R}$, and $N = n_x$. Note that ϕ_i's are $L^2(\Omega)$ orthogonal. For convenience, we collect all these basis functions in a vector $\Phi(x) \in \mathbb{R}^N$ such that

$$(\Phi(x))_i = \phi_i(x), \quad i \in \{1, \dots, N\}. \tag{2.2}$$

Using $\mathcal{X}^N$, we express the evolution equation for the FOM as

$$\left\langle \Phi, u^N(\cdot, t_{k+1}, \mu) \right\rangle = \left\langle \Phi, u^N(\cdot, t_k, \mu) \right\rangle + \Delta t \times \left\langle \Phi, L^N(u^N(\cdot, t_k, \mu), \mu) \right\rangle, \ \forall k \in \{1, \dots, K-1\}, \tag{2.3}$$

where $L^N : \mathcal{X}^N \to \mathcal{X}^N$ is a finite-volume discretization of the operator L, and the initial condition is given as $\left\langle \Phi, u^N(\cdot, t_1 = 0, \mu) \right\rangle = \langle \Phi, u_0(\cdot, \mu) \rangle$. Moreover, for any $w \in L^2(\Omega)$, we interpret $\langle \Phi, w \rangle$ as $(\langle \phi_1, w \rangle_{L^2(\Omega)}, \dots, \langle \phi_N, w \rangle_{L^2(\Omega)})^T$. We include a discretization of the boundary operator $\mathcal{G}$ in L^N. For simplicity, we consider a constant μ-independent time-step of $\Delta t > 0$. Between the discrete time-steps $\{t_k\}_{k=1,\dots,K}$, we extend $u^N(\cdot, t_k, \mu)$ by a constant i.e., $u^N(\cdot, t, \mu) = u^N(\cdot, t_k, \mu)$ for all $t \in [t_k, t_{k+1})$.

The operator L^N relies on a numerical flux, which we consider to be a local Lax-Friedrich (LLF) flux. An explicit form of the flux is not of importance here but can be found in [15]. With the LLF flux, and for a sufficiently regular initial and boundary data, the $L^2(\Omega)/L^\infty(\Omega)$-stability of the discretization (2.3) is ensured by choosing (see [10, 27])

$$\Delta t \leq \frac{\Delta x}{\max\limits_{(x,t,\mu) \in \Omega \times [0,T] \times \mathcal{P}} |f'(u^N(x, t, \mu), \mu)|}, \tag{2.4}$$

where $f'(u^N(x, t, \mu), \mu)$ represents the derivative of $f(\cdot, \mu)$ evaluated at $u^N(x, t, \mu)$. During numerical experiments (in section 7), we consider a one-dimensional parameter domain where $|f'(u^N(x, t, \mu), \mu)|$ is monotonic in μ. This ensures the above bound for μ equals the maximum or the minimum parameter value.

3. Reduced-Order Model (ROM)

In the following discussion, we develop a reduced approximation for the FOM where the approximation space is different for each μ and t_k, and is denoted by $\mathcal{X}^n_{\mu,t_k}$.

3.1 Approximation Space for the ROM

Consider a 1D parameter domain. An extension to multi-dimensions is possible using tensor products discussed in [29]. We partition the parameter domain as

$$\mathcal{P} = \bigcup_{i=1}^{n_\mu} \mathcal{I}_i \quad \text{where} \quad \mathcal{I}_i = [\mu_i, \mu_{i+1}]. \tag{3.1}$$

We will refer to $\mathcal{I}_i$ as the parameter elements. For every parameter $\mu \in \mathcal{P}$, there exists a $m \in \{1, \dots, n_\mu\}$ such that $\mu \in \mathcal{I}_m$. For convenience, we define

$$m = \gamma(\mu). \tag{3.2}$$

We do not have a restriction on the value of n_μ or the size of $\mathcal{I}_i$. In the numerical experiments, we choose some value for n_μ and consider parameter elements of the same size. Such a choice can also be made with a greedy-algorithm, which requires an a-posteriori error bounds. We postpone the development of such an a-posteriori error bound to our future works.

We motivate our approximation space with the help of an example, similar examples can be found in [4, 22, 30]. Consider the manifold $\mathcal{M} := \{f(\cdot, \mu) \ : \ \mu \in \mathcal{P}\} \subset L^2(\mathbb{R})$, where $f(\cdot, \mu)$ is a step function that scales and shifts to the right, and is given as

$$f(x, \mu) := \begin{cases} 1 + \mu, & x \leq \mu \\ 0, & x > \mu \end{cases}, \quad \mu \in \mathcal{P} := [0, 1].$$

Let $\{f(\cdot, \hat{\mu}_i)\}_{i=1,\dots,n}$, where $\hat{\mu}_i \in \mathcal{P}$, represent a set of n snapshots taken from $\mathcal{M}$; some of these snapshots are shown in Fig 1a. With a reduced-basis type approach, we can approximate $f(\cdot, \mu)$ (where $\mu \notin \{\hat{\mu}_i\}_{i=1,\dots,n}$) in the span of these snapshots. Such a span is linear and with $\mathcal{M}$ having a slow decaying Kolmogrov n-width (see [20]), we require a large value of n for acceptable accuracy. Now instead of the manifold $\mathcal{M}$, consider the following manifold $\mathcal{M}_\mu$ that consists of all the step functions shifted such that their discontinuities are aligned with the discontinuity in $f(\cdot, \mu)$

$$\begin{aligned} \mathcal{M}_\mu :=& \{f(\varphi(\cdot, \mu, \hat{\mu}), \hat{\mu}) \ : \ \varphi(\cdot, \mu, \hat{\mu}) = x - (\mu - \hat{\mu}), \ \hat{\mu} \in \mathcal{P}\}, \\ =& \{\alpha f(\cdot, \mu) \ : \ \alpha \in [1, 2]\}. \end{aligned}$$

The snapshots taken from the manifold $\mathcal{M}_\mu$ are shown in Fig 1b. From the above definition of $\mathcal{M}_\mu$ we conclude that $f(\cdot,\mu)$ is well-approximated in the span of a single snapshot (different from $f(\cdot,\mu)$) taken from $\mathcal{M}_\mu$.[1] In the terminology of [4], the spatial transform $\varphi(\cdot,\mu,\hat{\mu})$ calibrates the manifold $\mathcal{M}$ such that the snapshots from the resulting manifold $\mathcal{M}_\mu$ better approximate $f(\cdot,\mu)$.

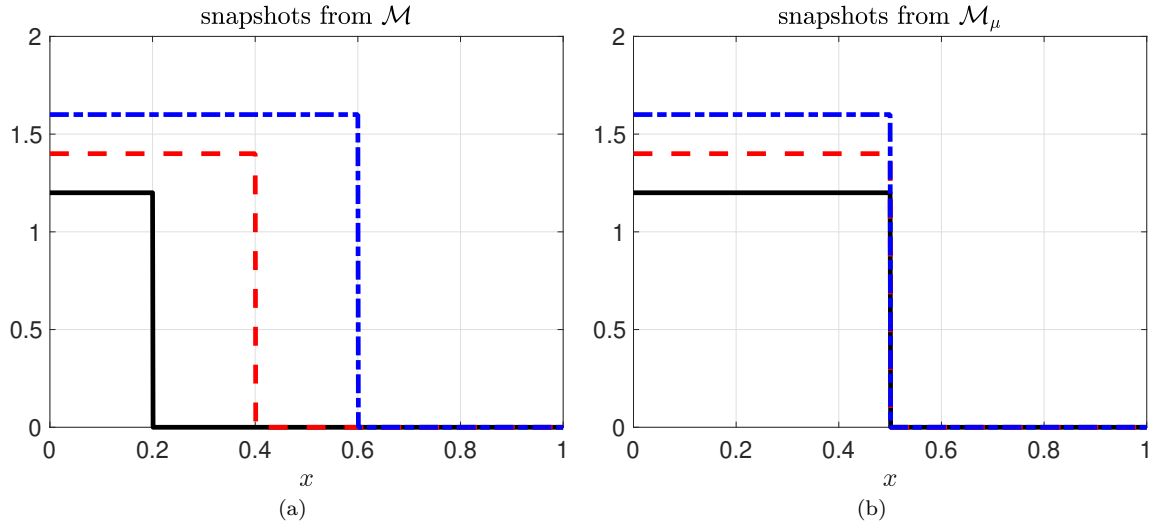


Figure 1: *Snapshots taken from (a) $\mathcal{M}$ and (b) $\mathcal{M}_\mu$.*

From previous numerical experiments and theoretical results (for e.g. see [18, 20, 30]) we know that $u^N(\cdot,t_k,\mu)$ is poorly approximated in the span of snapshots taken from the manifold $\mathcal{M}_{t_k} := \{u^N(\cdot,t_k,\hat{\mu}) \ : \ \hat{\mu}\in\mathcal{P}\}$. However, similar to [4, 30], we assume that there exists a spatial transform $\varphi(\cdot,\mu,\hat{\mu},t_k):\Omega\to\Omega$ that calibrates $\mathcal{M}_{t_k}$ such that $u^N(\cdot,t_k,\mu)$ is better approximated in the span of snapshots collected from

$$\mathcal{M}_{\mu,t_k} := \{u^N(\varphi(\cdot,\mu,\hat{\mu},t_k),t_k,\hat{\mu}) \ : \ \hat{\mu}\in\mathcal{P}\}. \tag{3.3}$$

To ensure that $u^N(\cdot,t_k,\mu)\in\mathcal{M}_{\mu,t_k}$, we require

$$\varphi(\cdot,\mu,\mu,t_k) = \mathrm{Id}, \tag{3.4}$$

where Id is an identity operator. We will compute $\varphi(\cdot,\mu,\hat{\mu},t_k)$ such that the above property is satisfied. Note that compared to our example, the additional time-dependency in the spatial transform accounts for the time-dependency in the solution.

To collect snapshots from the manifold $\mathcal{M}_{\mu,t_k}$, we use the observation made in [1, 2, 18] as per which snapshots corresponding to the parameters that are in the neighbourhood of μ are sufficient to approximate $u^N(\cdot,t_k,\mu)$. Therefore, we take snapshots at $M\geq 2$ different parameters in $\mathcal{I}_{\gamma(\mu)}$ given as $\mu_{\gamma(\mu)}\leq\hat{\mu}_1<\hat{\mu}_2<\cdots<\hat{\mu}_M\leq\mu_{\gamma(\mu)+1}$, where γ is defined in (3.2). Approximating $u^N(\cdot,t_k,\mu)$ in the span of these snapshots provides

$$\begin{aligned} &u^N(\cdot,t_k,\mu)\approx u^n(\cdot,t_k,\mu)\in\mathcal{X}^n_{\mu,t_k},\\ \text{where}\quad &\mathcal{X}^n_{\mu,t_k} := \mathrm{span}\{\psi^j_{\mu,t_k} \ : \ \psi^j_{\mu,t_k} = u^N(\varphi^M(\cdot,\mu,\hat{\mu}_j,t_k),t_k,\hat{\mu}_j),\ j\in\{1,\ldots,M\}\}. \end{aligned} \tag{3.5}$$

Above, $\varphi^M(\cdot,\mu,\hat{\mu}_j,t_k)$ is an approximation to $\varphi(\cdot,\mu,\hat{\mu}_j,t_k)$, and is given as follows. For some $\hat{\mu}_j\in\{\hat{\mu}_i\}_{i=1,\ldots,M}$, consider the function $\mu\mapsto\varphi(x,\mu,\hat{\mu}_j,t_k)$. Assume that we can compute this function for all $\mu\in\{\hat{\mu}_i\}_{i=1,\ldots,M}$ then we can approximate it using Lagrange interpolation as [30]

$$\varphi(x,\mu,\hat{\mu}_j,t_k)\approx\varphi^M(x,\mu,\hat{\mu}_j,t_k) := \sum_{i=1}^{M} l_{i-1}(\mu)\varphi(x,\hat{\mu}_i,\hat{\mu}_j,t_k). \tag{3.6}$$

[1] Indeed, the best L^2-approximation of $f(\cdot,\mu)$ in the span of a single snapshot ($\neq f(\cdot,\mu)$) from $\mathcal{M}_\mu$ is $f(\cdot,\mu)$.

Above, l_i is an i-th order Lagrange polynomial. We will refer to the different $\varphi(x, \hat{\mu}_i, \hat{\mu}_j, t_k)$ as the snapshots of the spatial transform. Later (in section 5) we assume a specific form for $\varphi(\cdot, \hat{\mu}_i, \hat{\mu}_j, t_k)$ and discuss its computation.

In our numerical experiments, we consider parameters $\{\hat{\mu}_i\}_{i=1,\dots,M}$ that are uniformly placed inside $\mathcal{I}_{\gamma(\mu)}$. With an a-posteriori error indicator, one can also choose the set $\{\hat{\mu}_i\}_{i=1,\dots,M}$ using the different techniques outlined in [8, 9, 16, 28]. The performance of each of these techniques changes with the test-case and the approximation space. Although an interesting question in its own right, we do not study the influence of these techniques upon the approximation quality of $\mathcal{X}^n_{\mu,t_k}$.

REMARK 1. *In (3.6), we approximate $\varphi(x, \cdot, \hat{\mu}_j, t_k)$ in the linear space $\text{span}\{l_i\}_{i=0,\dots,(M-1)}$, which is in contrast to our non-linear approximation for $u^N(\cdot, t_k, \mu)$. Current literature and this article offers no solution to problems (if they exist) where the spatial transform could also require a non-linear approximation.*

REMARK 2. *One can consider a different polynomial degree than M to approximate the spatial transform in (3.6). For simplicity, we consider this polynomial degree to be M.*

3.2 Residual Minimisation We compute a solution in $\mathcal{X}^n_{\mu,t_k}$ using residual-minimisation. Writing the finite-volume scheme (2.3) as a residual minimisation problem provides

$$u^N(\cdot, t_{k+1}, \mu) = \underset{w \in \mathcal{X}^N}{\arg\min} \|\text{Res}_{t_k}(w, u^N(\cdot, t_k, \mu))\|_{\mathbb{R}^N}, \quad \forall k \in \{0, \dots, K-1\}, \tag{3.7}$$

where $\mathcal{X}^N$ is the high-dimensional finite volume space given in (2.1). The residual $\text{Res}_{t_k} : \mathcal{X}^N \times \mathcal{X}^N \to \mathbb{R}^N$ follows from (2.3) and is given as

$$\text{Res}_{t_k}(w, v) := \begin{cases} \langle \Phi, w \rangle - \langle \Phi, v \rangle - \Delta t \times \left\langle \Phi, L^N(v, \mu) \right\rangle, & k \in \{1, \dots, K-1\}, \\ \langle \Phi, w \rangle - \langle \Phi, u_0(\cdot, \mu) \rangle, & k = 0. \end{cases} \tag{3.8}$$

All the other quantities are as given in (2.3). For simplicity of notation, in the following discussion we suppress the time dependency of Res_{t_k}.

Motivated from the residual formulation (3.7) of the FV scheme (2.3), we approximate $u^N(\cdot, t_{k+1}, \mu)$ in $\mathcal{X}^n_{\mu,t_{k+1}}$ by minimising the $\mathbb{R}^N$-norm of the residual. Similar to the formulations in [1, 2], this provides

$$u^N(\cdot, t_{k+1}, \mu) \approx u^n(\cdot, t_{k+1}, \mu) = \underset{w \in \mathcal{X}^n_{\mu,t_{k+1}}}{\arg\min} \|\text{Res}(\Pi_{\mu,t_{k+1}} w, \Pi_{\mu,t_k} u^n(\cdot, t_k, \mu))\|_{\mathbb{R}^N}, \tag{3.9}$$

where $\Pi_{\mu,t_k} : \mathcal{X}^n_{\mu,t_k} \to \mathcal{X}^N$ is a projection operator defined later, and $k \in \{0, \dots, K-1\}$. For convenience, we express the minimisation problem (3.9) in a matrix-vector product form. Every $w \in \mathcal{X}^n_{\mu,t_k}$ has the form $w(x) = \langle \alpha, \Psi_{\mu,t_k}(x) \rangle_{\mathbb{R}^M}$, where $\alpha \in \mathbb{R}^M$ contains the expansion coefficients, and $\Psi_{\mu,t_k}(x) \in \mathbb{R}^M$ is a vector containing all the basis functions given in (3.5). By substituting this expression for w into the minimisation problem (3.9) we find

$$\alpha_{\mu,t_{k+1}} = \underset{\alpha \in \mathbb{R}^M}{\arg\min} \|A_{\mu,t_{k+1}} \alpha - b_{\mu,t_k}\|_{\mathbb{R}^N}, \quad \forall k \in \{0, \dots, K-1\}, \tag{3.10}$$

where $\alpha_{\mu,t_{k+1}} \in \mathbb{R}^M$ and is such that $u^n(x, t_{k+1}, \mu) = \left\langle \alpha_{\mu,t_{k+1}}, \Psi_{\mu,t_{k+1}}(x) \right\rangle_{\mathbb{R}^M}$. The matrix $A_{\mu,t_{k+1}} \in \mathbb{R}^{N \times M}$ and the vector $b_{\mu,t_k} \in \mathbb{R}^N$ are defined as

$$\begin{aligned} A_{\mu,t_{k+1}} &:= \left(\left\langle \Phi, \Pi_{\mu,t_{k+1}} \psi^1_{\mu,t_{k+1}} \right\rangle, \dots, \left\langle \Phi, \Pi_{\mu,t_{k+1}} \psi^M_{\mu,t_{k+1}} \right\rangle \right), \\ b_{\mu,t_k} &:= \langle \Phi, \Pi_{\mu,t_k} u^n(\cdot, t_k, \mu) \rangle + \Delta t \times \left\langle \Phi, L^N(\Pi_{\mu,t_k} u^n(\cdot, t_k, \mu), \mu) \right\rangle, \end{aligned} \tag{3.11}$$

with $\psi^j_{\mu,t_{k+1}}$ as given in (3.5). Note that the definition of Res given in (3.8) implies $b_{\mu,t_0} = \langle \Phi, u_0(\cdot, \mu) \rangle$. Our later definition of $\Pi_{\mu,t_{k+1}}$ (given in subsection 5.3) will clarify that it is cheap to compute the inner-products appearing in the definition of $A_{\mu,t_{k+1}}$ and b_{μ,t_k}.

The dimension of the approximation space $\mathcal{X}^n_{\mu,t_k}$ is M, where we expect $M \ll N = \dim(\mathcal{X}^N)$. Despite of this, in terms of the computational efficiency, we do not gain much by computing a ROM using residual minimisation (3.10). Indeed, currently, our ROM is more expensive than the FOM. This is mainly because for the ROM, we require an online computation of $A_{\mu,t_{k+1}}$ and b_{μ,t_k}, both of which require a loop over all the basis functions in Φ. In addition, one needs to solve the least-squares problem in (3.10). In comparison, the FOM only requires a computation of b_{μ,t_k}. Later in section 4, we reduce the computational cost of the minimisation problem using hyper-reduction.

REMARK 3. *Although some authors consider residual minimisation in the L^2-norm (see [5, 23, 25]), recent results (see [1, 18]) indicate that using the L^1-norm could provide better accuracy. Ease of implementation and computational efficiency motivates our choice of the L^2-norm. Numerical results we present later might improve by using the L^1-norm or some other metric. Comparing the performance of different norms is not the goal here.*

3.3 Summary of the algorithm: Algorithm 3.1 presents the offline phase of the algorithm. `Line-1` is self-explanatory. `Line-2` provides all the spatial transforms required to construct the Lagrange interpolation given in (3.6). `Line-3` computes the offline phase of hyper-reduction and will be clearer later. Algorithm 3.2 presents the online phase of the algorithm. `Line-1/3` are self-explanatory. `Line-2` performs the online phase of hyper-reduction and will be clearer later.

Algorithm 3.1 Offline Phase: Algorithm for model reduction

1: For each $\mathcal{I}_i$ given in (3.1), compute the FOM for all $\mu \in \{\hat{\mu}_j\}_{j=1,\dots,M}$ using the time-evolution scheme given above in section 2.
2: For each $\mathcal{I}_i$, compute all the snapshots of the spatial transforms $\{\varphi(x, \hat{\mu}_j, \hat{\mu}_l, t_k)\}_{j,l=1,\dots,M}$ for all $k \in \{1, \dots, K\}$. Details are discussed below in section 5.
3: Perform the offline phase of hyper-reduction. Details are discussed below in section 4.

Algorithm 3.2 Online Phase: Algorithm for model reduction

1: For a given μ, approximate $\{\varphi(x, \mu, \hat{\mu}_j, t_k)\}_{j=1,\dots,M}$ using polynomial interpolation (3.6).
2: Perform the online phase of hyper-reduction. Details are discussed below in section 4.
3: Compute $u^n(\cdot, t_k, \mu)$ for all $k \in \{1, \dots, K\}$ using residual-minimisation and hyper-reduction. Details are discussed in subsection 3.2 and in section 4.

REMARK 4. *One can treat the time variable the same as a parameter, and compute the spatial transform for a few time instances while performing a polynomial approximation (same as (3.6)) for others. This reduces the offline computation cost at the expense of some accuracy. For the simplicity of exposition, with an additional offline cost, we compute the spatial transform for all time instances.*

4. Hyper-reduction

Let $\mathbb{P}_{\mu,t_{k+1}} : \mathbb{R}^N \to \mathbb{R}^N$ represent a (μ, t_{k+1})-dependent operator which is such that solving the following minimisation problem is (much) cheaper than solving the one given in (3.9)

$$u^{n,hyp}(\cdot, t_{k+1}, \mu) := \underset{w \in \mathcal{X}^n_{\mu,t_{k+1}}}{\arg\min} \|\mathbb{P}_{\mu,t_{k+1}} \operatorname{Res}(\Pi_{\mu,t_{k+1}} w, \Pi_{\mu,t_k} u^{n,hyp}(\cdot, t_k, \mu))\|_{\mathbb{R}^N}. \tag{4.1}$$

Above, Res is given in (3.8), and $u^{n,hyp}(\cdot, t_{k+1}, \mu)$ is an approximation to $u^n(\cdot, t_{k+1}, \mu)$. Following a collocation based approach, we consider

$$\mathbb{P}_{\mu,t_{k+1}} = P_{\mu,t_{k+1}}, \tag{4.2}$$

where

$$P_{\mu,t_{k+1}} \in \mathbb{R}^{N \times N}, \quad \left(P_{\mu,t_{k+1}}\right)_{ij} := \begin{cases} 1, & i \in \mathcal{E}_{\mu,t_{k+1}}, j = i, \\ 0, & \text{else} \end{cases}. \tag{4.3}$$

Thus, $P_{\mu,t_{k+1}}$ is a collocation matrix for the set $\mathcal{E}_{\mu,t_{k+1}} \subseteq \{1, \dots, N\}$ and has zero columns for the indices in $\left(\mathcal{E}_{\mu,t_{k+1}}\right)^c$, where $(\cdot)^c$ represents the complement of a set. For simplicity, whenever it is clear from the context, we will remove the zero columns from $P_{\mu,t_{k+1}}$. Let

$$N^{hyp}_{\mu,t_{k+1}} := \#\mathcal{E}_{\mu,t_{k+1}}, \tag{4.4}$$

where $\#(\cdot)$ represents the number of elements in a set. For $N^{hyp}_{\mu,t_{k+1}} < N$, with the above choice of $\mathbb{P}_{\mu,t_{k+1}}$, we compute only the $N^{hyp}_{\mu,t_{k+1}}$ entries in Res. This reduces both, the cost of evaluating the residual and the cost of solving the least-squares problem (3.10). We refer to $\mathcal{E}_{\mu,t_{k+1}}$ as a set of collocation points and compute it as follows. We emphasis that our computation of $\mathcal{E}_{\mu,t_{k+1}}$ does not assume the specific form of the spatial transform considered later in (5.1).

4.1 Computation of the collocation points: We divide the computation of $\mathcal{E}_{\mu,t_{k+1}}$ into an offline and an online stage. The offline and the online stage corresponds to `line-3` and `line-2` of Algorithm 3.1 and Algorithm 3.2, respectively, and are outlined as follows.

(i) **Offline stage:** Let $\{\tilde{\mu}_i\}_{i=1,\dots,M^{hyp}} \subset \mathcal{I}_{\gamma(\mu)}$ (see (3.1) for a definition of $\mathcal{I}_{\gamma(\mu)}$) be a set of parameters that does not overlap with the set $\{\hat{\mu}_i\}_{i=1,\dots,M}$ given in (3.5). The reason for considering non-overlapping sets is made clear below in remark 5. The value of $M^{hyp} \leq M$ is user-defined. Choosing $M^{hyp} > M$ did not provide any additional benefit in our numerical experiments. We compute $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ by minimising a bound on the error

$$E_{\tilde{\mu}_i,t_{k+1}} := \|u^N(\cdot,t_{k+1},\tilde{\mu}_i) - u^{n,hyp}(\cdot,t_{k+1},\tilde{\mu}_i)\|_{L^2(\Omega)}. \tag{4.5}$$

(ii) **Online stage:** Let $\mu \notin \{\tilde{\mu}_i\}_{i=1,\dots,M^{hyp}} \cup \{\hat{\mu}_i\}_{i=1,\dots,M}$ be the parameter of interest. Then, with the spatial transforms $\varphi(\cdot,\mu,\tilde{\mu}_i,t_{k+1})$, we account for transport in $\{\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}\}_{i=1,\dots,M^{hyp}}$. This finally provides us with $\mathcal{E}_{\mu,t_{k+1}}$.

Offline stage: A bound for $E_{\tilde{\mu}_i,t_{k+1}}$ follows from the result given below; similar result can be found in [9]. The result uses recursion and assumes the Lipschitz continuity of the operator $\mathrm{Id} + \Delta t \times L^N(\cdot,\mu)$, where $L^N(\cdot,\mu)$ is as given in (2.3). For the time-step restriction given in (2.4), the assumption of Lipschitz continuity is satisfied; see [10, 13] for further details. Note that the result holds independent of the choice of $\mathbb{P}_{\mu,t_{k+1}}$.

LEMMA 4.1. *Let* $\mathrm{Id} : \mathcal{X}^N \to \mathcal{X}^N$ *be the identity operator, and let* $L^N(\cdot,\mu)$ *be as given in* (2.3). *Assume that the operator* $\mathrm{Id} + \Delta t \times L^N(\cdot,\mu)$ *is Lipschitz continuous on* $\mathcal{X}^N$ *with a Lipschitz constant* $C > 0$ *i.e, for all* $u, v \in \mathcal{X}^N$ *it holds*

$$\|\big(u + \Delta t \times L^N(u,\mu)\big) - \big(v + \Delta t \times L^N(v,\mu)\big)\|_{L^2(\Omega)} \leq C\|u - v\|_{L^2(\Omega)}.$$

Let $\{v(t_j)\}_{j=1,\dots,K}$ *be a sequence in* $\mathcal{X}^n$, *then it holds*

$$\|u^N(\cdot,t_{k+1},\mu) - v(t_{k+1})\|_{L^2(\Omega)} \leq \sum_{j=0}^{k} C^{k-j}\,(c_j + d_j)\,, \tag{4.6}$$

where $k \in \{0,\dots,K-1\}$, *and*

$$\begin{aligned} c_j &:= \|\mathbb{P}_{\mu,t_{j+1}} \operatorname{Res}\big(\Pi_{\mu,t_{j+1}} v(t_{j+1}), \Pi_{\mu,t_j} v(t_j)\big)\|_{\mathbb{R}^N}, \\ d_j &:= \|\big(\mathrm{Id} - \mathbb{P}_{\mu,t_{j+1}}\big) \operatorname{Res}\big(\Pi_{\mu,t_{j+1}} v(t_{j+1}), \Pi_{\mu,t_j} v(t_j)\big)\|_{\mathbb{R}^N}. \end{aligned} \tag{4.7}$$

Proof. See Appendix A. ∎

Choosing $v(t_{k+1}) = u^{n,hyp}(\cdot,t_{k+1},\tilde{\mu}_i)$ in (4.6), provides a bound for the error $E_{\tilde{\mu}_i,t_{k+1}}$ given in (4.5). It is preferable to make this bound as small as possible. The definition of $u^{n,hyp}(\cdot,t_{k+1},\mu)$, for a given $\mathbb{P}_{\mu,t_{k+1}}$, minimises the c_k's appearing in (4.6). We choose the set $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ such that we minimise an upper bound on d_k.

To have an upper bound on d_k, we make the following assumption. We assume that if the total number of collocation points is larger than M then irrespective of the choice of the collocation points, there exists a time-step size smaller than or equal to the bound given in (2.4) such that the solution $u^{n,hyp}(\cdot,t_k,\mu)$ is L^2-stable. Equivalently, for a given grid size $\Delta x > 0$, we assume that

$$\exists\ 0 < \Delta t \leq \Delta t^{opt}\ :\ \|u^{n,hyp}(\cdot,t,\mu)\|_{L^2(\Omega)} \leq C, \quad \forall (t,\mu) \in [0,T] \times \mathcal{P}, \mathcal{E}_{\mu,t} \in \mathcal{E}, \tag{4.8}$$

where $\mathcal{E}$ is a collection of all possible collocation points with size larger than M, and Δt^{opt} is the upper-bound given in (2.4). remark 6 below further elaborates on the above assumption. Note that under the time-step restriction given in (2.4), the FOM also satisfies an estimate similar to the above [27].

Using the above assumption, the upper bound on d_k follows from the definition of Res given in (3.8), our choice of $v(t_k)$ and triangle's inequality

$$d_k^2 \leq C \left(\underbrace{\|(\mathrm{Id} - \mathbb{P}_{\tilde{\mu}_i,t_{k+1}}) A_{\tilde{\mu}_i,t_{k+1}}\|_F^2 + \|(\mathrm{Id} - \mathbb{P}_{\tilde{\mu}_i,t_{k+1}}) b_{\tilde{\mu}_i,t_k}\|_{\mathbb{R}^N}^2}_{=:E_1} \right). \tag{4.9}$$

Above, $A_{\tilde{\mu}_i,t_{k+1}}$ and $b_{\tilde{\mu}_i,t_k}$ are as defined in (3.11), and $\|\cdot\|_F$ represents the Frobenius norm of a matrix. Let $(a_1,\dots,a_N) = A^T_{\tilde{\mu}_i,t_{k+1}}$. We conclude that for the choice of $\mathbb{P}_{\tilde{\mu}_i,t_{k+1}}$ given in (4.2), E_1 is minimum when we define $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ as

$$\mathcal{E}_{\tilde{\mu}_i,t_{k+1}} := \underset{\omega\subseteq\{1,\dots,N\},\#\omega=N^{hyp}_{\tilde{\mu}_i,t_{k+1}}}{\arg\max} \sum_{p\in\omega} \left(\|a_p\|^2_{\mathbb{R}^M} + (b_{\tilde{\mu}_i,t_k})^2_p \right). \tag{4.10}$$

REMARK 5. *One can check that for $\mu \in \{\tilde{\mu}_i\}_{i=1,\dots,M^{hyp}} \cap \{\hat{\mu}_i\}_{i=1,\dots,M}$, we have*

$$\text{Res}\left(\Pi_{\mu,t_{k+1}} u^{n,hyp}(\cdot,t_{k+1},\mu), \Pi_{\mu,t_k} u^{n,hyp}(\cdot,t_k,\mu)\right) = 0,$$

where $k \in \{0,\dots,K-1\}$. The above relation makes the bound on $E_{\mu,t_{k+1}}$ trivial therefore, we choose $\{\tilde{\mu}_i\}_{i=1,\dots,M^{hyp}}$ such that it does not overlap with $\{\hat{\mu}_i\}_{i=1,\dots,M}$.

REMARK 6. *Presently, there lacks a theoretical justification for the assumption in (4.8). However, choosing $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ randomly, ensuring $\#\mathcal{E}_{\tilde{\mu}_i,t_{k+1}} \geq M$ and performing multiple runs of the ROM results in a solution that is bounded in $L^2(\Omega)$. This indicates that a proof of (4.8) could be possible. For $\#\mathcal{E}_{\tilde{\mu}_i,t_{k+1}} < M$ we have an under-determined least-squares problem (4.1), which, at least for our test cases, results in instability.*

Online stage: We compute $\mathcal{E}_{\mu,t_{k+1}}$ in the online phase as follows. As μ deviates from $\tilde{\mu}_i$, the snapshot $u^N(\cdot,t_{k+1},\tilde{\mu}_i)$ is transported along the spatial domain and we transport the entries in $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ along with it. Since the spatial transform $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$ captures the transport of $u^N(\cdot,t_{k+1},\mu)$ to $u^N(\cdot,t_{k+1},\tilde{\mu}_i)$, we approximate $\mathcal{E}_{\mu,t_{k+1}}$ by transforming every entry in $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ with $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$. We do so as follows. Let y_p denote the centroid of the mesh element $\mathcal{I}^x_p$. Every $p \in \mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ corresponds to a unique y_p. Then, $\varphi(y_p,\tilde{\mu}_i,\mu,t_{k+1})$ denotes the spatial location of a collocation point in $\mathcal{E}_{\mu,t_{k+1}}$. To get the collocation point corresponding to $\varphi(y_p,\tilde{\mu}_i,\mu,t_{k+1})$, we define $\Upsilon:\Omega\to\mathbb{N}$ such that $x \in \mathcal{I}^x_{\Upsilon(x)}$, where $\mathcal{I}^x_{\Upsilon(x)}$ is the $\Upsilon(x)$-th spatial element. Then, $\mathcal{E}_{\mu,t_{k+1}}$ is given as

$$\mathcal{E}_{\mu,t_{k+1}} \approx \bigcup_{i=1}^{M^{hyp}} \underline{\left\{\Upsilon(\varphi(y_p,\tilde{\mu}_i,\mu,t_{k+1})) \;:\; p \in \mathcal{E}_{\tilde{\mu}_i,t_{k+1}}\right\}}. \tag{4.11}$$

There are two ways to compute $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$: (i) compute the snapshots $\{\varphi(\cdot,\tilde{\mu}_j,\tilde{\mu}_i,t_{k+1})\}_{j=1,\dots,M^{hyp}}$ offline and approximate $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$ using the Lagrange interpolation (3.6); and (ii) use $\varphi(\cdot,\hat{\mu}_i,\mu,t_{k+1})$ (which we anyhow compute) to approximate $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$. We use the second option because it is cheaper. To approximate $\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1})$, we assume that a spatial transform follows the following chain relation [30]

$$\varphi(\cdot,\tilde{\mu}_i,\mu,t_{k+1}) \approx \varphi(\cdot,\tilde{\mu}_i,\hat{\mu}_i,t_{k+1}) \circ \varphi(\cdot,\hat{\mu}_i,\mu,t_{k+1}). \tag{4.12}$$

Both $\varphi(\cdot,\tilde{\mu}_i,\hat{\mu}_i,t_{k+1})$ and $\varphi(\cdot,\hat{\mu}_i,\mu,t_{k+1})$ then follow from the Lagrange interpolation in (3.6).

REMARK 7. *From (4.11) we find that the size of $\mathcal{E}_{\mu,t_{k+1}}$ is $M^{hyp} \times N^{hyp}_{\mu,t_{k+1}}$. However, in all our numerical experiments $\#\mathcal{E}_{\mu,t_{k+1}} \approx N^{hyp}_{\mu,t_{k+1}}$, which implies a large overlap in the underlined sets shown in (4.11). This is expected when the spatial transform accurately approximates the transport in the snapshots.*

REMARK 8. *The above definition of $\mathcal{E}_{\tilde{\mu}_i,t_{k+1}}$ given in (4.10) requires both $A_{\tilde{\mu}_i,t_{k+1}}$ and $b_{\tilde{\mu}_i,t_k}$. To compute $b_{\tilde{\mu}_i,t_k}$, one needs to solve the expensive least-square problem (3.10) for all $\mu \in \{\tilde{\mu}_i\}_{i=1,\dots,M^{hyp}}$. However, this computation is offline and is done only for a finite number of μ-values. This is (much) cheaper than solving the expensive least-squares problem (3.10) for every query parameter.*

4.2 Discussion:

In [18] authors choose $\mathcal{E}_{\mu,t_{k+1}}$ as the discrete empirical interpolation collocation points of the residual. This approach has a few shortcomings. Firstly, it does not need to minimise the error bound in (4.9). Secondly, to choose $N^{hyp}_{\mu,t_{k+1}}$ number of collocation points, one requires at least $M^{hyp} = N^{hyp}_{\mu,t_{k+1}}$ number of snapshots of the residual and a singular-value-decomposition of the resulting snapshot matrix. This results in an expensive offline computation if the singular values decay slowly, which results in $N^{hyp}_{\mu,t_{k+1}}$ being large. In our method, the error bound in (4.9) is minimum by construction. Moreover, the value of M^{hyp} and $N^{hyp}_{\mu,t_{k+1}}$ can be chosen independently. Even with $M^{hyp} = 1$, one can have a $N^{hyp}_{\mu,t_{k+1}}$ as large as possible.

Apart from the error $E_{\mu,t_{k+1}}$ defined in (4.5), we can define the following error that quantifies the accuracy lost by solving the hyper-reduced minimisation problem in (4.1) over the original one in (3.9)

$$\hat{E}_{\mu,t_{k+1}} := \|u^n(\cdot,t_{k+1},\mu) - u^{n,hyp}(\cdot,t_{k+1},\mu)\|_{L^2(\Omega)}. \tag{4.13}$$

Let $\tilde{N}_{\mu,t_{k+1}} = N - N^{hyp}_{\mu,t_{k+1}}$. As $\tilde{N}_{\mu,t_{k+1}} \to 0$, we expect $\hat{E}_{\mu,t_{k+1}} \to 0$. Unfortunately, there is an unavailability of a bound on $\hat{E}_{\mu,t_{k+1}}$ that could prove its convergence with $\tilde{N}_{\mu,t_{k+1}}$. We leave the development of such a bound for future works, and later study the convergence of $\hat{E}_{\mu,t_{k+1}}$ using numerical experiments. We speculate that it should be possible to bound $\hat{E}_{\mu,t_{k+1}}$ in terms of E_1 given in (4.9). In that case if E_1 converges slowly then to achieve an acceptable accuracy, one would require a large value of $N^{hyp}_{\mu,t_{k+1}}$. This would reduce the efficiency of our hyper-reduction. However, at least for linear problems, ensuring a decay in E_1 is simple. Consider, for example, the linear advection equation $\partial_t u + \partial_x u = 0$. Since $u(x,t) = u_0(x-t)$, an appropriate choice of initial conditions ensures a decay in E_1.

One can choose $\mathbb{P}_{t_{k+1}}$ with the gappy-POD (or the DEIM [6]) approach by projecting the residual onto its POD-basis [2, 5]. This provides

$$\mathbb{P}_{t_{k+1}} = U_{t_{k+1}}(P^T_{t_{k+1}}U_{t_{k+1}})^\dagger P^T_{t_{k+1}}, \tag{4.14}$$

where $P_{t_{k+1}}$ is the collocation matrix, $(\cdot)^\dagger$ denotes the Moore-Penrose inverse of a matrix, and $U_{t_{k+1}} \in \mathbb{R}^{N\times N^{hyp}_{t_{k+1}}}$ is a set of POD-modes for the snapshot matrix of residuals. Note that the collocation matrix $P_{t_{k+1}}$ is not necessarily the same as that resulting from (4.11). The DEIM approach differs from ours in the following sense. Firstly, it assumes that the residual is well-approximated in a linear finite-dimensional space, which is the span of the POD basis. Secondly, the collocation matrix is computed offline with a greedy-iteration and is μ-independent. Note that for a non-linear approximation of the form (3.5), similar to the solution, the residual can also show a moving wave-type behaviour along the μ-space. This can result in (i) poor approximability of the residual in a linear space, and (ii) an ill-suited μ-independent collocation matrix $P_{t_{k+1}}$. Later, through numerical examples we demonstrate the moving wave-type behaviour of the residual and the problems that arise from it. Note that also the DEIM approach does not provide a bound for the error $\hat{E}_{\mu,t_{k+1}}$ given above in (4.13).

5. Computation of the spatial transforms

We discuss the computation of $\varphi(x,\hat{\mu}_i,\hat{\mu}_j,t_k)$ appearing in (3.6). An algorithm to compute the spatial transform should provide

(i) a $\varphi(\cdot,\hat{\mu}_i,\hat{\mu}_j,t_k)$ for an arbitrarily large $|\hat{\mu}_i - \hat{\mu}_j|$;

(ii) an invertible $\varphi(\cdot,\hat{\mu}_i,\hat{\mu}_j,t_k)$.

Choosing the parameter samples $\{\hat{\mu}_j\}_{j=1,\dots,M}$ with a greedy-algorithm that has an arbitrary error tolerance could result in an arbitrarily large $|\hat{\mu}_i - \hat{\mu}_j|$. Similar observation holds for other methods used to sample $\{\hat{\mu}_j\}_{j=1,\dots,M}$. That is why we need the first requirement. The second requirement is motivated by intuition. To elaborate, let $u^N(\cdot,t_k,\hat{\mu}_j)$ represent the density of some fluid. Let $\varphi(x,\hat{\mu}_i,\hat{\mu}_j,t_k)$ be non-invertible, which implies the existence of a x_0 and a x_1 such that $y = \varphi(x_0,\hat{\mu}_i,\hat{\mu}_j,t_k) = \varphi(x_1,\hat{\mu}_i,\hat{\mu}_j,t_k)$. Then, composing $u^N(\cdot,t_k,\hat{\mu}_j)$ with $\varphi(\cdot,\hat{\mu}_i,\hat{\mu}_j,t_k)$ results in the density at y being transported to two different locations x_0 and x_1, which is physically unacceptable.

We satisfy the second requirement of the above two by choosing $\varphi(x,\hat{\mu}_i,\hat{\mu}_j,t_k)$ to be a shift in space i.e.,

$$\varphi(x,\hat{\mu}_i,\hat{\mu}_j,t_k) = \Theta(x,c(\hat{\mu}_i,\hat{\mu}_j,t_k)) \quad \text{where} \quad \Theta(x,c) := x - c. \tag{5.1}$$

Above, we still need to compute the shifts $c(\hat{\mu}_i,\hat{\mu}_j,t_k) \in \mathbb{R}^d$. Note that $u^N(\Theta(x,c),t,\mu)$ requires values from outside of Ω, which we prescribe as follows. We assume that there exists an $\epsilon \geq \Delta x$ such that the solution, for all time and parameter instances, stays constant inside $[x_{\min}, x_{\min}+\epsilon]$ and $[x_{\max}-\epsilon, x_{\max}]$. Equivalently,

$$u(x,t,\mu) = \begin{cases} U_0(\mu), & \forall (x,t,\mu) \in [x_{\min}, x_{\min}+\epsilon]\times[0,T]\times\mathcal{P}, \\ U_1(\mu), & \forall (x,t,\mu) \in [x_{\max}-\epsilon, x_{\max}]\times[0,T]\times\mathcal{P}, \end{cases} \tag{5.2}$$

where $U_0(\mu), U_1(\mu) \in \mathbb{R}$. With the above assumption, if $x - c < x_{\min}$, we set $u^N(x-c,t,\mu) = U_0(\mu)$, and if $x - c > x_{\max}$, we set $u^N(x-c,t,\mu) = U_1(\mu)$. Note that the assumption is true for Riemann problems, and Cauchy problems with compactly supported initial data.

The above mentioned first requirement is ensured by computing a shift $c(\hat{\mu}_i, \hat{\mu}_j, t_k)$ for any given parameter difference $|\hat{\mu}_i - \hat{\mu}_j|$. When $i = j$, we choose $c(\hat{\mu}_i, \hat{\mu}_j, t_k) = 0$. This ensures (3.4). For $i \neq j$, we compute $c(\hat{\mu}_i, \hat{\mu}_j, t_k)$ as follows. One way is to use the following minimisation problem that minimises the $L^2(\Omega)$ error

$$c(\hat{\mu}_i, \hat{\mu}_j, t_k) = \underset{c \in [-n_x \Delta x, n_x \Delta x]}{\arg\min} \underbrace{\| u^N(\Theta(\cdot, c), t_k, \hat{\mu}_j) - u^N(\cdot, t_k, \hat{\mu}_i) \|_{L^2(\Omega)}}_{=: \mathcal{R}(c)}. \tag{5.3}$$

One can solve the above problem using a fixed-point iteration [18, 30]. However, the following problems arise. Firstly, the function $\mathcal{R}$ does not need to be convex, which could result in a fixed-point iteration providing a sub-optimal local minima. Secondly, because $\mathcal{R}$ can have flat regions, a fixed-point iteration is (very) sensitive to its initial guess and the step-size. For example, consider the following two characteristic functions

$$u^N(\cdot, t_k, \hat{\mu}_j) = \mathbb{1}_{[0,1]}, \quad u^N(\cdot, t_k, \hat{\mu}_i) = \mathbb{1}_{[2,3]}. \tag{5.4}$$

The two functions are L^2-orthogonal for $c \in [0, 1]$, which results in $\mathcal{R}([0, 1]) = \sqrt{2}$. Therefore, with an initial guess of $c = 0$ (without any additional regularisation) one will never move past the initial guess. Choosing $|\hat{\mu}_j - \hat{\mu}_i|$ small enough ensures the strict convexity of the minimisation problem (5.3), but it is unclear how small should $|\hat{\mu}_j - \hat{\mu}_i|$ be [30]. Moreover, at least for the above example, a shifted $u^N(\cdot, t_k, \hat{\mu}_j)$ accurately approximates $u^N(\cdot, t_k, \hat{\mu}_i)$ and therefore, decreasing $|\hat{\mu}_j - \hat{\mu}_i|$ further is unnecessary. Indeed, $u^N(x - 2, t_k, \hat{\mu}_j) = u^N(x, t_k, \hat{\mu}_i)$. Note that even if we can compute a unique shift using (5.3), it does not ensure that the discontinuities in $u^N(\cdot, t_k, \hat{\mu}_i)$ and $u^N(\cdot, t_k, \hat{\mu}_j)$ are aligned. Such an alignment is needed to accurately capture the shock speeds and locations.

REMARK 9. *With our choice of the spatial transform (3.6), one can check that the chain relation in (4.12) simplifies to* $c(\tilde{\mu}_i, \mu, t_{k+1}) \approx c(\hat{\mu}_i, \tilde{\mu}_i, t_{k+1}) - c(\hat{\mu}_i, \mu, t_{k+1})$.

5.1 Computing spatial transforms using feature matching:

For the above reasons, we do not use a fixed-point iteration to find the shift values. Rather, we find a shift such that it aligns a dominant feature between the two snapshots. We elaborate on what we mean by a dominant feature. A feature is user-defined and refers to a local structure in the solution that one wishes to capture. For example, in fluid flow applications, a feature could mean a shock, a rarefaction fan, a vortex etc. Out of all the features, the dominant feature is the one, aligning which, results in the minimum L^2-distance between the snapshots. Note that although the definition of feature(s) is flexible, it should be such that for all $(\mu, t) \in \mathcal{P} \times [0, T]$ the FOM contains at least one feature. Else, one ends up with no features to align, which results in no shift values.

We consider a shock and a (strict) local minima/maxima in the solution as a feature. After some finite time (usually) solutions to non-linear hyperbolic problems develop shocks and it is desirable to capture these shocks accurately. For that reason we consider them as a feature. For a continuous initial data, it is possible that the solution does not contain a shock. This motivates us to consider a local minima/maxima in the solution as a feature. Moreover, since the absolute value of a solution is locally maximum near a local minima/maxima, we expect that aligning these local minima/maxima will significantly reduce the L^2-distance between the two snapshots. With an appropriate choice of the initial data (or the space-time domain $\Omega \times [0, T]$), we ensure that every snapshot has at least one shock or a local minima/maxima. Later, we present examples of such a situation.

We cast the computation of a shift using feature matching as a minimisation problem. Let $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$, which is a subset of $[-n_x \Delta x, n_x \Delta x]$, represent a finite $(\hat{\mu}_i, \hat{\mu}_j, t_k)$-dependent set that contains shifts that align all possible features between the snapshots $u^N(\cdot, t_k, \hat{\mu}_j)$ and $u^N(\cdot, t_k, \hat{\mu}_i)$. Then, following the above discussion, finding a shift through feature matching is equivalent to solving the following problem

$$c(\hat{\mu}_i, \hat{\mu}_j, t_k) = \underset{c \in \mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)}{\arg\min} \mathcal{R}(c), \tag{5.5}$$

where $\mathcal{R}(c)$ is as defined above in (5.3). We assume that the size of $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$, which we denote by $\#\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$, is small enough. Then, we can cheaply solve the above problem using enumeration i.e., we compute $\mathcal{R}(c)$ for all $c \in \mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ and pick the shift corresponding to the minimum value of $\mathcal{R}(c)$. We later elaborate on our assumption of $\#\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ being small. Below, we discuss how to compute $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$.

Identification of features: Let y_j denote the centroid of the mesh element $\mathcal{I}_j^x$. Let $du^j(t_k, \hat{\mu}_i)$ represent an approximation to the first-order space derivative in the j-th mesh element. We can compute such a derivative with (for example) central differences applied to cell-averages $u^N(y_j, t_k, \mu)$. Define the ratio $r^j(t_k, \hat{\mu}_i)$ as

$$r^j(t_k, \hat{\mu}_i) := \frac{\operatorname{sgn}(du^j(t_k, \hat{\mu}_i) - du^{j-1}(t_k, \hat{\mu}_i))}{\operatorname{sgn}(du^{j+1}(t_k, \hat{\mu}_i) - du^j(t_k, \hat{\mu}_i))},$$

where $\operatorname{sgn}(\cdot)$ represents a sign function. In $\mathcal{B}(\hat{\mu}_i, t_k)$ we collect the locations of the centroids of all those mesh-elements (i.e. $\mathcal{I}_j^x$) for which $r^j(\hat{\mu}_i, t_k) < 0$. Doing the same for $u^N(\cdot, t_k, \hat{\mu}_j)$ provides us with the set $\mathcal{B}(\hat{\mu}_j, t_k)$. Thus, the sets $\mathcal{B}(\hat{\mu}_i, t_k)$ and $\mathcal{B}(\hat{\mu}_j, t_k)$ contains shock locations and the location of the local minima/maxima occurring in $u^N(\cdot, t_k, \hat{\mu}_i)$ and $u^N(\cdot, t_k, \hat{\mu}_j)$, respectively. Aligning all possible locations in these two sets provides us with $(\#\mathcal{B}(\hat{\mu}_i, t_k) \times (\#\mathcal{B}(\hat{\mu}_j, t_k))$ number of shift values that we collect in the set $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$. We then remove the repeated shifts occurring in $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$, which provides us with the desired set.

5.2 Discussion:

In [23], $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ is a set of shifts that are integer multiples of Δx and lie inside $[-n_x\Delta x, n_x\Delta x]$. Such a choice results in $\#\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ scaling with n_x (or with n_x^d for d-dimensions), which leads to an expensive solution to (5.5). In contrast, for our choice of $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$, if the FOM does not oscillate along the entire spatial domain or does not have a large number of shocks then we expect $\#\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k) \ll n_x$. This is the case for all our numerical experiments. Moreover, for convergent FOMs, we expect that no new points are added to $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ beyond a certain n_x.

The minimisation problem (5.5) does not need to have a unique solution. Consider the two characteristic functions $u^N(\cdot, t_k, \hat{\mu}_i) = \mathbb{1}_{[0,1]}$ and $u^N(\cdot, t_k, \hat{\mu}_j) = \mathbb{1}_{[2,4]}$. One can check that $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k) = \{2, 3\}$, and that $\mathcal{R}\left(\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)\right) = \{1, 1\}$. Clearly, $\mathcal{R}$ is a constant on $B(\hat{\mu}_i, \hat{\mu}_j, t_k)$. If minimising $\mathcal{R}$ (i.e. the L^2-error) is the sole interest then both the shifts in $B(\hat{\mu}_i, \hat{\mu}_j, t_k)$ are equally acceptable. One can make a distinction between the two shifts by specifying additional quantities of interest. For example, the shifts $c = 2$ and $c = 3$ accurately capture the shocks at $x = 2$ and $x = 4$, respectively. Therefore, if one is interested in the shock at $x = 2$ then one must choose $c = 2$. In all our numerical experiments the solution is such that the minimisation problem (5.5) resulted in a unique solution, and we did not specify any additional quantity of interest.

The spatial transform (5.1) has d-degrees of freedom, which are the d-components of the shift $c(\hat{\mu}_i, \hat{\mu}_j, t_k)$. With d-degrees of freedom, in a d-dimensional spatial domain, we accurately capture one dominant feature of the solution. To capture more than one feature, one requires additional degrees of freedom in $\varphi(\cdot, \hat{\mu}_i, \hat{\mu}_j, t_k)$, which one can introduce with a higher-order polynomial (or Fourier-series) expansion for $\varphi(\cdot, \hat{\mu}_i, \hat{\mu}_j, t_k)$; see [18, 30]. One computes such a spatial transform by minimising a residual (see (5.3) above) with a fixed-point iteration, problems related to which are already discussed above. Moreover, it is unclear how to ensure the invertibility of such a spatial transform, which is undesirable. Another possibility to capture additional features could be to introduce spatial dependence in the shift. Such a shift would move different (spatial) "parts" of the solution differently, allowing one to capture more than one feature. This would be similar to the monotonic rearrangement based interpolation considered in [22]. We hope to consider a spatially dependent shift in our future work.

We present examples where the set $\mathcal{B}(\hat{\mu}_i, \hat{\mu}_j, t_k)$ given in (5.5) is non-empty, or equivalently, scenarios where every snapshot has at least one shock or a local minima/maxima.

EXAMPLE 5.1. *Consider linear advection $\partial_t u(\cdot, \cdot, \mu) + \beta(\mu)\partial_x u(\cdot, \cdot, \mu) = 0$. One can show that $u(x, t, \mu) = u_0(x - \beta(\mu)t)$. As a result, if $u_0(\cdot, \mu)$ has a discontinuity then so does $u(\cdot, \cdot, \mu)$. Moreover, if $u_0(\cdot, \mu)$ has strict local minima/maxima at the points $\mathcal{B} \subset \mathbb{R}$ then so does $u(\cdot, \cdot, \mu)$ at the points $\{x + \beta(\mu)t \ : \ x \in \mathcal{B}\}$.*

EXAMPLE 5.2. *Consider the Burger's equation $\partial_t u(\cdot, \cdot, \mu) + \frac{1}{2}\partial_x u(\cdot, \cdot, \mu)^2 = 0$. If $u_0(\cdot, \mu)$ is discontinuous and non-increasing then there exists a finite time before which the solution has a shock. If $u_0(\cdot, \mu)$ is smooth then either the solution develops a shock after some time or it remains smooth. Following the reasoning of the previous example, a smooth solution will preserve the strict local minima/maxima in the initial data.*

5.3 Projection Operator:

We define the projection operator $\Pi_{\mu,t_k} : \mathcal{X}^n_{\mu,t_k} \to \mathcal{X}^N$ appearing in (3.9). Substituting the expression for the snapshots of spatial transform from (5.1) into the Lagrange

interpolation in (3.6), we find a Lagrange interpolation for the shift values

$$c(\mu, \hat{\mu}_j, t_k) \approx c^M(\mu, \hat{\mu}_j, t_k) := \sum_{i=1}^{M} l_{i-1}(\mu) c^M(\hat{\mu}_i, \hat{\mu}_j, t_k). \tag{5.6}$$

Let $u^N(\cdot, t_k, \hat{\mu}_j) = \sum_{i=1}^{N} \beta_i(t_k, \hat{\mu}_j)\phi_i$ where $\beta_i(t_k, \hat{\mu}_j)$ are some expansion coefficients, and ϕ_i is given in (2.1). Then, from the definition of ψ^j_{μ,t_k} (given in (3.5), we find

$$\psi^j_{\mu,t_k} = \sum_{i=1}^{N} \beta_i(t_k, \hat{\mu}_j)\phi_i(\Theta(\cdot, c^M(\mu, \hat{\mu}_j, t_k))),$$

where Θ is as defined in (5.1). Let $\omega = c^M(\mu, \hat{\mu}_j, t_k)/\Delta x$. Assume that $\omega \in \mathbb{Z}$, and that $\omega > 0$. The construction for $\omega \leq 0$ is similar and is not discussed for brevity. We later discuss the case where $\omega \notin \mathbb{Z}$. One can show that for $i \in \{1, \ldots, N - \omega\}$, we have $\phi_i(\Theta(\cdot, \omega\Delta x)) = \phi_{i+\omega}$. This implies that when $\omega \in \mathbb{Z}$, we have

$$\psi^j_{\mu,t_k} = \sum_{i=1}^{N-\omega} \beta_i(t_k, \hat{\mu}_j)\phi_{i+\omega} + \underline{\sum_{i=1}^{\omega} U_0(\hat{\mu}_j)\phi_i\sqrt{\Delta x}}. \tag{5.7}$$

The underlined term of the above two follows from taking values from outside of Ω using (5.2). When $\omega \notin \mathbb{Z}$, we replace ω by $\lfloor\omega\rfloor$ in the above expression. Thus, for $\omega \geq 0$, we define Π_{μ,t_k} as

$$\Pi_{\mu,t_k}\psi^j_{\mu,t_k} = \sum_{i=1}^{N-\lfloor\omega\rfloor} \beta_i(t_k, \hat{\mu}_j)\phi_{i+\lfloor\omega\rfloor} + \underline{\sum_{i=1}^{\lfloor\omega\rfloor} U_0(\hat{\mu}_j)\phi_i\sqrt{\Delta x}}. \tag{5.8}$$

The above definition relies on shifting the indices of the basis functions ϕ_i, which we expect to be less expensive than computing L^2 inner-products in an orthogonal projection from $\mathcal{X}^n_{t_k,\mu}$ to $\mathcal{X}^N$.

REMARK 10. *Let $a_j^T \in \mathbb{R}^N$ be the j-th column of the matrix $A_{\mu,t_{k+1}}$ given in (3.11). With Π_{μ,t_k} as give above, a_j^T has the following (easy to compute) expression*

$$a_j^T = \left(\underbrace{U_0(\hat{\mu}_j)\sqrt{\Delta x}, \ldots, U_0(\hat{\mu}_j)\sqrt{\Delta x}}_{\lfloor\omega\rfloor - times}, \beta_1(t_k, \hat{\mu}_j), \ldots, \beta_{N-\lfloor\omega\rfloor}(t_k, \hat{\mu}_j) \right)^T. \tag{5.9}$$

6. Relation to the Previous Works

With formal arguments we show the similarities and the differences between the present work and the works related to symmetry reduction [17, 19, 25]. Assume that we can decompose the solution to the evolution equation (1.1) as

$$u(\cdot, t, \mu) = \mathcal{T}(t, \mu) f(t, \mu). \tag{6.1}$$

Above, $\mathcal{T}(t, \mu) : \mathcal{X} \to \mathcal{X}$ and $f(t, \mu) \in \mathcal{X}$, where $\mathcal{X}$ is the solution space to the evolution equation (1.1). We can interpret $f(t, \mu)$ as being representative of the "shape" of $u(\cdot, t, \mu)$, and the action of $\mathcal{T}(t, \mu)$ being representative of the "transport" in $u(\cdot, t, \mu)$. With the above decomposition, approximating the solution is equivalent to approximating $f(t, \mu)$ and $\mathcal{T}(t, \mu)$. One can assume that $f(t, \mu)$ is well-approximated in a linear space and the action of $\mathcal{T}(t, \mu)$ is well-approximated in a non-linear space.

In the present work (and also in [4, 18, 30]), we approximate the action of $\mathcal{T}(t, \mu)$ by shifting (or transforming) the snapshots along the spatial domain and $f(t, \mu)$ in the span of the shifted (or transformed) snapshots. Equivalently, calibrating the manifold $\mathcal{M}_t$ to $\mathcal{M}_{\mu,t}$ with spatial transforms approximates the action of $\mathcal{T}(t, \mu)$, and a linear reduced basis approximation of $\mathcal{M}_{\mu,t}$ approximates the evolution of $f(t, \mu)$. In [17, 19, 25], authors approximate the action of $\mathcal{T}(t, \mu)$ by the action of a Lie-group and $f(t, \mu)$ using a POD/KL-expansion.

Once we have an approximation space for $\mathcal{T}(t, \mu)$ and $f(t, \mu)$, we need to compute the two quantities in their respective approximation spaces. This is where the present work differs from that in [17, 19] where,

before performing any approximation, authors derive a governing equation for $f(t,\mu)$ by substituting the above decomposition (6.1) into the evolution equation (1.1) and multiplying by $\mathcal{T}(t,\mu)^{-1}$. This results in

$$\partial_t f(t,\mu) + \mathcal{T}(t,\mu)^{-1}\partial_t \mathcal{T}(t,\mu) f(t,\mu) + \underline{\mathcal{T}(t,\mu)^{-1} L(\mathcal{T}(t,\mu) f(t,\mu),\mu)} = 0.$$

After substituting the approximation for $\mathcal{T}(t,\mu)$, which results from the so-called reconstruction equation, one can reduce the above equation using any linear model order reduction technique. However, for an efficient ROM, the underlined term needs to be simplified by assuming that $L(\cdot,\mu)$ is invariant under the action of $\mathcal{T}(t,\mu)$. Equivalently,

$$\mathcal{T}(t,\mu)^{-1} L(\mathcal{T}(t,\mu) f(t,\mu),\mu) = L(f(t,\mu),\mu).$$

In the present work (and in [1, 18]), we do not treat the evolution of $\mathcal{T}(t,\mu)$ and $f(t,\mu)$ separately. Rather, we substitute our approximation for $\mathcal{T}(t,\mu)$ and $f(t,\mu)$ into the discretized evolution equation (2.3) that results in a residual. Minimisation of the residual provides us with our ROM. We reduce the computational cost of residual minimisation using hyper-reduction, which does not rely on any invariance property of the evolution operator.

Evolution operators $L(\cdot,\mu)$ of practical relevance are (mostly) only invariant under Galilean transformations i.e., under a rotation, a translation, and a uniform motion of space-time. Approximating the action of $\mathcal{T}(t,\mu)$ through Galilean transforms of $f(t,\mu)$ is accurate for most Cauchy problems but could be ineffective for boundary value problems. For such problems, one requires an approximation (similar to [18, 30]) that is different than a Galilean transform, and which is not necessarily invariant with $L(\cdot,\mu)$. This makes it crucial to develop ROMs that do not rely on the invariance properties of the evolution operator.

In [21, 23], for a given $\mu = \mu_0$, authors consider a snapshot matrix given as

$$\left(\left\langle \Phi, u^N(\cdot,t_1,\mu_0)\right\rangle, \ldots, \left\langle \Phi, u^N(\cdot,t_k,\mu_0)\right\rangle\right),$$

and shift the spatial domain to induce a singular value decay in the snapshot matrix; recall that Φ contains the basis of $\mathcal{X}^N$. Although authors ensure a fast singular value decay, they do not vary the parameter, nor do they propose an algorithm to compute a solution using shifted POD-modes. Both of these problems are considered here.

In [25], the authors consider a shifted KL-expansion. Similar to [17, 19], authors rely on the invariance of the evolution operator (mentioned above) the difference of which to our approach is discussed above. To compute the shifts, authors consider residual minimisation (5.3) and label the snapshot $u^N(\cdot,t_k,\hat{\mu}_i)$ as the template. For all time instances, authors choose the template as the initial data which results in inaccuracy if, with time, the solution changes dramatically in comparison to the initial data. This is true for non-linear problems, and therefore our template (i.e. $u^N(\cdot,t,\hat{\mu}_j)$) is both (t,μ)-dependent. For the same reason, the authors in [23] also consider time-dependent templates.

As to our knowledge, the idea of calibrating the manifold $\mathcal{M}_{t_k}$ first appeared in [4]. For Burger's equation, authors consider a shifted POD-basis as the non-linear approximation space and compute the ROM using residual minimisation. The shifts are computed iteratively and online, the computational cost of which is unclear. To speed-up residual minimisation, authors approximate the L^2 inner-products appearing in the residual, which is different from minimising the residual on a set of collocation points.

7. Numerical Experiments

We consider the following two different test-cases. The details of spatial and temporal discretization are test-case dependent, and are discussed later.

(i) **Test-1 (1D Linear advection):** Linear one-dimensional advection equation with parameterised advection speed

$$\partial_t u(\cdot,\cdot,\mu) + \beta(\mu)\partial_x u(\cdot,\cdot,\mu) = 0, \text{ on } \Omega\times[0,T]. \tag{7.1}$$

We consider $\beta(\mu) = \exp(\mu)/5$, and choose $\mathcal{P} \in [1,3]$. We set $\Omega = [0,4]$ and $T = 0.8$. As the initial data we consider

$$u_0(x) = \begin{cases} \exp\left(-1/\left(1-\left(\frac{x-\delta_1}{\delta_2}\right)^2\right)\right), & \left|\frac{x-\delta_1}{\delta_2}\right| < 1 \\ 0, & \text{else} \end{cases}. \tag{7.2}$$

One can check that for $\delta_1 = 0.5$ and $\delta_2 = 0.2$, u_0 is smooth and compactly supported inside Ω.

(ii) **Test-2 (1D Wave equation):** One dimensional wave-equation with parameterised wave speed

$$\partial_t \begin{pmatrix} u^{(1)}(\cdot,\cdot,\mu) \\ u^{(2)}(\cdot,\cdot,\mu) \end{pmatrix} + A\partial_x \begin{pmatrix} u^{(1)}(\cdot,\cdot,\mu) \\ u^{(2)}(\cdot,\cdot,\mu) \end{pmatrix} = 0, \text{ on } \Omega \times [0,T], \text{ where } A = \begin{pmatrix} 0 & 1 \\ \mu^2 & 0 \end{pmatrix}. \tag{7.3}$$

We choose $\mathcal{P} = [1,3]$, $\Omega = [-2,2]$ and $T = 0.45$. Our initial data is

$$u_0^{(1)}(x,\mu) = \sin(2\pi \times x) \times \mathbb{1}_{[-0.5,0.5]}, \quad u_0^{(2)}(x,\mu) = 0.$$

Although the above PDE is a system of equations, we write it as a system of two independent scalar conservation laws. To each of these conservation laws we independently apply the framework developed in the earlier sections. The details are discussed later.

(iii) **Test-3 (2D Burger's equation):** Two dimensional Burger's equation with parameterised initial data

$$\partial_t u(\cdot,\cdot,\mu) + \frac{1}{2}\partial_x u(\cdot,\cdot,\mu)^2 + \frac{1}{2}\partial_y u(\cdot,\cdot,\mu)^2 = 0, \text{ on } \Omega \times [0,T]. \tag{7.4}$$

We choose $\mathcal{P} = [1,3]$, $\Omega = [0,1]$ and $T = 0.8$. The initial data is given as

$$u_0(x,\mu) = \begin{cases} \mu \times \exp\left(-1/\left(1 - \left(\frac{\|x-\delta_1\|}{\delta_2}\right)^2\right)\right), & \frac{\|x-\delta_1\|}{\delta_2} < 1 \\ 0, & \text{else} \end{cases}. \tag{7.5}$$

We set $\delta_1 = (0.5, 0.5)^T$ and $\delta_2 = 0.2$. Note that the above initial data is the multi-dimensional version of the one considered above in (7.2).

In the following discussion, with `S-ROM` (snapshots based linear ROM) and `SS-ROM` (shifted snapshots based non-linear ROM) we refer to a ROM computed using the approximation space $\mathcal{X}^n_{t_k}$ and $\mathcal{X}^n_{\mu,t_k}$, respectively. Here, $\mathcal{X}^n_{t_k}$ is the approximation space based on dictionaries defined as [1]

$$\mathcal{X}^n_{t_k} := \text{span}\{u^N(\cdot, t_k, \hat{\mu}_j) \ : \ j \in \{1,\dots,M\}\}. \tag{7.6}$$

For a given $\mu \in \mathcal{P}$, we quantify the accuracy of a ROM with the space-time L^2-error

$$E_{ROM}(\mu) := \|u^N(\cdot,\cdot,\mu) - u^n(\cdot,\cdot,\mu)\|_{L^2(\Omega\times[0,T])}, \tag{7.7}$$

where u^n_μ could result either from `S-ROM` or `SS-ROM`. We consider the same parameter samples $\{\hat{\mu}_i\}_{i=1,\dots,M}$ for both the methods.

We implement our method in `matlab2018a`. To solve the least-squares problem (3.9) we use the `matlab` function `lsqminnorm`. None of the online computations use parallelization. All the simulations are run on a computer with two Intel Xeon Silver 4110 processors, 16 cores each and 92GB of RAM.

7.1 Test-1:

We choose a constant time-step size of $\Delta t = 4/(n_x \times \beta(3))$, which satisfies (2.4) and ensures the stability of the FOM. Although not proven, the same time-step was sufficient to ensure the stability of the solution resulting from residual minimisation (3.9). Recall that n_x is the number of spatial elements and its value is given later.

Study of shift computation: Consider two parameter instances $\hat{\mu}_1 = 1$ and $\hat{\mu}_2 = 3$. The exact solution to linear advection (7.1) satisfies $u(x,t,\mu) = u_0(x - \beta(\mu)t)$. Therefore, $u(x,t,\hat{\mu}_2) = u(x - c^{ex}(\hat{\mu}_2,\hat{\mu}_1,t), t, \hat{\mu}_1)$ where $c^{ex}(\hat{\mu}_2,\hat{\mu}_1,t)$ represents the exact shift value given as

$$c^{ex}(\hat{\mu}_2,\hat{\mu}_1,t) := (\beta(\hat{\mu}_2) - \beta(\hat{\mu}_1)) \times t. \tag{7.8}$$

Let $c(\hat{\mu}_2,\hat{\mu}_1,t_k)$ denote the shift resulting from the minimisation problem (5.5) with $\mathcal{B}(\hat{\mu}_2,\hat{\mu}_1,t_k)$ computed using feature matching as described in section 5. We label such a $\mathcal{B}(\hat{\mu}_2,\hat{\mu}_1,t_k)$ as $\mathcal{B}_{fm}$. Fig 2a shows the error $|c^{ex}(\hat{\mu}_2,\hat{\mu}_1,t_k) - c(\hat{\mu}_2,\hat{\mu}_1,t_k)|$ computed with $n_x = 10^3$. The error is either zero or equal to the grid-size $4/n_x$. This is acceptable because all the shift values in $\mathcal{B}_{fm}$ are integer multiples of n_x.

Instead of $\mathcal{B}_{fm}$, we can choose $\mathcal{B}(\hat{\mu}_2,\hat{\mu}_1,t_k)$ to be $\mathcal{B}_{\Delta x}$ that contains shifts that are integer multiples of Δx and lie inside $[-n_x\Delta x, n_x\Delta x]$. Note that the set $\mathcal{B}_{\Delta x}$ is also used in [23] for shift computation. The size of $\mathcal{B}_{\Delta x}$ scales with n_x whereas that of $\mathcal{B}_{fm}$ is independent of n_x, which contains the location of a single local maxima in u_0. Therefore, as n_x increases, using $\mathcal{B}_{fm}$ instead of $\mathcal{B}_{\Delta x}$ results in a significant

speed-up while solving the minimisation problem (5.5). The speed-up is shown in Fig 2b. For smaller values of n_x, using $\mathcal{B}_{fm}$ is more expensive than $\mathcal{B}_{\Delta x}$ because $\mathcal{B}_{fm}$ requires an approximation to the derivative of the solution, which dominates the cost for a small n_x.

There are time instances beyond which $u^N(\cdot, t_k, \hat{\mu}_1)$ and $u^N(\cdot, t_k, \hat{\mu}_2)$ are L^2-orthogonal. As mentioned earlier, L^2-orthogonality is problematic for fixed-points algorithms that solve (5.3) because it results in flat regions in the residual $\mathcal{R}$ given in (5.3). However, the enumeration based approach does not rely on a fixed-point iteration and provides a solution despite of the L^2-orthogonality.

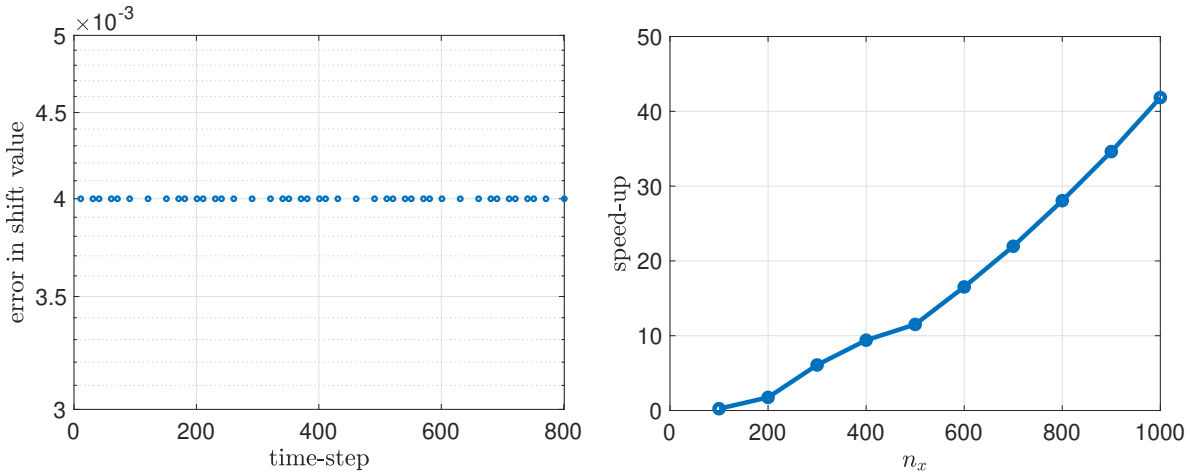


Figure 2: *Results for test-1. (b) the time variation of the error in shift computation; (c) speed-up in shift computation. Fig-(b) has a y-axis on a log-scale.*

Comparison to S-ROM: We study the error E_{ROM}, defined in (7.7), resulting from `SS-ROM` and `S-ROM`. We choose $n_\mu = 4$ in (3.1) and $M = 2$ in (3.6) which results in a piecewise linear approximation for the spatial transform. Moreover, $n_\mu = 4$ results in four parameter elements which we choose uniformly as

$$\mathcal{I}_1 = [1, 1.5],\ \mathcal{I}_2 = [1.5, 2],\ \mathcal{I}_3 = [2, 2.5],\ \mathcal{I}_4 = [2.5, 3]. \tag{7.9}$$

With $M = 2$ we need two parameter samples in each of the parameter elements. We let the endpoints of the parameter elements to be these parameter samples. At 45 uniformly sampled points inside $\mathcal{P}$, Fig 3a compares the error $E_{ROM}(\mu)$ resulting from `SS-ROM` and `S-ROM`.

We first understand the result for `SS-ROM`. Let $\hat{\mu}_1$ and $\hat{\mu}_2$ be the sample parameters that correspond to the endpoints of some $\mathcal{I}_{\gamma(\mu)}$. See (3.2) for a definition of $\gamma(\mu)$. Let $c^{ex}(\mu, \hat{\mu}_j, t_k)$ be as given in (7.8). Let $c^{ex,M}(\mu, \hat{\mu}_j, t_k)$ be the same as the Lagrange interpolation $c^M(\mu, \hat{\mu}_j, t_k)$ given in (5.6) but with $c(\hat{\mu}_i, \hat{\mu}_j, t_k)$ replaced by $c^{ex}(\hat{\mu}_i, \hat{\mu}_j, t_k)$. By triangle's inequality, we can bound the error $|c^M(\mu, \hat{\mu}_j, t_k) - c^{ex}(\mu, \hat{\mu}_j, t_k)|$ as

$$\begin{aligned} |c^M(\mu, \hat{\mu}_j, t_k) - c^{ex}(\mu, \hat{\mu}_j, t_k)| \leq & |c^{ex,M}(\mu, \hat{\mu}_j, t_k) - c^{ex}(\mu, \hat{\mu}_j, t_k)| \\ & + |c^{ex,M}(\mu, \hat{\mu}_j, t_k) - c^M(\mu, \hat{\mu}_j, t_k)|. \end{aligned} \tag{7.10}$$

Our previous study shows that $|c(\hat{\mu}_i, \hat{\mu}_j, t_k) - c^{ex}(\hat{\mu}_i, \hat{\mu}_j, t_k)|$ is $\mathcal{O}(\Delta x)$, which implies the same for $|c^{ex,M}(\mu, \hat{\mu}_j, t_k) - c^M(\mu, \hat{\mu}_j, t_k)|$. Therefore, ignoring errors from spatial discretization and using standard error bounds for Lagrange interpolation, we find

$$\begin{aligned} |c^M(\mu, \hat{\mu}_j, t_k) - c^{ex}(\mu, \hat{\mu}_j, t_k)| \leq & |c^{ex,M}(\mu, \hat{\mu}_j, t_k) - c^{ex}(\mu, \hat{\mu}_j, t_k)| \\ \leq & K \times (\mu - \hat{\mu}_1) \times (\hat{\mu}_2 - \mu) \times |\beta'(\xi)|, \end{aligned} \tag{7.11}$$

where $\xi \in \mathcal{I}_{\gamma(\mu)}$, K is a positive constant independent of μ and M, and $\beta'(\xi) = \exp(\xi)/5$.

From the bound in (7.11), two conclusions follow. Firstly, for a given $\gamma(\mu)$, the bound is maximum at the midpoint $\mu = (\hat{\mu}_1 + \hat{\mu}_2)/2$. Secondly, for a given $\mu - \hat{\mu}_1$ (or equivalently $\hat{\mu}_2 - \mu$), $\exp(\xi)$ increases with

Parameter element	1	2	3	4
Error ratio (Min/Max)	2.2/4.0	2.0/5.50	3.0/5.6	1.7/4.4

Table 1: *Let $e_1(\mu)$ and $\tilde{e}_1(\mu)$ denote the value of $E_{ROM}(\mu)$ computed with* `S-ROM` *and* `SS-ROM`*, respectively. Then, the maximum error ratio in the i-th parameter element $\mathcal{I}_i$ is $\|e_1\|_{L^\infty(\mathcal{I}_i)}/\|\tilde{e}_1\|_{L^\infty(\mathcal{I}_i)}$. Similarly, one can define the minimum error ratio.*

ξ and since ξ increases with $\gamma(\mu)$, the bound increases with $\gamma(\mu)$. We expect the bound for $E_{ROM}(\mu)$ resulting from `SS-ROM` to behave the same as the above error bound; similar results can be found in [30]. The result in Fig 3a corroborates our expectation. The error E_{ROM} is the maximum at the midpoint of every parameter element, and, for a given $(\mu - \hat{\mu}_1)$, the error increases with $\gamma(\mu)$.

Solution from `S-ROM` accurately approximates $u^N(\cdot, t_k, \mu)$ if it is not dominated by transport with respect to the snapshots $u^N(\cdot, t_k, \hat{\mu}_1)$ and $u^N(\cdot, t_k, \hat{\mu}_2)$. In our context, the shift $(\beta(\mu) - \beta(\hat{\mu}_1))t$ captures the transport of $u^N(\cdot, t_k, \hat{\mu}_1)$ to $u^N(\cdot, t_k, \mu)$. Therefore, we expect the error $E_{ROM}(\mu)$ from `L-ROM` to be behave as

$$\min\{\beta(\mu) - \beta(\hat{\mu}_1), \beta(\hat{\mu}_2) - \beta(\mu)\}, \tag{7.12}$$

where $\beta(\mu) = \exp(\mu)/5$. Similar to the bound in (7.11), for a given $\gamma(\mu)$, $\min\{\exp(\mu) - \exp(\hat{\mu}_1), \exp(\hat{\mu}_2) - \exp(\mu)\}$ is the maximum at $\mu = \ln(\exp(\hat{\mu}_1)/2 + \exp(\hat{\mu}_2)/2)$ resulting in $E_{ROM}(\mu)$ having a local maxima at this point. Note that for our parameter domain, these points of local maxima are close to the mid-points of $\mathcal{I}_{\gamma(\mu)}$. Moreover, for a given $(\mu - \hat{\mu}_1)$, $\min\{\exp(\mu) - \exp(\hat{\mu}_1), \exp(\hat{\mu}_2) - \exp(\mu)\}$ increases with $\gamma(\mu)$ resulting in $E_{ROM}(\mu)$ increasing with $\gamma(\mu)$.

The maximum and the minimum error from the `S-ROM`, in each of the parameter elements, is at least 3 and 1.5 times higher than that from `SS-ROM`, respectively. Ratio of the maximum/minimum value of the error from the two methods is given in Table 1. The shifting in `SS-ROM` calibrates the snapshots that results in its higher accuracy as compared to `S-ROM`. Consider Fig 3d, which shows the ROM for $\mu = 2.75$ computed with snapshots taken from $\mathcal{M}_{t_k}$. After (approximately) $t = 0.2$, the snapshots become $L^2(\Omega)$-orthogonal, which results in two wave-fronts in the ROM moving with different speeds. These two wave-fronts correspond to the two spatially disjoint rays seen in Fig 3d, and their speeds correspond to the (x, t)-slopes of these rays. They miss-represent the single wave front in the FOM (see Fig 3b), which has a wave speed in between of the two wave-fronts. In contrast, the snapshots from $\mathcal{M}_{\mu,t_k}$ accurately capture the single wave-front in the FOM; see Fig 3c.

Convergence with (n_μ, M): For different values of n_μ (defined in (3.1)) and M (defined in (3.6)), we compare the error $\|E_{ROM}\|_{L^\infty(\mathcal{P})}$ resulting from `SS-ROM` and `S-ROM`. We start with $n_\mu = 1$ and $M = 2$, and perform five uniform refinements of the parameter domain where we increase each n_μ and M by one. To estimate $\|E_{ROM}\|_{L^\infty(\mathcal{P})}$, we consider 150 uniformly placed samples inside $\mathcal{P}$. We choose $n_x = 2 \times 10^{-3}$.

The results are shown in Fig 4a. Clearly, the error from `SS-ROM` appears to converge to zero much faster in comparison to the error from `S-ROM`. As studied above, the error from `SS-ROM` includes the error in shift computation that results from spatial discretization. Therefore, under a further increment of n_x, it might be possible to get error values lower than those reported in Fig 4a.

Study of hyper-reduction: Let the parameter elements be as given in (7.9). We fix μ to 2.75 and choose $n_x = 10^3$. We compute the set $\{\mathcal{E}_{\tilde{\mu}_i,t_k}\}_{i=1,2,k=1,\ldots,K}$ given in (4.10) for $\tilde{\mu}_1 = 2.625$ and $\tilde{\mu}_2 = 2.8750$. Both $\tilde{\mu}_1$ and $\tilde{\mu}_2$ belong to $\mathcal{I}_4$ and not to $\{\hat{\mu}_i\}_{i=1,2}$. Using $\{\mathcal{E}_{\tilde{\mu}_i,t_k}\}_{i=1,2,k=1,\ldots,K}$, we estimate the set $\{\mathcal{E}_{\mu,t_k}\}_{k=1,\ldots,K}$ using the relation in (4.11). Let $N^{hyp}_{\tilde{\mu}_i,t_k}$ be as given in (4.4). We choose the same $N^{hyp}_{\tilde{\mu}_i,t_k}$ for all time instances and for all $\tilde{\mu}_i$'s. We denote this $N^{hyp}_{\tilde{\mu}_i,t_k}$ by N^{hyp}. Starting with $N^{hyp} = 5$, we increment it by $N^\delta = 5$ till it reaches 200. Note that a value for N^{hyp} implies that $\#\mathcal{E}_{\tilde{\mu}_i,t_k} = N^{hyp}$. Since $\mathcal{E}_{\mu,t_k}$ is a union over the elements of $\mathcal{E}_{\tilde{\mu}_i,t_k}$ (see (4.11)), it is not necessary that $\#\mathcal{E}_{\mu,t_k} = N^{hyp}$. To measure the deviation of $\#\mathcal{E}_{\mu,t_k}$ from n_x, we define

$$\tilde{N} := n_x - \frac{\sum_{k=1}^{K} \#\mathcal{E}_{\mu,t_k}}{K}. \tag{7.13}$$

The minimum value of $\tilde{N}$ is zero and at this minimum value, for all time instance, $\#\mathcal{E}_{\mu,t_k} = n_x$ i.e., we select all the grid-points as our collocation points.

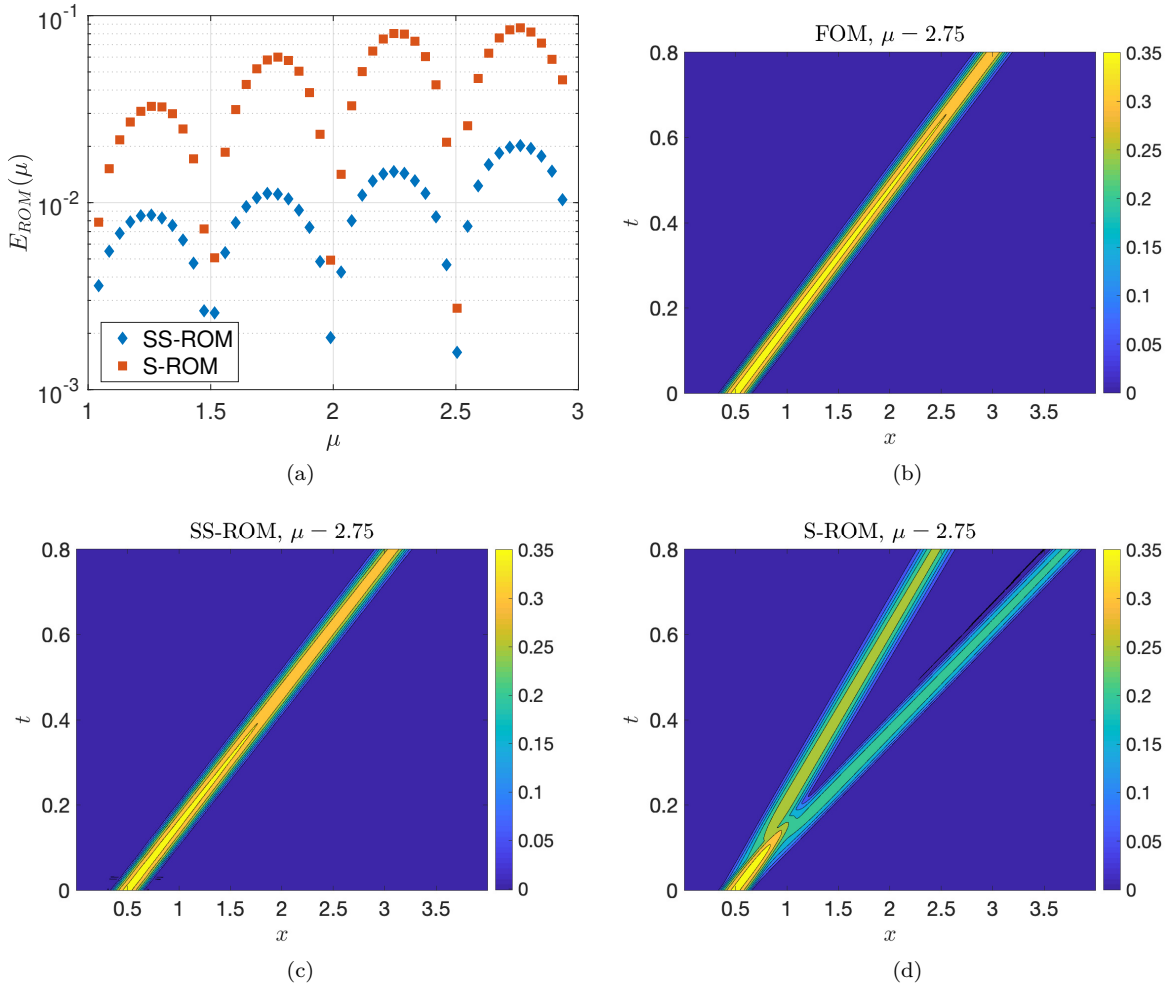


Figure 3: *Results for test-1. (a) E_{ROM}, given in (7.7), resulting from* `S-ROM` *and* `SS-ROM`*. (b) FOM, (c)* `SS-ROM`*, and (d)* `S-ROM` *for $\mu = 2.75$. Fig-(a) has a y-axis on a log-scale.*

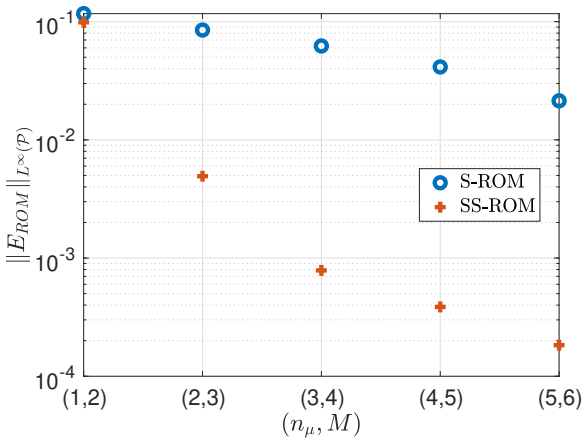


Figure 4: *Results for test-1. Error decay under (n_μ, M)-increment. Y-axis on a log-scale.*

Let $\hat{E}_{\mu,t_k}$ be as given in (4.13). As $\tilde{N} \to 0$, we expect $\max_{k\in\{1,\ldots,K\}} \hat{E}_{\mu,t_k} \to 0$. Fig 5a shows the convergence of $\max_{k\in\{1,\ldots,K\}} E_{\mu,t_k}$ with $\tilde{N}$. For the first few values of $\tilde{N}$, $\max_{k\in\{1,\ldots,K\}} \hat{E}_{\mu,t_k}$ converges slowly. However, (approximately) below $\tilde{N} = 900$, $\max_{k\in\{1,\ldots,K\}} \hat{E}_{\mu,t_k}$ converges fast and at a rate that

is close to 200 with respect to $\tilde{N}$. Already with $\tilde{N} = 900$, we reach an error value of $\mathcal{O}(10^{-4})$. Note that $\tilde{N} = 900$ corresponds to an average of $n_x - \tilde{N} = 100$ collocation points per time-step, which is 10% of the total grid-points. Recall that for $\mu = 2.75$ the error between the ROM and the FOM is $\mathcal{O}(10^{-2})$ therefore, an error of $\mathcal{O}(10^{-4})$ from hyper-reduction is acceptable.

For all the choices of N^{hyp}, $\#\mathcal{E}_{\mu,t_k}$ stays close to N^{hyp} implying a coincidence of most of the points in the union (4.10). Thus, even with a single parameter sample $\tilde{\mu}_1$, the set $\mathcal{E}_{\mu,t_k}$ remains almost the same and we can get the same results as reported here. For the first few times-steps, Fig 5b shows some of the entries in $\mathcal{E}_{\mu,t_k}$ for $N^{hyp} = 5$. Similar to the solution (see Fig 3b), the collocation points shift to the right as time progresses and follow the moving wave-front of the solution. Note that the shift in the collocation points is not the same for all time-steps. This results from an error in shift computation, which is of $\mathcal{O}(\Delta x)$, and from the error in approximating $c(\mu, \tilde{\mu}_i, t_k)$ in steps remark 9. At the expense of some offline cost, one can remove the later source of error by computing the shifts $c(\mu, \tilde{\mu}_i, t_k)$ using Lagrange interpolation.

We compare a gappy-POD/DEIM approximation of the residual to our approach. Such a DEIM-approximation results in the operator $\mathbb{P}_{t_{k+1}}$ given in (4.14). Here, for simplicity of notation, we suppress the projection operator in Res. We compute the residual $\mathrm{Res}(u^n(\cdot, t_{k+1}, \mu), u^n(\cdot, t_k, \mu))$, given in (3.8), for 100 equally spaced parameter points inside $\mathcal{I}_4$. Denoting these points by $\{\bar{\mu}_i\}_{i=1,\ldots,100}$, we define the snapshot matrix

$$\mathcal{U}_{t_{k+1}} := (\mathrm{Res}(u^n(\cdot, t_{k+1}, \bar{\mu}_1), u^n(\cdot, t_k, \bar{\mu}_1)), \ldots, \mathrm{Res}(u^n(\cdot, t_{k+1}, \bar{\mu}_{100}), u^n(\cdot, t_k, \bar{\mu}_{100}))) \,. \tag{7.14}$$

For every residual $\mathrm{Res}(u^n(\cdot, t, \bar{\mu}_1), u^n(\cdot, t, \bar{\mu}_1))$, we can define a piecewise constant function $\mathrm{res}(x, \mu, t)$ as

$$\mathrm{res}(\cdot, t, \mu) := \langle \mathrm{Res}(u^n(\cdot, t, \mu), u^n(\cdot, t, \mu)), \Phi \rangle_{\mathbb{R}^N} \,, \tag{7.15}$$

where Φ is a vector containing all the basis functions of a FOM and is given in (2.1). For $t = 0.8$ and $\mu \in \mathcal{I}_4$, $\mathrm{res}(\cdot, t, \cdot)$ is shown in Fig 6a. We scale all the values with $\|r(\cdot, t, \cdot)\|_{L^\infty(\Omega \times \mathcal{P})}$. Clearly, similar to the solution (see Fig 3b), which shifts to the right as t-increases, the residual also shifts to the right as μ-increases. This results in a slow decay in the singular values of $\mathcal{U}_{t_{k+1}}$. Fig 6b shows these singular values for $t_{k+1} = 0.8$. Although not shown in the plot, the decay gets slower as time progresses.

With a greedy-algorithm we compute the collocation points $\mathcal{E}_{t_{k+1}}$ corresponding to the collocation matrix $P_{t_{k+1}}$ given in (4.14). Details of the greedy-algorithm can be found in [5]. We perform 5 greedy iteration and in each of the iteration, we select 20 collocation points. This results in a total of 100 collocation points, which is equivalent to choosing $N^{hyp} = 100$. Owing to the slow decay of singular values of $\mathcal{U}_{t_{k+1}}$, we choose $U_{t_{k+1}}$ appearing in (4.14) as all the POD modes of $\mathcal{U}_{t_{k+1}}$.

Fig 6c shows the collocation points for $t = 0.8$, over-plotted on the residual for $(\mu, t) = (2.9, 0.8)$. The greedy-algorithm chooses points that are outside of the support of the residual while leaving out points where the residual is still non-zero. For the case shown in Fig 6c, almost 30% of the collocation points lie outside of the residual's support. This is because the greedy-algorithm selects the same collocation points for all the μ-values and does not adapt them to accommodate for a shifted residual. Therefore, we expect the error $\hat{E}_{\mu,t_{k+1}}$ from a DEIM approximation to decay slowly with $\tilde{N} = n_x - N^{hyp}$. This is also expected from the slow singular value decay of the snapshot matrix $\mathcal{U}_{t_{k+1}}$. In our approach, we shift the collocation points to the right with the solution. As a result, these points only populate the support of the residual. This is shown in Fig 6c. For both the DEIM approximation and our approach, we compute the error $\max_{k \in \{1,\ldots,K\}} \hat{E}_{\mu,t_k}$ at 45 uniformly placed parameter points inside $\mathcal{I}_4$. For the reasons mentioned above, the DEIM approximation has an error that is at least $\mathcal{O}(10^4)$ of our approach.

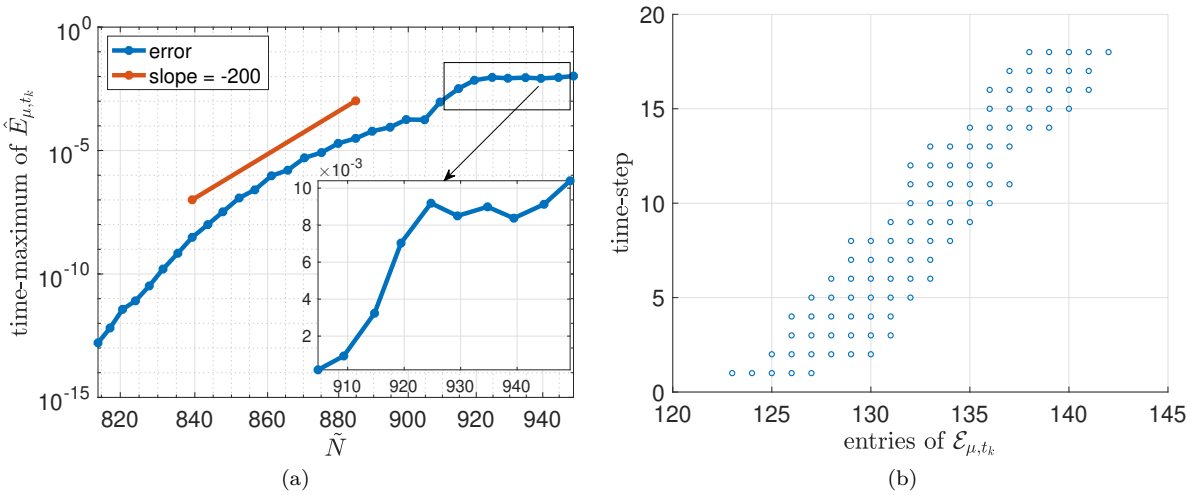
7.2 Test-2: Using the eigenvalue decomposition of the matrix A, we can express the wave-equation (7.3) as

$$\partial_t \begin{pmatrix} w^{(+)}(\cdot, \cdot, \mu) \\ w^{(-)}(\cdot, \cdot, \mu) \end{pmatrix} + \begin{pmatrix} \mu & 0 \\ 0 & -\mu \end{pmatrix} \partial_x \begin{pmatrix} w^{(+)}(\cdot, \cdot, \mu) \\ w^{(-)}(\cdot, \cdot, \mu) \end{pmatrix} = 0, \tag{7.16}$$

where $w^{(\pm)}$ are the characteristic variables. We reduce the above two equations independently using the framework discussed in earlier sections.

To account for the error in $w^{(\pm)}(\cdot, \cdot, \mu)$, we modify the error $E_{ROM}(\mu)$ given in (7.7) to

$$(E_{ROM}(\mu))^2 = \|w^{(+),N}(\cdot, \cdot, \mu) - w^{(+),n}(\cdot, \cdot, \mu)\|^2_{L^2(\Omega \times [0,T])} + \|w^{(-),N}(\cdot, \cdot, \mu) - w^{(-),n}(\cdot, \cdot, \mu)\|^2_{L^2(\Omega \times [0,T])}. \tag{7.17}$$



(a) (b)

Figure 5: *Results for test-1. (a) convergence of* $\max_{k\in\{1,\dots,N\}} \hat{E}_{\mu,t_k}$ *with* $\tilde{N}$ *for* $\mu = 2.75$*; (b) entries of* $\mathcal{E}_{\mu,t_k}$ *for the first few time steps. Fig-(a) has a y-axis on a log-scale.*

Above, $w^{(\pm),N}(\cdot,\cdot,\mu)$ and $w^{(\pm),n}(\cdot,\cdot,\mu)$ represent the FOM and the ROM of $w^{(\pm)}(\cdot,\cdot,\mu)$, respectively. We choose $n_x = 10^3$, $\Delta t = 4/(n_x \times 3)$, $n_\mu = 1$ in (3.1), and $M = 2$ in (3.6). This results in a piecewise linear approximation for the spatial transform. $n_\mu = 1$ results in a single parameter element given as $\mathcal{I}_1 = [1, 3]$. We let $\hat{\mu}_1 = 1$ and $\hat{\mu}_2 = 3$, where $\hat{\mu}_i$ are as given in (3.5).

We compute E_{ROM} at 45 uniformly placed parameter points different from the parameter samples $\hat{\mu}_1$ and $\hat{\mu}_2$. The error values at these parameter points resulting from `SS-ROM` and `S-ROM` are show in Fig 7a. Similar to the previous test case, `SS-ROM` performs much better than `S-ROM` and results in an error that is at least an order of magnitude lower than that resulting from `S-ROM`. The governing equations for w^+ and w^- (given in (7.16)) are the same as the linear advection equation (7.1) with the advection speeds $\beta_+(\mu) = \mu$ and $\beta_-(\mu) = -\mu$, respectively. Therefore, the explanation for the qualitative behaviour of the error is similar to the previous test case, and follows from the bound in (7.11) and (7.12).

The two transport modes of the current problem are shown in Fig 7b. In `SS-ROM`, the shifting of snapshots results in an accurate approximation of the two transport modes, see Fig 7c. In contrast, the result from `S-ROM` is a linear combination of two snapshots and since each of these snapshots have two distinct transport modes, their linear combination results in four distinct transport modes. These four transport modes are observable in Fig 7d after (approximately) $t = 0.3$. Results from hyper-reduction are similar to the previous test case and are not discussed for brevity.

7.3 Test-3: The previous test cases were one dimensional for which a FOM is already efficient. This makes it difficult to compare the efficiency of the FOM to that of the ROM, which we do so with the current test case. This test case also brings out a limitation of our method, which we discuss in detail later. We set $n_\mu = 1$ in (3.1) and $M = 2$ in (3.5). As sample parameters we choose $\hat{\mu}_1 = 1$ and $\hat{\mu}_2 = 3$. For spatial discretization we choose $n_x = 200$, which results in a spatial grid with 200×200 elements. For temporal discretization we choose $\Delta t = 10^{-3}$. Previous test cases show that choosing $N^{hyp} = 0.1 \times n_x^d$ number of collocation points for hyper-reduction provides acceptable results. Motivated from this observation we choose $N^{hyp} = 4000$. As sample parameters for hyper-reduction we choose $\tilde{\mu}_1 = 1.7$ and $\tilde{\mu}_2 = 2.7$, both of which do not belong to $\{\hat{\mu}_i\}_{i=1,2}$. To study the error resulting from the ROM and to analyse its performance, we compute the ROM for all $\mu \in \{1.2, 1.4, \dots, 2.8\}$. For computing the results from `S-ROM`, we do not use any hyper-reduction.

We define the speed-up $\mathcal{S}$ as

$$\mathcal{S} := \frac{\text{run-time of the FOM}}{\text{run-time of the online phase of the ROM}}.$$

Recall that the details of the online phase of `SS-ROM` are given in Algorithm 3.2. For `SS-ROM`, this speed-up is shown in Fig 8a. The speed-up is atleast 400, and results from introducing the operator $\mathbb{P}_{\mu,t_{k+1}}$ in the residual minimisation (4.1), which reduces the cost of both, evaluation of the residual and then

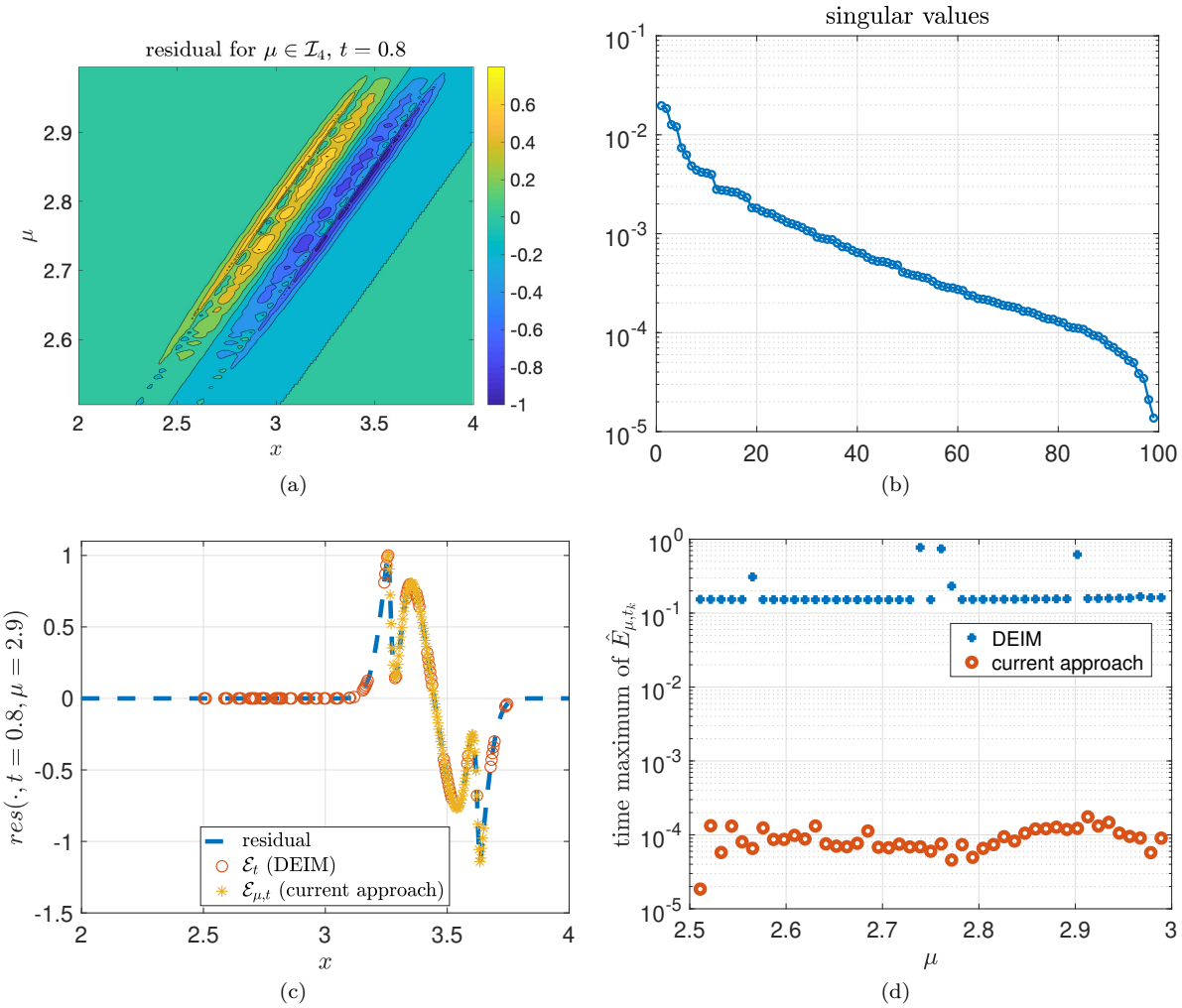


Figure 6: *Results for test-1. (a) The residual* $\text{res}(\cdot, t, \cdot)$ *given in* (7.15) *for* $\mu \in \mathcal{I}_4$, $x \in \Omega$, *and* $t_k = 0.8$. *(b) Singular values of the matrix* $\mathcal{U}_{t_{k+1}}$ *given in* (7.14) *for* $t_{k+1} = 0.8$. *(c) The location of the collocation points computed with a greedy DEIM approach and the current approach over-plotted on the residual for* $\mu = 2.9$ *and* $t = 0.8$. *(d) Comparison of* $\max_{k=1,\ldots,K} \hat{E}_{\mu,t_k}$. *Fig-(b) and (d) have a y-axis on a log-scale.*

its minimisation. Although we choose the same N^{hyp} and Δt for all parameter samples, the speed-up is not constant along $\mathcal{P}$. We provide the following explanation. The set of collocation points $\mathcal{E}_{\mu,t_k}$ (given in (4.11)) is a union over the set in $\{\mathcal{E}_{\tilde{\mu}_i,t_k}\}_{i=1,2}$, where the size of each of the sets is N^{hyp}. Taking a union results in $\#\mathcal{E}_{\mu,t_k}$ that is (in practice) slightly different from N^{hyp} and that changes with (μ, t). This results in the speed-up being non-constant along $\mathcal{P}$.

Fig 8b compares the error $E_{ROM}(\mu)$ between `SS-ROM` and `SS-ROM`. Similar to the previous test cases, error from both the methods drops close to the endpoints of the parameter domain and is maximum close the the mid-point. The error resulting from `S-ROM` is atleast 3.5 times higher than that resulting from `SS-ROM`. Superior performance of `SS-ROM` results from shifting the snapshots and is also observed in the previous test case.

As time progresses, the solution develops a shock. A part of the shock can be seen in the cross-section (along $x = y$) of the solution shown in Fig 9a. `SS-ROM` considers a span of shifted snapshots with a shift that aligns shocks between the snapshots. As a result, `SS-ROM` accurately captures the shock location, see Fig 9a. In contrast, `S-ROM` has the so-called staircase effect and wrongly captures the shock location. This results from approximating the solution in a span of non-shifted snapshots with each snapshot having a different shock location.

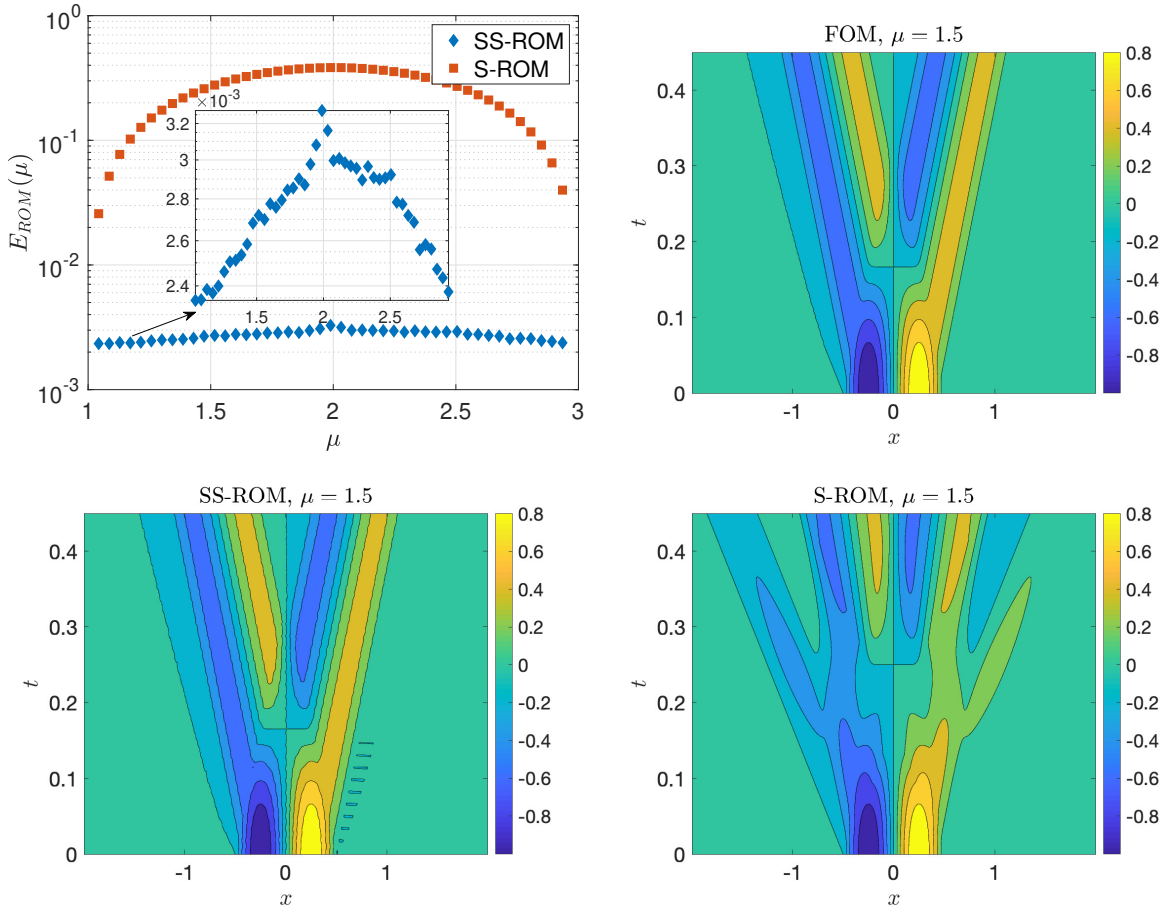


Figure 7: *Results for test-2. (a) E_{ROM}, given in (7.17), resulting from* `S-ROM` *and* `SS-ROM`*. (b) FOM, (c)* `SS-ROM` *and (d)* `S-ROM` *of $u^{(1)}(\cdot,\cdot,\mu)$ for $\mu = 1.5$. Fig-(a) has a y-axis on a log-scale. The results are similar for $u^{(2)}(\cdot,\cdot,\mu)$.*

7.4 Limitations: The previous test case brings out a drawback of our method. The support of the initial data $u_0(\cdot,\mu)$ given in (7.5) does not change with μ. This results in the solution having a support, a part of the boundary of which, does not change with μ. A part of this boundary is shown in Fig 9a around $x = 0.36$. We refer to this location as the starting location of the support. Despite of a difference in the shock locations, the starting location of the support is the same for every snapshot. Therefore, `S-ROM` captures this starting location accurately. However, because we shift the snapshots in `SS-ROM`, we capture the starting location of the support inaccurately.

The limitation of the method becomes clearer with the following example. Consider the one-dimensional Burger's equation, which is the same as (7.4) without the y-derivative, with the initial data

$$u_0(x) = \begin{cases} \mu \exp\left(-1/\left(1-\left(\frac{x-\delta_1}{\delta_2}\right)^2\right)\right), & \left|\frac{x-\delta_1}{\delta_2}\right| < 1 \\ -\exp\left(-1/\left(1-\left(\frac{x+\delta_1}{\delta_2}\right)^2\right)\right), & \left|\frac{x+\delta_1}{\delta_2}\right| < 1 \\ 0, & \text{else} \end{cases}. \tag{7.18}$$

We let $\delta_1 = 0.5$ and $\delta_2 = 0.2$. Let $\Omega = [-2, 2]$ and $T = 1.0$. Let $\mathcal{P} = [1, 3]$ be discretized with one parameter element i.e., $n_\mu = 1$ in (3.1). Moreover, let $M = 2$ in (3.5). Let Ω be discretized with $n_x = 10^3$ elements, and let $\Delta t = 10^{-3}$.

The FOM has two shocks, both of which are shown in Fig 10a for $\mu = 2$ and $t = T$. Since in `SS-ROM`

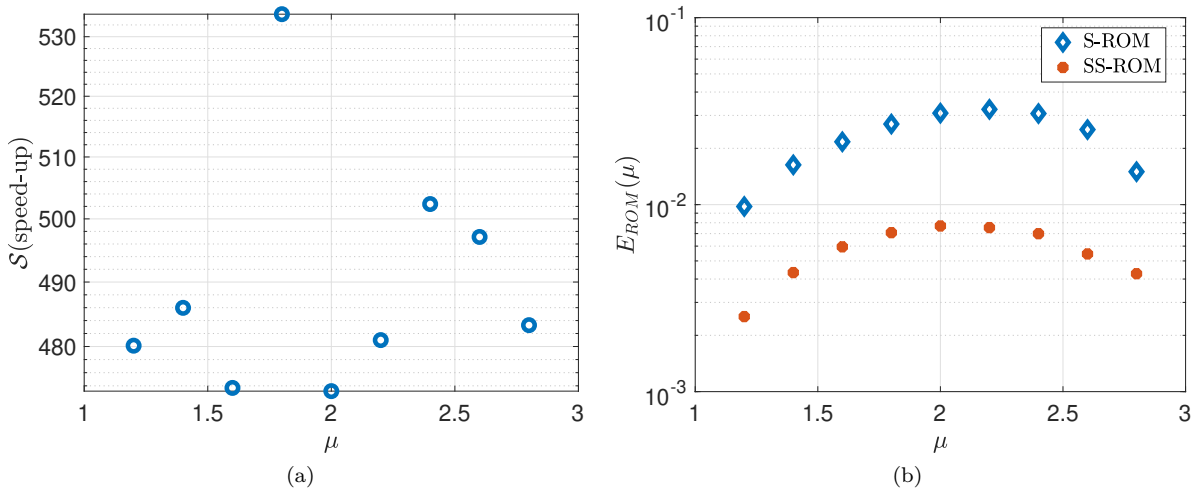


Figure 8: *Results for test-3. (a) Speed-up resulting from* `SS-ROM`*; and (b)* E_{ROM} *resulting from* `S-ROM` *and* `SS-ROM`*. Fig-(b) has a y-axis on a log-scale.*

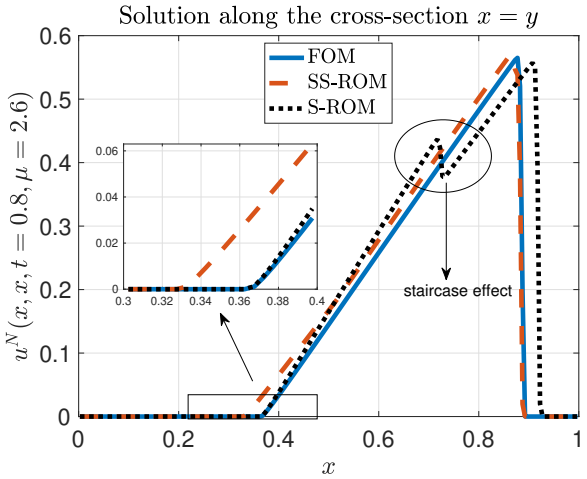


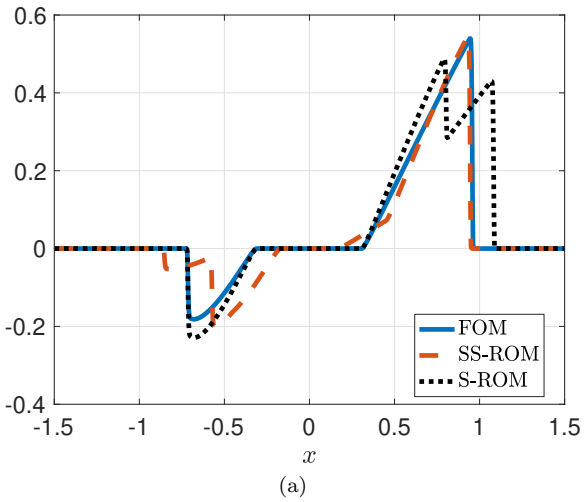
Figure 9: *Results for test-3. FOM and the ROM along the cross-section* $x = y$ *for* $\mu = 2.6$ *and* $t = 0.8$.

the snapshots shift in one direction, we capture only one of the shocks accurately. Aligning the shock on the right results in a lower L^2 error (i.e., $\mathcal{R}$ in (5.3)) than aligning the shock on the left. Therefore, `SS-ROM` accurately captures the shock on the right but has a staircase effect at the shock on the left.

Since the negative part of $u_0(\cdot, \mu)$ is independent of μ, the shock location on the left is μ-independent. Therefore, `S-ROM` captures it accurately. However, it is highly inaccurate at the right shock and shows the staircase effect. Despite the inaccuracy from `SS-ROM` at the left shock, it has an L^2-error of 0.06 that is less than the L^2 error of 0.08 from `S-ROM`. However, the benefit of using `SS-ROM` over `S-ROM` is not much, and both the ROMs perform poorly.

8. Conclusion

We considered a transformed snapshot based non-linear approximation for solution manifolds of hyperbolic equations. To compute the ROM, we used residual minimisation. Projecting the residual onto a low-dimensional linear space, which is usually a span of POD-basis, usually makes the residual minimisation efficient. However, computational examples showed that such a projection could be ineffective for hyperbolic problems. Broadly speaking, this is because (similar to the solution) the residual has a moving wave-type behaviour. To circumvent a projection, we considered hyper-reduction that evaluates and minimises the residual on a subset of mesh-elements or the so-called collocation points. For the



(a)

Figure 10: *Results that show the limitation of* `SS-ROM`*. Computed with the one-dimensional Burger's equation with the initial data as given in* (7.18)*. The solutions are for* $\mu = 2$ *and* $t = 1$*.*

computation of the collocation points, we considered an offline and an online stage. Offline, we computed the collocation points for a set of training parameters by minimising a bound on the L^2-error of our ROM. Moreover, online, we transported the set of collocation points computed offline.

As an instance of our non-linear approximation space, we considered a span of shifted snapshots with local in time and parameter shifts. We considered shifts that align shocks and local minima/maxima in the solution while minimising the L^2-error between the shifted snapshots. We were efficient with our shift computation for solutions that do not have a very large number of shocks and local minima/maxima. Moreover, for a certain class of problems, our algorithm provides a shift between snapshots that have arbitrarily separated parameters.

With numerical experiments, we compared the accuracy of our non-linear ROM to a ROM that uses the span of snapshots as its approximation space. The non-linear ROM had an L^2-error that was 2 to 10 times lower than that resulting from a snapshots based ROM. For a test-case involving the 2D Burger's equation, as compared to the FOM, the non-linear ROM showed a speed-up of at least 400. The speed-up was a result of hyper-reduction, which reduced the cost of both the evaluation and the minimisation of the residual.

9. Acknowledgements

Supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the joint project "MathEnergy - Mathematical Key Technologies for Evolving Energy Grids", sub-project: Model Order Reduction (Grant number: 0324019B).

Appendix A. Proof of Lemma 5.1. By definition

$$\left\langle \Phi, \Pi_{\mu,t_{j+1}} v(t_{j+1}) \right\rangle = \mathrm{Res}(\Pi_{\mu,t_{j+1}} v(t_{j+1}), \Pi_{\mu,t_j} v(t_j)) + \left\langle \Phi, \Pi_{\mu,t_j} v(t_j) \right\rangle + \Delta t \left\langle \Phi, L^N_\mu(\Pi_{\mu,t_j} v(t_j)) \right\rangle .$$

where $j \in \{1, \ldots, K-1\}$. The result for $j = 0$ is trivial. Also,

$$\left\langle \Phi, u^N(\cdot, t_{j+1}, \mu) \right\rangle = \left\langle \Phi, u^N(\cdot, t_j, \mu) \right\rangle + \Delta t_j \left\langle \Phi, L^N_\mu(u^N(\cdot, t_j, \mu)) \right\rangle$$

Subtracting the above two relations, taking the $\mathbb{R}^N$ norm on both sides and using triangle's inequality provides

$$\begin{aligned} \| \left\langle \Phi, u^N(\cdot, t_{j+1}, \mu) - \Pi_{\mu,t_{j+1}} v(t_{j+1}) \right\rangle \|_{\mathbb{R}^N} \leq & \left\| \left\langle \Phi, u^N(\cdot, t_k, \mu) - \Pi_{\mu,t_j} v(t_j) \right\rangle \right. \\ & \left. + \Delta t_j \left\langle \Phi, L^N_\mu(u^N(\cdot, t_j, \mu)) - L^N_\mu(\Pi_{\mu,t_j} v(t_j)) \right\rangle \right\|_{\mathbb{R}^N} \\ & + \| \mathrm{Res}(\Pi_{\mu,t_{j+1}} v(t_{j+1}), \Pi_{\mu,t_j} v(t_j)) \|_{\mathbb{R}^N} . \end{aligned} \tag{A.1}$$

Using Lipschitz continuity of $\mathrm{Id} + \Delta t_j L_\mu^N$ provides

$$\begin{aligned} \left\| \left\langle \Phi, u^N(\cdot, t_k, \mu) - \Pi_{\mu,t_j} v(t_j) \right\rangle + \Delta t_j \left\langle \Phi, L_\mu^N(u^N(\cdot, t_j, \mu)) - L_\mu^N(\Pi_{\mu,t_j} v(t_j)) \right\rangle \right\|_{\mathbb{R}^N} \\ \leq C \| \left\langle \Phi, u^N(\cdot, t_j, \mu) - \Pi_{\mu,t_j} v(t_j) \right\rangle \|_{\mathbb{R}^N}. \end{aligned} \tag{A.2}$$

Substituting the above bound into (A.1) provides

$$\begin{aligned} \| \left\langle \Phi, u^N(\cdot, t_{j+1}, \mu) - \Pi_{\mu,t_{j+1}} v(t_{j+1}) \right\rangle \|_{\mathbb{R}^N} \leq C \| \left\langle \Phi, u^N(\cdot, t_j, \mu) - \Pi_{\mu,t_j} v(t_j) \right\rangle \|_{\mathbb{R}^N} \\ + \| \mathrm{Res}(\Pi_{\mu,t_{j+1}} v(t_{j+1}), \Pi_{\mu,t_j} v(t_j)) \|_{\mathbb{R}^N} \end{aligned} \tag{A.3}$$

The result follows by applying recursion to the above bound.

References.

[1] Abgrall, R., Amsallem, D., and Crisovan, R. (2016). Robust model reduction by L1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):1.

[2] Amsallem, D., Zahr, M. J., and Farhat, C. (2012). Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916.

[3] Benner, P., Gugercin, S., and Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531.

[4] Cagniart, N., Maday, Y., and Stamm, B. (2019). Model order reduction for problems with large convection effects. In *Contributions to Partial Differential Equations and Applications*, pages 131–150. Springer International Publishing, Cham.

[5] Carlberg, K., Farhat, C., Cortial, J., and Amsallem, D. (2013). The GNAT method for non-linear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623 – 647.

[6] Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764.

[7] Cohen, A. and DeVore, R. (2015). Kolmogorov widths under holomorphic mappings. *IMA Journal of Numerical Analysis*, 36(1):1–12.

[8] Dahmen, W., Plesken, C., and Welper, G. (2014). Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: M2AN*, 48(3):623–663.

[9] Drohmann, M., Haasdonk, B., and Ohlberger, M. (2012). Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969.

[10] Eymard, R., Gallouët, T., and Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, 7:713–1018.

[11] Gerbeau, J. F. and Lombardi, D. (2014). Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *Journal of Computational Physics*, 265:246 – 269.

[12] Greif, C. and Urban, K. (2019). Decay of the Kolmogorov N-width for wave problems. *Applied Mathematics Letters*, 96:216 – 222.

[13] Haasdonk, B. and Ohlberger, M. (2008). Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302.

[14] Kunisch, K. and Volkwein, S. (2001). Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148.

[15] LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press.

[16] McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

[17] Mowlavi, S. and Sapsis, T. P. (2018). Model order reduction for stochastic dynamical systems with continuous symmetries. *SIAM Journal on Scientific Computing*, 40(3):A1669–A1695.

[18] Nair, N. J. and Balajewicz, M. (2019). Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *International Journal for Numerical Methods in Engineering*, 117(12):1234–1262.

[19] Ohlberger, M. and Rave, S. (2013). Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathematique*, 351(23):901 – 906.

[20] Peherstorfer, B. (2018). Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *arXiv:1812.02094*.

[21] Reiss, J., Schulze, P., Sesterhenn, J., and Mehrmann, V. (2018). The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344.

[22] Rim, D. and Mandli, K. T. (2018). Displacement interpolation using monotone rearrangement. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1503–1531.

[23] Rim, D., Moe, S., and LeVeque, R. J. (2018). Transport reversal for model reduction of hyperbolic partial differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):118–150.

[24] Rowley, C. W. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013.

[25] Rowley, C. W. and Marsden, J. E. (2000). Reconstruction equations and the Karhunen–Loéve expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1):1 – 19.

[26] Rozza, G., Huynh, D. B. P., and Patera, A. T. (2008). Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229.

[27] Tadmor, E. (2003). Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12:451–512.

[28] Veroy, K., Prud'Homme, C., Rovas, D., and Patera, A. (2003). A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*, page 3847.

[29] Welper, G. (2017a). h and hp-adaptive interpolation by transformed snapshots for parametric and stochastic hyperbolic pdes. *arXiv:1710.11481*.

[30] Welper, G. (2017b). Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250.