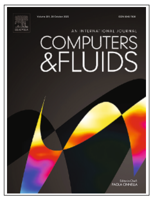
Computers and Fluids 305 (2026) 106911

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid



Modeling advection-dominated flows with space-local reduced-order models

T. Van Gastelen [a,b,*], W. Edeling [a], B. Sanderse [a,b]

[a] Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, The Netherlands
[b] Centre for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, PO Box 513, 5600 MB, Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Reduced-order models
Proper orthogonal decomposition
Advection-dominated flows
Space-local basis functions
Energy conservation

ABSTRACT

Reduced-order models (ROMs) are often used to accelerate the simulation of large physical systems. However, traditional ROM techniques, such as proper orthogonal decomposition (POD)-based methods, often struggle with advection-dominated flows due to the slow decay of singular values. This results in high computational costs and potential instabilities.

This paper proposes a novel approach using space-local POD to address the challenges arising from the slow singular value decay. Instead of global basis functions, our method employs local basis functions that are applied across the domain, analogous to the finite element method, but with a data-driven basis. By dividing the domain into subdomains and applying the space-local POD, we obtain a sparse representation that generalizes better outside the training regime. This allows the use of a larger number of basis functions compared to standard POD, without prohibitive computational costs. To ensure smoothness across subdomain boundaries, we introduce overlapping subdomains inspired by the partition of unity method.

Our approach is validated through simulations of the 1D and 2D advection equation. We demonstrate that using our space-local approach, we obtain a ROM that generalizes better to flow conditions not included in the training data. In addition, we show that the constructed ROM inherits the energy conservation and non-linear stability properties from the full-order model. Finally, we find that using a space-local ROM allows for larger time steps.

1. Introduction

Simulating large physical systems is an ongoing challenge in the field of computational sciences. This especially becomes challenging when dealing with multiscale systems. Such systems exhibit interesting behavior at various spatial and temporal scales. A prominent example and our main incentive for this work are turbulent flows described by the incompressible Navier-Stokes equations. Systems described by these equations feature the formation of turbulent eddies of a range of different sizes. The significant difference in size between the largest and the smallest eddies gives rise to the multiscale nature of the problem. The problem with simulating such systems is that they require high-resolution computational meshes to obtain accurate simulations and small time steps. This places a significant burden on the available computational resources [1,2].

To make simulation of turbulent flows feasible, one typically resorts to Reynolds averaged Navier-Stokes [3], large eddy simulation [4], and reduced-order models (ROMs) [5,6]. Here we focus on the latter approach. In reduced-order modelling, data is used to speed up simulations. The data can be obtained from simulations or experiments. The collected data is then used to identify the most critical features of the flow. This is typically done by a proper orthogonal decomposition (POD) of the collected flow data. The resulting features are then used to construct a reduced basis. By projecting the fluid flow equations on this basis one obtains the widely used POD-Galerkin ROM [6–8]. However, two common issues with POD-Galerkin ROMs are their stability and their accuracy for convection-dominated systems. In [9], it is shown that the stability issue can be resolved by making sure that the energy-conserving property of the Navier-Stokes equations is still satisfied under Galerkin projection. For the incompressible Navier-Stokes equations, the only prerequisite for an energy-conserving ROM is that the discretization is structure-preserving, i.e. it is such that it inherits the energy conservation property from the continuous equations.

However, the stability property derived in [9] does not guarantee accuracy; it is possible to have energy-stable simulations that are highly inaccurate. This issue can already be observed on academic test cases

* Corresponding author.
E-mail address: tobyvangastelen@gmail.com (T. Van Gastelen).

https://doi.org/10.1016/j.compfluid.2025.106911
Received 22 July 2025; Received in revised form 6 November 2025; Accepted 10 November 2025
Available online 15 November 2025
0045-7930/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

such as the linear advection equation. For example, in [5], the authors discuss the problem of a single traveling wave through a periodic domain. Even though it is clear that a one-dimensional representation of the system exists, it is not recovered by the POD algorithm. The result is a slow decay in singular values of the snapshot matrix. This means that many POD modes are required to accurately describe the flow. The slow decay in singular values is often related to the Kolmogorov N-width. This is a measure of how well the solution space of a partial differential equation (PDE) can be represented by a linear combination of N basis functions [10,11]. Advection-dominated flows are notorious for displaying a slow Kolmogorov N-width decay, requiring a large N for accurate simulation. The resulting ROM is then expensive to evaluate [12,13]. When not including a sufficient number of basis functions, inaccurate results are obtained, e.g., they contain oscillations [14,15].

Different approaches to deal with this problem have been suggested. One way of dealing with this is by taking a local in time approach [16]. In this approach, the ROM switches basis for different time intervals. This means that the ROM can employ a smaller basis, since each basis is specialized to deal with a specific time interval. Switching between bases during simulation can also be done in a structure-preserving (energy-conserving) manner [17]. A related approach is to update the basis during the ROM simulation to make it more robust to changes in simulation conditions [18]. In [19], a Petrov-Galerkin approach is suggested, which leads to more stable ROMs when using a small POD basis than the standard Galerkin approach. In [20], it is suggested to add a closure model to the ROM to account for a small POD basis. The closure model is then tasked with modeling the interaction between the part of the flow that is covered by the POD basis and the part that is removed. Another suggestion is to construct ROMs on non-linear reduced subspaces instead of linear ones. In [21], a more accurate representation of the solution space is obtained using rational quadratic manifolds. Here, the ROM construction was also performed in a structure-preserving (entropy-stable) manner, yielding stability of the ROM.

Machine learning approaches have also gained traction for the construction of ROMs. In [22], a long short-term memory (LSTM) neural network is used instead of Galerkin projection, as the computational complexity of LSTM is more favorable than Galerkin projection-based ROMs. They also allow one to take larger time steps [23]. In [23], this approach is combined with an autoencoder to reduce the dimensionality of the system. It is demonstrated that a significantly greater reduction in degrees of freedom (DOF) can be achieved using this approach compared to the standard POD-Galerkin approach. In [10], an advection-aware autoencoder is suggested to reduce the dimensionality of this system. This is achieved by introducing an additional decoder which is trained to reconstruct a “shifted” version of the encoded snapshots. This shift can, for example, be a snapshot taken at a later time. This forces the latent space of the autoencoder to become aware of the dominant advective features of the flow.

In this work, we focus on an alternative approach to address the issue of ROM accuracy in advection-dominated flows. Namely, we propose to use the idea of a space-local POD to tackle the slow singular value decay of advection-dominated flows. The idea is as follows: instead of obtaining a set of global basis functions that span the entire domain, we obtain a set of space-local basis functions. These basis functions are repeated across the entire domain, similarly to how a finite element basis covers the domain [24]. Furthermore, they are only nonzero on their designated subdomain. The advantage of this is that the resulting Galerkin projected operators are sparse. This makes them much cheaper to evaluate when simulating the ROM. For example, in [25] a speedup of up to 1.5 orders of magnitude is reported with respect to a standard POD-Galerkin ROM. Our approach differs from the one presented in [25]. In our approach, the solution data in each subdomain is treated with the same local POD basis, rather than obtaining a different local POD basis for each subdomain. This approach ensures that a feature observed in part of the domain can also be represented in a different part of the domain using the same basis. In this way, less data is required to obtain a POD basis that generalizes well. Note that this approach is designed specifically for problems where the dynamics are similar throughout the domain, such as a traveling wave. For problems where the dynamics are more variable throughout the domain, the approach suggested by [25] is likely still superior, as it builds a specialized basis for each subdomain. For diffusion-dominated problems the standard space-global POD approach is likely still superior due to the rapid Kolmogorov N-width decay [26]. Note that while our approach does not directly solve the slow Kolmogorov N-width decay (we still use linear approximations), the sparsity of the basis allows us to use a much larger number of basis functions. Sparsity and generalizability are therefore key features of our approach. Furthermore, the authors note this approach is similar to the methodology described in [27] where different types of subdomains were specified, each with a shared POD basis. These subdomains were then used as building blocks to extrapolate the POD basis obtained from small spatial domains to larger domains. Our work differs in the fact that it focuses on temporal extrapolation for time-dependent PDEs, as opposed to steady-state problems. In addition, we introduce another novel idea in this work: the use of overlapping subdomains (to avoid discontinuities at the subdomain boundaries). Lastly, we show that our space-local ROMs satisfy energy conservation if the full-order model (FOM) does. In this way, stability of the ROM is guaranteed.

This paper is structured in the following way. In Section 2 we introduce the FOM, namely a central difference discretization of the 1D advection equation. The central difference discretization ensures the scheme is energy conserving. In Section 3 we introduce the POD approaches. We begin with the standard POD approach and then introduce two space-local approaches, one with and one without overlapping subdomains. In Section 4 we use Galerkin projection to project the FOM onto the POD basis and show that the resulting ROMs satisfy energy conservation. Finally, in Section 5, we evaluate the different POD-based approaches in numerical experiments using a set of different metrics. In addition, we assess the performance of the ROMs on a case where we extrapolate beyond the data used to construct the POD basis. Furthermore, we investigate the energy-conserving properties of the ROMs. To conclude this section, we apply the introduced methodologies to a 2D advection equation test case and evaluate the computational efficiency, among other metrics. Finally, in Section 6, we present our main findings and suggest future research topics.

2. Full-order model

2.1. Advection equation

In this work, we focus on developing a reduced-order model for the linear advection equation in both one-dimensional (1D) and two-dimensional (2D) settings. This equation is chosen as it exhibits similar difficulties as the incompressible Navier-Stokes equations when applying model reduction. For the sake of simplicity, we discuss the properties of the system for 1D, but the ideas easily carry over to 2D. To start, we consider a scalar solution $u(x,t)$ to the linear advection equation

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}, \tag{1}$$

with initial condition $u(x,0) = u_0(x)$ and constant c. In this work we stick to periodic boundary conditions (BCs). An important property of this equation is that the total energy of the system

$$E := \frac{1}{2}(u,u), \tag{2}$$

is conserved, where

$$(a(x), b(x)) := \int_\Omega a(x)b(x)\mathrm{d}\Omega \tag{3}$$

on the spatial domain Ω. This can easily be shown using the product rule of differentiation:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(u,u) = \left(u, \frac{\partial u}{\partial t}\right) = -c\left(u, \frac{\partial u}{\partial x}\right) = c\left(\frac{\partial u}{\partial x}, u\right) = 0. \tag{4}$$

In the final step, we carried out integration by parts and used the fact that the boundary term cancels on periodic domains. The energy-conserving property of this equation will be mimicked by the discretization and by the ROMs developed in this work. This leads to unconditionally stable methods, as in [9].

2.2. Finite difference discretization

Although Eq. (1) can be easily solved exactly, for more complex equations, this is not the case. In general, we approximate the solution by representing $u(x,t)$ on a grid. In this case, we employ a uniform grid with grid spacing $h = \frac{|\Omega|}{N}$ such that $\mathrm{u}_i(t) \approx u(x_i,t)$. The approximated solution is contained within the state vector $\mathbf{u}(t) \in \mathbb{R}^N$. To approximate the spatial derivative, we use a central difference approximation:

$$\frac{\partial u}{\partial x}\Big|_{x_i} \approx \frac{\mathrm{u}_{i+1} - \mathrm{u}_{i-1}}{2h}. \tag{5}$$

This leads to the following semi-discrete system of equations

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -c\mathbf{D}\mathbf{u}, \tag{6}$$

where the linear operator $\mathbf{D} \in \mathbb{R}^{N\times N}$ is skew-symmetric and encodes the stencil in (5). For the time integration, we use a classic Runge-Kutta 4 (RK4) scheme [28]. This time integration scheme introduces a small energy conservation error, which is negligible in our test cases. Alternatively, one can use the implicit midpoint method for exact energy conservation in time [29]. Eq. (6) will be regarded as the full-order model (FOM). We note that other discretization techniques, such as the finite element method, can also be employed to derive a FOM; our framework in Section 3 is also applicable in this context.

2.3. Energy conservation of the FOM

It is well known that this FOM mimics the energy-conservation property (4) in a discrete setting. This can be shown by defining the discretized energy as

$$E_h := \frac{h}{2}\mathbf{u}^T\mathbf{u}, \tag{7}$$

where the inclusion of h is such that this definition discretely represents the inner product in (2).

Using the product rule, we obtain the following evolution equation for the energy

$$\frac{\mathrm{d}E_h}{\mathrm{d}t} = \frac{h}{2}\frac{\mathrm{d}\mathbf{u}^T\mathbf{u}}{\mathrm{d}t} = h\mathbf{u}^T\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -ch\mathbf{u}^T\mathbf{D}\mathbf{u} = ch\mathbf{u}^T\mathbf{D}^T\mathbf{u} = 0. \tag{8}$$

In the final step, we used the fact that $\mathbf{D}$ is skew-symmetric, i.e., $\mathbf{D} = -\mathbf{D}^T$, to see that the energy is conserved using this stencil. This yields both stability and consistency with the continuous equation.

3. Local and global POD

To reduce the computational cost of solving the FOM, we construct a ROM. For this purpose, we use simulation data to build a data-driven basis. This basis is obtained through a POD of the simulation data [5,6]. The most common approach is to employ a global POD basis, defined over the entire simulation domain, similar to a Fourier basis. An alternative is a local basis, as proposed in [25,27]. In this section, we discuss both approaches. We will present our version of the space-local POD framework for finite difference discretizations, see (5). However, the ideas can also be applied to different discretization techniques, as shown in [27]. Our space-local approach has some key differences from the one presented in [25], which will be highlighted.

3.1. Global POD

In the global approach we first express the discrete solution $u_h(x,t)$ in terms of an orthogonal basis $\{\psi_i\}$:

$$u(x,t) \approx u_h(x,t) = \sum_{i=1}^{N} \mathrm{c}_i(t)\psi_i(x). \tag{9}$$

Here ψ_i can represent, for example, a Fourier basis or a localized box function in the case of the presented finite difference discretization. In the finite difference case we have $\mathbf{c}(t) = \mathbf{u}(t)$ and

$$\psi_i(x) = \begin{cases} 1 & \text{if } x_i - \frac{h}{2} \le x < x_i + \frac{h}{2}, \\ 0 & \text{elsewhere.} \end{cases} \tag{10}$$

For finite element discretizations, one could use, e.g., Löwdin orthogonalization to express the solution in terms of an orthogonal basis [30].

The snapshot matrix $\mathbf{X}$ is constructed from the coefficient vector at different points in time

$$\mathbf{X} = \begin{bmatrix}\mathbf{c}(t_1) & \mathbf{c}(t_2) & \dots & \mathbf{c}(t_s)\end{bmatrix} \quad \in \mathbb{R}^{N\times s}, \tag{11}$$

where s is the number of snapshots. We decompose this snapshot matrix using a singular value decomposition (SVD) [31]

$$\mathbf{X} = \tilde{\mathbf{\Phi}}\mathbf{\Sigma}\mathbf{V}^T. \tag{12}$$

We use the first r left-singular vectors in $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{N\times N}$ to obtain a reduced set of orthonormal basis vectors $\mathbf{\Phi} \in \mathbb{R}^{N\times r}$. This basis minimizes the projection error in the Frobenius norm:

$$\mathbf{\Phi} = \underset{\mathbf{M}\in\mathbb{R}^{N\times r}}{\arg\min}\, ||\mathbf{X} - \mathbf{M}\mathbf{M}^T\mathbf{X}||_F^2, \tag{13}$$

under the orthonormality constraint $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$ [5–8]. We can project the coefficient vector onto this basis as follows:

$$\mathbf{a}(t) = \mathbf{\Phi}^T\mathbf{c}(t) \quad \in \mathbb{R}^r, \tag{14}$$

such that projecting back onto the FOM space yields the approximation

$$\mathbf{c}(t) \approx \mathbf{c}_r(t) := \mathbf{\Phi}\mathbf{a}(t). \tag{15}$$

By substituting this into (9) we obtain the approximated solution $u_r(x,t)$:

$$u_h(x,t) \approx u_r(x,t) = \sum_{i=1}^{N} \mathrm{c}_{r,i}(t)\psi_i(x). \tag{16}$$

We can also write this approximation in terms of the POD basis expansion:

$$\begin{aligned} u_r(x,t) &= \sum_{i=1}^{N} \mathrm{c}_{r,i}(t)\psi_i(x) = \sum_{i=1}^{N}(\mathbf{\Phi}\mathbf{a}(t))_i\psi_i(x) = \sum_{i=1}^{N}(\sum_{j=1}^{r}\Phi_{ij}\mathrm{a}_j(t))\psi_i(x) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{r}\Phi_{ij}\mathrm{a}_j(t)\psi_i(x) = \sum_{j=1}^{r}\mathrm{a}_j(t)\sum_{i=1}^{N}\Phi_{ij}\psi_i(x) = \sum_{j=1}^{r}\mathrm{a}_j(t)\phi_j(x), \end{aligned} \tag{17}$$

where the reduced POD basis $\{\phi_j\}$ is obtained by applying the following transformation:

$$\phi_j(x) = \sum_{i=1}^{N}\Phi_{ij}\psi_i(x). \tag{18}$$

As the obtained basis functions span the entire domain, we refer to this approach as space-global proper orthogonal decomposition (G-POD).

3.2. Space-local POD

In space-local proper orthogonal decomposition (L-POD) we take a different approach from G-POD. We start by assuming a uniform finite difference discretization for the discrete FOM solution $u_h(x,t)$.[1] Once

[1] If $u_h(x,t)$ would stem from a simulation on an unstructured grid, one could potentially first project the discrete solution on a uniform grid, after which the presented methodology could still be applied. We consider this outside the scope of this research.

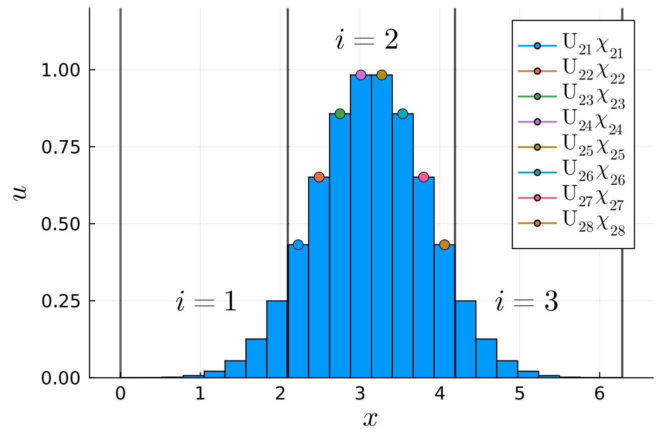


Fig. 1. A Gaussian wave discretized by a finite difference scheme represented by a local basis of box functions for $I = 3$ subdomains and $J = 8$ points per subdomain. The edges of the subdomains are indicated by the vertical grey lines.

the solution is represented on this grid, we subdivide the domain Ω into I non-overlapping subdomains $\Omega_i = [\alpha_i, \beta_i)$, $i = 1 \dots I$, such that $\alpha_i < \beta_i = \alpha_{i+1} < \beta_{i+1}$. Later in this section, these subdomains will be used to construct the L-POD basis. Furthermore, we assume each of these subdomains contains exactly J grid points. This means the total number of grid points N has to equal the product $N = I \cdot J$. This is essential for our methodology to work, as for the L-POD procedure, each subdomain is treated equally in the snapshot matrix, which is only possible if they contain the same number of grid points. This helps us obtain a basis that generalizes better outside the training data for advection-dominated problems, where the dynamics are similar throughout the domain. This approach is fundamentally different from what is presented in [25], where each subdomain has its own POD basis. The latter is more suitable for problems where the dynamics differ throughout the domain, such that the local bases can be specialized to these dynamics. If N/I is not an integer, one could possibly resolve this by projecting the FOM solution on a compatible grid of M grid points for which M/I is an integer.

As stated earlier, we use a common finite difference basis $\{\chi_{ij}\}$ of size J within each subdomain Ω_i to describe the solution:

$$u_h(x,t) = \sum_{i=1}^{I} \sum_{j=1}^{J} \mathrm{U}_{ij}(t)\chi_{ij}(x), \tag{19}$$

where χ_{ij} is only nonzero within Ω_i and $\mathbf{U}(t) \in \mathbb{R}^{I\times J}$ contains the coefficient values. Note that in this case $\mathbf{U}(t)$ is simply a reshaped version of the solution vector $\mathbf{u}(t)$. The finite difference basis functions are defined as

$$\chi_{ij}(x) = \begin{cases} 1 & \text{if } \alpha_i + (j - \frac{1}{2})h \le x < \alpha_i + (j + \frac{1}{2})h, \\ 0 & \text{elsewhere,} \end{cases} \tag{20}$$

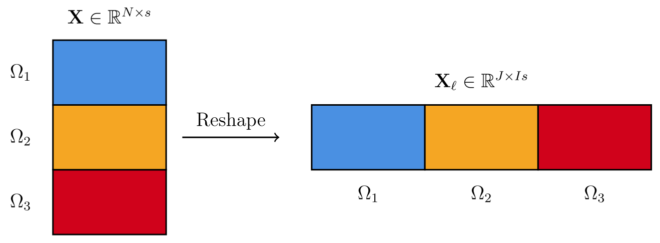
similarly to (10). An example for a Gaussian solution profile for $I = 3$ and $J = 8$ is given in Fig. 1.

After obtaining $\mathbf{U}(t)$ at different points in time, the snapshot matrix is constructed as follows

$$\mathbf{X}_\ell = \begin{bmatrix} \mathbf{U}^T(t_1) & \mathbf{U}^T(t_2) & \dots & \mathbf{U}^T(t_s) \end{bmatrix} \in \mathbb{R}^{J\times Is}. \tag{21}$$

In this way, each subdomain is treated equally in the snapshot matrix. This is what differentiates our work from [25] and similar to what is done [27]. A schematic representation of this is shown in Fig. 2, where the G-POD snapshot matrix $\mathbf{X}$ is reshaped into the L-POD snapshot matrix $\mathbf{X}_\ell$.

Using a single common snapshot matrix allows us to obtain a space-local POD basis that generalizes well outside the training data. The way we divide the domain into subdomains is decided by what results in an L-POD basis that generalizes best on a validation data set, see Section 5.2.



$\mathbf{X} \in \mathbb{R}^{N\times s}$
Ω_1
Ω_2
Ω_3
Reshape
$\mathbf{X}_\ell \in \mathbb{R}^{J\times Is}$
Ω_1
Ω_2
Ω_3

Fig. 2. Schematic representation of the G-POD snapshot matrix $\mathbf{X}$ being reshaped into the L-POD snapshot matrix $\mathbf{X}_\ell$ for $I = 3$.

Note that this matrix has fewer rows but more columns than the G-POD snapshot matrix, see (11). This makes the SVD cheaper to compute for a large number of snapshots s [31]. As in the global case, we use a SVD of the snapshot matrix to obtain a truncated basis $\hat{\mathbf{\Gamma}} \in \mathbb{R}^{J\times q}$ with $q < J$ from the left-singular vector. In terms of the local basis $\{\chi_{ij}\}$ the POD basis is written as

$$\gamma_{ij}(x) = \sum_{k=1}^{J} \chi_{ik}(x)\hat{\Gamma}_{kj}, \tag{22}$$

analogous to (18). In this basis, the approximated solution u_ℓ is obtained as

$$u_h(x,t) \approx u_\ell(x,t) = \sum_{i=1}^{I} \sum_{j=1}^{q} \mathrm{A}_{ij}(t)\gamma_{ij}(x), \tag{23}$$

similarly to the global case, see (17). The coefficients A_{ij} are obtained as

$$\mathbf{a}_\ell = \mathbf{\Gamma}^T \mathbf{u}, \tag{24}$$

where

$$\mathbf{\Gamma} = \begin{bmatrix} \hat{\mathbf{\Gamma}} & & \\ & \ddots & \\ & & \hat{\mathbf{\Gamma}} \end{bmatrix} \in \mathbb{R}^{N\times r}. \tag{25}$$

This matrix, containing non-overlapping blocks, is constructed such that multiplying by its transpose projects $\mathbf{u}$ onto the L-POD basis. For the mapping from $\mathbf{a}_\ell \in \mathbb{R}^{Iq}$ to $\mathbf{A} \in \mathbb{R}^{I\times q}$ we follow the same convention as for $\mathbf{u}$ and $\mathbf{U}$. The effective number of basis functions for L-POD is $Iq = r$ with $q < J$. This basis is orthonormal, i.e. $\mathbf{\Gamma}^T\mathbf{\Gamma} = \mathbf{I}$. The sparsity of $\mathbf{\Gamma}$, as opposed to $\mathbf{\Phi}$, is what yields a ROM which is cheaper to evaluate than a G-POD-based ROM [25].

An example of an L-POD approximation u_ℓ to a Gaussian wave is shown in Fig. 3. For the L-POD we used $I = 10$ subdomains with $q = 6$ basis functions per subdomain. The training data used to obtain the basis are discussed in the results section (Section 5), as shown in Fig. 5. Although the L-POD approximation is accurate in most of the domain, some oscillations appear near the boundaries. In addition, the approximation is not guaranteed to be smooth on the edges of the subdomains, see Fig. 3. Discontinuities appear when transitioning from one subdomain to the next. In Section 5, we will show that these discontinuities tend to grow increasingly severe when using the L-POD basis to construct a ROM, see Fig. 8. To remedy this issue, we introduce space-local POD with overlapping subdomains in the next section.

3.3. *Local POD with overlapping subdomains*

In finite element discretizations, similar discontinuities are resolved by imposing continuity of the approximation, while using a local basis. In this work, we aim to achieve the same by introducing a novel space-local POD formulation with *overlapping* subdomains. We will refer to this approach as LO-POD. To start off, we subdivide the domain Ω into I overlapping subdomains $\Omega_i = [\alpha_i, \beta_i)$. This means $\beta_{i-2} = \alpha_i < \beta_{i-1} = \alpha_{i+1} < \beta_i = \alpha_{i+2}$. Once again, each subdomain contains J grid points. Note that due to the overlap, each grid point is now located in two subdomains. This means $\frac{I\cdot J}{2} = N$, as opposed to $I \cdot J = N$ for L-POD.

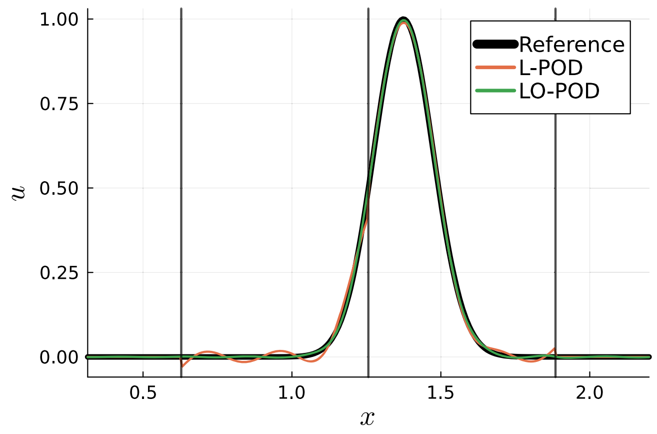


Fig. 3. L-POD and LO-POD representation of a Gaussian wave. The data used to obtain the local POD basis is explained in Section 5. The depicted snapshot is part of the snapshot matrix used to obtain the POD basis. The edges of the subdomains are indicated by the vertical grey lines. Only part of the domain $\Omega = [0, 2\pi)$ is shown.

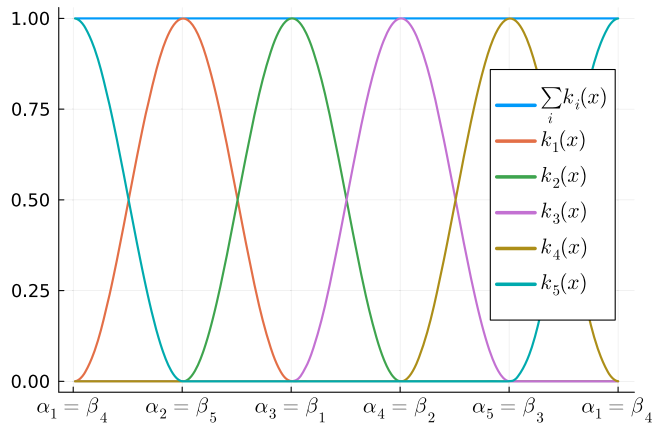


Fig. 4. Kernels k_i for a subdivision of the periodic domain into five overlapping subdomains of equal size.

To obtain a smooth approximation u_ℓ, we require the LO-POD basis to smoothly decay to zero on the edge of the subdomain. This is enforced through a post-processing step of the local snapshot matrix given by (21). For this purpose, we introduce a kernel $k_i(x)$ to divide the solution between the subdomains. This places the following constraint on the kernels:

$$\sum_{i=1}^{I} k_i(x) = 1, \tag{26}$$

such that this set of functions forms a partition of unity. Here, we propose the following kernel

$$k_i(x) = \begin{cases} \sin^2\left(\frac{1}{2}\frac{x-\alpha_i}{\alpha_i-\alpha_{i+1}}\pi\right) & \text{if } \alpha_i \le x < \alpha_{i+1}, \\ \sin^2\left(\frac{1}{2}\frac{x-\alpha_{i+1}}{\alpha_{i+2}-\alpha_{i+1}}\pi + \frac{1}{2}\pi\right) & \text{if } \alpha_{i+1} \le x < \alpha_{i+2}, \\ 0 & \text{elsewhere}, \end{cases} \tag{27}$$

which is chosen as it smoothly decays to zero at the subdomain boundaries. This approach is inspired by the partition of unity method [32]. Here, the right half of each subdomain is overlapped by the subdomain to its right, and the left half by the subdomain to its left. A visualization of the kernels is displayed in Fig. 4.

Different kernels and different amounts of overlap between subdomains can also be considered. However, we consider this outside the scope of this paper.

Using the introduced kernel, the coefficients $\mathbf{U}$ for the local expansion in (19) are obtained as

$$\mathrm{U}_{ij}(t) = \frac{(\chi_{ij}(x), k_i(x)u_h(x,t))}{(\chi_{ij}(x), \chi_{ij}(x))}. \tag{28}$$

These integrals can simply be approximated using the midpoint rule for integration [33]. The remaining procedure stays the same as for L-POD, i.e., we build the snapshot matrix in (21) and carry out a SVD to obtain $\hat{\boldsymbol{\Gamma}}$. However, what changed is that the blocks in $\boldsymbol{\Gamma}$, see (25), are now overlapping. This means the basis is no longer orthonormal, i.e. $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} \neq \mathbf{I}$. Due to the non-orthonormality, the coefficients $\mathbf{a}_\ell$ for the expansion in (23) are obtained by solving the following linear system

$$(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})\mathbf{a}_\ell = \boldsymbol{\Gamma}^T\mathbf{u}, \tag{29}$$

as opposed to (24). Obtaining the coefficients from the FOM solution is therefore more expensive. In addition, evaluating the resulting ROM requires solving a linear system, as will be discussed in Section 4. This makes the LO-POD ROM more expensive to evaluate than its non-overlapping counterpart L-POD, see Sections 4.2 and 5.5. However, looking at Fig. 3, we find that using the LO-POD results in a much smoother approximation of the Gaussian wave. Note that the parameters are kept the same, i.e., $I = 10$ and $q = 6$. In Section 5.2, the associated error will be quantified more precisely.

4. Space-local, energy-conserving reduced-order model

4.1. Projection on reduced basis

Consider again $\mathbf{c}$, the state vector of the full-order model, and $\boldsymbol{\Phi} \in \mathbb{R}^{N\times r}$ a reduced basis where $r < N$, obtained from either G-POD, L-POD, or LO-POD. For the space-local approaches, this operator was referred to as $\boldsymbol{\Gamma}$, see Section 3.2. The corresponding 'Gram' matrix $\mathbf{S} \in \mathbb{R}^{r\times r}$ of this basis is computed as

$$\mathbf{S} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}. \tag{30}$$

We can project $\mathbf{c}$ onto the subspace spanned by the POD basis as

$$\mathbf{c}_r = \underbrace{\boldsymbol{\Phi}\mathbf{S}^{-1}\boldsymbol{\Phi}^T}_{=:\mathbf{P}}\mathbf{c}. \tag{31}$$

Note that for both G-POD and L-POD, computing the inverse of $\mathbf{S}$ is trivial, as it is simply the identity. This is because the basis is orthonormal. However, for LO-POD the basis is non-orthogonal, which makes computing its inverse less trivial. As stated earlier, this is a downside to LO-POD and increases its computational cost. It is quite straightforward to see that $\mathbf{P}$ is idempotent as $\mathbf{P}^2 = \mathbf{P}$. The POD coefficient vector is obtained as

$$\mathbf{a} = \mathbf{S}^{-1}\boldsymbol{\Phi}^T\mathbf{c}. \tag{32}$$

4.2. Galerkin projection

Having obtained a reduced basis, we construct a ROM for the advection equation as follows [5–8]: based on (6) we define the residual $\mathbf{r} \in \mathbb{R}^N$ as

$$\mathbf{r}(\mathbf{u}_r) := \frac{\mathrm{d}\mathbf{u}_r}{\mathrm{d}t} + c\mathbf{D}\mathbf{u}_r \neq \mathbf{0}, \tag{33}$$

where we replaced $\mathbf{u}$ by the POD approximation $\mathbf{u}_r = \boldsymbol{\Phi}\mathbf{a}$. As $\mathbf{u}$ comes from the finite difference discretization of the advection equation, we have $\mathbf{c}(t) = \mathbf{u}(t)$. Next, we carry out the Galerkin projection of the residual on the POD basis, according to (32), and set this to zero:

$$\mathbf{S}^{-1}\boldsymbol{\Phi}^T\mathbf{r}(\mathbf{u}_r) = \mathbf{0}. \tag{34}$$

This ensures the residual is orthogonal to the basis. This results in the following ROM:

$$\frac{\mathrm{d}\mathbf{a}^{\mathrm{ROM}}}{\mathrm{d}t} := -c\mathbf{S}^{-1}\mathbf{A}\mathbf{a}^{\mathrm{ROM}}, \tag{35}$$

where the ROM operator $\mathbf{A} \in \mathbb{R}^{r\times r}$ is defined as

$$\mathbf{A} := \mathbf{\Phi}^T \mathbf{D}\mathbf{\Phi}. \tag{36}$$

Note that we introduced $\mathbf{a}^{\mathrm{ROM}}$ here as the POD state vector predicted by ROM. This is typically not equal to the true $\mathbf{a}$, see (32), past $t = 0$. Going from the FOM to the ROM, we reduced the DOF in the system from N to r. In the G-POD case $\mathbf{A}$ is typically dense, whereas in the L-POD/LO-POD it is sparse. The sparsity decreases the cost of evaluating the ROM [25]. The amount of nonzero entries in the G-POD ROM operator scales with $\mathcal{O}(r^2)$. For L-POD it scales with $\mathcal{O}(((n+1)q)^2 I)$, where n is the number of neighbors per subdomain. This accounts for the interaction between the subdomains. For LO-POD, the neighbors of the neighbors have to be included in the interactions due to the overlap. This results in a scaling of $\mathcal{O}(((\tilde{n}+1)q)^2 I)$, where $\tilde{n}$ is the number of subdomains that can be reached by crossing at most one subdomain starting from one of the subdomains. For 1D problems $n = 2$ and $\tilde{n} = 4$. For a large number of subdomains I and a small number of POD basis functions per subdomain q, with the total number of POD modes being $I \cdot q = r$, this scales more favorably than G-POD. This allows us to include a larger number of basis functions in the POD basis without sacrificing computational efficiency. Hyper-reduction techniques, including energy-conserving ones, can also be employed to further sparsen the ROM operators [17,34,35].

4.3. Energy conservation of the ROM

To ensure stability of the ROMs, we aim to mimic the energy conservation property of the FOM, see (8). To investigate if the constructed ROMs satisfy this property we define the ROM energy E_r^{ROM} as

$$E_r^{\mathrm{ROM}} = \frac{h}{2}(\mathbf{u}_r^{\mathrm{ROM}})^T(\mathbf{u}_r^{\mathrm{ROM}}), \tag{37}$$

where $\mathbf{u}_r^{\mathrm{ROM}} := \mathbf{\Phi}\mathbf{a}^{\mathrm{ROM}}$. Employing (35) we obtain the ROM evolution of $\mathbf{u}_r^{\mathrm{ROM}}$ as

$$\frac{\mathrm{d}\mathbf{u}_r^{\mathrm{ROM}}}{\mathrm{d}t} = \mathbf{\Phi}\frac{\mathrm{d}\mathbf{a}^{\mathrm{ROM}}}{\mathrm{d}t} = -c\mathbf{\Phi}\mathbf{S}^{-1}\mathbf{A}\mathbf{a}^{\mathrm{ROM}}. \tag{38}$$

The evolution of the ROM energy follows as

$$\begin{aligned}\frac{\mathrm{d}E_r^{\mathrm{ROM}}}{\mathrm{d}t} &= h(\mathbf{u}_r^{\mathrm{ROM}})^T\frac{\mathrm{d}\mathbf{u}_r^{\mathrm{ROM}}}{\mathrm{d}t} = -ch(\mathbf{u}_r^{\mathrm{ROM}})^T\mathbf{\Phi}\mathbf{S}^{-1}\mathbf{A}\mathbf{a}^{\mathrm{ROM}}\\ &= -ch(\mathbf{a}^{\mathrm{ROM}})^T\underbrace{\mathbf{\Phi}^T\mathbf{\Phi}}_{=\mathbf{S}}\mathbf{S}^{-1}\mathbf{\Phi}^T\mathbf{D}\mathbf{\Phi}\mathbf{a}^{\mathrm{ROM}}\\ &= -ch(\mathbf{a}^{\mathrm{ROM}})^T\mathbf{\Phi}^T\mathbf{D}\mathbf{\Phi}\mathbf{a}^{\mathrm{ROM}} = -ch(\mathbf{u}_r^{\mathrm{ROM}})^T\mathbf{D}\mathbf{u}_r^{\mathrm{ROM}} = 0,\end{aligned} \tag{39}$$

employing the product rule of differentiation. From the final expression, we conclude that a ROM based on Galerkin projection inherits energy conservation and stability from the FOM. This is true for both orthogonal and non-orthogonal projections. This means that not only the G-POD ROM inherits this property, as shown in [9], but also the newly introduced space-local approaches L-POD and LO-POD.

5. Results & discussion

5.1. Test case setup

To construct a ROM, we first require data from the FOM. As stated earlier, we use the finite difference discretization of the advection equation, detailed in Section 2.2, for this purpose. The system is simulated on a periodic domain $\Omega = [0, 2\pi)$ discretized with $N = 1000$ grid points for $t = [0, 5]$ and constant $c = 1$. A time step size of $\Delta t = 0.01$ is used for the time integration, using a RK4 scheme. This is the largest time step size that still yielded stable simulations. Snapshot data for the ROM construction is collected in the interval $t = [0, 1]$. We refer to this as the training data. Data generated in the interval $t = (1, 2]$ will be referred to as validation data. This data is used to evaluate the generalization of the POD basis. The remaining part of the simulation data, $t = (2, 5]$, will be used to evaluate the extrapolation capabilities of the ROMs. As an initial condition, we use a Gaussian wave centered around $x = \frac{1}{4}\pi$, namely

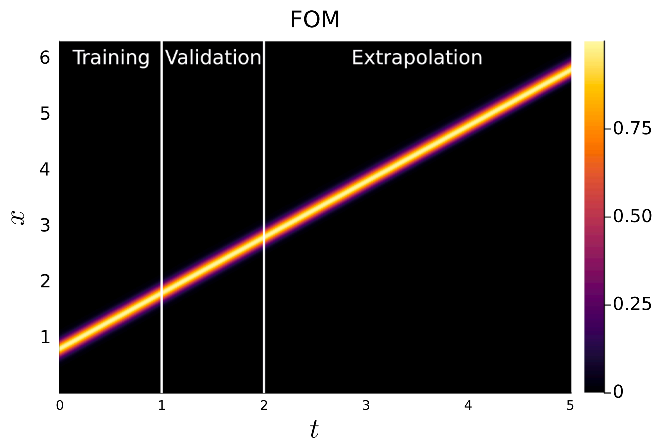


Fig. 5. Reference simulation of a Gaussian wave being advected throughout the domain. White bars separate the training data from the validation data and the validation data from the extrapolation data, respectively.

$$u(x, 0) = \exp(-50(x - \frac{1}{4}\pi)^2). \tag{40}$$

A visualization of this simulation is displayed in Fig. 5.

For the ROMs we consider the three bases discussed in this work: the global POD basis G-POD, the space-local POD basis L-POD, and the space-local POD basis with overlapping subdomains LO-POD. For the space-local approaches, the domain is subdivided into I subdomains. For the local basis $\{\chi_{ij}\}$, we use J box functions contained within each subdomain, such that $IJ = N$. In this way, the local basis aligns with the FOM finite difference basis, see (10). The integrals in the LO-POD post-processing step, Eq. (28), are approximated using the midpoint rule for integration [33]. For the ROM time integration, we use the same RK4 scheme as for the FOM.

To evaluate the ROMs we evaluate the difference between the ROM solution $\mathbf{u}_r^{\mathrm{ROM}} := \mathbf{\Phi}\mathbf{a}^{\mathrm{ROM}}$ and the FOM solution $\mathbf{u}^{\mathrm{FOM}}$:

$$\underbrace{\frac{\mathbf{u}_r^{\mathrm{ROM}}(t) - \mathbf{u}^{\mathrm{FOM}}(t)}{||\mathbf{u}^{\mathrm{FOM}}(t)||_2}}_{:=\text{solution error}} = \underbrace{\frac{\mathbf{u}_r^{\mathrm{ROM}}(t) - \mathbf{P}\mathbf{u}^{\mathrm{FOM}}(t)}{||\mathbf{u}^{\mathrm{FOM}}(t)||_2}}_{:=\text{ROM error}} + \underbrace{\frac{\mathbf{P}\mathbf{u}^{\mathrm{FOM}}(t) - \mathbf{u}^{\mathrm{FOM}}(t)}{||\mathbf{u}^{\mathrm{FOM}}(t)||_2}}_{:=\text{projection error}}, \tag{41}$$

where $\mathbf{P}$ projects the solution on the POD basis, see (31). This difference will be referred to as the solution error. In (41), this error is decomposed as the sum of the ROM error (the error made by the ROM during the time integration) and the projection error (the error made by the reduced basis approximation). Our implementation, in Julia [36], of the introduced methodologies and experiments is freely available on Github, see https://github.com/tobyvg/local_POD_overlap.jl.

5.2. Projection error

For the construction of the L-POD and LO-POD ROMs, we have to determine the number of subdomains I and the number of modes per subdomain q, with $Iq = r$, which results in a basis that generalizes best outside the training data. To do this, we evaluate the projection error, see (41), on both the training and validation data. The results are depicted in Fig. 6 for different I, averaged over both the training (solid line) and validation data (dashed line).

A basis that generalizes well should perform well on both the training and validation data. We observe that for $I = 5$, the training error is lowest, but the validation error is rather large, indicating that the basis does not generalize well. For higher values of I, the training and validation errors are much closer, but tend to increase with increasing I. In this case, $I = 10$ is considered optimal, as the training and validation

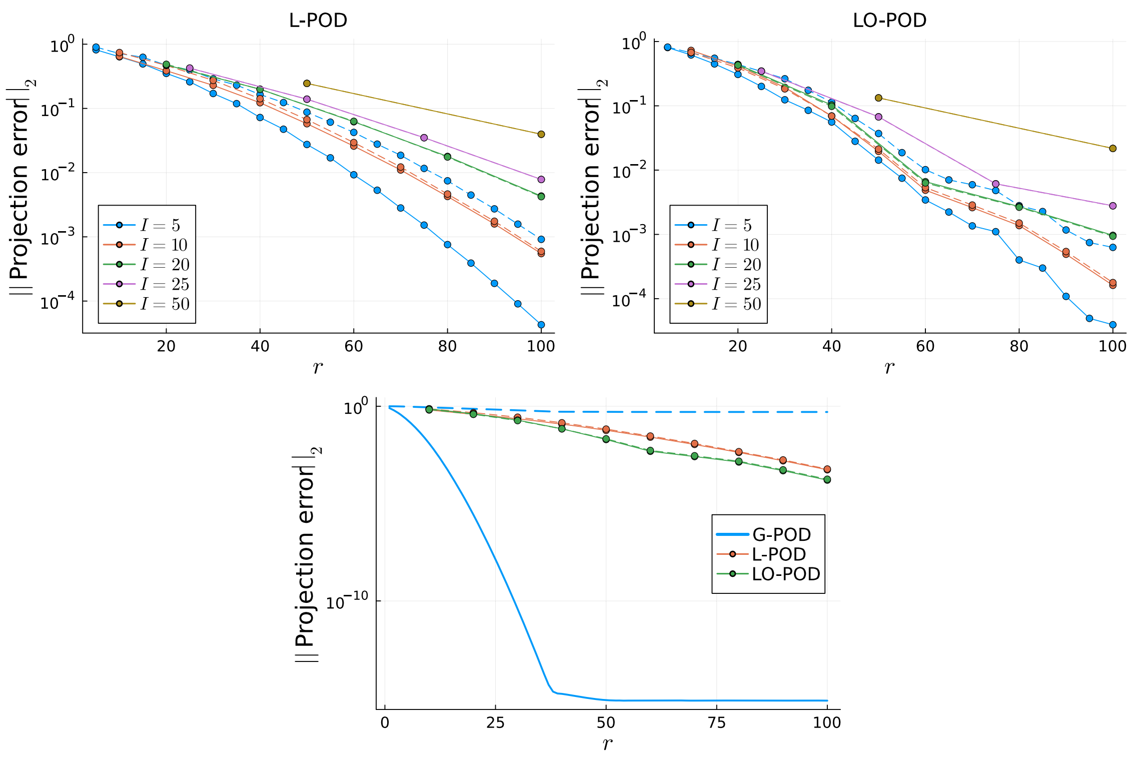


Fig. 6. (Top) Projection error evaluated over the training (solid line) and validation (dashed line) data for different I for the space-local approaches. (Bottom) Projection error for $I = 10$ for L-POD and LO-POD evaluated over the training (solid line) and validation (dashed line) data. The projection error for G-POD is also depicted.

errors are small and of similar size. This is true for both L-POD and LO-POD. For the remainder of this text, we will therefore stick to $I = 10$ for the construction of the local ROMs. In general, the choice of I is likely problem-dependent, and an a priori study similar to Fig. 6 needs to be performed to determine an optimal value.

Next, we compare the performance of the space-local approaches against each other, as well as to G-POD. This is also displayed in Fig. 6 (bottom plot). We observe that the projection error converges faster for LO-POD, as compared to L-POD. This can be explained by the fact that the LO-POD basis functions are constructed from twice as many finite difference points as for L-POD, due to the overlapping subdomains. This means the fit is carried out over twice as many DOF, which yields a lower projection error. The difference between convergence on the training and validation data is slight for both space-local approaches. On the other hand, for G-POD, the convergence of the projection error on the training data is very fast, see (13). However, on the validation data, the error hardly converges. This is because the G-POD is only suited for representing the wave on the left side of the domain, as detailed in the next section. We conclude that the space-local approaches yield a basis that generalizes better than the global approach for this particular problem.

5.3. Resulting basis

The resulting local basis functions for $q = 6$ are displayed in Fig. 7 along with the first six G-POD modes.

We find that the G-POD modes are only nonzero where the traveling wave is represented in the training data. This explains why the error on the validation data hardly converged in Fig. 6. For the space-local approaches, where a common basis is “copied/pasted” across the entire domain, this is not an issue. Regarding L-POD, we observe that the obtained basis functions do not smoothly decay to zero at the edge of the subdomain, but instead end abruptly. Finally, we observe that for LO-POD the post-processing procedure in (28) indeed results in a basis that smoothly decays on the edge of the subdomains.

5.4. Accuracy and energy conservation of ROM

Having obtained a basis for each of the POD approaches we construct a set of ROMs. Based on the results in Section 5.2, we select $r = 60$, $I = 10$, and $q = 6$ for the space-local approaches. To keep the size of the basis the same, we choose $r = 60$ for G-POD. The ROM results are obtained by evaluating (35) from the initial condition in (40). The resulting simulations up to $t = 5$ are presented in Fig. 8.

For G-POD we find that after the training data the performance degrades significantly. This exposes the limitations of G-POD, as it is unable to adapt to wave traveling outside the training range. The space-local approaches perform better in this regard. Both L-POD and LO-POD are capable of extrapolating the traveling of the wave past the training region. However, L-POD seems to suffer from discontinuities in $\mathbf{u}_r^{\text{ROM}}$, as the edges of the subdomains become increasingly visible as the simulation progresses. LO-POD does not suffer from this issue and smoothly extrapolates the solution past the training region. This means the smooth reconstruction coming from LO-POD indeed improves the quality of the simulation for the same number of DOF.

Next, we evaluate the ROMs on a set of performance metrics. The first metric is the solution error, see (41). The other metrics (defined on the figure axes) focus on how well the energy is conserved during the simulation, namely the total change in energy and the instantaneous change in energy. The results are shown in Fig. 9.

The results are depicted for both the FOM time step size Δt (solid line) and a five times larger time step $5 \cdot \Delta t$ (dashed line). This larger time step is based on an eigenvalue analysis of the ROM operator, also presented in Fig. 9. For this analysis, we determined the eigenvalues of the operator $\mathbf{D}$ projected on the ROM basis. This operator is given by $\mathbf{PDP}$. It can be shown that replacing $\mathbf{D}$ in the FOM, see (6), by this

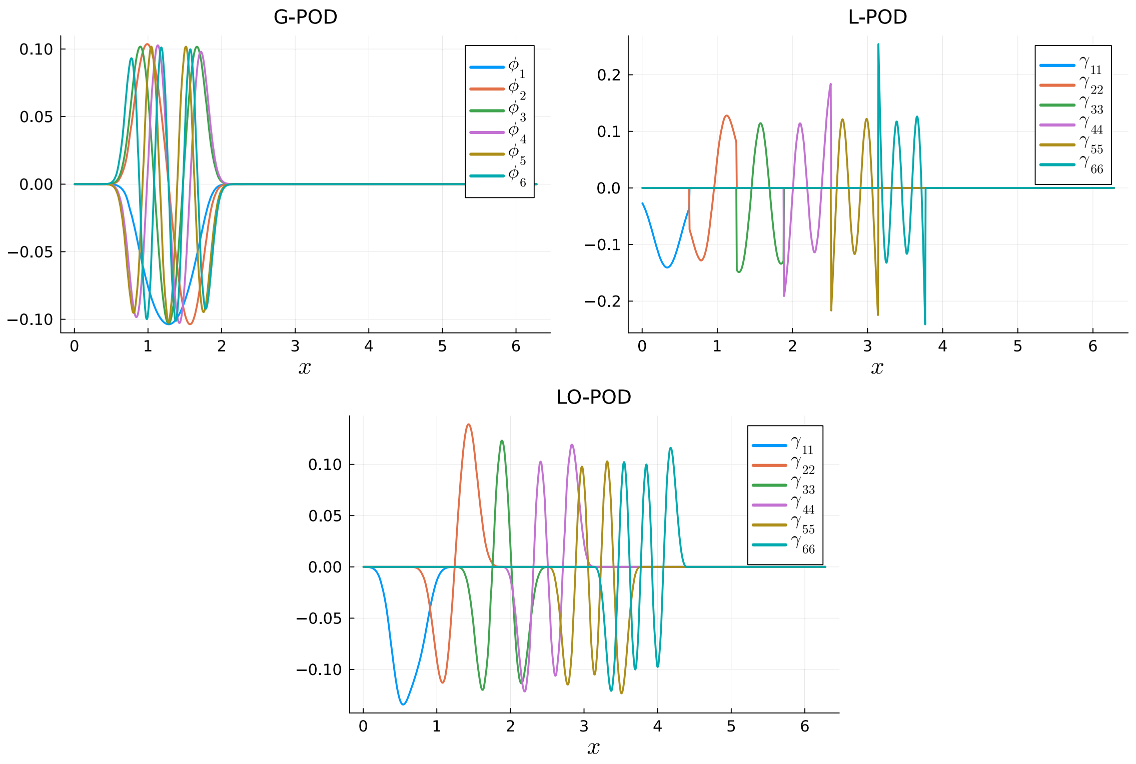


Fig. 7. The first six G-POD modes and the first six space-local POD modes for L-POD and LO-POD. For visualization purposes, the space-local POD modes are displayed on adjacent subdomains.

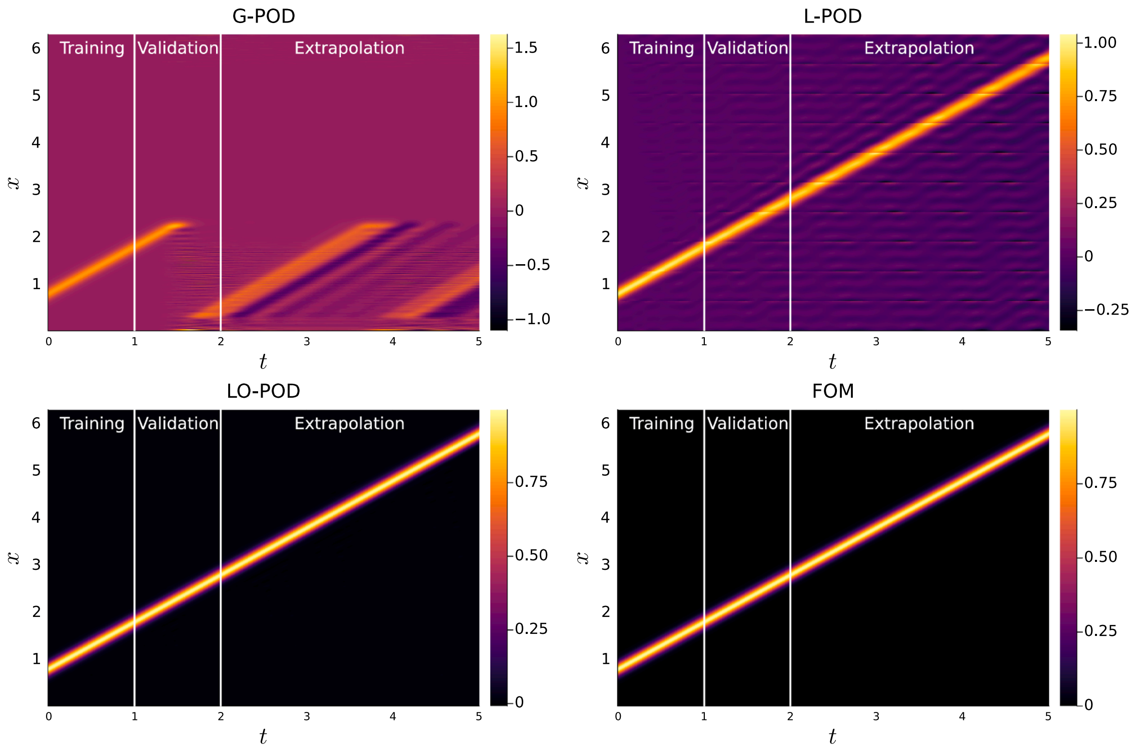


Fig. 8. Trajectories of $\mathbf{u}_r$ for the different ROMs, along with the FOM trajectory. From left to right, the white lines indicate the end of the training data and validation data, respectively.

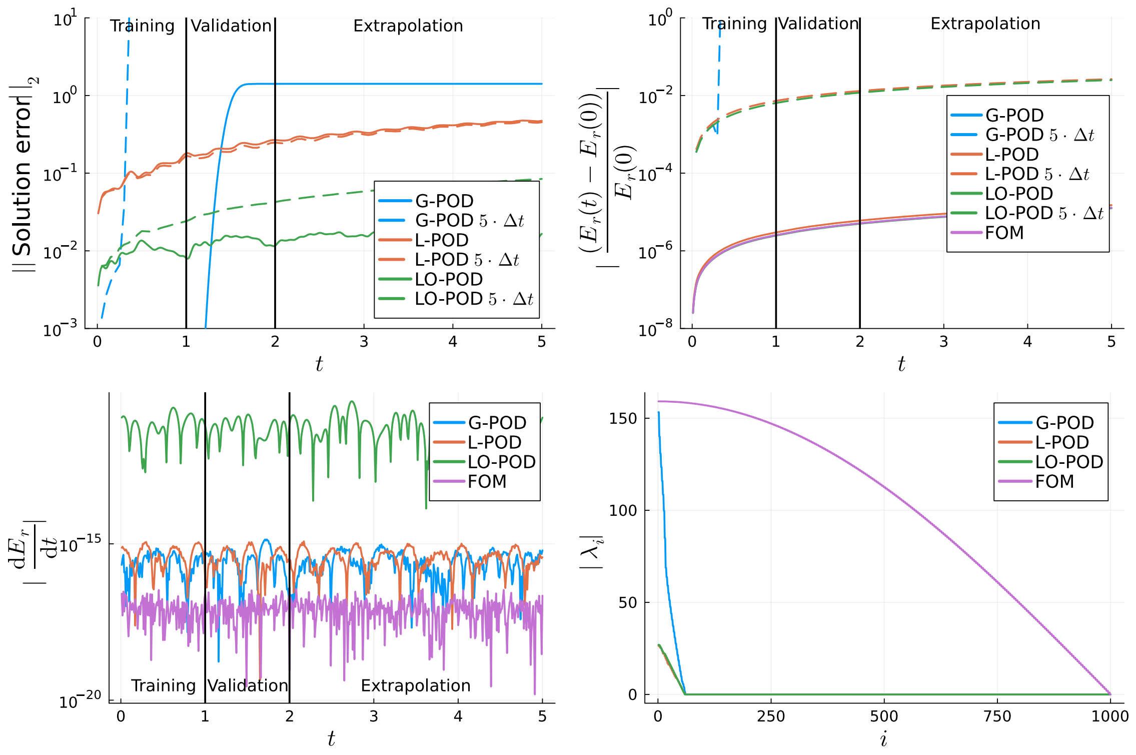


Fig. 9. (Top-left) Solution error for each of ROMs during the simulation, presented for both $\Delta t = 0.01$ and $\Delta t = 0.05$. From left to right, the black lines indicate the end of the training data and validation data, respectively. (Top-right) Relative change in energy with respect to the start of the simulation. (Bottom-left) Instantaneous change in energy, computed by evaluating (39). (Bottom-right) Eigenvalues of the projected FOM operator.

projected operator is equivalent to integrating the ROM, see (35). The eigenvalues λ_i are ordered according to the magnitude of their absolute values. We observe that the largest eigenvalue for the space-local approaches is roughly five times smaller than that of G-POD and the FOM. This is likely due to the fact that the G-POD basis functions contain higher frequencies than the space-local basis functions, see Fig. 7. Using a smaller G-POD basis could alleviate this. Based on this analysis, we also evaluate the performance of the ROMs for a five times larger time step [37].

Looking at the solution error, we observe that G-POD performs best in the training region (the error is so small that it is outside the plotting range). However, after leaving the training region ($t > 1$), the performance degrades rapidly. On the other hand, for the space-local approaches, we do not observe this jump of error outside the training region, but rather a steady increase. Importantly, outside the training region, the space-local approaches are much more accurate than G-POD. In particular, LO-POD improves by more than one order of magnitude upon both G-POD and L-POD outside the training region.

When increasing the time step size by fivefold, the simulation quickly becomes unstable for G-POD. For L-POD, both time steps yield stable simulations, giving results that are very close to each other. For LO-POD and a smaller time step size, the error seems to converge to an equilibrium. However, for a larger time step, it increases steadily. This means there is likely still a benefit to taking a smaller time step. An interesting continuation of this research would be to find a way to systematically determine the "sweet spot" between increasing the time step and maintaining accuracy of the ROM. This could for example, be done by considering several different time step sizes and both implicit and explicit integration methods, or adaptive time-stepping based on an error estimator [38].

In terms of energy conservation, we observe that all ROMs conserve the energy as predicted by the theoretical analysis, except for a time discretization error incurred by the use of RK4. This error increases when the time step size is increased. Only for G-POD with an increased time step size, the simulation becomes unstable. For LO-POD, the numerical error in the instantaneous change in energy, see (39), is the largest. This is likely due to the linear system that needs to be solved to evaluate the ROM for a non-orthogonal basis, see (35). However, the time discretization error is the primary source of error, as the change in energy during the simulation is roughly the same for L-POD and LO-POD.

5.5. Convergence with increasing ROM dimension

Next, we look at the convergence of the solution error as we increase the size of ROM basis r. In addition, we compare the ROMs to a finite difference discretization with the same number of DOF. Note that for the ROMs DOF $= r$. For the finite difference simulations, we first project the solution on the FOM grid using linear interpolation. We then compute the difference, such as in (41), and compute the L_2-norm to quantify the error. For the local ROMs we stick to $I = 10$ subdomains, while increasing the number of basis functions per subdomain q to increase r. We consider a maximum of $r = 100$ for the ROMs.

The results are depicted in Fig. 10.

Regarding the local ROMs, we observe rapid convergence of the solution error as we increase the number of DOF, with LO-POD consistently outperforming L-POD. This is in line with the projection error convergence discussed in Section 5.2. Regarding G-POD, we observe no convergence of the solution error. Regarding the finite difference discretization, we find it converges at a much slower rate than the space-local ROMs.

To obtain a conclusive answer about the scaling of the ROMs, we considered the number of nonzero entries in the ROM operator $\mathbf{A}$. This is also depicted in Fig. 10. Here we find that the number of nonzero entries scales according to a power law, with L-POD scaling the most favorably. This is in line with the discussion presented in Section 4.2. Hyper-reduction can be carried out to further reduce the computational cost of the ROMs [17,34,35]. Finally, one could also increase the time

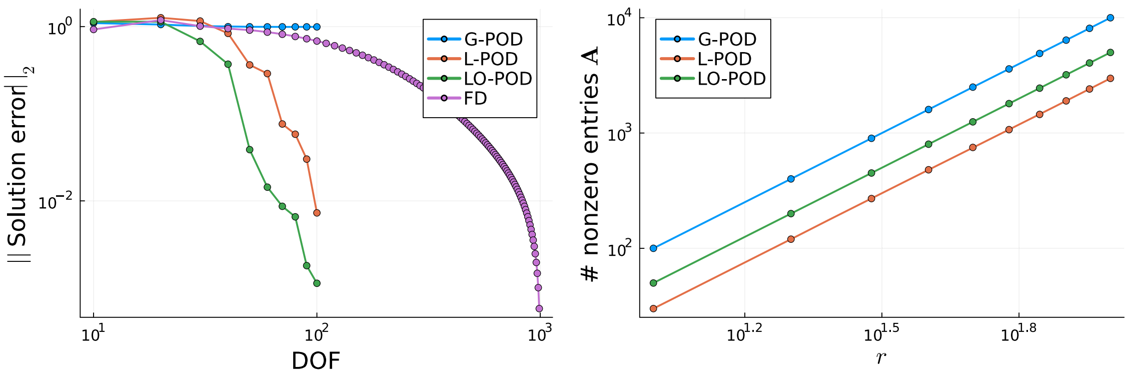


Fig. 10. (Top-left) Solution error averaged over a simulation up to $t = 5$. Results are presented for the different ROMs, as well as a finite difference discretization (FD) with the same number of DOF. (Bottom-right) Number of nonzero entries in the ROM operator **A**.

step size, as discussed in Section 5.4, to further decrease the computational cost of the space-local ROMs.

5.6. 2D advection-diffusion equation

To further evaluate the viability of the space-local ROMs, we consider the 2D advection-diffusion equation:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = \underbrace{-\nabla\cdot(\mathbf{V}(\mathbf{x})u(\mathbf{x},t))}_{\text{advection}} + \underbrace{\nu\nabla^2 u(\mathbf{x},t)}_{\text{diffusion}}, \tag{42}$$

where some quantity $u(\mathbf{x},t)\in\mathbb{R}$ is advected through space $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2$ by a stationary velocity field $\mathbf{V}\in\mathbb{R}^2$ which is divergence free, i.e. $\nabla\cdot\mathbf{V} = 0$. The fact that the velocity field is now space-dependent, as opposed to the 1D case where it was uniform, adds to the difficulty of the test case. As opposed to the 1D case this system also contains diffusion. This results in a passive spread of u throughout the domain. The diffusion rate is determined by the scalar viscosity $\nu \geq 0$. In our case, we employ a finite volume discretization with a staggered grid for the velocity field [39]. This is done to preserve the skew-symmetry of the operator. The diffusive term is discretized using a simple second-order scheme.

For the FOM we consider the following setting: A periodic domain $\Omega = [-\pi,\pi]\times[-\pi,\pi]$, a velocity field given by $\mathbf{V} = \begin{bmatrix}\cos(x-y) & \cos(x-y)\end{bmatrix}^T$ discretized on a 256×256 uniform grid, and a viscosity of $\nu = 10^{-3}$. The initial condition is given by the sum of three Gaussians, each centered on one of the streamlines of the flow where the velocity is highest:

$$\begin{aligned} u(\mathbf{x},0) = &-\exp(-2x^2-2y^2) + \exp(-2\left(x-\frac{\pi}{2}\right)^2 - 2\left(y+\frac{\pi}{2}\right)^2) \\ &+ \exp(-2\left(x+\frac{\pi}{2}\right)^2 - 2\left(y-\frac{\pi}{2}\right)^2). \end{aligned} \tag{43}$$

The FOM is integrated in time using a RK4 scheme with $\Delta t = 0.025$. Every 16 time steps, we save a snapshot of $u(x,t)$ for the construction of the ROM basis. Note that the performance of the ROMs converge to the FOM, when doing the Galerkin projection. To resolve this closure, models can be included in the ROM [40]. The snapshots collected in the interval $t = [0,12]$ are used as training data, the interval $t = (16,20]$ is used as validation data, and the interval $t = (20,40]$ is used to test the extrapolation capabilities of the ROMs. The ROMs are also integrated in time explicitly using a RK4 scheme. However, for the LO-POD ROM we resort to the implicit Crank-Nicolson method, see [24], to limit the computational cost. This scheme enables us to achieve second-order accuracy in time, while solving only a single linear system at each time step, rather than at each evaluation of (35) in the RK4 scheme. To solve the resulting system efficiently, we employ an LU-decomposition [41]. This is required, as the LO-POD basis is non-orthogonal, i.e. $\mathbf{S}\neq\mathbf{I}$.

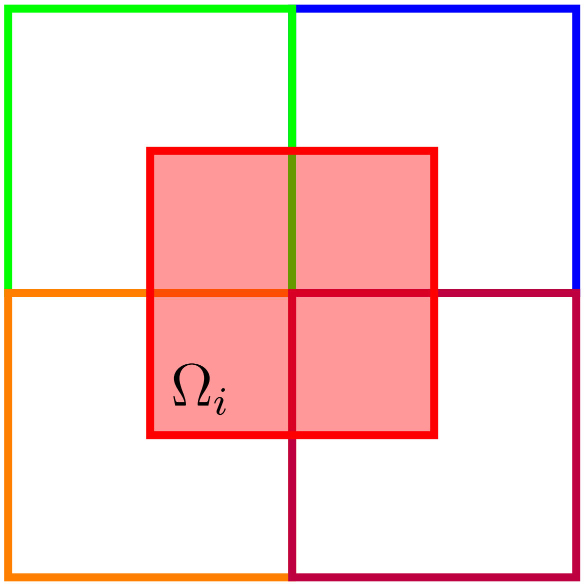


Fig. 11. Subdomain Ω_i and its overlapping subdomains for LO-POD in 2D.

Regarding the subdomain decomposition, we use the fact that the FOM is discretized on a uniform grid. In this way, we can simply subdivide the domain uniformly into subdomains. For L-POD, this results in a straightforward decomposition. For LO-POD the decomposition is chosen such that each subdomain overlaps with its four neighboring subdomains which share a vertex in the middle of the considered domain. This is depicted in Fig. 11.

The number of subdomains I for the space-local approaches and the number of modes q is chosen according to the performance on the validation set. This is presented in Appendix B. Based on this, we settle on $I = 8\times8$ and $q = 20$ for L-POD and $q = 15$ for LO-POD. We settle on these values for q as these are the lowest values which surpass a projection error threshold of 10^{-2} on the validation set for this value of I. The value of I is chosen as it generalizes well from the training data set to the validation data set. For G-POD, we choose $r = q = 31$, as the snapshot matrix contains 31 snapshots for G-POD such that the SVD only produces 31 basis functions. For the size of the time step, we choose the largest value that does not degrade the ROM performance. This is presented in Appendix C. The time step sizes used are presented in Table 1.

5.6.1. 2D basis

The resulting basis computed with the SVD for each of the POD approaches is depicted in Fig. 12.

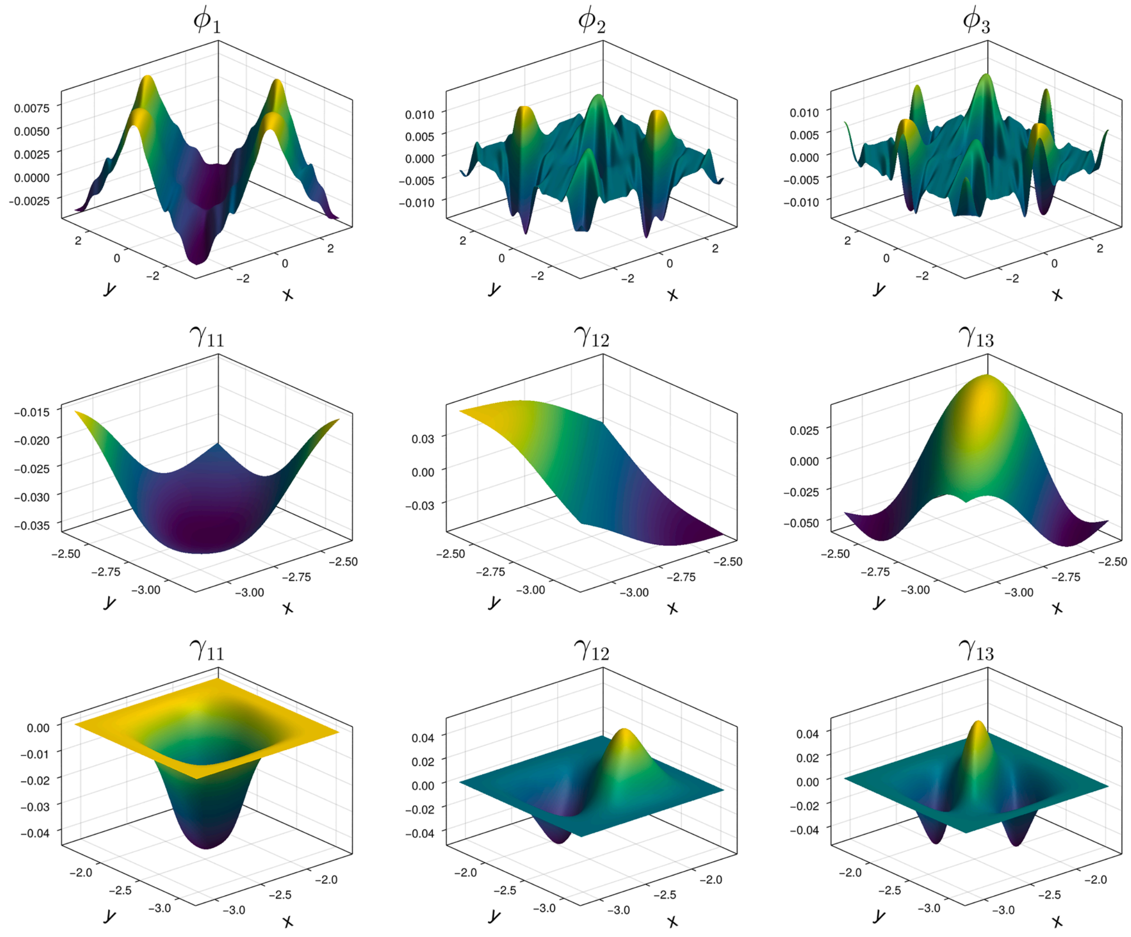


Fig. 12. The first three G-POD (top) modes and the first three space-local POD modes for L-POD (middle) and LO-POD (bottom) for the 2D advection test case.

From the basis functions, we can see that the flow runs diagonally across the domain. This is expected from the prescribed velocity field for which $V_1 = V_2$. The G-POD modes contain information on the most dominant features of the flow. From these modes, we can see that some interesting dynamics occur in the region where the flow changes direction. The second and the third mode also seem to be quite similar, but slightly shifted. This is common for advection-dominated flows and results in a slow singular value decay [5]. For the space-local approaches, we find that the basis functions contain less information about the flow, but seem more similar to a generic basis. This likely allows them to generalize better on the validation set. We provide more insight on this in Appendix B. For LO-POD we find that the basis functions smoothly decay to zero at the edge of the subdomain, as enforced by the kernel introduced in Appendix A. This allows LO-POD to produce smooth approximations. The basis resulting from L-POD does not guarantee this.

5.6.2. ROM performance

In this section, we discuss the performance of different ROMs and compare to both the FOM and a finite volume discretization, which is twice as coarse as the FOM in each direction (128×128 as compared to 256×256 for the FOM). This will be referred to as coarse FOM. The results are summarized in Table 1 for a full simulation up to $t = 40$ (including the training, validation, and extrapolation region).

For the G-POD ROM we find a short computation time, due to the limited number of modes, but also a large solution and gradient error. The latter is defined as

$$\text{gradient error} := \frac{\mathbf{G}\mathbf{u}_r^{\text{ROM}}(t) - \mathbf{G}\mathbf{u}^{\text{FOM}}(t)}{||\mathbf{G}\mathbf{u}^{\text{FOM}}(t)||_2}, \tag{44}$$

Table 1
Computation times (**comp time**) in seconds on a standard laptop CPU and average solution errors (**sol err**) computed for the entire simulation $t \in [0, 40]$ for each of the ROMs for the 2D advection test case. A coarse FOM with a resolution of 128×128 is also included for comparison. For the space-local approaches, the number of DOF is reported as the multiplication $I \times q = r$. Reported computation times are an average of 5 replica simulations. The average error for the gradient of the solution (**grad err**) is also depicted.

	DOF ($I \times q$)	comp time	$\|\|\text{sol err}\|\|_2$	$\|\|\text{grad err}\|\|_2$	Δt
G-POD	**31**	$\mathbf{2.5 \times 10^{-3}}$	0.267	0.636	**0.2**
L-POD	$8^2 \times 20 = 1280$	0.207	**0.0353**	0.213	**0.2**
LO-POD	$8^2 \times 15 = 960$	1.02	0.0354	**0.164**	0.05
coarse FOM	$128^2 = 16384$	4.57	0.226	0.514	0.1
FOM	$256^2 = 65536$	25.9	–	–	0.025

where $\mathbf{G} \in \mathbb{R}^{2N \times N}$ is a forward difference approximation of the gradient operator, such that the elements of $\mathbf{G}\mathbf{u}_r^{\text{ROM}}$ approximate ∇u in different parts of the domain. These large errors are likely the result of the G-POD basis's limited generalization capabilities. For the space-local approaches, we observe that the computation time is roughly two orders of magnitude larger than for G-POD. As we have access to more space-local POD modes than global ones, we choose to include enough to accurately represent the solution, see Appendix B for details. This increases the computation time with respect to G-POD, see Table 1 for the exact number of DOF for each ROM. However, using the space-local approaches, we still obtain a speedup between one and two orders of magnitude with respect to the FOM. The L-POD ROM is roughly 5 times faster than the LO-POD ROM. This is mostly because the explicit RK4 scheme (fourth order accurate) used for L-POD allows for larger

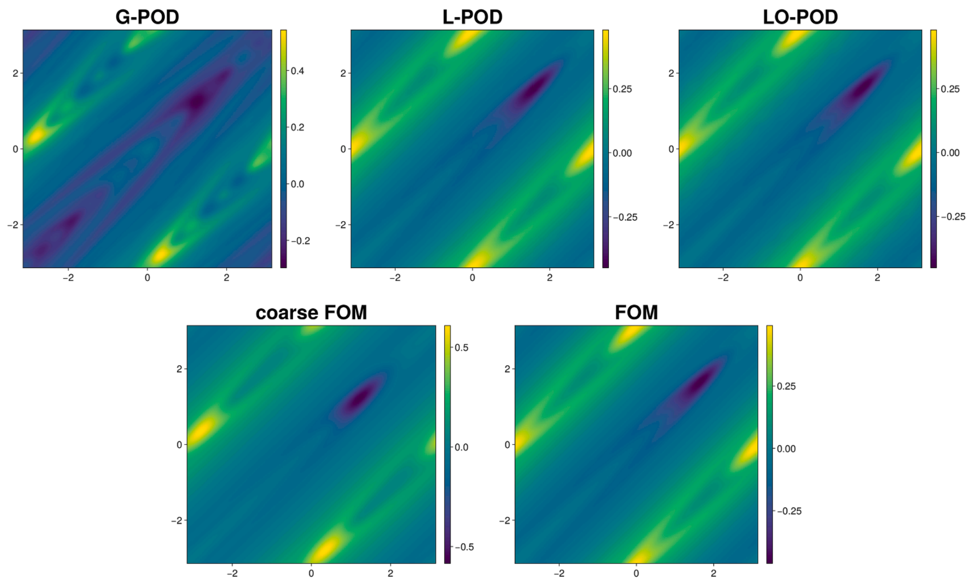


Fig. 13. Solution at the end of the simulation at $t = 40$ for the 2D advection test case produced by the different ROMs as well as a coarse FOM.

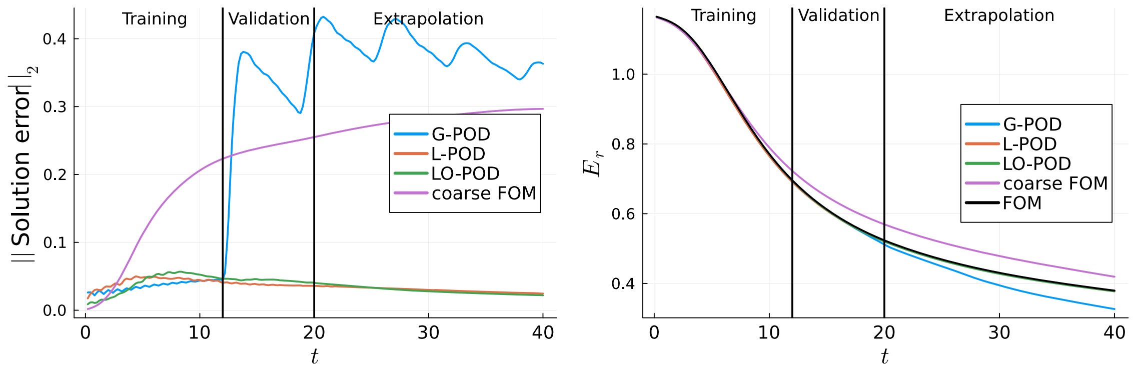


Fig. 14. (Left) Solution error for each of the ROMs during the simulation of the 2D advection test case. (Right) Energy trajectories during the simulation. Energy trajectories from L-POD and LO-POD overlap with the FOM trajectory. From left to right, the black lines indicate the end of the training data and validation data, respectively.

time steps without loss of accuracy, as compared to the Crank-Nicolson scheme (only second order accurate) used for LO-POD, see Appendix C for an analysis on the time step size. As mentioned earlier, the Crank-Nicolson scheme is used for LO-POD as it limits the number of linear systems needed to be solved per time step to one. L-POD and LO-POD both produce a similar solution error. However, looking at the gradient error LO-POD obtains a lower error. This is likely because LO-POD does not suffer from discontinuities at the subdomain boundaries, while L-POD does, see Fig. 3. Both space-local approaches outperform the coarse grid finite volume discretization, both in computation time and error metrics.

Next, we consider the solution at the end of the simulation ($t = 40$), see Fig. 13.

Here we find that G-POD does not capture the solution well and contains unphysical artifacts, whereas the space-local approaches do reproduce the solution well. The coarse-grid finite volume discretization also captures the solution quite well, but is slightly smoothed and shifted (due to numerical diffusion and dispersion).

Finally, we consider the solution error and energy trajectory for each of these models, see Fig. 14.

We find that the G-POD ROM performs well within the training region, but it does not extrapolate well. For the space-local approaches, this is not an issue as both perform comparably, both inside and outside the training region, and for both the solution error and the energy. These approaches also outperform the coarse FOM, which slowly accumulates error during the simulation.

5.7. Applicability to 2D Navier-Stokes

As a final test case, we evaluate the applicability to the 2D incompressible Navier-Stokes equations by assessing how well the POD bases generalize to representing the solution. The incompressible Navier-Stokes equations read:

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{V} + \mathbf{f}, \tag{45a}$$

$$\nabla \cdot \mathbf{V} = 0, \tag{45b}$$

for a 2D velocity field $\mathbf{V}(\mathbf{x}, t) \in \mathbb{R}^2$. For the density we choose $\rho = 1$ and for the kinematic viscosity $\nu = \frac{1}{1000}$. For the forcing we take $\mathbf{f}(\mathbf{V}, \mathbf{x}, t) = \begin{bmatrix} \sin(4y) & 0 \end{bmatrix}^T - 0.1\mathbf{V}$, as to simulate Kolmogorov flow. This setup is often used for evaluating data-driven turbulence modeling strategies [42–44]. The PDE is solved numerically using the second-order accurate scheme presented in [45] on a periodic domain $\Omega = [-\pi, \pi] \times [-\pi, \pi]$. The computational grid consists of 2048×2048 grid cells. The simulation is initialized with energy content in the large wave-numbers, and a warm-up period of 25 time units is employed before collecting the training data. After the warm-up period, training data is collected on the interval $t = [0, 10]$ and validation data on the interval $t = [100, 110]$,

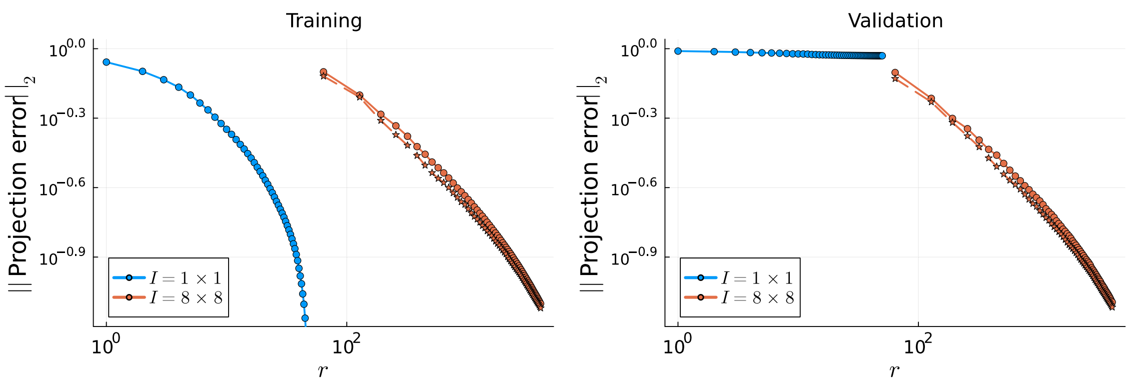


Fig. 15. Projection error evaluated over the training (left) and validation (right) data set for the 2D Navier-Stokes test case. $I = 1 \times 1$ corresponds to G-POD. Solid dots correspond to L-POD, stars to LO-POD.

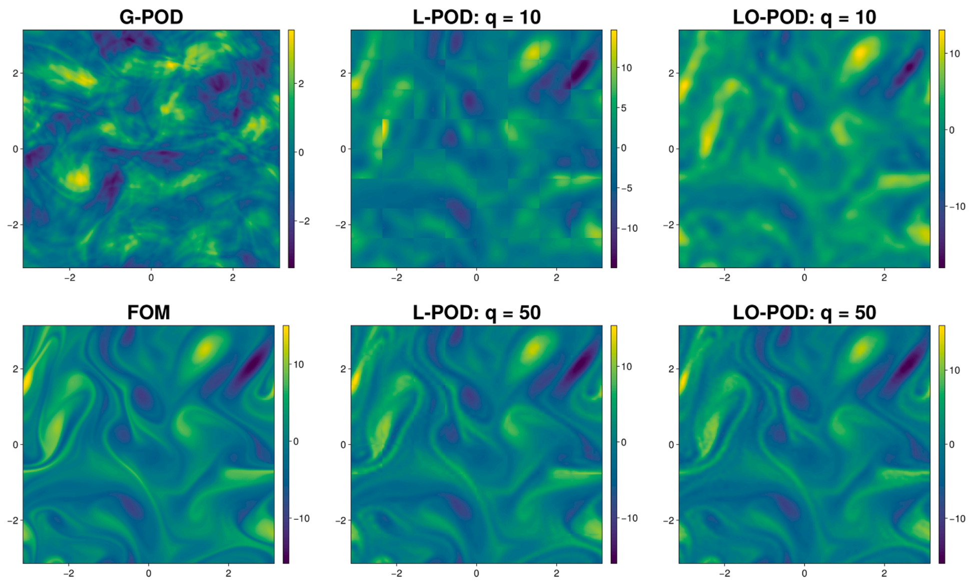


Fig. 16. First vorticity snapshot of the validation data projected on the different POD bases with varying number of modes per subdomain q for the space-local approaches.

to remove any temporal correlation. Both the validation and training data consist of 50 snapshots of the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{V} \in \mathbb{R}^3$. For 2D, only the third component of this vector is nonzero, which we shall refer to simply as ω. We choose to represent the vorticity as for periodic domains in 2D, all the information of the flow is contained within this scalar field [46].

To begin our analysis, we examine the projection error for both the training and validation datasets using an 8×8 decomposition of subdomains, which is identical to the 2D advection-diffusion case, see Fig. 15.

Similar to the previous test cases, we find that the projection error of the G-POD converges fast on the training set. However, upon examining the validation set, we find that the space-local approaches generalize significantly better.

In Fig. 16, we display the reconstruction of the first snapshot of the validation set produced by the different bases.

We depict the reconstructions for both $q = 10$ and $q = 50$ modes per subdomain for the space-local approaches, and all the available POD modes, i.e. $r = 50$, for G-POD. We find that the space-local approaches capture the flow characteristics quite well, with $8 \times 8 \times 50 = 3200$ DOF, whereas G-POD does not capture any of the characteristics. Regarding L-POD, we find that using a lower number of modes per subdomain q worsens the discontinuities at the subdomain boundaries, whereas LO-POD suffers from noisier reconstructions for low q.

6. Conclusions

In this work, we presented a novel way of constructing ROMs for PDEs describing advection-dominated problems. The key properties of our proposed ROM are: *sparsity* and *generalizability* through a space-local ROM with overlapping subdomains; *stability* by embedding energy conservation in the ROM. The space-local ROMs are achieved by building a POD basis within each subdomain, as compared to a single common basis for the standard space-global approach. In [25], it was shown that such a basis results in a sparse and computationally efficient ROMs. As suggested in [27], we modified this approach to generate a common local basis for the entire domain, similar to a finite element basis. This work differs from [27] in that we applied this to time-dependent PDEs, as opposed to steady-state problems. In addition, we introduced overlapping subdomains to ensure a smooth representation of the solution within the space-local POD basis. By generating a common basis, we achieve generalizability in time, as in this way, a feature observed in one subdomain can now be represented in any of the subdomains. In addition, we introduced overlapping subdomains to prevent discontinuities at the subdomain boundaries. To test our methodologies, we made use of the linear advection equation in both 1D and 2D. One of the properties of this system is that the energy is conserved. Our space-local ROMs satisfy this property exactly, even when the basis is non-orthogonal (as is the case for the overlapping approach).

We observed that the resulting space-local ROMs generalize much better for advection-dominated problems than the standard space-global approach. We also demonstrated that such a space-local approach yields sparser and, consequently, more computationally efficient ROMs. In addition, we showed that the space-local ROMs also satisfy energy conservation and allow for larger time steps. The latter further decreases the computational cost of the ROMs. Regarding computational time, we observed an improvement of one to two orders of magnitude with respect to the FOM in the 2D test case. By introducing overlapping subdomains, we demonstrated that we obtain smoother approximations of the solution and require fewer basis functions to represent it compared to the non-overlapping approach. However, this comes at the cost of solving a linear system at each time step. Depending on the application, either overlapping or non-overlapping subdomains might be most suitable. If smoothness of the solution is required, overlapping subdomains might be more suitable, while if computational efficiency is of higher priority, non-overlapping subdomains are likely preferred.

To further evaluate the generalizability of the space-local POD approaches, we applied it to the 2D incompressible Navier-Stokes equations in a Kolmogorov flow setup. In this case we solely considered the ability of POD basis to represent the solution, without building the ROMs. We consider constructing the actual ROM for this non-linear 2D test case outside the scope of this research, as it requires dealing with the divergence freeness condition, and possibly introducing hyperreduction methods for the non-linear term [34,35]. Regarding the representation of the solution space-local methods (L-POD and LO-POD) clearly outperformed G-POD, which showed strong overfitting to the training data. In contrast, the local variants maintained low projection errors and accurately reproduced flow features on unseen, temporally decorrelated data. These results indicate that the space-local framework generalizes well for advection-dominated systems and holds strong potential for reduced-order modeling of turbulent flows.

For future research, we consider constructing a space-local ROM for an actual turbulence test case described by the Navier-Stokes equations. To achieve energy conservation, one can make use of the observation presented in [9], where it was shown that carrying out a Galerkin projection on an energy-conserving FOM (with quadratic non-linearity) resulted in an energy-conserving ROM. However, one important requirement is that the POD basis needs to be divergence-free, meaning that the space-local POD basis also needs to be divergence-free. Currently, this is still being actively researched in our group. To resolve this, we could take inspiration from [27] and construct a POD basis, not only for the velocity, but also for the pressure [47]. Another possible research direction would be to circumvent or speed up the solution of the linear system required in LO-POD. For this purpose, one could possibly take inspiration from the finite element community [24]. Finally, the application of the space-local POD methods to steady-state problems could be investigated. The work in [27] offers an interesting starting point for this, as they investigated the generalization of the space-local POD basis when increasing the size of the domain for steady-state problems. An interesting research avenue would be to compare the space-local ROMs to space-global ones, when the spatial domain is kept the same and one of the model parameters is varied.

Acronyms

BC	boundary condition.
DOF	degrees of freedom.
FOM	full-order model.
G-POD	space-global proper orthogonal decomposition.
LO-POD	space-local proper orthogonal decomposition with overlapping subdomains.
L-POD	space-local proper orthogonal decomposition.
LSTM	long short-term memory.
PDE	partial differential equation.
POD	proper orthogonal decomposition.
RK4	Runge-Kutta 4.
ROM	reduced-order model.
SVD	singular value decomposition.

CRediT authorship contribution statement

T. Van Gastelen: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization; **W. Edeling:** Writing – review & editing; **B. Sanderse:** Writing – review & editing, Supervision, Methodology.

Data availability

The code used to generate the training data, construct the ROMs, and replicate the presented experiments can be found at https://github.com/tobyvg/local_POD_overlap.jl.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT to improve language and grammar. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This publication is part of the project "Unraveling Neural Networks with Structure-Preserving Computing" (with project number OCENW.GROOT.2019.044 of the research programme NWO XL, which is financed by the Dutch Research Council (NWO)). Part of this publication is funded by Eindhoven University of Technology. Finally, we thank the reviewer and editor for their feedback, enhancing the quality of the article.

Appendix A. 2D Bump kernel

Here we discuss the kernel used to construct the LO-POD basis for the 2D advection test case. This kernel is required to obtain basis functions that smoothly decay to zero at the edge of the subdomain. To achieve this, we use a standard bump function:

$$\tilde{\psi}(x) = \begin{cases} \exp(\frac{-1}{1-|x|}) & \text{if } |x| < 1, \\ 0 & \text{elsewhere}, \end{cases} \tag{A.1}$$

to construct the kernel [48]. This function, along with its derivatives, smoothly decays to zero as $|x|$ approaches 1. Next, we apply normalization

$$\psi(x) = \begin{cases} \frac{\tilde{\psi}(x)}{\tilde{\psi}(x)+\tilde{\psi}(1-|x|)} & \text{if } |x| < 1, \\ 0 & \text{elsewhere}, \end{cases} \tag{A.2}$$

to ensure the kernels form a partition of unity, see (26). The 2D kernel is built using a product of two normalized bump functions

$$k(\mathbf{x}) = \psi(x)\psi(y), \tag{A.3}$$

where $\mathbf{x} = (x, y)^T \in \mathbb{R}^2$ are the spatial coordinates. For the presentation of this kernel, we exploit the fact that we use a uniform grid. This allows us to conveniently subdivide the domain into square subdomains Ω_i of the same size. Let $\boldsymbol{\alpha}^i, \boldsymbol{\beta}^i \in \mathbb{R}^2$ denote the coordinates of the bottom-left and top-right vertex. Each subdomain overlaps with its four neighboring subdomains, which share a vertex in the middle of the considered domain. The kernel associated with subdomain Ω_i is

$$k_i(\mathbf{x}) = \psi(\hat{x}_i)\psi(\hat{y}_i), \tag{A.4}$$

where $\hat{x}_i = 2\frac{x-\alpha_1^i}{\beta_1^i-\alpha_1^i} - 1$ and $\hat{y}_i = 2\frac{x-\alpha_2^i}{\beta_2^i-\alpha_2^i} - 1$ are the normalized coordinates. The kernel is depicted in Fig. A.1.

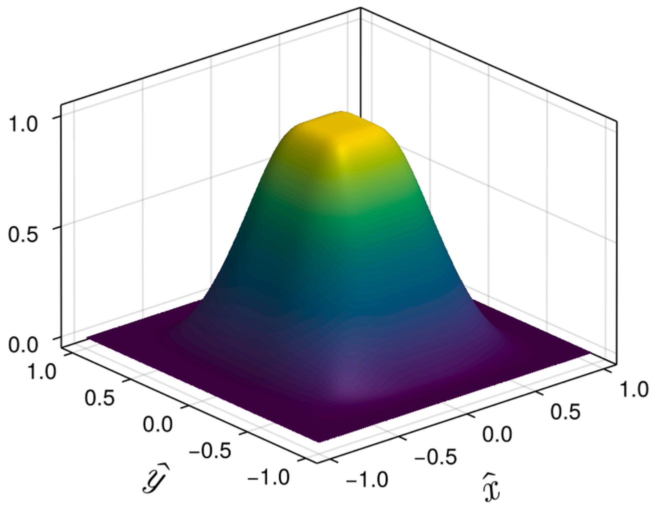


Fig. A.1. Bump kernel used to construct the LO-POD basis for the 2D advection test case.

Appendix B. Choice of subdomains and modes for 2D test case

In this section, we present the projection error on the training and validation sets for the 2D advection test case. This is presented in Fig. B.1.

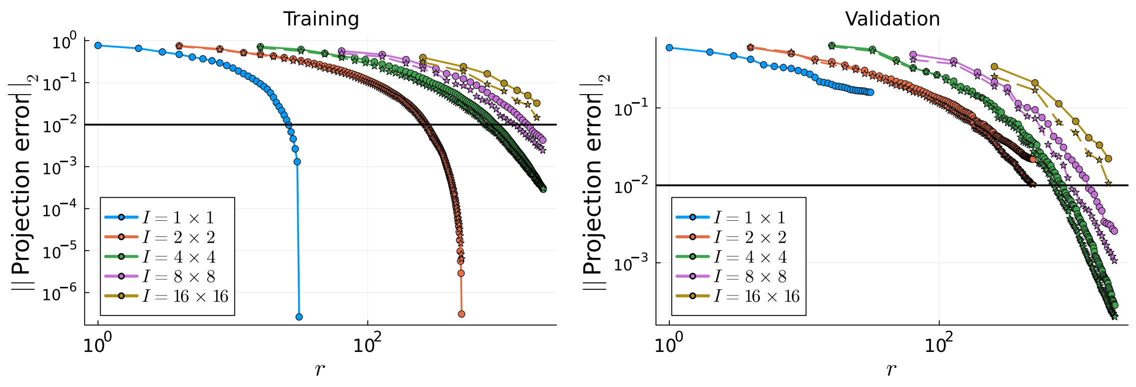


Fig. B.1. Projection error evaluated over the training (left) and validation (right) data set for the 2D advection test case. $I = 1 \times 1$ corresponds to G-POD. Solid dots correspond to L-POD and stars to LO-POD. The horizontal black line corresponds to a projection error of 10^{-2}.

Based on these results, we select the number of modes q which first surpasses the projection error of 10^{-2} for $I = 8 \times 8$. This value of I is chosen as it generalizes well, i.e., the performance on the validation set is similar to the training set and the resulting ROM is sparser than for $I = 4 \times 4$. For L-POD this corresponds to $q = 20$ and for LO-POD to $q = 15$.

Appendix C. Time step size

In Fig. C.1, we present the error of ROMs integrated using different time step sizes. The maximum time step sizes are selected so that performance degradation or the occurrence of instabilities is avoided. These are presented in Table 1.

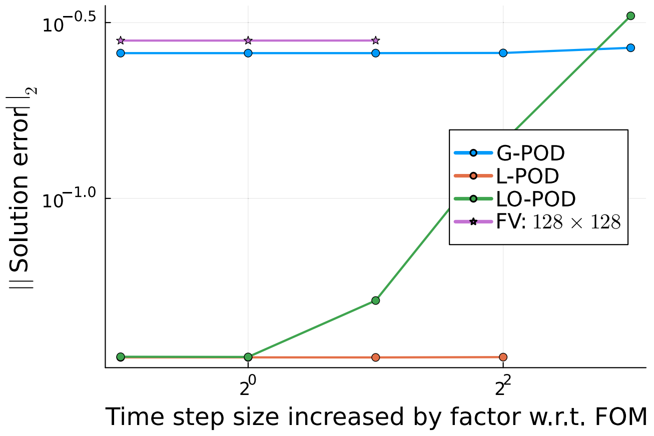


Fig. C.1. Solution error averaged over the simulation for the 2D advection test case, the ROMs for each time step size. Results for a finite volume (FV) discretization on a coarser grid than the FOM resolution (256×256) are also depicted. Absence of markers indicates an unstable simulation. The LO-POD ROM is integrated using a Crank-Nicolson scheme, as opposed to RK4 for the others, and is therefore unconditionally stable.

References

[1] Sasaki D, Obayashi S, Nakahashi K. Navier-Stokes optimization of supersonic wings with four objectives using evolutionary algorithm. J Aircr 2002;(4):621–9.
[2] Agdestein SD, Sanderse B. Discretize first, filter next: learning divergence-consistent closure models for large-eddy simulation. 2024. https://arxiv.org/abs/2403.18088.
[3] Alfonsi G. Reynolds-averaged Navier-Stokes equations for turbulence modeling. Appl Mech Rev 2009;62. https://doi.org/10.1115/1.3124648
[4] Sagaut P, Lee Y-T. Large eddy simulation for incompressible flows: an introduction. scientific computation series. Appl Mech Rev 2002;55:115. https://doi.org/10.1115/1.1508154
[5] Brunton SL, Kutz JN. Data-driven science and engineering: machine learning, dynamical systems, and control. USA: Cambridge University Press; 1st ed.; 2019. ISBN 1108422098.
[6] Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 2015;57 (4):483–531. https://doi.org/10.1137/130932715
[7] Burkardt J, Gunzburger M, Lee H-C. POD and CVT-based reduced-order modeling of Navier-Stokes flows. Comput Methods Appl Mech Eng 2006;196:337–55. https://doi.org/10.1016/j.cma.2006.04.004
[8] Rapun M-L, Vega J. Reduced order models based on local POD plus Galerkin projection. J Comput Phys 2010;229. https://doi.org/10.1016/j.jcp.2009.12.029
[9] Sanderse B. Non-linearly stable reduced-order models for incompressible flow with energy-conserving finite volume methods. J Comput Phys 2020;421:109736. https://doi.org/10.1016/j.jcp.2020.109736
[10] Dutta S, Rivera-Casillas P, Styles B, Farthing MW. Reduced order modeling using advection-aware autoencoders. Math. Comput. Appl. 2022;27 (3). https://doi.org/10.3390/mca27030034
[11] Arbes F, Greif C, Urban K. The Kolmogorov N-width for linear transport: exact representation and the influence of the data. 2024. https://arxiv.org/abs/2305.00066.
[12] Taddei T, Perotto S, Quarteroni A. Reduced basis techniques for nonlinear conservation laws. ESAIM: Math. Modell. Numer. Anal. 2015;49 (3):787–814.
[13] Greif C, Urban K. Decay of the Kolmogorov N-width for wave problems. Appl Math Lett 2019;96:216–22.
[14] Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. J Comput Phys 2013;242:623–47.
[15] Nair NJ, Balajewicz M. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. Int J Numer Methods Eng 2019;117 (12):1234–62.
[16] Washabaugh K, Amsallem D, Zahr M, Farhat C. Nonlinear Model Reduction for CFD Problems Using Local Reduced Order Bases. 2012, https://doi.org/10.2514/6.2012-2686
[17] Klein RB, Sanderse B. Energy-conserving hyper-reduction and temporal localization for reduced order models of the incompressible Navier-Stokes equations. J Comput Phys 2024;499:112697. https://doi.org/10.1016/j.jcp.2023.112697
[18] Peherstorfer B, Willcox K. Online adaptive model reduction for nonlinear systems via low-rank updates. SIAM J Sci Comput 2015;37 (4):A2123–A2150. https://doi.org/10.1137/140989169
[19] Grimberg S, Farhat C, Youkilis N. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. J Comput Phys 2020;419:109681. https://doi.org/10.1016/j.jcp.2020.109681
[20] Wang Z, Akhtar I, Borggaard J, Iliescu T. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. Comput Methods Appl Mech Eng 2012;237-240:10–26. https://doi.org/10.1016/j.cma.2012.04.015
[21] Klein R, Sanderse B, Costa P, Pecnik R, Henkes R. Entropy-stable model reduction of one-dimensional hyperbolic systems using rational quadratic manifolds. 2024. https://arxiv.org/abs/2407.12627.
[22] Mohan AT, Gaitonde DV. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. 2018. https://arxiv.org/abs/1804.09269.
[23] Mücke NT, Bohté SM, Oosterlee CW. Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning. J Comput Sci 2021;53:101408. . https://doi.org/10.1016/j.jocs.2021.101408
[24] Reddy JN. Introduction to the finite element method. New York: McGraw-Hill Education; 4th edition ed.; 2019. ISBN 9781259861901. https://www.accessengineeringlibrary.com/content/book/9781259861901.
[25] Anderson S, White T, Farhat C. Space-local reduced-order bases for accelerating reduced-order models through sparsity. Int J Numer Methods Eng 2022;124. https://doi.org/10.1002/nme.7179
[26] Quarteroni A, Manzoni A, Negri F. Reduced basis methods for partial differential equations: an introduction. 2015. ISBN 978-3-319-15430-5. https://doi.org/10.1007/978-3-319-15431-2
[27] Chung SW, Choi Y, Roy P, Moore T, Roy T, Lin TY, et al. Train small, model big: scalable physics simulators via reduced order modeling and domain decomposition. Comput Methods Appl Mech Eng 2024;427:117041. https://doi.org/10.1016/j.cma.2024.117041
[28] Butcher J. Runge-Kutta methods. Scholarpedia 2007;2 (9):3147. . Revision #91735. https://doi.org/10.4249/scholarpedia.3147
[29] Sanderse B. Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. J Comput Phys 2013;233:100–31. https://doi.org/10.1016/j.jcp.2012.07.039
[30] Aiken JG, Erdos JA, Goldstein JA. On Löwdin orthogonalization. Int J Quantum Chem 1980;18 (4):1101–8.
[31] Li X, Wang S, Cai Y. Tutorial: complexity analysis of singular value decomposition and its variants. 2019. https://arxiv.org/abs/1906.12085.
[32] Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. Comput Methods Appl Mech Eng 1996;139 (1):289–314. https://doi.org/10.1016/S0045-7825(96)01087-0
[33] Dragomir SS, Cerone P, Sofo A. Some remarks on the midpoint rule in numerical integration. RGMIA Res Rep Collect 1998;1 (2):25–33.
[34] Ştefănescu R, Navon IM. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. J Comput Phys 2013;237:95–114.
[35] Nguyen VB, Tran S BQ, Khan SA, Rong J, Lou J. POD-DEIM model order reduction technique for model predictive control in continuous chemical processing. Comput. Chem. Eng. 2020;133:106638. https://doi.org/10.1016/j.compchemeng.2019.106638
[36] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev 2017;59 (1):65–98. https://doi.org/10.1137/141000671
[37] Baldauf M. Stability analysis for linear discretisations of the advection equation with Runge-Kutta time integration. J Comput Phys 2008;227 (13):6638–59. . https://doi.org/10.1016/j.jcp.2008.03.025
[38] Johnson C. Error estimates and automatic time step control for nonlinear parabolic problems, I. SIAM J Numer Anal 1987;24 (1):1–14. https://doi.org/10.1137/0724001
[39] Sanderse B. Energy-conserving discretization methods for the incompressible Navier-Stokes equations : application to the simulation of wind-turbine wakes. Phd thesis 2 (research not tu/e / graduation tu/e); Centrum voor Wiskunde en Informatica; 2013. https://doi.org/10.6100/IR750543
[40] Snyder W, Mou C, Liu H, San O, De Vita R, Iliescu T. Reduced order model closures: a brief tutorial. 2022. https://arxiv.org/abs/2202.14017.
[41] Mittal RC, Al-Kurdi A. LU-decomposition and numerical structure for solving large sparse nonsymmetric linear systems. Comput Math Appl 2002;43 (1):131–55. https://doi.org/10.1016/S0898-1221(01)00279-6
[42] Kochkov D, Smith JA, Alieva A, Wang Q, Brenner MP, Hoyer S. Machine learning-accelerated computational fluid dynamics. Proc Natl Acad Sci 2021;118 (21). . https://doi.org/10.1073/pnas.2101784118
[43] Shankar V, Chakraborty D, Viswanathan V, Maulik R. Differentiable turbulence: closure as a partial differential equation constrained optimization. 2024. https://arxiv.org/abs/2307.03683.
[44] Agdestein SD, Sanderse B. Discretize first, filter next: learning divergence-consistent closure models for large-eddy simulation. J Comput Phys 2025;522:113577. https://doi.org/10.1016/j.jcp.2024.113577
[45] Harlow FH, Welch JE, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys Fluids 1965;8 (12):2182.
[46] Lequeurre J, Munnier A. Vorticity and stream function formulations for the 2D Navier-Stokes equations in a bounded domain. 2018. https://arxiv.org/abs/1810.04222.
[47] Rosenberger H, Sanderse B. No pressure? Energy-consistent ROMs for the incompressible Navier-Stokes equations with time-dependent boundary conditions. 2022. https://arxiv.org/abs/2212.12036.
[48] Lee J. Introduction to smooth manifolds. 2nd revised ed; vol. 218. New York: Springer; 2012. ISBN 978-1-4419-9981-8. https://doi.org/10.1007/978-1-4419-9982-5