

Gradient preserving Operator Inference: Data-driven reduced-order models for equations with gradient structure

Yuwei Geng^a, Jasdeep Singh^a, Lili Ju^a, Boris Kramer^b, Zhu Wang^{a,*}

^a Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

^b Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, USA

ARTICLE INFO

MSC:

65P99

65M15

Keywords:

Dissipative and conservative systems

Gradient-flow equation

Operator inference

Reduced-order modeling

Data-driven modeling

ABSTRACT

Hamiltonian Operator Inference has been introduced in Sharma et al. (2022) to learn structure-preserving reduced-order models (ROMs) for Hamiltonian systems. This approach constructs a low-dimensional model using only data and knowledge of the Hamiltonian function. Such ROMs can keep the intrinsic structure of the system, allowing them to capture the physics described by the governing equations. In this work, we extend this approach to more general systems that are either conservative or dissipative in energy, and which possess a gradient structure. We derive the optimization problems for inferring structure-preserving ROMs that preserve the gradient structure. We further derive an *a priori* error estimate for the reduced-order approximation. To test the algorithms, we consider semi-discretized partial differential equations with gradient structure, such as the parameterized wave and Korteweg–de-Vries equations, and equations of three-dimensional linear elasticity in the conservative case and the one- and two-dimensional Allen-Cahn equations in the dissipative case. The numerical results illustrate the accuracy, structure-preservation properties, and predictive capabilities of the gradient-preserving Operator Inference ROMs.

1. Introduction

Consider evolutionary partial differential equations (PDEs) of the following form

$$\frac{\partial y(x, t; \mu)}{\partial t} = \mathcal{L} \frac{\delta H[y; \mu]}{\delta y}, \quad (1)$$

where the state variable y depends on the spatial variable $x \in \Omega \subset \mathbb{R}^d$ ($d = 1, 2$ or 3), time variable $t \in I \subset [t_0, \infty)$ and parameter $\mu \in P \subset \mathbb{R}^p$, $p \geq 1$. Moreover, $\mathcal{L}$ represents a constant linear differential operator, and $\frac{\delta H}{\delta y}$ is the variational derivative of the functional

$$H[y; \mu] := \int_{\Omega} H(y, y_x, y_{xx}, \dots; \mu) dx$$

with $y_x, y_{xx}, \dots$, denoting the first- and second-order (and higher order) partial derivatives of y with respect to x . Depending on the properties of $\mathcal{L}$, the PDE can possess different features. If $\mathcal{L}$ is skew adjoint, H , referred to as the Hamiltonian, is a constant and the PDE is conservative. If $\mathcal{L}$ is negative semi-definite (resp. definite), H , referred to as a Lyapunov function or energy function, is nonincreasing (resp. monotonically decreasing) and the PDE is dissipative.

* Corresponding author.

E-mail addresses: ygeng@email.sc.edu (Y. Geng), jasdeep@email.sc.edu (J. Singh), ju@math.sc.edu (L. Ju), bmramer@ucsd.edu (B. Kramer), wangzhu@math.sc.edu (Z. Wang).

<https://doi.org/10.1016/j.cma.2024.117033>

Received 21 January 2024; Received in revised form 18 April 2024; Accepted 28 April 2024

Available online 8 May 2024

0045-7825/© 2024 Elsevier B.V. All rights reserved.

Either class of PDEs has broad applications ranging from mechanics [1], plasma physics [2,3] to ocean modeling [4–7]. Numerical methods have been developed to approximate and predict solutions to such PDEs. It is well known that numerical schemes that do not preserve the continuous models' properties are prone to suffer from error accumulation and generate unphysical solutions over a long-term integration. Therefore, structure-preserving algorithms have been designed to keep the right features at a discrete level, see the review papers [8,9] and the reference therein. In particular, structure-preserving spatial discretization schemes include finite difference methods such as [10,11] and finite element methods such as [8,12–15]. Structure-preserving time discretization schemes include geometric integrators [16–18], discrete gradient methods [19,20], average vector field (AVF) methods [21,22], and discrete variational derivative methods [23].

The cost (e.g., measured in CPU hours) to simulate a high-dimensional discretization of (1) can become a limiting factor when one wants to simulate either for long time horizons, and/or obtain many parametric solutions for different μ_i in the context of exploration and uncertainty quantification. To accelerate the simulation of high-dimensional systems, reduced-order models (ROMs) can provide orders-of-magnitude speedups while providing accurate approximate solutions, see the surveys/books [24–28]. However, ROMs constructed by projecting the equation onto a linear subspace spanned by a reduced basis, or onto a nonlinear manifold, could destroy the features of mechanical systems. In the context of Lagrangian structure, this has been shown in [29], and motivated the development of projection-based model reduction techniques that preserve the Lagrangian structure [30]. For systems with Hamiltonian and port-Hamiltonian systems, structure-preserving projection-based model reduction has been developed, including the interpolatory projection method proposed in [31] that preserves the symmetry of linear dynamical systems and is extended to port-Hamiltonian systems via tangential rational interpolation in [32]; the proper orthogonal decomposition (POD)-based Galerkin projection method in [33–35] for Hamiltonian systems and the POD/ H_2 -based Petrov–Galerkin project method in [36,37] for port-Hamiltonian; the proper symplectic decomposition in [38] and a Petrov–Galerkin projection-based, variationally consistent approach in [39] for canonical Hamiltonian systems; and the reduced basis method for Hamiltonian systems in [40–43]. Moreover, there has been increased interest lately in projection-based model reduction on nonlinear manifolds for Hamiltonian systems, see [44–46].

ROMs can also be constructed non-intrusively, which is appealing in situations where only snapshot data of state variables is available, paired with potentially some prior knowledge of the PDE model. In particular, the operator inference (OpInf) method provides such a nonintrusive framework for model order reduction, which was introduced in [47] and expanded in many directions, c.f. the survey paper [48]. The recent work [49] extended OpInf to affine-parametric systems of PDEs, yet the adaption of the framework to problems with unknown parametric structures is nontrivial. Within the OpInf framework, learning canonical Hamiltonian ROMs on linear subspaces is considered in [50], where the linear part of the gradient of the Hamiltonian is inferred through constrained least-squares solutions. This has been extended to noncanonical Hamiltonian systems in [51], where the entire Hamiltonian function is assumed to be known, and the reduced operator associated to the linear differential operator $\mathcal{L}$ is inferred. Port-Hamiltonian [52] and Lagrangian ROMs can also be inferred directly from time-domain data, see [53–55].

Motivated by these developments, in this paper, we propose *gradient-preserving Operator Inference (GP-OPINF)*, a more general framework to nonintrusively construct ROMs that preserve the gradient structure of the PDE model (1). In this nonintrusive setting, we assume that we are given snapshot data and some prior knowledge of the continuous model and the functional $\mathcal{H}$, yet that we do not have access to the spatially-discretized forms in high-fidelity solvers. We note that the proposed approach is not limited to Hamiltonian systems. The novelties of this work include the following:

1. We propose suitable optimization problems for the gradient-preserving Operator Inference method. In particular, for conservative equations, the solution to the optimization problem is post-processed for improving the gradient structure of ROMs without introducing any consistency errors.
2. We derive an *a priori* error estimate for the inferred GP-OPINF ROM, which we demonstrate by numerical experiments.
3. Our framework applies to dissipative and conservative systems with a gradient structure.

The rest of this paper is organized as follows. In Section 2, we introduce the full-order model form that we consider and, for comparison, an intrusive structure-preserving ROM for Eq. (1). In Section 3, we introduce the gradient-preserving Operator Inference and present the associated optimization problems. In Section 4 we derive an *a priori* error estimate for the learned GP-OPINF ROM. We then demonstrate the effectiveness of the proposed ROM through several numerical examples in Section 5. A few concluding remarks are drawn in the last section.

2. Full-order model and structure-preserving reduced-order model

We present the full-order model setting in Section 2.1 and then discuss the traditional structure-preserving intrusive ROM approach in Section 2.2. This provides the background for the new method in the following Section 3.

To ease the exposition, in situation when no confusion arises, we omit the explicit reference to the parameter and adopt the following notations: at any $\mu \in P$, for any vector-valued function $\mathbf{f}$ and scalar-valued function g ,

$$\mathbf{f}(t) \equiv \mathbf{f}(t; \mu), \quad g(\mathbf{f}) \equiv g(\mathbf{f}(t; \mu); \mu), \quad \forall t \in I.$$

2.1. Full-order model

For any parameter (vector) $\mu \in P \subset \mathbb{R}^p, p \geq 1$, we call the system of finite dimensional ODEs obtained from (1) after a structure-preserving spatial discretization the full-order model (FOM). Here, we consider the FOM

$$\dot{\mathbf{y}} = \mathbf{D} \nabla_{\mathbf{y}} H(\mathbf{y}), \quad (2)$$

where $\mathbf{y}(t) \equiv \mathbf{y}(t; \mu) \in \mathbb{R}^n$ is the n -dimensional state vector and the initial condition is $\mathbf{y}(t^0) = \mathbf{y}^0$. The right-hand-side of (2) represents the velocity field (or drift), in which $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a real matrix, $H(\mathbf{y}) \equiv H(\mathbf{y}; \mu) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function of the state and $\nabla_{\mathbf{y}} H(\mathbf{y})$ is the gradient of $H(\mathbf{y})$. The gradient structure of Eq. (2) plays a key role in characterizing the dynamics:

- (i) When $\mathbf{D}$ is anti-symmetric (also referred to as skew-symmetric or skew-adjoint), that is, $\mathbf{D} = -\mathbf{D}^T$, the system is Hamiltonian. A special case is when $\mathbf{D} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix}$ then Eq. (2) represents a canonical Hamiltonian system and the solution flow is symplectic. Due to such a structure, the internal energy of the system, $H(\mathbf{y})$, is conserved. Mathematically, for any $t_1, t_2 \in I$ with $t_1 < t_2$, we have

$$H(\mathbf{y}(t_2)) - H(\mathbf{y}(t_1)) = \int_{t_1}^{t_2} \frac{d}{dt} H(\mathbf{y}(t)) dt = \int_{t_1}^{t_2} (\nabla_{\mathbf{y}} H(\mathbf{y}))^T \mathbf{D} \nabla_{\mathbf{y}} H(\mathbf{y}) dt = 0.$$

- (ii) When $\mathbf{D}$ is negative semi-definite, the system represents a gradient flow and is dissipative. To see this, we can easily check

$$H(\mathbf{y}(t_2)) - H(\mathbf{y}(t_1)) = \int_{t_1}^{t_2} \frac{d}{dt} H(\mathbf{y}(t)) dt = \int_{t_1}^{t_2} (\nabla_{\mathbf{y}} H(\mathbf{y}))^T \mathbf{D} \nabla_{\mathbf{y}} H(\mathbf{y}) dt \leq 0.$$

Note that if $\mathbf{D}$ is negative definite, $H(\mathbf{y})$ is strictly decreasing. Numerical schemes to solve these equations recognize the special gradient structure in time discretization, e.g., geometric integrators and average vector field methods.

If the size of the dynamical system is large, it is expensive to simulate (2), especially for many parameters or for long time intervals. We next review intrusive projection-based ROMs that can provide accurate approximations at much lower computational cost. To preserve physics of the system, they should possess the same gradient structure.

2.2. Intrusive structure-preserving ROMs

We review an intrusive Galerkin-projection-based approach to build the structure-preserving ROM, termed *SP-G ROM*, which was introduced in [33]. The first step in the ROM construction is to extract the low-dimensional basis in which we derive the ROM. Here, we use the widely used snapshot-based POD method [56]. For a non-parametric problem, we consider μ as fixed, and the snapshots (state vectors at select time instances) are stored in the matrix

$$\mathbf{Y} \equiv \mathbf{Y}(\mu) := [\mathbf{y}(t^0; \mu), \mathbf{y}(t^1; \mu), \dots, \mathbf{y}(t^s; \mu)]. \quad (3)$$

For simplicity, we assume the snapshot set contains the FOM solutions at all the uniformly distributed discrete time discretization points, that is, $t^{i+1} = t^i + \Delta t$ for $i = 0, \dots, s-1$. However, in practice one may use only a portion of them as the snapshots. For a parametric problem, one is interested in a set of parameters $\{\mu^0, \mu^1, \dots, \mu^p\}$ from P , for which full-order solutions are generated. The associated snapshot data matrix is defined as

$$\mathbf{Y} := [\mathbf{Y}(\mu^0), \mathbf{Y}(\mu^1), \dots, \mathbf{Y}(\mu^p)]. \quad (4)$$

The POD basis matrix $\Phi = [\phi_1, \phi_2, \dots, \phi_r]$ contains the leading left singular vectors of $\mathbf{Y}$ as its columns, which can be computed by the method of snapshots [57] or singular value decomposition (SVD) and related randomized algorithms.

The reduced-order approximation of the state variable is defined by a linear ansatz:

$$\hat{\mathbf{y}}(t; \mu) := \Phi \mathbf{y}_r(t; \mu), \quad (5)$$

where $\mathbf{y}_r(t; \mu) \in \mathbb{R}^r$ are the unknowns, which are determined by integrating the reduced-order model in time. A centering trajectory (e.g., the average of snapshots) $\bar{\mathbf{y}}$ is usually subtracted from each snapshot before calculating the POD modes, which helps in enforcing fixed inhomogeneous Dirichlet boundary conditions. In that case, $\hat{\mathbf{y}}(t; \mu) = \bar{\mathbf{y}} + \Phi \mathbf{y}_r(t; \mu)$. We assume $\bar{\mathbf{y}} = \mathbf{0}$ in this work for simplicity. This linear ansatz has been adapted in [58] for reduced order modeling of problems with parametric, time-varying, inhomogeneous Dirichlet-type boundary conditions.

The *SP-G ROM* applies the ansatz (5). To distinguish it from the non-intrusive reduced-order ROMs developed later in this paper, we let $\tilde{\mathbf{y}}_r(t; \mu)$ denote the *SP-G ROM*'s reduced-order state. The ROM is then constructed by first projecting the FOM onto the low-dimensional basis given by Φ , and subsequently adjusting it to ensure the correct gradient structure appears in the resulting ROM. This yields the *SP-G ROM*

$$\dot{\tilde{\mathbf{y}}}_r = \tilde{\mathbf{D}}_r \nabla_{\tilde{\mathbf{y}}_r} H_r(\tilde{\mathbf{y}}_r) \quad (6)$$

for $\tilde{\mathbf{y}}_r(t) \equiv \tilde{\mathbf{y}}_r(t; \mu) \in \mathbb{R}^r$ with $\tilde{\mathbf{y}}_r(t^0) = \Phi^T \mathbf{y}^0$, where $\tilde{\mathbf{D}}_r = \Phi^T \mathbf{D} \Phi$, $H_r(\tilde{\mathbf{y}}_r) := H(\Phi \tilde{\mathbf{y}}_r)$, and $\nabla_{\tilde{\mathbf{y}}_r} H_r(\tilde{\mathbf{y}}_r) = \Phi^T \nabla_{\mathbf{y}} H(\Phi \tilde{\mathbf{y}}_r)$. With this construction, the right-hand side of the ROM (6) has the same gradient structure as the FOM (2). Thus, the ROM has the same properties as its full-order counterpart. If the system is Hamiltonian, H_r is also guaranteed to be constant.

We remark that an error exists between $H_r(\tilde{\mathbf{y}}_r)$ and $H(\mathbf{y})$, which is due to the projected initial condition $\tilde{\mathbf{y}}_r(t^0)$ in the ROM. However, this error shrinks as r increases, as a larger ROM basis produces a more accurate ROM initial condition. Specifically, when the initial condition is independent of any parameters, it can be used as the centering trajectory, eliminating the error caused by projecting the initial condition. For details, the reader is referred to [33].

The above construction of *SP-G ROM* requires access to the full-order coefficient matrix $\mathbf{D}$, knowledge of H or its gradient $\nabla_{\mathbf{y}} H$, and snapshot data. In the next section, we introduce an Operator Inference approach to build a data-driven ROM for (2) with the appropriate gradient structure, but without accessing the full-order coefficient matrix.

3. Gradient-preserving operator inference

A structure-preserving Operator Inference for canonical Hamiltonian systems, named *H-OpInf*, has been developed in [50], based on the unconstrained operator inference from [47,48]. The symplecticity-preserving method learns the reduced-order operator associated with the quadratic part of the Hamiltonian function and assumes knowledge of the nonquadratic part of the Hamiltonian function. The motivational example in [50, Sec. 3.1] shows that the OpInf ROM without preserving the symplectic structure produces an increasing Hamiltonian energy for a canonical linear wave equation, resulting in inaccurate and unphysical reduced-order simulations. This occurs already within the training data regime. Non-canonical Hamiltonian systems are considered in [51], where the authors construct a ROM that preserves the (Hamiltonian) energy under the assumption of prior knowledge of the entire Hamiltonian function. In this work, we propose a more general gradient-preserving Operator Inference (GP-OpInf) approach for both conservative and dissipative systems governed by the general FOM (2). In Section 3.1 we introduce the new framework. Section 3.2 presents solution approaches for the conservative case and Section 3.3 for the dissipative case.

3.1. Proposed GP-OPINF framework

We assume that snapshot data $\mathbf{Y}$ given in Eqs. (3) or (4) as well as the symbolic expression of $H(\mathbf{y})$ or the associated gradient is given. The goal is to develop a non-intrusive gradient-preserving ROM. We do this by employing the linear ansatz (5), characterized by the expression $\hat{\mathbf{y}}(t; \mu) = \Phi \mathbf{y}_r(t; \mu)$, to find $\mathbf{y}_r(t) \equiv \mathbf{y}_r(t; \mu) \in \mathbb{R}^r$ satisfying

$$\dot{\mathbf{y}}_r = \mathbf{D}_r \nabla_{\mathbf{y}_r} H(\mathbf{y}_r) \quad (7)$$

with $\mathbf{y}_r(t^0) = \Phi^\top \mathbf{y}^0$. Here $\mathbf{D}_r$ is unknown and will be inferred from data. Specifically, $\mathbf{D}_r$ is anti-symmetric if the system is conservative; and is negative semi-definite if the system is dissipative. We refer to Eq. (7) as the *GP-OpInf* ROM since it has the same gradient structure as the FOM (2).

To learn $\mathbf{D}_r$, the gradient data are computed based on knowledge of the gradient of the Hamiltonian function and available snapshot data. In the non-parametric case,

$$\mathbf{F} \equiv \mathbf{F}(\mu) := [\nabla_{\mathbf{y}} H(\mathbf{y}(t^0; \mu); \mu), \nabla_{\mathbf{y}} H(\mathbf{y}(t^1; \mu); \mu), \dots, \nabla_{\mathbf{y}} H(\mathbf{y}(t^s; \mu); \mu)] . \quad (8)$$

The first-order time derivative data can be generated by applying a finite difference operator, denoted as $D_t[\cdot]$, on the state vectors:

$$\dot{\mathbf{Y}} \equiv \dot{\mathbf{Y}}(\mu) := [D_t[\mathbf{y}(t^0; \mu)], D_t[\mathbf{y}(t^1; \mu)], \dots, D_t[\mathbf{y}(t^s; \mu)]] .$$

For instance, $D_t[\cdot]$ can be chosen to be the following second-order finite difference operator, satisfying

$$D_t[\mathbf{y}(t^j; \mu)] = \begin{cases} (-3\mathbf{y}(t^0; \mu) + 4\mathbf{y}(t^1; \mu) - \mathbf{y}(t^2; \mu))/(2\Delta t) & j = 0, \\ (\mathbf{y}(t^{j+1}; \mu) - \mathbf{y}(t^{j-1}; \mu))/(2\Delta t) & j = 1, \dots, s-1, \\ (\mathbf{y}(t^{s-2}; \mu) - 4\mathbf{y}(t^{s-1}; \mu) + 3\mathbf{y}(t^s; \mu))/(2\Delta t) & j = s. \end{cases} \quad (9)$$

In the parametric case, $\mathbf{F} := [\mathbf{F}(\mu^0), \mathbf{F}(\mu^1), \dots, \mathbf{F}(\mu^p)]$ and $\dot{\mathbf{Y}} := [\dot{\mathbf{Y}}(\mu^0), \dot{\mathbf{Y}}(\mu^1), \dots, \dot{\mathbf{Y}}(\mu^p)]$. In either case, using the POD basis Φ , we project each of the above data matrices onto the POD basis and obtain the projected data

$$\mathbf{Y}_r = \Phi^\top \mathbf{Y}, \quad \mathbf{F}_r = \Phi^\top \mathbf{F}, \quad \text{and} \quad \dot{\mathbf{Y}}_r = \Phi^\top \dot{\mathbf{Y}}.$$

The *GP-OpInf* learns the reduced operator $\mathbf{D}_r$ from the following optimization problems:

(i) When the system is conservative:

$$\min_{\mathbf{D}_r \in \mathbb{R}^{r \times r}} \frac{1}{2} \|\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r\|_F^2 \quad \text{s.t.} \quad \mathbf{D}_r^\top = -\mathbf{D}_r. \quad (10)$$

(ii) When the system is dissipative:

$$\min_{\mathbf{D}_r \in \mathbb{R}^{r \times r}} \frac{1}{2} \|\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r\|_F^2 \quad \text{s.t.} \quad \mathbf{D}_r \preceq 0, \quad (11)$$

where $\mathbf{D}_r \preceq 0$ indicates that $\mathbf{D}_r$ is semi-negative definite.

The next two sections present suitable solutions to these respective optimization problems.

3.2. Solution of the GP-OpInf optimization for the conservative case

The solution to the optimization problem (10) can be derived using the method of Lagrange multipliers. Consider the Lagrangian function

$$L(\mathbf{D}_r) = \frac{1}{2} \left\langle \dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r, \dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r \right\rangle_F + \left\langle \Lambda, \mathbf{D}_r + \mathbf{D}_r^\top \right\rangle_F,$$

where the Frobenius inner product is defined as $\langle \mathbf{A}, \mathbf{B} \rangle_F := \sum_{i,j} A_{ij} B_{ij}$ for two real matrices $\mathbf{A}$ and $\mathbf{B}$. The stationary point of the Lagrangian then satisfies

$$(\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r) \mathbf{F}_r^\top = \mathbf{\Lambda} + \mathbf{\Lambda}^\top, \quad (12)$$

$$\mathbf{D}_r^\top = -\mathbf{D}_r. \quad (13)$$

Using the symmetry of the right hand side in Eq. (12) together with Eq. (13), we get the Lyapunov equation

$$\mathbf{F}_r \mathbf{F}_r^\top \mathbf{D}_r + \mathbf{D}_r \mathbf{F}_r \mathbf{F}_r^\top = \dot{\mathbf{Y}}_r \mathbf{F}_r^\top - \mathbf{F}_r \dot{\mathbf{Y}}_r^\top. \quad (14)$$

We next outline three solution approaches to obtain $\mathbf{D}_r$, each leading to a different ROM. We then provide numerical comparisons in Section 5.

GP-OpInf-V: This straightforward (“vanilla”) approach directly solves the Lyapunov equation (14). Given square matrices $\mathbf{A}$, $\mathbf{Q} \in \mathbb{R}^{n \times n}$, for a general continuous-time Lyapunov equation

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^\top = \mathbf{Q},$$

it is known that (1) if $\mathbf{A}$ and $-\mathbf{A}^\top$ have no eigenvalues in common, then the solution $\mathbf{X}$ is unique; (2) if $\mathbf{Q}$ is (anti-)symmetric, then $\mathbf{X}$ must be (anti-)symmetric, see, e.g., [59]. Since $\mathbf{F}_r \mathbf{F}_r^\top$ in (14) is only positive semi-definite, there might be zero eigenvalues, violating (1), and the system would become singular. In this case, standard software libraries for numerical linear algebra, such as LAPACK [60], provide a unique solution $\mathbf{D}_r$, yet to a slightly perturbed system. This solution may not be symmetric to machine precision anymore, and due to numerical errors in the solver, $\mathbf{F}_r \mathbf{F}_r^\top$ can have small negative eigenvalues. This could introduce additional errors in the numerical simulation of the GP-OpInf ROM. To resolve this issue, we next consider a regularized optimization.

GP-OpInf-P: We introduce a regularization term to the optimization: for some $\epsilon > 0$

$$\min_{\mathbf{D}_r \in \mathbb{R}^{r \times r}} \frac{1}{2} \|\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r\|_F^2 + \frac{\epsilon}{2} \|\mathbf{D}_r\|_F^2 \quad \text{s.t.} \quad \mathbf{D}_r^\top = -\mathbf{D}_r. \quad (15)$$

We use similar arguments as above and define the Lagrangian

$$L(\mathbf{D}_r) = \frac{1}{2} \left\langle \dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r, \dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r \right\rangle_F + \frac{\epsilon}{2} \langle \mathbf{D}_r, \mathbf{D}_r \rangle_F + \langle \mathbf{\Lambda}, \mathbf{D}_r + \mathbf{D}_r^\top \rangle_F,$$

whose stationary point satisfies

$$(\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r) \mathbf{F}_r^\top - \epsilon \mathbf{D}_r = \mathbf{\Lambda} + \mathbf{\Lambda}^\top, \\ \mathbf{D}_r^\top = -\mathbf{D}_r.$$

This leads to the “perturbed” Lyapunov equation

$$(\mathbf{F}_r \mathbf{F}_r^\top + \epsilon \mathbf{I}) \mathbf{D}_r + \mathbf{D}_r (\mathbf{F}_r \mathbf{F}_r^\top + \epsilon \mathbf{I}) = \dot{\mathbf{Y}}_r \mathbf{F}_r^\top - \mathbf{F}_r \dot{\mathbf{Y}}_r^\top, \quad (16)$$

where $\mathbf{I}$ is the $r \times r$ identity matrix.

In order to prevent the eigenvalues of the coefficient matrix $\mathbf{F}_r \mathbf{F}_r^\top + \epsilon \mathbf{I}$ from being non-positive, we define

$$\epsilon = \begin{cases} \max(c_0, \alpha |\min(\lambda(\mathbf{F}_r \mathbf{F}_r^\top))|) & \text{if } \min(\lambda(\mathbf{F}_r \mathbf{F}_r^\top)) \leq c_0, \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

with a small positive constant c_0 and a positive constant α . This approach improves the anti-symmetry of the inferred $\mathbf{D}_r$. However, comparing with (14), the perturbed equation could introduce a consistency error.

GP-OpInf: We propose an alternative approach to eliminate this common consistency error. Suppose that $\hat{\mathbf{D}}_r$ satisfies (14), that is,

$$\mathbf{F}_r \mathbf{F}_r^\top \hat{\mathbf{D}}_r + \hat{\mathbf{D}}_r \mathbf{F}_r \mathbf{F}_r^\top = \dot{\mathbf{Y}}_r \mathbf{F}_r^\top - \mathbf{F}_r \dot{\mathbf{Y}}_r^\top. \quad (18)$$

We assume that the equation can be uniquely solved. In the case where $\mathbf{F}_r \mathbf{F}_r^\top$ has zero eigenvalues, the approach used in LAPACK should be applied to provide a unique solution [60]. As discussed above, $\hat{\mathbf{D}}_r$ may not be exactly anti-symmetric. After transposing both sides of (18) and taking the negative, we obtain

$$-\mathbf{F}_r \mathbf{F}_r^\top \hat{\mathbf{D}}_r - \hat{\mathbf{D}}_r \mathbf{F}_r \mathbf{F}_r^\top = \dot{\mathbf{Y}}_r \mathbf{F}_r^\top - \mathbf{F}_r \dot{\mathbf{Y}}_r^\top. \quad (19)$$

By averaging (18) and (19), we find that (14) admits the anti-symmetric solution

$$\mathbf{D}_r = (\hat{\mathbf{D}}_r - \hat{\mathbf{D}}_r^\top)/2. \quad (20)$$

Thus, the inferred $\mathbf{D}_r$, defined in (20), is superior to the one obtained from (16) because it possesses the right property without causing any consistency errors to the optimization problem.

3.3. Solution of the GP-OpInf optimization for the dissipative case

The solution to the optimization problem (11) for the dissipative case can be obtained through an interior-point method [61,62]. By introducing $f(\mathbf{X}) = \|\dot{\mathbf{Y}}_r - \mathbf{X}\mathbf{F}_r\|_F^2$ and the logarithmic barrier

$$g(\mathbf{X}) = \begin{cases} \sum_{i=1}^r \log(-\lambda_i(\mathbf{X})) & \text{if } \lambda_i(\mathbf{X}) < 0 \\ -\infty & \text{otherwise,} \end{cases}$$

where $\mathbf{X}$ is an $r \times r$ matrix and $\lambda_i(\mathbf{X})$ is the i th eigenvalue of the matrix $\mathbf{X}$, we can solve, instead of (11), an unconstrained optimization with an augmented objective function:

$$\mathbf{D}_r = \arg \min_{\tilde{\mathbf{D}}_r \in \mathbb{R}^{r \times r}} \left\{ f_{\beta}(\tilde{\mathbf{D}}_r) := f(\tilde{\mathbf{D}}_r) - \beta g(\tilde{\mathbf{D}}_r) \right\}, \quad (21)$$

where $\beta > 0$ is the barrier parameter. The corresponding (dissipative) ROM (7) using $\mathbf{D}_r$ determined from (21) is referred to as the GP-OpInf ROM.

Remark 1. The online simulation time of the nonintrusive GP-OpInf ROM (7) is similar to that of the intrusive SP-G ROM for the same r , as both ROMs have the same structure yet may have different model parameters. If the Hamiltonian is nonlinear, one can use hyper-reduction techniques to accelerate the evaluation of the nonlinear term. For instance, the DEIM approach presented in [63] can be used to reduce the cost for evaluating the nonlinear Hamiltonian while preserving the structure.

Compared to SP-G ROM, the GP-OpInf ROM requires additional offline computation to infer $\mathbf{D}_r$. In the conservative case, we need to solve a $r \times r$ Lyapunov equation. Since the matrix $\mathbf{F}_r \mathbf{F}_r^T$ is usually dense, the Bartels–Stewart algorithm can be applied [64] with an associated computational complexity of $\mathcal{O}(r^3)$ flops. In the dissipative case, one can solve (21) by a gradient descent method as outlined in Appendix. The associated computational complexity is $\mathcal{O}(r^2 sp)$ per iteration, where sp represents the number of snapshots.

4. Error estimation

Previous work in [44,51] established an asymptotic (as $r \rightarrow n$) convergence of learned reduced operators to the associated projection-based reduced operators. However, this analysis can sometimes not explain the error in the solution of the ROM, as small errors in the operators may cause large solution errors, specifically for structured (e.g., gradient) systems. Here, we take a step further and analyze the approximate solution error of the inferred ROMs in the *preasymptotic* regime. Specifically, we estimate the *a priori* error of the GP-OpInf ROM approximation relative to the FOM (7). The derivation mainly follows the error estimation developed in [33] for analyzing the projection-based SP-G ROM. However, in addition to considering the error from POD truncation, there is a need to estimate the error caused by Operator Inference.

Given a mapping $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, the Lipschitz constant and the logarithmic Lipschitz constant of the mapping are defined as

$$C_{\text{Lip}}[f] := \sup_{\mathbf{u} \neq \mathbf{v}} \frac{\|f(\mathbf{u}) - f(\mathbf{v})\|}{\|\mathbf{u} - \mathbf{v}\|}, \quad \text{and} \quad C_{\log\text{-Lip}}[f] := \sup_{\mathbf{u} \neq \mathbf{v}} \frac{\langle \mathbf{u} - \mathbf{v}, f(\mathbf{u}) - f(\mathbf{v}) \rangle}{\|\mathbf{u} - \mathbf{v}\|^2},$$

where $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ for any positive integer n denotes the Euclidian inner product. The logarithmic Lipschitz constant could be negative, which was used in [65] to show that the error of reduced-order solution is uniformly bounded over the time interval of simulations.

Theorem 1. Let $\mathbf{y}(t)$ be the solution of the FOM (2) on the time interval $[0, T]$ and $\mathbf{y}_r(t)$ be the solution of the ROM (7) on the same interval. Suppose $\nabla_{\mathbf{y}} H(\mathbf{y})$ is Lipschitz continuous, then the ROM approximation error satisfies

$$\int_0^T \|\mathbf{y} - \Phi \mathbf{y}_r\|^2 dt \leq C(T) \left(\underbrace{\int_0^T \|\mathbf{y} - \Phi \Phi^T \mathbf{y}\|^2 dt}_{\text{projection error}} + \underbrace{\int_0^T \|\dot{\mathbf{y}} - \mathbf{D}_r \mathbf{D}_r^T [\mathbf{y}]\|^2 dt}_{\text{data error}} + \underbrace{\int_0^T \|\Phi^T \mathbf{D}_r [\mathbf{y}] - \mathbf{D}_r \Phi^T \nabla_{\mathbf{y}} H(\mathbf{y})\|^2 dt}_{\text{optimization error}} \right), \quad (22)$$

where $C(T) = \max\{1 + C_2^2, 2\} T \alpha(T)$, $\alpha(T) = 2 \int_0^T e^{2C_1(T-\tau)} d\tau$, and the constants $C_1 = C_{\log\text{-Lip}}[\Phi \mathbf{D}_r \Phi^T \nabla_{\mathbf{y}} H]$ and $C_2 = \|\Phi \mathbf{D}_r \Phi^T\| C_{\text{Lip}}[\nabla_{\mathbf{y}} H]$.

There are three terms in the error bound: the first term $\int_0^T \|\mathbf{y} - \Phi \Phi^T \mathbf{y}\|^2 dt$ is the **projection error** caused by projecting $\mathbf{y}(t; \mu)$ onto the subspace spanned by the POD basis Φ , that interprets the POD approximability and has been analyzed thoroughly in [66,67]; the second one $\int_0^T \|\dot{\mathbf{y}} - \mathbf{D}_r \mathbf{D}_r^T [\mathbf{y}]\|^2 dt$ is the **data error** caused by the approximation of time derivative snapshots using the finite difference method; the third one $\int_0^T \|\Phi^T \mathbf{D}_r [\mathbf{y}] - \mathbf{D}_r \Phi^T \nabla_{\mathbf{y}} H(\mathbf{y})\|^2 dt$ represents the **optimization error** generated by learning $\mathbf{D}_r$ from data.

Proof. Consider the ROM of a fixed dimension r , and define its approximate solution error by

$$\mathbf{e} := \mathbf{y} - \Phi \mathbf{y}_r.$$

It can be decomposed as $\mathbf{e} = \boldsymbol{\rho} + \boldsymbol{\theta}$ with $\boldsymbol{\rho} := \mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y}$ and $\boldsymbol{\theta} := \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y} - \boldsymbol{\Phi} \mathbf{y}_r$. Define $\boldsymbol{\zeta} := \dot{\mathbf{y}} - D_t[\mathbf{y}]$ and $\boldsymbol{\eta} := \boldsymbol{\Phi}^\top D_t[\mathbf{y}] - \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\mathbf{y})$. Consider the time derivative of $\boldsymbol{\theta}$

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \dot{\mathbf{y}} - \dot{\boldsymbol{\Phi}} \dot{\mathbf{y}}_r \\ &= \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \dot{\mathbf{y}} - \boldsymbol{\Phi} \mathbf{D}_r \nabla_{\mathbf{y}_r} H(\mathbf{y}_r) \\ &= \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \dot{\mathbf{y}} - \boldsymbol{\Phi} \mathbf{D}_r \nabla_{\mathbf{y}_r} H(\boldsymbol{\Phi} \mathbf{y}_r) \\ &= \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \dot{\mathbf{y}} - \boldsymbol{\Phi} \boldsymbol{\Phi}^\top D_t[\mathbf{y}] \\ &\quad + \boldsymbol{\Phi} \boldsymbol{\Phi}^\top D_t[\mathbf{y}] - \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\mathbf{y}) \\ &\quad + \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\mathbf{y}) - \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y}) \\ &\quad + \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y}) - \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \mathbf{y}_r). \end{aligned} \quad (23)$$

Note that

$$\frac{d}{dt} \|\boldsymbol{\theta}\| = \frac{1}{2\|\boldsymbol{\theta}\|} \frac{d}{dt} \|\boldsymbol{\theta}\|^2 = \frac{1}{\|\boldsymbol{\theta}\|} \langle \boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \rangle. \quad (24)$$

Taking the inner product of Eq. (23) with $\boldsymbol{\theta}$, the terms on the right-hand side of (23) are

$$\langle \boldsymbol{\theta}, \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \dot{\mathbf{y}} - \boldsymbol{\Phi} \boldsymbol{\Phi}^\top D_t[\mathbf{y}] \rangle = \langle \boldsymbol{\theta}, \boldsymbol{\Phi} \boldsymbol{\Phi}^\top (\dot{\mathbf{y}} - D_t[\mathbf{y}]) \rangle \leq \|\boldsymbol{\theta}\| \|\boldsymbol{\zeta}\|, \quad (25)$$

$$\langle \boldsymbol{\theta}, \boldsymbol{\Phi} \boldsymbol{\Phi}^\top D_t[\mathbf{y}] - \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H(\mathbf{y}) \rangle = \langle \boldsymbol{\theta}, \boldsymbol{\Phi} \boldsymbol{\eta} \rangle \leq \|\boldsymbol{\theta}\| \|\boldsymbol{\eta}\|, \quad (26)$$

$$\langle \boldsymbol{\theta}, \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top (\nabla_{\mathbf{y}} H(\mathbf{y}) - \nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y})) \rangle \leq \|\boldsymbol{\theta}\| \|\boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top\| C_{\text{Lip}}[\nabla_{\mathbf{y}} H] \|\boldsymbol{\rho}\|, \quad (27)$$

$$\langle \boldsymbol{\theta}, \boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top (\nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \boldsymbol{\Phi}^\top \mathbf{y}) - \nabla_{\mathbf{y}} H(\boldsymbol{\Phi} \mathbf{y}_r)) \rangle \leq C_{\log-\text{Lip}}[\boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H] \|\boldsymbol{\theta}\|^2. \quad (28)$$

Let $C_1 = C_{\log-\text{Lip}}[\boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top \nabla_{\mathbf{y}} H]$, $C_2 = \|\boldsymbol{\Phi} \mathbf{D}_r \boldsymbol{\Phi}^\top\| C_{\text{Lip}}[\nabla_{\mathbf{y}} H]$, and combine (24) with (23) and (25)–(28), we have

$$\frac{d}{dt} \|\boldsymbol{\theta}\| \leq C_1 \|\boldsymbol{\theta}\| + C_2 \|\boldsymbol{\rho}\| + \|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|.$$

Applying the classical differential version of Gronwall lemma (see, for example, Appendix B in [68]) over the interval $[0, t]$, for $t \leq T$, and using the fact that $\boldsymbol{\theta}(0) = 0$, we get

$$\|\boldsymbol{\theta}(t)\| \leq \int_0^t e^{C_1(t-\tau)} (C_2 \|\boldsymbol{\rho}\| + \|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|) d\tau.$$

Applying the Cauchy–Schwarz inequality on the RHS and squaring both sides, we have

$$\begin{aligned} \|\boldsymbol{\theta}(t)\|^2 &\leq \int_0^t e^{2C_1(t-\tau)} d\tau \int_0^t [C_2 \|\boldsymbol{\rho}\| + (\|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|)]^2 d\tau \\ &\leq \alpha(t) \left[\int_0^T C_2^2 \|\boldsymbol{\rho}\|^2 dt + \int_0^T (\|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|)^2 dt \right] \\ &\leq \alpha(T) \left[C_2^2 \int_0^T \|\boldsymbol{\rho}\|^2 dt + \int_0^T (\|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|)^2 dt \right], \end{aligned}$$

where $\alpha(t) = 2 \int_0^t e^{2C_1(t-\tau)} d\tau$. Hence,

$$\int_0^T \|\boldsymbol{\theta}(t)\|^2 dt \leq T \alpha(T) \left[C_2^2 \int_0^T \|\boldsymbol{\rho}\|^2 dt + \int_0^T (\|\boldsymbol{\zeta}\| + \|\boldsymbol{\eta}\|)^2 dt \right].$$

This, together with the triangular inequality and orthogonality of $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$, yields

$$\int_0^T \|\mathbf{e}(t)\|^2 dt \leq (1 + C_2^2 T \alpha(T)) \int_0^T \|\boldsymbol{\rho}\|^2 dt + 2T \alpha(T) \int_0^T (\|\boldsymbol{\zeta}\|^2 + \|\boldsymbol{\eta}\|^2) dt,$$

which proves the theorem. $\square$

Remark 2. In comparison to the asymptotic error estimation of the *SP-G* ROM in [33], the *GP-OpInf* ROM is different in that it only involves the projection error of the state, whereas the error of projecting the gradient of the Hamiltonian onto the reduced basis space is included in the *SP-G*. Additionally, it incorporates the data error and optimization error. We will numerically investigate these errors in the next section.

Remark 3. The FOM (2) and ROM (7) are continuous in time. In numerical simulations, time-marching schemes with identical or different time step sizes are applied. Consequently, the time discretization errors appear in the upper bound of the ROM approximation error. This time discretization error, along with the data error, can be controlled by choosing a sufficiently small time step size.

Remark 4. It is worth mentioning that, in addition to (8), one could evaluate the gradient data on the projected snapshots, e.g.,

$$\mathbf{F} \equiv \mathbf{F}(\mu) := [\nabla_{\mathbf{y}} H(\Phi \Phi^T \mathbf{y}(t^0; \mu); \mu), \nabla_{\mathbf{y}} H(\Phi \Phi^T \mathbf{y}(t^1; \mu); \mu), \dots, \nabla_{\mathbf{y}} H(\Phi \Phi^T \mathbf{y}(t^s; \mu); \mu)] \quad (29)$$

when inferring the reduced operator. Using it does not alter the proposed *GP-OpInf* framework, yet for the error estimation, Eqs. (26) and (27) can be grouped together. This results in a similar error bound to (22), with the only difference being that the optimization error term becomes $\int_0^T \|\Phi^T \mathcal{D}_t[\mathbf{y}] - \mathbf{D}_r \Phi^T \nabla_{\mathbf{y}} H(\Phi \Phi^T \mathbf{y})\|^2$.

5. Numerical experiments

In this section, we provide many numerical experiments to demonstrate the effectiveness of the proposed method. In Section 5.1 we first define the error measures used throughout to assess quality and accuracy of a learned ROM. Section 5.2 contains conservative dynamical systems and Section 5.3 evaluates dissipative dynamical systems.

5.1. Error measures

Given a set of snapshots, the performance of a ROM depends on its dimension and the numerical schemes used for simulations. To test the performance of the various *GP-OpInf* ROMs and to illustrate the error analysis, for a given time discretization, we compute the (squared) *ROM approximation error* as a function of r :

$$\mathcal{E}(r) := \frac{T\Delta A}{N} \sum_{j=1}^N \|\mathbf{y}(t^j; \mu) - \Phi \mathbf{y}_r(t^j; \mu)\|^2, \quad (30)$$

where $\|\cdot\|$ represents the vector 2-norm, $\mathbf{y}(t^j; \mu)$ and $\mathbf{y}_r(t^j; \mu)$ are the full-order and reduced-order solutions at t^j , respectively, for $j = 1, \dots, N$, $\Phi = [\phi_1, \phi_2, \dots, \phi_r]$ is the reduced basis matrix, and ΔA represents the associated element size in a uniform grid. Following the error bound (22), we additionally compute the (squared) *projection error* and (squared) *optimization error*:

$$\mathcal{E}_{\text{proj}}(r) := \frac{T\Delta A}{N} \sum_{j=1}^N \|\mathbf{y}(t^j; \mu) - \Phi \Phi^T \mathbf{y}(t^j; \mu)\|^2, \quad (31)$$

$$\mathcal{E}_{\text{opt}}(r) := \frac{T\Delta A}{N} \sum_{j=1}^N \|\Phi^T \mathcal{D}_t[\mathbf{y}(t^j; \mu)] - \mathbf{D}_r \Phi^T \nabla_{\mathbf{y}} H(\mathbf{y}(t^j; \mu); \mu)\|^2. \quad (32)$$

The norms are evaluated by the same quadrature rule such as the composite rectangle method. In the following numerical experiments, we frequently use two matrices in discrete schemes:

$$\mathbf{L}_n = \begin{pmatrix} -2 & 1 & 0 & \dots & 1 \\ 1 & -2 & 1 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \dots & 1 & -2 & 1 \\ 1 & \dots & 0 & 1 & -2 \end{pmatrix}_{n \times n} \quad \text{and} \quad \mathbf{S}_n = \begin{pmatrix} 0 & 1 & 0 & \dots & -1 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 0 & 1 \\ 1 & 0 & \dots & -1 & 0 \end{pmatrix}_{n \times n};$$

and we denote the spatial domain of an equation by Ω and the time domain by I . Specifically, in the FOM simulations the time interval $I = [0, T_{\text{FOM}}]$, while in the ROM simulations $I = [0, T_{\text{ROM}}]$.

For our numerical tests, the Lyapunov equation is solved by using the `lyap` function when using Matlab, or the `linalg.solve_continuous_lyapunov` function when using python/scipy. Moreover, we use the licensed MOSEK [69] optimization software developed by CVXPY [70,71] to solve the semi-definite programming problem (21). If this software is unavailable, one can obtain the optimal solution through gradient descent, as discussed in Appendix.

5.2. Conservative PDEs

We consider the wave equation, the Korteweg–de Vries (KdV) equation and the linear elastic equations as a test-bed for investigating the *GP-OpInf* ROM when approximating conservative systems. After a spatial discretization of the PDEs, the former yields a canonical Hamiltonian system while the latter results in a non-canonical Hamiltonian system. We note that both equations have also been considered in [50,51] with similar configurations.

5.2.1. Parameterized wave equation

Consider the one-dimensional linear wave equation with a constant wave speed c :

$$u_{tt} = c^2 u_{xx}, \quad (x, t) \in \Omega \times I,$$

where $u = u(x, t; \mu)$ is the parameter-dependent solution and $\mu \in P$ a parameter appearing in the initial condition, defined below alongside the boundary and initial conditions. The wave equation can be recast to the *canonical* Hamiltonian formulation (see, e.g. [22]) with the Hamiltonian $H(u, v) = \int_{\Omega} (\frac{1}{2} v^2 + \frac{c^2}{2} u_x^2) dx$, so that

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\delta H}{\delta u} \\ \frac{\delta H}{\delta v} \end{bmatrix},$$

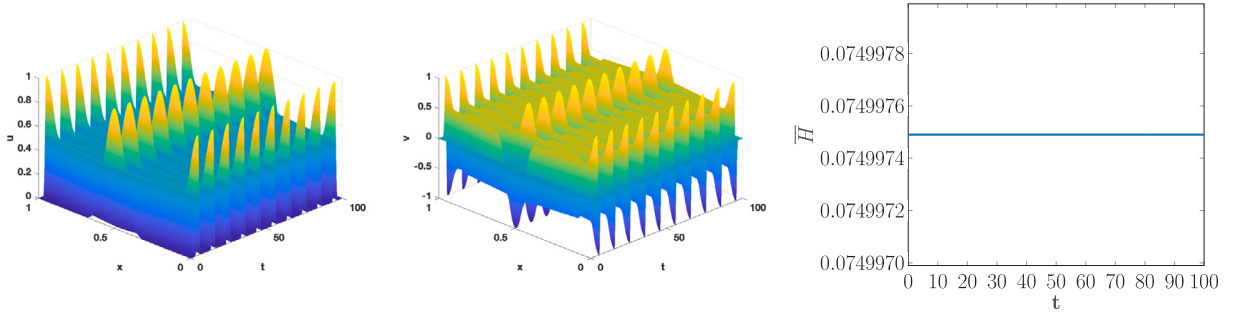


Fig. 1. (Wave Equation) Full-order model simulation when $T_{\text{FOM}} = 100$: time evolution of $\mathbf{u}$ (left), $\mathbf{v}$ (middle), and energy $\bar{H}(t)$ (right).

which has the symplectic (gradient) structure.

Taking a uniform partition in the spatial domain with the mesh size Δx and defining a consistent discrete Hamiltonian $\bar{H}(t) := \Delta x H(\mathbf{u}, \mathbf{v})$ with $H(\mathbf{u}, \mathbf{v}) = \sum_j [\frac{1}{2} \mathbf{v}_j^2 + \frac{c^2}{2(\Delta x)^2} (\mathbf{u}_{j+1} - \mathbf{u}_j)^2]$ leads to the Hamiltonian system of ODEs

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_n \\ -\mathbf{I}_n & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{A}\mathbf{u} \\ \mathbf{v} \end{bmatrix}. \quad (33)$$

Here, n is the number of degrees of freedom, $\mathbf{I}_n$ the $n \times n$ identity matrix, and $\mathbf{A}$ is a discretization of the scaled, one-dimensional second order differential operator. The above system is of the form (2) with $\mathbf{y} = [\mathbf{u}^T, \mathbf{v}^T]^T$ and $\mathbf{D}$ is the $2n \times 2n$ skew-symmetric block matrix from above. This implies the system is conservative and the Hamiltonian is a constant function.

Computational setting. Consider the case in which $c = 0.1$ and $\Omega = [0, 1]$. The boundary condition is set to be periodic and the initial condition satisfies $u_0(x; \mu) = h(s(x; \mu))$ and $v_0(x; \mu) = 0$, in which $h(s)$ is a cubic spline function defined by

$$h(s) = \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & \text{if } 0 \leq s \leq 1, \\ \frac{1}{4}(2-s)^3 & \text{if } 1 < s \leq 2, \\ 0 & \text{if } s > 2, \end{cases} \quad \text{and} \quad s(x; \mu) = \mu \left| x - \frac{1}{2} \right|$$

with μ a parameter that can take values from $P = [5, 15]$. In the FOM, the finite difference method is used for spatial discretization and

$$\mathbf{A} = \frac{c^2}{\Delta x^2} \mathbf{L}_n.$$

For time integration, the midpoint rule is applied with the time step Δt . The resulting linear system is solved by the built-in direct solver in MATLAB. For $T_{\text{FOM}} = 100$, $\Delta t = 1 \times 10^{-3}$ and $\mu = 10$, the time evolution of the full-order states and the energy $\bar{H}(t)$ are plotted in Fig. 1. Note that the Hamiltonian energy is preserved as $\bar{H}(t) \approx 7.5 \times 10^{-2}$.

ROM construction. We assemble the snapshot matrices $\mathbf{U}$ and $\mathbf{V}$ from full-order simulations and generate the basis matrices $\Phi_u \in \mathbb{R}^{n \times r_1}$ and $\Phi_v \in \mathbb{R}^{n \times r_2}$ that contain, respectively, the r_1 and r_2 leading left singular vectors of $\mathbf{U}$ and $\mathbf{V}$. We define the POD basis $\Phi = \begin{bmatrix} \Phi_u & \mathbf{0} \\ \mathbf{0} & \Phi_v \end{bmatrix}$ and obtain the projected data $\mathbf{Y}_r = \Phi^T \mathbf{Y} = \Phi^T [\mathbf{U}^T, \mathbf{V}^T]^T$ as well as the associated gradient of the Hamiltonian and the time derivative data matrices (using D_t defined in (9)). For simplicity, we choose $r_1 = r_2 = r$ in the following tests. However, different basis sizes can be chosen to better represent data. We note that other types of reduced basis can be used, such as the cotangent lift [38]. We refer the interested readers to [51] for a numerical investigation on different types of reduced bases. Next, we provide several comparisons of the ROMs generated by *GP-OpInf*, *GP-OpInf-V* and *GP-OpInf-P* and investigate their performance over $[0, T_{\text{ROM}}]$.

Test 1. Effects of regularization. We first consider the non-parametric case in which $\mu = 10$ is fixed. The mesh size for the FOM is $\Delta x = 10^{-3}$ (correspondingly, $n = 1000$) and the same time step $\Delta t = 1 \times 10^{-3}$ is used in both the FOM and the ROMs. We use *GP-OpInf*, *GP-OpInf-V* and *GP-OpInf-P* to create ROMs of dimension $2r$ (r bases for u and r bases for v) and let r vary from $r = 5$ to $r = 200$.

We first select $T_{\text{FOM}} = 5$, $T_{\text{ROM}} = 5$ and check whether the inferred $\mathbf{D}_r$ possesses the desired skew-symmetric structure by plotting the maximum magnitude of $|\mathbf{D}_r + \mathbf{D}_r^T|$ in Fig. 2 (left). The inferred $\mathbf{D}_r$ from *GP-OpInf-P* (with $\alpha = 2$ and $c_0 = 10^{-13}$ in the definition of ϵ (17)) achieves better anti-symmetric structure than *GP-OpInf-V* when $r > 30$, but both methods are not producing satisfactory results and are far off from being skew-symmetric. In contrast, the proposed *GP-OpInf* produces exact anti-symmetric matrices up to machine precision. The associated ROM approximation errors $\mathcal{E}$ are plotted in Fig. 2 (right). For the *GP-OpInf-V* method, the error grows unbounded (no entry indicates a NaN) when $r > 40$. The *GP-OpInf-P* method yields better performance as it provides stable numerical solutions at almost all of the considered r -values, yet it becomes inaccurate at $r = 105$ and $r = 155$. In contrast, the *GP-OpInf* method is able to obtain stable numerical approximations for all the r -values.

We now consider the case of $T_{\text{FOM}} = 10$ and $T_{\text{ROM}} = 10$. The associated maximum magnitude of $|\mathbf{D}_r + \mathbf{D}_r^T|$ and the approximation errors of the ROMs based on three inferred $\mathbf{D}_r$ are plotted in Fig. 3. From the results, we find that all three methods learn $\mathbf{D}_r$ with

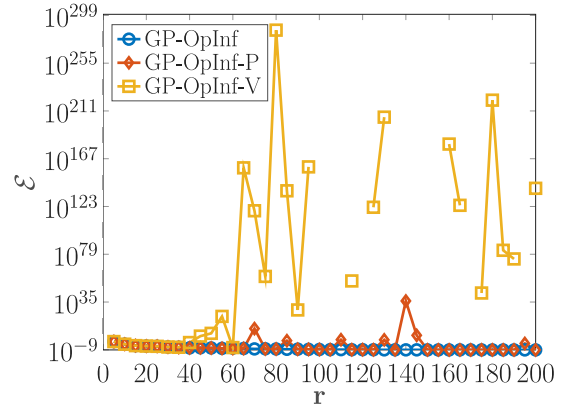
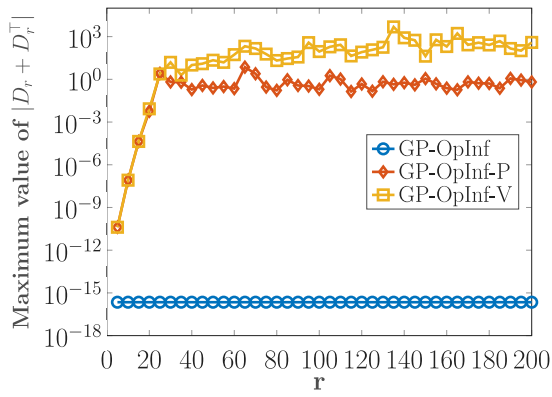


Fig. 2. (Wave Equation) ROMs of 2r-dimensions when $T_{\text{FOM}} = 5$ and $T_{\text{ROM}} = 5$: (left) maximum magnitude of $|\mathbf{D}_r + \mathbf{D}_r^T|$ for the inferred $\mathbf{D}_r$, when r varies; (right) ROM approximation error ε from (30) when r varies from 5 to 200 in increments of five. The inserted plots cover the portion of r between 75 and 175.

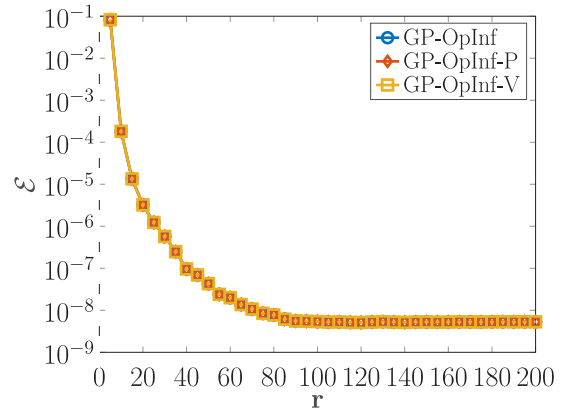
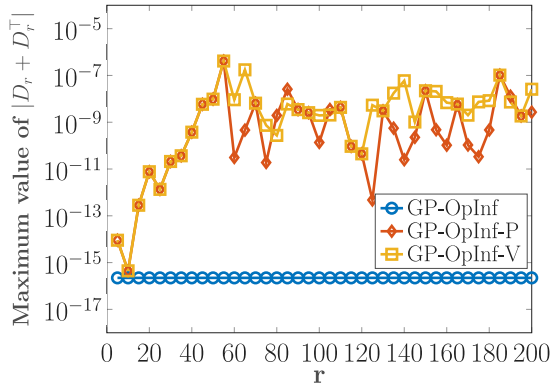


Fig. 3. (Wave Equation) ROMs of 2r-dimensions when $T_{\text{FOM}} = 10$ and $T_{\text{ROM}} = 10$: (left) maximum magnitude of $|\mathbf{D}_r + \mathbf{D}_r^T|$ for the inferred $\mathbf{D}_r$, when r varies; (right) numerical error ε from (30) when r varies from 5 to 200 in increments of five.

Table 1

Smallest eigenvalues of $\mathbf{F}_r \mathbf{F}_r^T$ at several small r values.

Case	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$	$r = 30$
$T_{\text{FOM}} = T_{\text{ROM}} = 5$	1.36e-07	8.16e-10	3.55e-12	9.57e-15	-6.97e-13	-6.41e-12
$T_{\text{FOM}} = T_{\text{ROM}} = 10$	1.16e-04	6.74e-09	2.48e-10	1.98e-11	5.55e-12	1.26e-12

skew-symmetric structure, however, with noticeable differences. The maximum magnitude of $|\mathbf{D}_r + \mathbf{D}_r^T|$ is about 10^{-6} in *GP-OpInf-V* and *GP-OpInf-P*. Once again *GP-OpInf* is the most accurate yielding machine precision, 10^{-16} , for the skew-symmetry test. From Fig. 3 (right), we observe that the three ROMs yield similar numerical accuracy in this case.

The different behavior of *GP-OpInf-V* in the second case is mainly a result of the inferred $\mathbf{D}_r$ having a better structure. In Table 1, the minimum eigenvalues of $\mathbf{F}_r \mathbf{F}_r^T$ are listed at several small r values. We observe that small negative eigenvalues appear when $r = 25$ and $r = 30$ which result in the direct solution to the Lyapunov equation, $\mathbf{D}_r$, not being skew-symmetric.

In sum, we find that the *GP-OpInf* approach computes $\mathbf{D}_r$ with skew-symmetric structure up to machine precision in both tests, which is markedly better than *GP-OpInf-V* and *GP-OpInf-P*. While more training data improved the *GP-OpInf-V* and *GP-OpInf-P* structural properties, noticeable (10 orders of magnitude) differences remained. The resulting *GP-OpInf* ROMs are more accurate, which is especially evident in the first test case where the other two approaches failed. In the following tests, we thus only focus on the *GP-OpInf* approach and investigate it in more detail.

Test 2. Illustration of error estimation. Here, we investigate the error estimation for the *GP-OpInf* ROM for the same non-parametric case with $\mu = 10$, and compare the approximation error with the intrusive structure-preserving ROM (*SP-G* ROM) of the same dimension.

We choose $\Delta x = \Delta t = 2 \times 10^{-4}$ so that the discretization errors are negligible relative to the POD projection error and the optimization error. We set $T_{\text{FOM}} = T_{\text{ROM}} = 10$ and vary the dimension r of the ROM. The three ROM errors (30)–(32) are plotted in

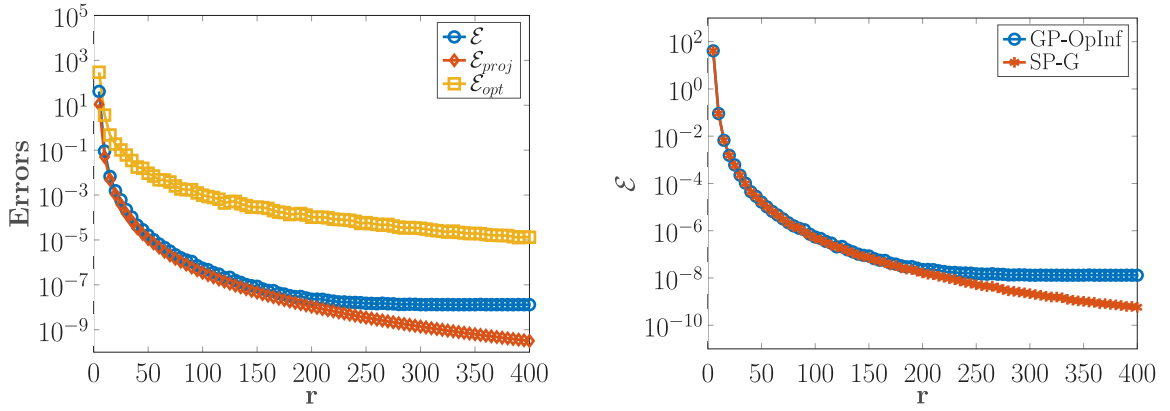


Fig. 4. (Wave Equation) Numerical errors of the 2D-dimensional *GP-OpInf* ROM when $T_{\text{FOM}} = T_{\text{ROM}} = 10$ and $\Delta x = \Delta t = 2 \times 10^{-4}$; (left) the ROM approximation error (30) together with POD projection error (31) and optimization error (32); (right) comparison of the ROM approximation error for the *GP-OpInf* and *SP-G* ROM.

Fig. 4 (left), which shows that the error decays monotonically as r increases. When $r > 200$, the ROM approximation error saturates, although the POD projection error is still decreasing, which indicates the error is dominated by the saturated optimization error and the fixed time discretization error. We compare the *GP-OpInf* ROM approximation error with the *SP-G* ROM in **Fig. 4** (right). The *SP-G* approximation error decreases in a tendency similar to the POD projection error in **Fig. 4** (left), and *GP-OpInf* achieves similar numerical accuracy to *SP-G* when r is less than 200. However, as r increases, the *GP-OpInf* ROM approximation accuracy stays around 10^{-8} . Nevertheless, such large r values lead to impractically expensive ROMs, and are of little practical use.

In this numerical test, $T_{\text{FOM}} = T_{\text{ROM}}$, as assumed when deriving the error estimation. Since Operator Inference is a data-driven technique, the optimization error and POD projection error depend on the given snapshot data. Consequently, there is no general accuracy guarantee for the ROM simulations beyond the time duration on which the snapshots are gathered. Nonetheless, if the snapshot data represent the dynamical system behavior well, it is possible to attain accurate ROM predictions. Nevertheless, the predictive capabilities of the ROMs outside the training intervals are enhanced by enforcing the correct structure in the inferred ROMs. We investigate predictions for longer time intervals in the next test.

Test 3. Long-term predictive capabilities of the *GP-OpInf* ROM. We set $\Delta x = \Delta t = 10^{-3}$ and fix $\mu = 10$ and generate snapshots from the FOM with the final time $T_{\text{FOM}} = 10$. However, we simulate the *GP-OpInf* ROM on a much longer interval with $T_{\text{ROM}} = 100$ to demonstrate the long-term predictive capabilities of structure-preserving ROMs. The time evolution of the ROM approximation errors, using the FOM solution as the benchmark, at $r = 10, 20, 40$ and 80 are plotted in **Fig. 5** (left). The associated approximate Hamiltonian values are plotted in **Fig. 5** (right), where the dashed line indicates the end of the training time interval. We observe that for $r = 10$ and 20 , the errors stay at the same level over time, while for $r = 40$ and 80 , the errors increase gradually as t increases. Meanwhile, since the appropriate (gradient) structure is captured in the inferred *GP-OpInf* ROM, the approximate Hamiltonian is conservative, which approaches the benchmark value (from the FOM) as r increases. We remark that these Hamiltonian approximations can be improved by choosing $\bar{y} = y^0$ and using a POD basis generated from the shifted snapshots as introduced in [33].

Test 4. Parametric predictions away from training data. We parameterize the initial condition $u_0(x; \mu) = h(s(x; \mu))$ and let $\mu \in [5, 15]$. To generate snapshots, 11 uniformly distributed training samples are collected from the parameter interval and the FOM is simulated at all samples with $\Delta x = \Delta t = 10^{-3}$ and $T_{\text{FOM}} = 10$. Then, we compute the POD basis from the collected snapshots, learn the *GP-OpInf* ROM, from which we predict solutions at different test parameters over the same time interval ($T_{\text{ROM}} = T_{\text{FOM}}$). We consider three randomly selected test parameters, $\mu = 6.7, 9.5$ and 14.1 .

Fig. 6 (top left) compares the approximation errors of the *SP-G* ROM and the *GP-OpInf*. The figure shows that at all test parameters μ , the error of the *GP-OpInf* ROM is close to that of *SP-G* ROM when $r \leq 80$. The error, however, saturates when r becomes bigger and, thus, turns larger than that of the *SP-G* ROM. Nevertheless, such large basis dimensions produce inefficient ROMs and are of little practical value. We choose the more realistic $r = 40$ case and plot the benchmark FOM solution and the ROM solution in **Fig. 6** (top right and bottom); only a few snapshots are shown for illustration. In each case, the *GP-OpInf* ROM achieves accurate approximations close to the FOM solution.

We next consider a more challenging parametric problem in which both the wave speed and the initial condition are parameterized: $c(\mu) = 0.1\mu$ and $u_0(x; \mu) = (1 + \mu)h(s(x; \mu))$, where $s(x; \mu) = \frac{20}{\mu} |x - \frac{1}{2} - \frac{\mu}{10}|$ and $\mu \in P = [5/12, 5/6]$. Compared to the preceding test, the support of the initial pulse is thinner and its location, magnitude, and traveling speed vary with the parameter μ . We infer the *GP-OpInf* ROM based on the training set of 11 uniformly distributed parameter samples in P . For three randomly chosen testing samples outside the training set, $\mu = 0.43, 0.60$ and 0.82 , we plot the approximation errors of the *GP-OpInf* ROM in **Fig. 7** (top left) together with those of the *SP-G* ROM. Several time snapshots of the *GP-OpInf* with $r = 55$ and the FOM solutions are presented in **Fig. 7** (top right and bottom), demonstrating the good performance of the *GP-OpInf* ROM in this case as well.

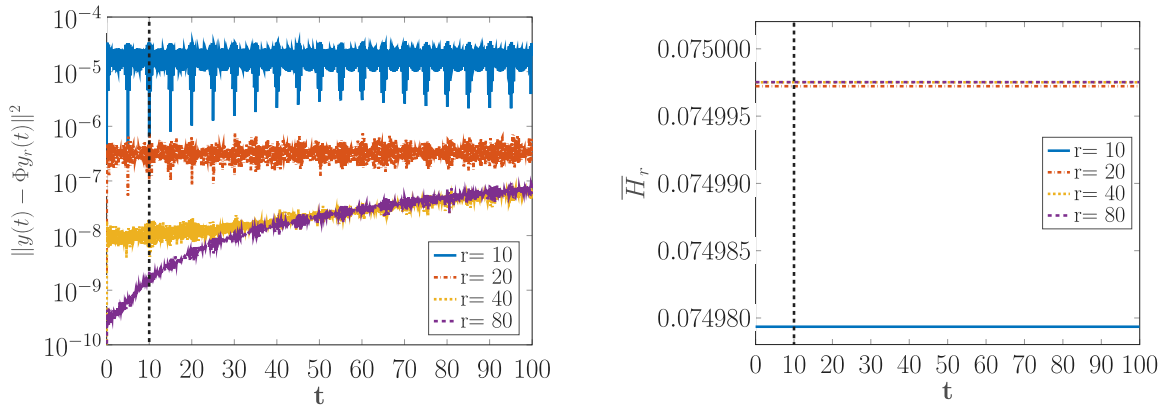


Fig. 5. (Wave Equation) Numerical accuracy of the 2r-dimensional GP-OpInf: (left) time evolution of the summand of the ROM approximation error (30); (right) time evolution of the reduced-order approximate Hamiltonian energy. The vertical dashed line indicates the end of the training interval.

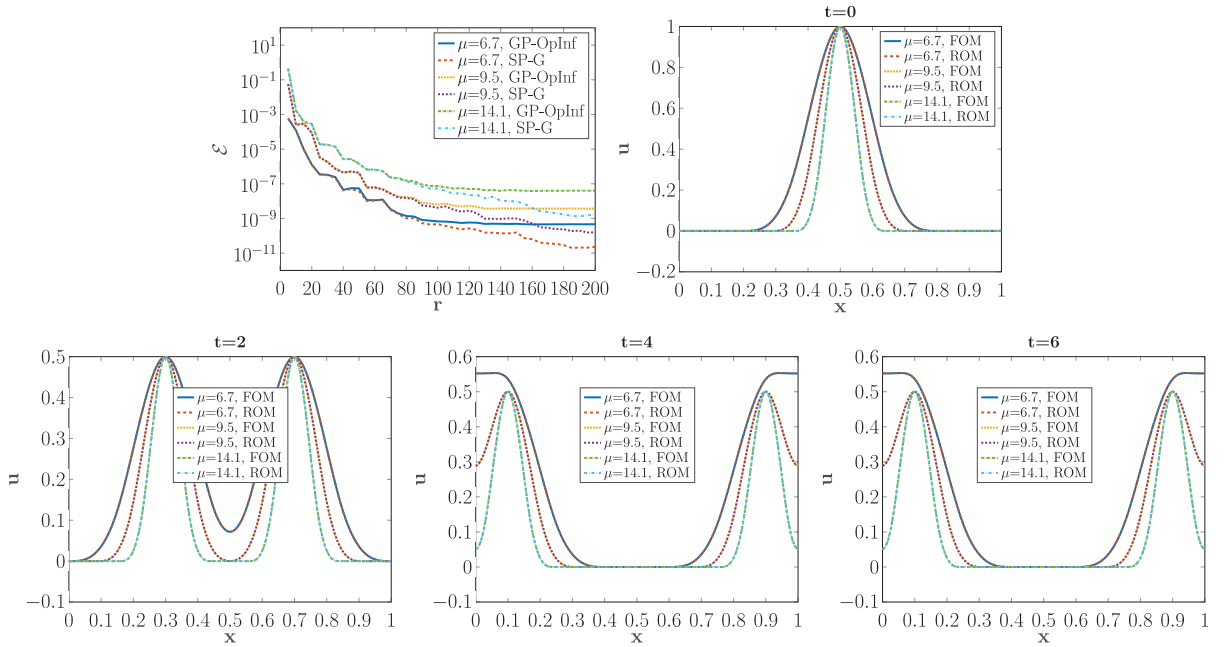


Fig. 6. (Wave Equation) Performance at three test parameters: (top left) comparison of the ROM approximation errors (30) for the GP-OpInf with the SP-G ROM of the same dimension 2r; (top right and bottom) the GP-OpInf ROM solution and the FOM solution at $t = 0, 2, 4, 6$.

5.2.2. Parameterized Korteweg–de Vries (KdV) equation

Next, we consider the one-dimensional KdV equation

$$u_t = \alpha u u_x + \nu u_{xxx}, \quad (x, t) \in \Omega \times I,$$

where $u = u(x, t; \mu)$ is the parameter-dependent solution and $\mu \in P$ a parameter appearing in the initial condition, defined below alongside the boundary and initial conditions. The KdV equation can be rewritten as a non-canonical Hamiltonian system (see, e.g., [22]):

$$u_t = S \frac{\delta H}{\delta u},$$

where $S = \partial_x$ denotes the first-order derivative operator with respect to space, and the Hamiltonian function $\mathcal{H}(u) = \int_{\Omega} \left(\frac{\alpha}{6} u^3 - \frac{\nu}{2} u_x^2 \right) dx$. After taking a spatial discretization with a uniform mesh size Δx and defining a consistent discrete Hamiltonian $\bar{H}(t) := \Delta x H(\mathbf{u})$ with $H(\mathbf{u}) = \sum_j \left[\frac{\alpha}{6} u_j^3 - \frac{\nu}{2} (u_{j+1} - u_j)^2 \right]$, we obtain the semi-discrete Hamiltonian system

$$\dot{\mathbf{u}} = \mathbf{A} \left(\frac{\alpha}{2} \mathbf{u}^2 + \nu \mathbf{B} \mathbf{u} \right), \quad (34)$$

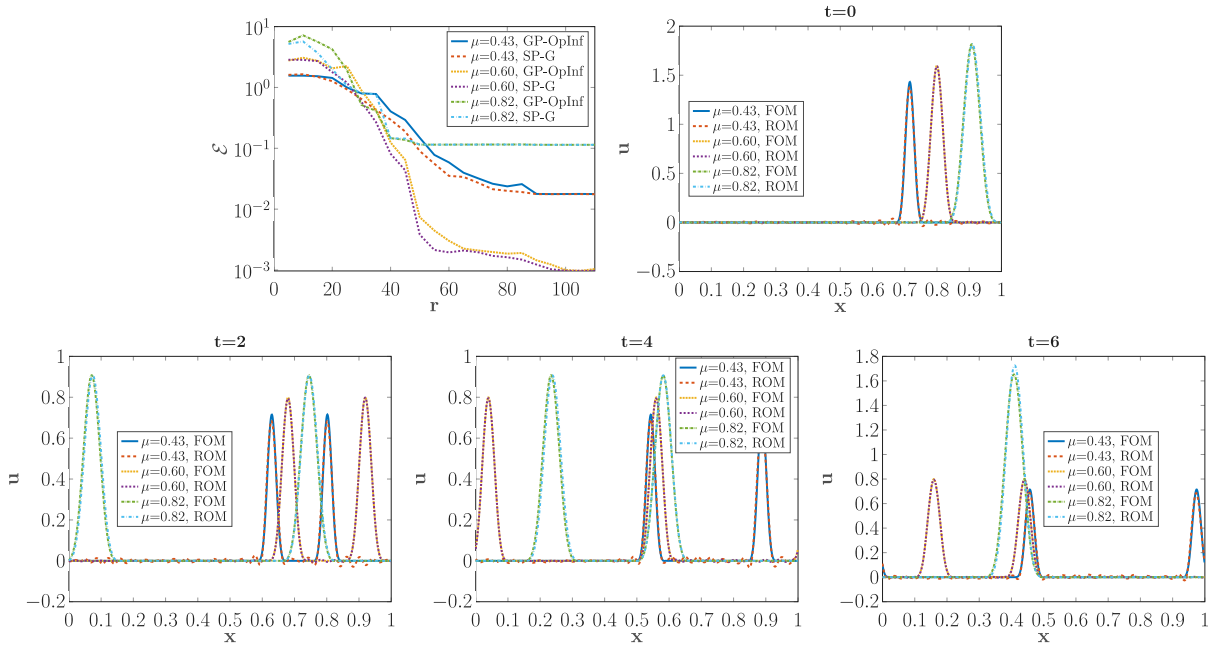


Fig. 7. (Wave Equation) Performance at three test parameters: (top left) comparison of the ROM approximation errors (30) for the *GP-OpInf* with the *SP-G* ROM of the same dimension $2r$; (top right and bottom) the *GP-OpInf* ROM solution and the FOM solution at $t = 0, 2, 4, 6$.

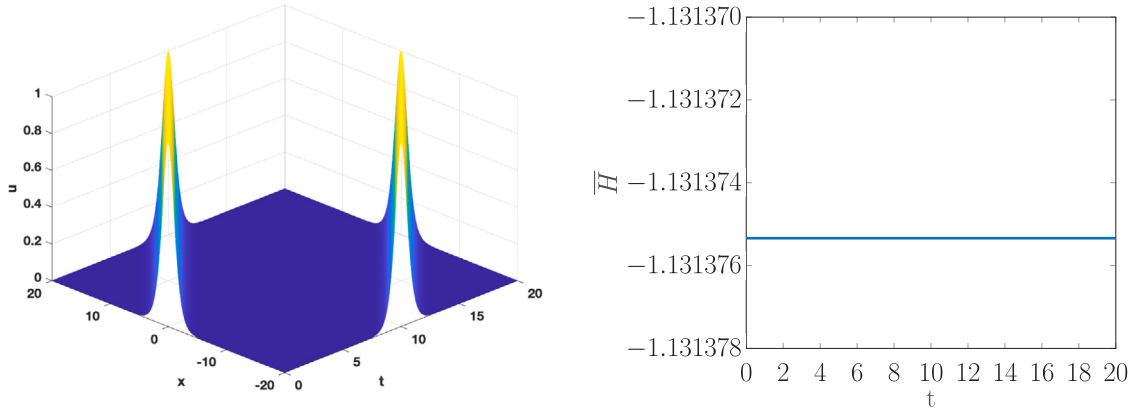


Fig. 8. (KdV Equation) Full-order model simulation up to final time $T_{\text{FOM}} = 20$: time evolution of $\mathbf{u}$ (left) and approximate Hamiltonian $\bar{H}(t)$ (right).

where $\mathbf{A}$ and $\mathbf{B}$ are the matrices associated to the discretization of the skew-adjoint operator S and the second-order derivative by central differences, respectively. Since $\mathbf{A}$ is skew-symmetric, this dynamical system conserves the discrete Hamiltonian $\bar{H}(t)$.

Computational setting. We set $\Omega = [-20, 20]$, $\alpha = -6$, $\nu = -1$, and consider periodic boundary conditions $u(-20, t; \mu) = u(20, t; \mu)$ for any $t \in I$ and we parameterize the initial condition $u_0(x; \mu) = \text{sech}^2\left(\frac{x}{\mu}\right)$, for any $\mu \in P = [1, 5]$. In the FOM (34), we have

$$\mathbf{A} = \frac{1}{2\Delta x} \mathbf{S}_n, \quad \text{and} \quad \mathbf{B} = \frac{1}{4\Delta x^2} \mathbf{L}_n.$$

The mesh size $\Delta x = 0.01$ (correspondingly, $n = 4000$) is used in all simulations. For time integration, we use the average vector field (AVF) method [22] together with a Picard iteration to solve the nonlinear system.

When $T_{\text{FOM}} = 20$, $\Delta t = 0.01$ and $\mu = \sqrt{2}$, the time evolution of the full-order state and approximate Hamiltonian $\bar{H}(t)$ are shown in Fig. 8. We observe that the system is conservative, with the Hamiltonian staying around $\bar{H}(t) \approx -1.13$ during the simulation.

Test 1. Illustration of error estimation. We demonstrate the estimation of the *GP-OpInf* ROM approximation error in a non-parametric case with $\mu = \sqrt{2}$, and compare the approximation error with the *SP-G* ROM of the same dimension. To verify the error estimation, we choose a small time step $\Delta t = 2.5 \times 10^{-3}$, let $T_{\text{FOM}} = T_{\text{ROM}} = 20$ and vary the dimension of the ROM. Fig. 9 (left) shows the

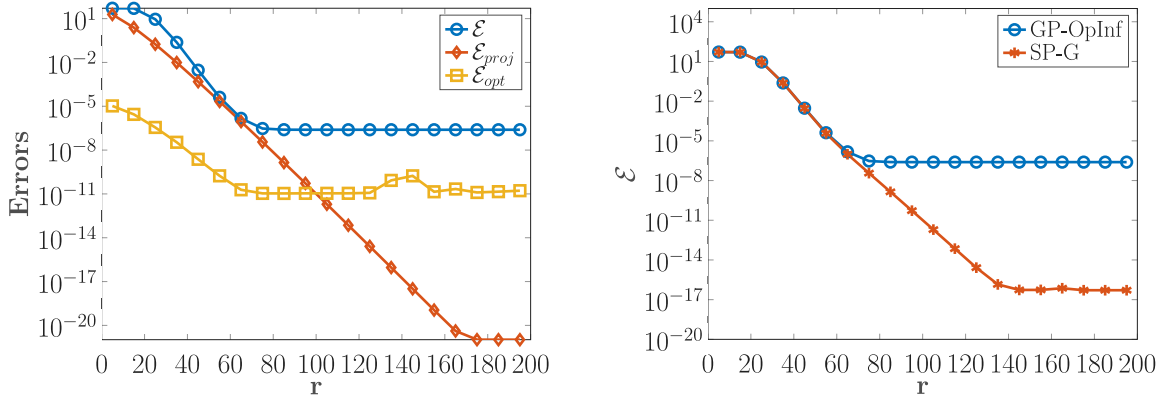


Fig. 9. (KdV Equation) Numerical errors of the r -dimensional GP-OpInf ROM when $T_{FOM} = T_{ROM} = 20$ and $\Delta t = 2.5 \times 10^{-3}$: (left) the ROM approximation error (30) together with POD projection error (31) and optimization error (32); (right) comparison of the ROM approximation error of the GP-OpInf ROM with the SP-G ROM.

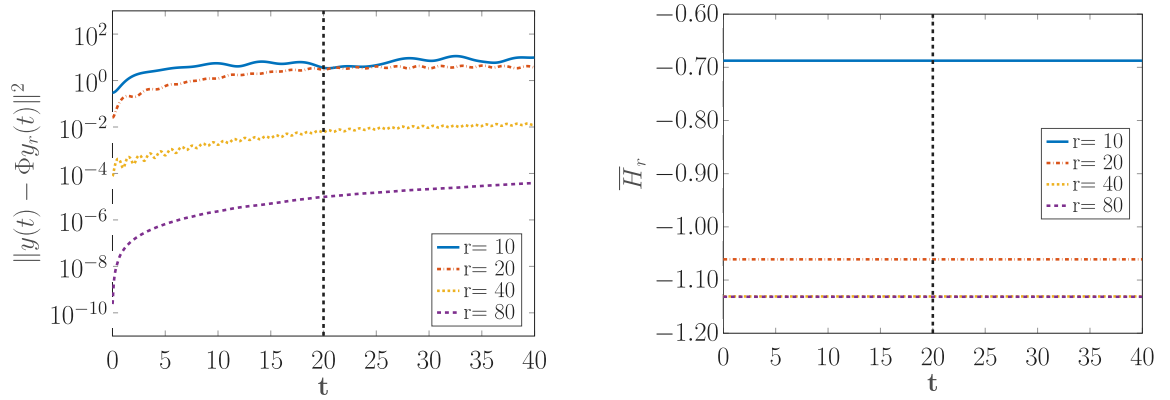


Fig. 10. (KdV Equation) Numerical accuracy of the r -dimensional GP-OpInf ROM: (left) time evolution of the summand of the ROM approximation error (30); (right) time evolution of the values of reduced-order approximate Hamiltonian functions. The dashed line indicates the end of the training interval.

three error measures from Eqs. (30)–(32). The ROM approximation error decays monotonically until $r > 80$ and then levels off. The optimization error follows a similar trend, yet the projection error continues to decay until $r \approx 170$. Fig. 9 (right) compares the GP-OpInf ROM approximation error with that of the SP-G ROM. We observe that SP-G and GP-OpInf achieve the same accuracy for $r < 70$, which covers the model dimension where most practical ROMs would be selected from, whereas for larger r the SP-G ROM performs better.

Test 2. Long-term predictive capabilities of the GP-OpInf ROM. We set $\Delta t = 10^{-2}$ and fix $\mu = \sqrt{2}$ and generate snapshots from FOM with the final time $T_{FOM} = 20$. We simulate the GP-OpInf ROM 100% past the training interval, so $T_{ROM} = 40$. Fig. 10 (left) shows the time evolution of the ROM approximation errors at $r = 10, 20, 40$ and 80 , where the FOM solution is the benchmark. The associated approximate Hamiltonian values are plotted in Fig. 10 (right), where the dashed line indicates the end of the training interval. The errors increase gradually as t increases due to usual error accumulation. Since the appropriate (gradient) structure is captured in the inferred GP-OpInf ROMs, the approximate Hamiltonian functions are constant and approach the benchmark value for r that is sufficiently large.

Test 3. Parametric predictions away from training data. We parameterize the initial condition as $u_0(x; \mu) = \text{sech}^2\left(\frac{x}{\mu}\right)$ for $\mu \in P = [1, 5]$. To generate snapshots, 9 training samples are uniformly collected from the interval and the FOM is simulated at all samples with $\Delta t = 10^{-2}$ and $T_{FOM} = 20$. We compute the POD basis from the collection of snapshot matrices.

Next, we infer the GP-OpInf ROM and use it to make predictions at the test parameters, where we keep the time interval fixed ($T_{ROM} = T_{FOM}$). We evaluate the ROMs accuracy at three randomly selected test parameters $\mu = 1.4, 2.8$ and 4.7 . Fig. 11 (top left) shows a comparison of the ROM approximation errors (30) of the SP-G and the GP-OpInf ROMs. At the test parameters, the error of the GP-OpInf ROM is close to the SP-G ROM when $r \leq 50$ (which is where most practical ROM model dimensions would be selected from), yet the error saturates when r gets bigger and thus becomes larger than that of SP-G. We fix $r = 40$ and plot the FOM solution and the GP-OpInf ROM solution at several selected time instances in Fig. 11 (top right and bottom). This shows that the ROM produces accurate approximations of the FOM solution at these test parameters.

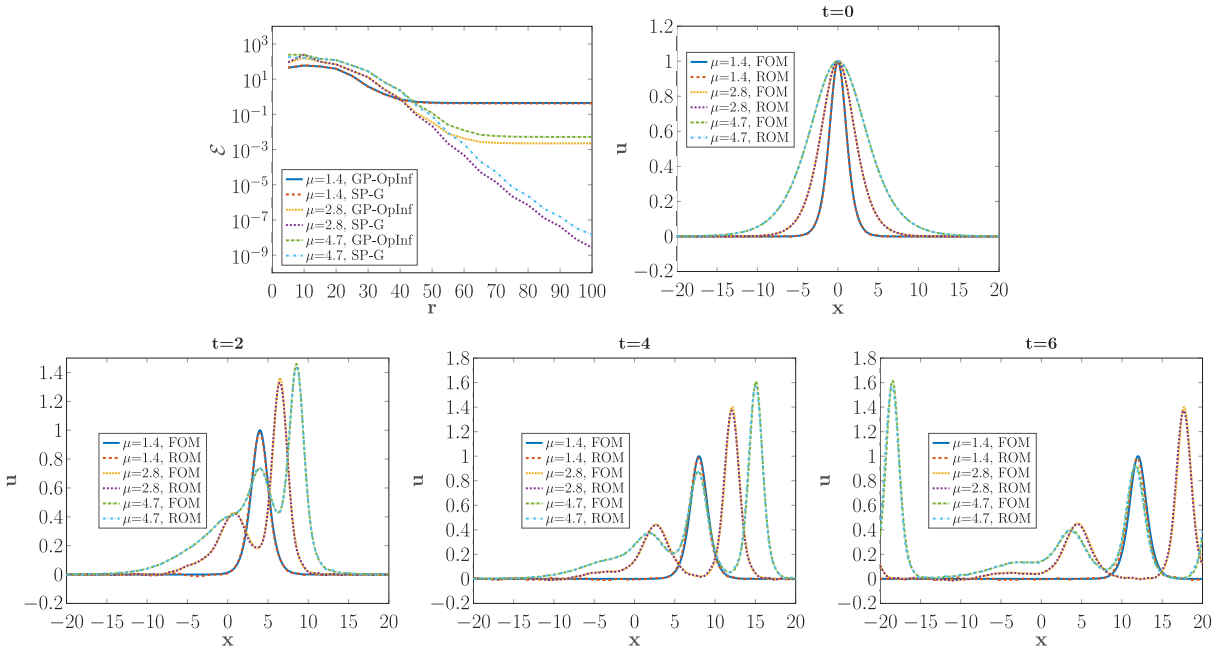


Fig. 11. (KdV Equation) Performance of the r -dimensional $GP\text{-}OpInf$ ROM at several test parameters: (top left) comparison of the ROM approximation error (30) for the $SP\text{-}G$ ROM of the same dimension r ; (top right and bottom) snapshots of the $GP\text{-}OpInf$ ROM solution and the FOM solution at several time instances.

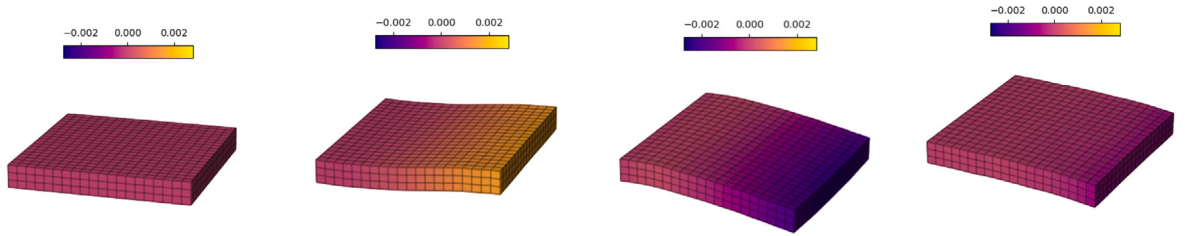


Fig. 12. (3D Cantilever Plate) Plate's surface plots at $t = 0, 0.005, 0.0125$, and 0.02 (from left to right), in which the color bar indicates the displacement in the vertical direction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2.3. Three-dimensional linear elastic cantilever plate

We consider the 3D linear elastic cantilever plate problem from [51]. The governing equation can be recast in the conservative Hamiltonian form

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_n \\ -\mathbf{I}_n & 0 \end{bmatrix} \begin{bmatrix} \mathbf{K} & 0 \\ 0 & \mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix},$$

where $\mathbf{u}$ is the displacement, $\mathbf{v} = \mathbf{M}\dot{\mathbf{u}}$ is the momentum, $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrix obtained from a finite element discretization. The Hamiltonian of the semi-discrete system is $H(\mathbf{u}, \mathbf{v}) = \frac{1}{2} (\mathbf{u}^T \mathbf{K} \mathbf{u} + \mathbf{v}^T \mathbf{M}^{-1} \mathbf{v})$. Following [51], we consider a rectangular steel plate of size $0.2 \text{ m} \times 0.2 \text{ m} \times 0.03 \text{ m}$, whose “left” side is set in place by a clamp, and the “right” side is given an initial upward velocity $(0, 0, 100)$. For the spatial discretization, we use $\Delta x = \Delta y = \Delta z = 0.01$ (correspondingly $n = 5, 292$), which is identical to the numerical setting in [51].¹ We simulate the full-order model over the time interval $[0, 0.02]$ with the implicit midpoint method, with time step size $\Delta t = 1 \times 10^{-4}$ to collect snapshots. Fig. 12 shows the 3D surface of the plate at time instances $t = 0, 0.005, 0.0125$ and 0.02 .

Test 1. Effects of regularization. Similar to the linear wave case, we obtain the reduced basis $\Phi = \begin{bmatrix} \Phi_u & 0 \\ 0 & \Phi_v \end{bmatrix}$ with $r_1 = r_2 = r$, and simulate the $GP\text{-}OpInf$, $GP\text{-}OpInf\text{-}V$, and $GP\text{-}OpInf\text{-}P$ (with $\alpha = 2$ and $c_0 = 10^{-13}$ in the definition of ϵ (17)) ROMs, respectively,

¹ We use the data from the github repository: <https://github.com/ikalash/HamiltonianOpInf>. We note that regularization is also used in their code to infer Hamiltonian reduced-order operators via optimization.

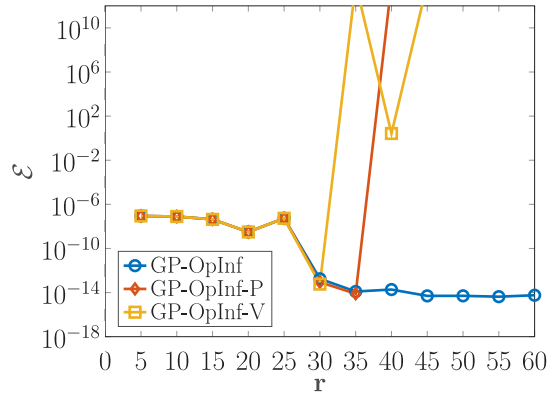


Fig. 13. (3D Cantilever Plate) Approximation errors for *GP-OpInf*, *GP-OpInf-V* and *GP-OpInf-P* as r increases.

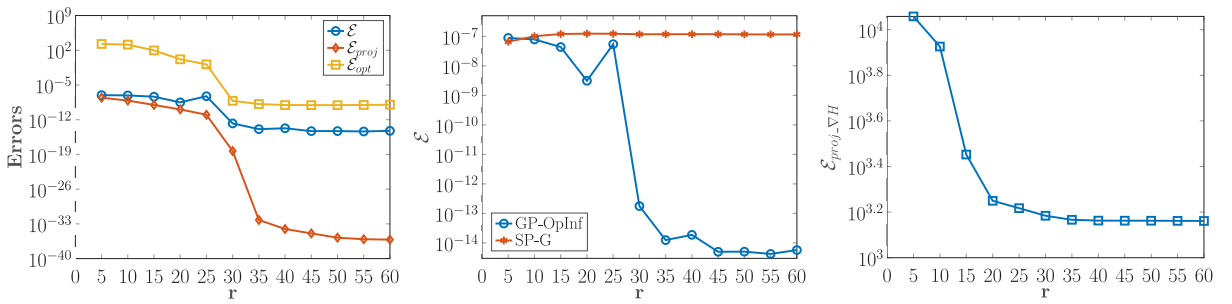


Fig. 14. (3D Cantilever Plate) Numerical errors of the *GP-OpInf*: (left) the ROM approximation error (30) together with POD projection error (31) and optimization error (32); (Middle) comparison with approximation errors of the *SP-G*; (Right) the projection error of ∇H onto the reduced basis space.

within the training data regime. The corresponding approximation errors are shown in Fig. 13 as r varies. It is evident that *GP-OpInf* surpasses the other two with its accuracy. Especially, as r increases beyond 35, the errors of *GP-OpInf-V* and *GP-OpInf-P* grow uncontrollably, while *GP-OpInf* maintains a small error.

Test 2. Illustration of error estimation. We now investigate the *GP-OpInf* approximation error, projection error, and optimization error in the training data regime. This comparison is illustrated in Fig. 14 (left), while r varies from 5 to 60, increasing incrementally by 5. The optimization error stagnates for $r \geq 30$ at a level below 10^{-6} which for many applications is sufficiently low.

We further compare the accuracy of the nonintrusive *GP-OpInf* with the intrusive *SP-G* in Fig. 14 (middle). Despite increasing r , the error of *SP-G* does not improve, whereas the *GP-OpInf* error decays to almost machine precision. A potential explanation for this phenomenon is discussed in Remark 2. In light of that discussion, we calculate the projection errors of ∇H -snapshots onto the reduced basis subspace,

$$\mathcal{E}_{\text{proj}, \nabla H}(r) = \frac{TAA}{N} \sum_{j=1}^N \|\nabla H(\mathbf{y}^j) - \Phi \Phi^T \nabla H(\mathbf{y}^j)\|^2,$$

and plot it in Fig. 14 (right). We observe that the magnitude of this projection error is on the order of 10^3 even when $r = 60$, which could limit the accuracy of *SP-G*. However, it does not affect the *GP-OpInf* as indicated by the error bound in Theorem 1.

We remark that we also tested a similar parametric problem, in which the initial velocity vector was parameterized as $(0, 0, \mu)$, with the parameter $\mu \in P := [80, 120]$. The *GP-OpInf* ROM, inferred based on a training set of 11 uniformly distributed parameter samples, accurately predicted the solution for randomly selected parameters from P . Since this phenomenon closely resembles our observations in the linear wave test case, we omit further discussion on it here.

5.3. Dissipative PDEs

We consider the Allen-Cahn equation to assess the *GP-OpInf* ROM's performance on a dissipative system.

5.3.1. One-dimensional Allen-Cahn equation

Consider the one-dimensional Allen-Cahn equation

$$u_t = \epsilon^2 u_{xx} + (u - u^3), \quad (x, t) \in \Omega \times I \text{ and } \mu \in P,$$

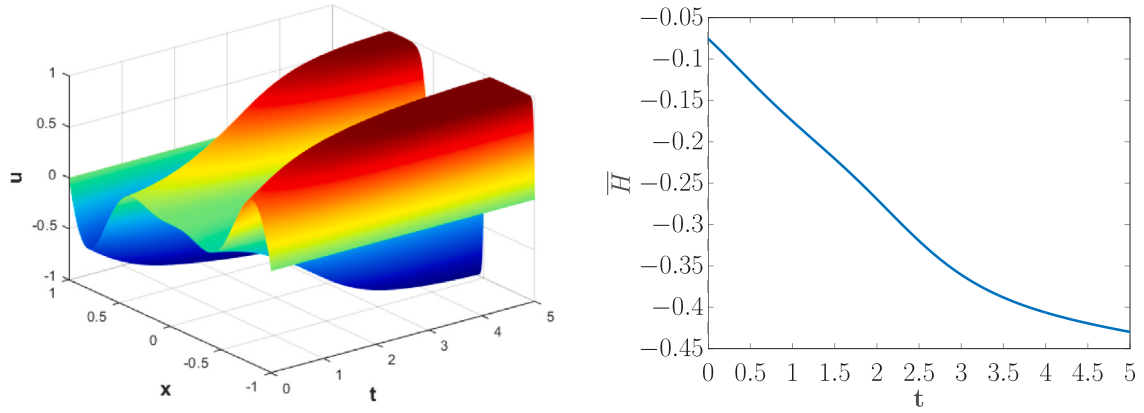


Fig. 15. (1D Allen-Cahn Equation) Full-order model simulation with final time $T_{\text{FOM}} = 5$: time evolution of the state $\mathbf{u}$ (left) and the approximate Lyapunov function $\bar{H}(t)$ (right).

where $u = u(x, t; \mu)$ with μ a parameter in the initial condition. The equation can be recast to the form (1), that is, $u_t = -\frac{\delta H}{\delta u}$, with the Lyapunov function $H(u) = \int_{\Omega} \left(\frac{\epsilon^2}{2} u_x^2 - \frac{1}{2} u^2 + \frac{1}{4} u^4 \right) dx$ (see, e.g., [22]). After applying a uniform spatial discretization with the mesh size Δx and defining a consistent discrete Lyapunov function $\bar{H}(t) := \Delta x H(\mathbf{u})$ with $H(\mathbf{u}) = \sum_j \left[\frac{\epsilon^2}{2\Delta x^2} (u_{j+1} - u_j)^2 - \frac{1}{2} u_j^2 + \frac{1}{4} u_j^4 \right]$, we obtain the semi-discrete system

$$\dot{\mathbf{u}} = -\mathbf{I}_n(-\mathbf{A}\mathbf{u} - \mathbf{u} + \mathbf{u}^3), \quad (35)$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\mathbf{A}$ is associated with a discrete, one-dimensional, second-order differential operator, and $\mathbf{u}^3$ is the component-wise cubic function of the vector $\mathbf{u}$. Since $-\mathbf{I}_n$ is negative definite, the Lyapunov function decreases in time.

Computational setting. Set $\Omega = [-1, 1]$, the interface parameter $\epsilon = 0.01$, and consider periodic boundary conditions $u(-1, t; \mu) = u(1, t; \mu)$ and $\frac{\partial u}{\partial n}(-1, t; \mu) = \frac{\partial u}{\partial n}(1, t; \mu)$ for $t \in I$. The initial condition is $u_0(x; \mu) = \mu x^2 \sin(2\pi x)$, for $\mu \in P = [0.2, 2]$ (see [72] for numerical simulations of a non-parametric configuration). In the FOM (35), we have

$$\mathbf{A} = \frac{\epsilon^2}{\Delta x^2} \mathbf{L}_n.$$

In all full-order simulations, we choose the mesh size $\Delta x = 10^{-3}$ and use the AVF method for time integration with a time step size Δt . To solve the nonlinear systems of equations, we apply Picard iteration. Fig. 15 shows the time evolution of the full-order state and discrete Lyapunov function $\bar{H}(t)$ where we set $\mu = 1$, $T_{\text{FOM}} = 5$ and $\Delta t = 10^{-3}$.

Test 1. Illustration of error estimation. We set $T_{\text{FOM}} = T_{\text{ROM}} = 3$, fix the parameter $\mu = 1$, and choose a small time step $\Delta t = 2.5 \times 10^{-4}$ so that the time discretization error is negligible in the error estimate. Varying the dimension r of the ROM, we compute the *GP-OpInf* ROM approximation error (30), projection error (31) and optimization error (32), which are shown in Fig. 16 (left). In this case, for $r \leq 15$, the optimization error is smaller than the POD projection error; however, for a bigger r , the former becomes larger and levels off. Correspondingly, the *GP-OpInf* ROM approximation error first decreases monotonically and then reaches a plateau. Therefore, compared to the *SP-G* ROM, the *GP-OpInf* ROM achieves the same accuracy for small r , but yields larger numerical error when its dimension is large, as shown in Fig. 16 (right).

Test 2. Prediction-in-time capabilities of the *GP-OpInf* ROM. Setting $\Delta t = 10^{-3}$, we generate snapshots from the full-order simulation with the final time $T_{\text{FOM}} = 3$ and simulate the ROM to the final time $T_{\text{ROM}} = 5$, which is 66% past the training data. The ROM approximation errors (30) at $r = 10, 20, 40$ and 80 are plotted in Fig. 17 (left) alongside the FOM solutions. The associated approximate Lyapunov function is plotted in Fig. 17 (right). The dashed line indicates the end of the training time interval. It is evident that the ROM is accurate within the interval $[0, 3]$, but its accuracy degrades when the simulation time exceeds this range. Meanwhile, the approximate Lyapunov is decreasing in time as guaranteed by the gradient structure of the ROM.

Test 3. Parametric predictions away from training data. We parameterize the problem with the interface parameter $\mu \in P = [0.2, 2]$, which also affects the initial condition. To generate snapshots, we simulated the FOM at 10 uniformly distributed training parameter samples in P with the time step $\Delta t = 10^{-3}$ and final simulation time $T_{\text{FOM}} = 3$. We construct the *GP-OpInf* ROM from the collected snapshot data and use it to obtain predictions at any given test parameter over the same time interval ($T_{\text{ROM}} = T_{\text{FOM}}$). Here, we select three random test parameters, $\mu = 0.34, 0.96$ and 1.87 .

Fig. 18 (top left) shows the approximation errors of the *SP-G* and *GP-OpInf* ROMs. This illustrates that, at all test parameters, the error of *GP-OpInf* is close to that of *SP-G* when $r \leq 30$. The former levels off at values around 10^{-15} when r becomes large, which is well beyond a typically required accuracy. Therefore, the *GP-OpInf* is very accurate. Fig. 18 (top right and bottom) compares the FOM solution with the *GP-OpInf* ROM solution of dimension $r = 30$ at several time instances. This confirms that the *GP-OpInf* ROM provides an accurate approximation to the FOM solution in all cases.

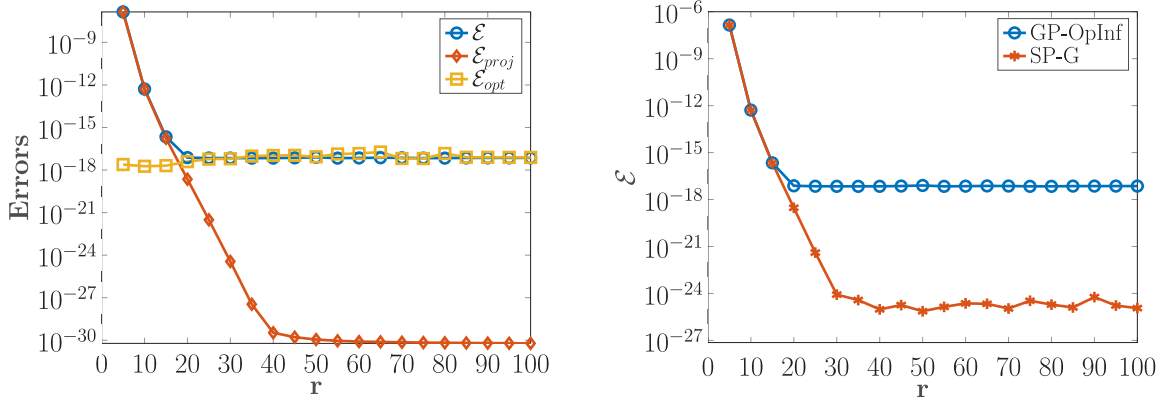


Fig. 16. (1D Allen-Cahn Equation) Numerical errors for the r -dimensional GP-OpInf when $T_{\text{FOM}} = T_{\text{ROM}} = 3$ and $\Delta t = 2.5 \times 10^{-4}$: (left) the ROM error together with the POD projection error (31) and the optimization error (32); (right) comparison of the GP-OpInf ROM approximation error (30) with that of the SP-G ROM.

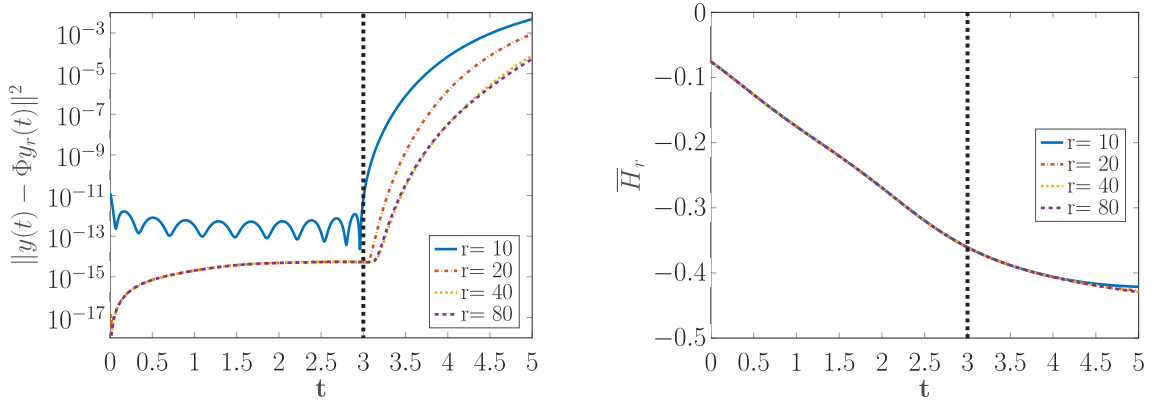


Fig. 17. (1D Allen-Cahn Equation) Numerical accuracy of the r -dimensional GP-OpInf ROM: (left) time evolution of the summand of the ROM approximation error (30); (right) time evolution of the reduced-order approximate Lyapunov function. The dashed line indicates the end of the training interval.

5.3.2. Two-dimensional parameterized Allen-Cahn equation

Consider the two-dimensional Allen-Cahn equation on a rectangular domain Ω :

$$u_t = \epsilon^2 \Delta u + u - u^3, \quad (x, y) \in \Omega, t \in I,$$

where $u = u(x, t; \mu)$ is the parameter-dependent solution and $\mu \in P$ a parameter appearing in the initial condition, defined below alongside the boundary and initial conditions. The Allen-Cahn equation can be recast into the form (1), in a similar way as in the 1D case, with the Lyapunov function $H(u) = \int_{\Omega} \left(\frac{\epsilon^2}{2} |\nabla u|^2 - \frac{1}{2} u^2 + \frac{1}{4} u^4 \right) dx$. Let $\mathbf{u}$ be a vectorization of the approximate state defined on the rectangular grid, obtained by applying a spatial discretization with a uniform mesh size Δx and Δy in the horizontal and vertical directions. The semi-discrete system is then

$$\dot{\mathbf{u}} = -\mathbf{I}_{n^2}(-\epsilon^2 \mathbf{L} \mathbf{u} - \mathbf{u} + \mathbf{u}^3),$$

where $\mathbf{L} = \mathbf{I}_n \otimes \mathbf{D}_x^2 + \mathbf{D}_y^2 \otimes \mathbf{I}_n$ with $\mathbf{D}_x^2 = \frac{1}{(\Delta x)^2} \mathbf{L}_n$, $\mathbf{D}_y^2 = \frac{1}{(\Delta y)^2} \mathbf{L}_n$ and $\otimes$ denotes the Kronecker product [11].

Computational setting. Let the problem be defined on $\Omega = [-0.5, 0.5]^2$ and $I = [0, T_{\text{FOM}}]$. Consider a periodic boundary condition and $\epsilon = 0.02$. The initial condition (following [73]) is given by

$$u_0(x, y; \mu) = \max \left(\tanh \left(\frac{r - \sqrt{(x - r(0.7 - \mu))^2 + (y + r\mu)^2}}{\epsilon} \right), \tanh \left(\frac{r - \sqrt{(x + r(0.7 - \mu))^2 + (y - r\mu)^2}}{\epsilon} \right) \right)$$

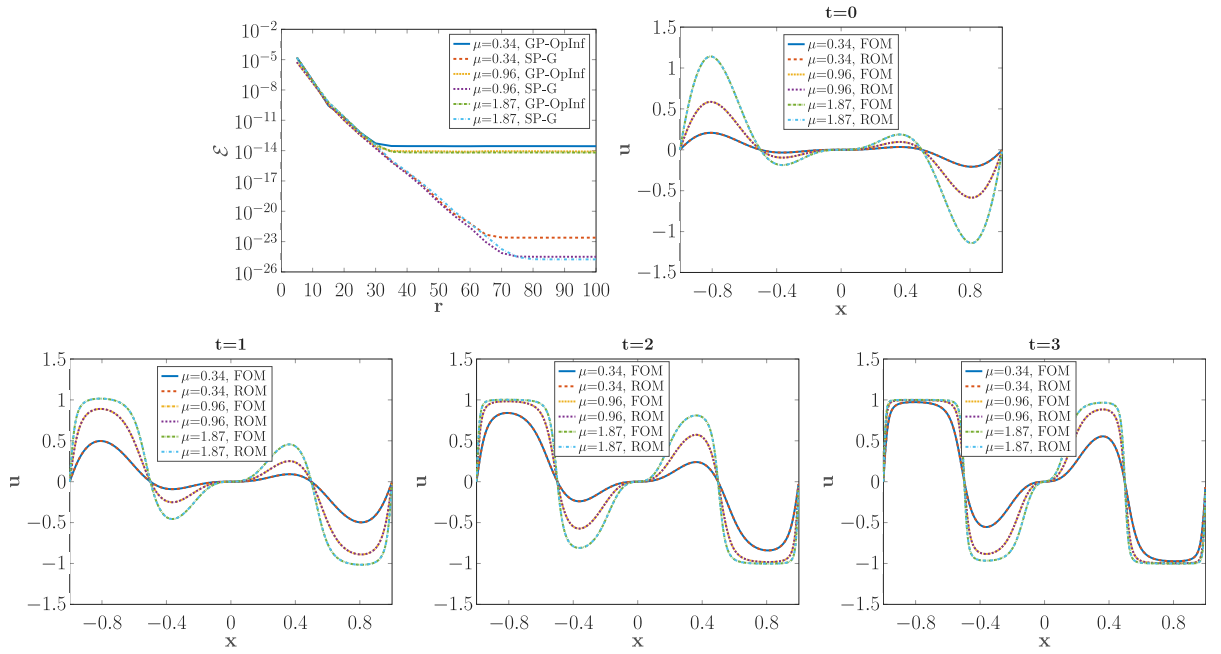


Fig. 18. (1D Allen-Cahn Equation) Performance of r -dimensional *GP-OpInf* ROM: (top left) comparison of the ROM approximation error (30) for the *SP-G* ROM in problems with different μ ; (top right and bottom) snapshots of the ROM and FOM solutions at $t = 0, 1, 2$ and 3 .

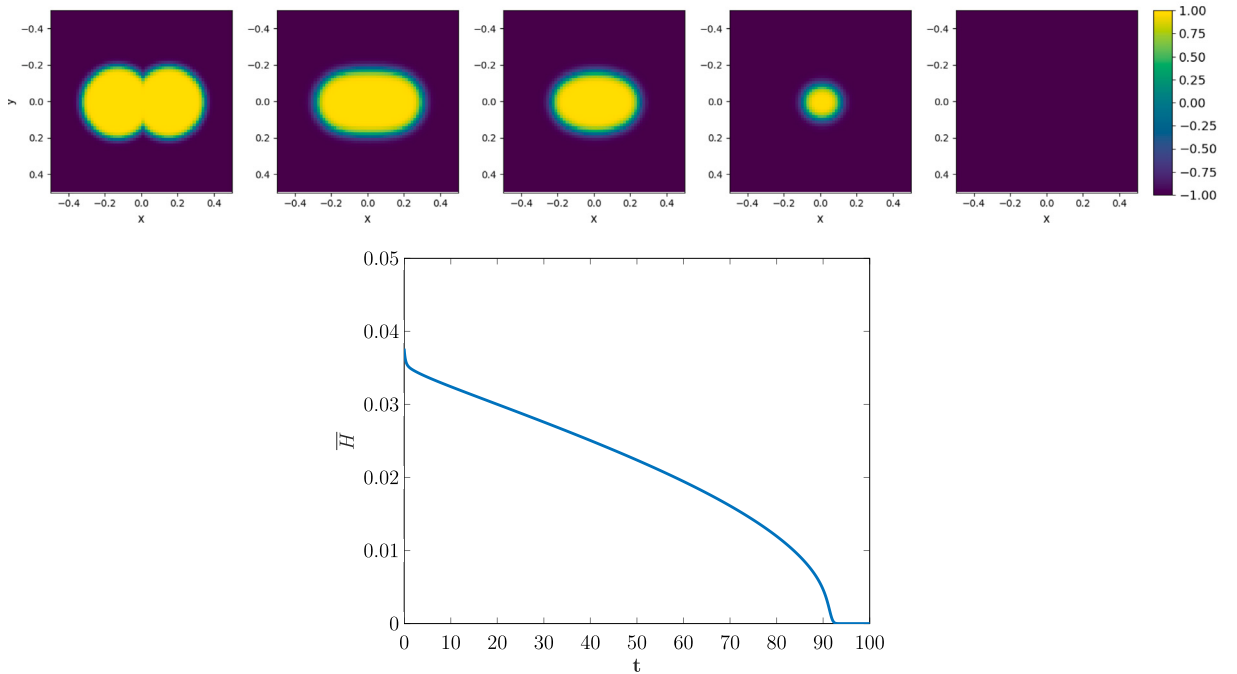


Fig. 19. (2D Allen-Cahn Equation) Full-order simulation when $T_{\text{FOM}} = 100$ and $\Delta t = 10^{-2}$: (top) approximate states at $t = 0, 20, 40, 80, 100$; (bottom) time evolution of the discrete Lyapunov function.

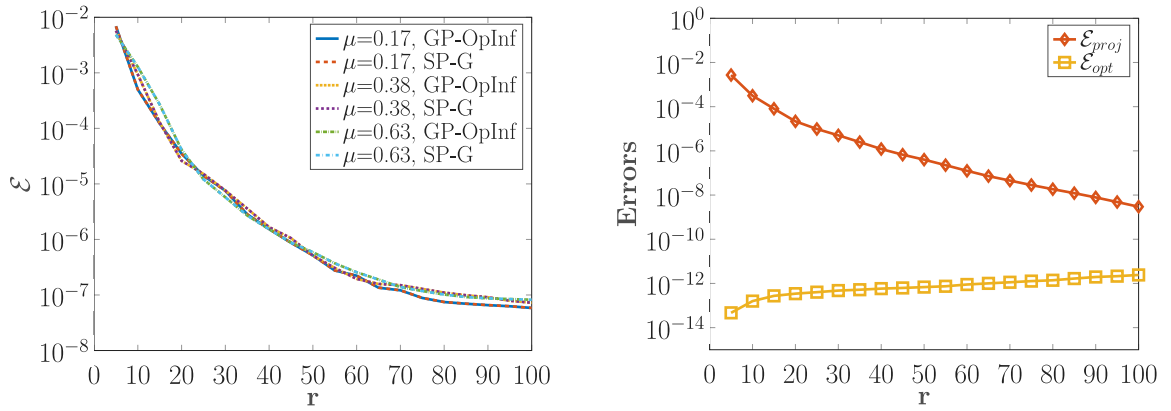


Fig. 20. (2D Allen-Cahn Equation) Numerical errors for the r -dimensional $GP\text{-}OpInf$ ROM: (left) comparison of the $GP\text{-}OpInf$ ROM approximation error (30) with that of the $SP\text{-}G$ ROM; (right) the projection error (31) and optimization error (32).

with $r = 0.2$ and μ a parameter in $P = [0, 0.7]$. The initial condition has two disks centered at $((0.7 - \mu)r, -\mu r)$ and $(-(0.7 - \mu)r, \mu r)$ with radius r . In all FOM simulations, the spatial domain is partitioned by a rectangular grid with the mesh size $\Delta x = \Delta y = 2^{-6}$ and the AVF method is used for time integration with a time step Δt . The nonlinear system is solved by the Picard iteration.

Fig. 19 shows the time evolution of the FOM state and the discrete Lyapunov function $\bar{H}(t)$ when $T_{\text{FOM}} = 100$, $\Delta t = 1 \times 10^{-2}$ and $\mu = 0$.

Similar to the 1D Allen-Cahn equation discussed in Section 5.3.1, we investigated the performance of the $GP\text{-}OpInf$ ROM for 2D Allen-Cahn equation through three tests where we illustrate the error estimation, evaluate the $GP\text{-}OpInf$'s capability for time prediction, and examine parametric predictions beyond the training data. We observed that the numerical behaviors of the $GP\text{-}OpInf$ ROM for the 2D Allen-Cahn equation closely align with those reported for the 1D case. Therefore, in the sequel, we only present the numerical test on the parametric problem in which $T_{\text{FOM}} = T_{\text{ROM}} = 20$ and $\Delta t = 10^{-2}$.

Test 1. Parametric predictions away from training data. To generate snapshots, 15 training samples are uniformly selected from P . The $GP\text{-}OpInf$ ROM is then inferred from the data and used to predict solutions at any given testing samples from P . Here, the approximation errors of $SP\text{-}G$ and $GP\text{-}OpInf$ at three test parameters $\mu = 0.17, 0.38$ and 0.63 are compared in Fig. 20 (left). The figure shows that the error of $GP\text{-}OpInf$ is close to that of $SP\text{-}G$ with the same dimension r at all test parameters. As indicated in Fig. 20 (right), the optimization error is significantly lower than the POD projection error, thereby aligning the trends of errors for both $GP\text{-}OpInf$ and $SP\text{-}G$ with the POD projection error. Fixing $r = 40$, we plot the ROM solutions and associated absolute errors at time instances $t = 0, 5, 10, 15, 20$ (from top to bottom) in Fig. 21. We observe that the $GP\text{-}OpInf$ ROM produces accurate solutions for the parametric problem at the selected testing samples.

6. Conclusions

In this work, we considered evolutionary PDEs with a gradient structure with the goal to infer reduced-order models with the same gradient structure from simulated data of the semi-discretized PDEs. We first projected the high-dimensional snapshot data onto a POD basis, and from the projected data we infer a low-dimensional operator by solving a constrained optimization problem. The resulting operators then define a low-dimensional model, termed the gradient-preserving Operator Inference ($GP\text{-}OpInf$) ROM. By enforcing the proper constraints in the inference of the low-dimensional operators, we ensure that the $GP\text{-}OpInf$ ROM has the appropriate gradient structure, thereby preserving the essential physics in the reduced-order dynamics. We further analyzed the associated approximation error of the $GP\text{-}OpInf$ ROM. The analysis shows that an upper bound for the error consists of the combination of the POD projection error, the data error, and the $OpInf$ optimization error. We test the accuracy, structure-preservation properties, and predictive capabilities of the $GP\text{-}OpInf$ ROM on several PDE examples. For conservative test cases, we consider the parameterized wave equation and Korteweg-de-Vries equation, as well as a three-dimensional linear elasticity problem, and for dissipative test cases we consider the one-and two-dimensional Allen-Cahn equation. The numerical experiments show that $GP\text{-}OpInf$ can achieve the same performance as the intrusive, projection-based ROM when the optimization error and data error are dominated by the POD projection error, which is typically true if the dimension of the ROM is low. This observation aligns with our error estimation. Moreover, for parametric problems, the low-dimensional $GP\text{-}OpInf$ ROMs can attain accuracy close to that of the full-order simulations even for unseen parameters.

Going forward, it would be interesting to learn the Hamiltonian or its gradient function entirely from data, without relying on any prior knowledge. For this purpose, the work in [55] could be extended to the Hamiltonian case. Moreover, a joint learning of the dissipative and conservative matrices, similar to the port-Hamiltonian DMD framework from [52] would be an important next step.

CRediT authorship contribution statement

Yuwei Geng: Visualization, Software, Investigation. **Jasdeep Singh:** Software, Investigation. **Lili Ju:** Writing – review & editing, Methodology. **Boris Kramer:** Writing – review & editing, Validation, Funding acquisition. **Zhu Wang:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization.

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.

Data availability

Data will be made available on request.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable suggestions and feedback. We are also grateful to Dr. Irena Tezaur (Sandia National Laboratories) for making their codes and data for the 3D plate simulations available online, and for patiently providing us with instructions on setting up the experiment.

B.K. and Z.W. were in part supported by the U.S. Office of Naval Research under award number N00014-22-1-2624. Moreover, B.K. was in part supported by the Ministry of Trade, Industry and Energy (MOTIE) and the Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (No. P0019804, Digital twin based intelligent unmanned facility inspection solutions). Z. W. was in part supported by the U.S. National Science Foundation under award number DMS-2038080 and an ASPIRE grant from the Office of the Vice President for Research at the University of South Carolina. L. J. was in part supported by the U.S. National Science Foundation under award number DMS-2109633 and the U.S. Department of Energy under award number DE-SC0022254.

Appendix. Iterative algorithm for solving (21)

To solve (21), we first find the gradient of the objective function. The barrier function $g(\mathbf{X})$ can be written as follows.

$$g(\mathbf{X}) = \sum_{i=1}^r \log(-\lambda_i(\mathbf{X})) = \log \left(\prod_{i=1}^r (-\lambda_i(\mathbf{X})) \right) = \log((-1)^r \det(\mathbf{X})).$$

Without loss of generality, we consider an even r here. Differentiating it with respect to the ij entry yields

$$\frac{\partial \log(\det(\mathbf{X}))}{\partial \mathbf{X}_{ij}} = \frac{1}{\det(\mathbf{X})} \frac{\partial \det(\mathbf{X})}{\partial \mathbf{X}_{ij}}.$$

Based on the cofactor formula for determinant, $\det(\mathbf{X}) = \sum_{k=1}^s \mathbf{X}_{ik} \mathbf{C}_{ik}$, we have

$$\frac{\partial \det(\mathbf{X})}{\partial \mathbf{X}_{ij}} = \frac{\partial (\sum_{k=1}^r \mathbf{X}_{ik} \mathbf{C}_{ik})}{\partial \mathbf{X}_{ij}} = \sum_{k=1}^s \left(\frac{\partial \mathbf{X}_{ik}}{\partial \mathbf{X}_{ij}} \mathbf{C}_{ik} + \mathbf{X}_{ik} \frac{\partial \mathbf{C}_{ik}}{\partial \mathbf{X}_{ij}} \right) = \mathbf{C}_{ij}.$$

Therefore, we have

$$\frac{\partial \log(\det(\mathbf{X}))}{\partial \mathbf{X}_{ij}} = \frac{1}{\det(\mathbf{X})} \mathbf{C}_{ij}.$$

Due to the fact that $\mathbf{X}^{-1} = \frac{1}{\det(\mathbf{X})} \mathbf{C}^T$, we have $\nabla g(\mathbf{X}) = \mathbf{X}^{-1}$, and

$$\nabla g(\mathbf{D}_r) = \mathbf{D}_r^{-T}. \quad (\text{A.1})$$

Next, $f(\mathbf{D}_r)$ is recast into the following form using the trace:

$$\begin{aligned} f(\mathbf{D}_r) &= \text{tr} \left((\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r) (\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r)^T \right) \\ &= \text{tr} \left(\dot{\mathbf{Y}}_r \dot{\mathbf{Y}}_r^T - \dot{\mathbf{Y}}_r \mathbf{F}_r^T \mathbf{D}_r^T - \mathbf{D}_r \mathbf{F}_r \dot{\mathbf{Y}}_r^T + \mathbf{D}_r \mathbf{F}_r \mathbf{F}_r^T \mathbf{D}_r^T \right). \end{aligned}$$

Then the gradient of $f(\mathbf{D}_r)$ can be obtained as follows:

$$\nabla f(\mathbf{D}_r) = -\dot{\mathbf{Y}}_r \mathbf{F}_r^T - \dot{\mathbf{Y}}_r \mathbf{F}_r^T + (\mathbf{F}_r \mathbf{F}_r^T \mathbf{D}_r^T)^T + \mathbf{D}_r \mathbf{F}_r \mathbf{F}_r^T = -2(\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r) \mathbf{F}_r^T. \quad (\text{A.2})$$

By combining (A.2) and (A.1), we have

$$\nabla f_\beta(\mathbf{D}_r) = \nabla f(\mathbf{D}_r) - \beta \nabla g(\mathbf{D}_r) = -2(\dot{\mathbf{Y}}_r - \mathbf{D}_r \mathbf{F}_r) \mathbf{F}_r^T - \beta \mathbf{D}_r^{-T}.$$

Secondly, we use the gradient descent method [74] to approximate the solution $\mathbf{D}_r^{k+1}$ by the given $\mathbf{D}_r^k$, which follows

$$\mathbf{D}_r^{k+1} = \mathbf{D}_r^k - \tau^k \nabla f_\beta(\mathbf{D}_r^k)$$

where $\tau^k \geq 0$ is the step size. Instead of a constant barrier parameter β , one could use a decaying parameter with $\beta^{k+1} = \sigma \beta^k$, where $\sigma \in (0, 1)$ and its value depends on r [75,76].

This iteration continues till the magnitude of gradient $\nabla f_\beta(\mathbf{D}_r)$ is sufficiently small and the barrier parameter β drops below a user-defined tolerance.

References

- [1] J.E. Marsden, T.S. Ratiu, *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*, vol. 17, Springer Science & Business Media, 2013.
- [2] A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I, *J. Comput. Phys.* 1 (1) (1966) 119–143.
- [3] C. Pagliantini, G. Manzini, O. Koshkarov, G.L. Delzanno, V. Roytershteyn, Energy-conserving explicit and implicit time integration methods for the multi-dimensional Hermite-DG discretization of the Vlasov–Maxwell equations, *Comput. Phys. Comm.* 284 (2023) 108604.
- [4] R. Salmon, The shape of the main thermocline, *J. Phys. Oceanogr.* 12 (12) (1982) 1458–1479.
- [5] A.L. Stewart, P.J. Dellar, Multilayer shallow water equations with complete Coriolis force. Part 1. Derivation on a non-traditional beta-plane, *J. Fluid Mech.* 651 (2010) 387–413.
- [6] A.L. Stewart, P.J. Dellar, An energy and potential enstrophy conserving numerical scheme for the multi-layer shallow water equations with complete Coriolis force, *J. Comput. Phys.* 313 (2016) 99–120.
- [7] Q. Chen, L. Ju, R. Temam, Conservative numerical schemes with optimal dispersive wave relations: Part I. Derivation and analysis, *Numer. Math.* 149 (1) (2021) 43–85.
- [8] S.H. Christiansen, H.Z. Munthe-Kaas, B. Owren, Topics in structure-preserving discretization, *Acta Numer.* 20 (2011) 1–119.
- [9] H. Sharma, M. Patil, C. Woolsey, A review of structure-preserving numerical methods for engineering applications, *Comput. Methods Appl. Mech. Engrg.* 366 (2020) 113067.
- [10] T. Fukao, S. Yoshikawa, S. Wada, Structure-preserving finite difference schemes for the Cahn–Hilliard equation with dynamic boundary conditions in the one-dimensional case, *Commun. Pure Appl. Anal.* 16 (5) (2017) 1915–1938.
- [11] W. Cai, C. Jiang, Y. Wang, Y. Song, Structure-preserving algorithms for the two-dimensional sine-Gordon equation with Neumann boundary conditions, *J. Comput. Phys.* 395 (2019) 166–185.
- [12] M. Groß, P. Betsch, P. Steinmann, Conservation properties of a time FE method. Part IV: Higher order energy and momentum conserving schemes, *Internat. J. Numer. Methods Engrg.* 63 (13) (2005) 1849–1897.
- [13] Y. Xu, J.J. van der Vegt, O. Bokhove, Discontinuous Hamiltonian finite element method for linear hyperbolic systems, *J. Sci. Comput.* 35 (2008) 241–265.
- [14] Z. Sun, Y. Xing, On structure-preserving discontinuous Galerkin methods for Hamiltonian partial differential equations: energy conservation and multi-symplecticity, *J. Comput. Phys.* 419 (2020) 109662.
- [15] M.A. Sánchez, B. Cockburn, N.-C. Nguyen, J. Peraire, Symplectic Hamiltonian finite element methods for linear elastodynamics, *Comput. Methods Appl. Mech. Engrg.* 381 (2021) 113843.
- [16] R.I. McLachlan, G. Quispel, et al., *Six Lectures on the Geometric Integration of ODEs*, Citeseer, 1998.
- [17] R.I. McLachlan, G.R.W. Quispel, Geometric integrators for ODEs, *J. Phys. A: Math. Gen.* 39 (19) (2006) 5251.
- [18] E. Hairer, M. Hochbruck, A. Iserles, C. Lubich, Geometric numerical integration, *Oberwolfach Rep.* 3 (1) (2006) 805–882.
- [19] G. Quispel, G.S. Turner, Discrete gradient methods for solving ODEs numerically while preserving a first integral, *J. Phys. A: Math. Gen.* 29 (13) (1996) L341.
- [20] R.I. McLachlan, G.R.W. Quispel, N. Robidoux, Geometric integration using discrete gradients, *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 357 (1754) (1999) 1021–1045.
- [21] G. Quispel, D.I. McLaren, A new class of energy-preserving numerical integration methods, *J. Phys. A* 41 (4) (2008) 045206.
- [22] E. Celledoni, V. Grimm, R.I. McLachlan, D. McLaren, D. O’Neale, B. Owren, G. Quispel, Preserving energy resp. dissipation in numerical PDEs using the average vector field method, *J. Comput. Phys.* 231 (20) (2012) 6770–6789.
- [23] D. Furihata, T. Matsuo, *Discrete Variational Derivative Method: A Structure-Preserving Numerical Method for Partial Differential Equations*, CRC Press, 2010.
- [24] A.C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, 2005.
- [25] A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: an Introduction*, vol. 92, Springer, 2015.
- [26] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.* 57 (4) (2015) 483–531.
- [27] A.C. Antoulas, C.A. Beattie, S. Gügercin, *Interpolatory Methods for Model Reduction*, SIAM, 2020.
- [28] S.L. Brunton, J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2022.
- [29] S. Lall, P. Krysl, J.E. Marsden, Structure-preserving model reduction for mechanical systems, *Physica D* 184 (1–4) (2003) 304–318.
- [30] K. Carlberg, R. Tuminaro, P. Boggs, Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics, *SIAM J. Sci. Comput.* 37 (2) (2015) B153–B184.
- [31] C. Beattie, S. Gugercin, Interpolatory projection methods for structure-preserving model reduction, *Systems Control Lett.* 58 (3) (2009) 225–232.
- [32] S. Gugercin, R.V. Polyuga, C. Beattie, A. Van Der Schaft, Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems, *Automatica* 48 (9) (2012) 1963–1974.
- [33] Y. Gong, Q. Wang, Z. Wang, Structure-preserving Galerkin POD reduced-order modeling of Hamiltonian systems, *Comput. Methods Appl. Mech. Engrg.* 315 (2017) 780–798.
- [34] B. Karasözen, M. Uzunca, Energy preserving model order reduction of the nonlinear Schrödinger equation, *Adv. Comput. Math.* 44 (2018) 1769–1796.
- [35] Y. Miyatake, Structure-preserving model reduction for dynamical systems with a first integral, *Japan J. Ind. Appl. Math.* 36 (2019) 1021–1037.
- [36] C. Beattie, S. Gugercin, Structure-preserving model reduction for nonlinear port-Hamiltonian systems, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 6564–6569.
- [37] S. Chaturantabut, C. Beattie, S. Gugercin, Structure-preserving model reduction for nonlinear port-Hamiltonian systems, *SIAM J. Sci. Comput.* 38 (5) (2016) B837–B865.
- [38] L. Peng, K. Mohseni, Symplectic model reduction of Hamiltonian systems, *SIAM J. Sci. Comput.* 38 (1) (2016) A1–A27.
- [39] Gruber, A. and Tezaur, I., Variationally consistent Hamiltonian model reduction, 2024. *ArXiv preprint arXiv:2024.15315*.
- [40] B.M. Afkham, J.S. Hesthaven, Structure preserving model reduction of parametric Hamiltonian systems, *SIAM J. Sci. Comput.* 39 (6) (2017) A2616–A2644.
- [41] J. Hesthaven, C. Pagliantini, Structure-preserving reduced basis methods for Poisson systems, *Math. Comp.* 90 (330) (2021) 1701–1740.

- [42] C. Pagliantini, Dynamical reduced basis methods for Hamiltonian systems, *Numer. Math.* 148 (2) (2021) 409–448.
- [43] J.S. Hesthaven, C. Pagliantini, G. Rozza, Reduced basis methods for time-dependent problems, *Acta Numer.* 31 (2022) 265–345.
- [44] H. Sharma, H. Mu, P. Buchfink, R. Geelen, S. Glas, B. Kramer, Symplectic model reduction of Hamiltonian systems using data-driven quadratic manifolds, *Comput. Methods Appl. Mech. Engrg.* 417 (2023) 116402.
- [45] P. Buchfink, S. Glas, B. Haasdonk, Symplectic model reduction of Hamiltonian systems on nonlinear manifolds and approximation with weakly symplectic autoencoder, *SIAM J. Sci. Comput.* 45 (2) (2023) A289–A311.
- [46] S. Yildiz, P. Goyal, T. Bendokat, P. Benner, Data-driven identification of quadratic symplectic representations of nonlinear Hamiltonian systems, 2023, arXiv preprint arXiv:2308.01084.
- [47] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, *Comput. Methods Appl. Mech. Engrg.* 306 (2016) 196–215.
- [48] B. Kramer, B. Peherstorfer, K.E. Willcox, Learning nonlinear reduced models from data with operator inference, *Annu. Rev. Fluid Mech.* 56 (1) (2024) 521–548.
- [49] S.A. McQuarrie, P. Khodabakhshi, K.E. Willcox, Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference, *SIAM J. Sci. Comput.* 45 (4) (2023) A1917–A1946.
- [50] H. Sharma, Z. Wang, B. Kramer, Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems, *Physica D* 431 (2022) 133122.
- [51] A. Gruber, I. Tezaur, Canonical and noncanonical Hamiltonian operator inference, *Comput. Methods Appl. Mech. Engrg.* 416 (2023) 116334.
- [52] R. Morandin, J. Nicodemus, B. Unger, Port-Hamiltonian dynamic mode decomposition, *SIAM J. Sci. Comput.* 45 (4) (2023) A1690–A1710.
- [53] H. Sharma, B. Kramer, Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale mechanical systems, *Physica D* 462 (2024) 134128.
- [54] Y. Filanova, I.P. Duff, P. Goyal, P. Benner, An operator inference oriented approach for linear mechanical systems, *Mech. Syst. Signal Process.* 200 (2023) 110620.
- [55] K. Sharma, D.A. Najera-Flores, M.D. Todd, B. Kramer, Lagrangian operator inference enhanced with structure-preserving machine learning for nonintrusive model reduction of mechanical systems, *Comput. Methods Appl. Mech. Engrg.* 423 (2024) 116865.
- [56] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.* 25 (1) (1993) 539–575.
- [57] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, *Q. Appl. Math.* 45 (3) (1987) 561–571.
- [58] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent pdes with multiple parameters in the boundary data, *Comput. Methods Appl. Mech. Engrg.* 196 (4–6) (2007) 1030–1047.
- [59] P.J. Antsaklis, A.N. Michel, *Linear Systems*, vol. 8, Springer, 1997.
- [60] B. Kågström, P. Poromaa, Lapack-style algorithms and software for solving the generalized sylvester equation and estimating the separation between regular matrix pairs, *ACM Trans. Math. Softw.* 22 (1) (1996) 78–103.
- [61] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, 2001.
- [62] C. Helmberg, F. Rendl, R.J. Vanderbei, H. Wolkowicz, An interior-point method for semidefinite programming, *SIAM J. Optim.* 6 (2) (1996) 342–361.
- [63] C. Pagliantini, F. Vismara, Gradient-preserving hyper-reduction of nonlinear dynamical systems via discrete empirical interpolation, *SIAM J. Sci. Comput.* 45 (5) (2023) A2725–A2754.
- [64] R.H. Bartels, G.W. Stewart, Algorithm 432 [c2]: solution of the matrix equation $ax + xb = c$ [f4], *Commun. ACM* 15 (9) (1972) 820–826.
- [65] S. Chaturantabut, D.C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction, *SIAM J. Numer. Anal.* 50 (1) (2012) 46–63.
- [66] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, *Numer. Math.* 90 (2001) 117–148.
- [67] J.R. Singler, New pod error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs, *SIAM J. Numer. Anal.* 52 (2) (2014) 852–876.
- [68] L.C. Evans, *Partial Differential Equations*, vol. 19, American Mathematical Society, 2022.
- [69] MOSEK ApS, The MOSEK optimization toolbox for MATLAB manual. Version 9.0, 2019.
- [70] S. Diamond, S. Boyd, CVXPY: A python-embedded modeling language for convex optimization, *J. Mach. Learn. Res.* 17 (83) (2016) 1–5.
- [71] A. Agrawal, R. Verschuere, S. Diamond, S. Boyd, A rewriting system for convex optimization problems, *J. Control Decis.* 5 (1) (2018) 42–60.
- [72] C.L. Zhao, Solving Allen–Cahn and Cahn–Hilliard equations using the adaptive physics informed neural networks, *Commun. Comput. Phys.* 29 (3) (2020).
- [73] Y. Geng, Y. Teng, Z. Wang, L. Ju, A deep learning method for the dynamics of classic and conservative Allen–Cahn equations based on fully-discrete operators, *J. Comput. Phys.* (2023) 112589.
- [74] S.-i. Amari, Backpropagation and stochastic gradient descent method, *Neurocomputing* 5 (4–5) (1993) 185–196.
- [75] F.A. Potra, S.J. Wright, Interior-point methods, *J. Comput. Appl. Math.* 124 (1–2) (2000) 281–302.
- [76] C.C. Gonzaga, An algorithm for solving linear programming problems in $O(n^3)$ operations, in: *Progress in Mathematical Programming: Interior-Point and Related Methods*, Springer, 1989, pp. 1–28.