SIAM J. SCI. COMPUT.
Vol. 48, No. 5, pp. A2391–A2416

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

EXACT OPERATOR INFERENCE WITH MINIMAL DATA*

HENRIK ROSENBERGER†, BENJAMIN SANDERSE†, AND GIOVANNI STABILE‡

Abstract. This work introduces a novel method to generate snapshot data for operator inference that guarantees the exact reconstruction of intrusive projection-based reduced-order models (ROMs). To ensure exact reconstruction, the operator inference least squares matrix must have full rank, without regularization. Existing works have achieved this full rank using heuristic strategies to generate snapshot data and a-posteriori checks on full rank but without a guarantee of success. Our novel snapshot data generation method provides this guarantee thanks to two key ingredients: first, we identify ROM states that induce full rank, then we generate snapshots corresponding to exactly these states by simulating multiple trajectories for only a single time step. This way, the number of required snapshots is minimal and orders of magnitude lower than typically reported with existing methods. The method avoids non-Markovian terms and does not require reprojection. Since the number of snapshots is minimal, the least squares problem simplifies to a linear system that is numerically more stable. In addition, because the inferred operators are exact, properties of the intrusive ROM operators such as symmetry or skew-symmetry are preserved. Numerical results for differential equations involving 2nd, 3rd, and 8th order polynomials demonstrate that the novel snapshot data generation method leads to exact reconstruction of the intrusive reduced order models.

Key words. operator inference, data-driven modeling, nonintrusive model reduction, reduced basis method

MSC codes. 65Y99, 65F22, 35R30, 65D05

DOI. 10.1137/25M1772484



See reproducibility of computational results at end of the article.

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

1. Introduction.

Computational efficiency is of paramount importance in complex engineering tasks such as optimization, data assimilation, uncertainty quantification, and the development of digital twins. All these tasks require multiple simulations with similar settings. While first principle-derived models have primarily been developed to run single simulations, data-driven models are being developed to exploit the similarity among such multiple simulations.

These data-driven surrogate models vary in how much they preserve the structure of the corresponding first principle-derived model [13]. Statistical data-fit models such as Gaussian process models [32] generally do not preserve such structure and solely approximate the mapping between simulation inputs and outputs. Reduced-order models (ROMs), on the other hand, are closely based on first principle-derived, full-order models (FOMs) and perform simulations on a lower-dimensional subspace, for example, by projecting the first principle-derived model onto this subspace [1, 10, 23]. Advantages of such projection-based ROMs are the availability of rigorous theory

*Submitted to the journal's Numerical Algorithms for Scientific Computing section June 27, 2025; accepted for publication (in revised form) March 16, 2026; published electronically September 1, 2026.
https://doi.org/10.1137/25M1772484
Funding: This publication is part of the project "Discretize first, reduce next" (with project number VI.Vidi.193.105 of the research programme NWO Talent Programme Vidi which is (partly) financed by the Dutch Research Council (NWO).

†Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, and Centre for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, Eindhoven, The Netherlands (hker@cwi.nl, sanderse@cwi.nl).
‡The Biorobotics Institute, Sant'Anna School of Advanced Studies, Pisa, Italy (giovanni.stabile@santannapisa.it).

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

that provides a posteriori error estimates [9, 11, 20, 28, 31] and the smaller demand for training data compared with surrogate models that preserve less structure. A disadvantage of projection-based ROMs is that they are intrusive in the sense that they require access to the numerical operators of the FOM to perform the projection onto the lower-dimensional subspace. This access is often not possible in software used in industry and hence impedes the use of intrusive reduced-order models in practice.

To address this practical problem, methods have been developed to approximate intrusive-reduced models in a nonintrusive way. These nonintrusive methods do not require access to the numerical operators of the FOM but only to the generated output data. For linear models, dynamic mode decomposition finds the linear operator that fits the data best [14]. For nonlinear models, operator inference generalizes this approach to polynomial operators [19]. For nonpolynomial models, a variable transformation can be used to obtain a polynomial model [21], and so operator inference can still be applied.

However, two major difficulties currently impede operator inference in accurately reconstructing the intrusive ROM operators. First, the FOM snapshot data contain non-Markovian information that is inconsistent with the Markovian intrusive ROM. This non-Markovian information perturbs the operator inference minimization problem such that the intrusive ROM generally does not minimize this perturbed problem. To eliminate this perturbation, Peherstorfer has proposed to generate additional FOM snapshot data whereby the FOM state is projected onto the ROM basis after each time step [18], known as the reprojection approach. The resulting FOM snapshot data do not contain non-Markovian information and are equivalent to snapshot data generated by the intrusive ROM. Consequently, the minimization problem is unperturbed, and the intrusive ROM does minimize it. However, operator inference is only guaranteed to yield this solution if this minimizer is unique. Concretely, the operator inference minimization problem is a least squares problem and has a unique solution only if the least squares matrix has full rank.

This full rank requirement is the second major difficulty in accurately reconstructing the intrusive ROM operators. To the authors' knowledge, no method has been proposed yet to ensure this full rank requirement a priori. In practice, two different approaches have been suggested to address this issue. The first approach is to generate large numbers of snapshots and test a posteriori whether the resulting least squares problem has full rank [18, 19]. However, this approach does not guarantee full rank, even when the number of generated snapshots exceeds the required rank by orders of magnitude and hence causes large computational costs. The second approach is to circumvent the need for full rank by adding a regularization term to the operator inference least squares problem [25, 26]. However, this modification inevitably causes the resulting inferred ROM operators to deviate from the intrusive ROM operators.

In this work, we propose a novel snapshot data generation method that guarantees the full rank of the operator inference least squares problem. The number of generated snapshots is equal to this rank and hence minimal. Like Peherstorfer's approach [18], our method avoids non-Markovian information and hence reconstructs the intrusive ROM operators exactly, i.e., up to machine precision. Our method has two key ingredients. First, we leverage the knowledge about the specific polynomial structure of the intrusive ROM to identify a set of ROM states and input signals that define snapshots that induce a full rank least squares problem. Second, we generate snapshots corresponding to these identified ROM states and input signals by simulating multiple trajectories, initialized with the identified ROM states as initial conditions and the identified input signals, for only a single explicit Euler time step.

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

In contrast to the usual approach that generates snapshot data along long trajectories, this single-step approach gives us control over which snapshot data we obtain. Additionally, this single-step approach avoids any non-Markovian terms, so we do not need reprojection [18], and the explicit Euler scheme prevents time discretization inconsistencies. Since the number of snapshots is minimal, the least squares problem simplifies to a linear system that is numerically more stable. Since the method exactly reconstructs intrusive ROM operators, it preserves structures such as symmetry and skew-symmetry of these intrusive operators [24] without the need to enforce this preservation explicitly [7, 8, 12].

This article is organized as follows. In section 2, we introduce dynamical systems with polynomial terms, intrusive projection-based model reduction, operator inference and the limitation of insufficient snapshot data. In section 3, we propose our novel method for rank-sufficient snapshot data generation. In section 4, we demonstrate the novel method in numerical experiments. In section 5, we draw conclusions and give an outlook on future work.

2. Preliminaries. Our notation is based on [18], but we start at the time-continuous level.

2.1. Dynamical systems with polynomial terms. As full-order model (FOM), we consider a time-continuous dynamical system

$$\dot{\boldsymbol{x}}(t;\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{x}(t;\boldsymbol{\mu}), \boldsymbol{u}(t;\boldsymbol{\mu});\boldsymbol{\mu}), \quad \boldsymbol{x}(0;\boldsymbol{\mu}) = \boldsymbol{x}_0(\boldsymbol{\mu}), \tag{2.1}$$

with state $\boldsymbol{x}(t;\boldsymbol{\mu}) \in \mathbb{R}^N$, input $\boldsymbol{u}(t;\boldsymbol{\mu}) \in \mathbb{R}^{N_u}$, time $t \in [0,T]$, and parameter $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d$. We assume the nonlinear function $\boldsymbol{f} : \mathbb{R}^N \times \mathbb{R}^{N_u} \times \mathcal{D} \to \mathbb{R}^N$ to be polynomial of order $l \in \mathbb{N}_0$ in the state $\boldsymbol{x}$ and linear in the input signal $\boldsymbol{u}$, so there exist $\mathbf{A}_i(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N_i}$ with $N_i = \binom{N+i-1}{i}$ for $i \in \mathcal{I}$, and $\mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N_u}$ for $\boldsymbol{\mu} \in \mathcal{D}$ such that

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u};\boldsymbol{\mu}) = \sum_{i \in \mathcal{I}} \mathbf{A}_i(\boldsymbol{\mu})\boldsymbol{x}^i + \mathbf{B}(\boldsymbol{\mu})\boldsymbol{u}, \tag{2.2}$$

with the degree set $\mathcal{I} \subset \{0, 1, \ldots, l\}$. The vector $\boldsymbol{x}^i \in \mathbb{R}^{N_i}$ comprises all unique products of i entries of $\boldsymbol{x}$ ($\boldsymbol{x}^0$ is scalar 1). We can construct this vector from the i-fold Kronecker product $\boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x} =: \boldsymbol{x}^{i\otimes}$ by removing all duplicate entries due to commutativity of the multiplication [19]. Formally, we can write $\boldsymbol{x}^i = \mathbf{I}_N^{(i)} \boldsymbol{x}^{i\otimes}$ with $\mathbf{I}_N^{(i)} \in \{0,1\}^{N_i \times N^i}$ an identity matrix with certain rows removed and conversely $\boldsymbol{x}^{i\otimes} = \mathbf{J}_N^{(i)} \boldsymbol{x}^i$ with $\mathbf{J}_N^{(i)} \in \{0,1\}^{N^i \times N_i}$ an identity matrix with certain rows copied and permuted. Here, N^i denotes as usual the ith power of N.

2.2. Intrusive projection-based model reduction. To obtain a projection-based reduced order model (ROM), a common intrusive approach is to project the FOM (2.1) onto $\mathbf{V}_n$ and replace the FOM state $\boldsymbol{x}(t;\boldsymbol{\mu})$ by the lower-dimensional approximation $\mathbf{V}_n \tilde{\boldsymbol{x}}(t;\boldsymbol{\mu})$,

$$\dot{\tilde{\boldsymbol{x}}}(t;\boldsymbol{\mu}) = \mathbf{V}_n^T \boldsymbol{f}(\mathbf{V}_n \tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}), \boldsymbol{u}(t;\boldsymbol{\mu});\boldsymbol{\mu}), \quad \tilde{\boldsymbol{x}}(0;\boldsymbol{\mu}) = \mathbf{V}_n^T \boldsymbol{x}_0(\boldsymbol{\mu}). \tag{2.3}$$

Here, $\mathbf{V}_n = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{N \times n}$ is an orthonormal basis of an n-dimensional subspace of $\mathbb{R}^N$ with $n \ll N$, and $\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}) \in \mathbb{R}^n$ the ROM state. The orthonormal basis $\mathbf{V}_n$ can, for example, be obtained from proper orthogonal decomposition (POD) [11] of snapshots $\boldsymbol{x}_k(\boldsymbol{\mu}) \in \mathbb{R}^N$, $k = 0, \ldots, K_{\text{POD}} - 1$ of a time discretization of the FOM (2.1). Other methods to construct reduced bases and details on how to sample parameters and the ROM dimension n are discussed in [4].

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

To allow efficient simulations of the ROM (2.3), the cost of evaluating the right-hand side must be independent of the FOM dimension N. For this purpose, the intrusive approach is to precompute reduced operators,

$$\tilde{\mathbf{B}}(\boldsymbol{\mu}) := \mathbf{V}_n^T \mathbf{B}(\boldsymbol{\mu}) \quad \in \mathbb{R}^{n \times N_u}, \tag{2.4}$$

and

$$\tilde{\mathbf{A}}_i(\boldsymbol{\mu}) := \mathbf{V}_n^T \mathbf{A}_i(\boldsymbol{\mu}) \mathbf{I}_N^{(i)} \mathbf{V}_n^{i\otimes} \mathbf{J}_N^{(i)} \in \mathbb{R}^{n \times n_i}, \quad \text{with } n_i = \binom{n+i-1}{i}, \text{ for all } i \in \mathcal{I}, \tag{2.5}$$

whose dimensions only depend on the ROM dimension n (and the unreduced input signal dimension N_u) [18]. Note that these precomputations must in general be performed for each choice of the parameter $\boldsymbol{\mu} \in \mathcal{D}$ separately. The resulting efficient ROM for fixed $\boldsymbol{\mu}$ is given by

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}(t;\boldsymbol{\mu}) &= \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}), \boldsymbol{u}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \sum_{i \in \mathcal{I}} \tilde{\mathbf{A}}_i(\boldsymbol{\mu}) \tilde{\boldsymbol{x}}^i(t;\boldsymbol{\mu}) + \tilde{\mathbf{B}}(\boldsymbol{\mu}) \boldsymbol{u}(t;\boldsymbol{\mu}) \\ &= \tilde{\mathbf{O}}(\boldsymbol{\mu}) \boldsymbol{p}(\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}), \boldsymbol{u}(t;\boldsymbol{\mu})), \end{aligned} \tag{2.6}$$

with the aggregated ROM operator $\tilde{\mathbf{O}}(\boldsymbol{\mu}) := [[\tilde{\mathbf{A}}_i(\boldsymbol{\mu})]_{i \in \mathcal{I}} \;\; \tilde{\mathbf{B}}(\boldsymbol{\mu})] \in \mathbb{R}^{n \times n_f}$, $n_f = n_p + N_u$, $n_p = \sum_{i \in \mathcal{I}} n_i$ and the aggregated ROM state (and input) polynomial

$$\boldsymbol{p} : \mathbb{R}^n \times \mathbb{R}^{N_u} \to \mathbb{R}^{n_f}, \quad \boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u}) = \begin{bmatrix} [\tilde{\boldsymbol{x}}^i]_{i \in \mathcal{I}} \\ \boldsymbol{u} \end{bmatrix}. \tag{2.7}$$

2.3. Operator inference. As described in the previous subsection, intrusive model reduction requires access to the FOM operators $\mathbf{A}_i(\boldsymbol{\mu})$, $\mathbf{B}(\boldsymbol{\mu})$ to precompute the ROM operators $\tilde{\mathbf{A}}_i(\boldsymbol{\mu})$, $\tilde{\mathbf{B}}(\boldsymbol{\mu})$ via (2.4) and (2.5). In practice, this access is not possible in many FOM solvers. Instead, operator inference aims to learn approximations $\hat{\mathbf{A}}_i(\boldsymbol{\mu}) \in \mathbb{R}^{n \times n_i}$, $\hat{\mathbf{B}}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times N_u}$ of the intrusive ROM operators from data [19]. These data consist of triples of a ROM state snapshot $\breve{\boldsymbol{x}}_k(\boldsymbol{\mu}) \in \mathbb{R}^n$, a ROM state time derivative snapshot $\dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu}) \in \mathbb{R}^n$, and an input signal snapshot $\boldsymbol{u}_k(\boldsymbol{\mu})$, $k = 0, \ldots, K-1$. For these triples, operator inference minimizes the residual of the ROM ODE (2.6) to infer the aggregated ROM operator,

$$\min_{\hat{\mathbf{O}}(\boldsymbol{\mu}) = [[\hat{\mathbf{A}}_i]_{i \in \mathcal{I}} \;\; \hat{\mathbf{B}}(\boldsymbol{\mu})]} \sum_{k=0}^{K-1} \left\| \hat{\mathbf{O}}(\boldsymbol{\mu}) \boldsymbol{p}(\breve{\boldsymbol{x}}_k(\boldsymbol{\mu}), \boldsymbol{u}_k(\boldsymbol{\mu})) - \dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu}) \right\|_2^2. \tag{2.8}$$

If snapshot triples are obtained from the intrusive ROM ODE (2.6) such that $\breve{\boldsymbol{x}}_k(\boldsymbol{\mu}) = \tilde{\boldsymbol{x}}(t_k;\boldsymbol{\mu})$, $\dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu}) = \dot{\tilde{\boldsymbol{x}}}(t_k;\boldsymbol{\mu})$, and $\boldsymbol{u}_k(\boldsymbol{\mu}) = \boldsymbol{u}(t_k;\boldsymbol{\mu})$ at time steps t_k, $k = 0, \ldots, K-1$, then the intrusive aggregated ROM operator $\tilde{\mathbf{O}}(\boldsymbol{\mu}) = [[\tilde{\mathbf{A}}_i]_{i \in \mathcal{I}} \;\; \tilde{\mathbf{B}}(\boldsymbol{\mu})]$ is a solution to this minimization problem.

In practice, however, snapshots of the ROM ODE (2.6) are usually not available, but only snapshots of time discretizations of the FOM (2.1), $\boldsymbol{x}_k(\boldsymbol{\mu}) \in \mathbb{R}^N$, $\boldsymbol{u}_k(\boldsymbol{\mu}) \in \mathbb{R}^{N_u}$ corresponding to time steps t_k, $k = 0, \ldots, K$. These snapshots can be used to approximate the ROM snapshots $\breve{\boldsymbol{x}}_k(\boldsymbol{\mu})$ in (2.8) by projecting onto the ROM basis,

$$\breve{\boldsymbol{x}}_k(\boldsymbol{\mu}) = \mathbf{V}_n^T \boldsymbol{x}_k(\boldsymbol{\mu}), \quad k = 0, \ldots, K, \tag{2.9}$$

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

and to approximate the time derivative snapshots $\dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu})$ via some finite difference, in the simplest case explicit Euler,

$$\dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu}) = \frac{\breve{\boldsymbol{x}}_{k+1}(\boldsymbol{\mu}) - \breve{\boldsymbol{x}}_k(\boldsymbol{\mu})}{t_{k+1} - t_k}, \quad k = 0, \dots, K-1. \tag{2.10}$$

When snapshots (2.9) and (2.10) are used in the operator inference minimization problem (2.8), the inferred operator $\hat{\mathbf{O}}(\boldsymbol{\mu})$ resulting from solving this minimization problem is generally not equal to the intrusive operator $\tilde{\mathbf{O}}(\boldsymbol{\mu})$ but is only an approximation. This approach is nonintrusive because it does not require access to the FOM operators $\tilde{\mathbf{A}}_i(\boldsymbol{\mu})$, $\tilde{\mathbf{B}}(\boldsymbol{\mu})$ but only the knowledge about $\mathcal{I}$ and N_u and FOM snapshot data.

Note that operator inference for ROMs with POD bases take two sets of snapshots as input. The first set, the POD snapshot data, consists of FOM state snapshots $\boldsymbol{x}_k(\boldsymbol{\mu}) \in \mathbb{R}^N$, $k = 0, \dots, K_{\mathrm{POD}} - 1$ that are used to compute the POD basis $\mathbf{V}_n$ exactly in the same way as for the corresponding intrusive ROM with POD basis. The second set, the operator inference snapshot data, consists of triples of a ROM state snapshot $\breve{\boldsymbol{x}}_k \in \mathbb{R}^n$, a ROM state time derivative snapshot $\dot{\breve{\boldsymbol{x}}}_k \in \mathbb{R}^n$, and an input signal snapshot $\boldsymbol{u}_k \in \mathbb{R}^{N_u}$. This set defines the minimization problem (2.8).

Note further that, in general, the inference of the ROM operators by solving (2.8) must be performed separately for each choice of the parameter $\boldsymbol{\mu}$, just like the intrusive operator precomputations (2.4) and (2.5). For parameter values for which inference data are not available, the corresponding operators are then obtained by interpolation of inferred operators [19]. Given certain assumptions, the parameter dependency of the ROM operators can explicitly be incorporated in the operator inference minimization problem (2.8) [16]. In this work, we consider the parameter $\boldsymbol{\mu}$ fixed and will omit it in the following sections.

2.4. Insufficiently informative snapshot data. As has been observed by several researchers [7, 15, 18], standard operator inference via (2.8) often fails to accurately reproduce the intrusive ROM operators. There are several causes of such failures, such as inconsistency between the FOM time discretization and the discretization of the time derivative snapshots $\dot{\breve{\boldsymbol{x}}}_k(\boldsymbol{\mu})$ or non-Markovian information in the snapshot data [18]. In this work, we want to highlight the remaining cause that has not been resolved yet: insufficiently informative snapshot data. For this purpose, we write (2.8) equivalently as

$$\min_{\hat{\mathbf{O}} \in \mathbb{R}^{n \times n_f}} \left\| \hat{\mathbf{O}}\mathbf{P} - \dot{\breve{\mathbf{X}}} \right\|_F^2, \tag{2.11}$$

with

$$\mathbf{P} := [\boldsymbol{p}(\breve{\boldsymbol{x}}_0, \boldsymbol{u}_0) \ \dots \ \boldsymbol{p}(\breve{\boldsymbol{x}}_{K-1}, \boldsymbol{u}_{K-1})] \in \mathbb{R}^{n_f \times K} \quad \text{and}$$
$$\dot{\breve{\mathbf{X}}} := \left[\dot{\breve{\boldsymbol{x}}}_0 \ \dots \ \dot{\breve{\boldsymbol{x}}}_{K-1} \right] \in \mathbb{R}^{n \times K}. \tag{2.12}$$

This optimization problem has a unique solution $\hat{\mathbf{O}}$ if and only if the matrix $\mathbf{P}$ has full rank n_f [18]. To compute this solution numerically, also the numerical rank of $\mathbf{P}$—defined as the number of singular values larger than a specified tolerance—must be n_f. In general, there is no guarantee that this requirement is satisfied even if the number of snapshots K is much larger than n_f. As described in [18], strategies to mitigate such rank deficiency include regularization and reduction of the ROM

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

dimension n. However, these strategies do not guarantee accurate reconstruction of the intrusive ROM operators. Regularization can limit the accuracy of the resulting operators. Indeed, many works on regularized operator inference report significant differences between the nonintrusive ROMs and their intrusive counterparts [3, 16, 25]. Reducing the ROM dimension, on the other hand, can result in a least squares matrix of even smaller rank than the initial matrix. Therefore, iterative reduction of the ROM dimension can result in a significantly smaller dimension n corresponding to an intrusive ROM of insufficient accuracy.

In the numerical experiments in [18], full numerical rank has been achieved by generating and concatenating multiple trajectories of snapshots with randomized input signals. However, the number and length of the trajectories and the distributions of the random input signals are hyperparameters that have to be tuned, and whether the resulting least squares matrix has full numerical rank can only be checked a posteriori.

3. Exact operator inference. In this work, we propose a novel method to generate snapshots that is a priori guaranteed to result in a full rank matrix $\mathbf{P}$. Our method has two key ingredients. First, in section 3.1, we leverage the knowledge about how the matrix $\mathbf{P}$ depends on the ROM states $\breve{\boldsymbol{x}}_k$ and input signals $\boldsymbol{u}_k$ via (2.12) and (2.7) to find pairs of ROM states $\bar{\boldsymbol{x}}_s \in \mathbb{R}^n$ and input signals $\bar{\boldsymbol{u}}_s \in \mathbb{R}^{N_u}$, $s = 1, \dots, n_f$ that ensure full rank of $\mathbf{P}$. Second, in section 3.2, we generate time derivative snapshots $\dot{\bar{\boldsymbol{x}}}_s$ corresponding to these rank-ensuring pairs $(\bar{\boldsymbol{x}}_s, \bar{\boldsymbol{u}}_s)$ by simulating n_f trajectories for only a single time step each, with initial conditions $\bar{\boldsymbol{x}}_s$ and input signals $\bar{\boldsymbol{u}}_s$.

3.1. Rank-ensuring ROM states and input signals. The motivation for our choice of rank-ensuring ROM states is to isolate single entries of the aggregated polynomial $\boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u})$ defined in (2.7). If we could find inputs $\tilde{\boldsymbol{x}}$ and $\boldsymbol{u}$ such that $\boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u})$ is 1 in the jth entry and zero in all other entries for all $j = 1, \dots, n_f$, then the product $\hat{\mathbf{O}}\boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u})$ would equal the jth column of the operator $\hat{\mathbf{O}}$. Hence, we could easily infer all columns of the operator $\hat{\mathbf{O}}$.

Example 1. If $\mathcal{I} = \{1\}$ and $N_u = 0$, so $\boldsymbol{p}(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}$, we can achieve such an isolation of entries by using the unit vectors in $\mathbb{R}^n$, here denoted by $\boldsymbol{e}_j$, $j = 1, \dots, n$, as inputs $\tilde{\boldsymbol{x}}$.

Example 2. If $\mathcal{I} = \{2\}$ and $N_u = 0$, so $\boldsymbol{p}(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^2$, perfect isolation of entries is not possible. For example, when choosing $\tilde{\boldsymbol{x}} = [\tilde{x}_1, \dots, \tilde{x}_n]^T = \boldsymbol{e}_i + \boldsymbol{e}_j$ such that $\boldsymbol{p}(\tilde{\boldsymbol{x}})$ is 1 in the entry $\tilde{x}_i\tilde{x}_j$ with $i \neq j$, also the entries $\tilde{x}_i^2$ and $\tilde{x}_j^2$ are 1. However, the columns of $\hat{\mathbf{O}}$ corresponding to $\tilde{x}_i\tilde{x}_j$, $\tilde{x}_i^2$, and $\tilde{x}_j^2$ can still be inferred when employing the data $\hat{\mathbf{O}}\boldsymbol{p}(\boldsymbol{e}_i + \boldsymbol{e}_j)$, $\hat{\mathbf{O}}\boldsymbol{p}(2\boldsymbol{e}_i)$, and $\hat{\mathbf{O}}\boldsymbol{p}(2\boldsymbol{e}_j)$.

Based on this motivation, we define for arbitrary $i \in \mathbb{N}^+$ the set $\mathcal{X}^i$ that consists of all sums of i of these vectors,

$$\mathcal{X}^i := \left\{ \sum_{j=1}^{i} \boldsymbol{a}_j \,\middle|\, \boldsymbol{a}_1, \dots, \boldsymbol{a}_i \in \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\} \right\} \quad \text{and consistently } \mathcal{X}^0 := \{\mathbf{0} \in \mathbb{R}^n\}. \tag{3.1}$$

The main result of this article is as follows: the vectors in $\cup_{i \in \mathcal{I}} \mathcal{X}^i$ induce full rank of the matrix $\mathbf{P}$ as defined in (2.12) and (2.7) for any degree set $\mathcal{I} \subset \mathbb{N}_0$ and $N_u = 0$.

If $N_u > 0$, we achieve full rank of $\mathbf{P}$ by pairing these ROM states $\bar{\boldsymbol{x}}_j$ with zero input signals, $(\bar{\boldsymbol{x}}_j, \mathbf{0})$, and adding the pairs $(\mathbf{0}, \boldsymbol{u}_j)$ with zero ROM state and $\boldsymbol{u}_j$,

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

$j = 1, \ldots, N_u$ the unit vectors in $\mathbb{R}^{N_u}$. This way, we get the rank-ensuring snapshot pairs

$$\{(\bar{\boldsymbol{x}}_j, \bar{\boldsymbol{u}}_j)\}_{j=1}^{n_f} = \left\{(\bar{\boldsymbol{x}}_j, \boldsymbol{0}) | \tilde{\boldsymbol{x}}_j \in \cup_{i \in \mathcal{I}} \mathcal{X}^i\right\} \cup \{(\boldsymbol{0}, \boldsymbol{u}_j)\}_{j=1}^{N_u}. \tag{3.2}$$

Example 3. For $n = 2$, $N_u = 2$, and $\mathcal{I} = \{1, 2\}$ (such that $n_p = 5$, $n_f = 7$), we get the pairs

$$\underbrace{\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right), \left(\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right)}_{\{(\bar{\boldsymbol{x}}_j, \boldsymbol{0}) | \tilde{\boldsymbol{x}}_j \in \mathcal{X}^1\}}, \underbrace{\left(\begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right), \left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right), \left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}\right)}_{\{(\bar{\boldsymbol{x}}_j, \boldsymbol{0}) | \tilde{\boldsymbol{x}}_j \in \mathcal{X}^2\}},$$
$$\underbrace{\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right), \left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right)}_{\{(\boldsymbol{0}, \boldsymbol{u}_j)\}_{j=1}^{N_u}},$$

that result in the full-rank matrix

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where} \quad \boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u}) = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_1^2 \\ \tilde{x}_1 \tilde{x}_2 \\ \tilde{x}_2^2 \\ u_1 \\ u_2 \end{bmatrix}.$$

In general, the following theorem holds.

THEOREM 3.1 (Full rank of matrix $\mathbf{P}$). *For any $n \in \mathbb{N}^+$, $N_u \in \mathbb{N}_0$, and $\mathcal{I} \subset \mathbb{N}_0$, the matrix $\mathbf{P} := [\boldsymbol{p}(\breve{\boldsymbol{x}}_0, \boldsymbol{u}_0) \quad \ldots \quad \boldsymbol{p}(\breve{\boldsymbol{x}}_{K-1}, \boldsymbol{u}_{K-1})] \in \mathbb{R}^{n_f \times K}$ defined in* (2.12) *with $K = n_f$, $\boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u}) = \left[{}^{[\tilde{\boldsymbol{x}}^i]_{i \in \mathcal{I}}}_{\boldsymbol{u}}\right]$ as defined in* (2.7) *and with the ROM states and input signals $\{(\bar{\boldsymbol{x}}_j, \bar{\boldsymbol{u}}_j)\}_{j=1}^{n_f} = \{(\bar{\boldsymbol{x}}_j, \boldsymbol{0}) | \tilde{\boldsymbol{x}}_j \in \cup_{i \in \mathcal{I}} \mathcal{X}^i\} \cup \{(\boldsymbol{0}, \boldsymbol{u}_j)\}_{j=1}^{N_u}$ defined in* (3.2) *has full rank.*

The proof is given in Appendix A. For moderate values of n, N_u, and $\mathcal{I}$ which are relevant in practice, the rank and the numerical rank of $\mathbf{P}$ coincide, so we will only refer to the rank in the following.

Note that this choice of ROM states and input signals results in a matrix $\mathbf{P}$ that has as many columns as rows. This means that our choice uses the minimal number of snapshots necessary to achieve full rank. In addition, because $\mathbf{P}$ is square, the operator inference minimization problem (2.11) is equivalent to the matrix-valued linear system

$$\hat{\mathbf{O}}\mathbf{P} = \dot{\hat{\mathbf{X}}}. \tag{3.3}$$

Since the matrix $\mathbf{P}$ has a smaller condition number than the matrix $\mathbf{P}\mathbf{P}^T$ of the normal equations

$$\hat{\mathbf{O}}\mathbf{P}\mathbf{P}^T = \dot{\hat{\mathbf{X}}}\mathbf{P}^T \tag{3.4}$$

corresponding to the least squares problem (2.11), the linear system (3.3) is in general less sensitive to error amplifications.

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

3.2. Snapshot data generation via single-step ensemble. To formulate the operator inference minimization problem (2.8), we not only need the pairs of ROM states and input signals denoted by $(\bar{\boldsymbol{x}}_j, \bar{\boldsymbol{u}}_j)$, $j = 1, \ldots, n_f$ defined in (3.2), but also approximations of the ROM state time derivative $\dot{\bar{\boldsymbol{x}}}_j$ that correspond to these pairs. In existing works on operator inference [7, 15, 18, 19], triples of these quantities are obtained by projection of the corresponding FOM vector that are generated along trajectories of the FOM. Hence, the FOM state $\boldsymbol{x}_j$ at a given time step t_j depends on the FOM state $\boldsymbol{x}_{j-1}$ at the previous time step t_{j-1} and the FOM dynamics (2.2). These FOM dynamics are unknown (otherwise we would not need to perform operator inference). Consequently, we cannot predict or control which FOM states are attained along such trajectories. In particular, we cannot guarantee that FOM states corresponding to the ROM states $\bar{\boldsymbol{x}}_j$ described in the previous section are attained.

In our approach, we do not generate FOM snapshot data along one (or few) long trajectories but instead use an ensemble of many short trajectories, each containing *only a single time step.* Each trajectory is initialized with

$$\begin{aligned} &\text{the rank-ensuring initial condition } \boldsymbol{x}_0^s \\ &\quad = \mathbf{V}_n \bar{\boldsymbol{x}}_s, \text{ and input signal } \boldsymbol{u}_0^s = \bar{\boldsymbol{u}}_s,\ s = 1, \ldots, n_f, \end{aligned} \tag{3.5}$$

with $\bar{\boldsymbol{x}}_s$ and $\bar{\boldsymbol{u}}_s$ as defined in (3.2) and integrated in time for only a single explicit Euler step of size Δt. This approach results in FOM snapshots

$$\boldsymbol{x}_1^s = \boldsymbol{x}_0^s + \Delta t \boldsymbol{f}(\boldsymbol{x}_0^s, \boldsymbol{u}_0^s), \qquad s = 1, \ldots, n_f. \tag{3.6}$$

From each pair of FOM snapshots $\boldsymbol{x}_0^s$ and $\boldsymbol{x}_1^s$, we can compute the ROM time derivative snapshot

$$\dot{\bar{\boldsymbol{x}}}_s = \mathbf{V}_n^T \frac{\boldsymbol{x}_1^s - \boldsymbol{x}_0^s}{\Delta t}, \qquad s = 1, \ldots, n_f. \tag{3.7}$$

The resulting triples $(\bar{\boldsymbol{x}}_s, \dot{\bar{\boldsymbol{x}}}_s, \boldsymbol{u}_s)_{s=1}^{n_f}$ form the input to the operator inference optimization problem (2.8). Since the matrix $\mathbf{P}$ (2.12) only depends on the snapshots $\bar{\boldsymbol{x}}_s$ and $\boldsymbol{u}_s$, $s = 1, \ldots, n_f$, it does not depend on the unknown FOM operators. Hence, we can guarantee full rank of $\mathbf{P}$ by the choice of the initial conditions and input signals.

Note that our novel approach for generating snapshot data for operator inference does not affect the snapshot data used for POD, so the POD basis $\mathbf{V}_n$ remains unchanged. For example, the implicit Euler method is used to generate POD snapshot data in section 4.2.

The idea to generate snapshot data in multiple trajectories, each containing only a single time step, has already been used in numerical experiments in [19]. However, randomized instead of well-chosen initial conditions are used in that work. Later, FOM snapshots projected onto the ROM space have been used as initial conditions in the context of lifted nonlinearities [21] and noisy data [29]. All three approaches enable checking the full rank a priori but do not guarantee full rank. Indeed, far more than n_f trajectories have been used to achieve full rank of $\mathbf{P}$ in the numerical experiments in these works.

The complete method is summarized in Algorithm 3.1.

3.3. Exact reconstruction of the intrusive ROM operators. By combining the rank-ensuring initial conditions and input signals in section 3.1 and the single-step ensemble in section 3.2, our novel approach exactly reconstructs the intrusive ROM operators, as stated by the following theorem.

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Algorithm 3.1. Exact operator inference.

procedure EXACTOPINF(
ROM basis $\mathbf{V}_n \in \mathbb{R}^{N \times n}$,
degree set $\mathcal{I} \subset \mathbb{N}_0$,
input signal dimension $N_u \in \mathbb{N}_0$
time step size Δt
)
  `# Generate rank-ensuring snapshot data.`
  Initialize $\mathbf{P} \leftarrow \emptyset$
  Initialize $\dot{\breve{\mathbf{X}}} \leftarrow \emptyset$
  for $i \in \mathcal{I}$ **do**
    `# Generate ROM state sets` $\mathcal{X}^i$ `as defined in (3.1) (see Appendix B)`
    $\mathcal{X}^i \leftarrow$ GENERATEROMSTATESET(ROM dimension n, polynomial degree i)
  end for
  for $\bar{\boldsymbol{x}} \in \cup_{i \in \mathcal{I}} \mathcal{X}^i$ **do**
    $\boldsymbol{x}_1 \leftarrow$ FOM(initial condition $\boldsymbol{x}_0 = \mathbf{V}_n \bar{\boldsymbol{x}}$,
      input signal $\boldsymbol{u} = \mathbf{0}$,
      time step size Δt, end time $T = \Delta t$, explicit Euler time discretization)
    $\mathbf{P} \leftarrow [\mathbf{P},\ \boldsymbol{p}(\tilde{\boldsymbol{x}}_0, \mathbf{0})]$
    $\dot{\breve{\mathbf{X}}} \leftarrow [\dot{\breve{\mathbf{X}}},\ \frac{\mathbf{V}_n^T \boldsymbol{x}_1 - \bar{\boldsymbol{x}}}{\Delta t}]$
  end for
  for $j = 1, \ldots, N_u$ **do**
    $\boldsymbol{u} \leftarrow j$-th unit vector in $\mathbb{R}^{N_u}$
    $\boldsymbol{x}_1 \leftarrow$ FOM(initial condition $\boldsymbol{x}_0 = \mathbf{0}$,
      input signal $\boldsymbol{u}$,
      time step size Δt, end time $T = \Delta t$, explicit Euler time discretization)
    $\mathbf{P} \leftarrow [\mathbf{P},\ \boldsymbol{p}(\mathbf{0}, \boldsymbol{u})]$
    $\dot{\breve{\mathbf{X}}} \leftarrow [\dot{\breve{\mathbf{X}}},\ \frac{\mathbf{V}_n^T \boldsymbol{x}_1 - \mathbf{0}}{\Delta t}]$
  end for
  Solve linear system (3.3)
  return $\hat{\mathbf{O}}$
end procedure

THEOREM 3.2 (Exact reconstruction of intrusive ROM operators). *For any* $n \in \mathbb{N}^+$, $N_u \in \mathbb{N}_0$, $\mathcal{I} \subset \mathbb{N}_0$, *and* $\Delta t > 0$, *the operator inference minimization problem* (2.8) *with snapshot data* $(\breve{\bar{\boldsymbol{x}}}_k, \dot{\breve{\bar{\boldsymbol{x}}}}_k, \boldsymbol{u}_k) = (\bar{\boldsymbol{x}}_s, \dot{\bar{\boldsymbol{x}}}_s, \bar{\boldsymbol{u}}_s)$ *defined in* (3.2) *and* (3.7) *has a unique solution. This solution is equivalent to the intrusive ROM operators defined in* (2.4) *and* (2.5).

Proof. We first show that the FOM snapshot data generated in our proposed approach are equivalent to snapshot data generated from the intrusive ROM itself, which constitutes the ideal inference data. For this purpose, we insert the definition of the rank-ensuring initial conditions $\bar{\boldsymbol{x}}_s$ (3.5) and the explicit Euler FOM time step (3.6) into the definition of the explicit Euler time derivative snapshot (3.7) to find

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

$$\dot{\bar{\boldsymbol{x}}}_s = \frac{1}{\Delta t}\mathbf{V}_n^T\left(\boldsymbol{x}_0^s + \Delta t \boldsymbol{f}(\boldsymbol{x}_0^s, \boldsymbol{u}_0^s) - \boldsymbol{x}_0^s\right) = \mathbf{V}_n^T \boldsymbol{f}(\boldsymbol{x}_0^s, \boldsymbol{u}_0^s)$$
$$= \mathbf{V}_n^T \boldsymbol{f}(\mathbf{V}_n \bar{\boldsymbol{x}}_s, \boldsymbol{u}_0^s) = \tilde{\mathbf{O}}\boldsymbol{p}(\bar{\boldsymbol{x}}_s, \boldsymbol{u}_0^s). \tag{3.8}$$

Hence, (2.8) simplifies to

$$\min_{\hat{\mathbf{O}}} \sum_{s=1}^{n_f} \left\| \hat{\mathbf{O}}\boldsymbol{p}(\bar{\boldsymbol{x}}_s, \boldsymbol{u}_0^s) - \tilde{\mathbf{O}}\boldsymbol{p}(\bar{\boldsymbol{x}}_s, \boldsymbol{u}_0^s) \right\|_2^2, \tag{3.9}$$

or equivalently,

$$\min_{\hat{\mathbf{O}}} \left\| \hat{\mathbf{O}}\mathbf{P} - \tilde{\mathbf{O}}\mathbf{P} \right\|_F^2. \tag{3.10}$$

Since $\mathbf{P}$ has full rank as shown in Theorem 3.1, the solution $\hat{\mathbf{O}} = \tilde{\mathbf{O}}$ is unique, so the method exactly reconstructs the intrusive ROM operator. ∎

Note that our approach not only guarantees the full rank of $\mathbf{P}$ but also eliminates two other error sources common in operator inference. First, the consistency of using explicit Euler schemes for both the snapshot generation (3.6) and the computation of the time derivative snapshots (3.7) avoids the introduction of time discretization inconsistencies. Second, by performing only a single explicit Euler time step on each trajectory, we avoid introducing non-Markovian terms that would cause the projected FOM snapshot data (2.9) to deviate from actual ROM snapshot data, as described in [18].

Since our approach exactly reconstructs intrusive ROM operators, it also preserves any structure present in these intrusive operators such as symmetry and skew-symmetry [24]. In contrast to other works [7, 8, 12], these structures are preserved automatically without the need to enforce their preservation explicitly.

3.4. Assumptions on the FOM solver. Operator inference is a nonintrusive model reduction concept developed for FOM solvers that do not provide the access to FOM operators required by intrusive model reduction. Consequently, imposing strong assumptions on the FOM solver contradicts the core motivation for operator inference. In our proposed approach, we only impose two assumptions.

First, we assume that specific initial conditions can be imposed; concretely, linear combinations of the ROM basis modes (3.5). For POD bases, these modes are linear combinations of FOM snapshots. Since these snapshots are physically realizable, the specific initial conditions are physically realizable as well in most cases. Exceptions include models that only allow a certain range of state values, for example positivity of temperature, or require specific boundary conditions. These exceptions are beyond the scope of this article since they typically require some additional treatment already on the intrusive ROM level, for example the introduction of a lifting function [22] to satisfy boundary conditions.

Second, we assume that explicit Euler can be chosen as a time discretization method to generate the operator inference snapshot data. However, this restriction can be relaxed to higher-order Runge-Kutta schemes if input signals are interpolated between time steps. These schemes can be written as explicit Euler schemes whose right-hand side has higher polynomial degree [18]. Consequently, our approach can be applied to explicit Runge-Kutta methods (and multistep methods that use an explicit Runge-Kutta method to compute the first time step). For implicit methods, on the other hand, the full rank of $\mathbf{P}$ might not be guaranteed. For example, if the FOM time steps (3.6) are replaced by implicit Euler steps,

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

$$\boldsymbol{x}_1^s = \boldsymbol{x}_0^s + \Delta t \boldsymbol{f}(\boldsymbol{x}_1^s, \boldsymbol{u}_1^s), \tag{3.11}$$

then the matrix $\mathbf{P}$ should, for time discretization consistency, be composed of vectors $\boldsymbol{p}(\boldsymbol{x}_1^s, \boldsymbol{u}_1^s)$ instead of $\boldsymbol{p}(\boldsymbol{x}_0^s, \boldsymbol{u}_0^s)$. However, in contrast to the initial conditions $\boldsymbol{x}_0^s$, we do not have full control over the states $\boldsymbol{x}_1^s$ after one time step. Consequently, we generally cannot guarantee the full rank of $\mathbf{P}$. Alternatively, we can construct the matrix $\mathbf{P}$ from the vectors $\boldsymbol{p}(\boldsymbol{x}_0^s, \boldsymbol{u}_0^s)$. This way, we guarantee the full rank of $\mathbf{P}$ but introduce a time discretization inconsistency that impedes exact reconstruction of the intrusive ROM operators.

Note that this requirement on the time discretization only affects the generation of operator inference snapshot data but neither the generation of POD snapshot data nor the simulations that are performed with the resulting ROM. The POD snapshot data can be generated from any FOM time discretization, and—like intrusive ROM operators—the inferred operators can be used for any ROM time discretization.

Note further that the time step size Δt is provided as an input in Algorithm 3.1 only for ease of presentation. Our proposed approach is also applicable to a FOM solver that does not allow the users to specify the time step size themselves. In such cases, Δt is not provided as an input, and the end time T cannot be chosen to be equal to Δt. Consequently, we can generally not guarantee that the FOM solver only performs a single time step, but by setting T to a small value, we can limit the number of excess time steps. From the solver output, we then only use the state snapshot $\boldsymbol{x}_1$ after the first time step and the corresponding time t_1 to compute the time step size Δt.

While our approach differs decisively from the reprojection approach [18] in terms of control over which snapshots are generated, both approaches are very similar in terms of intrusiveness. In fact, our approach can be seen as the reprojection approach performed for n_f separate trajectories but stopped after a single time step and before the first reprojection step. Consequently, the described assumptions on the FOM solver are almost the same as for the reprojection approach. The only exception are implicit time discretization schemes for linear models which are allowed in reprojection but not in our approach because they inhibit control over the generated snapshots.

3.5. Remarks on changing ROM dimension. When ROMs are used in practice, it can be useful to change their dimension. For instance, the ROM dimension might be increased to improve accuracy or reduced to improve computational efficiency. The proposed data generation method is flexible toward such dimension changes. Concretely, the set $\mathbf{X}_n$ of FOM snapshots $\boldsymbol{x}_1^s$ as defined in (3.6) for any ROM dimension $n \in \mathbb{N}$ is a subset of the corresponding set $\mathbf{X}_{n+1}$ of FOM snapshots for dimension $n+1$. Generally, $\mathbf{X}_{n_-} \subset \mathbf{X}_n \subset \mathbf{X}_{n_+}$ for all $n_- < n < n_+$. Consequently, we do not need to generate new data to infer ROM operators of dimension n_- when the snapshot data $\mathbf{X}_n$ are already generated but can select the subset $\mathbf{X}_{n_-}$ of $\mathbf{X}_n$. Likewise, we do not need to generate the complete snapshot data $\mathbf{X}_{n_+}$ when $\mathbf{X}_n$ is given but can completely reuse $\mathbf{X}_n$ and only have to generate $\mathbf{X}_{n_+} \setminus \mathbf{X}_n$ additionally.

Note that reprojection data [18] do not have this flexibility because the reprojection process is specific to a fixed ROM dimension.

3.6. Choice of time step size Δt. As stated in Theorem 3.2, our proposed method achieves exact reconstruction for any $\Delta t > 0$. Indeed, in exact arithmetic, the size of Δt also does not affect the accuracy of the inferred operators since the effect of Δt in computing $\boldsymbol{x}_1^s$ in (3.6) cancels out when computing the time derivative snapshot $\dot{\boldsymbol{x}}_1^s$ in (3.7). In floating point arithmetic, however, the accuracy of the inferred

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

operators depends on how accurate the intermediate results can be represented as floating-point numbers. If Δt is determined by the FOM solver, we have to accept the inaccuracy in the ROM operator reconstruction induced by this limitations of floating-point number representation. However, if Δt can be specified by the user, we can choose Δt such that these limitations are mitigated.

Concretely, the differences $\breve{\boldsymbol{x}}_1^s - \breve{\boldsymbol{x}}_0^s$ cannot be represented well as floating-point numbers if $\|\tilde{\mathbf{O}}\|_2$ is very large or very small. To counteract this effect, $\Delta t = \frac{1}{\|\tilde{\mathbf{O}}\|_2}$ would be a good choice. Since we do not know $\tilde{\mathbf{O}}$ a priori, we can only approximate $\|\tilde{\mathbf{O}}\|_2$ based on the POD snapshot data.

In this work, we use the estimate

$$\Delta t = \left(\max_{k=0,\dots,K_{POD}-1} \frac{\left\| \boldsymbol{v}_1^T \frac{x_{k+1}-x_k}{t_{k+1}-t_k} \right\|_2}{\left\| \boldsymbol{p}(\boldsymbol{v}_1^T x_k, u(t_k)) \right\|_2} \right)^{-1}, \tag{3.12}$$

based on the lower bound

$$\begin{aligned} \left\| \tilde{\mathbf{O}} \right\|_2 := \sup_{\boldsymbol{y} \in \mathbb{R}^{n_f}} \frac{\left\| \tilde{\mathbf{O}} \boldsymbol{y} \right\|_2}{\|\boldsymbol{y}\|_2} &\geq \sup_{\tilde{\boldsymbol{x}} \in \mathbb{R}^n, \boldsymbol{u} \in \mathbb{R}^{N_u}} \frac{\left\| \tilde{\mathbf{O}} \boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u}) \right\|_2}{\|\boldsymbol{p}(\tilde{\boldsymbol{x}}, \boldsymbol{u})\|_2} \\ &\geq \max_{k=0,\dots,K_{POD}-1} \frac{\left\| \tilde{\mathbf{O}} \boldsymbol{p}(\mathbf{V}_n^T \boldsymbol{x}_k, \boldsymbol{u}(t_k)) \right\|_2}{\|\boldsymbol{p}(\mathbf{V}_n^T \boldsymbol{x}_k, \boldsymbol{u}(t_k))\|_2}, \end{aligned} \tag{3.13}$$

and the approximation

$$\tilde{\mathbf{O}} \boldsymbol{p}(\mathbf{V}_n^T \boldsymbol{x}_k, \boldsymbol{u}(t_k)) \approx \mathbf{V}_n^T \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{t_{k+1} - t_k}, \quad k = 0, \dots, K_{POD} - 1, \tag{3.14}$$

for ROM dimension $n = 1$, so $\mathbf{V}_n = \boldsymbol{v}_1$.

Note that we do not need to take stability into account when determining the time step size Δt since we only perform a single time step per trajectory.

4. Numerical experiments. The following numerical experiments demonstrate that Algorithm 3.1 exactly reconstructs the corresponding intrusive ROM operators. Many existing works on operator inference assess the accuracy of the inferred ROMs based on trajectory errors. As shown in [25], trajectory errors can be misleading because they are sensitive to the choice of trajectories. Furthermore, the objective of operator inference is to reconstruct the intrusive ROM, so this intrusive ROM should serve as ground truth, not the underlying FOM (see Appendix C). We therefore assess the reconstruction accuracy with relative operator errors $\frac{\|\hat{\mathbf{O}} - \tilde{\mathbf{O}}\|_F}{\|\tilde{\mathbf{O}}\|_F}$.

The code is available on https://github.com/h3rror/exactOpInf.

4.1. Chafee-Infante equation / Allen–Cahn equation. We consider the Chafee-Infante equation / Allen–Cahn equation in a setup based on [18, 25],

$$\frac{\partial}{\partial t} x(\xi, t) = \frac{\partial^2}{\partial \xi^2} x(\xi, t) - x^3(\xi, t) + x(\xi, t), \tag{4.1}$$

with spatial coordinate $\xi \in \Omega = (0, 1)$ and time $t \in [0, 0.1]$. We impose the boundary conditions

$$x(0, t) = u(t), \quad \frac{\partial}{\partial \xi} x(1, t) = 0, \qquad t \in [0, 0.1], \tag{4.2}$$

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

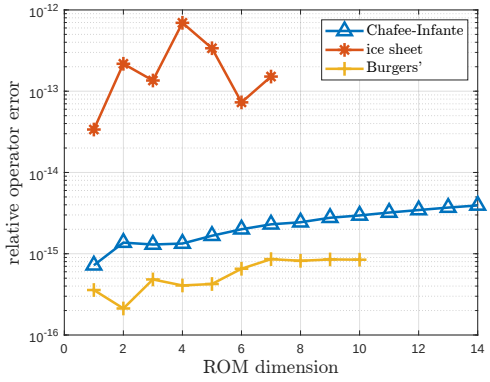

FIG. 1. *Relative operator errors* $\frac{\|\hat{\mathbf{O}}-\tilde{\mathbf{O}}\|_F}{\|\tilde{\mathbf{O}}\|_F}$.

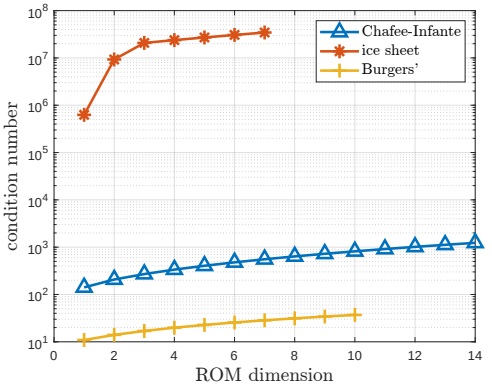

FIG. 2. *Condition numbers of data matrices* $\mathbf{P}$.

and zero initial condition $x(\xi,0)=0$ for $\xi\in\Omega$. We discretize with central finite differences on an equidistant grid in space with mesh width $\Delta\xi=2^{-7}$ and with explicit Euler in time with time step size $\Delta t=10^{-5}$ to obtain a time-discrete dynamical system of dimension $N=128$,

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\Delta t\left(\mathbf{A}_1\boldsymbol{x}_k+\mathbf{A}_2\boldsymbol{x}_k^2+\mathbf{A}_3\boldsymbol{x}_k^3+\mathbf{B}u_k\right),\quad k=0,\dots,K-1. \tag{4.3}$$

This system is a polynomial of degree $l=3$ with $\mathbf{A}_1\in\mathbb{R}^{N\times N}$, $\mathbf{A}_2\in\mathbb{R}^{N\times N_2}$ with $N_2=\frac{N(N+1)}{2}$, $\mathbf{A}_3\in\mathbb{R}^{N\times N_3}$ with $N_3=\frac{N(N+1)(N+2)}{6}$, $\mathbf{B}\in\mathbb{R}^{N\times N_u}$ with $N_u=1$, the input signal $u_k=10(\sin(k\pi\Delta t)+1)$, $k=0,\dots,K-1$, and $K=10^4$. The resulting 10^4+1 snapshots are used to compute the POD bases $\mathbf{V}_n\in\mathbb{R}^{N\times n}$ with $n=1,\dots,14$.

We apply Algorithm 3.1 with $\Delta t=3.2741\cdot10^{-5}$ according to estimate (3.12) and compute the relative operator errors for ROM dimensions $n=1,\dots,14$. Figure 1 shows that the operator error is on the order of machine precision, and Figure 2 shows that the data matrices $\mathbf{P}$ have moderate condition numbers.

A crucial advantage of our method is the minimal number of FOM time steps needed for exact reconstruction of the intrusive ROM. For ROM dimension $n=14$, we only need $n_f=n_1+n_2+n_3+N_u=680$ FOM time steps. This number is orders of magnitude smaller than in previous works with similar setups. For example, in [18], 10^7 FOM time steps with reprojection have been used to reconstruct the operators exactly for ROM dimension $n=12$. In [25], 10^5 FOM time steps are used to infer ROM operators for ROM dimension $n=14$ from a regularized operator inference minimization problem that significantly differ from the intrusive counterparts.

Note that $\mathbf{A}_2=\mathbf{0}$ in our discretization, so we could omit this term in the operator inference and would even need $n_2=105$ less FOM time steps. We refrain from this simplification to keep the comparison with [18] and [25] fair.

4.2. Ice sheet model. We consider the shallow ice equations in a setup based on [2],

$$\frac{\partial}{\partial t}x(\xi,t)=c_1\frac{\partial}{\partial\xi}\left(x^2(\xi,t)\frac{\partial}{\partial\xi}x(\xi,t)\right)+c_2\frac{\partial}{\partial\xi}\left(x^5(\xi,t)\left|\frac{\partial}{\partial\xi}x(\xi,t)\right|^2\frac{\partial}{\partial\xi}x(\xi,t)\right) \tag{4.4}$$

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

with the spatial coordinate $\xi \in [0, 1000]$, time $t \in [0, 2]$, and the constants $c_1 = 8.9 \cdot 10^{-13}$ and $c_2 = 2.8 \cdot 10^7$. We impose the initial condition

$$x(\xi, 0) = 10^{-2} + 630 \left(\frac{\xi}{2000} + 0.25 \right)^4 \left(\frac{\xi}{2000} - 0.75 \right)^4 \tag{4.5}$$

and homogeneous Neumann boundary conditions. We discretize with finite differences in space with mesh width $\Delta\xi = \frac{1000}{2^9}$, and with implicit Euler in time with time step size $\Delta t = 10^{-3}$, to obtain the time-discrete dynamical system of dimension $N = 512$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t \left(\mathbf{A}_3 \boldsymbol{x}_k^3 + \mathbf{A}_8 \boldsymbol{x}_k^8 \right), \quad k = 0, \ldots, K-1. \tag{4.6}$$

This system is polynomial of degree $l = 8$ with $\mathbf{A}_3 \in \mathbb{R}^{N \times N_3}$ with $N_3 = \frac{N(N+1)(N+2)}{6}$ and $\mathbf{A}_8 \in \mathbb{R}^{N \times N_8}$ with $N_8 = \binom{N+7}{8}$ and $K = 2000$. The resulting 2001 snapshots from $t = 0$ to $t = 2$ are used to compute the POD bases $\mathbf{V}_n \in \mathbb{R}^{N \times n}$ with $n = 1, \ldots, 7$.

We apply Algorithm 3.1 with $\Delta t = 1.4773 \cdot 10^{13}$ according to estimate (3.12). This extreme value can be explained by the extreme operator norm $\|\tilde{\mathbf{O}}\|_2 = 4.271 \cdot 10^{-14}$ for ROM dimension $n = 1$. The relative operator errors for ROM dimensions $n = 1, \ldots, 7$ are depicted in Figure 1. Like in the previous test case, these operator errors are essentially on the order of machine precision. The fact that these errors are a few orders of magnitude larger than the errors in the previous test case is presumably related to the larger condition numbers as depicted in Figure 2.

For ROM dimension $n = 7$, the proposed method performs $n_f = 3087$ FOM time steps. This number is larger than the 2001 FOM time steps with re-projection reported in [2]. However, the authors also report that this budget of snapshots is depleted before the inference of the ROM operators of dimension $n = 6$ is completed. Hence, their ROM operators of dimensions $n = 6$ and $n = 7$ are partially computed with regularized operator inference and snapshot data polluted with non-Markovian terms. Consequently, their inferred operators of dimension $n = 6$ and $n = 7$ are not reconstructed exactly. Their operators of dimension $n \leq 5$ might be reconstructed exactly, but the used 2001 snapshots are more than the $n_f = 365$ snapshots used by our method for dimension $n = 5$.

Note that the cubic term could be neglected in practice due to the extremely small value of c_1. This simplification would reduce the number of required FOM time steps further.

4.3. Burgers' equation. We consider the viscous Burgers' equation in a setup based on [12, 18],

$$\frac{\partial}{\partial t} x(\xi, t) + x(\xi, t) \frac{\partial}{\partial \xi} x(\xi, t) - \frac{\partial^2}{\partial \xi^2} x(\xi, t) = 0, \tag{4.7}$$

with spatial coordinate $\xi \in \Omega = (-1, 1)$, time $t \in [0, 1]$. We impose periodic boundary conditions and the initial condition $x(\xi, 0) = -\sin(\pi\xi)$ for all $\xi \in \Omega$.

We discretize with finite differences on an equidistant grid in space with mesh width $\Delta\xi = 2^{-6}$, and with explicit Euler in time with step size $\Delta t = 10^{-4}$, to obtain the time-discrete dynamical system of dimension $N = 128$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t \left(\mathbf{A}_1 \boldsymbol{x}_k + \mathbf{A}_2 \boldsymbol{x}_k^2 \right), \quad k = 0, \ldots, K-1. \tag{4.8}$$

This system is polynomial of degree $l = 2$ with $\mathbf{A}_1 \in \mathbb{R}^{N \times N}$, $\mathbf{A}_2 \in \mathbb{R}^{N \times N_2}$ with $N_2 = \frac{N(N+1)}{2}$ and $K = 10^4$. Concretely, we use a spatial discretization as described

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

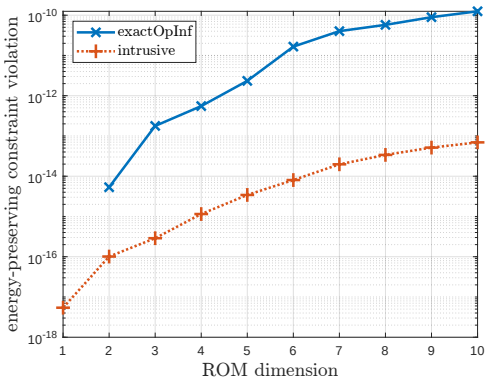

FIG. 3. *Violation of the energy-preserving constraint of the convection operator of Burgers' equation.*

in [30] such that the diffusion matrix $\mathbf{A}_1$ is symmetric and negative semidefinite, and the convection operator $\mathbf{A}_2$ satisfies the skew-symmetry property $\boldsymbol{x}^T\mathbf{A}_2\boldsymbol{x}^2 = 0$ for all $\boldsymbol{x} \in \mathbb{R}^N$. The resulting $10^4 + 1$ snapshots are used to compute the POD bases $\mathbf{V}_n \in \mathbb{R}^{N\times n}$, with $n = 1, \ldots, 10$.

We perform Algorithm 3.1 with $\Delta t = 1.0134 \cdot 10^{-1}$ according to estimate (3.12) and compute the relative operator errors for ROM dimensions $n = 1, \ldots, 10$. Figure 1 shows operator errors on the order of machine precison, and Figure 2 shows very small condition numbers. In addition, we observe that both the intrusive and the inferred operators preserve the skew-symmetry of the FOM convection operator and the symmetry and negative semidefiniteness of the FOM diffusion operator. Figure 3 shows the energy-preserving constraint violation $\sum_{1\le i,j,k\le n} |\hat{h}_{ijk} + \hat{h}_{jik} + \hat{h}_{kji}|$ as defined in [12] for the inferred operator with $\hat{h}_{ijk} = \boldsymbol{e}_i\hat{\mathbf{A}}_2\mathbf{I}_N^{(2)}(\boldsymbol{e}_j \otimes \boldsymbol{e}_k)$ and for the intrusive operator with $\hat{h}_{ijk} = \boldsymbol{e}_i\tilde{\mathbf{A}}_2\mathbf{I}_N^{(2)}(\boldsymbol{e}_j \otimes \boldsymbol{e}_k)$. As can be seen, this violation is essentially on the order of machine precision for both inferred and intrusive operators. Figure 4 shows the symmetry violation of the inferred and the intrusive diffusion matrix, $\frac{\|\hat{\mathbf{A}}_1 - \hat{\mathbf{A}}_1^T\|}{\|\hat{\mathbf{A}}_1\|}$ and $\frac{\|\tilde{\mathbf{A}}_1 - \tilde{\mathbf{A}}_1^T\|}{\|\tilde{\mathbf{A}}_1\|}$, respectively. Also, this violation is on the order of machine precision. Consequently, the diffusion matrices are symmetric, so their eigenvalues are real. These eigenvalues—multiplied with -1—are depicted in Figure 5. As can be seen, the eigenvalues of the inferred diffusion matrices coincide with the eigenvalues of the corresponding intrusive matrices. Hence, both ROMs in particular preserve the negative semidefiniteness of the FOM diffusion operator.

5. Conclusion. This work introduces a method to generate rank-ensuring snapshot data for operator inference that guarantees the exact reconstruction of the corresponding intrusive reduced order model. This method performs the minimal number of FOM time steps and yields a square data matrix, so the operator inference least squares problem can be simplified to a linear system that has better numerical stability. Numerical experiments confirm that the novel snapshot data generation method leads to exact reconstruction of the intrusive ROM operators. Consequently, also properties such as skew-symmetry, symmetry, and negative semidefiniteness of the intrusive operators are preserved by the nonintrusive operators without the need to enforce this structure preservation explicitly.

To improve the condition number of the matrix corresponding to the operator inference linear system, future work could investigate the exploitation of nestedness

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

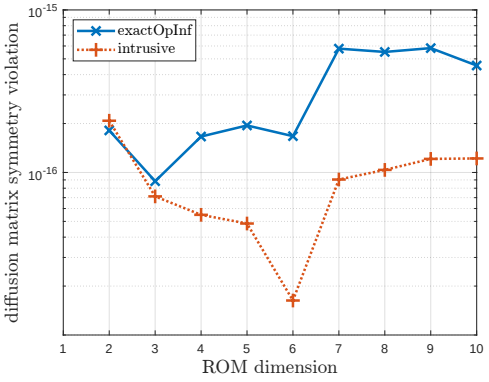

FIG. 4. *Symmetry violation of the diffusion matrix of Burgers' equation.*

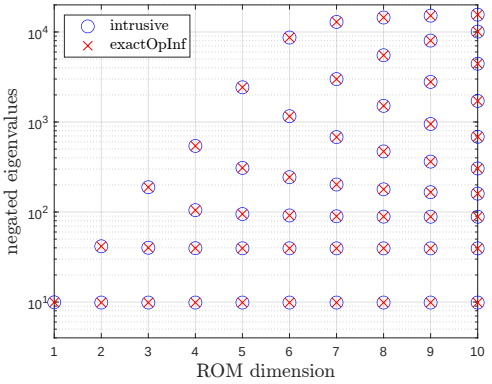

FIG. 5. *Eigenvalues of diffusion matrix of Burgers' equation multiplied with* -1.

to solve subproblems sequentially [2] or different choices of rank-ensuring ROM states, for example, based on Chebychev points [27]. Future work should also investigate how to enable that the proposed rank-ensuring states can be specified as initial conditions in solvers that allow only a certain range of state values or require specific boundary conditions.

In this work, we have not considered the parameter dependency of ROM operators. For that purpose, our work could be combined with [16] to generate rank-sufficient operator inference snapshot data for parametric problems. We expect the full rank guarantee to be transferable to this setting.

Appendix A. Proof of full rank of matrix P. We prove that the ROM states and input signals proposed in section 3.1 guarantee that the matrix $\mathbf{P}$ has full rank. For this purpose, we first note that $\mathbf{P}$ is composed of the submatrices $\mathbf{P} \in \mathbb{R}^{n_p \times n_p}$, $\mathbf{Z} \in \mathbb{R}^{n_p \times N_u}$, $\mathbf{0} \in \mathbb{R}^{N_u \times n_p}$, and $\mathbf{U} \in \mathbb{R}^{N_u \times N_u}$,

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}(\tilde{\boldsymbol{x}}_1, \mathbf{0}) & \dots & \boldsymbol{p}(\tilde{\boldsymbol{x}}_{n_p}, \mathbf{0}) & \boldsymbol{p}(\mathbf{0}, \boldsymbol{u}_1) & \dots & \boldsymbol{p}(\mathbf{0}, \boldsymbol{u}_p) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{P}} & \mathbf{Z} \\ \mathbf{0} & \mathbf{U} \end{bmatrix}. \tag{A.1}$$

The matrix $\bar{\mathbf{P}} \in \mathbb{R}^{n_p \times n_p}$ is the interpolation matrix corresponding to the multivariate interpolation problem of finding the coefficients c_α of the polynomial

$$p : \mathbb{R}^n \to \mathbb{R}, \quad \tilde{\boldsymbol{x}} \mapsto \sum_{|\alpha| \in \mathcal{I}} c_\alpha \tilde{\boldsymbol{x}}^\alpha \tag{A.2}$$

such that

$$p(\tilde{\boldsymbol{x}}_i) = b_i, \quad i = 1, \dots, n_p, \tag{A.3}$$

for the interpolation nodes $\tilde{\boldsymbol{x}}_i \in \cup_{i \in \mathcal{I}} \mathcal{X}^i$ and arbitrary given interpolation values $b_i \in \mathbb{R}$. All entries of $\mathbf{Z}$ are equal to $p(\mathbf{0})$. $\mathbf{U} \in \mathbb{R}^{N_u \times N_u}$ is the N_u-dimensional identity matrix.

Since $\mathbf{P}$ is a block-triangular matrix, it has full rank if its diagonal blocks $\bar{\mathbf{P}}$ and $\mathbf{U}$ have full rank. While the identity matrix $\mathbf{U}$ trivially has full rank, proving that the polynomial interpolation matrix $\bar{\mathbf{P}}$ has full rank is more involved.

Most works on multivariate polynomial interpolation consider the special case $\mathcal{I} = \mathbb{N}_l := \{0, 1, \dots, l\}$ for a given $l \in \mathbb{N}$ [5, 6]. In this work, however, we allow *gaps* in

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

the degree set $\mathcal{I}$, i.e., not all elements of $\mathbb{N}_l$ are necessarily contained in $\mathcal{I}$. Therefore, we define for all $n \in \mathbb{N}^+$, $l \in \mathbb{N}_0$ and any $\mathcal{I} \subset \mathbb{N}_l$, the set of *gappy polynomials* in $\mathcal{I}$,

$$\mathcal{P}^n_{\mathcal{I}} := \left\{ p : \mathbb{R}^n \to \mathbb{R}, \quad \tilde{\boldsymbol{x}} \mapsto \sum_{|\boldsymbol{\alpha}| \in \mathcal{I}} c_{\boldsymbol{\alpha}} \tilde{\boldsymbol{x}}^{\boldsymbol{\alpha}} \;\middle|\; c_{\boldsymbol{\alpha}} \in \mathbb{R}, \boldsymbol{\alpha} \in \{0, 1, \ldots, l\}^n \right\}. \tag{A.4}$$

Then, the following theorem implies that $\bar{\mathbf{P}}$ has full rank.

THEOREM A.1 (Gappy multivariate polynomial interpolation). *For any $n \in \mathbb{N}^+$, $l \in \mathbb{N}_0$, for any degree set $\mathcal{I} \subset \mathbb{N}_l$, there exists a unique polynomial $p \in \mathcal{P}^n_{\mathcal{I}}$ such that*

$$p(\boldsymbol{q}_j) = b_j, \quad j = 1, \ldots, n_p, \tag{A.5}$$

for the interpolation nodes $\boldsymbol{q}_j$ in $\cup_{i \in \mathcal{I}} \mathcal{X}^i$ and arbitrary interpolation values b_j.

A.1. Outline of proof. The special case of degree set s without gaps (i.e. $\mathcal{I} = \mathbb{N}_l$) is covered by an existing result on existence and uniqueness of an interpolating multivariate polynomial [17] described in section A.2. In section A.3, we use this result to prove Theorem A.1 for the special case that $\mathcal{I}$ contains only one element. I.e., we can find for all $i \in \mathcal{I}$ a homogeneous polynomial $p_i \in \mathcal{P}^n_{\{i\}}$ that interpolates in the nodes in $\mathcal{X}^i$. We use this result as the base case in a proof by induction in section A.5 to prove the general case where $\mathcal{I}$ contains multiple elements. There, the induction hypothesis states that Theorem A.1 holds for a fixed degree set $\mathcal{I}^-$; i.e., there exists a unique polynomial p^- that interpolates in the interpolation nodes in $\cup_{i \in \mathcal{I}^-} \mathcal{X}^i$. In the induction step, we show that Theorem A.1 also holds for the new degree set $\mathcal{I} := \{i^*\} \cup \mathcal{I}^-$ for any $i^* \in \mathbb{N}_0$ with $i^* < \min \mathcal{I}^-$. For this purpose, we add a new polynomial p that has two key properties. First, p attains suitable values in the new interpolation nodes in $\mathcal{X}^{i^*}$ such that the resulting polynomial $p^- + p$ attains the prescribed interpolation values b_j. Second, p attains 0 in the old interpolation nodes in $\cup_{i \in \mathcal{I}^-} \mathcal{X}^i$ to not interfere with the interpolation property of p^- in these nodes. In section A.4, we show that such a polynomial p exists in $\mathcal{P}^n_{\mathcal{I}}$.

A.2. Multivariate polynomial interpolation without gaps. The following theorem is given in [17] for the special case without gaps in the degree set, so $\mathcal{I} = \mathbb{N}_l$.

THEOREM A.2 (Multivariate polynomial interpolation without gaps). *For any $m \in \mathbb{N}_0$, let $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \in \mathbb{R}^m$ be the vertices of a nondegenerate m-dimensional simplex in $\mathbb{R}^m$. For any $l \in \mathbb{N}_0$, we define the lattice*

$$B(l, m) := \left\{ \boldsymbol{x} \in \mathbb{R}^m \;\middle|\; \boldsymbol{x} = \sum_{i=0}^{m} \lambda_i \boldsymbol{y}_i,\ \lambda_i \in \mathbb{N}_0,\ \sum_{i=0}^{m} \lambda_i = l \right\}. \tag{A.6}$$

Then, there exists a unique polynomial $p \in \mathcal{P}^m_{\mathbb{N}_l}$ such that

$$p(\boldsymbol{q}_i) = f_i, \quad i = 1, 2, \ldots, \binom{m+l}{m}, \tag{A.7}$$

for the interpolation nodes $\boldsymbol{q}_i$ in $B(l, m)$ and arbitrary interpolation values $f_i \in \mathbb{R}$.

A.3. Homogeneous multivariate polynomial interpolation. Next, we consider the special case that the degree set $\mathcal{I}$ contains only one element l, so the interpolant is a homogeneous polynomial in $\mathcal{P}^n_{\{l\}}$ and the interpolation nodes are given by $\mathcal{X}^l$.

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

THEOREM A.3 (Homogeneous multivariate polynomial interpolation). *For any $n \in \mathbb{N}^+$, $l \in \mathbb{N}_0$, there exists a unique polynomial $p \in \mathcal{P}^n_{\{l\}}$ such that*

$$p(\boldsymbol{q}_i) = f_i, \quad i = 1, 2, \ldots, \binom{n+l-1}{l}, \tag{A.8}$$

for the interpolation nodes $\boldsymbol{q}_i$ in $\mathcal{X}^l$ as defined in (3.1) *and arbitrary interpolation values $f_i \in \mathbb{R}$.*

To prove this theorem, we trace it back to Theorem A.2 on multivariate polynomial interpolation without gaps. First, we note that $\mathcal{X}^l$ lies in an $(n-1)$-dimensional subspace of $\mathbb{R}^n$ and constitutes a lattice of degree l and dimension $n-1$. Hence, we can construct a multivariate polynomial $p \in \mathcal{P}^n_{\mathbb{N}_l}$ that interpolates arbitrary values in $\mathcal{X}^l$. Then, we use the linear dependence of the nodes in $\mathcal{X}^l$ to reformulate p into an equivalent polynomial in $\mathcal{P}^n_{\{l\}}$.

Proof of Theorem A.3. We employ Theorem A.2 with $m = n-1$ and the vertices $\boldsymbol{y}_0 = \boldsymbol{0} \in \mathbb{R}^{n-1}$ and $\boldsymbol{y}_j$ the unit vectors in $\mathbb{R}^{n-1}$ for $j = 1, \ldots, n-1$. The theorem states that there exists a polynomial

$$\hat{p} : \mathbb{R}^{n-1} \to \mathbb{R}, \quad \hat{p}(\hat{\boldsymbol{x}}) = \sum_{|\hat{\boldsymbol{\alpha}}| \le l} \hat{a}_{\hat{\boldsymbol{\alpha}}} \hat{\boldsymbol{x}}^{\hat{\boldsymbol{\alpha}}}, \tag{A.9}$$

with $\hat{a}_{\hat{\boldsymbol{\alpha}}} \in \mathbb{R}$ for all multiexponents $\hat{\boldsymbol{\alpha}} \in \{0, 1, \ldots, l\}^{n-1}$ with $|\hat{\boldsymbol{\alpha}}| \le l$, such that

$$\hat{p}(\hat{\boldsymbol{q}}_i) = f_i, \quad i = 1, 2, \ldots, \binom{n-1+l}{n-1} \tag{A.10}$$

for the prescribed interpolation values f_i and the interpolation nodes $\hat{\boldsymbol{q}}_i$ in $B(l, n-1)$.

We define the linear map $\boldsymbol{m}$ that maps the vertices $\boldsymbol{y}_j$ to the unit vectors in $\mathbb{R}^n$. Concretely, $\boldsymbol{m}(\boldsymbol{y}_j) = \boldsymbol{e}_j$, $j = 1, \ldots, n-1$ and $\boldsymbol{m}(\boldsymbol{y}_0) = \boldsymbol{e}_n$. In view of the definition of $\mathcal{X}^i$ (3.1), we can rewrite the definition of $B(l, m)$ (A.6),

$$B(l, m) = \left\{ \sum_{j=1}^{l} \boldsymbol{a}_j \,\middle|\, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_l \in \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_m\} \right\}. \tag{A.11}$$

Hence, we find

$$\boldsymbol{m}(B(l, n-1)) = \left\{ \sum_{j=1}^{l} \boldsymbol{a}_j \,\middle|\, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_l \in \{\boldsymbol{m}(\boldsymbol{y}_0), \ldots, \boldsymbol{m}(\boldsymbol{y}_{n-1})\} \right\} = \mathcal{X}^l. \tag{A.12}$$

Consequently, the polynomial

$$p(\boldsymbol{x}) := \hat{p}(\boldsymbol{m}^{-1}(\boldsymbol{x})) \tag{A.13}$$

satisfies (A.8) with the $\boldsymbol{q}_i$ in $\mathcal{X}^l$.

To show that this polynomial p is an element of $\mathcal{P}^n_{\{l\}}$, we note that the map $\boldsymbol{m}$ is given by

$$\boldsymbol{m} : \mathbb{R}^{n-1} \to \mathbb{R}^n, \ \hat{\boldsymbol{x}} \mapsto \begin{bmatrix} I_{n-1} \\ \boldsymbol{0}^T \end{bmatrix} \hat{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{0} \\ l \end{bmatrix}, \tag{A.14}$$

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

with the inverse map

$$\boldsymbol{m}^{-1}:\mathbb{R}^n\to\mathbb{R}^{n-1},\ \boldsymbol{x}\mapsto[I_{n-1}\ \mathbf{0}]\boldsymbol{x}. \tag{A.15}$$

We apply the definition of $\boldsymbol{m}^{-1}$ to find

$$p(\boldsymbol{x})=\hat{p}(\boldsymbol{m}^{-1}(\boldsymbol{x}))=\sum_{|\hat{\boldsymbol{\alpha}}|\le l}\hat{a}_{\hat{\boldsymbol{\alpha}}}(\boldsymbol{m}^{-1}(\boldsymbol{x}))^{\hat{\boldsymbol{\alpha}}}=\sum_{|\hat{\boldsymbol{\alpha}}|\le l}\hat{a}_{\hat{\boldsymbol{\alpha}}}\prod_{i=1}^{n-1}x_i^{\hat{\alpha}_i}, \tag{A.16}$$

with the entries x_i of $\boldsymbol{x}\in\mathbb{R}^n$, $i=1,\dots,n$. By definition, all $\boldsymbol{x}\in\mathcal{X}^l$ satisfy

$$\sum_{i=1}^{n}x_i=l, \tag{A.17}$$

so we can multiply $1=\frac{1}{l}\sum_{i=1}^{n}x_i$ to get

$$p(\boldsymbol{x})=\sum_{|\hat{\boldsymbol{\alpha}}|\le l}\hat{a}_{\hat{\boldsymbol{\alpha}}}\left(\prod_{i=1}^{n-1}x_i^{\hat{\alpha}_i}\right)\left(\frac{1}{l}\sum_{i=1}^{n}x_i\right)^{l-\sum_{i=1}^{n}\hat{\alpha}_i}. \tag{A.18}$$

As a result, each summand is a monomial of degree l, so we can write

$$p(\boldsymbol{x})=\sum_{|\boldsymbol{\alpha}|=l}a_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}} \tag{A.19}$$

with suitable $a_{\boldsymbol{\alpha}}\in\mathbb{R}$, for all $\boldsymbol{\alpha}\in\{0,1,\dots,l\}^n$ with $|\boldsymbol{\alpha}|=l$. Thereby, we have proven the existence of a polynomial $p\in\mathcal{P}^n_{\{l\}}$ that satisfies (A.8) for any interpolation values f_i, $i=1,2,\dots,\binom{n+l-1}{l}$. Since the corresponding interpolation matrix is square, existence of a solution implies also uniqueness. ∎

A.4. Specific gappy multivariate polynomial interpolation. As explained in section A.1, the proof of Theorem A.1 in section A.5 involves a polynomial $p\in\mathcal{P}^n_{\mathcal{I}}$ with $\mathcal{I}:=\{i^*\}\cup\mathcal{I}^-$ such that

$$p(\boldsymbol{q}_j)=0,\quad \text{for all } \boldsymbol{q}_j\in\mathcal{X}^i,\ \text{for all } i\in\mathcal{I}^-, \tag{A.20}$$

$$p(\boldsymbol{q}_j)=f_j,\quad \text{for all } \boldsymbol{q}_j\in\mathcal{X}^{i^*}, \tag{A.21}$$

for prescribed interpolation values $f_j\in\mathbb{R}$. To show that such a polynomial p exists, we make the Ansatz

$$p(\boldsymbol{q})=p_{i^*}(\boldsymbol{q})\hat{p}\left(\sum_{k=1}^{n}q_k\right)\quad\forall\boldsymbol{q}\in\mathbb{R}^n, \tag{A.22}$$

with $p_{i^*}\in\mathcal{P}^n_{i^*}$ that satisfies (A.21) according to Theorem A.3, and a univariate polynomial $\hat{p}$ that solves the specific interpolation problem with interpolation values 0 and 1 described in the following lemma.

LEMMA A.4 (Specific gappy univariate polynomial interpolation). *For any $l\in\mathbb{N}_0$, for any $\mathcal{I}:=\{i^*\}\cup\mathcal{I}^-\subset\mathbb{N}_l$ with $i^*<\min\mathcal{I}^-$, there exists a polynomial $p\in\mathcal{P}^1_{\mathcal{I}^{-i^*}}$ with $\mathcal{I}^{-i^*}:=\{i-i^*|i\in\mathcal{I}\}$ such that*

$$p(i)=0,\quad \textit{for all } i\in\mathcal{I}^-\ \textit{and} \tag{A.23}$$

$$p(i^*)=1. \tag{A.24}$$

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

To prove this lemma, we show in Lemma A.5 that for fixed $l \in \mathbb{N}_0$, the existence of solutions to such specific interpolation problems with interpolation values 0 and 1 as described above implies the existence of solutions to the general interpolation problems with arbitrary interpolation values. We use this result to prove Lemma A.4 with a case discrimination: In Lemma A.6, we consider the case $i^* \neq 0$ and in Lemma A.7 the case $i^* = 0$.

LEMMA A.5 (Specific and general gappy univariate interpolation). *If for some $l \in \mathbb{N}_0$, for any $\mathcal{I} \subset \mathbb{N}_l := \{0, 1, \dots, l\}$ and any pairwise distinct nodes $x_1, \dots, x_{n_\mathcal{I}} \in \mathbb{R}$, there exists a polynomial $p \in \mathcal{P}^1_\mathcal{I}$ such that*

$$p(x_j) = 0 \quad \text{for } j = 1, \dots, n_\mathcal{I} - 1, \text{ and} \tag{A.25}$$

$$p(x_{n_\mathcal{I}}) = 1, \tag{A.26}$$

then there also exists for any $\mathcal{I} \subset \mathbb{N}_l$ a unique polynomial $\bar{p} \in \mathcal{P}^1_\mathcal{I}$ such that

$$\bar{p}(x_j) = f_j, \quad \text{for } j = 1, \dots, n_\mathcal{I}, \tag{A.27}$$

for arbitrary interpolation values $f_j \in \mathbb{R}$.

Proof. Let $i_1 < i_2 < \dots i_{n_\mathcal{I}}$ be the elements of a given $\mathcal{I} \subset \mathbb{N}_l$. For $k = 1, \dots, n_\mathcal{I}$, we define the set $\mathcal{I}_k$ of the smallest k of these elements. For each of these sets $\mathcal{I}_k$, let p_k be the polynomial in $\mathcal{P}^1_{\mathcal{I}_k}$ that satisfies

$$p_k(x_j) = 0 \quad \text{for } j = 1, \dots, k-1, \text{ and} \tag{A.28}$$

$$p_k(x_k) = 1. \tag{A.29}$$

To find the polynomial $\bar{p} \in \mathcal{P}^1_\mathcal{I}$ that satisfies (A.27), we make the Ansatz

$$\bar{p}(x) = \sum_{k=1}^{n_\mathcal{I}} c_k p_k, \tag{A.30}$$

to obtain an interpolation problem described by the linear system

$$\mathbf{A}\boldsymbol{c} = \boldsymbol{f}, \tag{A.31}$$

with $\boldsymbol{c} = [c_1 \ \dots \ c_{n_\mathcal{I}}]^T \in \mathbb{R}^{n_\mathcal{I}}$, $\boldsymbol{f} = [f_1 \ \dots \ f_{n_\mathcal{I}}]^T \in \mathbb{R}^{n_\mathcal{I}}$ and the entries of the matrix $\mathbf{A} \in \mathbb{R}^{n_\mathcal{I} \times n_\mathcal{I}}$,

$$\mathbf{A}_{jk} := p_k(x_j), \qquad j, k = 1, \dots, n_\mathcal{I}. \tag{A.32}$$

Because of (A.28) and (A.29), $\mathbf{A}$ is a lower triangular matrix with entries 1 on its diagonal. Hence, $\mathbf{A}$ has full rank, and the linear system (A.31) has a unique solution $\boldsymbol{c}$ for any right-hand side $\boldsymbol{f}$. ∎

The following two lemmas extend this result to arbitrary degrees l; Lemma A.6 for the case that all nodes x_j are positive, Lemma A.7 for the case that $x_{n_\mathcal{I}} = 0$ and $0 \in \mathcal{I}$.

LEMMA A.6 (Specific gappy univariate interpolation for arbitrary l). *For any $l \in \mathbb{N}_0$, for any $\mathcal{I} \in \mathbb{N}_l := \{0, 1, \dots, l\}$, there exists a polynomial $p \in \mathcal{P}^1_\mathcal{I}$ such that the interpolation properties* (A.25) *and* (A.26) *hold for any pairwise distinct, positive nodes $x_1, \dots, x_{n_\mathcal{I}} \in \mathbb{R}^+$.*

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

Proof. The proof proceeds by induction along l.

- Base case: For $l = 0$, $p(x) = 1$ satisfies the interpolation properties for the only existing set $\mathcal{I} = \{0\}$.
- Induction hypothesis: For fixed l, there exists $p \in \mathcal{P}^1_{\mathcal{I}}$ such that the interpolation properties (A.25) and (A.26) hold for any $\mathcal{I} \in \mathbb{N}_l$ and any pairwise distinct, positive nodes $x_1, \dots, x_{n_\mathcal{I}} \in \mathbb{R}$.
- Induction step: We consider a set $\mathcal{I} \subset \mathbb{N}_{l+1} := \{0, 1, \dots, l, l+1\}$ and the pairwise distinct, positive nodes $x_1, \dots, x_{n_\mathcal{I}} \in \mathbb{R}^+$. If $l+1 \notin \mathcal{I}$, then the induction hypothesis directly applies. Otherwise, if $l+1 \in \mathcal{I}$, then the induction hypothesis holds for $\mathcal{I}^- := \mathcal{I} \setminus \{l+1\}$ and the nodes $x_1, \dots, x_{n_\mathcal{I}-1}$. Hence, Lemma A.5 implies that there exists a polynomial $\bar{p} \in \mathcal{P}^1_{\mathcal{I}^-}$ such that

$$\bar{p}(x_j) = f_j, \quad \text{for } j = 1, \dots, n_\mathcal{I} - 1, \tag{A.33}$$

for arbitrary interpolation values $f_i \in \mathbb{R}$. Consequently, there exists $\bar{p} \in \mathcal{P}^1_{\mathcal{I}^-}$ such that

$$\bar{p}(x_j) = -x_j^{l+1}, \quad \text{for } j = 1, \dots, n_\mathcal{I} - 1. \tag{A.34}$$

Then,

$$\hat{p}(x) := \bar{p}(x) + x^{l+1} \tag{A.35}$$

satisfies (A.25); i.e., $x_1, \dots, x_{n_\mathcal{I}-1}$ are roots of $\hat{p}$. By Descartes' rule of signs, $\hat{p}$ has at most $n_\mathcal{I} - 1$ positive roots, hence $\hat{p}(x_{n_\mathcal{I}}) \neq 0$. Consequently, we can normalize $\hat{p}$ to find the desired polynomial,

$$p(x) := \frac{\hat{p}(x)}{\hat{p}(x_{n_\mathcal{I}})}. \tag{A.36}$$

∎

LEMMA A.7 (Specific gappy univariate interpolation for arbitrary l with $\boldsymbol{x}_{n_\mathcal{I}} = 0$). *For any $l \in \mathbb{N}_0 := \{0, 1, \dots, l\}$, for any $\mathcal{I} \in \mathbb{N}_l$ with $0 \in \mathcal{I}$, there exists a polynomial $p \in \mathcal{P}^1_{\mathcal{I}}$ such that the interpolation properties* (A.25) *and* (A.26) *hold for any pairwise distinct, positive nodes $x_1, \dots, x_{n_\mathcal{I}-1} \in \mathbb{R}^+$ and $x_{n_\mathcal{I}} = 0$.*

Proof. Let $\mathcal{I}^- := \mathcal{I} \setminus \{0\}$. According to Lemma A.5 and Lemma A.6, there exists a polynomial $\bar{p} \in \mathcal{P}^1_{\mathcal{I}^-}$ such that

$$\bar{p}(x_j) = -1 \quad \text{for } j = 1, \dots, n_\mathcal{I} - 1. \tag{A.37}$$

Then,

$$p(x) := \bar{p}(x) + 1 \tag{A.38}$$

is an element of $\mathcal{P}^1_{\mathcal{I}}$ and satisfies (A.25). Because $\bar{p} \in \mathcal{P}^1_{\mathcal{I}^-}$ with $0 \notin \mathcal{I}^-$, x is a factor of $\bar{p}$, so $\bar{p}(0) = 0$. Consequently p also satisfies (A.26). ∎

With these preparations, we can prove Lemma A.4.

Proof of Lemma A.4. For any $l \in \mathbb{N}$, for any $\hat{\mathcal{I}} = \{i^*\} \cup \mathcal{I}^- \subset \mathbb{N}_l$ with $i^* < \min \mathcal{I}^-$, let $\mathcal{I} = \{i - i^* | i \in \hat{\mathcal{I}}\}$ and $\{x_1, \dots, x_{n_\mathcal{I}}\} = \hat{\mathcal{I}}$. Then, Lemma A.6 implies the statement if $i^* \neq 0$ and Lemma A.7 if $i^* = 0$. ∎

A.5. Gappy multivariate polynomial interpolation. Now, we have all ingredients to prove Theorem A.1.

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

Proof of Theorem A.1. The proof is performed by induction along the number of elements in $\mathcal{I}$.

- Base case: Let $\mathcal{I} = \{i\}$ for any $i \in \mathbb{N}_l$ with arbitrary $l \in \mathbb{N}_0$, then Theorem A.3 implies the existence and uniqueness of a polynomial $p \in \mathcal{P}^n_{\{i\}}$ that satisfies the interpolation property (A.5) in the interpolation nodes in $\mathcal{X}^i$.
- Induction hypothesis: Let for a fixed degree set $\mathcal{I}^- \subset \mathbb{N}_l$ with arbitrary $l \in \mathbb{N}_0$ exist a unique polynomial $p^- \in \mathcal{P}^n_{\mathcal{I}^-}$ that satisfies the interpolation property (A.5) in the interpolation nodes $\cup_{i\in\mathcal{I}^-}\mathcal{X}^i$.
- Induction step: We consider the degree set $\mathcal{I} := \{i^*\} \cup \mathcal{I}^-$ with arbitrary $i^* \in \mathbb{N}_l$. Without loss of generality, we can assume $i^* < \min \mathcal{I}^-$. As described in section A.4, we can construct a polynomial

$$p(\boldsymbol{q}) = p_{i^*}(\boldsymbol{q})\hat{p}\left(\sum_{k=1}^{n} q_k\right) \quad \forall \boldsymbol{q} \in \mathbb{R}^n, \tag{A.39}$$

with $p_{i^*} \in \mathcal{P}^n_{i^*}$ and $\hat{p} \in \mathcal{P}^1_{\mathcal{I}-i^*}$, so $p \in \mathcal{P}^n_{\mathcal{I}}$. Because of Lemma A.4 and $\sum_{k=1}^n q_k = i$ for all $\boldsymbol{q} \in \mathcal{X}^i$ for any $i \in \mathbb{N}_0$, we can choose $\hat{p}$ such that

$$\hat{p}\left(\sum_{k=1}^{n} q_k\right) = 0, \quad \text{for all } \boldsymbol{q} \in \mathcal{X}^i, \text{ for all } i \in \mathcal{I}^-, \tag{A.40}$$

$$\hat{p}\left(\sum_{k=1}^{n} q_k\right) = 1, \quad \text{for all } \boldsymbol{q} \in \mathcal{X}^{i^*}, \tag{A.41}$$

and p_{i^*} such that

$$p_{i^*}(\boldsymbol{q}_j) = b_j - p^-(\boldsymbol{q}_j), \quad \text{for all } \boldsymbol{q}_j \in \mathcal{X}^{i^*}, \tag{A.42}$$

with the interpolation values b_j prescribed in (A.5). Consequently, the polynomial p satisfies

$$p(\boldsymbol{q}_j) = 0, \quad \text{for all } \boldsymbol{q}_j \in \mathcal{X}^i, \text{ for all } i \in \mathcal{I}^- \text{ and} \tag{A.43}$$

$$p(\boldsymbol{q}_j) = b_j - p^-(\boldsymbol{q}_j), \quad \text{for all } \boldsymbol{q}_j \in \mathcal{X}^{i^*}. \tag{A.44}$$

As a result, the polynomial $p^- + p$ satisfies

$$p^-(\boldsymbol{q}_j) + p(\boldsymbol{q}_j) = b_j + 0 = b_j, \quad \text{for all } \boldsymbol{q}_j \in \mathcal{X}^i, \text{ for all } i \in \mathcal{I}^- \text{ and} \tag{A.45}$$

$$p^-(\boldsymbol{q}_j) = p(\boldsymbol{q}_j) = p^-(\boldsymbol{q}_j) + b_j - p^-(\boldsymbol{q}_j) = b_j, \quad \text{for all } \boldsymbol{q}_j \in \mathcal{X}^{i^*}, \tag{A.46}$$

and hence satisfies the interpolation property (A.5). Since the corresponding interpolation matrix is square, the existence of a solution also implies uniqueness. ∎

Appendix B. Algorithm for generating ROM state set. We present two different approaches to how to implement the procedure to generate the rank-ensuring ROM state sets $\mathcal{X}^i$ as defined in (3.1) in line 7 of Algorithm 3.1. As shown in Algorithm B.1 for polynomial degree $i = 3$, the procedure essentially comprises 3 nested **for**-loops. The first approach shown in Algorithm B.2 constructs these nested **for**-loops for arbitrary i via recursion. This recursion makes the algorithm rather nonintuitive. While this nonintuitivity might be unavoidable to handle arbitrary polynomial degrees i, a more intuitive approach is possible in practice. In practice,

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Algorithm B.1. Generate rank-ensuring ROM state set $\mathcal{X}^3$.

procedure GENERATEROMSTATESET3(ROM dimension n)
  Initialize $\mathcal{X}^3 \leftarrow \emptyset$
  for $j_1 = 1, \ldots, n$ **do**
    for $j_2 = j_1, \ldots, n$ **do**
      for $j_3 = j_2, \ldots, n$ **do**
        $\mathcal{X}^3 \leftarrow \mathcal{X}^3 \cup \boldsymbol{e}_{j_1} + \boldsymbol{e}_{j_2} + \boldsymbol{e}_{j_3}$
      end for
    end for
  end for
  return $\mathcal{X}^3$
end procedure

Algorithm B.2. Generate rank-ensuring ROM state set $\mathcal{X}^i$ - recursive algorithm.

procedure GENERATEROMSTATESET(
  ROM dimension n,
  polynomial degree $i \in \mathbb{N}_0$
)
  return RECURSIVEROMSTATESET(ROM dimension n, polynomial degree i,
    current for loop depth $i^* = 0$,
    current index sequence $\boldsymbol{j} = []$,
    current ROM state set $\mathcal{X}^i = \emptyset$)
end procedure
procedure RECURSIVEROMSTATESET(ROM dimension n, polynomial degree $i \in \mathbb{N}_0$,
    current for loop depth i^*,
    current index sequence $\boldsymbol{j} = [j_1, \ldots, j_{i^*}]$, # always has length i^*
    current ROM state set $\mathcal{X}^i$)
  if $i^* = i$ **then**
    Initialize $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{0} \quad \in \mathbb{R}^n$
    for k=1,...,i **do**
      $\bar{\boldsymbol{x}} \leftarrow \bar{\boldsymbol{x}} + \boldsymbol{e}_{j_k}$
    end for
    return $\mathcal{X}^i \cup \bar{\boldsymbol{x}}$
  else
    `# Perform next inner for-loop`
    for $j_{i^*+1} = j_{i^*}, \ldots, n$ **do** # with $j_0 = 1$
      return RECURSIVEROMSTATESET(ROM dimension n, polynomial degree i,
        current for loop depth $i^* + 1$,
        current index sequence $[j_1, \ldots, j_{i^*+1}]$,
        current ROM state set $\mathcal{X}^i$)
    end for
  end if
end procedure

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Algorithm B.3. Generate rank-ensuring ROM state set $\mathcal{X}^i$ - hard-coded.

procedure GENERATEROMSTATESET(
ROM dimension n,
polynomial degree $i \in \mathbb{N}_0$
)
  if $i = 1$ **then return** GENERATEROMSTATESET1(ROM dimension n)
  else if $i = 2$ **then return** GENERATEROMSTATESET2(ROM dimension n)
  else if $i = 3$ **then return** GENERATEROMSTATESET3(ROM dimension n)
  else if $i = 8$ **then return** GENERATEROMSTATESET8(ROM dimension n)
  end if
end procedure

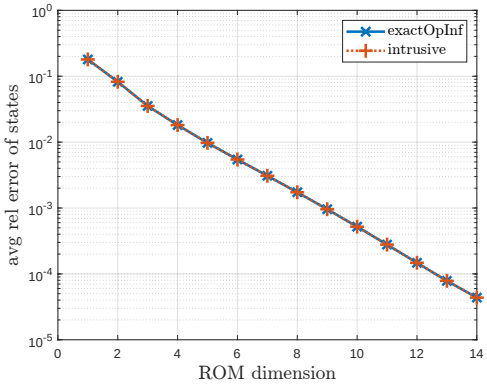

FIG. 6. *Relative average ROM state error* (C.1) *for Chafee–Infante equation.*

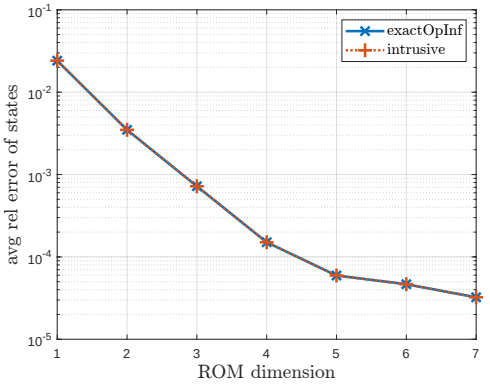

FIG. 7. *Relative average ROM state error* (C.1) *for the ice sheet model.*

we know a priori which polynomial degrees i are involved in the considered problems. For example, in the numerical experiments in section 4, the only polynomial degrees involved are $\{1, 2, 3, 8\}$. Consequently, we can use the hard-coded discrimination of cases shown in Algorithm B.3 as alternative to Algorithm B.2, with the procedures for $i = 1, 2, 8$ implemented analogous to Algorithm B.1.

Appendix C. ROM state errors with respect to the FOM. Figures 6, 7, and 8 show the relative average ROM state errors,

$$\frac{\left\| \mathbf{X} - \mathbf{V}_n \tilde{\mathbf{X}} \right\|_F}{\|\mathbf{X}\|_F}, \tag{C.1}$$

for the Chafee–Infante equation in section 4.1, ice sheet model in section 4.2, and Burgers' equation in section 4.3, respectively. The matrix $\mathbf{X} = [\boldsymbol{x}_0 \ \dots \ \boldsymbol{x}_{K_{\mathrm{POD}}-1}]$ comprises the FOM snapshots used to construct the POD basis and $\tilde{\mathbf{X}} = [\tilde{\boldsymbol{x}}_0 \ \dots \ \tilde{\boldsymbol{x}}_{K_{\mathrm{POD}}-1}]$ the corresponding ROM snapshots generated by integrating the ROM ODE (2.6) with the intrusive ROM operators and the inferred ROM operators, respectively. As can be seen, the ROM errors decrease monotonously for increasing ROM dimensions in all three test cases for both, intrusive and inferred ROMs. Additionally, we cannot see any difference between the errors of corresponding intrusive and inferred ROMs.

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

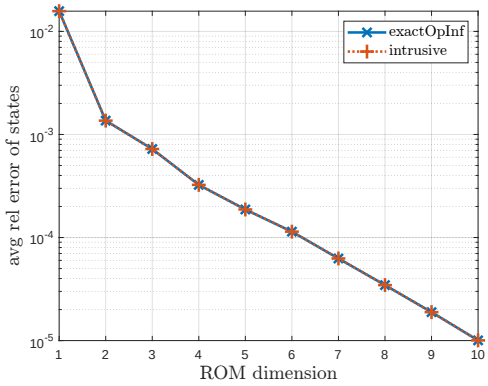

FIG. 8. *Relative average ROM state error* (C.1) *for Burgers' equation.*

In fact, the differences between these errors are all smaller than 10^{-10}. This alignment is due to the exact reconstruction of the intrusive ROM operators shown in Figure 1.

Reproducibility of computational results. This paper has been awarded the "SIAM Reproducibility Badge: Code and data available" as recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/h3rror/exactOpInf and in the supplementary materials (exactOpInf-main.zip [local/web 6.26MB]).

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, 2005.
[2] N. Aretz and K. Willcox, *Enforcing structure in data-driven reduced modeling through nested Operator Inference*, in 2024 IEEE 63rd Conference on Decision and Control (CDC), 2024, pp. 8046–8053, https://doi.org/10.1109/CDC56724.2024.10885857.
[3] P. Benner, P. Goyal, J. Heiland, and I. P. Duff, *Operator inference and physics-informed learning of low-dimensional models for incompressible flows*, Electron. Trans. Numer. Anal., 56 (2022), pp. 28–51, https://doi.org/10.1553/etna_vol56s28.
[4] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715.
[5] M. Gasca, *Multivariate polynomial interpolation*, in Computation of Curves and Surfaces, Springer, 1990, pp. 215–236.
[6] M. Gasca and T. Sauer, *Polynomial interpolation in several variables*, Adv. Comput. Math., 12 (2000), pp. 377–410, https://doi.org/10.1023/A:1018981505752.
[7] P. Goyal, I. P. Duff, and P. Benner, *Guaranteed stable quadratic models and their applications in SINDy and operator inference*, Phys. D, 483 (2025), 134893.
[8] A. Gruber and I. Tezaur, *Canonical and noncanonical Hamiltonian operator inference*, Comput. Methods Appl. Mech. Engrg., 416 (2023), 116334, https://doi.org/10.1016/j.cma.2023.116334.
[9] B. Haasdonk and M. Ohlberger, *Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Math. Comput. Model. Dynam., 17 (2011), pp. 145–161, https://doi.org/10.1080/13873954.2010.514703.
[10] J. S. Hesthaven, G. Rozza, B. Stamm, et al., *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Math. 590, Springer, 2016.
[11] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control*, in Dimension Reduction of Large-Scale Systems: Proceedings of a Workshop held in Oberwolfach, October 19–25, 2003, Springer, Germany, 2005, pp. 261–306.
[12] T. Koike and E. Qian, *Energy-preserving reduced operator inference for efficient design and control*, in AIAA SCITECH Forum, 2024, https://doi.org/10.2514/6.2024-1012.

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

[13] B. Kramer, B. Peherstorfer, and K. E. Willcox, *Learning nonlinear reduced models from data with operator inference*, Annu. Rev. Fluid Mech., 56 (2024), pp. 521–548, https://doi.org/10.1146/annurev-fluid-121021-025220.
[14] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM, 2016.
[15] S. A. McQuarrie, C. Huang, and K. E. Willcox, *Data-driven reduced-order models via regularised operator inference for a single-injector combustion process*, J. Roy. Soc. New Zeal., 51 (2021), pp. 194–211, https://doi.org/10.1080/03036758.2020.1863237.
[16] S. A. McQuarrie, P. Khodabakhshi, and K. E. Willcox, *Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference*, SIAM J. Sci. Comput., 45 (2023), pp. A1917–A1946, https://doi.org/10.1137/21M1452810.
[17] R. Nicolaides, *On a class of finite elements generated by Lagrange interpolation*, SIAM J. Numer. Anal., 9 (1972), pp. 435–445, https://doi.org/10.1137/0709039.
[18] B. Peherstorfer, *Sampling low-dimensional Markovian dynamics for preasymptotically recovering reduced models from data with operator inference*, SIAM J. Sci. Comput., 42 (2020), pp. A3489–A3515, https://doi.org/10.1137/19M1292448.
[19] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction*, Comput. Methods Appl. Mech. Engrg., 306 (2016), pp. 196–215, https://doi.org/10.1016/j.cma.2016.03.025.
[20] C. Prud'Homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, *Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods*, J. Fluids Eng., 124 (2002), pp. 70–80, https://doi.org/10.1115/1.1448332.
[21] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems*, Phys. D, 406 (2020), 132401, https://doi.org/10.1016/j.physd.2020.132401.
[22] H. Rosenberger and B. Sanderse, *No pressure? Energy-consistent ROMs for the incompressible Navier-Stokes equations with time-dependent boundary conditions*, J. Comput. Phys., 491 (2023), 112405, https://doi.org/10.1016/j.jcp.2023.112405.
[23] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275, https://doi.org/10.1007/s11831-008-9019-9.
[24] B. Sanderse, *Non-linearly stable reduced-order models for incompressible flow with energy-conserving finite volume methods*, J. Comput. Phys., 421 (2020), 109736, https://doi.org/10.1016/j.jcp.2020.109736.
[25] N. Sawant, B. Kramer, and B. Peherstorfer, *Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference*, Comput. Methods Appl. Mech. Engrg., 404 (2023), 115836, https://doi.org/10.1016/j.cma.2022.115836.
[26] R. Swischuk, B. Kramer, C. Huang, and K. Willcox, *Learning physics-based reduced-order models for a single-injector combustion process*, AIAA J., 58 (2020), pp. 2658–2672, https://doi.org/10.2514/1.J058943.
[27] L. N. Trefethen, *Approximation Theory and Approximation Practice*, extended ed., SIAM, 2019.
[28] K. Urban and A. T. Patera, *A new error bound for reduced basis approximation of parabolic partial differential equations*, C. R. Math., 350 (2012), pp. 203–207, https://doi.org/10.1016/j.crma.2012.01.026.
[29] W. I. T. Uy, Y. Wang, Y. Wen, and B. Peherstorfer, *Active operator inference for learning low-dimensional dynamical-system models from noisy data*, SIAM J. Sci. Comput., 45 (2023), pp. A1462–A1490, https://doi.org/10.1137/21M1439729.
[30] T. van Gastelen, W. Edeling, and B. Sanderse, *Energy-conserving neural network for turbulence closure modeling*, J. Comput. Phys., 508 (2024), 113003, https://doi.org/10.1016/j.jcp.2024.113003.
[31] K. Veroy and A. T. Patera, *Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: Rigorous reduced-basis a posteriori error bounds*, Internat. J. Numer. Methods Fluids, 47 (2005), pp. 773–788, https://doi.org/10.1002/fld.867.
[32] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, MA, 2006.

Downloaded 09/01/26 to 193.205.81.18 . Redistribution subject to CCBY license

© 2026 SIAM. Published by SIAM under Creative Commons Attribution NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) license