LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-834453

Parametric dynamic mode decomposition for reduced order modeling

Q. A. Huhn, M. E. Tano, J. C. Ragusa, Y. Choi

April 27, 2022

Journal of Computational Physics

Disclaimer

Parametric Dynamic Mode Decomposition for Reduced Order Modeling

Quincy A. Huhn[a], Mauricio E. Tano[a], Jean C. Ragusa[a,*], Youngsoo Choi[b]

[a] *Department of Nuclear Engineering, Texas A&M University, College Station, TX 77843*
[b] *Lawrence Livermore National Laboratory, 7000 East Ave, Livermore, CA 94550, USA*

Abstract

Dynamic Mode Decomposition (DMD) is a model-order reduction approach, whereby spatial modes of fixed temporal frequencies are extracted from numerical or experimental data sets. The DMD low-rank or reduced operator is typically obtained by singular value decomposition of the temporal data sets. For parameter-dependent models, as found in many multi-query applications such as uncertainty quantification or design optimization, the only parametric DMD technique developed was a stacked approach, with data sets at multiple parameter values were aggregated together, increasing the computational work needed to devise low-rank dynamical reduced-order models. In this paper, we present two novel approach to carry out parametric DMD: one based on the interpolation of the reduced-order DMD eigen-pair and the other based on the interpolation of the reduced DMD (Koopman) operator. Numerical results are presented for diffusion-dominated nonlinear dynamical problems, including a multiphysics radiative transfer example. All three parametric DMD approaches are compared.

Keywords: Dynamical Mode Decomposition, Parametric Model Dependence, Parametric Model-Order Reduction

*Corresponding author
Email addresses: quincy.huhn98@tamu.edu (Quincy A. Huhn), mtano@tamu.edu (Mauricio E. Tano), jean.ragusa@tamu.edu (Jean C. Ragusa), choi15@llnl.gov (Youngsoo Choi)

1. Introduction

High-fidelity, dynamic, multi-physics simulations are becoming more routinely used in engineering practice and research. Due to their large computational cost, typically requiring several processing nodes or supercomputers, these simulations are most often limited to providing a single, best-estimate answer to a given problem [1]. However, multi-query applications in engineering, such as uncertainty quantification, optimal control, or design optimization, require several, repeated evaluations of a computational model in which the uncertain or design parameters are assigned different values [2, 3, 4, 5, 6, 7]. Therefore, multi-query applications generally lie beyond the realm of high-fidelity simulation tools and rely on simplified, less-accurate computational models [8].

Reduced order models (ROMs) offer the possibility of finding reduced representations of high-fidelity, dynamic, multi-physics models. The ROM representations can significantly speed up computational times while introducing controllable prediction errors, when compared to the expected variations over the expected range of uncertain or design parameters in the multi-query problem [9, 10, 11, 12, 13, 14, 15, 16]. Broadly, ROMs can be classified as *intrusive* or *non-intrusive* regarding whether they need access or not to the discretized system of equations of the higher-fidelity simulations, respectively [17]. Both intrusive and non-intrusive ROMs involve the distillation of a reduced set of the most observable and/or controllable modes from a set of snapshots obtained by parametric realizations of the high-fidelity system. Typically, proper orthogonal decomposition (POD) [18] or non-linear manifold learning via artificial neural networks (ANNs) [19, 20] are used for computing the reduced set of modes. Then, intrusive ROMs use the computed modes for truncating, e.g., balanced truncation method [21], projecting, e.g., Galerkin or Petrov-Galerkin projections [22, 23, 24], or moment-matching, e.g., moment-matching method [25], the high-fidelity discretized system of equations into a reduced representation. Thanks to the temporal and parametric affinity of the high-fidelity model or by gappy-sampling of the nonlinear parametric terms, e.g., via the Discrete

Empirical Interpolation Method [26], intrusive ROMs preserve temporal and parametric dependence and, thus, have been effectively used as surrogate models in multi-query problems [27]. Note, however, that in the dynamic evolution of the system, the most influential or observable modes will change in time and, thus, the projection performance may deteriorate during transients. Nonetheless, variants for dynamically selecting the most influential modes have been introduced to overcome this issue in intrusive ROMs, such as spatio-temporal biorthogonal decomposition [28, 29, 30], spectral POD [31], time-dependent projection manifolds [32], and time windowing [33]. Intrusive ROMs have also been developed that take into account conservation constraints [34], and reduce training cost by building subdomain ROMs and assembling them in a least-squares sense when a large-scale problem needs to be reduced [35]. On the other hand, non-intrusive ROMs use the compressed set of modes and the dynamic evolution observed in the high-fidelity system realizations to derive a reduced expression for the evolution of the selected modes, see [36] as a review of the application of non-intrusive ROMs to the Navier Stokes equations.

As high-fidelity systems generally involve numerical artifacts for improving the computational performance or well-posedness of the system, e.g., matrix-free methods [37] or artificial viscosity [38], and are implemented on legacy code, the development and implementation of adapted, stable, intrusive ROMs require vast efforts [39]. Therefore, there is a growing interest in non-intrusive ROMs. Dynamic mode decomposition (DMD), ANNs, and Gaussian process regression (GPR) have been widely used for developing non-intrusive ROMs.

As a non-intrusive ROM, GPR utilizes Bayesian statistics to create a surrogate model that includes an estimate of the variance of the approximation [40]. To create ROMs for structural analysis GPR has been applied to combine modes generated from POD [41]. In nuclear engineering GPR has been applied to the multi-fidelity simulation of radiative shocks [42].

Regarding ANNs, the reduction method generally relies on the reduction of high-fidelity snapshots into a latent reduced representation, via an encoder-type ANN or POD, coupled with an ANN expansion of the latent space into

the high-fidelity space, via a decoder-type ANN [43]. The parametric dependence of the high-fidelity system is emulated in ANN-driven ROMs via a parameter-dependent variational regularization of the latent reduced representation [44, 45, 46]. Similarly, the time dependence is either emulated via variation regularization [47] or is introduced in the ANN by adding recurrent structures between the encoder and decoder parts of the ANN [48]. However, in their current state-of-the-art, ANN-driven ROMs are still subjected to numerical and performance constraints [49, 50, 51]. Numerically, the modes selected by the ANN are non-orthonormal and involve representations with a lower order of convergence than the high fidelity model and that do not preserve physical constraints [52]. Regarding performance, the ANN compression and expansion involve a large number of weights between hidden layers, which frequently yield small speed-ups in terms of number of floating point operations. Although several improvements have been recently made by incorporating physical constraints [53], improving convergence order [54], and increasing speed ups of ANN-driven ROMs [55], the development of efficient ANN-driven ROMs is still highly problem-dependent and requires a significant amount of user-tuning and training cost.

DMD methods have been popularly used for non-intrusive ROM over the past decade [56]. DMD aims at finding a reduced representation Koopman operator [57], which allows us to naturally represent the dynamic evolution of the system on the selected POD modes via a transient, mode-decaying algebraic equation based on the modes and frequencies of this operator [58]. Although Mezic [59] was the first one to use DMD as a ROM method, many variants have later appeared to improve DMD performance on large-stream data sets [60], prediction boundedness [61], ability to capture multiple scales [62, 63], and reduce sensitivity to noise [64], between others. The reader is referred to the review by Kutz et al. for details about these methods [56]. Despite accounting for the system's time dependence, for a single parameter realization, comes naturally in DMD via the reduced representation of the Koopman operator, it is significantly more challenging to introduce parametric dependence on DMD-

ROMs.

To the best of our knowledge, the only work proposing a parameter-dependent framework for DMD-ROM is the one by Syadi, Schmid et al. [65], where the DMD-ROM model shows a good performance for representing transient signals involving a bifurcation parameter. In the model proposed, the authors perform DMD on a virtual augmented snapshot matrix, created by time-stacking of high-fidelity snapshots for all parameter realizations, and then select the DMD modes by parametric-interpolation of the stacked DMD modes created. Albeit performant in the problem evaluated, the proposed method presents three key limitations. First, the computational cost of performing POD on the snapshot matrix grows at best linearly with the dimension of this matrix [66]. Thus, the memory trace and computational performance of this method may increase linearly with the number of parameters included in the training set of DMD, making it impractical for multi-query, multi-parametric applications. Second, by assuming that snapshots can be vertically stacked, the method forces a parameter-independent frequency in the evolution of the DMD modes, which may be inadequate in nonlinear problems. Finally, vertical stacking requires snapshots to be sampled at the same time, which introduces complications when time-adaptivity is used for efficiently solving the the high-fidelity models. Hence, in this paper, we propose two new algorithms for performing parametric DMD-ROMs. In these algorithms, DMD is performed independently per parameter realization and the parametric interpolation is performed at the level of the reduced Koopman operator. Whereas one of this methods proposes the parametric-interpolation of the reduced Koopman operator eigen-pair, the other one suggest an independent interpolation of each component of the Koopman operator. We then compare the accuracy and computational performance of the method proposed by Syadi et al. against the two proposed methods, using 2D and 3D test cases.

The rest of this article is organized as follows. In Section 2, we discuss Dynamical Mode Decomposition (DMD) as a data-driven means to linearly approximate nonlinear dynamics, in the context of discretized governing equations

arising from PDEs. In Section 3, we first review the state-of-the-art for parametric DMD ("stacked" DMD) for parametric PDEs. We then propose two new and computationally more efficient approaches for parametric DMD, the reduced eigen-pair Interpolation (rEPI) and the reduced Koopman Operator Interpolation (rKOI). In Section 4, we compare the various parametric DMD approaches for several parametric PDEs: (1) a nonlinear heat conduction problem, (2) an advection-reaction-diffusion problem, and (3) a coupled radiation energy/material temperature problem described by coupled nonlinear diffusion-reaction PDEs. We conclude in Section 5.

2. Background on Dynamical Mode Decomposition

As a starting point, we consider a possibly nonlinear dynamical system:

$$\frac{\partial y}{\partial t} = \mathcal{F}(y(\mathbf{x}, t), t)\,. \tag{1}$$

Eq. (1) is obtained by invoking governing equations, typically expressed as partial differential equations. Examples of such systems are provided in Section 4 for (1) nonlinear heat conduction, (2) advection-reaction-diffusion problem, (3) coupled temperature-radiation grey radiative transfer. In Eq. (1), t represents time, $\mathbf{x}$ is the phase-space independent variables (usually physical space coordinates, but other dimensions may be present, such as energy/frequency in non-grey radiative transfer, direction in particle transport, etc., when the solution belongs to a higher dimensional phase space), u is the solution (state) of the governing equation(s), and $\mathcal{F}(y, t)$ is a nonlinear operator that describes the governing equation(s). If the problem is multi-physics, then y is to be viewed as the solution of the whole multi-physics problem, where each component of y is related to a single physics. Oftentimes, Eq. (1) depends on parameters that are either uncertain or that one may want to change or modify, so Eq. (1) may have to be solved repeatedly for each parameter realization. This is discussed further in Section 3.

After discretization of the phase space, one obtains a system of coupled

ordinary differential equations (ODEs):

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{y}(t), t)\,, \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^N$ is a vector representing the state of the dynamical system for each of the N degrees of freedom at time t and $\mathbf{F}(\cdot)$ represents the dynamics in a discrete setting. The state vector $\mathbf{y}$ can be quite large, due to the large number of degrees of freedom involved in the discretization of the phase space, that is, we often have $N \gg 1$. Finally, Eq. (2) is discretized in time and solution at various time steps $i\Delta t$ $(0 \leq i \leq m)$ are obtained.

In Dynamical Mode Decomposition (DMD), one attempts are replacing the operator $\mathbf{F}(\cdot)$ of Eq. (2) with a linear operator to capture the time evolution of the solution $\mathbf{y}$. Hence, we seek the following surrogate system

$$\frac{d\mathbf{y}}{dt} = \hat{\mathbf{A}}\mathbf{y}\,, \tag{3}$$

which, in effect, assumes that the data collected was generated from linear dynamics and that linear operator $\hat{\mathbf{A}}$ approximates these dynamics. In DMD, the operator $\hat{\mathbf{A}}$ is learned in a data-driven fashion. We start with a time series of snapshots $\mathbf{S} = [\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_m] = [\mathbf{y}_i]_{i=0}^{m} \in \mathbb{R}^{n\times m}$, which can be obtained from either experimental data or from simulations. Here, m stands for the number of snapshots in time and n is the full-order dimension of the system, which can be either the number of degrees of freedom in numerical simulations, i.e., $n = N$, or the number of measuring devices in experiments. We then split the data matrix $\mathbf{S}$ into a *lagged matrix of snapshots* $\mathbf{S}^- = [\mathbf{y}_i]_{i=0}^{m-1} \in \mathbb{R}^{n\times(m-1)}$ and a *forward matrix of snapshots* $\mathbf{S}^+ = [\mathbf{y}_i]_{i=1}^{m} \in \mathbb{R}^{n\times(m-1)}$. In a continuous case, given the current state of a system, the operator that defines the new state of the system after an infinitesimal time is known as Koopman operator [67]. Using this concept and the linearized version of the surrogate system in Eq. 3, one can define a discrete version of the Koopman operator $\mathbf{A} \in \mathbb{R}^{n\times n}$ as linear map from old to new states as follows:

$$\mathbf{S}^+ = \mathbf{A}\mathbf{S}^-\,. \tag{4}$$

Note that $\mathbf{A}$ is the operator resulting from the time discretization of Eq. (3). Obtaining $\mathbf{A}$ using data may seem a relatively simple, invoking the pseudo-inverse of the data as $\mathbf{A} = \mathbf{S}^{+}(\mathbf{S}^{-})^{\dagger}$, where † denote the Moore-Penrose pseudo-inverse which minimizes Eq. 4 in a least-squares sense in the Frobenius norm. However, the snapshot matrix can be large, usually with the number of (spatial) unknowns much larger than the number of collection times, i.e., $n \gg m-1$. Hence, $\mathbf{A}$ is a low-rank matrix, whose rank is at most $m-1$. Thus $\mathbf{A}$ is not sought after directly but rather we seek to discover its low-rank subspace first and use the dynamics of low-rank approximation to represent the full state dynamics, i.e., we seek for a reduced representation of the discrete Koopman operator $\mathbf{A}_r \in \mathbb{R}^{r\times r}$ with $r \ll n$. With DMD, one proposes to find this reduced representation based on the principal modes in space and time of the matrix of snapshots $\mathbf{S}$. To that effect, we compute the Singular Value Decomposition (SVD) of the lagged matrix of snapshots $\mathbf{S}^{-}$:

$$\mathbf{S}^{-} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{T}\,, \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V}^{T} \in \mathbb{R}^{m\times m}$ are two orthogonal matrices that hierarchically contain the principal modes of the space and time behaviour of the system, respectively. The orthogonality of this matrix will be key to ensure that the algorithm will yield the correct modes for the reduced Koopman operator. Furthermore, $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix that stores the singular values, σ_i.

We then reduce the rank by truncating $\mathbf{U}$ and $\mathbf{V}$ to the first r columns, i.e., defining $\mathbf{U}_r = [\mathbf{u}]_{i=1}^{r} \in \mathbb{R}^{n\times r}$ and $\mathbf{V}_r = [\mathbf{v}]_{i=1}^{r} \in \mathbb{R}^{m\times r}$, respectively. In addition, we also select the first r singular values in $\mathbf{\Sigma}$ defining a diagonal matrix of singular values as $\mathbf{\Sigma}_r = \mathrm{diag}([\Sigma]_{i=1}^{r}) \in \mathbb{R}^{r\times r}$. The rank r is typically chosen so that a certain fraction of the information (proportional to the singular values) is retained, that is,

$$r = \underset{j}{\mathrm{argmin}} \frac{\sum_{i=1}^{i=j}\sigma_i}{\sum_{i=1}^{i=n}\sigma_i} < \tau\,, \tag{6}$$

where τ is a user-specified fraction between 0 and 1 (usually close to 1). With

the truncated SVD, we have

$$\mathbf{S}^{-} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T \,. \tag{7}$$

Then, using the orthogonality of $\mathbf{U}$ and $\mathbf{V}$ and plugging Eq. (7) into Eq. (4), we obtain the reduced Koopman operator as

$$\mathbf{A}_r \equiv \mathbf{U}_r^T \mathbf{A} \mathbf{U}_r = \mathbf{U}_r^T \mathbf{S}^{+} \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \,. \tag{8}$$

$\mathbf{A}_r$ is of size $r \times r$, with r significantly smaller than n ($r \ll n$), so $\mathbf{A}_r$ is a much smaller matrix than $\mathbf{A}$. It is, therefore, computationally inexpensive to perform the eigen-decomposition of $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, yielding:

$$\mathbf{A}_r \mathbf{W} = \mathbf{\Lambda} \mathbf{W} \,. \tag{9}$$

The projected DMD modes are then computed as $\mathbf{\Phi} = \mathbf{U}_r \mathbf{W}$ and the evolution of the reconstructed system can then be computed via the formula

$$\mathbf{y}(t) = \mathbf{\Phi} \mathbf{\Lambda}^{t/\Delta t} \mathbf{b}_0 = \sum_{i=1}^{r} b_{0i} \phi_i (\lambda_i)^{t/\Delta t} \,, \tag{10}$$

where $\mathbf{b}_0 = \mathbf{\Phi}^{\dagger} \mathbf{y}_0 \in \mathbb{R}^n$ is the vector of initial coefficients, $\mathbf{\Phi}^{\dagger}$ is the pseudo-inverse of the matrix of modes, and $\mathbf{y}_0 \in \mathbb{R}^n$ is the snapshot at the initial time. Algorithm 1 summarizes the steps for the classical DMD.

Algorithm 1 DMD Algorithm

1: Solve Eq. (2) and collect temporal snapshots $[\mathbf{y}(t_i)]_{i=0}^{m}$
2: Arrange snapshots in $\mathbf{S}^{+}$ and $\mathbf{S}^{-}$ data matrices
3: Perform SVD of $\mathbf{S}^{-}$: $\mathbf{S}^{-} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^{\mathbf{T}}$
4: Retain r modes and compute the reduced Koopman operator $\mathbf{A_r}$, Eq. (8)
5: Perform the eigen-decomposition of $\mathbf{A_r}$ to obtain the reduced eigen-modes, Eq. (9)
6: Recover the full-state modes $\mathbf{\Phi} = \mathbf{U_r} \mathbf{W}$
7: Reconstruct the full-state solution $\mathbf{y}(t)$, Eq. (10)

3. Parametric Dynamical Mode Decomposition

In this section, we expand upon the general algorithm for DMD, presented in Section 2, in order to determine the temporal evolution of the system under parametric dependence. Indeed, the governing equations of Eq. (1) may be dependent upon some parameters μ and can be re-stated in a parametric fashion as follows:

$$\frac{\partial y^{\mu}}{\partial t} = \mathcal{F}(y^{\mu}(\mathbf{x}, t; \mu), t; \mu)\,. \tag{11}$$

After discretization of the phase space, one obtains a system of coupled parametric ODEs:

$$\frac{d\mathbf{y}^{\mu}}{dt} = \mathbf{F}(\mathbf{y}^{\mu}(t; \mu), t; \mu)\,. \tag{12}$$

In parametric DMD, one seeks to replace the operator $\mathbf{F}(\cdot)$ with a linear surrogate

$$\frac{d\mathbf{y}^{\theta}}{dt} = \hat{\mathbf{A}}(\theta)\mathbf{y}^{\theta}\,, \tag{13}$$

where θ is a parameter realization (we used a different Greek letter to emphasize that the new parameter realization θ is not in the training set, which we denoted by μ).

To the best of our knowledge, the only parametric DMD algorithm is that of [65]. We will review that work as well as present two novel methods, developed in this paper. All three algorithms will be evaluated on time series from some parametric governing equations. The parametric DMD solutions will be assessed using parameter values different from that of the training set.

3.1. Stacked Parametric DMD

The current state-of-the-art in parametric DMD is the *stacked* DMD algorithm, proposed [65]. This approach is very similar to the non-parametric DMD of Section 2, except that the time series solutions for different parameter values are "stacked" to form an augmented snapshot matrix $\mathbf{S}$. The new snapshot matrix contains the vertically stacked time series for each parameter realization

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mu_1} \\ \vdots \\ \mathbf{S}_{\mu_i} \\ \vdots \\ \mathbf{S}_{\mu_{N_S}} \end{bmatrix}, \tag{14}$$

where N_S is the number of parametric realizations (i.e., samples) in the training set. Algorithm 1 is then applied using this matrix of stacked snapshots ($\mathbf{S} \in \mathbb{R}^{(N_S \times n) \times m}$), yielding the parametric projected DMD modes $\mathbf{\Phi}$ as follows:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{\mu_1} \\ \vdots \\ \mathbf{\Phi}_{\mu_i} \\ \vdots \\ \mathbf{\Phi}_{\mu_{N_S}} \end{bmatrix}. \tag{15}$$

For any new parametric realization θ, the projected DMD mode $\mathbf{\Phi}_\theta$ is obtained from a point-wise Lagrangian interpolation of a subset of the $\mathbf{\Phi}_{\mu_j}$ vectors. That subset is defined as the set of training values that corresponds to the ball $\mathcal{B}$ of nearest neighbors for the new parameter θ. That is, we pick parameters μ_j's from the training set such that $\mu_j \in \mathcal{B}(\theta)$. We define the set of such training parameters as: $\mathcal{S} = \{j \text{ such that } \mu_j \in \mathcal{B}(\theta)\}$ and thus use the corresponding components $\mathbf{\Phi}_{\mu_j}$ of $\mathbf{\Phi}$ ($j \in \mathcal{S}$) to obtain, by point-wise Lagrangian interpolation, $\mathbf{\Phi}_\theta$. A similar procedure is employed for the initial vector $\mathbf{b}_0^\theta$ that is interpolated from the $\mathbf{b}_0^{\mu_j}$ vectors.

In this approach, the DMD time eigen-values are shared among all snapshots. Therefore, we can then simply use the time evaluation formula of Eq. (16) to produce the solution at the new parameter value θ:

$$\mathbf{y}^\theta(t) = \mathbf{\Phi}_\theta \mathbf{\Lambda}_\theta^{t/\Delta t} \mathbf{b}_0^\theta = \sum_{i=1}^{r} b_{0i}^\theta \phi_{\theta i} (\lambda_{\theta i})^{t/\Delta t}, \tag{16}$$

Algorithm 2 summarizes the steps needed for the stacked DMD.

There are two limitations to the stacked DMD approach:

Algorithm 2 Parametric DMD Algorithm via snapshot stacking

1: Solve Eq. (12) for all training parameters $\{\mu_j\}_{k=1}^{N_S}$ and collect temporal snapshots $\mathbf{S}_{\mu_j} = [\mathbf{y}(t_i;\mu_j)]_{i=0}^{m}$ for $1 \leq j \leq N_S$ ▷ Steps 1-8 are offline
2: Arrange the $\mathbf{S}_{\mu_j}$ matrices into the stacked matrix of snapshots $\mathbf{S}$, Eq. (14)
3: Arrange stacked snapshots in $\mathbf{S}^+$ and $\mathbf{S}^-$ data matrices
4: Perform SVD of $\mathbf{S}^-$: $\mathbf{S}^- = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$
5: Retain r modes and compute the reduced Koopman operator $\mathbf{A_r}$, Eq. (8)
6: Perform the eigen-decomposition of $\mathbf{A_r}$ to obtain the reduced eigen-modes, Eq. (9)
7: Recover the full-state stacked DMD modes $\mathbf{\Phi} = \mathbf{U}_r\mathbf{W}$, Eq. (15)
8: Recover the initial coefficients for the DMD evolution as $\mathbf{b}_0 = \mathbf{\Phi}^{\dagger}\left[\left(\mathbf{y}_0^{\mu_1}\right)^T \cdots \left(\mathbf{y}_0^{\mu_{N_S}}\right)^T\right]^T$
9: Interpolate $\mathbf{\Phi}$ and $\mathbf{b}_0$ to find the new parameter realization's modes ($\mathbf{\Phi}_\theta$) and initial condition ($\mathbf{b}_0^\theta$) ▷ Steps 9-10 are online
10: Reconstruct the full-state solution $\mathbf{y}^\theta(t)$, Eq. (16)

1. The eigen-values are shared among all parametric snapshot realizations, forcing the dynamics of the DMD reconstruction to be independent of the parametric realization.
2. The online computational cost and memory requirements of stacked DMD is large because SVD is performed on the data matrices containing all of the parametric snapshots. This can be prohibitively expensive for simulations with a large number of realizations of parameter values.

The two new methods we introduce below are meant to combat the limitations stacked DMD.

3.2. Reduced Eigen-pair Interpolation (rEPI)

The first method we propose is the reduced Eigen-Pair Interpolation (rEPI) method. Given a new parameter realization θ, we only perform **individual** classical Dynamical Mode Decompositions, one for each training parameter μ_j that is a nearest neighbor of the new parameter θ (i.e., $\mu_j \in \mathcal{B}(\theta)$ or, equivalently

$j \in \mathcal{S}$). We denote by $J = \text{card}(\mathcal{S})$ the number of nearest neighbors. Then, we perform a point-wise Lagrangian interpolation of the eigen-pairs of the reduced Koopman operator. Denoting the dimension of the input parameter space by P, the number of nearest neighbors is $J \propto 2^P$ (e.g., as in the case of a tensor grid of the training set). So, the largest computational cost in this approach is the computation of J SVD calculations, each one for a snapshot matrix of size $n \times (m-1)$. However, the important distinction with the stacked DMD approach is that we do not perform a unique SVD decomposition for the whole training set (stacked snapshot matrix of size $(P \times n) \times (m-1)$) but we perform individual SVD decompositions, one per training parameter. We then reduce the rank of these J SVD decompositions according to the criterion in Eq (6) and select the largest of the J ranks ($r = \max_{j \in J}(r_j)$), yielding the reduced left- and right-singular vectors $\mathbf{U}_{jr}$ and $\mathbf{V}_{jr}^T$ as well as the reduced singular value matrix $\mathbf{\Sigma}_{jr}$. Now, we can construct the J individual reduced Koopman operators

$$\mathbf{A}_{jr} = \mathbf{U}_{jr}^T \mathbf{S}_j^+ \mathbf{V}_{jr} \mathbf{\Sigma}_{jr}^{-1} \,, \tag{17}$$

from which we obtain the reduced eigen-pairs:

$$\mathbf{A}_{jr} \mathbf{W}_{jr} = \mathbf{\Lambda}_{jr} \mathbf{W}_{jr} \,. \tag{18}$$

At this stage, the eigen-pair of interest is obtained ($\mathbf{\Lambda}_{\theta r}$, $\mathbf{W}_{\theta r}$) via interpolation, using the J eigen-pairs ($\mathbf{\Lambda}_{jr}$, $\mathbf{W}_{jr}$). Similarly, the interpolated SVD mode $\mathbf{U}_{\theta r}$ is computed via interpolation using the individual SVD modes $\mathbf{U}_{jr}$. Then, the projected DMD modes are computed as $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta r} \mathbf{W}_{\theta r}$. Finally, the initial value $\mathbf{b}_\theta$ is also interpolated using the J neighbors $\mathbf{b}_j$ and the expression of Eq. (16) can be employed to reconstruct the time evolution of $\mathbf{y}^\theta$ as we now have $\mathbf{\Phi}_\theta$, $\mathbf{\Lambda}_{\theta r}$, and $\mathbf{b}_\theta$. The process is summarized in Algorithm 3.

The rEPI method solves both limitations of stacked DMD for parametric applications in the most straightforward way. The time eigen-values are no longer shared between all parameters as they are calculated as part of the DMD performed at each parameter value in the training set. Furthermore, since DMD is performed for each parameter independently of the others, the computational

cost of this method scales linearly with the number of parameters used for the interpolation set (J) instead of the number of parameters used in the training set (N_S). Note, however, that when interpolating the eigen-pairs, this method assumes smoothness of the found eigen-pair along the parametric range. This limitation will limit the performance of this method for high-order parametric DMD as later observed in the results of this article.

Algorithm 3 Parametric DMD algorithm via reduced eigen-pair interpolation

1: Solve Eq. (12) for all training parameters $\{\mu_j\}_{k=1}^{N_S}$ and collect temporal snapshots $\mathbf{S}_{\mu_j} = [\mathbf{y}(t_i;\mu_j)]_{i=0}^{m}$ for $1 \leq j \leq N_S$ ▷ Steps 1-5 are offline
2: Arrange matrix of snapshots in $\mathbf{S}_j^+$ and $\mathbf{S}_j^-$ data matrices, $1 \leq j \leq J$
3: Perform SVD of $\mathbf{S}_j^-$: $\mathbf{S}_j^- = \mathbf{U}_j\mathbf{\Sigma}_j\mathbf{V}_j^T$, $1 \leq j \leq J$
4: Retain r modes and compute the reduced Koopman operator $\mathbf{A}_{jr}$, , $1 \leq j \leq J$, Eq. (17)
5: Perform the eigen-decomposition of $\mathbf{A}_{jr}$ to obtain the reduced eigen-modes, Eq. (18)
6: Interpolate SVD-modes $\mathbf{U}_j$, eigen-modes $\mathbf{W}_{jr}$, and eigen-values $\mathbf{\Lambda}_{jr}$ for to find $\mathbf{U}_{\theta r}$, $\mathbf{W}_{\theta r}$, and $\mathbf{\Lambda}_{\theta r}$, respectively ▷ Steps 6-9 are online
7: Construct the DMD-mode $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta r}\mathbf{W}_{\theta r}$
8: Recover the initial coefficients $\mathbf{b}_\theta$ by interpolating $\mathbf{b}_j$, $1 \leq j \leq J$
9: Reconstruct the full-state solution $\mathbf{y}^\theta(t)$, Eq. (16)

3.3. Reduced Koopman Operator Interpolation (rKOI)

In the second method we developed, we directly interpolate the reduced Koopman operator $\mathbf{A}_{\theta r}$ using the reduced operators $\mathbf{A}_{jr}$. That is, initially, the same procedure as in the rEPI method is followed. However, the reduced eigen-pair ($\mathbf{\Lambda}_{\theta r}$, $\mathbf{W}_\theta$) is obtained directly from the eigen-value problem $\mathbf{A}_{\theta r}\mathbf{W}_{\theta r} = \mathbf{\Lambda}_{\theta r}\mathbf{W}_{\theta r}$. The reduced Koopman operator $\mathbf{A}_{\theta r}$ is obtaining for a point-wise Lagrangian interpolation of the J nearest neighbors matrices, $\mathbf{A}_{\mathrm{j}r}$.

The remainder of the procedure is then identical to the rEPI scheme: we interpolate $\mathbf{U}_{\theta r}$ in order to compute the projected DMD modes $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta r}\mathbf{W}_{\theta r}$.

We also interpolate the initial vector $\mathbf{b}_\theta$. Then, all the necessary information is at hand and Eq. (16) can be employed to reconstruct the time evolution of $\mathbf{y}^\theta$. The procedure is summarized as Algorithm 4.

The rKOI method also solves both limitations of stacked DMD. The time eigen-values are no longer shared between all parameters as they are calculated as part of the eigen-decomposition of the interpolated reduced Koopman operator and this method scales linearly with the number of parameters in the interpolation set. Furthermore, note that this method assumes smoothness across the components of the reduced Koopman operator over the parametric range when performing interpolations. Since these components are solely a function of the projection of the matrix of snapshots over its SVD modes (Eq. (8)), this assumption is much less constraining that assuming smoothness over the reduced Koopman operator eigen-pair.

Algorithm 4 Parametric DMD algorithm via reduced (discrete) Koopman operator interpolation

1: Solve Eq. (12) for all training parameters $\{\mu_j\}_{k=1}^{N_S}$ and collect temporal snapshots $\mathbf{S}_{\mu_j} = [\mathbf{y}(t_i;\mu_j)]_{i=0}^{m}$ for $1 \leq j \leq N_S$ ▷ Steps 1-4 are offline
2: Arrange matrix of snapshots in $\mathbf{S}_j^+$ and $\mathbf{S}_j^-$ data matrices, $1 \leq j \leq J$
3: Perform SVD of $\mathbf{S}_j^-$: $\mathbf{S}_j^- = \mathbf{U}_j \mathbf{\Sigma}_j \mathbf{V}_j^T$, $1 \leq j \leq J$
4: Retain r modes and compute the reduced Koopman operator $\mathbf{A}_{jr}$, $1 \leq j \leq J$, Eq. (17)
5: Interpolate the components of $\mathbf{A}_{jr}$ for $\mu_j \in \mathcal{S}$ to find $\mathbf{A}_{\theta r}$ ▷ Steps 5-10 are online
6: Perform the eigen-decomposition of $\mathbf{A}_{\theta r}$ to obtain the reduced eigen-pair $(\mathbf{\Lambda}_{\theta r} \mathbf{W}_{\theta r})$
7: Interpolate SVD-modes U_j to find $\mathbf{U}_{\theta r}$
8: Construct the DMD-mode $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta r} \mathbf{W}_{\theta r}$
9: Recover the initial coefficients $\mathbf{b}_\theta$ by interpolating $\mathbf{b}_j$, $1 \leq j \leq J$
10: Reconstruct the full-state solution $\mathbf{y}^\theta(t)$, Eq. (16)

4. Results

In this section, we test the accuracy and performance of the 3 parametric DMD methods presented above. For test cases, we have selected problems that encompass various types of physical operators, including diffusion, advection, reaction, and coupled multi-physics problems. The problems are, nonetheless, diffusion-reaction dominated as it is well known that DMD for advection/wave propagation phenomena requires reorientation along the propagation characteristics [68]. The first two problems are variations on heat conduction, one being more diffusive and the other one more advection-dominated. The last problem and the main focus of the results section is a coupled multi-physics radiative diffusion problem, where radiation energy and material temperature form a coupled system of differential equations that describes the absorption/re-emission of high-energy X-rays by matter.

The training set denotes the collection of snapshots, generated by sampling the parameter space (of dimension P) and running finite-element simulations at the sampled training points. The MOOSE code [69] was used to generate the snapshots for the radiative diffusion problem and the FEniCSx [70] Python package was used to generate the snapshots for the heat conduction problems; in both cases, linear finite elements were employed. All tests are carried out for new realizations of the parameters, i.e., parameter values that are not in the training set. In order to assess the accuracy of the parametric DMD approaches, the DMD-reconstructed solution is compared against a reference solution u^{ref}, obtained by actually solving the full-order problem at the new parameter value. To quantify the reconstruction errors from the various methods, we use the relative L_2 error norm, given in Eq. (19a) where u^{ref} is the reference snapshot from the testing set and u^{DMD} is the DMD reconstructed solution at the parameter value. The relative L_2 error norm is calculated for each time step. We also provide the time-averaged relative L_2 error norm given in Eq. (19b) where E is the relative L_2 error norm and N_T is the number of time steps.

$$E(t_i, \mu_j) = \frac{\|u^{\text{DMD}}(t_i, \mu_j) - u^{\text{ref}}(t_i, \mu_j)\|_2}{\|u^{\text{ref}}(t_i)\|_2}\,, \tag{19a}$$

$$\bar{E}(\mu_j) = \frac{\sum_i^{N_T} E(t_i, \mu_j)}{N_T} \,. \tag{19b}$$

We also average these error norms over parameter values sampled from the testing set:

$$\langle E(t_i) \rangle = \frac{\sum_j^{N_S} E(t_i, \mu_j)}{N_S} \,, \tag{19c}$$

and

$$\langle \bar{E} \rangle = \frac{\sum_j^{N_S} \tilde{E}(\mu_j)}{N_S} \,. \tag{19d}$$

4.1. Transient Nonlinear Diffusion Problem

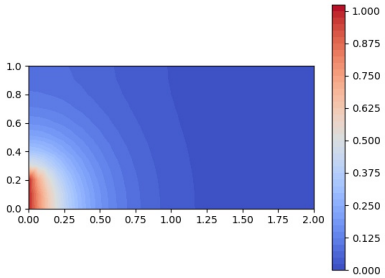
4.1.1. Problem Description

The first set of results deals with a transient nonlinear diffusion problem. The governing equation is given in Eq. (20):

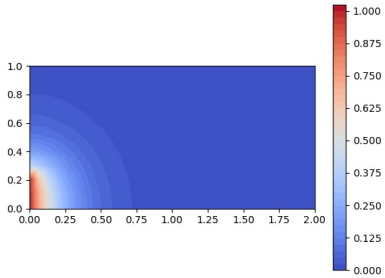
$$\frac{\partial T}{\partial t} + w \cdot \nabla T = \nabla \cdot k(T) \nabla T + f \qquad \forall \vec{r} \in \Omega \tag{20}$$

where $\Omega = \{(x, y) \in [0, 2] \otimes [0, 1]\}$ is the problem domain, with a Dirichlet boundary condition $T(x = 0, 0 < y < 0.2, t) = 1$ on the bottom of left wall, and homogeneous Neumann boundary condition elsewhere, $\nabla T \cdot n = 0$. The advection velocity for this problem is $w = 0.1\hat{i} + 0\hat{j}$. The thermal conductivity k is a non-linear function of temperature, given by Eq. (21), where a and b are parameters of the conductivity correlation. The unknown parameter we vary in this problem is the exponent coefficient b, while a is fixed to a value of 0.01; the parameter range for b is $0 \leq b \leq 4$. Note that, when $b = 0$, the problem is linear. Sample snapshot solutions for $b = 0$ and $b = 4$ are given in Fig. 1, at different time instants.

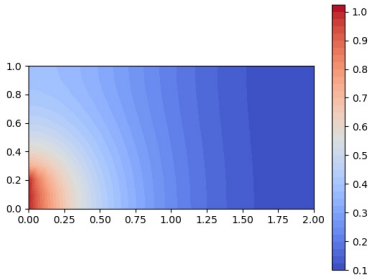
$$k(T) = a + T^b \,. \tag{21}$$



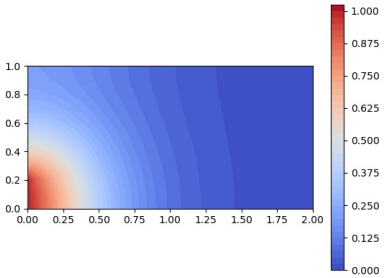
(a) Solution at $b = 0$, $t = 0.1$



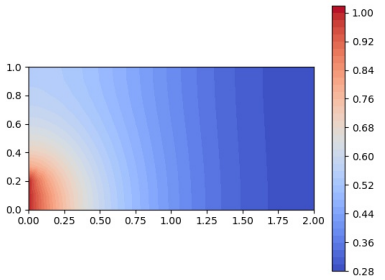
(b) Solution at $b = 4$, $t = 0.1$



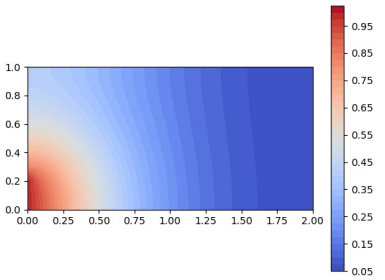
(c) Solution at $b = 0$, $t = 0.5$



(d) Solution at $b = 4$, $t = 0.5$



(e) Solution at $b = 0$, $t = 1.0$



(f) Solution at $b = 4$, $t = 1.0$

Figure 1: Sample Snapshot Solutions of the Nonlinear Diffusion Problem at the Extremes of the Parameter b Range Over Time.

4.1.2. Results

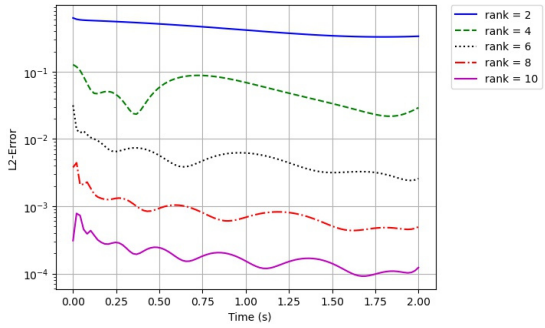
In this problem, we generated 50 snapshots with varying thermal conductivity non-linearity exponents b equally spaced in $[0, 4]$. A testing set of 20 snapshots, randomly sampled within the parameter range, was used. We compute the relative L_2 error norm over the testing set, $\langle E(t_i) \rangle$. Data points from the testing set are added until the change in the $\langle \bar{E} \rangle$ error is less than 0.1% or until all of the testing set samples have been utilized. This process is employed in all subsequent results as well, unless stated otherwise. In Fig. 2 we show the $\langle E(t_i) \rangle$ error and in Table 1 we show the $\langle \bar{E} \rangle$ error of each of the three methods as a function of DMD reconstruction rank.

Rank	2	4	6	8	10
Stacked Error	4.31×10^{-1}	5.21×10^{-2}	5.48×10^{-3}	8.86×10^{-4}	1.89×10^{-4}
rKOI Error	4.37×10^{-1}	5.21×10^{-2}	5.57×10^{-3}	8.82×10^{-4}	1.87×10^{-4}
rEPI Error	4.34×10^{-1}	5.24×10^{-2}	5.51×10^{-3}	7.91×10^{-3}	2.44×10^{-2}

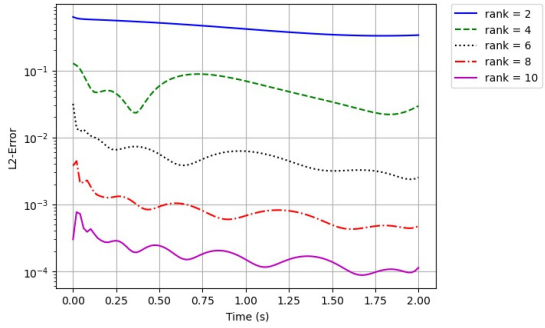
Table 1: $\langle \bar{E} \rangle$ Error in Rank for the Three Parametric DMD Methods (Nonlinear Diffusion Test Case)

For this problem, the error monotonically decreases as a function of the approximation rank r. This pattern will hold on all problems for most of the methods. On this problem, stacked DMD and reduced Koopman operator interpolation perform very similarly while the reduced Eigen-pair interpolation method presents higher errors. The three methods perform the same for rank 2, 4, and 6 reconstructions but rEPI has slightly worse error for rank 8 reconstruction and rank 10 reconstruction breaks with the general trend by having an increase in error with a higher rank reconstruction. This instability in rank is an undesirable trait because it is impossible to know a priori if increasing rank will decrease the error, making the rEPI method of questionable use for parametric DMD.

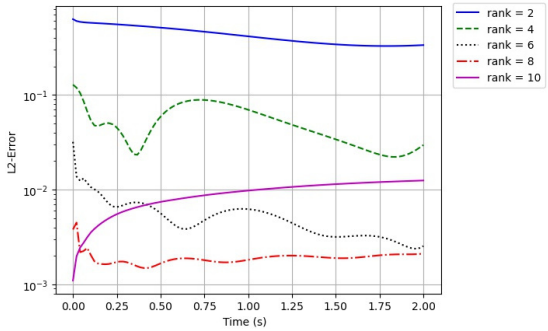
When any of the three parametric DMD methods are tested on a training point, they devolve to a non-parametric DMD method because there is no need



(a) Stacked DMD



(b) rKOI



(c) rEPI

Figure 2: Parameter-averaged, Time-dependent Relative L_2 Error, $\langle E(t_i)\rangle$, as a function of Rank for the Three Parametric DMD Methods on the Nonlinear Diffusion Problem.

for interpolation at the training points. For completeness, we show in Table 2 the average error for the non-parametric DMD at the training points as a function of rank. When comparing Table 1 and Table 2, we note that the error for the non-parametric DMD case is about smaller by a factor 2.

Rank	2	4	6	8	10
non-parametric DMD Error	2.61×10^{-1}	2.58×10^{-2}	3.13×10^{-3}	4.47×10^{-4}	7.22×10^{-5}

Table 2: $\langle \bar{E} \rangle$ Error in Rank for Non-Parametric DMD Method (Nonlinear Diffusion Test Case)

4.2. Incident-Jet Problem

4.2.1. Problem Description

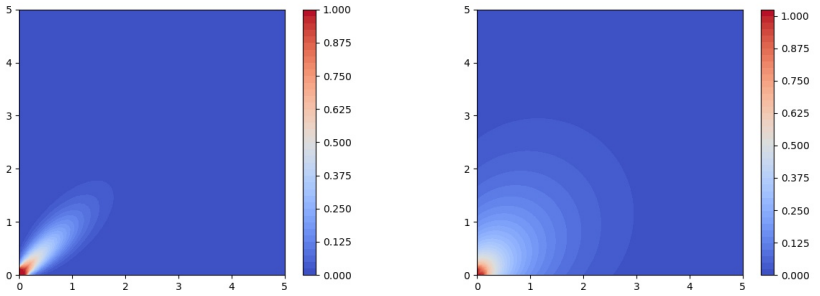
The second problem is an incident-jet problem. Compared to the previous nonlinear diffusion problem, the incident-jet test case is more advection-dominated. The thermal conductivity was chosen to be independent of temperature and varies in $0.2 \leq k \leq 5$. The governing equation is given in Eq. (22).

$$\frac{\partial T}{\partial t} + w \cdot \nabla T = \nabla \cdot k \nabla T \tag{22}$$

The domain for this problem is $\Omega = \{(x, y) \in [0, 5] \otimes [0, 5]\}$ with a Dirichlet boundary condition of $T(x < 0.1, y < 0.1, t) = sin(10\pi t)$ in the bottom left corner, and a homogeneous Neumann boundary condition of $\nabla T \cdot n = 0$ elsewhere. In this problem, the advection velocity w has x and y components such that $w = 5\hat{i} + 5\hat{j}$. The parameter we chose to vary in this problem is the thermal conductivity, the parameter range for k is $0.2 \leq k \leq 5$. Example solutions at $k = 0.2$ and $k = 5$ over time are given in Fig. 3. We compare the performance of the three parametric DMD methods on this problem in the following section.

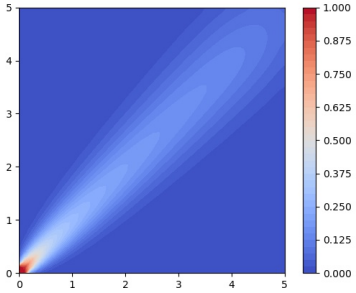
4.2.2. Problem Results

In this problem, we collected snapshots for 50 thermal conductivities k equally spaced in $[0.2, 5]$. A testing set of 20 snapshots, randomly sampled

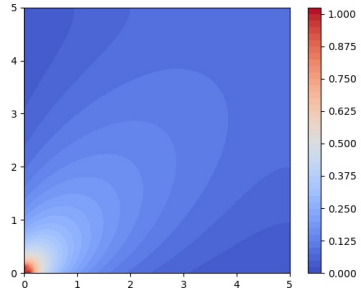


(a) $k = 0.2$, $t = 0.2$

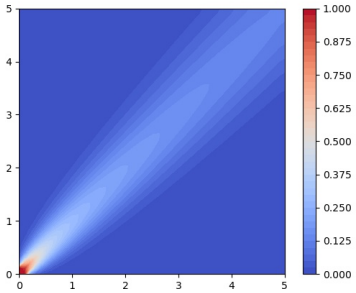
(b) $k = 5.0$, $t = 0.2$



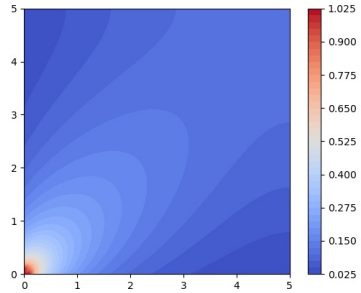
(c) $k = 0.2$, $t = 1.0$



(d) $k = 5.0$, $t = 1.0$



(e) $k = 0.2$, $t = 2.0$



(f) $k = 5.0$, $t = 2.0$

Figure 3: Sample Solutions of the Incident Jet Problem at the Extremes of the Parameter Range Over Time

Rank	2	4	6	8	10
Stacked Error	1.67×10^{-1}	8.45×10^{-1}	5.21×10^{-1}	5.00×10^{-1}	4.99×10^{-1}
rKOI Error	1.59×10^{-1}	6.57×10^{-1}	1.41×10^{-1}	2.14×10^{-3}	3.66×10^{-4}
rEPI Error	1.60×10^{-1}	6.52×10^{-2}	1.40×10^{-2}	2.47×10^{-3}	3.64×10^{-3}

Table 3: $\langle \bar{E} \rangle$ as a Function of Rank for the Three Parametric DMD Methods on the Incident-Jet Problem

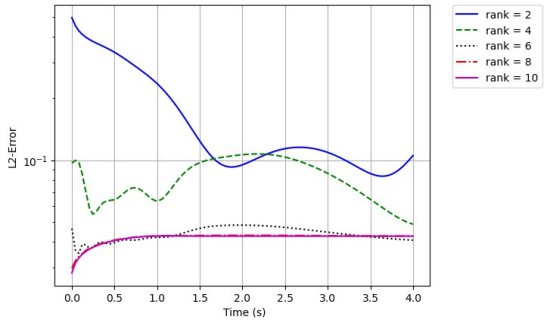
within the parameter range, was used. To get a relative L_2 error norm over the testing set we calculated both the $\langle E(t_i) \rangle$ error and the $\langle \bar{E} \rangle$ error. In Fig. 4 we show the ensemble-averaged error $\langle E(t_i) \rangle$ and in Table 3 we show the ensemble- and time-averaged error $\langle \bar{E} \rangle$ of each of the three methods, as a function of DMD reconstruction rank.

In this problem rKOI is the clear winner out of the three methods. The stacked DMD error saturates at a relatively large value, even when rank is increased. This may be due to the fact that DMD time eigen-values are generated and shared among all training snapshots, a drawback that is emphasized more in this advection-dominated problem. Because this method is the state of the art, we will continue employ it as baseline comparison in the remainder of this paper, even though its performance may be low at times. The rEPI method exhibits the same behavior as in the previous transient nonlinear diffusion problem, where the error does not monotonically decrease with rank. This behavior is disqualifying for a parametric DMD method due to the unpredictability of its behavior in rank. The rKOI method has also performed better than the rEPI approach for two test problems so far, thus we conclude that there are little benefits in the rEPI approach. As such, we will no longer present results using the rEPI approach.

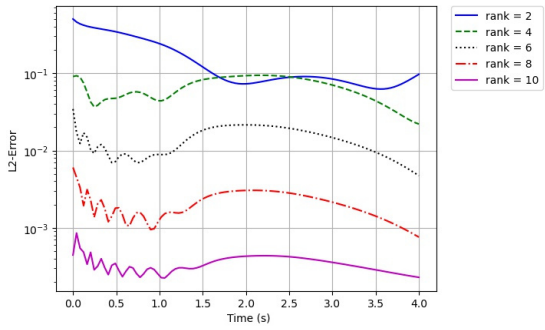
4.3. Radiative Diffusion Problem

4.3.1. Problem Description

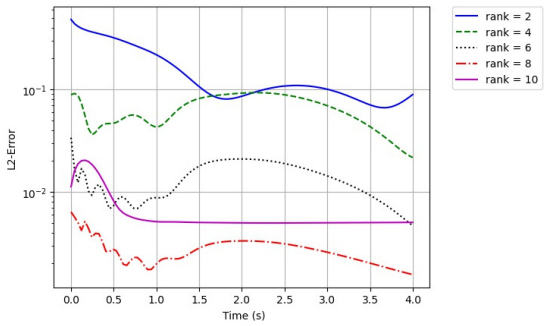
In this Section, we present an application of parametric DMD to a multi-physics radiative transfer problem. Radiation transport is approximated using



(a) Stacked DMD



(b) rKOI



(c) rEPI

Figure 4: $\langle E(t_i) \rangle$ Error as a Function of Rank for the Three Parametric DMD Methods (Incident-Jet test Case)

diffusion theory and the governing equations are given in Eqs. (23) (see also [71]):

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_T \nabla T) = -\sigma_a (T^4 - E)\,, \tag{23a}$$

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E)\,, \tag{23b}$$

where E is the radiation energy and T is the material temperature. The absorption opacity σ_a is given by Eq. (24) where α is a parameter, which we consider to be uncertain and thus can vary, and Z is the atomic number of the material, which will also be a parameter that varies as the material composition is modified in the problem:

$$\sigma_a(T) = \frac{Z^\alpha}{T^\alpha} \tag{24}$$

D_T is the material conduction coefficient, defined in Eq. (25a), and D_r is the radiation diffusion coefficient, defined in Eq. (25b).

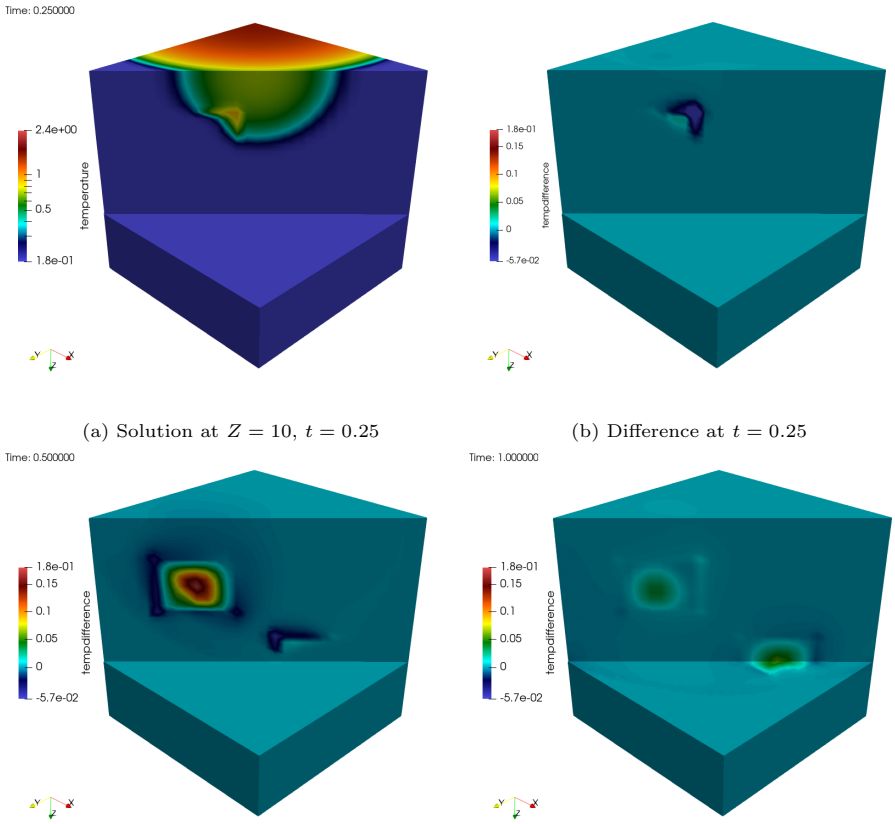
$$D_T(T) = 10^{-2} \times T^{5/2} \tag{25a}$$

$$D_r(E,T) = \frac{1}{3\sigma_a(T) + |\nabla E|/E} \tag{25b}$$

To test the various parametric DMD methods, parameters α and Z will be allowed to vary. A 3-D version of the 2-D problem given in [71] was created. Here, the 3-D background medium has a Z-value of 1, while two inclusions of high-Z material are present in the form of two cubes. The cubes are located at $(\frac{3}{32} \le x \le \frac{7}{32},\ \frac{9}{32} \le y \le \frac{13}{32},\ \frac{3}{32} \le z \le \frac{7}{32})$ and $(\frac{9}{32} \le x \le \frac{13}{32},\ \frac{3}{32} \le y \le \frac{7}{32},\ \frac{9}{32} \le z \le \frac{13}{32})$. Boundary conditions are reflective on the 3 adjacent faces of the domain and vacuum boundary conditions are applied on the 3 opposite faces. The initial condition for T and E are given in Eq. (26) and Eq. (27).

$$T(x,y,z,t=0) = 0.001 + 100 \times \left[\exp(-100(x^2+y^2+z^2)\right]^{\frac{1}{4}}\,, \tag{26}$$

$$E(x,y,z,t=0) = 0.001 + 100 \times \exp(-100(x^2+y^2+z^2)\,. \tag{27}$$



(a) Solution at $Z = 10$, $t = 0.25$

(b) Difference at $t = 0.25$

(c) Difference at $t = 0.5$

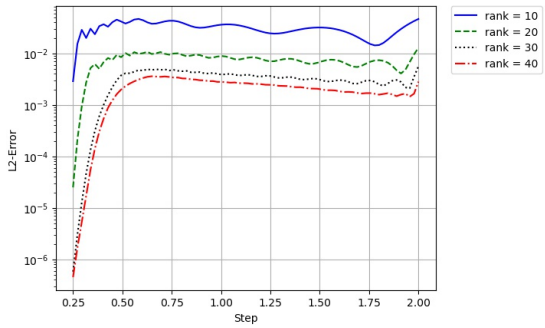
(d) Difference at $t = 1.0$

Figure 5: Sample Solution of the Radiative Diffusion Problem and Plots of the Difference Between the Solutions at $Z = 5$ and $Z = 15$ at Different Times

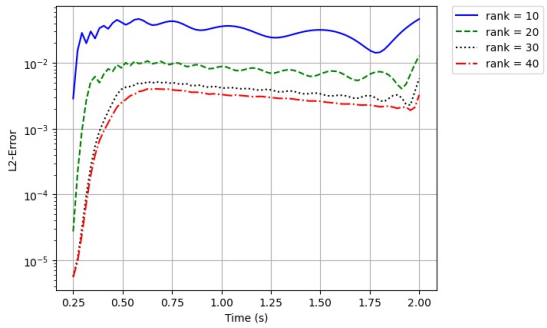
The input parameter space (α and Z) is two-dimensional in this problem. A sample solution with $\alpha = 3$ in the entire domain and $Z = 10$ in the two inclusions is shown in Figure 5a at time $t = 0.25$. In Figures 5b-5d, the difference between two solutions, with inclusion values of $Z = 5$ and $Z = 15$ respectively, is plotted at different time instants.

4.3.2. Method Comparison Results

In this problem, we first choose to vary only Z in the two inclusion, while keeping $\alpha = 3.0$ fixed. We study varying multiple parameters in the following



(a) Stacked DMD



(b) rKOI

Figure 6: $\langle E(t_i)\rangle$ Error in Rank for two Parametric DMD Methods (Radiative Diffusion Test Case)

sections. Due to the long run times of these simulations, a set of snapshots was divided into training and testing sets. The data set includes the range $Z \in [1, 15]$ at each integer value in that range. The odd-Z values are used for the training snapshots and the even-Z values for the testing values. In this problem we take the individual snapshots to include both the radiative energy E and the temperature T solution fields. The $\langle E(t_i)\rangle$ error is calculated as an average of E-values and T-values. In Fig. 6 we show the $\langle E(t_i)\rangle$ error of stacked DMD and rKOI as a function of rank. We also display the run time of the two methods in Table 4. This timing includes the SVD of the two nearest neighbors for the rKOI case and the single SVD containing all snapshots for the stacked DMD case.

Rank	10	20	30	40
Stacked Run Time (s)	8.6	8.7	8.8	8.9
rKOI Run Time (s)	3.4	3.5	3.5	3.5
% Speed Up	60.5%	59.8%	60.2%	60.7%

Table 4: Runtime of rKOI and Stacked DMD

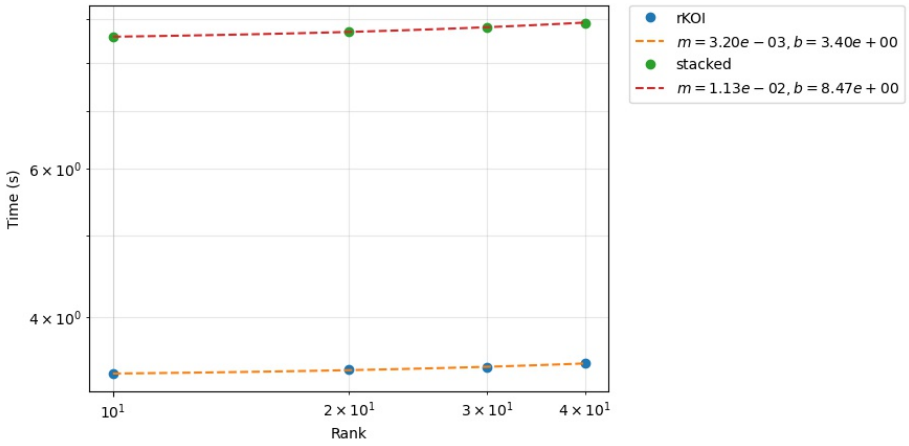


Figure 7: Rank Versus Run Time for rKOI and Stacked DMD

Rank	10	20	30	40
rKOI Error	3.11×10^{-2}	7.49×10^{-3}	3.49×10^{-3}	2.65×10^{-3}
Stacked Error	3.11×10^{-1}	7.50×10^{-3}	3.24×10^{-3}	2.16×10^{-3}

Table 5: $\langle \bar{E} \rangle$ Error in Rank for two Parametric DMD Methods (Radiative Diffusion Test Case)

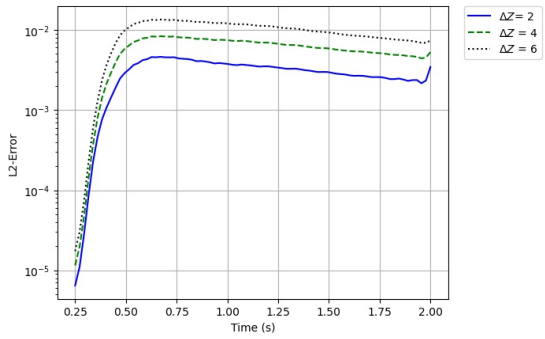
In this problem we see rKOI and stacked DMD perform similarly in terms of $\langle \bar{E} \rangle$ error with stacked DMD slightly beating out rKOI as shown in Table 5. However rKOI sees a speed up of up to 60.5% over stacked DMD, this is due to rKOI only requiring the two nearest neighbors to evaluate a point in-between while stacked DMD requires the SVD of all of the snapshots. In 7 we show the rank plotted versus time. The slope of this line will show how the two methods scale as a function of rank. It is clear that rKOI will scale more efficiently than stacked DMD from the respective 1.1×10^{-3} and 3.2×10^{-2} slopes. Our new method therefore is able to perform very close to the state of the art in terms of error while running up to 60% faster while also scaling favorably in terms of rank. These characteristics make our method much more attractive than stacked DMD if timing is of the upmost concern. This in addition to the superior performance on problems containing advection like the Incident-Jet Problem makes our new rKOI method more attractive in a large number of scenarios.

4.4. Parametric DMD Performance Study

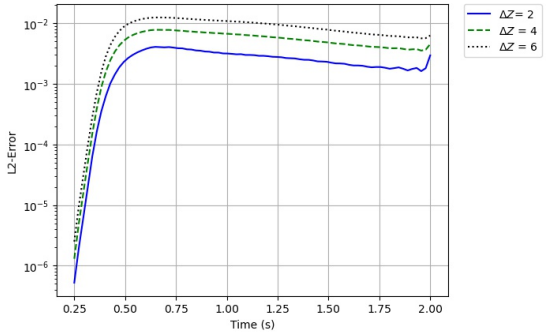
In this section we take the Radiative Diffusion problem and study the behavior of the rKOI and stacked DMD in methods more detail. We will look at the error as a function of number of snapshots, performance on a problem with 2 parameters, and how the multi-physics nature of the problem effects the behavior.

First we will look at the $\langle E(t_i) \rangle$ error as a function of the number of snapshots. Using the same parameter range for Z we use different ΔZ to generate 3 training sets, one with $\Delta Z = 2$ ($Z = [1, 3, 5, 7, 9, 11, 13, 15]$), one with $\Delta Z = 4$ ($Z = [1, 5, 9, 13]$), and one with $\Delta Z = 6$ ($Z = [1, 7, 13]$). The testing set for this problem is the even integers from 2 to 12. We used a rank 40 reconstruction for all values of ΔZ. The results are shown in Fig. 8.

Both methods perform similarly relative to their performance in rank. As could be expected the more snapshots that are used the lower the $\langle E(t_i) \rangle$ error of the solution is. This property could potentially be used in the future to



(a) $\langle E(t_i) \rangle$ Error in Rank for rKOI with Varied Parameter Spacing



(b) $\langle E(t_i) \rangle$ Error in Rank for Stacked DMD with Varied Parameter Spacing

Figure 8: $\langle E(t_i) \rangle$ Error in Rank for two Parametric DMD Methods for Varied Parameter Spacing

determine how many full order runs are needed in order to achieve a certain $\langle E(t_i) \rangle$ error in the the reduced solution over the entire parameter domain. This property is also very desirable because it means that before performing DMD the more snapshots that are generated the lower the error will be ensured to be at a new parameter value.

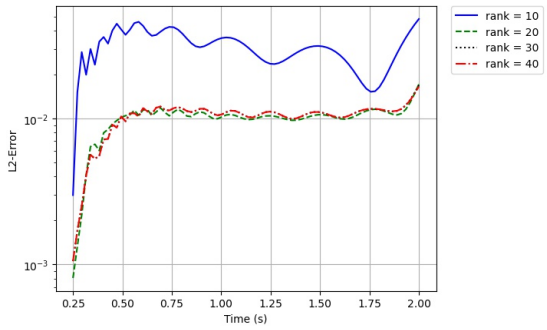
Next we will expand the problem to a 2-parameter problem where in addition to varying Z in the snapshots we also vary the α of the cubes. We will vary Z between 5 and 15 including every whole value of Z and 11 values of α equally spaced in $\alpha \in [2.5, 3.5]$. We took all even Z to be the testing set as well as every α with an even tenths place. The set containing each pairing of Z and α testing

points was used as the testing set and the set containing each paring of Z and α training points was used as the training set. We evaluated at each point using the same ending criteria as before, but with only using 5 testing points instead of the entire set and the results as a function of rank are shown in Fig. 9.

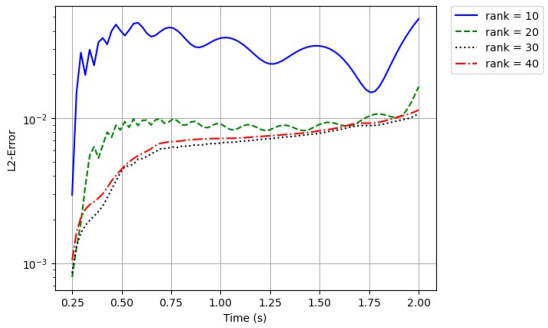
Even though this problem has many more snapshots than the single parameter case the error is increased with a much higher lower error bound. This error bound differs between the two methods with the stacked DMD method having a slightly lower bound. The distances in 2-D space are larger than in 1-D parameter space making the distance between parameters for the interpolation required to get the same performance much smaller requiring more snapshots overall. This method is still able to get a reasonable error much faster than the full order model however, meaning that there is still reason to consider DMD even in multi-parameter cases.

Next we will study how the coupling of this problem effects parametric DMD. As a reminder this problem is a set of coupled differential equations in both E and T. In all previous parts the error was reported averaged over both E and T. In this part we will present the error over only one portion of the solution at a time. In addition we will also perform DMD taking the snapshot to include both E and T as we have in the previous parts and perform DMD independently using only E or only T as the snapshot to predict only E or only T respectively. In this problem we go back to only vary Z as the parameter and keeping $\alpha = 3.0$. We take $Z \in [1, 15]$ with odd values composing the training set and even values composing the testing set. We use the same ending criteria as before and plot the results in Fig. 10.

In the rKOI case for both E and T the coupled DMD performs better at all time steps than the DMD performed with only E or T as the snapshot. This effect is less pronounced in the stacked DMD case which means it might be preferable to use the stacked DMD method for a multi-physics problem where only one of the physics is known. Better performance in the full data case makes sense given that the coupled nature of this physics would be better captured with both halves of the snapshot instead of only one.

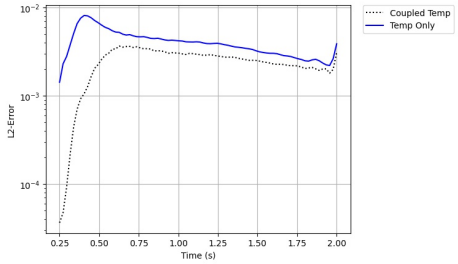


(a) $\langle E(t_i) \rangle$ Error in Rank for rKOI on 2-Parameter Problem

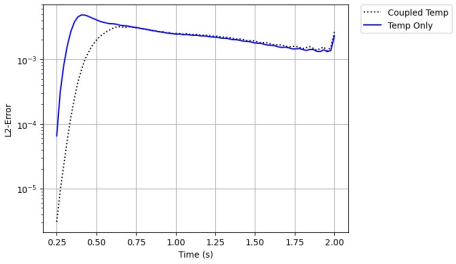


(b) $\langle E(t_i) \rangle$ Error in Rank for Stacked DMD on 2 Parameter Problem

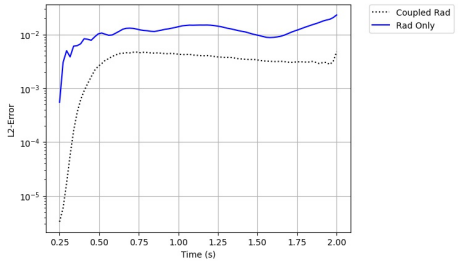
Figure 9: $\langle E(t_i) \rangle$ Error in Rank for two Parametric DMD Methods for 2 Parameter Problem



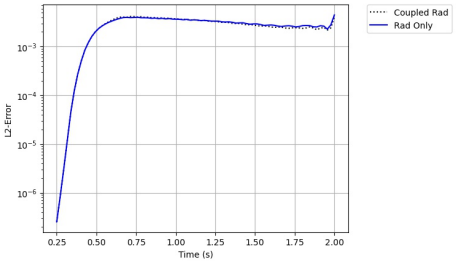
(a) rKOI Coupling Effects on Temperature Solution



(b) Stacked DMD Coupling Effects on Temperature Solution



(c) rKOI Coupling Effects on Radiation Solution



(d) Stacked DMD Coupling Effects on Radiation Solution

Figure 10: Study of Coupling Effects on the $\langle E(t_i) \rangle$ Error of two Parametric DMD Methods

We have shown that both rKOI and stacked DMD have a favorable relationship between error and parameter spacing, that in a multi-parameter case both parametric DMD methods have significantly worse performance than the single parameter case with the stacked DMD slightly out performing rKOI, and that both parametric DMD methods do significantly better when they are done with the full multi-physics snapshot instead of just one physics. All of these results show that the rKOI method is similarly robust to the state of the art method.

5. Conclusions

In this article, we analyzed three methods for extending dynamic mode decomposition (DMD) to deal with parametric problems. We analyzed the stacked-snapshots approach proposed by Sayadi et al. [65] and proposed two new methods based on the interpolation of the Koopman eigen-pair (rEPI) and the components of the reduced Koopman operator matrix (rKOI). Whereas the approach proposed by Sayadi et al. [65] scale, at best, linearly with the number of parameters in the training set, the two novel methods scale only with the number of training parameters used for the interpolation set. This makes the proposed methods more apt for multi-parametric problems where a large number of parameter samples are needed over the training set. In addition, for problems with large amounts of training points, it is possible that the memory requirements for performing the SVD in the case of the stacked DMD method exceed the RAM memory of the computer. Furthermore, a key limitation identified in the method proposed by Sayadi et al. [65] is that it forces the frequency of evolution in the DMD modes to be parameter-independent. The new methods overcome this limitation, but require smoothness in the reduced Koopman operator eigen-pair (rEPI) or components (rKOI) over the parameters in the interpolation set. Note that non-linear manifold interpolation, e.g., Grassmann manifold, matrix interpolation, e.g., solving the orthogonal Procrustes problem, or polynomial regression, e.g., radial-basis function regression, may mitigate these smoothness requirements. However, we have chosen to use simple linear

interpolation methods to better showcase the performance of the parametric-DMD methods, rather than the one of the interpolation methods used.

For the three problems analyzed, we observed that the smoothness requirements in the reduced Koopman operator eigen-pair is often too restrictive. The performance of the rEPI method deteriorated at high ranks in the DMD reconstruction. On the contrary, the smoothness assumption in the components of the reduced Koopman operator was often less constraining. In fact, the rKOI method surpassed in accuracy the stacked DMD method in problems where the solution field evolution was strongly dependent on the uncertain parameter, while it was significantly superior in computational performance in every problem. For all cases, the rKOI method was able to achieve a relative L_2 errors below 1% along the whole problem time-span. Moreover, for the multi-physics radiation diffusion problem, the rKOI method was also able to achieve average L_2 errors below 1% when reconstructing one field at the time, independently of the other. This single-physics reconstruction possibility offered by the rKOI method further reduces its computational burden. Note, however, that the performance of the rKOI method in single-physics reconstruction was inferior than the one when reconstructing the whole multi-physics problem. This deterioration in performance was not observed when using the stacked DMD approach.

Future work should look at improving the rKOI method and rEPI methods with adapted interpolation methods, e.g., non-linear Grassmann manifold or matrix interpolation methods. Also, these framework should be applied to different problems to continue testing and improving their performance.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344. Y.C. was funded by LDRD (21-FS-042; LLNL-JRNL-834453). Q.H., M.T., and J.R.

were funded by LLNL under research subcontract B640889.

References

[1] Steven F Ashby and John M May. Multiphysics simulations and petascale computing. In *Petascale Computing*, pages 101–120. Chapman and Hall/CRC, 2007.

[2] Michel Bergmann and Laurent Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics*, 227(16):7813–7840, 2008.

[3] M Avramova, K Ivanov, T Kozlowski, I Pasichnyk, W Zwermann, K Velkov, E Royer, A Yamaji, and J Gulliford. Multi-physics and multi-scale benchmarking and uncertainty quantification within oecd/nea framework. *Annals of Nuclear Energy*, 84:178–196, 2015.

[4] Sadia Younis, Muhammad Mubasher Saleem, Muhammad Zubair, and Syed Muhammad Tahir Zaidi. Multiphysics design optimization of rf-mems switch using response surface methodology. *Microelectronics Journal*, 71:47–60, 2018.

[5] Y Huang, MS Tillack, NM Ghoniem, JP Blanchard, LA El-Guebaly, and CE Kessel. Multiphysics modeling of the fw/blanket of the us fusion nuclear science facility (fnsf). *Fusion Engineering and Design*, 135:279–289, 2018.

[6] Yue Hu and Per Kvols Heiselberg. A new ventilated window with pcm heat exchanger—performance analysis and design optimization. *Energy and Buildings*, 169:185–194, 2018.

[7] Kaiyue Zeng, Jason Hou, Kostadin Ivanov, and Matthew Anderson Jessee. Uncertainty quantification and propagation of multiphysics simulation of the pressurized water reactor core. *Nuclear Technology*, 205(12):1618–1637, 2019.

[8] Bruno Sudret, Stefano Marelli, and Joe Wiart. Surrogate models for uncertainty quantification: An overview. In *2017 11th European conference on antennas and propagation (EUCAP)*, pages 793–797. IEEE, 2017.

[9] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[10] Dongli Huang, Hany Abdel-Khalik, Cristian Rabiti, and Frederick Gleicher. Dimensionality reducibility for multi-physics reduced order modeling. *Annals of Nuclear Energy*, 110:526–540, 2017.

[11] Rohit K Tripathy and Ilias Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375:565–588, 2018.

[12] Youngsoo Choi, Gabriele Boncoraglio, Spenser Anderson, David Amsallem, and Charbel Farhat. Gradient-based constrained optimization using a database of linear reduced-order models. *Journal of Computational Physics*, 423:109787, 2020.

[13] Youngsoo Choi David Amsallem, Matthew Zahr and Charbel Farhat. Design optimization using hyper-reduced-order models. *Structural and Multidisciplinary Optimization*, 51(4):919–940., 2015.

[14] Youngsoo Choi, Geoffrey Oxberry, Daniel White, and Trenton Kirchdoerfer. Accelerating design optimization using reduced order models, 2019.

[15] Sean McBane and Youngsoo Choi. Component-wise reduced order model lattice-type structure design. *Computer Methods in Applied Mechanics and Engineering*, 381:113813, 2021.

[16] Teeratorn Kadeethum, Daniel O'Malley, Jan Niklas Fuhg, Youngsoo Choi, Jonghyun Lee, Hari S. Viswanathan, and Nikolaos Bouklas. A framework for data-driven solution and parameter estimation of pdes using conditional generative adversarial networks, 2021.

[17] Francisco Chinesta, Antonio Huerta, Gianluigi Rozza, and Karen Willcox. Model order reduction. *Encyclopedia of Computational Mechanics, 2Eds., Wiley*, 2016.

[18] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[19] Khaled Salah Mohamed. *Machine learning for model order reduction*. Springer, 2018.

[20] Youngkyu Kim, Youngsoo Choi, David Widemann, and Tarek Zohdi. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 451:110841, 2022.

[21] Patrick Kürschner. Balanced truncation model order reduction in limited time intervals for large systems. *Advances in Computational Mathematics*, 44(6):1821–1844, 2018.

[22] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[23] Mauricio Tano, Jean Ragusa, Dominic Caron, and Patrick Behne. Affine reduced-order model for radiation transport problems in cylindrical coordinates. *Annals of Nuclear Energy*, 158:108214, 2021.

[24] Youngsoo Choi, Deshawn Coombs, and Robert Anderson. Sns: A solution-based nonlinear subspace method for time-dependent model order reduction. *SIAM Journal on Scientific Computing*, 42(2):A1116–A1146, 2020.

[25] Danish Rafiq and Mohammad Abid Bazaz. Model order reduction via moment-matching: A state of the art review. *Archives of Computational Methods in Engineering*, pages 1–21, 2021.

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

[27] Mohammad Abid Bazaz, S Janardhanan, et al. A review of parametric model order reduction techniques. In *2012 IEEE International Conference on Signal Processing, Computing and Control*, pages 1–6. IEEE, 2012.

[28] Nadine Aubry. On the hidden beauty of the proper orthogonal decomposition. *Theoretical and Computational Fluid Dynamics*, 2(5):339–352, 1991.

[29] Youngkyu Kim, Karen Wang, and Youngsoo Choi. Efficient space–time reduced order model for linear dynamical systems in python using less than 120 lines of code. *Mathematics*, 9(14), 2021.

[30] Youngsoo Choi and Kevin Carlberg. Space–time least-squares petrov–galerkin projection for nonlinear model reduction. *SIAM Journal on Scientific Computing*, 41:A26–A58, 01 2019.

[31] Moritz Sieber, C Oliver Paschereit, and Kilian Oberleithner. Spectral proper orthogonal decomposition. *Journal of Fluid Mechanics*, 792:798–828, 2016.

[32] Jean-Christophe Loiseau, Bernd R Noack, and Steven L Brunton. Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490, 2018.

[33] Dylan Matthew Copeland, Siu Wun Cheung, Kevin Huynh, and Youngsoo Choi. Reduced order models for lagrangian hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 388:114259, 2022.

[34] Kevin Carlberg, Youngsoo Choi, and Syuzanna Sargsyan. Conservative model reduction for finite-volume models. *Journal of Computational Physics*, 371:280–314, 2018.

[35] Chi Hoang, Youngsoo Choi, and Kevin Carlberg. Domain-decomposition least-squares petrov–galerkin (dd-lspg) nonlinear model reduction. *Computer Methods in Applied Mechanics and Engineering*, 384:113997, 2021.

[36] D Xiao, F Fang, AG Buchan, CC Pain, IM Navon, and A Muggeridge. Non-intrusive reduced order modelling of the navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 293:522–541, 2015.

[37] Marvin L Adams and Edward W Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in nuclear energy*, 40(1):3–159, 2002.

[38] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, 2011.

[39] Youngsoo Choi, Peter Brown, William Arrighi, Robert Anderson, and Kevin Huynh. Space–time reduced order model for large-scale linear dynamical systems with application to boltzmann transport problems. *Journal of Computational Physics*, 424:109845, 2021.

[40] Ryan McClarren. *Uncertainty Quantification and Predictive Computational Science: A Foundation for Physical Scientists and Engineers*. 01 2018.

[41] Mengwu Guo and Jan S. Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 2018.

[42] Joslin Goh, Derek Bingham, James Paul Holloway, Michael J. Grosskopf, Carolyn C. Kuranz, and Erica Rutter. Prediction and computer model calibration using outputs from multifidelity simulators. *Technometrics*, 55(4):501–512, 2013.

[43] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[44] Markus Mrosek, Carsten Othmer, and Rolf Radespiel. Variational autoencoders for model order reduction in vehicle aerodynamics. In *AIAA AVIATION 2021 FORUM*, page 3049, 2021.

[45] SiHun Lee, Kijoo Jang, Haeseong Cho, Haedong Kim, and SangJoon Shin. Parametric non-intrusive model order reduction for flow-fields using unsupervised machine learning. *Computer Methods in Applied Mechanics and Engineering*, 384:113999, 2021.

[46] T. Kadeethum, F. Ballarin, Y. Choi, D. O'Malley, H. Yoon, and N. Bouklas. Non-intrusive reduced order modeling of natural convection in porous media using convolutional autoencoders: Comparison with linear subspace techniques. *Advances in Water Resources*, 160:104098, feb 2022.

[47] Hamidreza Eivazi, Hadi Veisi, Mohammad Hossein Naderi, and Vahid Esfahanian. Deep neural networks for nonlinear model order reduction of unsteady flows. *Physics of Fluids*, 32(10):105104, 2020.

[48] Thomas Simpson, Nikolaos Dervilis, and Eleni Chatzi. Machine learning approach to model order reduction of nonlinear systems via autoencoder and lstm networks. *Journal of Engineering Mechanics*, 147(10):04021061, 2021.

[49] Felix Fritzen, Mauricio Fernández, and Fredrik Larsson. On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling. *Frontiers in Materials*, 6:75, 2019.

[50] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

[51] Hugo FS Lui and William R Wolf. Construction of reduced-order models for fluid flows using deep feedforward neural networks. *Journal of Fluid Mechanics*, 872:963–994, 2019.

[52] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[53] N Benjamin Erichson, Michael Muehlebach, and Michael W Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019.

[54] Shaowu Pan and Karthik Duraisamy. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 19(1):480–509, 2020.

[55] Pin Wu, Junwu Sun, Xuting Chang, Wenjie Zhang, Rossella Arcucci, Yike Guo, and Christopher C Pain. Data-driven reduced order model with temporal convolutional neural network. *Computer Methods in Applied Mechanics and Engineering*, 360:112766, 2020.

[56] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[57] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *arXiv preprint arXiv:1710.04340*, 2017.

[58] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

[59] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

[60] Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids*, 26(11):111701, 2014.

[61] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2):024103, 2014.

[62] J Nathan Kutz, Xing Fu, and Steven L Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

[63] Daniel Dylewsky, Molei Tao, and J Nathan Kutz. Dynamic mode decomposition for multiscale nonlinear physics. *Physical Review E*, 99(6):063311, 2019.

[64] Zhang Dang, Yong Lv, Yourong Li, and Cancan Yi. Optimized dynamic mode decomposition via non-convex regularization and multiscale permutation entropy. *Entropy*, 20(3):152, 2018.

[65] Taraneh Sayadi, Peter J Schmid, Franck Richecoeur, and Daniel Durox. Parametrized data-driven decomposition for bifurcation analysis, with application to thermo-acoustically unstable systems. *Physics of Fluids*, 27(3):037102, 2015.

[66] Satyavir Singh, M Abid Bazaz, and Shahkar Ahmad Nahvi. A scheme for comprehensive computational cost reduction in proper orthogonal decomposition. *dynamics*, 2(2):8, 2018.

[67] Igor Mezić. On applications of the spectral theory of the koopman operator in dynamical systems and control theory. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 7034–7041. IEEE, 2015.

[68] Hannah Lu and Daniel M. Tartakovsky. Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena. *Journal of Computational Physics*, 407:109229, 2020.

[69] Cody J. Permann, Derek R. Gaston, David Andrš, Robert W. Carlsen, Fande Kong, Alexander D. Lindsay, Jason M. Miller, John W. Peterson,

Andrew E. Slaughter, Roy H. Stogner, and Richard C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 2020.

[70] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The fenics book*. Springer, 2016.

[71] V.A. Mousseau and D. A. Knoll. New physics-based preconditioning of implicit methods for non-equilibrium radiation diffusion. *Journal of Computational Physics*, 190(1):42–51, 2003.