Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2023-0163

Vol. **35**, No. 1, pp. 38-69
January 2024

An Adaptive Method Based on Local Dynamic Mode Decomposition for Parametric Dynamical Systems

Qiuqi Li[1], Chang Liu[1], Mengnan Li[2,*] and Pingwen Zhang[3,4]

[1] *School of Mathematics, Hunan University, Changsha 410082, China.*
[2] *College of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China.*
[3] *School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China.*
[4] *School of Mathematical Sciences, Laboratory of Mathematics and Applied Mathematics, Peking University, Beijing 100871, China.*

Received 12 June 2023; Accepted (in revised version) 15 October 2023

Abstract. Parametric dynamical systems are widely used to model physical systems, but their numerical simulation can be computationally demanding due to nonlinearity, long-time simulation, and multi-query requirements. Model reduction methods aim to reduce computation complexity and improve simulation efficiency. However, traditional model reduction methods are inefficient for parametric dynamical systems with nonlinear structures. To address this challenge, we propose an adaptive method based on local dynamic mode decomposition (DMD) to construct an efficient and reliable surrogate model. We propose an improved greedy algorithm to generate the atoms set Θ based on a sequence of relatively small training sets, which could reduce the effect of large training set. At each enrichment step, we construct a local sub-surrogate model using the Taylor expansion and DMD, resulting in the ability to predict the state at any time without solving the original dynamical system. Moreover, our method provides the best approximation almost everywhere over the parameter domain with certain smoothness assumptions, thanks to the gradient information. At last, three concrete examples are presented to illustrate the effectiveness of the proposed method.

AMS subject classifications: 35R60, 60H35, 65M99, 68W99

Key words: Parametric dynamical systems, dynamic mode decomposition, greedy algorithm, Taylor expansion, surrogate model.

1 Introduction

Parametrized partial differential equations arise in various engineering and applied science problems, including heat and mass transfer, acoustics, solid and fluid mechanics,

[*]Corresponding author. *Email addresses:* qli28@hnu.edu.cn (Q. Li), chang16@hnu.edu.cn (C. Liu), mengnanli@njupt.edu.cn (M. Li), pzhang@pku.edu.cn (P. Zhang)

electromagnetics, and finance. Due to limited knowledge about physical properties, such as material properties and geometric features, the model inputs often contain parameters or uncertainties. Estimating the unknown parameter values and quantifying their effects usually require a large number of realizations, ranging from thousands to millions. Consequently, traditional full-order techniques, such as finite element or finite volume methods, are often too computationally expensive, especially when dealing with nonlinear, multiphysics, and time-dependent phenomena. To overcome these challenges, many model order reduction (MOR) methods have been developed to construct efficient surrogate models, including generalized Polynomial chaos expansion [1, 2], tensor decomposition-based methods (e.g., the Proper Generalized Decomposition [3, 4] and the Variable-separation method [5–7]), and projection-based methods (e.g., the Reduced Basis method [8–10] and the Proper orthogonal decomposition (POD) Galerkin method [11, 12]). These MOR methods aim to reduce computational costs while accurately capturing the most important features of the original system.

MOR methods aim to construct an approximate model in a low-dimensional subspace of the solution space [13–16]. The success of these methods relies on the assumption that the solution manifold can be embedded in a low-dimensional space. However, the important class of problems given by parametric dynamical systems usually induce rough solution manifold with slowly decaying Kolmogorov n-widths. This implies that traditional MOR methods are generally not effective. In recent years, there has been a growing interest in the development of MOR techniques for parametric dynamical systems to overcome the limitations of linear global approximations. A large class of methods consider the dynamical low rank (DLR) approximation (see [17–21]) which allows both the deterministic and stochastic basis functions to evolve in time. Other strategies based on deep learning (DL) algorithms were proposed in [22–24] to construct the efficient surrogate model for time-dependent parametrized PDEs. In this contribution, we try to combine dynamic mode decomposition with the local Taylor approximation to construct an efficient and reliable approximation of input-output relationship (i.e. surrogate model) for parametric dynamical systems.

Data driven methods have received widespread attention. Koopman operator [25] can be an effective data driven tool. It can transform a nonlinear system in state space into a linear system in observation function space. Koopman operator is an infinite dimensional linear operator acting on the observation function space. The spectral decomposition of the Koopman can capture linear systems in the observation function space. For numerical computation, it is necessary to approximate the Koopman operator in a finite dimensional subspace. Dynamic mode decomposition is used to approximate Koopman eigenvalues and eigenvectors in the subspace are spanned by a set of observation functions. DMD [26] describes the dynamical system in an equation-free manner and can be used for prediction and control. DMD is a spatio-temporal matrix decomposition method that connects spatial dimensionality-reduction technology and Fourier transforms in time. In the standard DMD method [26], identity functions are used as a finite dimensional set of observation functions for approximate the Koopman operator.

The DMD mode calculated using standard DMD is a projection approximation. Tu et al. [27] proposed that the exact DMD method can obtain accurate DMD modes. Williams et al. [28] developed the extended dynamic mode decomposition (EDMD) method by applying the Koopman operator to a given dictionary of observation functions. When these dictionary functions are sufficiently rich, the matrix obtained by EDMD converges to the Koopman operator [29]. Kernel DMD [30] is to reduce the computational complexity by intelligent choices of the observables. Koopman operator theory can be extended to nonlinear dynamical systems with inputs and control (KIC). Dynamic mode decomposition with control (DMDc) [31] is used to perform numerical computation KIC. DMDc is a modal decomposition method for discovering spatio-temporal modes from measured data of complex systems with inputs. DMDc utilizes both the measurements of the state and the external control to construct a dynamical system with inputs. It can be used to predict and design controllers for complex systems. For parametric dynamical systems with observation noise, there are many DMD variants based on Koopman operator theory, such as total least-squares DMD [32], subspace DMD [33] and so on. For dynamical systems with random parameters, combining K-nearest-neighbors regression [34] or parametric interpolation [35] with DMD can effectively reduce online computing costs.

In this work, we develop an adaptive method based on local dynamic mode decomposition for parametric dynamical systems. This method is an offline-online mechanism. In the offline stage, we first generate the set of optimal samples Θ based on a sequence of relatively small training sets. Then for each atom in Θ, we construct the corresponding local surrogate model using the Taylor linear expansion and DMD (or DMDc). In the online stage, for any instance of parameters, the local surrogate model is selected by finding the closest atom with the Euclidean measure, and the output is calculated by the corresponding local model.

The main contribution of this work is the development of an adaptive local DMD approximation for solving parametric dynamical systems. We propose an improved Greedy method to generate the atoms set Θ based on a sequence of relatively small training sets. The main idea of our improved method is running the greedy algorithm first on a relatively small training set and updating the output over a local denser training set at each enrichment step. Compared with the classic Greedy technique, our improved method could mitigate the effect of a large training set on the computation efficiency with the similar atoms set Θ. Using the Taylor expansion, we construct local DMD approximations of the parametric dynamical systems for each atom in Θ. Therefore, our method shares the same merits as DMD for dynamical systems. The important one is that it can be used to predict the state of the dynamical system at any time without solving the original dynamical system. Due to the local nature of our proposed method, the online computational cost does not increase as more atoms are added in the adaptive process. In addition, with certain smoothness assumptions, the approximation of our proposed method is guaranteed to converge to the reference solution almost everywhere over the parameter domain, thanks to the gradient information. To demonstrate the performance of our proposed method, we apply it to various nonlinear parametric dynamical systems and successively obtain the efficient and accurate surrogate models. This also demonstrates the accuracy of our adaptive local DMD approximation as a predictive tool outside of the training time region.

The paper is organized as follows. In Section 2, we describe the problem tackled in this work. In Section 3, the surrogate model based on local Taylor approximation for parametric dynamical systems is presented. In Section 4, we introduce the Koopman operator, DMD, and DMDc algorithm. In Section 5, we introduce the improved Greeddy method based on adaptive enriching strategy to generate the optimal atoms set and give a summary of our full approach. Section 6 provides three numerical examples to illustrate the performance of the proposed method. Finally, we make some conclusions and comments in Section 7.

2 Problem formulation

Let $(\Omega,\mathcal{B},\mathbb{P})$ be a probability space, where Ω is the sample space containing the set of elementary events ω, $\mathcal{B}$ is the σ- algebra on Ω, $\mathbb{P}$ is the probability measure on $\mathcal{B}$. Let D be a given bounded physical domain with Lipschitz continuous boundary ∂D. We consider the following parameter-dependent dynamical systems

$$u_t(x,t;\boldsymbol{\xi})=\mathcal{L}(x,t,u,I;\boldsymbol{\xi}), \quad x\in D, \quad \boldsymbol{\xi}\in\Omega, \quad t\in[0,T], \tag{2.1}$$

where $\mathcal{L}$ is a differential operator and may contain random coefficients and/or stochastic forces, and I is the model inputs, such as the boundary condition $x\in\partial D$ and initial condition $t=0$. Let $\boldsymbol{\xi}:\Omega\to\mathbb{R}^m$ be a set of real-valued random variables. We denote $u(x,t;\boldsymbol{\xi}):D\times[0,T]\times\Omega\to\mathbb{R}$ as the solution to this parameter-dependent dynamical systems.

High-fidelity simulation (also known as full model, e.g., finite element method and finite difference method) requires a large number of degrees of freedom and a huge number of time steps and iterations. Therefore, the computational cost is prohibitive when solving complex problems, especially for many-query problems such as quantifying the effects of parameters and estimating unknown parameter values. Our goal is to construct a surrogate model $\hat{u}$ of u so that the output of the system (2.1) can be evaluated without solving the full model. Based on the surrogate model $\hat{u}$, we can estimate the statistics of u efficiently. More generally, given a function $G(u)$, the expectation of $G(u)$ can be approximated using Monte Carlo as follows

$$\mathbb{E}[G(u)]\approx\mathbb{E}[G(\hat{u})]=\int_\Omega G(\hat{u}(\boldsymbol{\xi}))\mathbb{P}(\mathrm{d}\boldsymbol{\xi})\approx\frac{1}{N}\sum_{i=1}^{N}G(\hat{u}(\boldsymbol{\xi}_i)).$$

The recently developed dynamic mode decomposition is a powerful new technique for the discovery of dynamical systems from high-dimensional data and is an innovative tool for integrating data with dynamical systems theory. In this work, we attempt to combine

DMD with the local Taylor approximation to build a surrogate model $\hat{u}(x,t;\boldsymbol{\xi})$ of the state u in the system (2.1) such that

$$\hat{u}(x,t;\boldsymbol{\xi}) \approx \sum_{i=1}^{m+1} \boldsymbol{\Phi}_i(x,t)\alpha_i(\boldsymbol{\xi}), \tag{2.2}$$

where $\{\boldsymbol{\Phi}_i(x,t)\}_{i=1}^{m+1}$ are the spatial bases, and $\{\alpha_i(\boldsymbol{\xi})\}_{i=1}^{m+1}$ are the stochastic bases. In the next section, we briefly introduce the surrogate model based on local Taylor approximation.

3 Surrogate model based on local Taylor approximation

Assuming that the random field of PDE solution u possesses enough regularity [36]. Given a set of deterministic samples $\Theta := \{\boldsymbol{\xi}_k\}_{k=1}^M$, we construct the surrogate model based on local Taylor approximation. An offline-online computational decomposition is achieved to improve efficiency. In the offline stage, we need to determine the set of optimal samples from the training sets scattered in the parameter space and construct the corresponding local surrogate model. We attempt to seek an Taylor expansion of the solution around Θ as follows:

$$u(\boldsymbol{\xi}) = \sum_{k=1}^{M} \mathbf{I}_k \left[u_0 + \nabla_{\boldsymbol{\xi}} u(\boldsymbol{\xi}_k)(\boldsymbol{\xi}-\boldsymbol{\xi}_k) + \frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\xi}_k)^{\mathrm{T}} \mathbf{H}(\boldsymbol{\xi}_k)(\boldsymbol{\xi}-\boldsymbol{\xi}_k) + \cdots \right],$$

where $u_0 = u(\boldsymbol{\xi}_k)$, $\mathbf{H}$ is the Hessian matrix of $u(\boldsymbol{\xi})$, i.e.,

$$\mathbf{H}_{ij}(\boldsymbol{\xi}_k) = \frac{\partial^2 u}{\partial \xi_i \partial \xi_j}(\boldsymbol{\xi}_k), \quad \mathbf{I}_k = \begin{cases} 1 & \text{if } \boldsymbol{\xi} \in \Xi_{\text{local}}^k, \\ 0 & \text{else}, \end{cases}$$

and Ξ_{local}^k is the *local* testing set determined by $\boldsymbol{\xi}_k$. To facilitate implementation and simplify notation, for each atom $\boldsymbol{\xi}_k \in \Theta$, we consider the linear formulation $\hat{u}_k(\xi)$ of Taylor expansion to construct our local surrogate model such as

$$\hat{u}_k(\boldsymbol{\xi}) = u(\boldsymbol{\xi}_k) + \nabla_{\boldsymbol{\xi}} u(\boldsymbol{\xi}_k)(\boldsymbol{\xi}-\boldsymbol{\xi}_k), \quad \boldsymbol{\xi} \in \Xi_{\text{local}}^k. \tag{3.1}$$

Here $\nabla_{\boldsymbol{\xi}} u(\boldsymbol{\xi}_k) = [\partial_1 u(\boldsymbol{\xi}_k), \partial_2 u(\boldsymbol{\xi}_k), \cdots, \partial_m u(\boldsymbol{\xi}_k)]$, and ∂_i denotes the partial derivative with respect to the i-th component of random variables $\boldsymbol{\xi}$. The surrogate model (3.1) can be formulated as Eq. (2.2) by taking the spatial bases as

$$\boldsymbol{\Phi}_{\boldsymbol{i}}(\boldsymbol{x},\boldsymbol{t}) = \begin{cases} u(\boldsymbol{\xi}_k), & i=1, \\ \partial_{i-1} u(\boldsymbol{\xi}_k), & i=2,\cdots,m+1, \end{cases}$$

and the stochastic bases as

$$\alpha_i(\boldsymbol{\xi}) = \begin{cases} 1, & i=1, \\ \xi^{i-1} - \xi_k^{i-1}, & i=2,\cdots,m+1, \end{cases}$$

where ξ_k^i and ξ^i is the i-th components of $\boldsymbol{\xi}_k$ and $\boldsymbol{\xi}$. We note that the calculation cost of building $\hat{u}_k(\boldsymbol{\xi})$ is mainly dominated by the constructions of the spatial bases, that is,

$$\mathbf{\Phi}^k(x,t)=[u(\boldsymbol{\xi}_k),\partial_1 u(\boldsymbol{\xi}_k),\partial_2 u(\boldsymbol{\xi}_k),\cdots,\partial_m u(\boldsymbol{\xi}_k)].$$

In this attempt, we employ DMD to build the approximations of $\mathbf{\Phi}^k(x,t)$ and improve the efficiency of our surrogate model. The details of DMD will be introduced in Section 4.

Remark 3.1. For given atoms set Θ, we need to solve $M(m+1)$ deterministic problems to obtain the states $\{u(\boldsymbol{\xi}_k)\}_{k=1}^M$ and their derivatives $\{\partial_i u(\boldsymbol{\xi}_k)\}_{i=1,k=1}^{i=m,k=M}$. For each atom $\boldsymbol{\xi}_k$, $u(\boldsymbol{\xi}_k)$ is the solution of the original problem (2.1) with the fixed sample. By taking the derivative with respect to the i-th component of random variables ξ_i for Eq. (2.1), we have $\partial_i u(\boldsymbol{\xi}_k)$ satisfying the following equation

$$(\partial_i u)_t(x,t;\boldsymbol{\xi}_k)=\mathcal{L}_i(x,t,u,I;\boldsymbol{\xi}_k),\quad x\in D,\quad \boldsymbol{\xi}_k\in\Omega,\quad t\in[0,T], \tag{3.2}$$

where $\mathcal{L}_i$ denote the derivatives of $\mathcal{L}$ with respect to the i-th components, respectively. The $\mathcal{L}_i$ is similar to Eq. (2.1), relying on the space x, time t and model inputs I. We note that $u(\boldsymbol{\xi}_k)$ can be obtained by solving the original problem and is known in Eq. (3.2). Under some circumstances [37], $u(\boldsymbol{\xi}_k)$ and its derivatives $\{\partial_i u(\boldsymbol{\xi}_k)\}_{i=1}^{i=m}$ are equipped with the same equation, just with some different inputs, which could bring convenience to the calculation of offline stage.

Each atom $\boldsymbol{\xi}_k$ is associated with a cluster and acts as a good representation of the other elements of the cluster. In the online stage, the best local surrogate is automatically recommended by searching the corresponding atom $\boldsymbol{\xi}_k$, which is the closest one to the test sample $\boldsymbol{\xi}\in\Omega$ in the parameter space. The local approximation based on Taylor expansion is accurate only when the distance between the test sample and an atom is "sufficiently small". Therefore, selecting an optimal atom set is the key step for our method, which will be introduced in Section 5.

4 Dynamic mode decomposition

For a fixed sample $\boldsymbol{\xi}_k$, $u(\boldsymbol{\xi}_k)$ or $\{\partial_i u(\boldsymbol{\xi}_k)\}_{i=1}^{i=m}$ satisfies the following class of continuous time dynamical system,

$$u_t(x,t;\boldsymbol{\xi}_k)=\mathcal{F}(x,t,u,I;\boldsymbol{\xi}_k),\quad x\in D,\quad t\in[0,T]. \tag{4.1}$$

Here, $\mathcal{F}$ is a differential operator, which depends on the space x, time t, model inputs I, state u and its spatial partial derivatives. I represents the model inputs, such as source term, initial and boundary conditions. In this system, u is just an expression symbol. In Eq. (4.1), if u represents $u(\boldsymbol{\xi}_k)$, then dynamical system (4.1) represents Eq. (2.1) taking a fixed sample $\boldsymbol{\xi}_k$. If u represents $\partial_i u(\boldsymbol{\xi}_k)$, then Eq. (4.1) is equivalent to Eq. (3.2). Using the finite element method or finite difference method to solve Eq. (4.1) produces the following discrete dynamical system with external inputs

$$\boldsymbol{u}_{n+1}=F(\boldsymbol{u}_n,\boldsymbol{z}_n), \quad \boldsymbol{u}\in\mathcal{M}, \quad \boldsymbol{z}\in\mathcal{N}, \tag{4.2}$$

where $\boldsymbol{u}_n=\boldsymbol{u}(t_0+n\triangle t)$, $\mathcal{M}$ denotes the state space, and F is a map from $\mathcal{M}$ to itself, $\boldsymbol{z}_n=\boldsymbol{z}(t_0+n\triangle t)$ represents external input, which can be introduced by the source item and boundary conditions.

If the dynamical system (4.1) has no external inputs, for example, the source term equals zero and the boundary conditions are homogeneous, then we can obtain the following discrete dynamical system without external inputs

$$\boldsymbol{u}_{n+1}=F(\boldsymbol{u}_n), \quad \boldsymbol{u}\in\mathcal{M}. \tag{4.3}$$

F can be a linear mapping or nonlinear mapping, which depends on $\mathcal{F}$ in the dynamical system (4.1). When $\mathcal{F}$ is a nonlinear mapping, solving nonlinear dynamical system takes a large amount of computing resources. For different parameters $\boldsymbol{\xi}$, it is necessary to solve the nonlinear dynamic system to obtain $u(\boldsymbol{\xi})$. Therefore, for nonlinear dynamical systems with random parameters, it is very expensive to solve nonlinear dynamical systems multiple times. In this section, we will present a data-driven method based on the Koopman operator to solve the nonlinear dynamical system without knowing the specific form of the system.

Koopman operator is a linear operator acting on the observation function space. It does not linearize the dynamical system (4.2) or (4.3). Koopman operator can transform a nonlinear dynamical system into a linear system in the observation function space. For discrete dynamical systems without external inputs (4.3), the Koopman operator is defined as follows.

Definition 4.1. (Koopman operator) Let $\boldsymbol{G}(\mathcal{M})$ is an infinite dimensional observation function space for any scalar-valued observable function $g:\mathcal{M}\to\mathbb{R}$. The Koopman operator $\mathcal{K}:\boldsymbol{G}(\mathcal{M})\to\boldsymbol{G}(\mathcal{M})$ is defined by

$$\mathcal{K}g(\boldsymbol{u}):=g\big(F(\boldsymbol{u})\big), \quad \forall g\in\mathbf{G}.$$

So the nonlinear dynamical system (4.3) can be lifted to the following linear problem,

$$\boldsymbol{u}_{n+1}=F(\boldsymbol{u}_n) \;\;\Rightarrow\;\; g(\boldsymbol{u}_{n+1})=\mathcal{K}g(\boldsymbol{u}_n).$$

$\mathcal{K}$ is an infinite-dimensional operator. In order to be numerically computable, we restrict the infinite-dimensional system to a finite-dimensional invariant subspace $\mathcal{G}(\mathcal{M})\subseteq\boldsymbol{G}(\mathcal{M})$.

Suppose that there exists an invariant subspace $\mathcal{G}$ of $\mathcal{K}$, i.e.,

$$\mathcal{K}g\in\mathcal{G}, \quad \forall g\in\mathcal{G}.$$

Let a set of observation functions $\{g_1,\cdots,g_q\}(q<\infty)$ span $\mathcal{G}$. We restrict $\mathcal{K}$ to $\mathcal{G}$ and denote it by $\mathcal{K}|_{\mathcal{G}}$. Let $\boldsymbol{g}=[g_1,\cdots,g_q]^T$ be a vector-valued observation. Then $\mathcal{K}|_{\mathcal{G}}$ has a matrix-form representation $\boldsymbol{K}$ with respect to $\{g_1,\cdots,g_q\}$, i.e.,

$$\boldsymbol{g}(F(\boldsymbol{u}))=\begin{bmatrix} g_1(F(\boldsymbol{u})) \\ g_2(F(\boldsymbol{u})) \\ \vdots \\ g_q(F(\boldsymbol{u})) \end{bmatrix}=\begin{bmatrix} \mathcal{K}g_1(\boldsymbol{u}) \\ \mathcal{K}g_2(\boldsymbol{u}) \\ \vdots \\ \mathcal{K}g_q(\boldsymbol{u}) \end{bmatrix}=\boldsymbol{K}\boldsymbol{g}(\boldsymbol{u}).$$

The spectral decomposition theory of the Koopman operator can give an expression for the observation functions. Therefore, we consider the eigendecomposition of the matrix $\boldsymbol{K}$. Let,

$$\boldsymbol{K}\boldsymbol{\psi}_j=\lambda_j\boldsymbol{\psi}_j,\quad \boldsymbol{\omega}_j^T\boldsymbol{K}=\lambda_j\boldsymbol{\omega}_j^T,$$

where λ_j is the eigenvalue, $\boldsymbol{\psi}_j$ is the right eigenvector, $\boldsymbol{\omega}_j$ is the left eigenvector and $\boldsymbol{\omega}_i^T\boldsymbol{\psi}_j=\delta_{ij}$. If the matrix $\boldsymbol{K}\in\mathbb{R}^{q\times q}$ has q linearly independent eigenvectors, then any $\boldsymbol{g}\in\mathcal{G}$ can be expressed by a linear combination of the eigenvectors, i.e.,

$$\boldsymbol{g}(\boldsymbol{u})=\sum_{j=1}^{q}v_j(\boldsymbol{u})\boldsymbol{\psi}_j, \tag{4.4}$$

where $v_j(\boldsymbol{u})=\boldsymbol{\omega}_j^T\boldsymbol{g}(\boldsymbol{u})$. For a sequential time series, repeatedly applying the Koopman operator to Eq. (4.4) gives

$$\boldsymbol{g}(\boldsymbol{u}_n)=\sum_{j=1}^{q}\lambda_j^n v_j(\boldsymbol{u}_0)\boldsymbol{\psi}_j. \tag{4.5}$$

Therefore, we can use the eigenvalues and eigenvectors of the Koopman operator and $\boldsymbol{g}(\boldsymbol{u}_0)$ to evaluate the observation function $\boldsymbol{g}(\boldsymbol{u})$ at any time without knowing the specific expression of F.

Dynamic mode decomposition algorithm only uses the observation data to compute an approximation to the Koopman eigenvalues and eigenvectors. DMD was introduced in fluid mechanics by Schmidt [26]. It is an equation-free and data-driven method capable of providing accurate assessments of the spatio-temporal structures in a given complex system. Suppose we have a snapshot sequence of data $\{\boldsymbol{u}_0,\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_r\}$, and DMD is used to extract the features of data. Let $\boldsymbol{g}$ be the given observation function. We define the data matrices of observables $\mathbf{Y}_0$ and $\mathbf{Y}_1$ as follows:

$$\mathbf{Y}_0=\begin{bmatrix} | & | & & | \\ \boldsymbol{g}(\boldsymbol{u}_0) & \boldsymbol{g}(\boldsymbol{u}_1) & \cdots & \boldsymbol{g}(\boldsymbol{u}_{r-1}) \\ | & | & & | \end{bmatrix},$$

$$\mathbf{Y}_1=\begin{bmatrix} | & | & & | \\ \boldsymbol{g}(\boldsymbol{u}_1) & \boldsymbol{g}(\boldsymbol{u}_2) & \cdots & \boldsymbol{g}(\boldsymbol{u}_r) \\ | & | & & | \end{bmatrix}.$$

Let the DMD matrix

$$A := \mathbf{Y}_1 \mathbf{Y}_0^{\dagger},$$

where $\mathbf{Y}_0^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{Y}_0$. So,

$$g(u_{n+1}) \approx A g(u_n).$$

The eigenvectors and eigenvalues of A are the DMD modes and eigenvalues. DMD matrix A is an approximation of matrix K. It may be very expensive to do eigendecomposition directly on the matrix A. Tu et al. [27] proposed exact DMD, which reconstructs the nonzero eigenvalues and eigenvectors of A by calculating the eigendecomposition of a low-dimensional projection of A. The main steps of exact DMD are summarized in Algorithm 1.

Algorithm 1 Exact DMD [27]

```
Given snapshots$\{g(u_0), g(u_1), \cdots, g(u_r)\}$
Set $Y_0 = [g(u_0), g(u_1), \cdots, g(u_{r-1})]$ and $Y_1 = [g(u_1), g(u_2), \cdots, g(u_r)]$.
Compute the reduced SVD of $Y_0, Y_0 = U\Sigma V^*$, where $U \in \mathbb{R}^{q\times\tilde{r}}, \Sigma \in \mathbb{R}^{\tilde{r}\times\tilde{r}}, V \in \mathbb{R}^{r\times\tilde{r}}$, $\tilde{r}$ is the truncated rank.
Define $\widetilde{A} := U^T Y_1 V \Sigma^{-1}$.
Compute eigenvalues and eigenvectors of $\widetilde{A} W = W \Lambda$.
Set $\Psi = Y_1 V \Sigma^{-1} W$.
Then $g_{\mathrm{DMD}}(u_n) = \Psi \Lambda^n d$, where $d = \Psi^{\dagger} g(u_0), \quad n = 0, 1, 2, \cdots, r, r+1, \cdots$
Finally, $u_n = g^{-1}(g_{\mathrm{DMD}}(u_n)), n = 0, 1, \cdots, r+1, \cdots$, where the inverse function $g^{-1}$ is in the sense of least-squares if $g$ is not invertible.
```

For the discrete dynamical system that allows for external inputs

$$u_{n+1} = F(u_n, z_n), \quad u \in \mathcal{M}, \quad z \in \mathcal{N}.$$

Koopman operator theory is generalized to include exogenous inputs and control [31]. We define a set of scalar-valued observables that are functions of the state and the input where $g: \mathcal{M} \times \mathcal{N} \to \mathbb{R}$. Each observable is an element of an infinite dimensional Hilbert space $\mathcal{H}$. The following definition is given for the Koopman operator with inputs and control.

Definition 4.2. (Koopman operator with inputs and control) The KIC $\mathcal{K}: \mathcal{H} \to \mathcal{H}$ acts on the Hilbert space of observables

$$\mathcal{K} g(u_n, z_n) := g\big(F(u_n, z_n), z_{n+1}\big).$$

KIC is related to a method called DMDc [38]. DMD with control utilizes both the measurements of the system and the applied external control to extract the underlying dynamics and the input-output characteristics in an equation-free manner. The KIC can be connected to DMDc for linear observable measurements. This connection parallels the link between Koopman operator and DMD.

The observables $g(\boldsymbol{u},\boldsymbol{z})$ can be partitioned into those dependent on the state $\boldsymbol{g_u}$, control $\boldsymbol{g_z}$ and both $\boldsymbol{g_{uz}}$. Let

$$g(\boldsymbol{u}_n,\boldsymbol{z}_n)=\begin{bmatrix} \boldsymbol{g}_{u,n} \\ \boldsymbol{g}_{z,n} \\ \boldsymbol{g}_{uz,n} \end{bmatrix},$$

then similar to the Koopman operator in a finite dimensional invariant subspace, for KIC we have

$$g(\boldsymbol{u}_{n+1},\boldsymbol{z}_{n+1})=\begin{bmatrix} \boldsymbol{g}_{u,n+1} \\ \boldsymbol{g}_{z,n+1} \\ \boldsymbol{g}_{uz,n+1} \end{bmatrix}=\mathcal{K}g(\boldsymbol{u}_n,\boldsymbol{z}_n)=\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{D} & \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{G} & \boldsymbol{H} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{g}_{u,n} \\ \boldsymbol{g}_{z,n} \\ \boldsymbol{g}_{uz,n} \end{bmatrix}.$$

If we choose the observation function $\boldsymbol{g_{uz}}=\boldsymbol{0}$, we have

$$\boldsymbol{g}_{u,n+1}=\boldsymbol{A}\boldsymbol{g}_{u,n}+\boldsymbol{B}\boldsymbol{g}_{z,n}. \tag{4.6}$$

The DMDc can be established by choosing linear observables for the state and control, such that $\boldsymbol{g_u}=\boldsymbol{u},\boldsymbol{g_z}=\boldsymbol{z},\boldsymbol{g_{uz}}=\boldsymbol{0}$. With these linear observables, the KIC operator reduces to DMDc. Based on the linear system (4.6), we present the main steps of DMDc in Algorithm 2.

5 Adaptive enhancement strategies

We propose an adaptive approach to generate the atoms set $\Theta=\{\boldsymbol{\xi}_k\}_{k=1}^M$ sequentially. The selection of Θ is determined by the current surrogate solution $\hat{u}(\boldsymbol{\xi})$. Given a current set of atoms, the next atom $\boldsymbol{\xi}_{k+1}$ is identified by the current set of atoms with the largest error over a given training set.

5.1 Classical greedy method

An effective error estimator $\Delta_N(\boldsymbol{\xi})$ is crucial for both the efficiency and the reliability of our method. People usually utilize a posteriori error estimate to define the error estimator $\Delta_N(\boldsymbol{\xi})$ [39–41]. For the reasonable selection of $\Delta_N(\boldsymbol{\xi})$, we can refer to Appendix A in [7]. The classical greedy algorithm (CG) presented in Algorithm 3 can be invoked to build the atoms set Θ. We first define the parameter training set Ξ_{train}, which is a finite subset of Ω.

In a greedy method, the next atom is selected by computing the error estimator $\Delta_N(\boldsymbol{\xi})$ and searching for the largest one. An intuitive method is calculating the error estimators for each sample of the training set Ξ_{train}. We usually require a significant number of training samples to provide adequate representation. This will substantially impact on the computation efficiency. To alleviate the problem, we adopt the following adaptive strategy [41].

Algorithm 2 DMDc [38]

```
Given state snapshots$\{\boldsymbol{g_u}(\boldsymbol{u}_0),\cdots,\boldsymbol{g_u}(\boldsymbol{u}_r)\}$ and control snapshots: $\{\boldsymbol{g_z}(\boldsymbol{z}_0),\cdots,\boldsymbol{g_z}(\boldsymbol{z}_{r-1})\}$
1: Set $\boldsymbol{Y}_0=[\boldsymbol{g_u}(\boldsymbol{u}_0),\cdots,\boldsymbol{g_u}(\boldsymbol{u}_{r-1})]\in\mathbb{R}^{q\times r}$,  $\boldsymbol{Y}_1=[\boldsymbol{g_u}(\boldsymbol{u}_1),\cdots,\boldsymbol{g_u}(\boldsymbol{u}_r)]\in\mathbb{R}^{q\times r}$,
   and $\mathrm{Y}=\{\boldsymbol{g_z}(\boldsymbol{z}_0),\cdots,\boldsymbol{g_z}(\boldsymbol{z}_{r-1})\}\in\mathbb{R}^{s\times r}$.
2: Set the augmented data matrix $\boldsymbol{\Omega}=\begin{bmatrix}\boldsymbol{Y}_0\\ \mathrm{Y}\end{bmatrix}$ and the augmented operator matrix
   $\boldsymbol{G}=[\boldsymbol{A},\ \boldsymbol{B}]$.
3: Compute the reduced SVD of $\boldsymbol{\Omega}$, $\boldsymbol{\Omega}=\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^*$, where $\tilde{\boldsymbol{U}}\in\mathbb{R}^{(q+s)\times\tilde{r}}$,$\tilde{\boldsymbol{\Sigma}}\in\mathbb{R}^{\tilde{r}\times\tilde{r}}$,$\tilde{\boldsymbol{V}}\in\mathbb{R}^{r\times\tilde{r}}$, $\tilde{r}$
is the truncated rank. Let $\tilde{\boldsymbol{U}}=\begin{bmatrix}\tilde{\boldsymbol{U}}_1\\ \tilde{\boldsymbol{U}}_2\end{bmatrix}$, where $\tilde{\boldsymbol{U}}_1\in\mathbb{R}^{q\times\tilde{r}}$,$\tilde{\boldsymbol{U}}_2\in\mathbb{R}^{s\times\tilde{r}}$.
4: Compute the reduced SVD of $\boldsymbol{Y}_1$, $\boldsymbol{Y}_1=\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^*$, where $\hat{\boldsymbol{U}}\in\mathbb{R}^{q\times\hat{r}}$,$\hat{\boldsymbol{\Sigma}}\in\mathbb{R}^{\hat{r}\times\hat{r}}$,$\hat{\boldsymbol{V}}\in\mathbb{R}^{r\times\hat{r}}$, $\hat{r}$ is
the truncated rank.
5: Compute the reduced-order approximation of $\boldsymbol{A}$ and $\boldsymbol{B}$,

   $$\tilde{\boldsymbol{A}}=\hat{\boldsymbol{U}}^*\boldsymbol{Y}_1\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}_1^*\hat{\boldsymbol{U}},$$

   $$\tilde{\boldsymbol{B}}=\hat{\boldsymbol{U}}^*\boldsymbol{Y}_1\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}_2^*.$$

   We can form the ROM

   $$\tilde{\boldsymbol{g}}_{\boldsymbol{u},n+1}=\tilde{\boldsymbol{A}}\tilde{\boldsymbol{g}}_{\boldsymbol{u},n}+\tilde{\boldsymbol{B}}\boldsymbol{g}_{\boldsymbol{z},n},\quad n=0,1,\cdots,r+1,\cdots.$$

6: Compute $\boldsymbol{g}_{\mathrm{DMD}}(\boldsymbol{u}_n):=\boldsymbol{g}_{\boldsymbol{u},n}=\hat{\boldsymbol{U}}\tilde{\boldsymbol{g}}_{\boldsymbol{u},n},\quad n=0,1,\cdots,r+1,\cdots$.
7: Finally, $\boldsymbol{u}_n=\boldsymbol{g}^{-1}(\boldsymbol{g}_{\mathrm{DMD}}(\boldsymbol{u}_n))$, $n=0,1,\cdots,r+1,\cdots$, where the inverse function $\boldsymbol{g}^{-1}$ is in
the sense of least-squares if $\boldsymbol{g}$ is not invertible.
```

5.2 Adaptive enriching based on dynamic training sets

To mitigate the effect of a large Ξ_{train} on the computation efficiency and the limitation that a fixed smaller training set may not provide adequate atoms set, we perform the greedy algorithm on a relatively small training set and then sequentially update the current training set at each enrichment step. Let Ξ_{bkg} be a background samples set consisting of a sufficiently large number of Monte Carlo samples in Ω. Meanwhile, we define a coarse training set Ξ_{train} uniformly distributed over Ω, and its size is relatively small.

Now we develop an algorithm to obtain the atoms set Θ and our surrogate model $\hat{u}(\boldsymbol{\xi})$. To this end, we initialize the first atom as $\boldsymbol{\xi}_1=\mathbb{E}[\boldsymbol{\xi}]$, and $\Theta=\{\mathbb{E}[\boldsymbol{\xi}]\}$. Then the initial surrogate model $\hat{u}(\boldsymbol{\xi})$ is defined as

$$\hat{u}_1(\boldsymbol{\xi})=u(\mathbb{E}[\boldsymbol{\xi}])+\partial_{\boldsymbol{\xi}}u(\mathbb{E}[\boldsymbol{\xi}])\cdot(\boldsymbol{\xi}-\mathbb{E}[\boldsymbol{\xi}]),$$

where $u(\mathbb{E}[\boldsymbol{\xi}])$ and $\partial_{\boldsymbol{\xi}}u(\mathbb{E}[\boldsymbol{\xi}])$ are approximated by DMD or DMDc. At step k, we choose

```
Algorithm 3 Classical Greedy
Initialize
1: The error tolerance $\varepsilon_{\text{tol}}$, $N=1$.
2: Select an initial parameter value $\boldsymbol{\xi}_1 \in \Xi_{\text{train}}$ , set $\Theta = \{\boldsymbol{\xi}_1\}$.
3: Adaptive phase
  while $\max_{\boldsymbol{\xi}\in\Xi_{\text{train}}} \Delta_N(\boldsymbol{\xi}) > \varepsilon_{\text{tol}}$ do
    Choose $\boldsymbol{\xi}_{N+1} = \underset{\boldsymbol{\xi}\in\Xi_{\text{train}}}{\text{argmax}} \Delta_N(\boldsymbol{\xi})$.
    $\Theta = \Theta \bigcup \{\boldsymbol{\xi}_{N+1}\}$.
    $N = N+1$.
  end while
```

the quasi-optimal sample as

$$\boldsymbol{\xi}_k \quad = \begin{cases} \mathbb{E}[\boldsymbol{\xi}],\ \boldsymbol{\xi} \in \Xi_{\text{train}}, & k=1, \\ \arg\max\limits_{\boldsymbol{\xi}\in\Xi_{\text{train}}} \Delta_{k-1}(\boldsymbol{\xi}), & k \geq 2. \end{cases}$$

For the reasonable selection of $\Delta_k(\boldsymbol{\xi})$, please refer to Appendix A in [7]. Then we define the current *local* training set as $\Xi^k_{\text{local}} = \{\boldsymbol{\xi}_j \,|\, \delta(\boldsymbol{\xi}_j,\boldsymbol{\xi}_k) < \epsilon_k,\ \boldsymbol{\xi}_j \in \Xi_{\text{bkg}}\}$ with $\delta(\cdot,\cdot)$ being the Euclidean distance between two points. Here ϵ_k can be seen as the radius of Ξ^k_{local}. Based on the *local* training set Ξ^k_{local}, we update the current atom $\boldsymbol{\xi}_k$ as follows

$$\boldsymbol{\xi}_k = \underset{\boldsymbol{\xi}\in\Xi^k_{\text{local}}}{\text{argmax}} \Delta_{k-1}(\boldsymbol{\xi}),$$

and update atoms set as $\Theta = \Theta \cup \{\boldsymbol{\xi}_k\}$. Then the k-th sub-surrogate model is given by

$$\hat{u}_k(\boldsymbol{\xi}) = u(\boldsymbol{\xi}_k) + \partial_{\boldsymbol{\xi}} u(\boldsymbol{\xi}_k) \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}_k),$$

with $u(\mathbb{E}[\boldsymbol{\xi}])$ and $\partial_{\boldsymbol{\xi}} u(\mathbb{E}[\boldsymbol{\xi}])$ approximated by DMD or DMDc. We define the utilization rate of the background set Ξ_{bkg} at step k as

$$\eta = \frac{|\cup_{i=1}^k \Xi^k_{\text{local}}|}{|\Xi_{\text{train}}|},$$

where $|\alpha|$ denotes the sample size of α.

Remark 5.1. A training set that is too large can increase offline computation and hinder offline speed, while too small can reduce the efficiency of online approximations. Therefore, finding an appropriate balance between the size of the training set and computational resources is crucial in this method.

To overcome these limitations and obtain an approximate error accuracy that is no worse than the Classical Greedy method, we select the local training set Ξ^k_{local} based on all samples from the background set Ξ_{bkg} that a maximum neighborhood distance can cover. This maximum neighborhood distance ϵ_k is determined by the minimum Euclidean distance between the sample $\boldsymbol{\xi}_k$ and other sample points in the coarse training set Ξ_{train}.

Algorithm 4 describes the procedure for the adaptive construction of Θ and $\hat{u}(\boldsymbol{\xi})$. The whole approach introduced in Sections 3-5 are also summarized in Fig. 1.

Algorithm 4 Adaptive construction of Θ and $\hat{u}(\boldsymbol{\xi})$

```
Initialize
The maximum number of atom $N_{\text{atom}}$, the error tolerance $\varepsilon_{\text{tol}}$.
Form a background set $\Xi_{\text{bkg}}$ and a coarse training set $\Xi_{\text{train}}$.
Select an initial atom $\Theta = \{\mathbb{E}[\boldsymbol{\xi}]\}$.
Set $N = 1$, Flag $= 0$ and $\varepsilon_{\max} = +\infty$.
Adaptive phase
  while $N < N_{\text{atom}}$ and $\varepsilon_{\text{tol}} < \varepsilon_{\max}$ do
    Evaluate the error indicator $\Delta_N(\boldsymbol{\xi})$ of $\hat{u}(\boldsymbol{\xi})$ at each $\boldsymbol{\xi} \in \Xi_{\text{train}}$.
    Choose $\boldsymbol{\xi}_k = \underset{\boldsymbol{\xi} \in \Xi_{\text{train}}}{\operatorname{argmax}} \Delta_N(\boldsymbol{\xi})$, $\varepsilon_{\max} = \Delta_N(\boldsymbol{\xi}_k)$.
    Set a local training set $\Xi^k_{\text{local}}$ from $\Xi_{\text{bkg}}$ relies on $\boldsymbol{\xi}_k$.
    Compute $u(\boldsymbol{\xi}_i)$, $\boldsymbol{\xi}_i \in \Xi^k_{\text{local}}$ with Algorithm 1 or 2 and update Flag $=$ Flag $+1$.
    Find $\boldsymbol{\xi}_{N+1} = \underset{\boldsymbol{\xi} \in \Xi^k_{\text{local}}}{\operatorname{argmax}} \Delta_N(\boldsymbol{\xi})$, $\Theta = \Theta \cup \{\boldsymbol{\xi}_{N+1}\}$.
    Update $u(\boldsymbol{\xi}_{N+1})$ and $\nabla u(\boldsymbol{\xi}_{N+1})$ with Algorithm 1 or 2.
    $N = N+1$.
  end while
Store the atoms set $\Theta$ and the surrogate model $\hat{u}(\boldsymbol{\xi})$ for online calculation.
Output $\eta = \frac{\text{Flag}}{|\Xi_{\text{bkg}}|}$ denotes the utilization rate of $\Xi_{\text{bkg}}$.
```

6 Numerical results

In this section, we present a few numerical examples to demonstrate the performance of the proposed adaptive local DMD method. In Section 6.1, we apply the proposed method to linear heat equations with a random variable to verify the accuracy of the adaptive local DMD method. In Section 6.2, we consider the nonlinear Burgers equation to further demonstrate the effectiveness of the proposed method. In Section 6.3, the reaction-diffusion equation with two parameters is considered to present the performance of the adaptive local DMD method on high-dimensional parameters.

In this paper, we denote the reference solution as u_{ref}, which uses high-fidelity solutions such as Finite Element Method (FEM). The solution obtained using the local Taylor

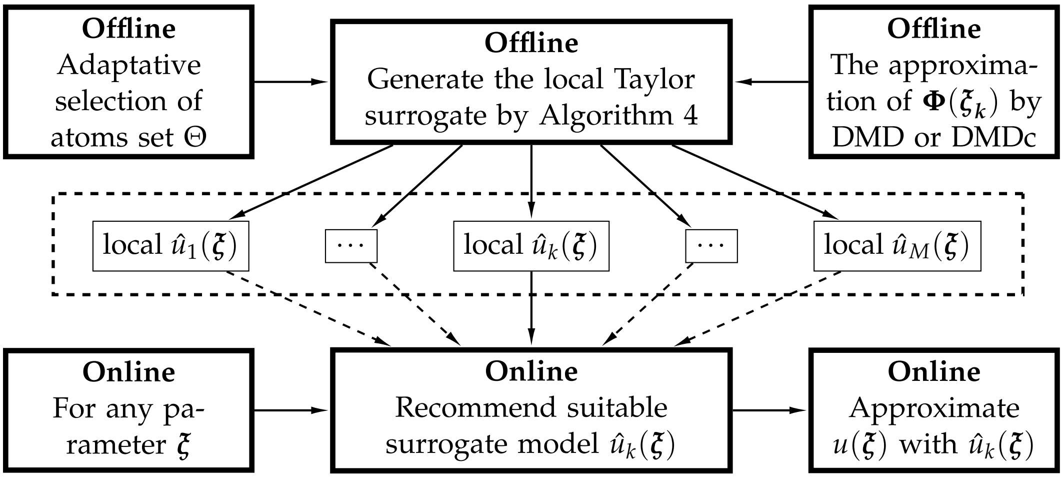

Offline
Adaptative selection of atoms set Θ
Offline
Generate the local Taylor surrogate by Algorithm 4
Offline
The approximation of $\mathbf{\Phi}(\xi_k)$ by DMD or DMDc
local $\hat{u}_1(\xi)$
$\cdots$
local $\hat{u}_k(\xi)$
$\cdots$
local $\hat{u}_M(\xi)$
Online
For any parameter ξ
Online
Recommend suitable surrogate model $\hat{u}_k(\xi)$
Online
Approximate $u(\xi)$ with $\hat{u}_k(\xi)$

Figure 1: Flowchart of our adaptive local DMD method.

surrogate model is denoted as u_{pre}. We are assessing the performance of the surrogate model by comparing the error between the solution of the local Taylor surrogate model and the reference solution. In Algorithm 4, for a fixed parameter ξ, when it comes to solving $u(\xi_k)$ or $\{\partial_i u(\xi_k)\}_{i=1}^{i=m}$, select DMD or DMDc based on the form of the equation. In Algorithms 1 and 2, the truncated rank r of reduced SVD is chosen by the hard threshold technique. In numerical examples, Σ is denoted as a singular value matrix, and we take

$$r=\left|\left\{\sigma_i\in diag(\Sigma)\,|\,\sigma_i>10^{(-10)}\right\}\right|.$$

To measure the approximation accuracy, the relative errors between the reference solution u_{ref} and the solution of the surrogate model u_{pre} are utilized and defined as follows:

For the given parameter ξ, the relative error (RE) at time t_k is defined as:

$$\text{RE}(x,t_k;\xi)=\frac{\|u_{\text{ref}}(x,t_k;\xi)-u_{\text{pre}}(x,t_k;\xi)\|_{L_2}}{\|u_{\text{ref}}(x,t_k;\xi)\|_{L_2}},$$

the mean relative L_2 error (MRE) over the whole time region is defined as:

$$\text{MRE}(\xi)=\frac{1}{N_t}\sum_{i=1}^{N_t}\text{RE}(x,t_i;\xi),$$

where N_t is the number of time steps.

The mean relative L_2 error over the parameters set at time t_k (MTE) is defined as:

$$\text{MTE}(t)=\frac{1}{N_\xi}\sum_{j=1}^{N_\xi}\text{RE}\left(x,t;\xi_j\right),$$

where N_ξ is the size of the parameters set, and the mean relative L_2 error over the whole test parameters set and time region (E) is defined as:

$$\mathrm{E}=\frac{\sum_j^{N_\xi}\sum_i^{N_t}\mathrm{RE}(x,t_i;\boldsymbol{\xi}_j)}{N_\xi N_t}.$$

6.1 Heat-equation

The heat equation is a simple linear parabolic equation describing heat diffusion over time. In this context, we use this toy example to verify the effectiveness of the proposed method. The heat equation is given by

$$u_t=\xi u_{xx},\quad (x,t)\in[0,1]\times[0,0.1],$$

with the following initial and Dirichlet boundary conditions:

$$\begin{aligned}&u(x,0)=0,\\&u(0,t)=0,\quad u(1,t)=1.\end{aligned}$$

Here, the random parameter ξ is uniformly distributed on $\mathcal{P}=[1,5]$. For the convenience of notation, we denote the solution of the heat equation as $u(x,t;\xi)$, simply as u.

In this example, we randomly select 600 samples from a uniform distribution on $\mathcal{P}$ as the background set (Ξ_{bkg}), and then select 60 uniform distribution samples as the coarse training parameter set $\Xi_{\text{train}}=\{\xi_1,\xi_2,\cdots,\xi_{N_{\text{train}}}\}$. For each sample ξ_i, the reference solution $u(x,t;\xi_i)$ is obtained through the FEM in spatial discretization and backward Euler in temporal discretization, where the spatial partition size $\Delta x=0.01$ and the time step $\Delta t=10^{-4}$. In the proposed adaptive local DMD method, the observable function $\boldsymbol{g}(\boldsymbol{u})=\boldsymbol{u}$, and snapshot data is taken from the time interval [0, 0.04]. In the DMD method, the approximate solution at any time can be obtained by using steps 6 and 7 in Algorithm 1 or 2 without solving the original equation. Therefore, the proposed adaptive local DMD method can realize the prediction in time only using snapshot data when the equation is unknown.

Fig. 2 shows the adaptive local DMD solution and the reference solution in the time interval [0, 0.1]. The first row represents the adaptive local DMD solutions at $\xi=1.5$ and $\xi=4.5$, and the second row represents the reference solutions, where $\{1.5,\ 4.5\}\in\Xi_{\text{train}}$. From this figure, we find that (1) the solution profiles of the equation show significant changes with respect to time; (2) there is no clear difference between the adaptive local DMD solution and reference solution; (3) the proposed method can achieve precise estimation within the time range of [0, 0.1], indicating that it not only demonstrates good accuracy in the snapshot data time range of [0, 0.04] but also delivers accurate predictions outside this range.

To visualize the individual errors of the testing set (Ξ_{test}) with $N_{\text{test}}=10^5$, we plot the frequency distribution of the mean relative L_2 error (MRE) over the whole time region

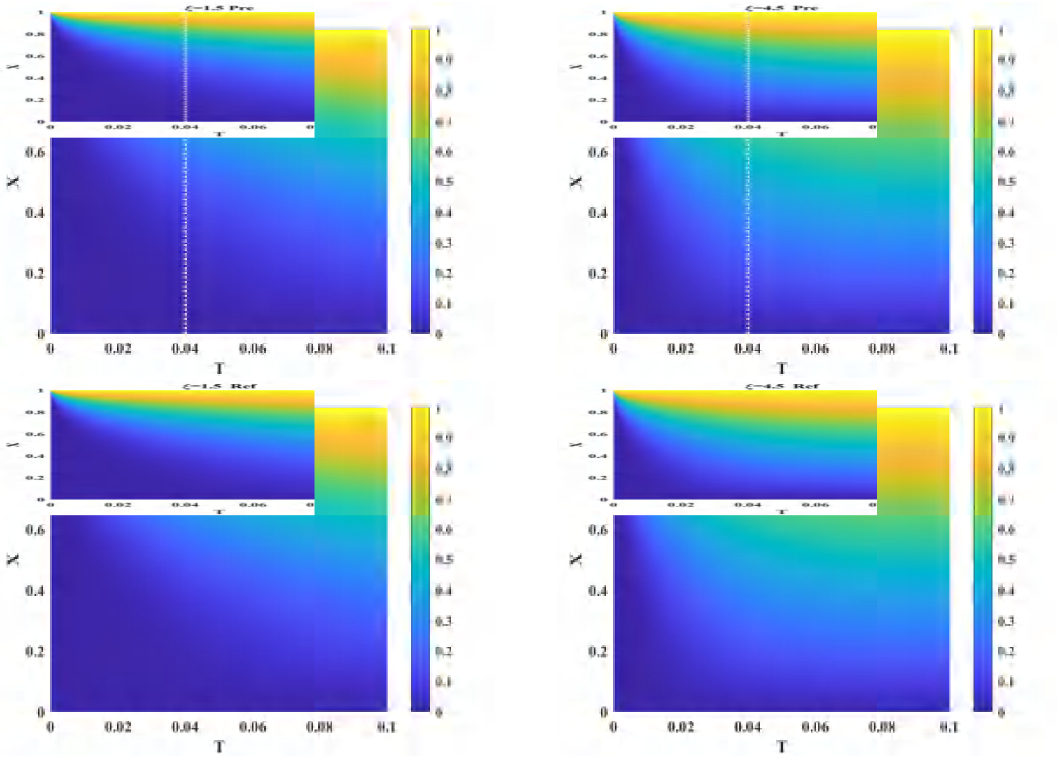


Figure 2: The solution of equation over the entire time region at $N_{\text{atom}} = 60$. Top left: the predicted solution (Pre) at $\xi = 1.5$. Top right: the predicted solution at $\xi = 4.5$ Bottom left: the reference solution (Ref) at $\xi = 1.5$. Bottom right: the reference solution at $\xi = 4.5$.

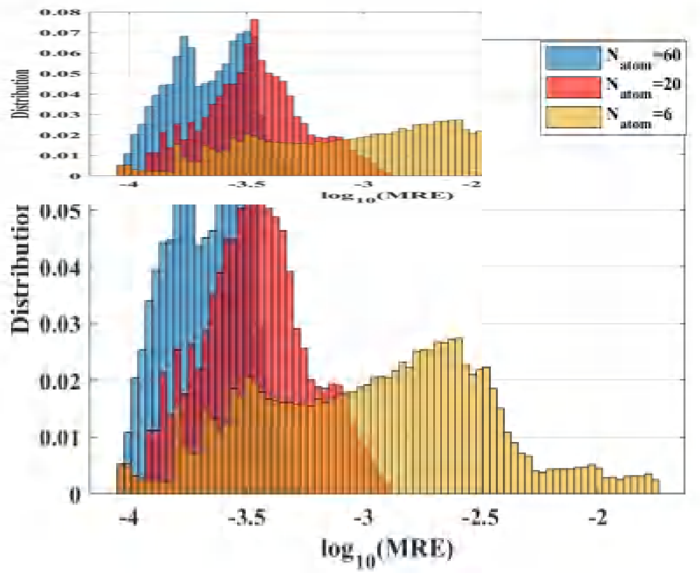


Figure 3: Distribution of the relative error (MRE) under different size of atoms set.

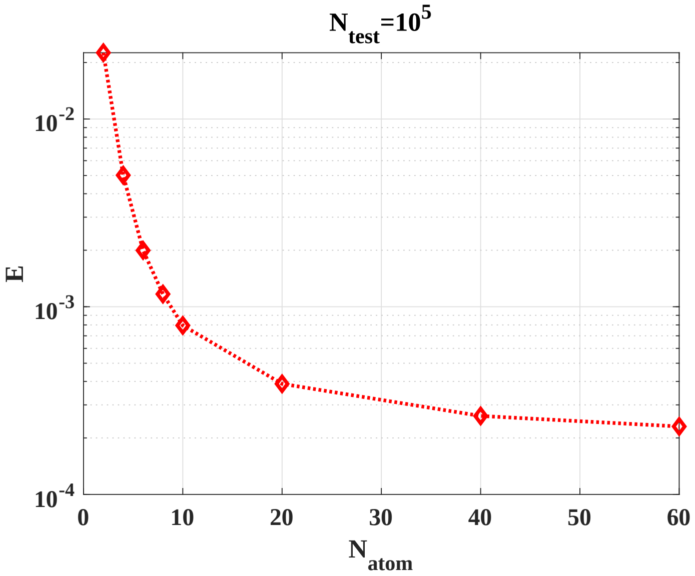


Figure 4: The relative error (E) with different size of atoms set.

on Ξ_{test} for different sizes of atoms set, with $N_{\text{atom}} = 60,20$ and 6. From this figure, we can see that (1) as the number of atoms increases, the mean value of MRE decreases, and the approximations become more accurate. That is to say, the more categories offline, the more accurate of online approximation; (2) the frequency histogram of MRE becomes more compact as the number of offline atoms increases; (3) MRE represents the mean relative error within the whole time region, when $N_{\text{atom}} = 40$, the value of MRE is relatively small, it indicates that the relative error at each moment in the whole time region is small. This reflects the stability of the proposed method.

In Fig. 4, we present the mean relative error over the whole test parameters set and time region (E) with different sizes of atoms. As we can see from this figure, the relative error E decreases substantially at the beginning as the size of atoms increases. When the size of atoms exceeds a certain number, the error E decreases very slowly. So we can get good accuracy by using only a few atoms. In this example, $N_{\text{atom}} = 20$ can achieve reasonable accuracy.

Fig. 5 presents the mean relative error (MTE) of the entire testing parameter set during the temporal range of $[0, 0.1]$. We find that (1) the MTE becomes smaller when the size of atoms increased; (2) the MTE of $N_{\text{atom}} = 60$ remains consistently below that of $N_{\text{atom}} = 6$; (3) the value of MTE is small in the snapshot data time interval $[0, 0.04]$. When $t > 0.04$, outside the snapshot data time interval, the value of MTE increased relative to $t < 0.04$, but it was below 10^{-3}. This implies the potential of the proposed method in predicting the model solutions outside of the snapshot data time interval.

During the testing phase, the adaptive local DMD method relies on two distinct training parameters sets: a coarse training set (Ξ_{coarse}) and a background training set (Ξ_{bkg}).

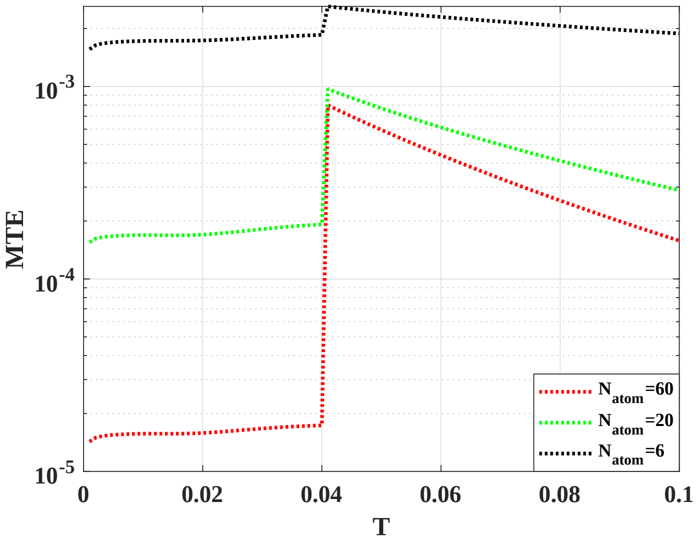


Figure 5: The mean relative L_2 error over the whole test parameters set (MTE) in the time interval $[0,\ 0.1]$.

Table 1: The utilization rate η and the comparison of errors between the there methods.

N_{atom}	2	5	10	25	30
η	1.5%	10.8%	26.7%	65.2%	65.8%
E	2.23×10^{-2}	3.60×10^{-3}	7.89×10^{-4}	2.85×10^{-4}	2.77×10^{-4}
E_{bkg}	2.23×10^{-2}	3.60×10^{-3}	9.46×10^{-4}	2.85×10^{-4}	2.78×10^{-4}
E_{coarse}	2.05×10^{-2}	3.77×10^{-2}	1.21×10^{-3}	9.52×10^{-4}	9.49×10^{-4}

Table 1 lists the errors of the considered methods. i.e. the adaptive local DMD method (E), the Classical Greedy method on Ξ_{bkg} (E_{bkg}) and the CG method on Ξ_{coarse} (E_{coarse}). We also evaluated the utilization of the adaptive local DMD method (η) in Table 1. We indicate that (1) the η exhibits an upward trend with the increase of N_{atom} but is always less than 1; (2) the error of the adaptive local DMD algorithm method is not significantly different from that of the CG method on Ξ_{bkg}, and the adaptive local DMD method requires fewer samples; (3) under the same N_{atom} conditions, both the adaptive local DMD method and the CG method on Ξ_{bkg} exhibit superior performance compared to the CG method on Ξ_{coarse}. This implies that the adaptive local DMD method is a promising alternative to the CG method and requires lower computational cost.

Fig. 6 shows the mean relative errors over the whole parameters set and time region (E) with three methods during the testing phase. By the figure, We find that (1) the adaptive local DMD method outperforms the CG method on Ξ_{coarse} when considering a small atom set Ξ_{coarse}; (2) the adaptive local DMD method requires fewer computational resources than the CG method on Ξ_{bkg} while maintaining similar accuracy. The results

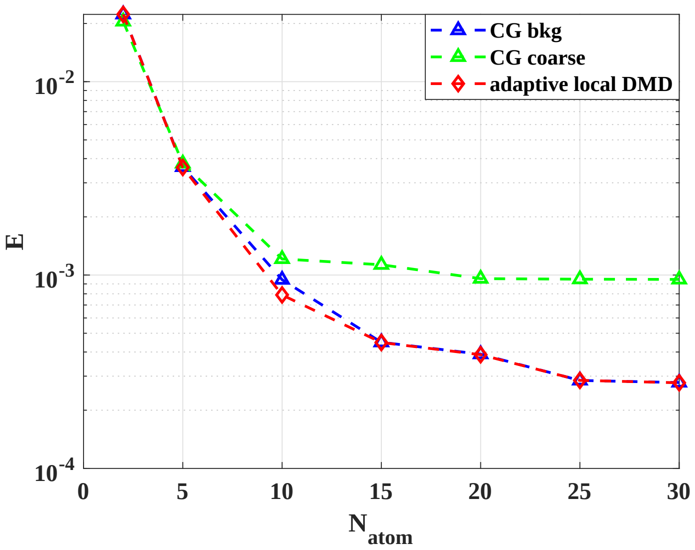


Figure 6: The mean relative error (E) comparison between the CG method and the adaptive local DMD method.

presented in Fig. 6 support using the adaptive local DMD algorithm for this particular problem.

6.2 Burgers-equation

The Koopman operator can transform a nonlinear dynamical system into a linear system, so the proposed method provides remarkable convenience for a parametric nonlinear dynamical system. In this section, we will show the performance of the proposed method on the Burgers equation, a popular nonlinear hyperbolic PDE. We consider the following Burgers equation:

$$u_t+uu_x=\frac{v}{50}u_{xx}, \quad (x,t)\in[0,1]\times[0,0.5], \tag{6.1}$$

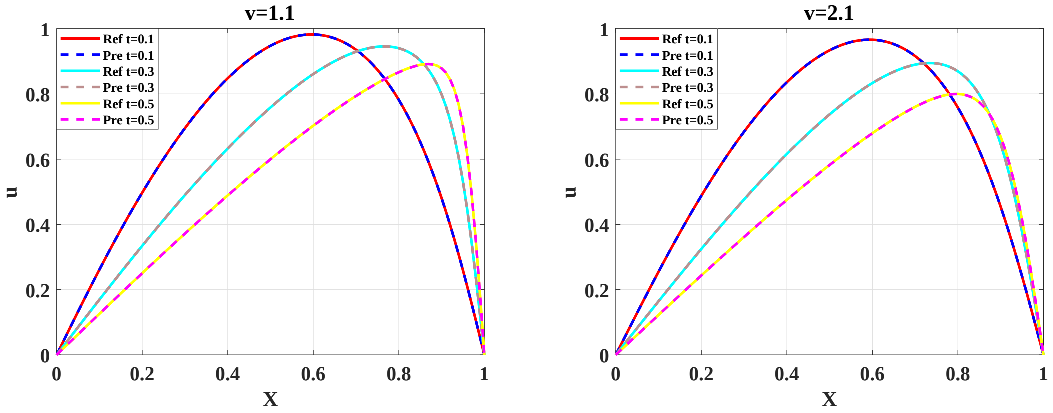
with the boundary and initial conditions defined as:

$$\begin{aligned} &u(0,t;v)=u(1,t;v)=0,\\ &u(x,0;v)=4x(1-x). \end{aligned}$$

Here, the viscous term v is uniform distribution on $\mathcal{P}=[1,3]$. The reference solutions are solved using FEM in space with $\Delta x=0.01$ and a backward Euler scheme in time with $\Delta t=0.0005$.

6.2.1 Original equation

To experiment, we generated a background set of 400 samples uniformly distributed in the parameter space $\mathcal{P}$, and an initial training set is formed by selecting 40 samples from



(a) Solution $v = 1.1$ (b) Solution $v = 2.2$

Figure 7: The solution of the equation over the entire time region at $N_{\text{atom}} = 40$. Left: solution at $v = 1.1$. Right: solution at $v = 2.2$.

$\mathcal{P}$ with uniform distribution. We randomly selected a testing set of 10^5 samples from $\mathcal{P}$ to evaluate the model's performance. In the proposed adaptive local DMD method, the observable function $\boldsymbol{g}(\boldsymbol{u}) = [\boldsymbol{u};\ \boldsymbol{u}^2]$ and snapshot data is taken from the time interval $[0,\ 0.3]$. It should be noted that the choice of observable function and snapshot data is crucial for the success of Algorithms 1 and 2.

Fig. 7 displays the reference and predicted solutions corresponding to the parameters $v = 1.1$ and $v = 2.1$, respectively, for a fixed atoms set $N_{\text{atom}} = 40$. From the figure, it can be seen that (1) at $t = 0.1, 0.3, 0.5$, the proposed adaptive local DMD method achieves pretty good agreement with the reference solution; (2) $t = 0.5$ outside the snapshot data time interval, indicating that the adaptive local DMD method is effective for future prediction; (3) even if the parameter v is changed, the proposed adaptive local DMD method can still achieve good computational accuracy.

We illustrates the frequency distribution of MRE for different sizes of atoms set with $N_{\text{atom}} = 40, 20$, and 6, over the entire testing set of $N_{\text{test}} = 10^5$ in Fig. 8. From this figure, we can observe that (1) the value of MRE decreases when the number of atoms increases. This indicates that the more atoms, the more accurate the computation; (2) as the number of atoms increases, the standard deviation of MRE decreases. This can be seen from figure that the histogram of the MRE becomes more compact as the number of atoms increases.

In Fig. 9, we depict the variation of error (E) in all samples with the change in the size of atoms set on the testing set. By the figure, we can see that as the size of the atoms set increases, the approximation becomes more accurate. However, once the number of atoms reaches a certain number, the degree of error reduction slows down significantly. In this example, $N_{\text{atom}} = 10$ can achieve desirable accuracy.

Fig. 10 shows the temporal evolution of the mean relative error (MTE) of all samples in the testing set for different atoms set sizes $N_{\text{atom}} = 6, 20, 40$ over the time interval $[0, 0.5]$.

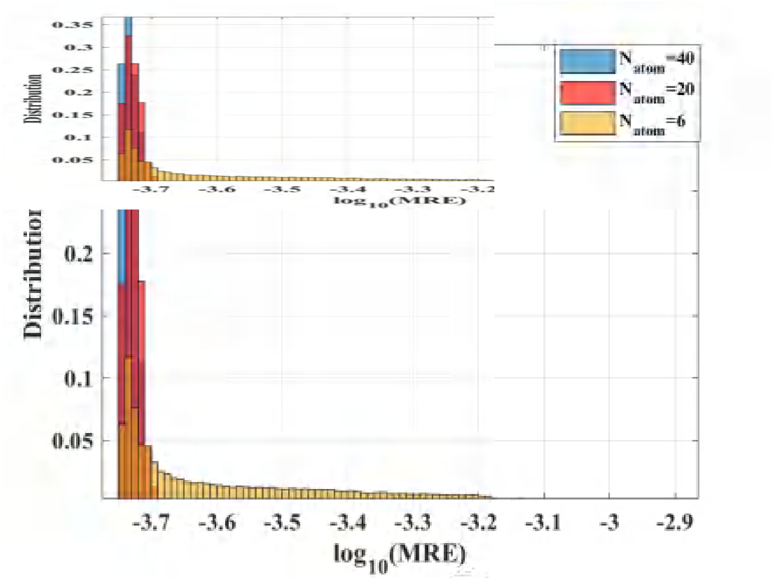


Figure 8: Distribution of the relative error (MRE) under different size of atoms set.

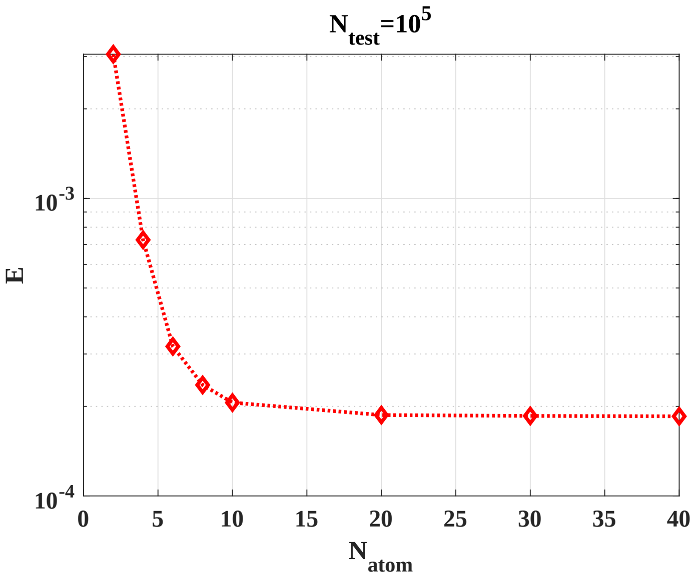


Figure 9: The relative error (E) with different sizes of atoms set.

We find that (1) the MTE becomes smaller when the number of atoms increase; (2) the MTE is very small within the snapshot data time interval $[0, 0.3]$. When $0.3 < t < 0.5$, outside the snapshot data time interval, although the MTE has increased, it is still below 10^{-3}. This indicates the proposed method can also achieve good prediction results outside the snapshot data time interval.

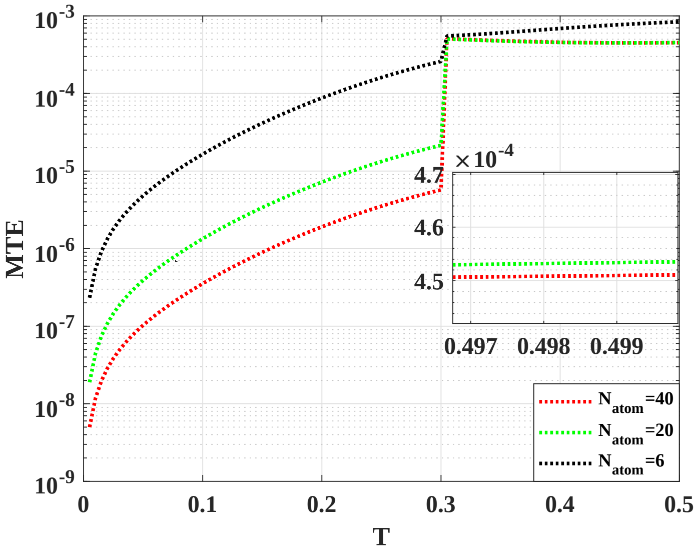


Figure 10: The mean relative L_2 error over the time interval $[0,0.5]$ (MTE) in the whole testing set.

Table 2: The utilization rate η of the background set Ξ_{bkg}.

N_{atom}	4	6	10	20	30	40
η	12%	21.5%	40.5%	78.75%	87.75%	96.25%

Table 2 presents the utilization rate η for different sizes of atom sets with a fixed background set size $N_{\text{bkg}}=400$. The result shows that (1) the utilization rate becomes higher when the number of atoms increases; (2) the utilization rate $\eta<1$, which indicates that the background set has not been fully utilized. During the training process, the size of the training set used can be adjusted as needed to improve the efficiency of offline computation.

6.2.2 Modifying the viscosity coefficient range

To demonstrate the performance of adaptive local DMD method in solving the Burgers equation with very small viscosity coefficients, we plot the errors (E) with the number of atoms (N_{atom}) for the viscosity coefficient being $v/50$, $v/1000$, and $v/10000$ in Fig. 11. which shows that (1) the errors for each viscosity coefficient becomes smaller when N_{atom} increase. (2) When N_{atom} reaches a certain threshold, the rate of error reduction diminishes, eventually stabilizing. (3) In the case of $v/1000$, the accuracy always outperforms the others cases, and the accuracy is more sensitive to the N_{atom}.

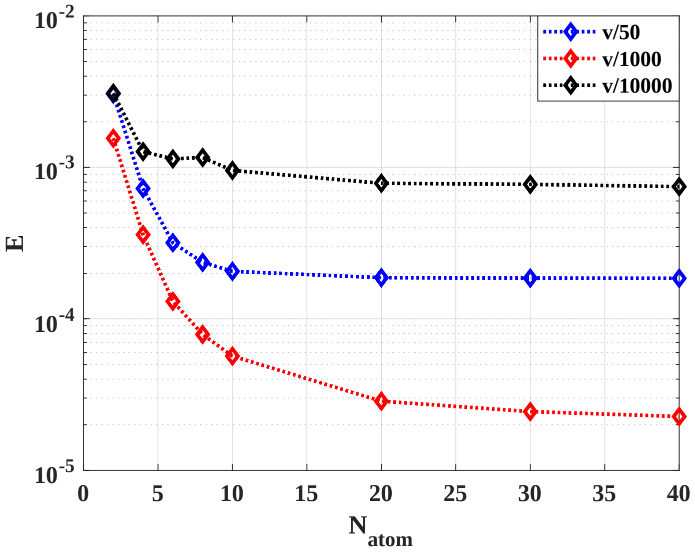


Figure 11: Burgers: The error corresponding to a smaller viscosity coefficient.

6.3 Allen-Cahn equation

In this section, we will consider an Allen-Cahn equation to test the proposed method's effectiveness for dealing with high-dimensional parameter problems. The Allen-Cahn equation is a special type of reaction-diffusion equation commonly used to describe phase separation phenomena in materials. It is described as follows:

$$u_t = v u_{xx} - \mu \left(u - u^3\right), \quad (x,t) \in [0,1] \times [0,1], \tag{6.2}$$

where time-independent diffusion coefficient v and reaction coefficient μ are uniformly distributed on $\mathcal{P}_1 = \mathcal{P}_2 \sim U[0.06,\ 1]$. The Dirichlet boundary condition and initial conditions are defined as follows:

$$\begin{aligned} &u(0,t) = 0, u(1,t) = 0, \\ &u(x,0) = 0.5 + 0.5\sin(\pi x). \end{aligned}$$

The reference solution is obtained using the FEM in space $\Delta x = 0.01$ and a backward Euler scheme in time with $\Delta t = 10^{-4}$.

6.3.1 Original equation

In experiment, the predicted solution is obtained using the adaptive local DMD method, where the observable function is $\boldsymbol{g} = \left[\boldsymbol{u};\ \boldsymbol{u}^3\right]$, and snapshots are taken in the time interval $t \in [0,0.5]$. Given the specified parameters, we compare the reference solution to the predicted solution with $N_{\text{atom}} = 625$, The results are presented in Fig. 12. Fig. 12 shows the predicted solutions using the adaptive local DMD method are compared to

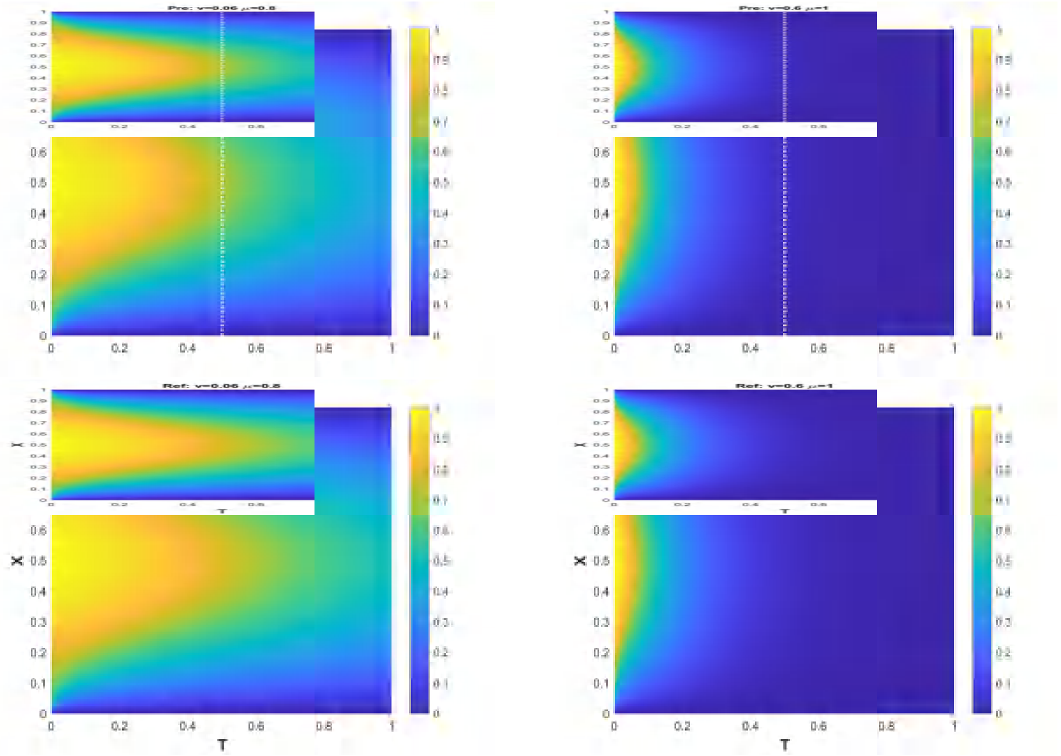


Figure 12: The solution of equation over the entire time region $N_{\text{atom}}=625$. Top left: the predicted solution (Pre) at $v=0.06$ $\mu=0.8$. Top right: the predicted solution at $v=0.6$ $\mu=1$ Bottom left: the reference solution (Ref) at $v=0.06$ $\mu=0.8$. Bottom right: the reference solution at $v=0.6$ $\mu=1$.

the reference solutions obtained using the FEM at two different parameter sets: $(v,\ \mu)=\{(0.06,\ 0.8),\ (0.6,\ 1)\}$. From Fig. 12, we can observe the following: (1) The solution changes significantly for different parameters; (2) the time layers of the equation's solutions change obviously over time; (3) the difference between the reference and predicted solutions is minimal both inside and outside the snapshot time. This implies that the solution is highly sensitive to the parameter values and varies significantly with time.

We have plotted the location of the atoms set with $N_{\text{atom}}=100$ in Fig. 13. As depicted in figure, it is evident that the adaptive local DMD method on A-C equation exhibits increased atom density in regions with larger parameter values.

Fig. 14 shows the distribution of mean relative errors (MRE) on all test samples $N_{\text{test}}=10^5$ over the whole time region, with different atom sets $N_{\text{atom}}=625$,225 and 25. We find that (1) the mean value of the MRE for the samples decreases as N_{atom} increase; (2) the maximum frequency corresponding to MRE also becomes smaller when N_{atom} increase. This indicates that a larger size of atoms size will result in better approximation performance for the adaptive local DMD method.

Fig. 15 presents the mean relative error for all parameters (E) at different sizes of N_{atom}. From Fig. 15, we observe that: (1) E decreases as the number of atoms increases;

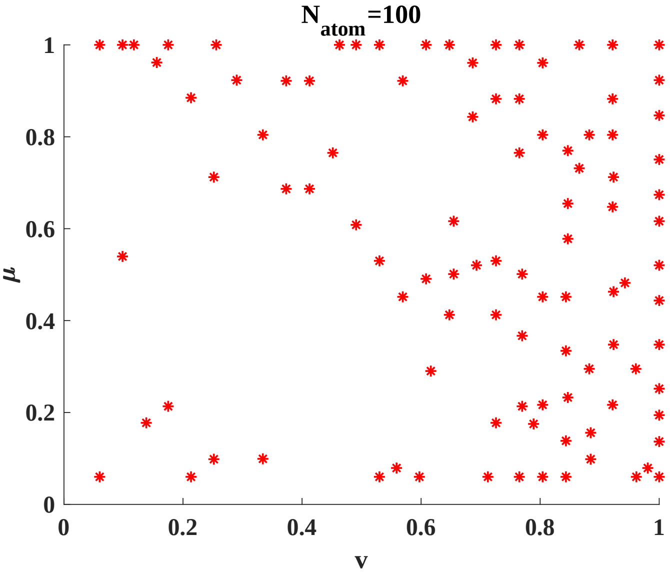


Figure 13: The location of the atoms set.

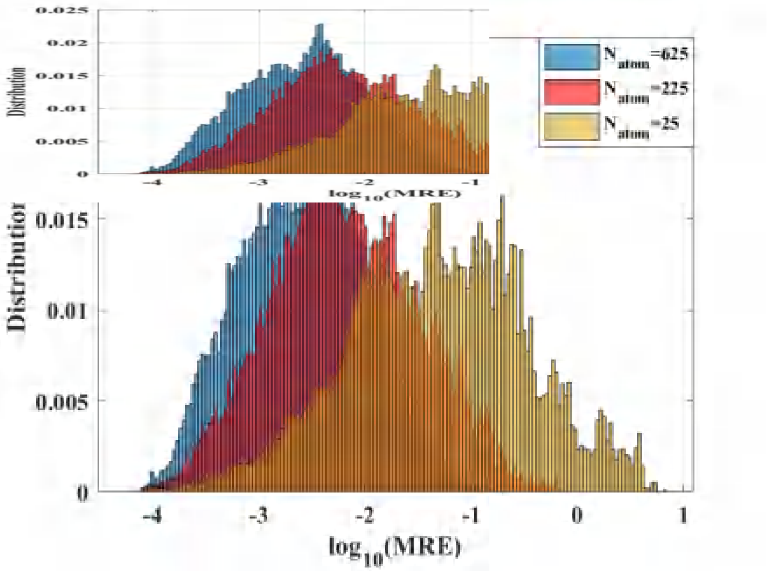


Figure 14: Distribution of the relative error (MRE) under different size of atoms set.

(2) when N_{atom} is at its maximum 700, the downward trend of the curve does not become smoother. This implies that adding more atomic points can reduce the error even further.

To visualize the error in time, we plot the mean relative error over all test parameters (MTE) of the atoms set size being $N_{\text{atom}} = 625,225$ and 25 in Fig. 16. These results indicate the performance of the adaptive local DMD method in predicting over the entire time

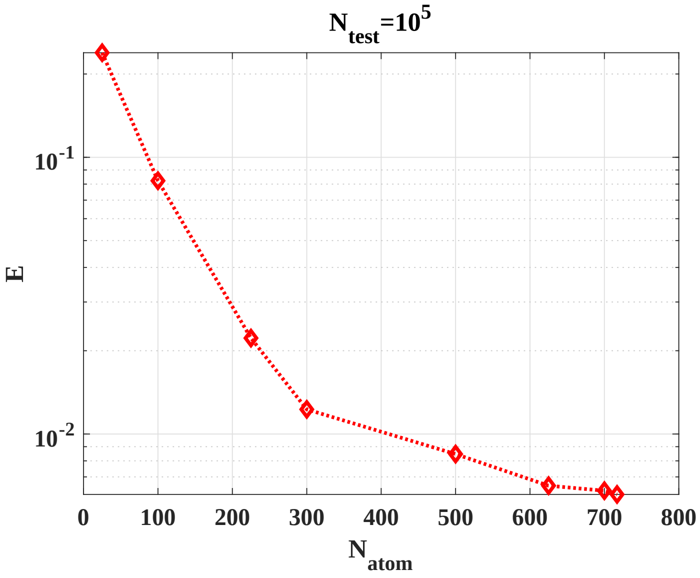


Figure 15: The relative error (E) with different size of atoms set.

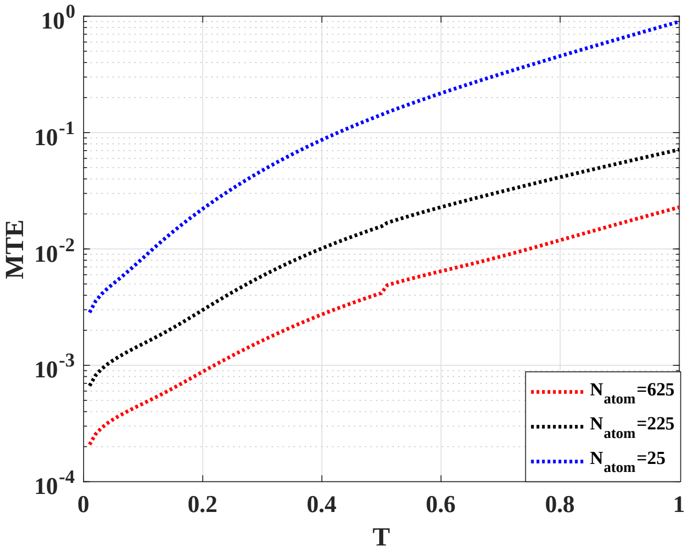


Figure 16: The mean relative L_2 error over the whole test parameters set (MTE) in the time interval $[0,0.5]$.

domain, which shows that (1) the MTE becomes smaller at any given time when N_{atom} increases; (2) with a fixed size of atoms set, the MTE within the snapshot data time interval $t\in[0,0.5]$ is much smaller than that outside the time interval $t\in[0.5,1]$. These findings suggest that the proposed method is reliable in high-dimensional parameter spaces.

In Table 3, we present the utilization (η) of Ξ_{bkg} corresponding to different atom set

Table 3: The utilization rate η of the background set Ξ_{bkg}.

N_{atom}	25	100	225	625
η	18.24%	53.64%	76.08%	91.12%

sizes: $N_{\text{atom}} = 25$, 100, 225 625 and $N_{\text{bkg}} = 900$. Through comparison, we find that (1) η becomes higher when N_{atom} increases; (2) even when the error is relatively small, the utilization rate remains below 1, i.e., $\eta < 1$, indicating that Ξ_{bkg} has not been fully utilized. From this, it can be inferred that the model can dynamically adjust to the desired training set size in high-dimensional parameter problems to improve offline computing efficiency.

6.3.2 Modifying the initial conditions

Considering the Allen-Cahn equation with rough initial conditions, we have initialized it with a normal distribution reflect on $[0, 1]$. The solution of parameter at $v=0.06$, $\mu=1$ are presented in Fig. 17. We randomly choose 10000 samples and compute the mean relative errors with time (MTE). Fig. 18 shows the MTE for the solutions versus different number of the atoms, with $N_{\text{atom}} = 625$,200 and 50. The temporal snapshots in $t \in [0, 0.7]$ were taken as the training set of DMD or DMDc.

By the Fig. 18, we find that (1) the MTE becomes smaller when the number of atoms set N_{atom} increases at any time; (2) with the fixed size of atoms sets, the MTE still below than 1. This implies that under the specified rough conditions, the error decreases with the increase in N_{atom}. To attain a reduced error, a greater quantity of atoms is required.

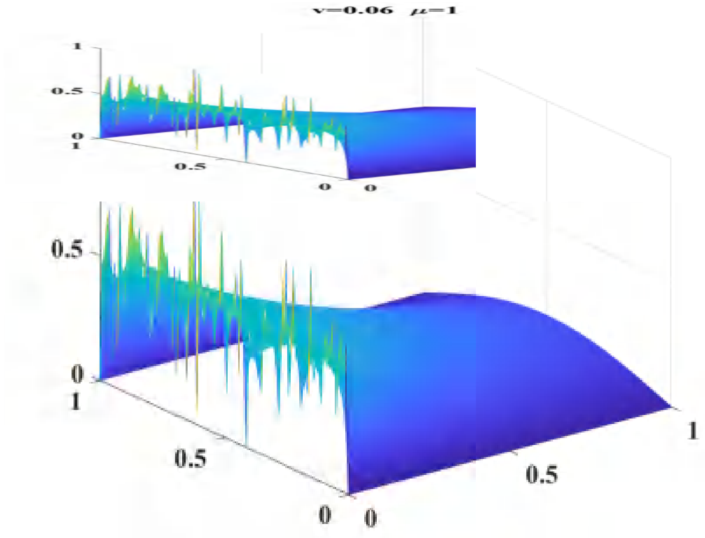


Figure 17: The solution of A-C equation.

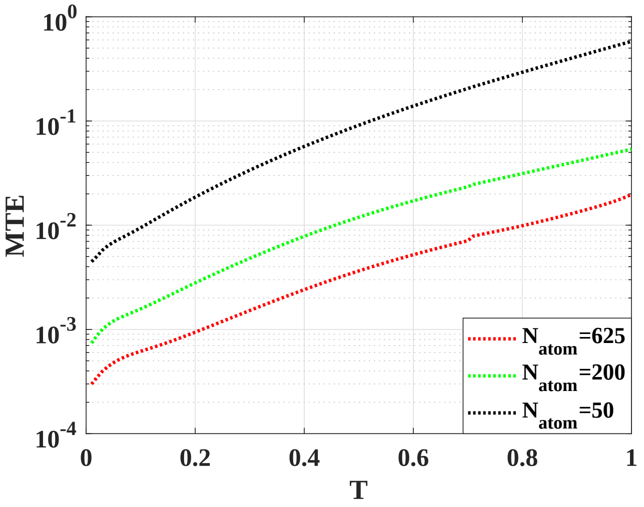


Figure 18: The error (E) of A-C equation.

6.3.3 Comparison with traditional methods

To compare the performance of traditional model reduction method (e.g. POD) and adaptive local DMD method, we have selected a specific parametric Allen-Cahn equation, described as follows:

$$u_t = v u_{xx} - \mu\left(u - u^3\right), \quad (x,t) \in [0,\, 1] \times [0,\, 0.5], \tag{6.3}$$

the Dirichlet boundary condition and initial conditions are defined as follows:

$$\begin{aligned} &u(0,t)=0,\ u(1,t)=0,\\ &u(x,0)=e^{\gamma x}. \end{aligned}$$

where time-independent diffusion coefficient v, reaction coefficient μ, and initial conditions coefficient γ are uniformly distributed on $\mathcal{P}_1 \sim U[0.0001,\, 1]$, $\mathcal{P}_2 \sim U[0.06,\, 1]$, $\mathcal{P}_3 \sim U[0,\, 1]$. The truncated rank r of POD method is chosen by the hard threshold technique. σ_i is denoted as a singular value.

$$r = \underset{j}{\operatorname{argmin}} \frac{\sum_{i=1}^{i=1} \sigma_i}{\sum_{i=1}^{i=n} \sigma_i} > 99.999\%. \tag{6.4}$$

We randomly choose 1000 samples and computer the mean relative error with time (MTE) and parameters (RE). Fig. 19 shows the MTE results for adaptive local DMD method and POD method. To further consider the accuracy of the adaptive local DMD method, we respectively compute the RE over the all test parameters for $t=0.2$ in Fig. 20.

From the figures, we find that (1) the adaptive local DMD method outperform POD in the entire time region; (2) the MTE increases as time increases; (3) the accuracy of the

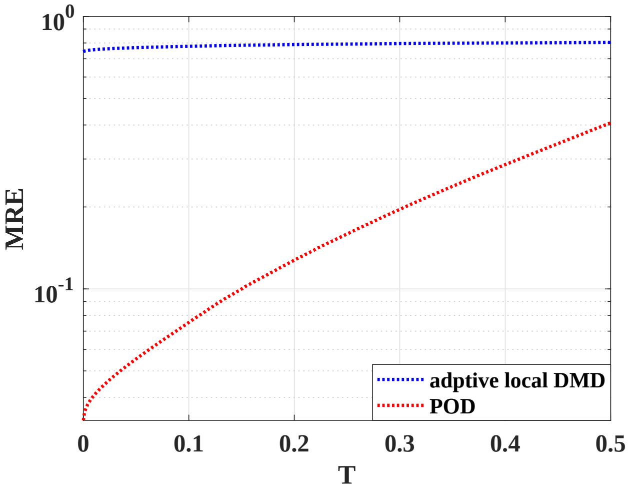


Figure 19: The error (MTE) of two methods.

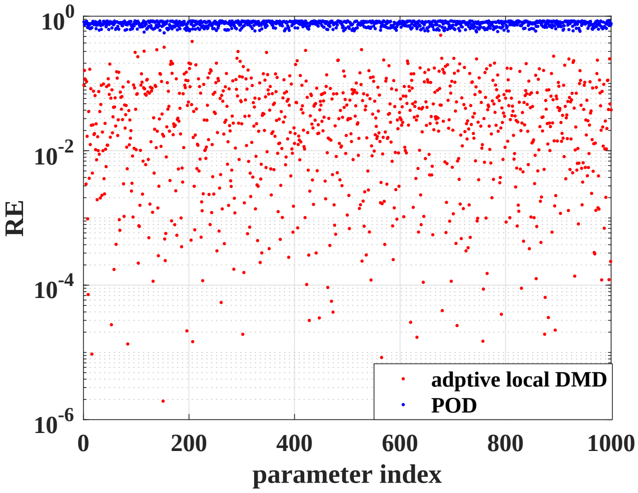


Figure 20: The error (RE) of testing set on $t=0.2$.

adaptive local DMD method is more sensitive to the parameter variation; (4) the adaptive local DMD accuracy still outperforms of POD for all test parameters when $t=0.2$.

7 Conclusion

In this work, we have developed an adaptive method based on local dynamic mode decomposition for parametric dynamical systems. The proposed method is devoted to achieving an efficient and reliable approximation of input-output relationship. The entire computational process follows an offline-online decomposition. In the offline phase, we propose an improved greedy algorithm to generate an atoms set Θ based on a series of relatively small training sets. This approach aims to mitigate the influence of large-scale training sets. Subsequently, for each atom in Θ, we construct the corresponding local surrogate model by combining Taylor linear expansion and DMD (or DMDc). Therefore, the proposed method retains the advantages of the DMD approach in deterministic problems. For instance, it allows for predicting the system's state at any given time without solving the original dynamic system. Moreover, due to the gradient information generated from the Taylor expansion, our method is capable of providing optimal approximations in nearly any location within the parameter domain, under certain smoothness assumptions. In the online stage, the surrogate model for any given parameter is determined by the local surrogate model corresponding to the atom that is the closest point to the given parameter according to the Euclidean distance criterion. The output is then calculated by the corresponding local model. Due to the local nature of our proposed method, the online computational cost does not increase as more atoms are added in the adaptive process. To demonstrate the performance of our proposed method, we applied it to various nonlinear parametric dynamical systems and obtained efficient and accurate surrogate models. This also demonstrates that our adaptive local DMD method gives rise to a good approximation when predicting the state outside the training time region.

Despite the significant reduction in online computations achieved by our proposed method, it requires considerable effort during the offline computation phase. In the future, we intend to explore new approaches to substantially alleviate the burden of offline computations.

Acknowledgments

The authors are grateful to the anonymous referees and the handling editor for their helpful suggestions. This work was support by National Key R&D Program of China (No. 2021YFA1001300), National Natural Science Foundation of China (Nos. 12288101, 12271150, 12101216), the Hunan Provincial Natural Science Foundation of China (No. 2022JJ40030), the Jiangsu Provincial Natural Science Foundation of China (No. BK20230346), and Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (No. NY222063).

References

[1] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys., 2 (2007), pp. 293–309.

[2] D. Xiu and J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.

[3] A. Nouy and O. P. Le Maître, Generalized spectral decomposition for stochastic nonlinear problems, J. Comput. Phys., 228 (2009), pp. 202–235.

[4] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems, Arch. Comput. Methods Eng., 17 (2010), pp. 403–434.

[5] Q. Li and L. Jiang, A novel variable-separation method based on sparse and low rank representation for stochastic partial differential equations, SIAM J. Sci. Comput., 39 (2017), pp. A2879–A2910.

[6] L. Jiang and Q. Li, Model reduction method using variable-separation for stochastic saddle point problems, J. Comput. Phys., 354 (2018), pp. 43–66.

[7] Q. Li and P. Zhang, A variable-separation method for nonlinear partial differential equations with random inputs, SIAM J. Sci. Comput., 42 (2020), pp. A723–A750.

[8] C. Canuto, T. Tonn and K. Urban, A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear pdes, SIAM J. Numer. Anal., 47 (2009), pp. 2001–2022.

[9] Y. Chen, J. S. Hesthaven, Y. Maday and J. Rodríguez, Certified reduced basis methods and output bounds for the harmonic Maxwell's equations, SIAM J. Sci. Comput., 32 (2010), pp. 970–996.

[10] P. Chen, A. Quarteroni and G. Rozza, A weighted reduced basis method for elliptic partial differential equations with random input data, SIAM J. Numer. Anal., 51 (2013), pp. 3163–3185.

[11] A. Quarteroni and G. Rozza, Reduced Order Methods for Modeling and Computational Reduction, Springer Publishing Company, 2014.

[12] L. Jiang and Q. Li, Model's sparse representation based on reduced mixed GMsFE basis methods, J. Comput. Phys., 338 (2017), pp. 285–312.

[13] A. Sartori, A. Cammi, L. Luzzi and G. Rozza, Reduced basis approaches in time-dependent non-coercive settings for modelling the movement of nuclear reactor control rods, Commun. Comput. Phys., 20(2016), pp. 23–59.

[14] J. Yu, D. Ray and J. S. Hesthaven, Fourier collocation and reduced basis methods for fast modeling of compressible flows, Commun. Comput. Phys., 32(2022), pp. 595–637.

[15] S. K. Star, G. Stabile, F. Belloni, G. Rozza and J. Degroote, A novel iterative penalty method to enforce boundary conditions in finite volume POD-Galerkin reduced order models for fluid dynamics problems, Commun. Comput. Phys., 30(2021), pp. 34–66.

[16] S. Chakraborty and N. Zabaras, Efficient data-driven reduced-order models for high-dimensional multiscale dynamical systems, Commun. Comput. Phys., 230(2018), pp. 70–88.

[17] M. Cheng, T. Y. Hou and Z. Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations I: Derivation and algorithms, J. Comput. Phys., 242 (2013), pp. 843–868.

[18] M. Cheng, T. Y. Hou and Z. Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations II: Adaptivity and generalizations, J. Comput. Phys., 242 (2013), pp. 753–776.

[19] E. Musharbash, F. Nobile and T. Zhou, Error analysis of the dynamically orthogonal approximation of time dependent random PDEs, SIAM J. Sci. Comput., 37 (2015), pp. A776–A810.

[20] Y. Kazashi, F. Nobile and E. Vidličková, Stability properties of a projector-splitting scheme for dynamical low rank approximation of random parabolic equations, Numer. Math., 149 (2021), pp. 973–1024.

[21] Y. Zhao, Z. Mao, L. Guo, Y. Tang and G. E. Karniadakis, A spectral method for stochastic fractional PDEs using dynamically-orthogonal/bi-orthogonal decomposition, J. Comput. Phys., 461(2022), 111213.

[22] K. S. Mohamed, Machine Learning for Model Order Reduction, Springer International Pub-

lishing, 2018.

[23] Y. Kim, Y. Choi, D. Widemann and T. Zohdi, A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, J. Comput. Phys., 451 (2022), 110841.

[24] C. Cui, K. Jiang and S. Shu, Solving time-dependent parametric PDEs by multiclass classification-based reduced order model, CSIAM Trans. Appl. Math., 4(2023), pp. 13–40.

[25] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, Proc. Natl. Acad. Sci. U.S.A., 17(1931), pp. 315–318.

[26] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech., 656(2010), pp. 5–28.

[27] J. H. Tu, C. W. Rowley, D. M. Luchtenberg, S. L. Brunton and J. N. Kutz, On dynamic mode decomposition: Theory and applications, J. Comput. Dyn., 1(2014), pp. 391–421.

[28] M. O. Williams, I. G. Kevrekidis and C. W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlinear Sci., 25 (2015), pp. 1307–1346.

[29] M. Korda and I. Mezić, On convergence of extended dynamic mode decomposition to the Koopman operator, J. Nonlinear Sci., 28 (2018), pp. 687–710.

[30] M. O. Williams, C. W. Rowley and I. G. Kevrekidis, A kernel-based method for data-driven koopman spectral analysis, J. Comput. Dyn., 2 (2015), pp. 247–265.

[31] J. L. Proctor, S. L. Brunton and J. N. Kutz, Generalizing Koopman theory to allow for inputs and control, SIAM J. Appl. Dyn. Syst., 17(2018), pp. 909–930.

[32] M. S. Hemati, C. W. Rowley, E. A. Deem and L. N. Cattafesta, De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets, Theor. Comput. Fulid Dyn., 31 (2017), pp. 349–368.

[33] N. Takeishi, Y. Kawahara and T. Yairi, Subspace dynamic mode decomposition for stochastic Koopman analysis, Phys. Rev. E (3), 96 (2017), 033310.

[34] Z. Gao, Y. Lin, X. Sun and X. Zeng, A reduced order method for nonlinear parameterized partial differential equations using dynamic mode decomposition coupled with k-nearest-neighbors regression, J. Comput. Phys., 452 (2022), 110907.

[35] Q. A. Huhn, M. E. Tano, J. C. Ragusa and Y. Choi, Parametric dynamic mode decomposition for reduced order modeling, J. Comput. Phys., 475 (2023), 111852.

[36] Z. Zou, D. Kouri and W. Aquino, An adaptive local reduced basis method for solving PDEs with uncertain inputs and evaluating risk, Comput. Methods Appl. Mech. Engrg., 345 (2019), pp. 302–322.

[37] A. Nouy, Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations, Arch. Comput. Methods Eng., 16 (2009), pp. 251–285.

[38] J. L. Proctor, S. L. Brunton and J. N. Kutz, Dynamic mode decomposition with control, SIAM J. Appl. Dyn. Syst., 15(2016), pp. 142–161.

[39] D. Wirtz, D. C. Sorensen and B. Haasdonk, A-posteriori error estimation for DEIM educed nonlinear dynamical systems, SIAM J. Sci. Comput., 36 (2014), pp. A311–A338.

[40] M. Billaud-Friess and A. Nouy, Dynamical model reduction method for solving parameter-dependent dynamical systems, SIAM J. Sci. Comput., 39 (2017), pp. A1766–A1792.

[41] J. Jiang and Y. Chen, Adaptive greedy algorithms based on parameter-domain decomposition and reconstruction for the reduced basis method, Internat. J. Numer. Methods Engrg., 121 (2020), pp. 5426–5445.