Thai Journal of **Math**ematics
Special Issue (2022): Annual Meeting in Mathematics 2021
Pages 38 – 52
http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Dictionary-based Model Order Reduction via POD-DEIM with Support Vector Machine for the Parametrized Burgers' Equation

Norapon Sukuntee and Saifon Chaturantabut*

Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani 12120, Thailand
e-mail : nsukuntee@gmail.com (N. Sukuntee); saifon@mathstat.sci.tu.ac.th (S. Chaturantabut)

Abstract In this article, we present a dictionary-based model order reduction approach applied to the parametrized Burgers' equation. This approach uses the support vector machine (SVM) to build a classifier, which efficiently selects the most suitable local reduced-order basis in a dictionary for a given parameter value. The dictionary of local reduced-order models is constructed by clustering the solution manifold enabling the identification of reduced-order bases that are obtained from proper orthogonal decomposition (POD). After a POD basis is chosen by the classifier, it is used in the construction of a reduced-order model corresponding to the parameter value of interest. For the reduced-order modeling task, the Galerkin projection is applied together with the selected POD basis to transform the full-order model to a low-dimensional system. In addition, the discrete empirical interpolation method (DEIM) is applied to further reduce computational complexity of the nonlinear term in Burger's Equation. The numerical experiments for the POD-DEIM reduced-order model assisted by SVM are shown to be efficient in reducing both dimension and simulation time while maintaining accuracy for the parametrized Burgers' equation. Our proposed method is also shown to be more accurate than the standard global basis approach when the same reduced dimension is used.

MSC: 65F30; 3540; 34A45
Keywords: model order reduction; proper orthogonal decomposition; discrete empirical interpolation method; deep neural network

1. Introduction

Numerical simulations have become an essential tool in many applications. However, such simulations are often time-consuming due to the large size and the complicated

*Corresponding author.

intrinsic structure of the corresponding systems that are used for describing these applications. Accelerating these complex simulations while retaining the accuracy therefore becomes a major challenge in many computational research works.

Model order reduction techniques have been proposed by many researchers to reduce simulation time. A commonly used approach is the projection-based technique. This technique employs a low-dimensional basis that can give the best approximation to the original solution space. There are several methods for constructing a reduced-order basis. In particular, this work focuses on using a well-known technique so-called proper orthogonal decomposition (POD). Recently, researchers have applied POD in many fields. In [1], the authors apply POD to predict in real-time steady-state capsule deformed shape. POD is also used in the construction of model reduction of particle processes in fluid flow described by Population Balance Equations (PBEs) as discussed in [2]. Moreover, the authors of [3] demonstrated how to modify stochastic tail buffeting prediction methods using POD. More details and applications of POD can be found in [4–8]. In this work, we utilize POD in the construction of reduced-order model. POD basis is used with the Galerkin projection to project the full-order model onto a low-dimensional subspace, which can extremely reduce the size of the original problem. However, the effective dimension reduction of POD-Galerkin approach is restricted to the linear part of the problem. To overcome the complexity issue on nonlinearity, an additional nonlinear model reduction method has to be applied to handle this problem.

In the case of nonlinear problem, several techniques have been introduced by many researchers. In general setting, the discrete empirical interpolation method (DEIM) is an efficient way to deal with nonlinearity. DEIM is employed to generate the selected interpolation indices that provide a nearly optimal subspace approximation to the nonlinear term. As a result, the numerical complexity becomes proportional to a small number of selected indices. DEIM procedure is used in conjunction with POD-Galerkin technique to further reduce the complexity of nonlinear term. The POD-DEIM approach has been applied in various applications. In [9], M. Dehghan, M. Abbaszadeh introduced a combination of POD-DEIM and meshless local RBF-DQ approach for prevention of groundwater contamination. Furthermore, POD-DEIM reduced-order method for stochastic Allen-Cahn equations with multiplicative noise is presented in [10]. The author of [11] applied POD to project the original large-scale full chemical process model onto a small system of a reduced model space, while DEIM is used to evaluate the nonlinear functions at a small set of the interpolation points. More articles on POD-DEIM can be found in [12–17].

Uncertainty quantification is another factor that would become practicable. In fact, quantities of interest in numerical simulations also depend on the environment of the problem and strongly influence simulation results. A common approach for this is to use one single global basis for constructing a reduce order model. However, this approach does not work well when uncertainties strongly affect the simulation results. Recently, machine learning techniques have been integrated to improve the reduced-order modeling. In [18], this work shows a recurrent neural network (RNN) closure of parametric POD-Galerkin reduced-order model. The parametric non-intrusive reduced order model (P-NIROM) based on Gaussian process regression method is presented as described in [19]. Furthermore in [20], a new deep-learning-based reduced-order modeling (ROM) framework is proposed for application in subsurface flow simulation. Most of these approaches are generally specific to certain classes of problems. This work constructs a POD-DEIM

reduced-order model that uses support vector machine (SVM) for the parametrized Burgers' equation, which is the standard test problem for general complex systems used in fluid dynamics. We first build a dictionary of projection basis sets corresponding to different parameter values. SVM is then employed to construct a classifier that assigns the best suitable basis in the dictionary prior to construct the POD-DEIM reduced-order model.

The article is arranged as follows. The problem formulation for the parametrized Burgers' equation is given in Section 2. In Section 3, we describe how to construct the POD-DEIM reduced system in Subsection 3.1 and 3.2. In Section 3.3, dictionary-based model order reduction via SVM is described and applied to the POD-DEIM reduced system. Section 4, numerical experiments illustrate the efficiency of our methodology as well as demonstrate the superiority of our approach over the standard global basis approach for approximating solutions for the Burgers' equations with various parameter values. Finally, the conclusion is given in Section 5.

2. Problem Formulation

Consider the Burgers' equation [21] given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \qquad x \in [0,1], \qquad t \in [0,1], \tag{2.1}$$

where Re is a Reynolds number. This PDE can be solved exactly to obtain the solution given by

$$u(x,t) = \frac{\frac{x}{t+1}}{1+\sqrt{\frac{t+1}{v}}\exp(Re\frac{x^2}{4t+4})} \tag{2.2}$$

where $v = \exp(\frac{Re}{8})$. Note that (2.2) will be used to compute the initial and boundary conditions. To construct the full-order model for (2.1), the first and second order central difference schemes are applied on the space domain. By using n space grid points, the PDE then becomes an n-dimension system of ODEs written by

$$\frac{d}{dt}\mathbf{u}(t) = \frac{1}{Re}\Big[\mathbf{B}\mathbf{u}(t) + \mathbf{b}(t)\Big] - \underbrace{\mathbf{u}(t).*\Big[\mathbf{A}\mathbf{u}(t) + \mathbf{a}(t)\Big]}_{\mathbf{F}(t,\mathbf{u}(t))}, \tag{2.3}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ are coefficient matrices, $\mathbf{a}(t), \mathbf{b}(t) : [0,1] \to \mathbb{R}^n$ reflect the boundary conditions and $\mathbf{u}(t) : [0,1] \to \mathbb{R}^n$ is the time-dependent solution vector. Finally, we apply the first order forward difference scheme to (2.3) on the time domain. As a result, the system can be solved to obtain the solution each time step. If we use m time steps, we will have m time-dependent solution vectors (often called m snapshots) containing n features denoted by

$$\mathbf{U} = \begin{bmatrix} u(x_1,t^{(1)}) & u(x_1,t^{(2)}) & & u(x_1,t^{(m)}) \\ u(x_2,t^{(1)}) & u(x_2,t^{(2)}) & & u(x_2,t^{(m)}) \\ \vdots & \vdots & \cdots & \vdots \\ u(x_n,t^{(1)}) & u(x_n,t^{(2)}) & & u(x_n,t^{(m)}) \end{bmatrix} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}. \tag{2.4}$$

3. Methodology

Our objective is to economically and accurately predict a quantity of interest that satisfies a PDE depending on an input parameter. Let consider our problem, where a quantity of interest, i.e., a solution $\mathbf{U}$ depends on an input parameter, i.e., a Reynolds number Re. More precisely, for a certain Reynolds number Re, a solution is given by

$$Re \in P \to \mathbf{U}(Re) \in U, \tag{3.1}$$

where P is the the parameter space and U is the solution manifold. As the input parameter $Re \in P$ is modified, the solution $\mathbf{U}(Re)$ evolves on a manifold U. In the following subsections, let us introduce some useful concepts and show how to apply them to our problem step-by-step.

3.1. Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) [20] is a method for constructing a low-dimensional basis (often called POD basis) of a subspace in Hilbert space. Assume that we want to find a basis of the space spanned by the solution $\mathbf{U} \in \mathbb{R}^{n\times m}$ defined in (2.4). Define $\mathcal{U} := \text{span}(\mathbf{U})$. To obtain POD basis of $\mathcal{U}$, the SVD of $\mathbf{U}$ is computed and it can be expressed as

$$\mathbf{U} = \mathbf{V}\Sigma\mathbf{W}^T, \tag{3.2}$$

where $\mathbf{V} \in \mathbb{R}^{n\times r}$, $\mathbf{W} \in \mathbb{R}^{m\times r}$ are matrices with orhonormal columns, called left and right singular matrices of $\mathbf{U}$, respectively, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ also known as singular values and $r = \text{rank}(\mathbf{U})$. The k-dimensional POD basis of $\mathcal{U}$ is the first $k \ll r$ columns of $\mathbf{V}$ denoted by $\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$. It is well-known that the k-dimensional POD basis $\mathbf{V}_k$ of $\mathcal{U}$ is the solution to the following minimization problem

$$\min_{\{\phi_i\}_{i=1}^k} \sum_{j=1}^{m} \|\mathbf{u}_j - \sum_{i=1}^{k} (\mathbf{u}_j^T \phi_i)\phi_i\|_2^2, \tag{3.3}$$

satisfying constrains $\phi_i^T \phi_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta funciton. Therefore, it can be used to approximate the snapshots as the following form

$$\mathbf{u}(t) \approx \mathbf{V}_k \mathbf{z}(t), \tag{3.4}$$

where $\mathbf{z}(t) : [0, 1] \to \mathbb{R}^k$ is the reduced representation of $\mathbf{u}(t)$. The minimum error of approximating the snapshots $\mathbf{U}$ by the k-dimensional POD basis $\mathbf{V}_k$ is given by

$$\sum_{j=1}^{m} \|\mathbf{u}_j - \sum_{i=1}^{k} (\mathbf{u}_j^T \mathbf{v}_i)\mathbf{v}_i\|_2^2 = \sum_{i=k+1}^{r} \sigma_i^2. \tag{3.5}$$

To construct the reduced-order model, we substitute (3.4) into (2.3). It yields that

$$\frac{d}{dt}\mathbf{V}_k\mathbf{z}(t) = \frac{1}{Re}\left[\mathbf{B}\mathbf{V}_k\mathbf{z}(t) + \mathbf{b}(t)\right] - \mathbf{F}(t, \mathbf{V}_k\mathbf{z}(t)), \tag{3.6}$$

$$\mathbf{V}_k\frac{d}{dt}\mathbf{z}(t) = \frac{1}{Re}\left[\mathbf{B}\mathbf{V}_k\mathbf{z}(t) + \mathbf{b}(t)\right] - \mathbf{F}(t, \mathbf{V}_k\mathbf{z}(t)). \tag{3.7}$$

By pre-multiplying both sides of (3.7) by $\mathbf{V}_k^T$ (often called Galerkin projection), the orthogonality condition of $\mathbf{V}_k$ ($\mathbf{V}_k^T\mathbf{V}_k = \mathbf{I}_k$) gives us that

$$\frac{d}{dt}\mathbf{z}(t) = \frac{1}{Re}\left[\underbrace{\mathbf{V}_k^T\mathbf{B}\mathbf{V}_k}_{\tilde{\mathbf{B}}}\mathbf{z}(t) + \underbrace{\mathbf{V}_k^T\mathbf{b}(t)}_{\tilde{\mathbf{b}}(t)}\right] - \underbrace{\mathbf{V}_k^T\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t))}_{\tilde{\mathbf{f}}(t,\mathbf{V}_k\mathbf{z}(t))}. \tag{3.8}$$

Hence, the full-order model as in (2.3) with very large size n is reduced to be (3.8) of size k.

However, the dimensionality reduction of POD-Galerkin approach does not cover the nonlinear term in (3.8). Particularly, to compute

$$\tilde{\mathbf{f}}(t,\mathbf{V}_k\mathbf{z}(t)) := \underbrace{\mathbf{V}_k^T}_{k\times n}\underbrace{\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t))}_{n\times 1}, \tag{3.9}$$

it is required to perform the multiplication depending on n which is still costly. Therefore, we employ another efficient technique to overcome the complexity that occurs on the nonlinear term as discussed in Subsection 3.2.

3.2. Discrete Empirical Interpolation Method

Discrete empirical interpolation method (DEIM) [22] is an approach to reduce the complexity for evaluating the nonlinear term. To illustrate this issue, we consider again the nonlinear part in (3.9), i.e.,

$$\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t)) := \mathbf{V}_k\mathbf{z}(t).*\left[\mathbf{A}\mathbf{V}_k\mathbf{z}(t) + \mathbf{a}(t)\right]. \tag{3.10}$$

Let $\{\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_m\} \subset \mathbb{R}^n$ be the set of nonlinear snapshots, where $\mathbf{f}_j = \mathbf{F}(t_j,\mathbf{u}(t_j))$ for all $j = 1,2,\ldots,m$. Note that nonlinear snapshots $\mathbf{f}_j$ are computed and collected from (2.3) simultaneously with snapshots $\mathbf{u}_j$. Define $\mathbf{F} := [\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_m] \in \mathbb{R}^{n\times m}$ with $r_f = \operatorname{rank}(\mathbf{F})$ and suppose that $\mathcal{F} := \operatorname{span}(\mathbf{F})$. We first apply the SVD on $\mathbf{F}$, i.e., $\mathbf{F} = \bar{\mathbf{V}}\bar{\Sigma}\bar{\mathbf{W}}^T$. Similarly, we utilize the POD basis of dimension $l \ll r_f$ of $\mathcal{F}$ denoted by $\bar{\mathbf{V}}_l$ to approximate the nonlinear term as well. This follows that

$$\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t)) \approx \bar{\mathbf{V}}_l\mathbf{c}(t), \tag{3.11}$$

where $\mathbf{c}(t) : [0,1] \to \mathbb{R}^l$ is the reduced representation of $\mathbf{f}(t,\mathbf{V}_k\mathbf{z}(t))$. The DEIM procedure is started here to select l rows of (3.11). Let $\mathbf{P}$ be a matrix used in the interpolation defined as

$$\mathbf{P} = [\mathbf{e}_{\wp_1},\mathbf{e}_{\wp_2},\ldots,\mathbf{e}_{\wp_l}] \in \mathbb{R}^{n\times l}, \tag{3.12}$$

where $\mathbf{e}_{\wp_i} = [0,\ldots,0,1,0,\ldots,0]^T$ is the $\wp_i$-th column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$. By pre-multiplying both sides of (3.11) by $\mathbf{P}^T$, the selection of components in the nonlinear term is made as follows

$$\mathbf{P}^T\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t)) \approx \underbrace{\mathbf{P}^T\bar{\mathbf{V}}_l}_{l\times l}\mathbf{c}(t). \tag{3.13}$$

Assume that $\mathbf{P}^T\bar{\mathbf{V}}_l$ is invertible. Then $\mathbf{c}(t)$ can be determined uniquely as

$$\mathbf{c}(t) \approx (\mathbf{P}^T\bar{\mathbf{V}}_l)^{-1}\mathbf{P}^T\mathbf{F}(t,\mathbf{V}_k\mathbf{z}(t)). \tag{3.14}$$

Note that the matrix $\mathbf{P}^T\bar{\mathbf{V}}_l$ is always nonsingular which the clarification can be found in [23]. As a result, (3.11) becomes

$$\mathbf{F}(t, \mathbf{V}_k\mathbf{z}(t)) \approx \bar{\mathbf{V}}_l(\mathbf{P}^T\bar{\mathbf{V}}_l)^{-1}\mathbf{P}^T\mathbf{F}(t, \mathbf{V}_k\mathbf{z}(t)). \tag{3.15}$$

The interpolation indices $\wp_1, \wp_2, \ldots, \wp_l$ are generated by the DEIM algorithm shown in Algorithm 1. The interpolation matrix $\mathbf{P}$ is provided to reduce the complexity in the

Algorithm 1 DEIM

Input : $l \ll p$, $\bar{\mathbf{V}} = [\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \ldots, \bar{\mathbf{v}}_p] \in \mathbb{R}^{n\times p}$
procedure
  $[|\rho|, \wp_1] = \max\{|\bar{\mathbf{v}}_1|\}$
  $\hat{\mathbf{V}} = [\bar{\mathbf{v}}_1]$, $\hat{\mathbf{P}} = [\mathbf{e}_{\wp_1}]$, $\hat{\wp} = [\wp_1]$
  for $i = 2 : p$ **do**
    $\mathbf{c} = (\hat{\mathbf{P}}^T\hat{\mathbf{V}})^{-1}\hat{\mathbf{P}}^T\bar{\mathbf{v}}_i$
    $\mathbf{r} = \bar{\mathbf{v}}_i - \hat{\mathbf{V}}\mathbf{c}$
    $[|\rho|, \wp_i] = \max\{|\mathbf{r}|\}$
    $\hat{\mathbf{V}} \leftarrow [\hat{\mathbf{V}} \;\; \bar{\mathbf{v}}_i]$, $\hat{\mathbf{P}} \leftarrow [\hat{\mathbf{P}} \;\; \mathbf{e}_{\wp_i}]$, $\hat{\wp} \leftarrow \begin{bmatrix} \hat{\wp} \\ \wp_i \end{bmatrix}$
  end for
end procedure
Output : $\mathbf{P} = \hat{\mathbf{P}}(:, 1:l) \in \mathbb{R}^{n\times l}$, $\vec{\wp} = \hat{\wp}(1:l) = [\wp_1, \wp_2, \ldots, \wp_l]^T \in \mathbb{R}^l$

nonlinear term. By substituting (3.15) into (3.8), this follows that the reduced-order model is completely independent of the full computation as shown below

$$\frac{d}{dt}\mathbf{z}(t) = \frac{1}{Re}\left[\underbrace{\tilde{\mathbf{B}}}_{k\times k}\underbrace{\mathbf{z}(t)}_{k\times 1} + \underbrace{\tilde{\mathbf{b}}(t)}_{k\times 1}\right] - \underbrace{\mathbf{V}_k^T\bar{\mathbf{V}}_l}_{k\times l}\underbrace{(\mathbf{P}^T\bar{\mathbf{V}}_l)^{-1}}_{l\times l}\underbrace{\mathbf{P}^T\mathbf{F}(t, \mathbf{V}_k\mathbf{z}(t))}_{l\times 1}. \tag{3.16}$$

Note that (3.16) is often called the POD-DEIM reduced-order model. This equation is used to update the reduced representation $\mathbf{z}_j := \mathbf{z}(t_j)$ for all $j = 1, 2, \ldots, m$. The original solution is approximated by projecting the reduced representation back to the original solution space, i.e.,

$$\mathbf{u}_j \approx \mathbf{V}_k\mathbf{z}_j, \quad j = 1, 2, \ldots, m. \tag{3.17}$$

Define $\mathbf{F}^{\mathrm{DEIM}}(t, \mathbf{u}(t)) := \bar{\mathbf{V}}_l(\mathbf{P}^T\bar{\mathbf{V}}_l)^{-1}\mathbf{P}^T\mathbf{F}(t, \mathbf{u}(t))$. An error bound between the nonlinear function $\mathbf{F}$ and its DEIM approximation $\mathbf{F}^{\mathrm{DEIM}}$ is given [23] by

$$\|\mathbf{F} - \mathbf{F}^{\mathrm{DEIM}}\|_2 \le \mathcal{C}\|(\mathbf{I}_n - \bar{\mathbf{V}}_l\bar{\mathbf{V}}_l^T)\mathbf{F}\|_2, \tag{3.18}$$

where $\mathcal{C} = \|(\mathbf{P}^T\bar{\mathbf{V}}_l)^{-1}\|_2$. More details on error analysis of DEIM and its extension can be found in [24, 25]

In general, the resulting POD-DEIM reduced-order model is not designed to approximate entirely the solution manifold U for the parameter space of Re. In the next subsection, we will introduce how to handle this problem.

3.3. Dictionary-based Model Order Reduction using SVM

As we have discussed in Subsection 3.2, it is quite challenging to build a single global reduced-order model that covers entirely the whole part of solution manifold U. However, U is not generally embedded in a low-dimensional vector space, using a single global reduced-order model would result inaccurate simulation. This leads us some ideas to split U into K parts and construct a dictionary of K local reduced-order models. Before giving a result of dictionary-based model order reduction, let us introduce some useful concepts.

Definition 3.1. The Grassmann manifold (Grassmannian) $Gr(k, \mathbf{U})$ is the set whose points are k-dimensional subspaces in $\mathbf{U}$.

Definition 3.2. The doubly-infinite Grassmannian $Gr(\infty, \infty)$ is the Grassmannian that models subspaces of all dimension regardless of ambient space.

Consider the doubly-infinite Grassmannian $Gr(\infty, \infty)$. Let $\mathcal{U}^a, \mathcal{U}^b \in Gr(\infty, \infty)$ be two subspaces with dimensions a and b, respectively. One can define a distance in $Gr(\infty, \infty)$ [26] by

$$d_{Gr(\infty,\infty)}(\mathcal{U}^a, \mathcal{U}^b) = \left(\frac{\pi^2}{4} |a - b| + \sum_{i=1}^{\min(a,b)} \alpha_i^2 \right)^{1/2}, \tag{3.19}$$

where $\alpha_i = \cos^{-1}(\bar{\sigma}_i)$ are the principal angles obtained by the SVD of $\mathbf{V}_a^T \mathbf{V}_b$ with $\mathbf{V}_a$ and $\mathbf{V}_b$ being orthogonal matrices whose columns form orthonormal bases of $\mathcal{U}^a$ and $\mathcal{U}^b$, respectively, and $\bar{\sigma}_i$'s are singular values of $\mathbf{V}_a^T \mathbf{V}_b$. In order to compute (3.19), we utilize the POD method in Subsection 3.1 to generate orthonormal bases for any subspaces. Note that (3.19) is called the Grassmann metric. To separate the solution manifold U into K clusters, we first generate N training samples of Reynolds number denoted by $\{Re_s\}_{s=1}^N$. Define snapshots corresponding to an input Re_s, $\mathbf{U}(Re_s) := \mathbf{U}^s = [\mathbf{u}_1^s, \mathbf{u}_2^s, \ldots, \mathbf{u}_m^s]$ with $r_s = \text{rank}(\mathbf{U}^s)$. We apply the POD method to obtain the k_s-dimensional POD basis $\mathbf{V}_{k_s}^s$ of $\mathcal{U}^s := \text{span}(\mathbf{U}^s)$. Since the dimension of POD basis can be variously considered, we define a criterion for selecting an unbiased choice of k_s as follows

$$I(e) = \frac{\sum_{i=1}^{e} \sigma_i(s)}{\sum_{i=1}^{r_s} \sigma_i(s)}, \tag{3.20}$$

where $k_s = \text{argmin}\{e : I(e) \geq \gamma\}$, $\gamma \in [0, 1]$ and $\sigma_i(s)$'s are singular values of $\mathbf{U}^s$. We often called (3.20) as the retained energy, which is the ratio of the information captured by the subspace spanned by k_s-dimensional POD basis. For a fixed γ, the Grassmann metric is computed for all pairs of Reynolds numbers, resulting in a dissimilarity matrix defined as

$$\boldsymbol{\delta} := [\delta_{ij}] = [d_{Gr(\infty,\infty)}(\mathcal{U}^s, \mathcal{U}^{s'})] \in \mathbb{R}^{N \times N}, \tag{3.21}$$

where $\mathbf{V}_{k_s}^s$ and $\mathbf{V}_{k_{s'}}^{s'}$ are POD bases for $\mathcal{U}^s$ and $\mathcal{U}^{s'}$, respectively. For the clustering task, the K-medoids clustering algorithm is applied together with the distance information in (3.21) and can be summarized as follows.

1. K training samples in the dataset are chosen randomly to be initial medoids.
2. Each point of the dataset is assigned to the cluster corresponding to its closest medoid.
3. For each cluster, a new medoid is updated by finding the best point that minimizes the sum of distances to all point in the cluster.

4. The last two steps are repeated until convergence.

After the clustering process is done, we have a disjoint union $U = \bigcup_{\ell=1}^{K} U_\ell$ satisfying $U_\ell \neq \emptyset$ for all $\ell = 1, 2, \ldots, K$. Also, we have a set of medoids $\{\mathbf{U}(\tilde{Re}_\ell) := \tilde{\mathbf{U}}^\ell\}_{\ell=1}^{K}$. Note that using the Grassmann metric implies that two inputs leading to solutions lying in nearby subspaces are then considered to be similar. More precisely, Re_s and $Re_{s'}$ are designated as the label ℓ if both $\mathbf{U}^s$ and $\mathbf{U}^{s'}$ are in the same part of solution manifold U_ℓ. The clustering results are provided to construct a classifier $C(U)_K : P \to \{1, 2, \ldots, K\}$. The classifier $C(U)_K$ is trained in a supervised fashion from pairs of $(Re_s, \ell) \in (P \times \{1, 2, \ldots, K\})$, where ℓ is the label of cluster containing Re_s satisfying

$$\ell = \underset{\flat \in \{1,2,\ldots,K\}}{\operatorname{argmin}} \left(d_{Gr(\infty,\infty)}(\mathcal{U}^s, \tilde{\mathcal{U}}^\flat)\right), \tag{3.22}$$

where $\tilde{\mathcal{U}}^\ell = \operatorname{span}(\tilde{\mathbf{U}}^\ell)$, $\ell = 1, 2, \ldots, K$. For the training process, it is done through classification process using linear support vector machine (SVM). Therefore, we utilize the classifier $C(U)_K$ as an automatic data labelling by transferring an unseen sample to the best suitable cluster. This implies that

$$Re \xrightarrow{C(U)_K} \ell \quad \Rrightarrow \quad \mathbf{u}(t) \approx \tilde{\mathbf{V}}_k^\ell \mathbf{z}(t), \tag{3.23}$$

where $\tilde{\mathbf{U}}^\ell = \tilde{\mathbf{V}}^\ell \tilde{\Sigma}^\ell \tilde{\mathbf{W}}^{\ell^T}$ and k can be arbitrarily chosen based on criterion (3.20). Note that using smaller k reduces computational time but this gives a trade-off in accuracy. Similarly, the classifier for selecting a POD basis for nonlinear term in the DEIM approximation is built in the same way. For any given input Reynolds number, this procedure is performed to select the best appropriate POD basis for solution snapshots and nonlinear snapshots in their dictionaries. The resulting POD bases are then used in the construction of the POD-DEIM reduced-order model as described before in Subsection 3.1 and 3.2.

Numerical results of dictionary-based model order reduction via POD-DEIM with SVM is demonstrated and discussed in the next section.

4. Numerical Results

In this section, we perform our methodology on the Burgers' equation as defined in (2.1). The results are shown for Reynolds number value which were not within training set. The model is set up as follows.

- Full-order model (FOM) : $n = 400$, $m = 4000$,
- Training samples : $N = 500$, where $Re \in [100, 2000]$,
- Clusters : $K_u = 6$ (snapshots), $K_f = 10$ (nonlinear snapshots),
- Retained energy (3.20): $\gamma = 0.95$.

In our proposed approach, the followings are the steps that have to be performed during the offline process.

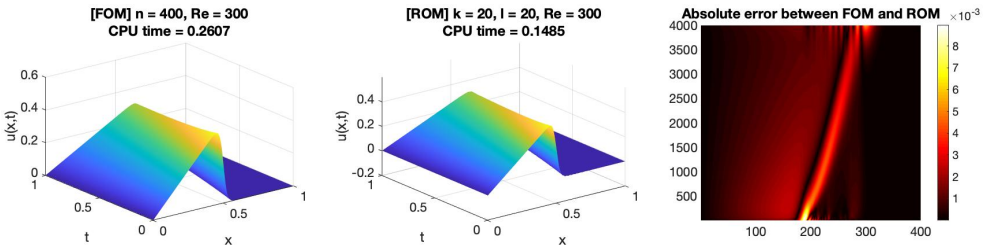
- Solving $N = 500$ training samples for the Burgers' equation (2.1) and collecting $\{\mathbf{U}^s\}_{s=1}^{500}$, $\{\mathbf{F}^s\}_{s=1}^{500}$.
- Computing dissimilarity matrices $\boldsymbol{\delta}_u$, $\boldsymbol{\delta}_f \in \mathbb{R}^{500\times500}$ for both snapshots and nonlinear snapshots via Grassmann metric (3.19).
- Clustering the dataset into $K_u = 6$ and $K_f = 10$ clusters via K-medoids clustering algorithm, resulting in Table 1 and 2, and recording their medoid POD bases into dictionaries.
- Training classifiers $C(U)_{K_u}$ and $C(F)_{K_f}$ via Linear SVM in MATLAB.

TABLE 1. K-medoids clustering result using $K_u = 6$ for snapshots.

Class label (ℓ_u)	Medoid Reynolds number ($\tilde{Re}_{\ell_u}$)
1	1326
2	384
3	1423
4	600
5	970
6	1843

TABLE 2. K-medoids clustering result using $K_f = 10$ for nonlinear snapshots.

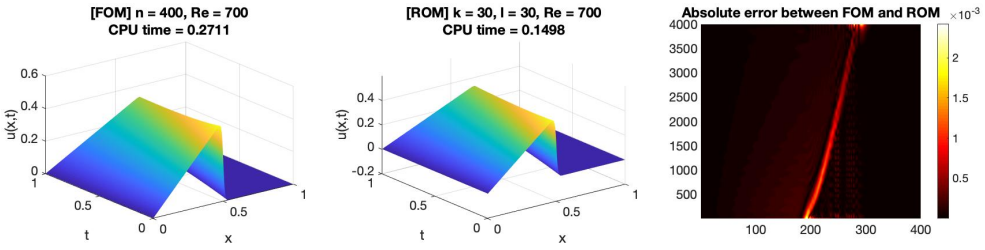
Class label (ℓ_f)	Medoid Reynolds number ($\tilde{Re}_{\ell_f}$)
1	418
2	245
3	1354
4	902
5	1273
6	1448
7	1105
8	1658
9	1843
10	594



(a) (b) (c)

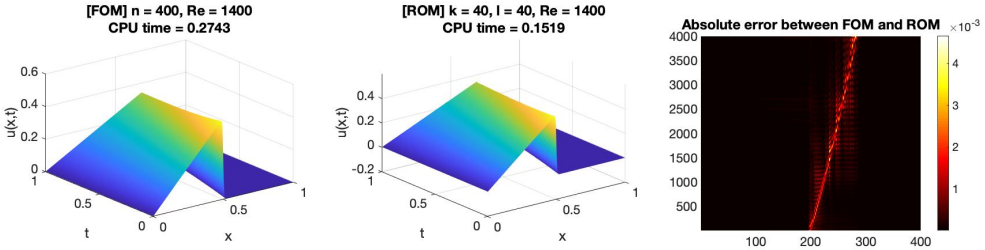
FIGURE 1. $Re = 300$ is considered. $Re = 300 \xrightarrow[C(F)_{10}]{C(U)_6} \begin{matrix} \ell_u = 2(384) \\ \ell_f = 2(245) \end{matrix}$. POD-DEIM reduced-order model is constructed using $k = l = 20$.

As illustrated in Figures 1, 2, 3 and 4, we show the results for different 4 values of Reynolds number $Re = 300, 700, 1400, 1900$. For the reduced-order model, the dimensions of POD bases for both linear (k) and nonlinear (l) terms are simply chosen to be the same ($k = l$) for each case. The CPU time for full-order models ($n = 400$) take approximately 0.26s$\sim$0.3s while the CPU time for POD-DEIM reduced-order models take around 0.15s depending on the number of POD bases used. For a fixed $k = l = 40$, the average



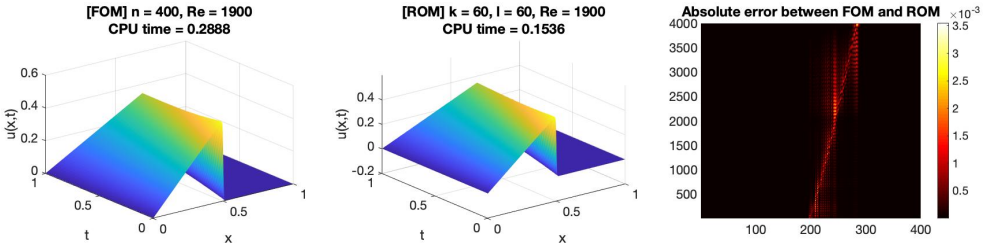
(a) (b) (c)

FIGURE 2. $Re = 700$ is considered. $Re = 700 \xrightarrow[C(F)_{10}]{C(U)_6} \begin{matrix} \ell_u = 4(600) \\ \ell_f = 10(594) \end{matrix}$. POD-DEIM reduced-order model is constructed using $k = l = 30$.



(a) (b)

FIGURE 3. $Re = 1400$ is considered. $Re = 1400 \xrightarrow[C(F)_{10}]{C(U)_6} \begin{matrix} \ell_u = 3(1423) \\ \ell_f = 3(1354) \end{matrix}$. POD-DEIM reduced-order model is constructed using $k = l = 40$.



(a) (b) (c)

FIGURE 4. $Re = 1900$ is considered. $Re = 1900 \xrightarrow[C(F)_{10}]{C(U)_6} \begin{matrix} \ell_u = 6(1843) \\ \ell_f = 9(1843) \end{matrix}$. POD-DEIM reduced-order model is constructed using $k = l = 60$.

CPU times for POD-DEIM reduced-order model are compared to the full-order models for different Reynolds numbers illustrated as in Table 3. Particularly, higher Reynolds number needs more number of POD bases to preserve the accuracy. This means that the part of solution manifold corresponding to large Reynolds number is embedded in the high-dimensional vector space.

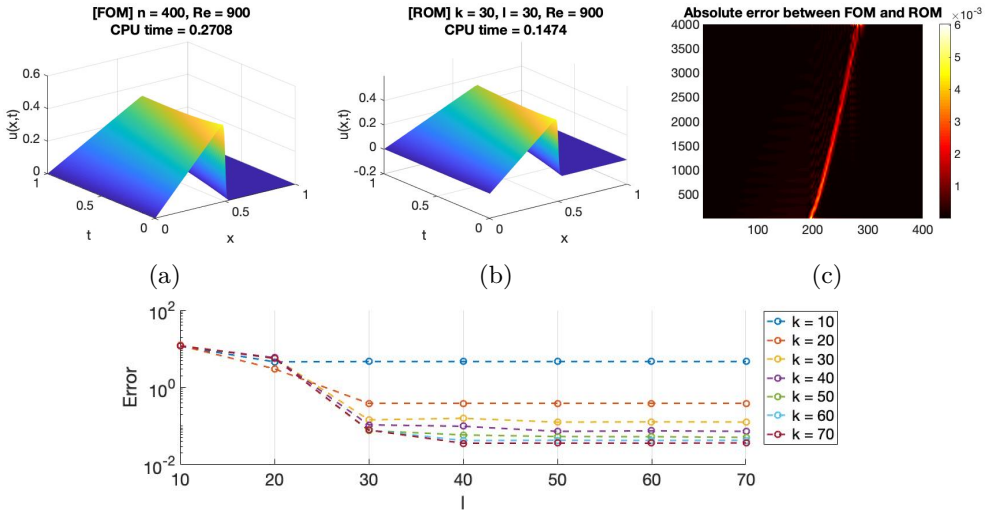


FIGURE 5. $Re = 900$ is considered. $Re = 900 \xrightarrow[C(F)_{10}]{C(U)_6} \begin{matrix} \ell_u = 5(970) \\ \ell_f = 4(902) \end{matrix}$. POD-DEIM reduced-order model is constructed using $k = l = 30$ (Top). Error defined in (4.1) is plotted for fixed values $k = 10, 20, \ldots, 70$ with different values $l = 10, 20, \ldots, 70$ (Bottom).

TABLE 3. Comparing the average CPU time between FOM ($n = 400$) and ROM ($k = l = 40$) for different Reynolds number values.

	Avg. CPU time	
Re	FOM	ROM
300	0.2853s	0.1571s
700	0.2817s	0.1486s
900	0.2762s	0.1488s
1200	0.2795s	0.1616s
1400	0.2767s	0.1528s
1700	0.2839s	0.1527s
1900	0.2816s	0.1483s

We define an error between the full-order solution and the POD-DEIM approximation as

$$\text{Error} = \sum_{k=1}^{m} \|\mathbf{u}_j - \mathbf{V}_k \mathbf{z}_j\|_2, \tag{4.1}$$

where $\mathbf{u}_j$ is the numerical solution from the full-order system at time t_j and $\mathbf{z}_j$ is the numerical solution from a reduced-order system at time t_j.

As shown in Figure 5, using $k = l = 30$ is identified as a good compromise for $Re = 900$ since the Error defined in (4.1) is of order $\mathcal{O}(10^{-2})$ and there is roughly no further decay for $k, l \geq 30$. Similarly, Figure 6 and Figure 7 indicate that $k = l = 40$ and $k = l = 50$

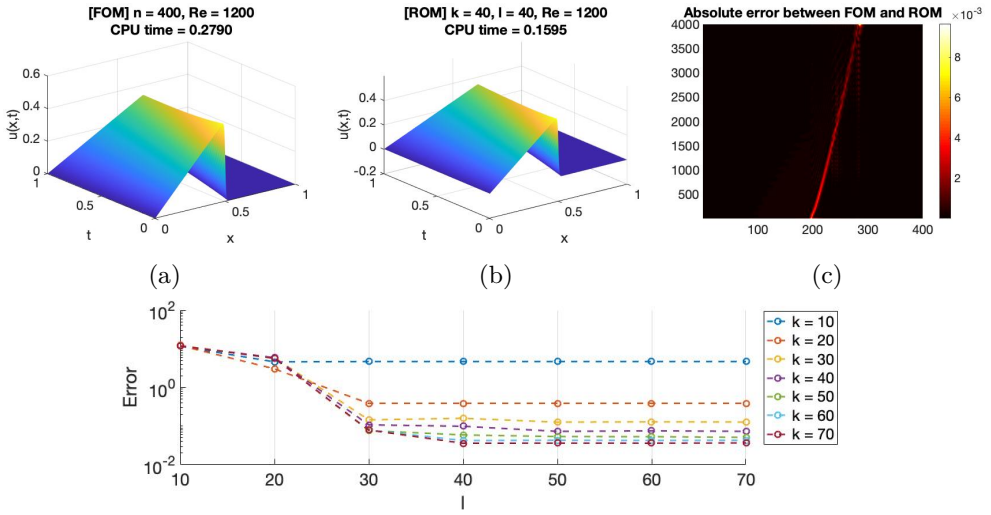


FIGURE 6. $Re = 1200$ is considered. $Re = 1200 \xrightarrow[C(F)_{10}]{C(U)_6} \dfrac{\ell_u = 1(1326)}{\ell_f = 5(1273)}$. POD-DEIM reduced-order model is constructed using $k = l = 40$ (Top). Error defined in (4.1) is plotted for fixed values $k = 10, 20, \ldots, 70$ with different values $l = 10, 20, \ldots, 70$ (Bottom).

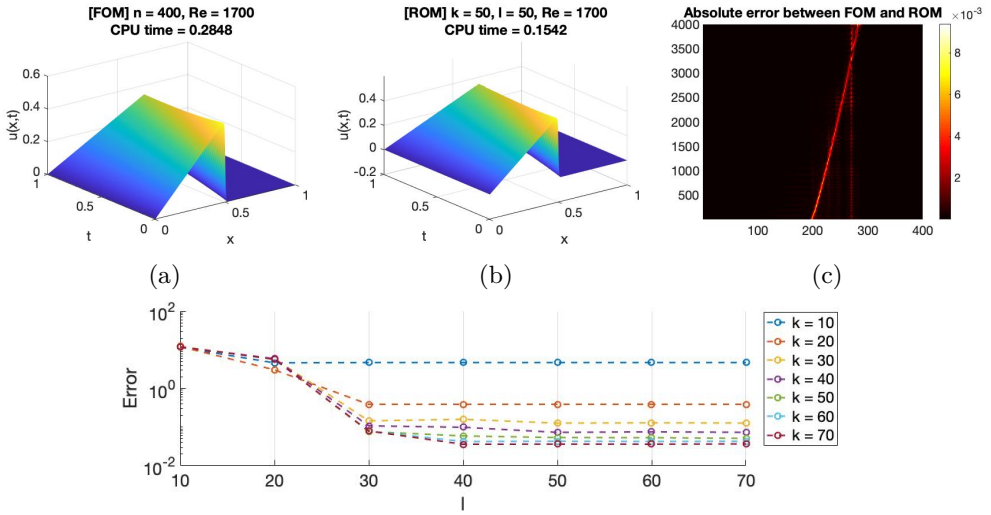
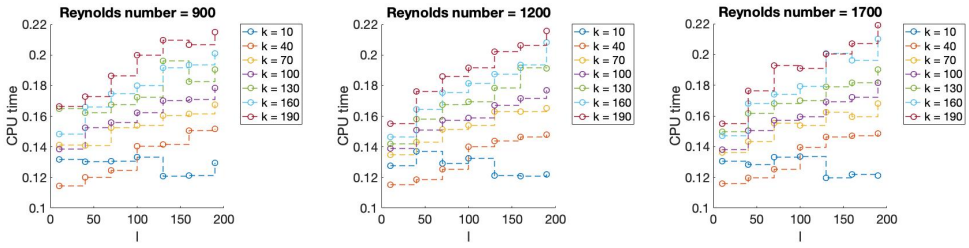


FIGURE 7. $Re = 1700$ is considered. $Re = 1700 \xrightarrow[C(F)_{10}]{C(U)_6} \dfrac{\ell_u = 6(1843)}{\ell_f = 8(1658)}$. POD-DEIM reduced-order model is constructed using $k = l = 50$ (Top). Error defined in (4.1) is plotted for fixed values $k = 10, 20, \ldots, 70$ with different values $l = 10, 20, \ldots, 70$ (Bottom).

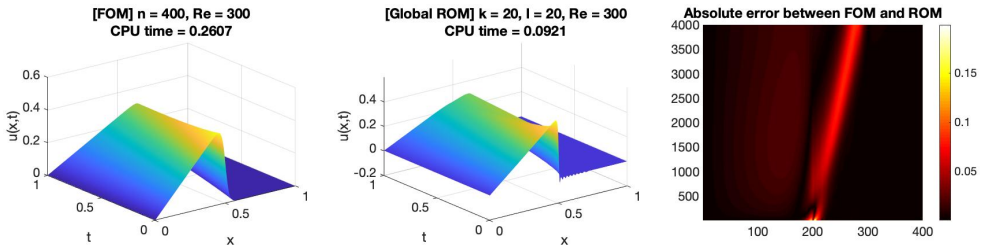


(a) (b) (c)

FIGURE 8. Plotting the CPU time for $Re = 900$ (Figure 8a), 1200 (Figure 8b), 1700 (Figure 8c) corresponding to fixed values $k = 10, 30, \ldots, 190$ with different values $l = 10, 30, \ldots, 190$.

provide the most suitable dimension for $Re = 1200$ and $Re = 1700$, respectively. The CPU time for respective cases are also plotted as displayed in Figure 8. Obviously, the CPU times have tendency to be increasing when we use higher number of k and l. Although we could increase the number of POD bases k and l as much as we want, we strongly recommend to select the dimensions k and l that give errors proportional to truncation error arising from numerical schemes to avoid unnecessary increase in simulation time.

To illustrate the advantage of using the proposed method over the standard approach that uses the global basis, we consider the case of Reynolds number $Re = 300$. The global bases for both POD and DEIM approximation are constructed from equally-spaced Reynolds numbers $Re = 100, 200, \ldots, 2000$. We compare the approximate solution for Burgers' equation with $Re = 300$ by using reduced systems constructed from the global basis approach (shown in Figure 9b) and our proposed approach (shown in Figure 1b) with the same reduced dimensions $k = l = 20$. Notice that, even if the CPU time usage for these two approaches are equivalent, the global basis approach gives unstable approximate solution with larger error than our proposed approach, as shown in Figure 9c and Figure 1c, respectively. In summary, the numerical experiments in Figures 1 and 9 demonstrate that the global reduced-order basis may not be good enough to approximate the entire solution manifold and using local reduced-order basis to approximate each part of solution manifold can improve the accuracy.



(a) (b) (c)

FIGURE 9. $Re = 300$ is considered. POD-DEIM reduced-order model is constructed by global POD bases from both snapshots and nonlinear snapshots using $k = l = 20$, respectively.

5. Conclusion

In this work, the concept of dictionary-based model order reduction via POD-DEIM together with SVM is applied to the parametrized Burgers' equation. The dictionary of local reduced-order models is constructed by clustering the solution manifold where the distance is measured in doubly-infinite Grassmann manifold. Once the clusters have been defined, SVM is then employed to create an automatic data labelling by transferring input parameter to select the most appropriate reduced-order basis in dictionary to be used in the construction of POD-DEIM reduced-order model. The numerical experiments on the parametrized Burgers' equation demonstrate desirable results in terms of accuracy and speed. Our methodology is also shown to provide more accurate approximations than those obtained from reduced-order model using a single global basis.

Acknowledgements

This study was supported by Thammasat University Research Fund, Contracted No. 76/2564.

References

[1] C. Quesada, P. Villon, A. Salsac, Real-time prediction of the deformation of microcapsules using Proper Orthogonal Decomposition, Journal of Fluids and Structures, 101 (2021).

[2] D. Khlopov, M. Mangold, Automatic Model Reduction of Linear Population Balance Models by Proper Orthogonal Decomposition, IFAC-PapersOnLine, 48 (1)(2015) 11-16.

[3] L. Katzenmeier, C. Vidy, C. Breitsamter, Using a Proper Orthogonal Decomposition representation of the aerodynamic forces for stochastic buffeting prediction, Journal of Fluids and Structures, 99 (2020).

[4] M. Onori, N. J. Hills, Reduced order modelling for a rotor-stator cavity using proper orthogonal decomposition, Computers & Fluids, 216 (2021).

[5] A. Mohammadi, K. Shimoyama, M. S. Karimi, M. Raisee, Efficient uncertainty quantification of CFD problems by combination of proper orthogonal decomposition and compressed sensing, Applied Mathematical Modelling, 2021.

[6] S. Wang, S. Khatir, M. A. Wahab, Proper Orthogonal Decomposition for the prediction of fretting wear characteristics, Tribology International, 152 (2020).

[7] N. Sukuntee, and S. Chaturantabut. An Application of Proper Orthogonal Decomposition for Estimating Missing Data of Patients in Different Cause Groups, Thai Journal of Mathematics (2018) 21-33.

[8] S. Intawichai and S. Chaturantabut, A numerical study of efficient sampling strategies for randomized singular value decomposition, Thai Journal of Mathematics, (2020) 371-385.

[9] M. Dehghan, M. Abbaszadeh, A combination of proper orthogonal decomposition–discrete empirical interpolation method (POD-DEIM) and meshless local RBF-DQ approach for prevention of groundwater contamination, Computers & Mathematics with Applications, Volume 75 (4)(2018) 1390-1412.

[10] D. Chen, H. Song, The POD-DEIM reduced-order method for stochastic Allen–Cahn equations with multiplicative noise, Computers & Mathematics with Applications, Volume 80 (12)(2020) 2691-2706.
[11] V. B. Nguyen, S. B. Q. Tran, S. A. Khan, J. Rong, J. Lou, POD-DEIM model order reduction technique for model predictive control in continuous chemical processing, Computers & Chemical Engineering, 133 (2020).
[12] M. Isoz, POD-DEIM based model order reduction for speed-up of flow parametric studies, Ocean Engineering, 186 (2019).
[13] J. Bremer, P. Goyal, L. Feng, P. Benner, K. Sundmacher, POD-DEIM for efficient reduction of a dynamic 2D catalytic reactor model, Computers & Chemical Engineering, 106 (2017).
[14] Z. P. Stanko, S. E. Boyce, W. W.-G. Yeh, Nonlinear model reduction of unconfined groundwater flow using POD and DEIM, Advances in Water Resources, 97 (2016).
[15] N. Sukuntee, and S. Chaturantabut, Model Order Reduction for Sine-Gordon Equation Using POD and DEIM, Thai Journal of Mathematics (2018).
[16] C. Dechanubeksa and S. Chaturantabut, An Application of a Modified Gappy Proper Orthogonal Decomposition on Complexity Reduction of Allen-Cahn Equation, Algorithms, 13 (6)(2020) 148.
[17] N. Ploymaklam and S. Chaturantabut, Reduced-Order Modeling of a Local Discontinuous Galerkin Method for Burgers-Poisson Equations. Thai Journal of Mathematics, 18 (4)(2020) 2053-2069.
[18] Q. Wang, N. Ripamonti, J. S. Hesthaven, Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, Journal of Computational Physics, 410 (2020).
[19] D. Xiao, Error estimation of the parametric non-intrusive reduced order model using machine learning, Computer Methods in Applied Mechanics and Engineering, 355 (2019) 513-534.
[20] A. Chatterjee, An Introduction to the Proper Orthogonal Decomposition, Current Science, 78 (2000) 808-817.
[21] H. Bateman, Some recent researches on the motion of fluids, Monthly Weather Review, 43 (1915) 163-170.
[22] S. Chaturantabut and D.C. Sorensen, Discrete Empirical Interpolation for nonlinear model reduction, Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference (2009) 4316-4321.
[23] S. Chaturantabut, D. C. Sorensen, Nonlinear Model Reduction via Discrete Empirical Interpolation, SIAM Journal on Scientific Computing, 32 (2010) 2737-2764.
[24] S. Chaturantabut and D.C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction. SIAM Journal on numerical analysis, 50 (1)(2012) 46-63.
[25] S. Chaturantabut, Stabilized model reduction for nonlinear dynamical systems through a contractivity-preserving framework, International Journal of Applied Mathematics and Computer Science, 30 (4)(2020) 615-628.
[26] T. Daniel, F. Casenave, A. Nissrine, D. Ryckelynck, Model order reduction assisted by deep neural networks (ROM-net), Advanced Modeling and Simulation in Engineering Sciences, 7 (2020) 1-27.