Thai Journal of Mathematics
Volume 19 Number 4 (2021)
Pages 1729–1762

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Model Reduction for Fisher's Equation with an Error Bound

Suchinthra Rungpitaxmana and Saifon Chaturantabut*

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

Abstract This work considers a model-order reduction (MOR) for Fisher's equation, which is generally used to describe many physical systems, such as chemical reactions, flame propagation, neurophysiology, nuclear reactors, and tissue engineering. Due to the nonlinearity in this type of system, solving the resulting discretized model for accurate solution could be time-consuming as the dimension gets large. Model-order reduction can be applied to improve the process of solving this large discretized model. In this work, a projection-based method called Proper Orthogonal Decomposition (POD) will be used first to project the state variables of the system on a low dimensional subspace, which will result in the decrease of unknowns in the systems. However, the computational complexity of the discretized nonlinear term still depends on the original large dimension. Discrete Empirical Interpolation Method (DEIM) is therefore used to eliminate this inefficiency. This POD-DEIM approach is applied on Fisher's equation with discontinuous initial conditions. An apriori error bound is derived for the approximations from POD-DEIM reduced system for the semi-implicit numerical scheme. The usefulness of this approach is illustrated through the parametric study of the varying boundary conditions. This work also investigates the effect of adding the snapshot difference quotients to construct basis sets used in POD and POD-DEIM reduced systems. The numerical results show that this POD-DEIM can substantially decrease the computational time while providing accurate numerical solution.

MSC: 65F30; 3540; 34A45

Keywords: Discrete Empirical Interpolation Method (DEIM); Finite Difference Methods, Fisher's Equation, Model Order Reductions (MORs); Ordinary Differential Equations (ODEs); Partial Differential Equations (PDEs); Proper Orthogonal Decomposition (POD)

Submission date: 15.01.2017 / Acceptance date: 02.06.2019

1. Introduction

The goal for applying model order reduction is to construct reduced system with small dimension and guarantee that it still maintains the accuracy when compared to the original full-order model. Since many nonlinear partial differential equations in practical applications are often required to use high dimensional discretized system for obtaining

*Corresponding author.

Published by The Mathematical Association of Thailand.

accurate numerical solutions, the simulation can be extremely long. This motivates us to apply model reduction techniques to decrease the computational complexity.

Model reduction techniques such as Proper Orthogonal Decomposition (POD) [9, 30, 42, 44], Galerkin POD [17, 38, 52] , Petrov-Galerkin POD [22, 36, 74], Balanced Truncation [27, 41, 48, 53, 78], Balanced POD [61, 66], Transfer Function Interpolation [19, 24, 57] and Piecewise Tangential Interpolation [75] can be used to find a low-order model which approximates the high-dimensional full-order model. These techniques are projection-based reduction approaches. So far, the model reduction techniques are important in many applications such as the analysis of network modeling [34, 76], biochemical reaction networks [51, 55], and flow dynamics [10, 11]. This work will initially focus on one of the most commonly used model reduction approaches, called the Proper Orthogonal Decomposition (POD) technique.

Proper Orthogonal Decomposition (POD), also known as principal component analysis, the Karhunen-Lòeve expansion, Hotelling transform, or singular value decomposition, is one of the most popular basis selection methods for nonlinear models that provides low dimensional estimate of high dimensional subspace by extracting the main characteristics from the system of interest. The low-order approximation is obtained by projecting the full-order system onto a set of basis functions computed from empirical data. POD can give a set of basis, called POD basis, by using Singular Value Decomposition (SVD). POD constructs a low-dimensional basis that minimizes error in $2-$norm for a given fixed rank of basis. POD has been first used by J. L. Lumley in the turbulence flow [44]. It was used in the incomplete (gappy) data for compressible external aerodynamic problem [9]. POD has been successfully used in many works, such as in the analysis of turbulent flows [8], the analysis the complex flow phenomenon in a horizontal chemical vapor deposition reactor [46], modeling nonlinear flows with deforming meshes [26], particle image velocimetry wall-gradient measurements flow field [49]. Moreover, POD is extended by combining with balanced realization theory in the application of two-dimensional airfoil [71]. The model reduction approach that uses POD basis with the Galerkin projection is very efficient and accurate for linear dynamical systems. However, this approach may not be able to reduce the simulation time for solving nonlinear problems because the computational complexity of nonlinear term still depend on the large dimension of the original systems. Therefore, additional nonlinear model reduction approaches have to be considered.

There are some existing model reduction techniques for nonlinear systems, such as Empirical Interpolation Method (EIM) [7], Trajectory Piecewise-Linear [56, 69], Missing Point Estimation [67], Discrete Empirical Interpolation Method (DEIM) [1, 13, 14]. We will mainly focus on Discrete Empirical Interpolation Method (DEIM), since it can reduce the computational complexity of general nonlinear terms.

Discrete Empirical Interpolation Method (DEIM) [14] can be considered as an improvement of the POD algorithm. It estimates nonlinear term by finding projection basis from POD and selecting the interpolation indices by a greedy algorithm. The aim of DEIM is to select the interpolation indices by trying to minimize the error heuristically. It has been used in many applications, such as non-linear miscible viscous fingering in porous media [15] 1-D FitzHugh-Nagumo equations[13], morphological structure spiking neurons [37], non-linear miscible viscous fingering in a 2-D porous medium [15], 2-D shallow-water equations [18], Navier-Stokes equations [1], four-dimensional variational data assimilation [64], three-dimensional nonlinear aeroelasticity model [23], electrical, thermal, and micro-electromechanical models [32].

Fisher's equation, also known as Fisher-Kolmogorov-Petrowskii-Piscounov equation, Fisher-Kolmogorov equation, or Fisher-KPP equation (FKPP), is a nonlinear parabolic PDE. It is one of the simplest classical reaction diffusion equations and it is useful such as in physical, chemical reaction processes, heat and mass transfer [77], biological phenomena, optics, and combustion. Fisher's equation has been studied in many fields such as first suggested by R.A.Fisher in propagation of a gene within a population [25]. It was solved numerically by pseudospectral method [50] and finite volume method. However, so far, there is no existing work that applies MOR on Fisher's equation to reduce complexity during the simulation together with providing an apriori error analysis. This work focuses on applying model order reduction technique combining POD and DEIM on the finite difference discretization of the Fisher's equation, as well as deriving an error bound for the corresponding reduced-order solutions.

The accuracy of the POD-Galerkin approach has been studied in a number of previous works, such as the error analysis for POD model reduction of Navier-Stokes equations [31, 70], error estimate for tropical pacific ocean reduced gravity model [45], error of semidiscretized FitzHugh-Nagumo system [65], wave-like equations [12], error equilibration for linear and semilinear parabolic PDE, error of nonlinear parabolic PDE [35, 39], optical diffusion tomography [5]. The stability of the reduced order system is also an important aspect in numerical analysis. This work extends the error analysis in [16] and derives an a-priori error bound for the POD-DEIM approximations obtained from the semi-implicit numerical scheme used for solving Fisher's equation.

This work is organized as follows. In Section 2, model reduction techniques used in this work are introduced. These techniques are Proper Orthogonal Decomposition (POD) and Discrete Empirical Interpolation Method (DEIM) approximation. Then, an error bound for the solutions from POD-DEIM approach is derived in Section 3. Section 4 provides more details on Fisher's equation, as well as its discretization. Section 5 investigates the numerical application of POD-DEIM method on Fisher's eqation. The effect of adding snapshot difference equations in the process of constructing POD basis is studied. The efficiency of POD-DEIM approach is illustrated through the parametric study of varying boundary conditions. Finally, Section 6 gives the conclusions and possible extension of this work.

2. Preliminary

This section considers two model reduction techniques for nonlinear ordinary differential equations (ODEs). The model reduction techniques or model order reduction aims to reduce the number of unknowns and not to reduce the order of the derivatives. A common technique used in dimension reduction is Proper Orthogonal Decomposition (POD) combined with Galerkin projection. For nonlinear problems, this technique can reduced only dimensions of linear term, because POD cannot reduce the complexity of nonlinear term. Therefore, we apply POD approximation with Discrete Empirical Interpolation Method (DEIM), which can reduce the complexity of nonlinear term so that it does not depend on the large dimensions of full-order system.

The goal of model reduction techniques used in this work is to decrease the dimension of the discretized systems from nonlinear partial differential equations (PDEs). These discretized system is in the form of nonlinear ordinary differential equations (ODEs) in the following form

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{F}(\mathbf{y}(t)), \tag{2.1}$$

where $\mathbf{y}(t) = [\mathbf{y}_1(t), \ldots, \mathbf{y}_n(t)]^T \in \mathbb{R}^n$ is the state variable iwith initial condition $\mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^n, t$ is time, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a constant matrix, and $\mathbf{F}$ is a nonlinear vector-valued function evaluated at $\mathbf{y}(t)$ componentwise. This type of system often arises from the discretization of nonlinear PDEs. The dimension n, which is the number of unknowns of this system, is generally required to be very large to improve the accuracy. We call (2.1) as original full-order system or full-order system of dimension n, which will be costly to compute in pratice. We can reduce the computational complexity and simulation time by using the following methods.

2.1. Projection-Based Model Order Reduction

Projection-based method can construct a reduced-order system by projecting (2.1) onto a low dimensional subspace. Let $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ be a matrix whose columns form a set of an orthonormal basis of dimension k, where $k < n$. Then, we can approximate the state variable $\mathbf{y}(t)$ in the space spanned by the columns of $\mathbf{V}_k$ in the form of (2.2)

$$\mathbf{y}(t) \approx \mathbf{V}_k\tilde{\mathbf{y}}(t), \tag{2.2}$$

where $\tilde{\mathbf{y}}(t) \in \mathbb{R}^k$. By substituting (2.2) into (2.1), we obtain the following reduced system with k unknows in $\tilde{\mathbf{y}}(t)$.

$$\frac{d}{dt}\mathbf{V}_k\tilde{\mathbf{y}}(t) = \mathbf{A}\mathbf{V}_k\tilde{\mathbf{y}}(t) + \mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)), \tag{2.3}$$

with initial condition

$$\mathbf{V}_k\tilde{\mathbf{y}}(0) = \mathbf{y}_0. \tag{2.4}$$

Then, applying the Galerkin projection which will give the smallest error of the residual in the direction of $span\{\mathbf{V}_k\}$. The POD reduced system is of the form:

$$\mathbf{V}_k^T\frac{d}{dt}\mathbf{V}_k\tilde{\mathbf{y}}(t) = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k\tilde{\mathbf{y}}(t) + \mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)), \tag{2.5}$$

and the initial condition (2.4) of the POD reduced system becomes

$$\mathbf{V}_k^T\mathbf{V}_k\tilde{\mathbf{y}}(0) = \mathbf{V}_k^T\mathbf{y}_0. \tag{2.6}$$

Since $\mathbf{V}_k^T\mathbf{V}_k = \mathbf{I}_k$, (2.3) and (2.4) can be written as

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \underbrace{\mathbf{V}_k^T\mathbf{A}\mathbf{V}_k}_{\tilde{\mathbf{A}}}\tilde{\mathbf{y}}(t) + \mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)), \tag{2.7}$$

$$\tilde{\mathbf{y}}(0) = \mathbf{V}_k^T\mathbf{y}_0, \tag{2.8}$$

where $\tilde{\mathbf{A}} = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k \in \mathbb{R}^{k\times k}$ can be precomputed because it does not depend on time and (2.7) is called POD reduced system. In this setting, $\mathbf{V}_k$ can be obtained from any orthogonal basis. However, to get a good approximation from this reduced system, we will consider the basis constructed by Proper Orthogonal Decomposition (POD) which will be described in Section 2.2.

2.2. Proper Orthogonal Decomposition (POD)

This section considers the procedure of POD. In 1937, John Lumley initially proposed POD in the context of inhomogeneous structure turbulent flows [44] and stochastic tools in turbulence (1970) [43]. POD is also known by other names, for example, Karhunen-Love decomposition (KLD), Principal Component Analysis (PCA), or Singular Value Decomposition (SVD). POD has been used in many applications, e.g. [8, 30, 39, 40, 60, 72]. We will next consider construct a low dimensional by using POD with the Galerkin projection.

POD Basis

The aim of POD is to construct a set of global basis functions by extracting basis that describes the main dynamics from the system of interest, which can be obtained by the singular value decomposition (SVD) of solutions or snapshots:$\mathbf{Y} \in \mathbb{R}^{n\times n_s}$. The singular value decomposition of a rectangular matrix $\mathbf{Y} \in \mathbb{R}^{n\times n_s}$ is given by the following theorem.

Theorem 2.1 (Singular value decomposition,[28]). *Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}] \in \mathbb{R}^{n\times n_s}$ be a snapshot matrix of rank r with $\mathbf{y}_i \cong y(t_i), t_i \in \mathcal{I}, i = 1, \ldots, n_s$. Then there exists a decomposition of the form*

$$\mathbf{Y} = \hat{\mathbf{U}} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} \mathbf{Z}^T = \hat{\mathbf{U}}\Sigma\mathbf{Z}^T \tag{2.9}$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{n\times r}$ and $\mathbf{Z} \in \mathbb{R}^{n_s\times r}$ are orthogonal matrices and $\Sigma = diag(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$. The columns in $\hat{\mathbf{U}} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ are called the (left) singular vectors of $\mathbf{Y}$ and for the singular values σ_i it holds: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

Proof. The formal derivation can, for example, be found in [28]. ■

Notice that, for the SVD of $\mathbf{Y} = \hat{\mathbf{U}}\Sigma\mathbf{Z}^T$, the following diagonalization holds

$$\mathbf{Y}\mathbf{Y}^T = (\hat{\mathbf{U}}\Sigma\mathbf{Z}^T)(\hat{\mathbf{U}}\Sigma\mathbf{Z}^T)^T = \hat{\mathbf{U}}\Sigma^2\hat{\mathbf{U}}^T, \tag{2.10}$$

and therfore, the columns of $\hat{\mathbf{U}}$ are eigenvectors of the matrix $\mathbf{Y}\mathbf{Y}^T$ with corresponding eigenvalues $\lambda_i = \sigma_i^2 > 0, i = 1, \ldots, r$. The calculation of POD basis can be done by using the following steps.

Algorithm 1 Algorithm for constructing POD basis

- INPUT : Snapshot vectors $\{\mathbf{y}_j\}_{j=1}^{n_s} \subset \mathbb{R}^n$
- OUTPUT : POD basis matrix $\mathbf{V}_k \in \mathbb{R}^{n\times k}$

1. Create snapshot matrix : $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}] \in \mathbb{R}^{n\times n_s}$and let $r = rank(\mathbf{Y})$
2. Compute SVD: $\mathbf{Y} = \mathbf{V}\Sigma\mathbf{W}^T$ and choose dimension $k \leq r$
3. POD basis of rank k : $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k] = \mathbf{V}(:, 1 : k)$

From POD basis Algorithm 1, we first create the snapshots which are the solutions in the different time steps. Then, we find POD basis $\mathbf{V}_k$ from the snapshots by using SVD or POD. Likewise, in the case of nonlinear term, we find POD basis from Algorithm 1.

One of the most important properties of POD is that it can construct an approximation that minimizes the error in 2−norm for a given fixed basis rank. More details on this will be discussed next in Section 2.3.

2.3. POD Error

We have presented the computation for a POD basis by using SVD. Alternately, it can be shown that the POD basis matrix $\mathbf{V}_k$ is the solution to the following optimization problem (2.13)

Theorem 2.2 (POD basis,[68]). *Let* $\mathbf{Y} \in \mathbb{R}^{n \times n_s}$ *be a snapshot matrix* $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_{n_s}]$ *with rank* $r \leq min\{n, n_s\}$*. Further, let* $\mathbf{Y} = \hat{\mathbf{U}}\Sigma\mathbf{Z}^T$ *be the singular value decomposition of* $\mathbf{Y}$ *with orthogonal matrices* $\hat{\mathbf{U}} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ *and* $\hat{\mathbf{Z}} = [\mathbf{z}_1, \dots, \mathbf{z}_{n_s}]$ *as in* (2.9)*. Then, for any* $\ell \in \{1, \dots, r\}$ *the solution to the optimization problem*

$$\max_{\varphi_i,\dots,\varphi_\ell} \sum_{i=1}^{\ell}\sum_{j=1}^{n_s} \|\langle \mathbf{y}_j, \varphi_i\rangle\|^2 \tag{2.11}$$

$$s.t. \quad \langle \varphi_i, \varphi_j\rangle = \varphi_i^T, \varphi_j = \delta_{i,j} = \begin{cases} 1, & if \quad i \neq j \\ 0, & if \quad i = j \end{cases}, \quad for \quad 1 \leq i, j \leq \ell \tag{2.12}$$

is given by the left singular vectors $\{\mathbf{u}_i\}_{i=1}^{\ell}$*. The set of vectors* $\varphi_1, \dots, \varphi_\ell$ *are called the POD basis of rank* ℓ*. Here,* $\delta_{i,j}$ *denotes the Kronecker delta.*

Proof. The proof is given in [[68], p. 5-6] ∎

Moreover, the optimization problem (2.13) will have minimum error when the POD basis approximation $\mathbf{y}_j \cong \mathbf{V}_k\mathbf{V}_k^T\mathbf{y}_j$ is used for $j = 1, \dots, n_s$. This error is given by

$$\sum_{j=1}^{n_s} \|\mathbf{y}_j - \mathbf{V}_k\mathbf{V}_k^T\mathbf{y}_j\|_2^2 = \sum_{\ell=k+1}^{r} \sigma_\ell^2, \tag{2.13}$$

which is the sum of the neglected singular values $\sigma_{k+1}, ..., \sigma_r$ from SVD of $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_{n_s}]$. We can proof by the method of low-rank approximation which can be found in [59].

Although we use POD to reduce the number of unknowns of the full-order system and POD can reduce the large dimension of linear term, it cannot reduce computational complexity for nonlinear term, which may still depend on the full dimension such as when computing the term $\mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))$ from (2.7). For this reason, we will combine POD aprroximation with Discrete Empirical Interpolation Method (DEIM), which will be described in Section 2.4.

2.4. Discrete Empirical Interpolation Method (DEIM)

This section considers the nonlinear term $\mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))$ in (2.7). Notice that the computational complexity for evaluating this term still depends on the full dimension n. To eliminate this dependence, we combine POD approximation with the Discrete Empirical Interpolation method (DEIM), which is recently proposed in [6, 14]. DEIM has been used in many applications [1, 13, 15, 18, 37] discussed earlier. We first consider $\mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))$ in the form

$$\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)) = \mathbf{f}(t) \tag{2.14}$$

and

$$\mathbf{N}(t) = \mathbf{V}_k^T \mathbf{F}(\mathbf{V}_k \tilde{\mathbf{y}}(t)). \tag{2.15}$$

Estimate (2.14) by projecting $\mathbf{f}(t)$ onto subspace $span\{\mathbf{U}\}$ of the form

$$\mathbf{f}(t) \approx \mathbf{U}\mathbf{c}(t) \tag{2.16}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$ is the projection basis with $m \ll n$. The basis matrix $\mathbf{U}$ can be found by using SVD of $[\mathbf{F}(\mathbf{y}_1), \ldots, \mathbf{F}(\mathbf{y}_{n_s})], \mathbf{y}_i \cong \mathbf{y}(t_i)$. Then we can calculate $\mathbf{c}(t)$ from the following interpolation method. First, consider matrix

$$\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m} \tag{2.17}$$

where $\mathbf{e}_{\wp_i} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^n$ is the $\wp_i$-th column of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, for $i = 1, \ldots, m$, for selecting m rows of $\mathbf{U}$. Then assume that $\mathbf{P}^T\mathbf{U}$ is nonsingular and solve for $\mathbf{c}(t)$ from

$$\mathbf{P}^T\mathbf{f}(t) = (\mathbf{P}^T\mathbf{U})\mathbf{c}(t) \tag{2.18}$$

so,

$$\mathbf{c}(t) = (\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(t). \tag{2.19}$$

Finally, the approximation is given by

$$\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{Y}}(t)) = \mathbf{f}(t) \approx \mathbf{U}\mathbf{c}(t) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\underbrace{\mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))}_{m\times 1}. \tag{2.20}$$

In the case when the nonlinear function $\mathbf{F}$ is componentwise, we have

$$\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{Y}}(t)) = \mathbf{f}(t) \approx \mathbf{U}\mathbf{c}(t) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\underbrace{\mathbf{F}(\mathbf{P}^T\mathbf{V}_k\tilde{\mathbf{y}}(t))}_{m\times 1}. \tag{2.21}$$

Discrete Empirical Interpolation Method (DEIM) estimates nonlinear term by finding projection basis from POD and selecting the interpolation indices by a greedy algorithm The interpolated indices $\wp_1, \ldots, \wp_m$, can be obtained form the following DEIM algorithm [6].

Algorithm 2 DEIM

- INPUT : $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ linearly independent
- OUTPUT : $\vec{\wp} = [\wp_1, \ldots, \wp_m]^T \in \mathbb{R}^m$

1. $[|\rho|, \wp_1] = \max\{|\mathbf{u}_1|\}$
2. $\bar{\mathbf{V}} = [\bar{\mathbf{v}}_1]$, $\bar{\mathbf{P}} = [\mathbf{e}_{\wp_1}]$, $\vec{\wp} = [\wp_1]$
3. **for** $\ell = 2 : m$ **do**
    - Solve $(\mathbf{P}^T\mathbf{U})\mathbf{c} = \mathbf{P}^T\mathbf{u}_\ell$;
    - $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$
    - $[|\rho|, \wp_j] = \max\{|\mathbf{r}|\}$
    - $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_\ell], \mathbf{P} \leftarrow [\mathbf{P} \quad \mathbf{e}_{\wp_\ell}], \vec{\wp} \leftarrow \begin{bmatrix} \vec{\wp} \\ \wp_\ell \end{bmatrix}$

   end for

The aim of DEIM algorithm is to select the interpolation indices so that the approximation has smallest error $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$ in each interation ℓ. The procedure of DEIM Algorithm 2 is as follows: First, we start with a basis of rank m, which can be obtained by using POD of nonlinear term. Then, select the index of a component in the first basis vector $\mathbf{u}_1$ with the largest absolute value. Next, we select the other indices so that we have minimum residual error $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$ in each step.

2.5. DEIM Error

The general form of DEIM approximation is summarized in Definition 2.3 and the corresponding error of DEIM proposed in [14] is shown in Theorem 2.4. The extension of this error bound of to the state-space error estimate can be found in [16]. We will extend this error analysis in the next section.

Definition 2.3 (DEIM approximation, [14])**.** Let $\mathbf{f} : \mathscr{D} \mapsto \mathbb{R}^n$ be a nonlinear vector-valued function with $\mathscr{D} \subset \mathbb{R}^d$ for some positive integer d. Let $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ be a linearly independent set for $\ell \in \{1, \ldots, m\}$. For $t \in \mathscr{D}$, the DEIM approximation of order m for $\mathbf{f}(t)$ in the space spanned by $\{\mathbf{u}_\ell\}_{\ell=1}^m$ is given by

$$\hat{\mathbf{f}}(t) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(t), \tag{2.22}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m}$, with $\{\wp_1, \ldots, \wp_m\}$ being the output from Algorithm 2 with the input basis $\{\mathbf{u}_i\}_{i=1}^m$.

Theorem 2.4 (Error bound of DEIM approximation, [14])**.** *Let $\mathbf{f} \in \mathbb{R}^n$ be arbitrary vector. Let $\{\mathbf{u}_\ell\}_{\ell=1}^m \subset \mathbb{R}^n$ be a given orthonormal set of vectors. From Definition 3.1, the DEIM approximation of order $m \le n$ for $\mathbf{f}$ in the space spanned by $\{\mathbf{u}_\ell\}_{\ell=1}^m$ is $\hat{\mathbf{f}} = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m}$, with $\{\wp_1, \ldots, \wp_m\}$ being the output from Algorithm 2.2 with the input basis $\{\mathbf{u}_i\}_{i=1}^m$. An error bound for $\hat{\mathbf{f}}$ is then given by*

$$\|\mathbf{f} - \hat{\mathbf{f}}\|_2 \le \mathcal{C}\mathscr{E}_*(\mathbf{f}), \tag{2.23}$$

where

$$\mathcal{C} = \|(\mathbf{P}^T\mathbf{U})^{-1}\|_2 \quad \text{and} \quad \mathscr{E}_*(\mathbf{f}) = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}\|_2 \tag{2.24}$$

is the error of the best 2-norm approximation for $\mathbf{F}$ from the space Range$(\mathbf{U})$.

The constant $\mathcal{C}$ is bounded [16] by

$$\mathcal{C} \le \frac{(1+\sqrt{2n})^{m-1}}{\|\mathbf{e}_{\wp_1}^T\mathbf{U}_1\|_2} = (1+\sqrt{2n})^{m-1}\|\mathbf{u}_1\|_\infty^{-1}. \tag{2.25}$$

Note that Theorem 2.4 gives the error bound for the approximation obtained by DEIM algorithm 2, but the proof established that the choice of the DEIM indices algorithm 2 minimizes $\|(\mathbf{P}^T\mathbf{U})^{-1}\|_2$. Therefore, the approximation error is minimized in each iteration of Algorithm 2. This work compares the theoretical error bounds in Theorem 2.4 in [[14], lemma 3.2] with the exact error in Section 5.

Let $\tilde{\mathbf{f}}(t)$ be the interpolation apprximation of $\mathbf{f}(t)$ in the form

$$\tilde{\mathbf{f}}(t) = \mathbf{U}\mathbf{c}(t) = \mathbb{P}\mathbf{f}(t), \tag{2.26}$$

where projection $\mathbb{P} = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T$. Therefore, $\mathbf{N}(t) = \mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))$ in (2.15) can be written in the form

$$\mathbf{N}(t) \approx \underbrace{\mathbf{V}_k^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}}_{\text{precomputed}:k\times m}\underbrace{\mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))}_{m\times 1} \tag{2.27}$$

and in the caes that $\mathbf{F}$ is componentwise, we have

$$\mathbf{N}(t) \approx \underbrace{\mathbf{V}_k^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}}_{\text{precomputed:}k\times m}\underbrace{\mathbf{F}(\mathbf{P}^T\mathbf{V}_k\tilde{\mathbf{y}}(t))}_{m\times 1} \tag{2.28}$$

where $\mathbf{V}_k^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}$ can be precomputed because there is no dependence on t. The resulting reduced system is given by

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{y}}(t) + \mathbb{E}\mathbf{F}_{\wp}(\tilde{\mathbf{y}}(t)) \tag{2.29}$$

where $\tilde{\mathbf{A}} = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k \in \mathbb{R}^{k\times k}, \mathbb{E} = \mathbf{V}_k^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1} \in \mathbb{R}^{k\times m}, \mathbf{F}_{\wp}(\tilde{\mathbf{y}}(t)) = \mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)) \in \mathbb{R}^{m\times 1}$, and $\tilde{\mathbf{A}}$ and $\mathbb{E}$ can be precomputed without the dependence on t.

Therefore, computing the nonlinear term in equation (2.29) has no dependence on n and this system is called POD-DEIM reduced system. Notice that this computational efficiency requires two parts in the approximation:

1. POD basis $\mathbf{U}$ which can be obtained by using POD of $\mathbb{F} = [\mathbf{F}(\mathbf{y}(t_1)), ..., \mathbf{F}(\mathbf{y}(t_{n_t}))]$.
2. Interpolate indices for selecting m rows by DEIM Algorithm 2.

3. State-Space Error Estimate for POD-DEIM Reduced System in Semi-Implicit Discretized Setting

For POD reduced systems, the state-space error bounds of can be found in [59]. This work extends the derivation in [[16], Section 4.2] to the discretized system from semi-implicit Euler numerical scheme for POD-DEIM reduced systems. Consider the following full-order and reduced-order systems:

$$\frac{d}{dt}\mathbf{y} = \mathbf{A}\mathbf{y} + \mathbf{g}(t) + \mathbf{F}(\mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0 \tag{3.1}$$

and

$$\frac{d}{dt}\tilde{\mathbf{y}} = \tilde{\mathbf{A}}\tilde{\mathbf{y}} + \tilde{\mathbf{g}}(t) + \tilde{\mathbf{F}}(\tilde{\mathbf{y}}(t)), \quad \tilde{\mathbf{y}}(0) = \mathbf{V}_k^T\mathbf{y}_0, \tag{3.2}$$

where $\tilde{\mathbf{A}} = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k$, $\tilde{\mathbf{g}}(t) = \mathbf{V}_k^T\mathbb{J}\mathbf{g}(t)$, $\tilde{\mathbf{F}}(\tilde{\mathbf{y}}(t)) = \mathbf{V}_k^T\mathbb{P}\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}})$, $\mathbb{J} = \mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1}\mathbf{J}^T$, $\mathbb{P} = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T$, (3.1) is the full-order system, and (3.2) is the POD-DEIM reduced system. Here, we discretized the system by using the semi-implicit Euler method for the full-order system, the POD-DEIM reduced system in the form: for $Y_0 = \mathbf{y}_0$ and $\tilde{Y}_0 = \mathbf{V}_k^T\mathbf{y}_0$,

$$\frac{Y_{j+1} - Y_j}{\Delta t} = \mathbf{A}Y_{j+1} + \mathbf{g}(t_j) + \mathbf{F}(Y_j), \tag{3.3}$$

$$\frac{\tilde{Y}_{j+1} - \tilde{Y}_j}{\Delta t} = \tilde{\mathbf{A}}\tilde{Y}_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\tilde{Y}_j), \tag{3.4}$$

for $j = 0, 1, \ldots, n_t$, where $n_t = \frac{T}{\Delta t}$ is the number of time steps with time steps Δt. Let $M[\mathbf{F}] = \sup\limits_{\mathbf{u}\neq\mathbf{v}} \frac{\left\langle \mathbf{u}-\mathbf{v}, \mathbf{F}(\mathbf{u})-\mathbf{F}(\mathbf{v})\right\rangle}{\left\|\mathbf{u}-\mathbf{v}\right\|^2}$ be the least upper bound (lub) logarithmic Lipschitz constant with respect to the inner product $\langle\cdot,\cdot\rangle$, $\langle\mathbf{u},\mathbf{v}\rangle = \mathbf{u}^T\mathbf{v}$, and $\|\mathbf{u}\| = \sqrt{\langle\mathbf{u},\mathbf{u}\rangle}$. The approximation of $\mathbf{y}(t_{j+1})$ and $\tilde{\mathbf{y}}(t_{j+1})$ are Y_{j+1} and $\tilde{Y}_{j+1}$, respectively. Therefore, the error is

$$E_{j+1} = Y_{j+1} - \mathbf{V}_k\tilde{Y}_{j+1}, \tag{3.5}$$

where Y_{j+1} is the solution of full system and $\tilde{Y}_{j+1}$ is the solution of the POD-DEIM reduced system in (3.3) for $j+1 = 1,\ldots,n_t$.

We can write

$$E_{j+1} = \underbrace{Y_{j+1} - \mathbf{V}_k\mathbf{V}_k^T Y_{j+1}}_{\rho_{j+1}} + \underbrace{\mathbf{V}_k\mathbf{V}_k^T Y_{j+1} - \mathbf{V}_k\tilde{Y}_{j+1}}_{\theta_{j+1}} \tag{3.6}$$

$$E_{j+1} = \rho_{j+1} + \theta_{j+1} \tag{3.7}$$

where $\rho_{j+1} = Y_{j+1} - \mathbf{V}_k\mathbf{V}_k^T Y_{j+1}$, $\theta_{j+1} = \mathbf{V}_k\mathbf{V}_k^T Y_{j+1} - \mathbf{V}_k\tilde{Y}_{j+1}$. Since $\rho_{j+1}^T\theta_{j+1} = 0$, $\|\mathbf{E}_{j+1}\|_2^2 = \|\rho_{j+1}\|_2^2 + \|\theta_{j+1}\|_2^2$. Let $\|\tilde{\theta}_{j+1}\| = \mathbf{V}_k^T\|\theta_{j+1}\|$.
From $\theta_{j+1} = \mathbf{V}_k\mathbf{V}_k^T Y_{j+1} - \mathbf{V}_k\tilde{Y}_{j+1}$, we have $\theta_{j+1} = \mathbf{V}_k\tilde{\theta}_{j+1}$ and it can be shown that $\tilde{\theta}_{j+1} = \mathbf{V}_k^T\theta_{j+1}$. That is,

$$\tilde{\theta}_{j+1} = \mathbf{V}_k^T(\mathbf{V}_k\mathbf{V}_k^T Y_{j+1} - \mathbf{V}_k\tilde{Y}_{j+1}) = \mathbf{V}_k^T Y_{j+1} - \tilde{Y}_{j+1}, \tag{3.8}$$

and therefore, for all $j = 0, 1, ..., n_t$,

$$\tilde{\theta}_j = \mathbf{V}_k^T Y_j - \tilde{Y}_j, \tag{3.9}$$

for $\mathbf{V}_k^T\mathbf{V}_k = \mathbf{I}$. Using (3.9) with (3.1) and (3.2) gives

$$\begin{aligned}
\tilde{\theta}_{j+1} - \tilde{\theta}_j &= \Big[\mathbf{V}_k^T Y_{j+1} - \tilde{Y}_{j+1}\Big] - \Big[\mathbf{V}_k^T Y_j - \tilde{Y}_j\Big]\\
\frac{\tilde{\theta}_{j+1} - \tilde{\theta}_j}{\Delta t} &= \mathbf{V}_k^T\Big(\frac{Y_{j+1} - Y_j}{\Delta t}\Big) - \Big(\frac{\tilde{Y}_{j+1} - \tilde{Y}_j}{\Delta t}\Big)\\
&= \mathbf{V}_k^T\Big[\mathbf{A}Y^{j+1} + \mathbf{g}(t_j) + \mathbf{F}(Y^j)\Big] - \Big[\tilde{\mathbf{A}}\tilde{Y}^{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\tilde{Y}^j)\Big].
\end{aligned}$$

Consider

$$\begin{aligned}
\mathbf{V}_k^T\Big(\frac{Y_{j+1} - Y_j}{\Delta t}\Big) =& \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j)\\
&+ \underbrace{\mathbf{V}_k^T\Big[\mathbf{A}Y_{j+1} + \mathbf{g}(t_j) + \mathbf{F}(Y_j)\Big] - \Big[\tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j)\Big]}_{\tilde{R}_{j+1}}\\
=& \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) + \tilde{R}_{j+1}.
\end{aligned}$$

Therefore,

$$\frac{\tilde{\theta}_{j+1} - \tilde{\theta}_j}{\Delta t} = \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \Big[\tilde{\mathbf{A}}\tilde{Y}_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\tilde{Y}_j)\Big] + \tilde{R}_{j+1}. \tag{3.10}$$

Since $\|\tilde{\theta}_{j+1}\|^2 = \langle\theta_{j+1}, \theta_{j+1}\rangle$, then $\frac{\|\tilde{\theta}_{j+1}\|}{\Delta t} = \frac{\langle\tilde{\theta}_{j+1}, \tilde{\theta}_{j+1}\rangle}{\Delta t\|\tilde{\theta}_{j+1}\|}$
and from the CauchySchwarz inequality, we have $\langle\tilde{\theta}_{j+1}, \tilde{\theta}_j\rangle \leq \|\tilde{\theta}_{j+1}\|\|\tilde{\theta}_j\|$.
That is,

$$\frac{\|\tilde{\theta}_j\|}{\Delta t} \geq \frac{\langle\tilde{\theta}_{j+1}, \tilde{\theta}_j\rangle}{\Delta t\|\tilde{\theta}_{j+1}\|}. \tag{3.11}$$

Therefore,

$$
\begin{aligned}
\frac{\|\tilde{\theta}_{j+1}\| - \|\tilde{\theta}_j\|}{\Delta t} &\leq \frac{1}{\Delta t}\Big[\frac{\langle \tilde{\theta}_{j+1}, \tilde{\theta}_{j+1}\rangle}{\|\tilde{\theta}_{j+1}\|} - \frac{\langle \tilde{\theta}_{j+1}, \tilde{\theta}_{j}\rangle}{\|\tilde{\theta}_{j+1}\|}\Big] \\
&= \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \frac{\tilde{\theta}_{j+1} - \tilde{\theta}_j}{\Delta t}\Big\rangle \\
&= \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \Big[\tilde{\mathbf{A}}\tilde{Y}_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\tilde{Y}_j)\Big] + \tilde{R}_{j+1}\Big\rangle \\
&= \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \Big[\tilde{\mathbf{A}}\tilde{Y}_{j+1} + \tilde{\mathbf{g}}(t_j) + \tilde{\mathbf{F}}(\tilde{Y}_j)\Big]\Big\rangle \\
&+ \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{R}_{j+1}\Big\rangle.
\end{aligned}
$$

From [[16], Section 4], by defining $M[\mathbf{F}] = \sup_{\mathbf{u}\neq\mathbf{v}} \frac{\langle \mathbf{u}-\mathbf{v}, \mathbf{F}(\mathbf{u})-\mathbf{F}(\mathbf{v})\rangle}{\|\mathbf{u}-\mathbf{v}\|^2}$, and using $\mathbf{u} = \mathbf{V}_k^T Y_{j+1}$ and $\mathbf{v} = \tilde{Y}_{j+1}$, we have

$$
\begin{aligned}
\frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} - \tilde{\mathbf{A}}\tilde{Y}_{j+1}\Big\rangle &+ \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \tilde{\mathbf{F}}(\tilde{Y}_j)\Big\rangle \\
&\leq M[\tilde{\mathbf{A}}]\frac{\|\tilde{\theta}_{j+1}\|^2}{\|\tilde{\theta}_{j+1}\|} + \frac{\|\tilde{\theta}_{j+1}\|}{\|\tilde{\theta}_{j+1}\|}\|\tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \tilde{\mathbf{F}}(\tilde{Y}_j)\|. \qquad (3.12)
\end{aligned}
$$

The Lipschitz continuity of $\mathbf{F}$ implies $\|\tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \tilde{\mathbf{F}}(\tilde{Y}_j)\| \leq L_f \underbrace{\|\mathbf{V}_k^T Y_j - \tilde{Y}_j\|}_{\|\tilde{\theta}_j\|}$, so that

$$
\begin{aligned}
\frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{A}}\mathbf{V}_k^T Y_{j+1} - \tilde{\mathbf{A}}\tilde{Y}_{j+1}\Big\rangle &+ \frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_j) - \tilde{\mathbf{F}}(\tilde{Y}_j)\Big\rangle \\
&\leq M[\tilde{\mathbf{A}}]\|\tilde{\theta}_{j+1}\| + L_f\|\tilde{\theta}_j\| \qquad (3.13)
\end{aligned}
$$

and

$$
\frac{1}{\|\tilde{\theta}_{j+1}\|}\Big\langle \tilde{\theta}_{j+1}, \tilde{R}_{j+1}\Big\rangle \leq \frac{1}{\|\tilde{\theta}_{j+1}\|}\|\tilde{\theta}_{j+1}\|\|\tilde{R}_{j+1}\| = \|\tilde{R}_{j+1}\|. \qquad (3.14)
$$

Therefore,

$$
\frac{\|\tilde{\theta}_{j+1}\| - \|\tilde{\theta}_j\|}{\Delta t} \leq M[\tilde{\mathbf{A}}]\|\tilde{\theta}_{j+1}\| + L_f\|\tilde{\theta}_j\| + \|\tilde{R}_{j+1}\|. \qquad (3.15)
$$

From $\theta_{j+1} = \mathbf{V}_k\tilde{\theta}_{j+1}$, we have $\|\theta_{j+1}\| = \sqrt{(\mathbf{V}_k\tilde{\theta}_{j+1})^T(\mathbf{V}_k\tilde{\theta}_{j+1})} = \|\tilde{\theta}_{j+1}\|$ and

$$\begin{aligned}
\frac{\|\tilde{\theta}_{j+1}\| - \|\tilde{\theta}_j\|}{\Delta t} &\leq M[\tilde{\mathbf{A}}]\|\tilde{\theta}_{j+1}\| + L_f\|\tilde{\theta}_j\| + \|\tilde{R}_{j+1}\| \\
\|\tilde{\theta}_{j+1}\| - \|\tilde{\theta}_j\| &\leq \Delta t M[\tilde{\mathbf{A}}]\|\tilde{\theta}_{j+1}\| + \Delta t L_f\|\tilde{\theta}_j\| + \Delta t\|\tilde{R}_{j+1}\| \\
\|\tilde{\theta}_{j+1}\| - \Delta t M[\tilde{\mathbf{A}}]\|\tilde{\theta}_{j+1}\| &\leq \|\tilde{\theta}_j\| + \Delta t L_f\|\tilde{\theta}_j\| + \Delta t\|\tilde{R}_{j+1}\| \\
\Big(1 - \Delta t M[\tilde{\mathbf{A}}]\Big)\|\tilde{\theta}_{j+1}\| &\leq \Big(1 + \Delta t L_f\Big)\|\tilde{\theta}_j\| + \Delta t\|\tilde{R}_{j+1}\| \\
\|\tilde{\theta}_{j+1}\| &\leq \frac{1}{1 - \Delta t M[\tilde{\mathbf{A}}]}\Big[\Big(1 + \Delta t L_f\Big)\|\tilde{\theta}_j\| + \Delta t\|\tilde{R}_{j+1}\|\Big],
\end{aligned}$$

or

$$\|\tilde{\theta}_{j+1}\| \leq \zeta\Big(\hat{\zeta}\|\tilde{\theta}_j\| + \Delta t\|\tilde{R}_{j+1}\|\Big), \tag{3.16}$$

where $\zeta = \frac{1}{1-\Delta t M[\tilde{\mathbf{A}}]}$ and $\hat{\zeta} = 1 + \Delta t L_f$. Note that $\theta_0 = 0$, since

$$\theta_0 = \mathbf{V}_k\mathbf{V}_k^T Y_0 - \mathbf{V}_k\tilde{Y}_0 = \mathbf{V}_k\Big[\underbrace{\mathbf{V}_k^T Y_0 - \tilde{Y}_0}_{\mathbf{V}_k^T Y_0 = \tilde{Y}_0}\Big] = 0.$$

Therefore, from (3.16),

$$\|\tilde{\theta}_j\| \leq \zeta\Big(\hat{\zeta}\|\tilde{\theta}_{j-1}\| + \Delta t\|\tilde{R}_j\|\Big) \leq \zeta^j\|\theta_0\| + \Delta t\sum_{\ell=1}^{j}\zeta^\ell\hat{\zeta}^{\ell-1}\|\tilde{R}_{j-\ell+1}\| \leq \Delta t\Big(q_i\sum_{\ell=1}^{j}\|\tilde{R}_\ell\|^2\Big)^{1/2},$$

where $q_j = \sum_{\ell=1}^{j}\zeta^{2\ell}\hat{\zeta}^{2(\ell-1)}$.

The term $\|\tilde{R}_\ell\|$ will be written as a sum of differences that can be estimated using the neglected singular values.

$$\begin{aligned}
\tilde{R}_\ell &= \mathbf{V}_k^T\Big[\mathbf{A}Y_\ell + \mathbf{g}(t_\ell) + \mathbf{F}(Y_\ell)\Big] - \Big[\tilde{\mathbf{A}}\mathbf{V}_k^T Y_\ell + \tilde{\mathbf{g}}(t_\ell) + \tilde{\mathbf{F}}(\mathbf{V}_k^T Y_\ell)\Big] \\
&= \mathbf{V}_k^T\Big[\mathbf{A}Y_\ell + \mathbf{g}(t_\ell) + \mathbf{F}(Y_\ell)\Big] - \Big[\tilde{\mathbf{A}}\mathbf{V}_k^T Y_\ell + \tilde{\mathbf{g}}(t_\ell) + \mathbf{V}_k^T\mathbb{P}\mathbf{F}(\mathbf{V}_k\mathbf{V}_k^T Y_\ell)\Big] \\
&= \mathbf{V}_k^T\Big[\mathbf{A}Y_\ell + \mathbf{g}(t_\ell) + \mathbf{F}(Y_\ell)\Big] - \mathbf{V}_k^T\mathbb{P}\mathbf{F}(Y_\ell) + \mathbf{V}_k^T\mathbb{P}\mathbf{F}(Y_\ell) \\
&\quad - \Big[\tilde{\mathbf{A}}\mathbf{V}_k^T Y_\ell + \tilde{\mathbf{g}}(t_\ell) + \mathbf{V}_k^T\mathbb{P}\mathbf{F}(\mathbf{V}_k\mathbf{V}_k^T Y_\ell)\Big] \\
&= \mathbf{V}_k^T\Big[\mathbf{A}Y_\ell + \mathbf{g}(t_\ell)\Big] - \Big[\tilde{\mathbf{A}}\mathbf{V}_k^T Y_\ell + \tilde{\mathbf{g}}(t_\ell)\Big] \\
&\quad + \mathbf{V}_k^T\Big[\mathbf{F}(Y_\ell) - \underbrace{\mathbb{P}\mathbf{F}(Y_\ell)}_{\hat{\mathbf{F}}}\Big] + \mathbf{V}_k^T\mathbb{P}\Big[\mathbf{F}(Y_\ell) - \mathbf{F}(\mathbf{V}_k\mathbf{V}_k^T Y_\ell)\Big] \\
&= \Big[\mathbf{V}_k^T\mathbf{A}Y_\ell - \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k\mathbf{V}_k^T Y_\ell\Big] + \mathbf{V}_k^T\Big[\mathbf{g}(t_\ell) - \mathbb{J}\mathbf{g}(t_\ell)\Big] + \mathbf{V}_k^T\Big[\mathbf{F}(Y_\ell) - \mathbb{P}\mathbf{F}(Y_\ell)\Big] \\
&\quad + \mathbf{V}_k^T\mathbb{P}\Big[\mathbf{F}(Y_\ell) - \mathbf{F}(\mathbf{V}_k\mathbf{V}_k^T Y_\ell)\Big] \\
&= \mathbf{V}_k^T\mathbf{A}\Big(Y_\ell - \mathbf{V}_k\mathbf{V}_k^T Y_\ell\Big) + \mathbf{V}_k^T\Big[\mathbf{g}(t_\ell) - \mathbb{J}\mathbf{g}(t_\ell)\Big] + \mathbf{V}_k^T\Big(\mathbf{F}(Y_\ell) - \hat{\mathbf{F}}\Big) \\
&\quad + \mathbf{V}_k^T\mathbb{P}\Big[\mathbf{F}(Y_\ell) - \mathbf{F}(\mathbf{V}_k\mathbf{V}_k^T Y_\ell)\Big].
\end{aligned}$$

Since $\mathbf{z}_\ell = (\mathbf{I} - \mathbf{W}\mathbf{W}^T)\mathbf{g}(t_\ell)$ and $\mathbf{w}_\ell = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{F}(Y_\ell)$ from Theorem 2.4, we have

$$\tilde{R}_\ell = \mathbf{V}_k^T\mathbf{A}\Big(Y_\ell - \mathbf{V}_k\mathbf{V}_k^TY_\ell\Big) + \mathbf{V}_k^T(\mathbf{I} - \mathbb{J})\mathbf{z}_\ell + \mathbf{V}_k^T(\mathbf{I} - \mathbb{P})\mathbf{w}_\ell + \mathbf{V}_k^T\mathbb{P}\Big[\mathbf{F}(Y_\ell) - \mathbf{F}(\mathbf{V}_k\mathbf{V}_k^TY_\ell)\Big] \quad (3.17)$$

The Lipschitz continuity of $\mathbf{F}$ implies $\|\mathbf{F}(Y_\ell) - \mathbf{F}(\mathbf{V}_k\mathbf{V}_k^TY_\ell)\| \le L_f\|Y_\ell - \mathbf{V}_k\mathbf{V}_k^TY_\ell\| = L_f\|\rho_\ell\|$, so that the norm associated with an inner product satisfies the triangle inequality

$$\|\tilde{R}_\ell\| \le \gamma\|\rho_\ell\| + \omega\|\mathbf{z}_\ell\| + \beta\|\mathbf{w}_\ell\| + \alpha\|\rho_\ell\| = (\gamma+\alpha)\|\rho_\ell\| + \omega\|\mathbf{z}_\ell\| + \beta\|\mathbf{w}_\ell\|, \quad (3.18)$$

where $\gamma = \|\mathbf{V}_k^T\mathbf{A}\|$, $\omega = \|\mathbf{V}_k^T(\mathbf{I} - \mathbb{J})\|$, $\alpha = \|\mathbf{V}_k^T\mathbb{P}\|L_f$, $\beta = \|\mathbf{V}_k^T(\mathbf{I} - \mathbb{P})\|$. From (3.16), since $\theta_0 = 0$. Then, for $j = 0, \dots, n_t$,

$$\|\theta_j\|^2 \;\le\; (\Delta t)^2 q_j\Big(\sum_{\ell=1}^{j}\|\tilde{R}_\ell\|^2\Big) \;\le\; (\Delta t)^2\bar{a}_M((\gamma+\alpha)^2\bar{\mathcal{E}}_\mathbf{y} + \omega^2\bar{\mathcal{E}}_\mathbf{g} + \beta^2\bar{\mathcal{E}}_\mathbf{f}),$$

where $\bar{a}_M = 2q_j$ and $\bar{\mathcal{E}}_\mathbf{y} = \sum_{\ell=1}^{j}\|\rho_\ell\|^2$, $\bar{\mathcal{E}}_\mathbf{g} = \sum_{\ell=1}^{j}\|\mathbf{z}_\ell\|^2$, $\bar{\mathcal{E}}_\mathbf{f} = \sum_{\ell=1}^{j}\|\mathbf{w}_\ell\|^2$, defined by

$$\bar{\mathcal{E}}_\mathbf{y} = \sum_{j=0}^{n_t}\|Y_j - \mathbf{V}_k\mathbf{V}_k^TY_j\|^2,\quad \bar{\mathcal{E}}_\mathbf{f} = \sum_{j=0}^{n_t}\|F_j - \mathbf{U}\mathbf{U}^TF_j\|^2,\quad \bar{\mathcal{E}}_\mathbf{g} = \sum_{j=0}^{n_t}\|g_j - \mathbf{W}\mathbf{W}^Tg_j\|^2 \quad (3.19)$$

where $\mathbf{f}(t) = \mathbf{F}(\mathbf{y}(t))$, $F_j = \mathbf{F}(Y_j)$, $g_j = \mathbf{g}(t_j)$, $\mathbf{V}_k$ is the POD basis of the snapshot set for $\mathbf{y}(t)$, $\mathbf{U}$ is the POD basis of nonlinear snapshots $\mathbf{f}(t) = \mathbf{F}(\mathbf{y}(t))$, and $\mathbf{W}$ is the POD basis of nonlinear snapshots $\mathbf{g}(t)$.

Finally, using $\sum_{\ell=0}^{n_t}\|E_\ell\|^2 = \sum_{\ell=0}^{n_t}\|\rho_\ell\|^2 + \sum_{\ell=0}^{n_t}\|\theta_\ell\|^2$ gives

$$\sum_{j=0}^{n_t}\|Y_j - \mathbf{V}_k\tilde{Y}_j\|^2 = \sum_{\ell=0}^{n_t}\|E_\ell\|^2 \le \bar{\mathcal{C}}(\bar{\mathcal{E}}_\mathbf{y} + \bar{\mathcal{E}}_\mathbf{f} + \bar{\mathcal{E}}_\mathbf{g}), \quad (3.20)$$

where $\bar{\mathcal{C}} = \max\{1 + \bar{a}_M\Delta t(\gamma+\alpha)^2T, \bar{a}_M\Delta t\beta^2T, \bar{a}_M\Delta t\omega^2T\}$ and for $T = n_t\Delta t$.

Therefore the norm of the error $\|E_j\|$ is bounded on $[0, T]$, that is $\|E_\ell\|^2 = \|\rho_\ell\|^2 + \|\theta_\ell\|^2$ gives the error bound $\sum_{j=0}^{n_t}\|Y_j - \mathbf{V}_k\tilde{Y}_j\|^2 \le \bar{\mathcal{C}}(\bar{\mathcal{E}}_\mathbf{y} + \bar{\mathcal{E}}_\mathbf{f} + \bar{\mathcal{E}}_\mathbf{g})$, which is summarized in the following theorem.

Theorem 3.1 (Error bound: Convergence of POD-DEIM reduced system)**.** *Let $\mathbf{y}(t)$ be the solution of the full-order system (3.1) and $\tilde{\mathbf{y}}(t)$ be the solution of the POD-DEIM reduced system (3.2), for $t \in [0, T]$. Let Y_j be the solutions of the dizcretized systems of (3.1) and $\tilde{Y}_j$ be the solutions of the dizcretized systems of (3.2) by using semi-implicit Euler, $\Delta t = \frac{T}{n_t}$. Let $M[\tilde{\mathbf{A}}] = \sup_{\mathbf{u}\neq\mathbf{v}}\frac{\langle \mathbf{u}-\mathbf{v}, \tilde{\mathbf{A}}(\mathbf{u})-\tilde{\mathbf{A}}(\mathbf{v})\rangle}{\|\mathbf{u}-\mathbf{v}\|^2}$, $\mathbf{u} = \mathbf{V}_k^TY_j$, and $\mathbf{v} = \tilde{Y}_j$ be the logarithmic Lipschitz constant of $\tilde{\mathbf{A}}$ and assume that $\mathbf{A}$ in (3.1) is Lipschitz continuous and $\mathbf{F}(\mathbf{y}(t))$ in (3.1) is Lipschitz continuous with Lipschitz constant L_f as $\|\mathbf{F}(\mathbf{y})_j - \mathbf{F}(\mathbf{y})_{j+1}\| \le L_f\|\mathbf{y}_j - \mathbf{y}_{j+1}\|$. Then*

$$\sum_{j=0}^{n_t}\|Y_j - \mathbf{V}_k\tilde{Y}_j\|^2 \le \bar{\mathcal{C}}(\bar{\mathcal{E}}_\mathbf{y} + \bar{\mathcal{E}}_\mathbf{f} + \bar{\mathcal{E}}_\mathbf{g}), \quad (3.21)$$

where

$$\begin{aligned}\bar{\mathcal{C}} &:= \max\{1+\bar{a}_M\Delta t(\gamma+\alpha)^2T, \bar{a}_M\Delta t\beta^2T, \bar{a}_M\Delta t\omega^2T\}, &(3.22)\\ \gamma &:= \|\mathbf{V}_k^T\mathbf{A}\|,\ \omega := \|\mathbf{V}_k^T(\mathbf{I}-\mathbb{J})\|,\ \alpha := \|\mathbf{V}_k^T\mathbb{P}\|L_f,\ \beta := \|\mathbf{V}_k^T(\mathbf{I}-\mathbb{P})\| &(3.23)\\ \bar{a}_M &:= 2q_j = 2\sum_{\ell=1}^{j}\zeta^{2\ell}\hat{\zeta}^{2(\ell-1)},\ \zeta := \frac{1}{1-\Delta t M[\tilde{\mathbf{A}}]},\ \hat{\zeta} := 1+\Delta t L_f, &(3.24)\end{aligned}$$

and $\bar{\mathcal{E}}_{\mathbf{y}}$, $\bar{\mathcal{E}}_{\mathbf{f}}$, $\bar{\mathcal{E}}_{\mathbf{g}}$ *are defined in* (3.19).

Remark 3.2. When $\mathbf{V}_k$, $\mathbf{U}$, $\mathbf{W}$ are POD basis matrices of snapshot sets $\{Y_j\}$, $\{F_j\}$, $\{g_j\}$, respectively, then, $\bar{\mathcal{E}}_{\mathbf{y}} = \sum_{\ell=k+1}^{r}\lambda_\ell$, $\bar{\mathcal{E}}_{\mathbf{f}} = \sum_{\ell=m+1}^{r_s}s_\ell$, $\bar{\mathcal{E}}_{\mathbf{g}} = \sum_{\ell=q+1}^{r_\nu}\nu_\ell$ where $\{\lambda_\ell\}_{\ell=1}^{r}$, $\{s_\ell\}_{\ell=1}^{r_s}$, and $\{\nu_\ell\}_{\ell=1}^{r_\nu}$ are eigenvalues of YY^T, FF^T, and gg^T, respectively.

In addition to the state space error estimate for POD-DEIM reduced system, there is an a-posteriori error estimate for POD-DEIM reduced system which can be found in the work by D. Wirtz, D. C. Sorensen and B. Haasdonk [73].

In the next section, we will show the application of POD-DEIM model reduction through Fisher's equation.

4. Applications of Model Reduction on Fisher's equation

This work uses Fisher's equation as the main model problem for nonlinear partial differential equation (PDE). Here, we will describe how to use the model reduction techniques via POD and DEIM for Fisher's equation. Fisher's equation can be written in the form:

$$\frac{\partial y}{\partial t} = \alpha y(1-y) + D\frac{\partial^2 y}{\partial x^2}, \tag{4.1}$$

where D is the diffusion coefficient (positive), α is the reactive factor (positive), t is time, x is the spatial location and $y = y(x,t)$ is the state variable (e.g., population density, particles of chemicals, of a bacteria colony, population of organisms) at location x and time t. The term $\alpha y(1-y)$ is also called logistic growth. In 1937, equation (4.1) was first proposed by Fisher [2] as a model for the spread of an advantageous gene in a population, and in the same year Kolmogoroff, Petrovsky and Piscounoff [3] also proposed the same equation of the form (4.1). As a result, (4.1) is called Fisher's equation, Fisher-KPP equation, or also known as the diffusional logistic equation which is the simplest equation with diffusion, growth and self-regulation of a species.

4.1. Non-Dimensionalization for Fisher's Equation

We can rescale the variables as in [47, 62]. Let $t^* = \alpha t$, $\quad x^* = x(\frac{\alpha}{D})^{1/2}$. We use the variables t^* and x^* to nondimensionalize the equation (4.1). Note that

$$x = \left(\frac{D}{\alpha}\right)^{1/2} x^*, \quad t = \frac{1}{\alpha}t^*. \tag{4.2}$$

By substituting (4.2) into each part of (4.1), we have

$$\frac{\partial y}{\partial t} = \alpha \frac{\partial y}{\partial t^*}, \tag{4.3}$$

$$\frac{\partial y}{\partial x} = \left(\frac{\alpha}{D}\right)^{1/2} \frac{\partial y}{\partial x^*}, \tag{4.4}$$

$$\begin{aligned} \frac{\partial^2 y}{\partial x^2} &= \frac{\partial}{\partial x}\left(\frac{\partial y}{\partial x}\right) = \frac{\partial}{\partial x}\left[\left(\frac{\alpha}{D}\right)^{1/2} \frac{\partial y}{\partial x^*}\right] \\ &= \left(\frac{\alpha}{D}\right)^{1/2} \frac{\partial}{\partial x^*}\left[\left(\frac{\alpha}{D}\right)^{1/2} \frac{\partial y}{\partial x^*}\right] = \frac{\alpha}{D}\frac{\partial^2 y}{\partial x^{*2}}. \end{aligned} \tag{4.5}$$

Substituting the above forms of $\frac{\partial y}{\partial t}$, $\frac{\partial^2 y}{\partial x^2}$, and $\frac{\partial^2 y}{\partial x^2}$ into (4.1) gives

$$\alpha \frac{\partial y}{\partial t^*} = \alpha y(1-y) + D\left(\frac{\alpha}{D}\right)\frac{\partial^2 y}{\partial x^{*2}} \tag{4.6}$$

After simplifying the above equation and dropping the superscript star notation, we have

$$\frac{\partial y}{\partial t} = y(1-y) + \frac{\partial^2 y}{\partial x^2}. \tag{4.7}$$

Fisher's equation was numerically solved by various methods such as pseudospectral method [50], finite volume method, the Sinc collocation method [4], the propagation properties of nonnegative and bounded solutions [63], a moving mesh method in cylindrical coordinates [54], and fully discrete finite element approximations [29]. Fisher's equation was also studied in 2D setting [58] and its critical wave speed has been investigated in [20]. Here, we consider initial value problem in the form of

$$\frac{\partial y}{\partial t} = y(1-y) + \frac{\partial^2 y}{\partial x^2} \tag{4.8}$$

with $x \in [-25, 50], t \in [0, 15]$, boundary condition $y(-25, t) = 1, y(50, t) = 0$ and initial condition $y(x, 0) = \begin{cases} 1 & ; x < -10 \\ \frac{1}{4} & ; 10 < x < 20 \\ 0 & ; \text{otherwise} \end{cases}$.

Notice that initial condition has the discontinuities at $-10, 10$, and 20. The initial and boundary conditions are obtained from [21].

4.2. Time Discretization

By using finite difference (FD) discetization, we have

$$\frac{dy_i}{dt} = y_i(1-y_i) + \frac{y_{i+1} - 2y_i + y_{i-1}}{(\Delta x)^2} \tag{4.9}$$

which can be written in the matrix form as in (2.1) in Section 2 as

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y} + \mathbf{g}(t) + \mathbf{F}(\mathbf{y}(t)) \tag{4.10}$$

where $\mathbf{A}$ is a constant matrix from FD discretization and $\mathbf{g}(t)$ is a vector-valued function defined on a variable $t \in \mathbb{R}$, or a constant vector obtained from the boundary conditions. When the model problem and the initial condition given in this section are used, we have

$$\mathbf{A} = \frac{1}{(\Delta x)^2}\begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad \mathbf{g}(t) = \frac{1}{(\Delta x)^2}\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n\times 1}.$$

$\mathbf{F}(\mathbf{y}(t))$ is the nonlinear function:

$$\mathbf{F}(\mathbf{y}(t)) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{bmatrix} .* \left\{ \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{bmatrix} \right\} \in \mathbb{R}^{n\times 1}.$$

Noticed that the general form of ODE (2.1) has only one nonlinear term $\mathbf{F}(\mathbf{y}(t))$ but our model problem (4.10) has also the term $\mathbf{g}(t)$, which comes from the inhomogeneous boundary conditions.

We solve ODE full-order system (4.10) by semi-implicit Euler method, by using the following time discretization

$$\frac{y_i^{j+1} - y_i^j}{\Delta t} = \mathbf{A}y_i^{j+1} + \mathbf{g}(t) + \mathbf{F}(y_i^j(t)) \tag{4.11}$$

or

$$y_i^{j+1} = (\mathbf{I} - \Delta t\mathbf{A})^{-1}[y_i^j + \Delta t\mathbf{g}(t) + \Delta t\mathbf{F}(y_i^j(t))] \tag{4.12}$$

where $\mathbf{F}(\mathbf{y}(t)) = \mathbf{y}.*(1-\mathbf{y})$, and the notation " .* " denotes the componentwise evaluation.

4.3. Model Reduction for Fisher's Equation

The projection basis or POD basis $\mathbf{V}_k$ of the dimension $k, k \ll n$ for reduced system is constructed by using steps from Algorithm 1. As done in Section 2, we substitute (2.2) into (2.1) and applying the Galerkin projection. The POD reduced system is of the form:

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \underbrace{\mathbf{V}_k^T\mathbf{A}\mathbf{V}_k}_{\tilde{\mathbf{A}}}\tilde{\mathbf{y}}(t) + \mathbf{V}_k^T\mathbf{g}(t) + \mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)), \tag{4.13}$$

where $\tilde{\mathbf{A}} = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k \in \mathbb{R}^{k\times k}$ can be precomputed because it does not depend on time variable t. Then we solve POD reduced system for (4.13) by the semi-implicit Euler method.

$$\frac{y_i^{j+1} - y_i^j}{\Delta t} = \tilde{\mathbf{A}}y_i^{j+1} + \mathbf{V}_k^T\mathbf{g}(t) + \mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k y_i^j(t)) \tag{4.14}$$

or

$$y_i^{j+1} = (\mathbf{I}_k - \Delta t\tilde{\mathbf{A}})^{-1}[y_i^j + \Delta t\mathbf{V}_k^T\mathbf{g}(t) + \Delta t\mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k y_i^j(t))]. \tag{4.15}$$

As described in Section 2, computing POD reduced system still depends on n. Hence we combine POD with DEIM as in section 2.4 and Algorithm 2. We compute projection basis $\mathbf{U}$ from $\mathbb{F} = [\mathbf{F}(\mathbf{y}(t_1)), \ldots, \mathbf{F}(\mathbf{y}(t_{n_t}))]$. Note that the function $\mathbf{g}(t)$ in this section is a constant vector and therefore $\mathbf{V}_k^T\mathbf{g}(t)$ can be precomputed. However, it is possible that $\mathbf{g}(t)$ depends on the variable t, which could require us to compute $\mathbf{V}_k^T\mathbf{g}(t)$ every time step

t. Therefore, we will also present here a DEIM approximation for $\mathbf{V}_k^T\mathbf{g}(t)$ as in the case of $\mathbf{V}_k^T\mathbf{F}(\mathbf{V}_k\mathbf{y}\tilde{(t)})$. The function $\mathbf{g}(t)$, is first approximated in the form

$$\mathbf{g}(t) \approx \mathbf{W}\mathbf{d}(t) \tag{4.16}$$

where $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_q] \in \mathbb{R}^{n\times q}$ is the projection basis, $\mathbf{d}(t) \in \mathbb{R}^{q\times 1}$ is the coefficient vector with $q \ll n$. Note that, above estimates (4.16) by projecting $\mathbf{g}(t)$ subspace $span\{\mathbf{W}\}$. The basis matrix $\mathbf{W}$ can be found by using SVD of $[\mathbf{g}(t_1), \dots, \mathbf{g}(t_{n_s})]$ and he coefficient vetor $\mathbf{d}(t) \in \mathbb{R}^{q\times 1}$. Then we can calculate $\mathbf{d}(t)$ from the following interpolation method. Consider the matrix

$$\mathbf{J} = [\mathbf{e}_{\wp_1}, \dots, \mathbf{e}_{\wp_q}] \in \mathbb{R}^{n\times q} \tag{4.17}$$

where $\wp_j = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ is the $\wp_j$-th column of the identity matrix $\mathbf{I}_n \in \mathbf{R}^{n\times n}$, for $j = 1, \dots, q$ by selecting q rows for $\mathbf{W}$. Then assume that $\mathbf{J}^T\mathbf{W}$ is nonsingular and solve for $\mathbf{d}(t)$ from

$$\mathbf{J}^T\mathbf{g}(t) = (\mathbf{J}^T\mathbf{W})^{-1}\mathbf{d}(t). \tag{4.18}$$

That is,

$$\mathbf{d}(t) = (\mathbf{J}^T\mathbf{W})^{-1}\mathbf{J}^T\mathbf{g}(t). \tag{4.19}$$

Lastly, the approximation is given by

$$\mathbf{g}(t) \approx \mathbf{W}\mathbf{d}(t) = \mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1}\underbrace{\mathbf{J}^T\mathbf{g}(t)}_{q\times 1}. \tag{4.20}$$

Note that, multiplying $\mathbf{J}^T$ (4.18) is equivalent to extracting the q rows corresponding to the interpolation indices $\wp_1, \dots, \wp_q$ from Algorithm 2. After that, we let $\tilde{\mathbf{g}}(t)$ to be the interpolation approximation of $\mathbf{g}(t)$ in the form

$$\tilde{\mathbf{g}}(t) = \mathbf{W}\mathbf{d}(t) = \mathbb{J}\mathbf{g}(t), \tag{4.21}$$

where $\mathbb{J} = \mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1}\mathbf{J}^T$ is a projection. Therefore, $\mathbf{V}_k^T\mathbf{g}(t)$ in (4.13) can be written in the form

$$\mathbf{V}_k^T\mathbf{g}(t) \approx \underbrace{\mathbf{V}_k^T\mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1}}_{\text{precomputed}:k\times q}\underbrace{\mathbf{J}^T\mathbf{g}(t)}_{q\times 1} \tag{4.22}$$

where $\mathbf{V}_k^T\mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1}$ can be precomputed because there is no dependence on t. The resulting reduced system, which has no dependence on n is given by

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{y}}(t) + \mathbb{H}\mathbf{g}_\wp(t) + \mathbb{E}\mathbf{F}_\wp(\tilde{\mathbf{y}}(t)) \tag{4.23}$$

where $\tilde{\mathbf{A}} = \mathbf{V}_k^T\mathbf{A}\mathbf{V}_k \in \mathbb{R}^{k\times k}, \mathbb{H} = \mathbf{V}_k^T\mathbf{W}(\mathbf{J}^T\mathbf{W})^{-1} \in \mathbb{R}^{k\times q}, \mathbf{g}_\wp(t) = \mathbf{V}_k^T\mathbf{g}(t) \in \mathbb{R}^{q\times 1}, \mathbb{E} = \mathbf{V}_k^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1} \in \mathbb{R}^{k\times m}, \mathbf{F}_\wp(\tilde{\mathbf{y}}(t)) = \mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)) \in \mathbb{R}^{m\times 1}$ and $\tilde{\mathbf{A}}, \mathbb{H}$, and $\mathbb{E}$ can be precomputed no depend on t. The corresponding semi-implicit discretization is given by

$$\frac{y_i^{j+1} - y_i^j}{\Delta t} = \tilde{\mathbf{A}}y_i^{j+1} + \mathbb{H}\mathbf{J}^T\mathbf{g}(t) + \mathbb{E}\mathbf{F}_\wp(\mathbf{V}_k y_i^j(t)) \tag{4.24}$$

or

$$y_i^{j+1} = (\mathbf{I}_k - \Delta t\tilde{\mathbf{A}})^{-1}[y_i^j + \Delta t\mathbb{H}\mathbf{J}^T\mathbf{g}(t) + \Delta t\mathbb{E}\mathbf{F}_\wp(\mathbf{V}_k y_i^j(t))]. \tag{4.25}$$

The next section shows some numerical results form applying this POD-DEIM approach on Fisher's equation.

5. Numerical Results

This section demonstrates the numerical results of Fisher's equation by using MATLAB in computation. There are three applications considered in this section. The first numerical experiment applies POD and DEIM to directly construct a reduced system with the same setting as the original full-order system. The second numerical test investigates the effect of basis used for constructing reduced systems by considering "POD basis with snapshot difference quotients" as suggested in [33]. The last experiment illustrates the usefulness of the POD-DEIM approach for Fisher's equation with various boundary conditions.

5.1. Application 1 : the POD and POD-DEIM Reduced System for Fisher's Equation

In this numerical experiment, we present the applcation of POD and DEIM reduced system for Fisher's equation with the same parameter and the same boundary conditions. Finite difference discretization with semi-implicit scheme is used to obtain numerical results. Figure 1 is the numerical solution of the original FD system of Fisher's equation with dimension 500. We use 500 solutions or snapshots to construct reduced basis by using SVD or POD. Figure 2 shows the singular values of the 500 solution snapshots of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ which can be obtained simultaneously from the full-order FD discretization of the Fisher's equation. We can then construct POD reduced system from this basis. Figure 3 is the numerical solution of the POD reduced system of the Fisher's equation with different POD dimensions.

Figure 4 is the numerical solution of the POD-DEIM reduced system of the Fisher's equation with different POD and DEIM dimensions. Notice that solutions from both POD and POD-DEIM reduced systems look the same as the solution of full-order system in Figure 1. The errors of POD and POD-DEIM reduced system and simulation time are shown in Table 1.

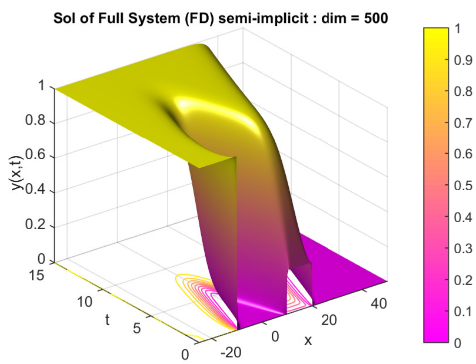
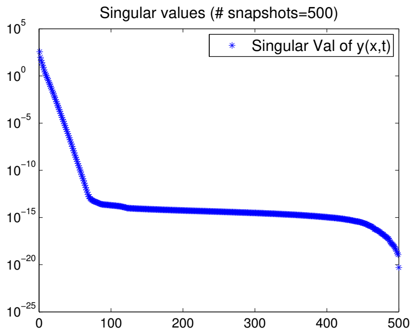
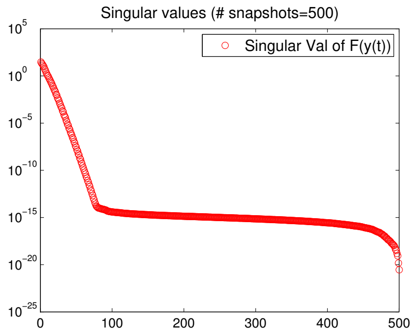


Figure 1. [Application 1] Numerical solution of the original FD system (dim 500) of Fisher's equation.

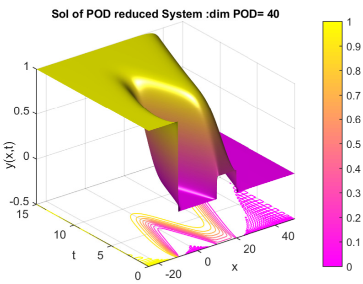


(a) The singular values of $\mathbf{y}(t)$.

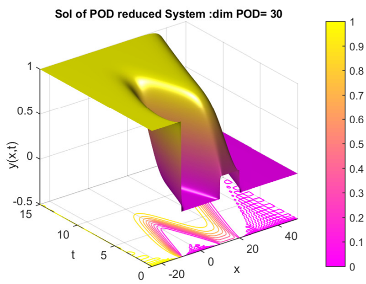


(b) The singular values of $\mathbf{F}(\mathbf{y}(t))$.

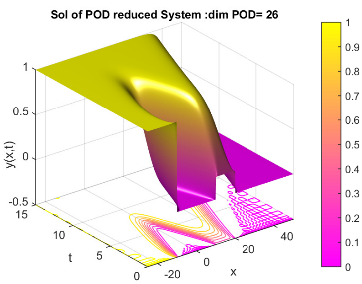
FIGURE 2. [Application 1] The singular values of the 500 snapshots solution of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ from the full order FD discretization of the Fisher's equation.



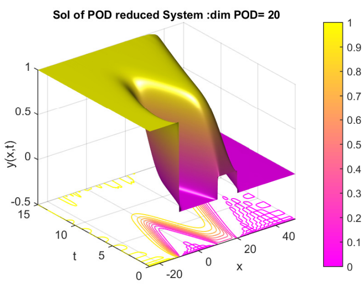
(a) POD dim = 40.



(b) POD dim = 30.

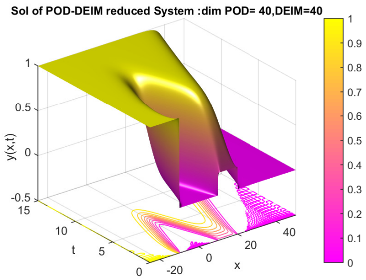


(c) POD dim = 26.

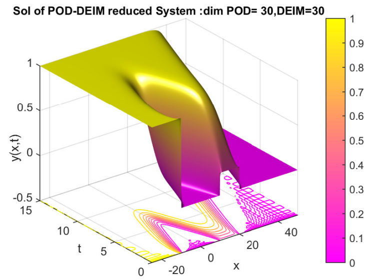


(d) POD dim = 20.

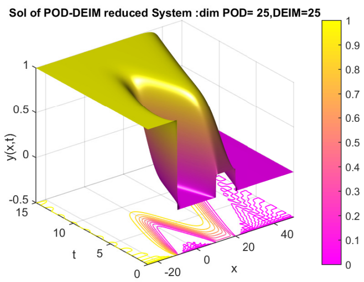
FIGURE 3. [Application 1] Solution of POD reduced system of Fisher's equation with different POD and DEIM dimensions.



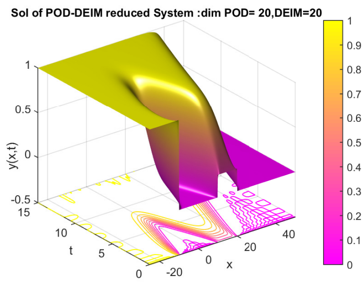
(a) POD dim = 40, DEIM dim = 40.



(b) POD dim = 30, DEIM dim = 30.



(c) POD dim = 25, DEIM dim = 25.



(d) POD dim = 25, DEIM dim = 25.

FIGURE 4. [Application 1] Solution of the POD-DEIM reduced system of Fisher's equation with different POD and DEIM dimensions.

Dimension	Error (Average)	CPU Time (sec)	Ratio CPU Time
Full 500 (FD)	–	6.0730	1
POD 40	5.0719×10^{-16}	6.0730×10^{-1}	1/10
POD 30	5.0606×10^{-10}	5.4889×10^{-1}	1/11
POD 26	1.0355×10^{-6}	4.0487×10^{-1}	1/15
POD 20	2.3550×10^{-4}	2.8938×10^{-1}	1/21
POD 40/DEIM 40	5.8264×10^{-16}	2.0243×10^{-1}	1/30
POD 30/DEIM 30	8.2261×10^{-10}	1.9251×10^{-1}	1/32
POD 25/DEIM 25	1.4419×10^{-9}	1.5183×10^{-1}	1/40
POD 20/DEIM 20	9.8284×10^{-3}	1.3496×10^{-1}	1/45

TABLE 1. [Application 1] CPU time (and its corresponding ratio) of the full system, POD reduced system, and POD-DEIM reduced system; Average error of solutions.

Table 1 shows the accuracy via the average error of the solutions of POD and POD-DEIM reduced systems. Notice that the errors of POD reduced system with POD dimensions 20, which is approximately $\mathcal{O}(10^{-4})$ are less than the error of POD-DEIM reduced system with POD dimensions 20 and DEIM dimensions 20, which is approximately $\mathcal{O}(10^{-3})$. The simulation time of POD reduced system decreases by a factor of approximately 32 and CPU time of the POD-DEIM reduced system decreases by a factor of approximately 45. Obviously, the POD-DEIM reduced system spends less time than the POD reduced system. This can be explained by the fact that the DEIM approximation can efficiently approximate the nonlinear term as shown in Section 2. Figure 5 is the DEIM points where we plot the first 30 interpolation points and the first 10 points are labelled as $P_1, P_2, \ldots, P_{10}$. Figure 6 and 7 show POD basis vectors of the snapshots of linear and nonlinear term from (4.10). Figure 8 shows errors for DEIM reduced system which are the exact errors, error bounds in Theorem 2.4, and used $\max(\mathbf{F}(\mathbf{y}))$ combine with Theorem 2.4.

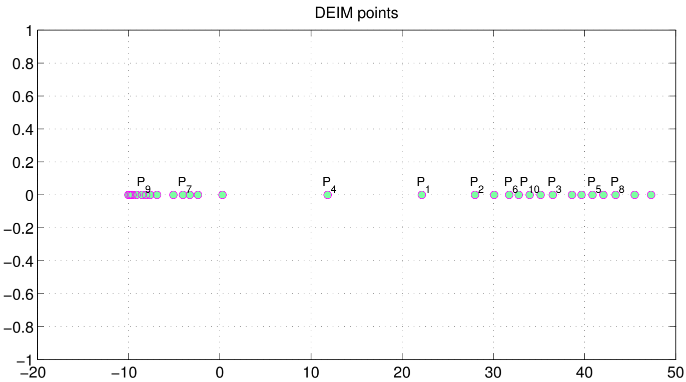
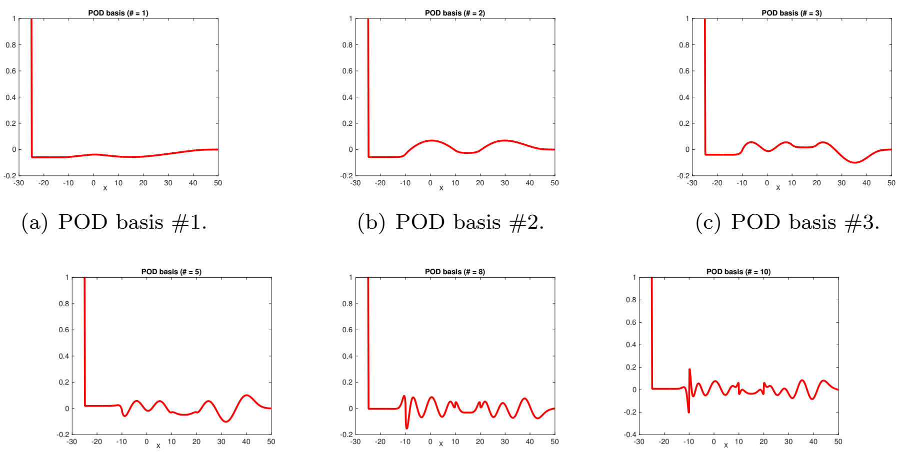


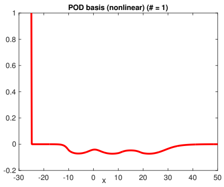
FIGURE 5. [Application 1] The first 30 points in the discretized spatial domain selected by the DEIM algorithm (the first 10 points are labelled by $P_1, P_2, \ldots, P_{10}$).



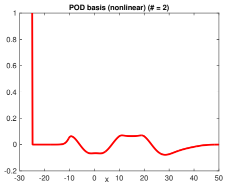
(a) POD basis #1. (b) POD basis #2. (c) POD basis #3.

(d) POD basis #5. (e) POD basis #8. (f) POD basis #10.

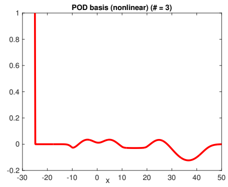
FIGURE 6. [Application 1] The POD basis vectors of the snapshots from (4.10).



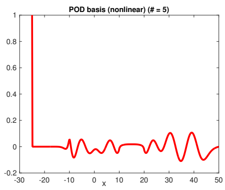
(a) POD basis (nonlinear) #1.



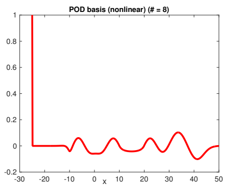
(b) POD basis (nonlinear) #2.



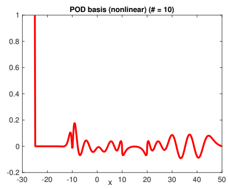
(c) POD basis (nonlinear) #3.



(d) POD basis (nonlinear) #5.



(e) POD basis (nonlinear) #8.



(f) POD basis (nonlinear) #10.

FIGURE 7. [Application 1] The POD basis vectors of the snapshots of nonlinear term from (4.10).

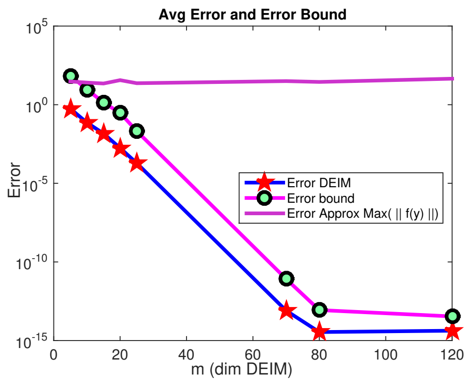


FIGURE 8. [Application 1] Average errors, error bounds, and approximate error bounds.

In the next section, we will use different snapshot sets to construct the POD basis. In particular, Section 5.2 will use additional "POD basis with snapshot difference quotients" to construct reduced systems.

5.2. Application 2 : POD Basis with Snapshot Difference Quotients for Fisher's Equation

In the second numerical experiment, we use the snapshots $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}]$ from nonlinear original system with additional finite difference of the adjacent snapshots as done in [33] $\left[\mathbf{Y}, \hat{\mathbf{Y}}\right]$, $\hat{\mathbf{Y}} = \left[\frac{\mathbf{y}_2 - \mathbf{y}_1}{\Delta t}, \ldots, \frac{\mathbf{y}_{n_s} - \mathbf{y}_{n_s-1}}{\Delta t}\right] \in \mathbb{R}^{n \times n_s - 1}$. We will refer to these snapshots $\left(\frac{\mathbf{y}_{j+1} - \mathbf{y}_j}{\Delta t}\right), j = 1, 2, \ldots, n_s$ as the "POD basis with snapshot difference quotients." A similar approach is used for the snapshots of nonlinear term to construct POD basis for DEIM approximation. Then these basis sets are used to construct POD and POD-DEIM reduced systems.

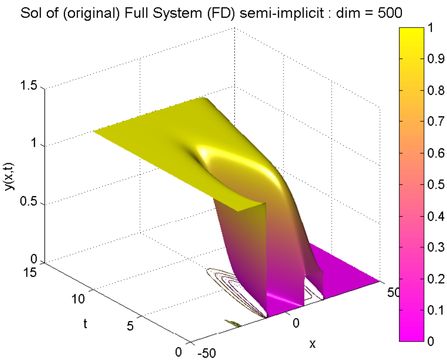
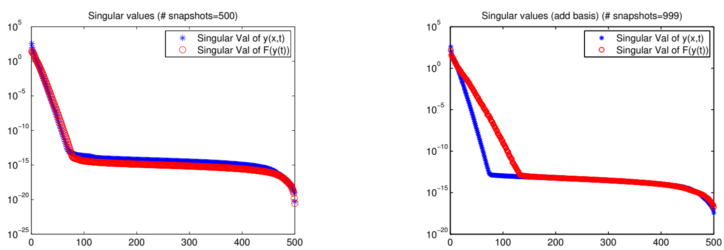


FIGURE 9. [Application 2] Numerical solution of the original FD system (dim 500) of Fisher's equation.

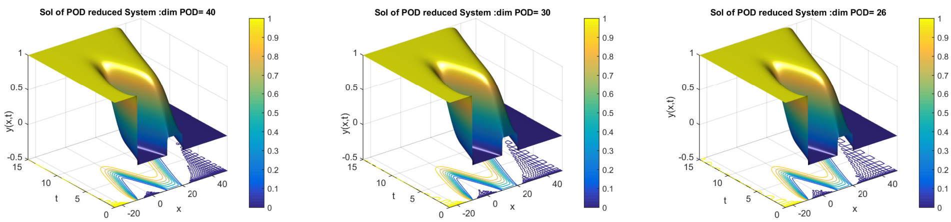
Figure 9 shows the numerical solution of Fisher's equation with dimension 500. We use 500 snapshots to find POD basis as explained in Algorithm 1. The singular values of the 500 solutions snapshots of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ are shown in Figure 10(a). The decay of these plots suggests that using 60 basis vectors is enough to capture the dynamics of the collected snapshots. However, when 999 solution snapshots of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ are used to construct POD basis, the decay of singular values in Figure 10(b) suggests that we should use approximately 120 basis vectors to capture all features of the full-order system.



(a) The singular values of the 500 snapshots.

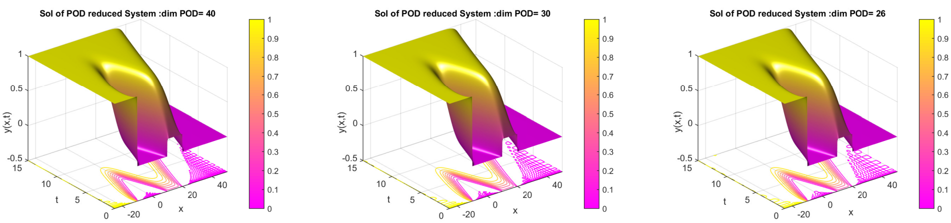
(b) The singular values of the 999 snapshots.

FIGURE 10. [Application 2] The singular values of the number snapshots of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ from the difference POD basis of the Fisher's equation.



(a) POD dim = 40. (b) POD dim = 30. (c) POD dim = 26.

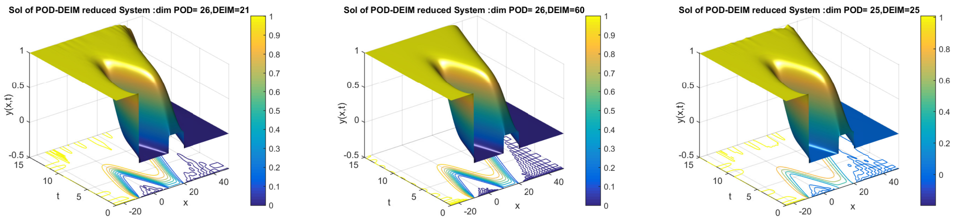
Figure 11. [Application 2] Solution of POD reduced system for Fisher's equation with different POD and DEIM dimensions by using the 999 snapshot solutions of the original FD system with POD basis with snapshot difference quotients.



(a) POD dim = 40. (b) POD dim = 30. (c) POD dim = 26.

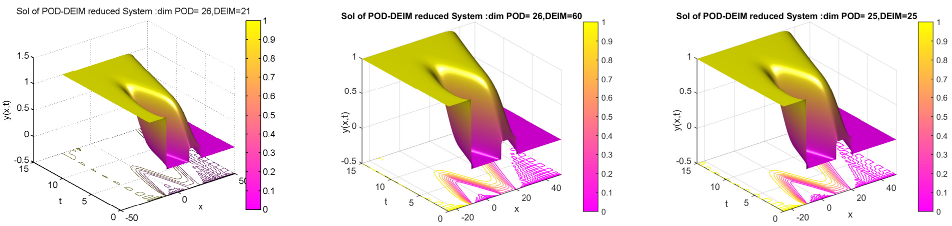
Figure 12. [Application 2] Solution of POD reduced system for Fisher's equation with different POD and DEIM dimensions by using the 500 snapshot solutions of the original FD system.

Figure 11 illustrates the numerical solutions of the POD reduced systems with the POD basis constructed from snapshot difference quotients of the Fisher's equation. Notice that they are similar to the solution of POD reduced system for Fisher's equation with original POD basis in Figure 12 and full-order system in Figure 9. This shows that POD reduced systems can be computed with less CPU time than full-order system while still maintaining the accuracy. However, the complexity of nonlinear term is not truely reduced. So we combine POD approximation with DEIM algorithm to obtain a POD-DEIM reduced system. Figure 13 shows the numerical solutions of the POD-DEIM reduced systems with POD basis constructed from snapshot difference quotients of the Fisher's equation. Notice that they are similar to the solution of POD reduced system for Fisher's equation with original POD basis as shown in Figure 12 and full-order system as shown in Figure 9. In addition, the solutions from both the POD and POD-DEIM reduced systems look the same as the solution of full-order system in Figure 9. The errors of the POD and POD-DEIM reduced systems and simulation time are shown in Table 2.



(a) POD = 26, DEIM = 21. (b) POD = 26, DEIM = 60. (c) POD = 25, DEIM = 25.

Figure 13. [Application 2] Solution of the POD-DEIM reduced system for Fisher's equation with different POD and DEIM dimensions by using the 999 snapshot solutions of the original FD system with POD basis with snapshot difference quotients.



(a) POD = 26, DEIM = 21. (b) POD = 26, DEIM = 60. (c) POD = 25, DEIM = 25.

Figure 14. [Application 2] Solution of the POD-DEIM reduced system for Fisher's equation with different POD and DEIM dimensions by using the 500 snapshot solutions of the original FD system.

Table 2 presents the simulation time and the accuracy of the POD and POD-DEIM reduced systems, which shows that basis constructed with additional POD basis with snapshot difference quotients gives more accurate results for POD reduced system but not for POD-DEIM reduced system in the case of dimension POD 26 and DEIM 21. However, for dimension POD 26 and DEIM 60, the numerical approximations are more accurate when using these additional snapshots quotients for both POD and POD-DEIM reduced systems. These results show that adding snapshot difference quotients can increase the accuracy of POD reduced system, but does not neccessary improve the accuracy of POD-DEIM reduced system.

In the next section, we will experiment various boundary conditions with Fisher's equation.

Dimension	Error (Average)	CPU Time (sec)	Ratio CPU Time
Full 500 (FD)	–	6.0730	1
POD 40	5.0719×10^{-16}	6.0730×10^{-1}	1/10
POD 30	5.0606×10^{-10}	5.4889×10^{-1}	1/11
POD 26	1.0355×10^{-6}	4.0487×10^{-1}	1/15
POD 20	2.3550×10^{-4}	2.8938×10^{-1}	1/21
POD 40/DEIM 40	5.8264×10^{-16}	2.0243×10^{-1}	1/30
POD 30/DEIM 30	8.2261×10^{-10}	1.9251×10^{-1}	1/32
POD 26/DEIM 21	1.1042	1.8403×10^{-1}	1/33
POD 26/DEIM 60	1.4737×10^{-9}	2.3358×10^{-1}	1/26
POD 25/DEIM 25	1.4419×10^{-9}	1.5183×10^{-1}	1/40
POD 20/DEIM 20	9.8284×10^{-3}	1.3496×10^{-1}	1/45

TABLE 2. [Application 2] Average error of solutions; CPU time (and its corresponding ratio) of the full system, POD reduced system, and POD-DEIM reduced system.

5.3. Application 3 : the Varying Boundary Conditions for Fisher's Equation

In the last application, we consider many boundary conditions for Fisher's equation. The numerical test in this section have the same initial conditions as the experiments in Section 5.1, but they use different boundary conditions.

Figure 15 shows the solution of full-order system with dimension 500 with boundary conditions: $y(-25,t) = \sin(\frac{4\pi}{5}t), y(50,t) = 0$. Then, we construct snapshots matrix: $\mathbf{Y} = \left[\mathbf{Y}_1, \mathbf{Y}_2\right]$ from Figure 16, where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are the solutions shown in Figure 16(a) and 16(b), respectively. After that we compute the POD basis from the snapshots of the two full-order systems, one uses boundary condition $y(-25,t) = \sin(\frac{\pi}{2}t)$ and the other uses $y(-25,t) = \sin(\pi t)$. The resulting singular values are shown in Figure 17 which use 1,000 snapshots of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ from Figure 16. Notice that it is enough to just use 120 basis for describing the original solutions. When we compute POD basis, we can use construct a POD reduced system.

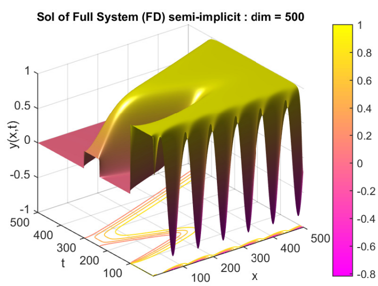
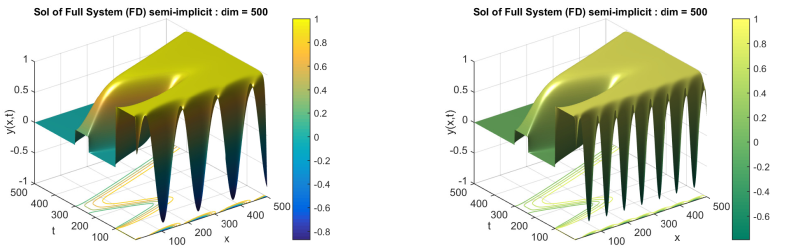


FIGURE 15. [Application 3] Solution of full-order system for Fisher's equation with boundary conditions is $y(-25,t) = \sin(\frac{4\pi}{5}t), y(50,t) = 0$.



(a) Boundary conditions is $y(-25,t) = \sin(\frac{\pi}{2}t), y(50,t) = 0$

(b) Boundary conditions is $y(-25,t) = \sin(\pi t), y(50,t) = 0$

FIGURE 16. [Application 3] Solution of full-order systems for Fisher's equation with the different boundary conditions.

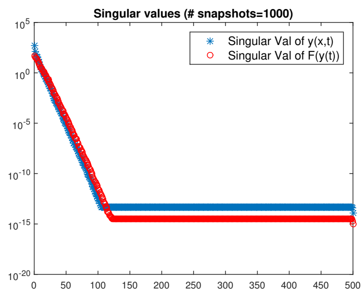
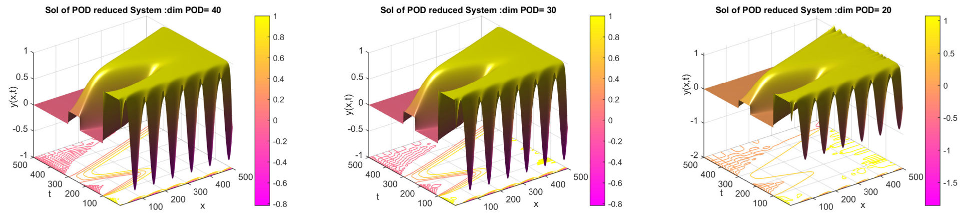


FIGURE 17. [Application 3] The singular values of the 1,000 snapshots solution of $\mathbf{y}(t)$ and $\mathbf{F}(\mathbf{y}(t))$ from the full-order FD discretization of the Fisher's equation with the boundary conditions from the solution in Figure 16.

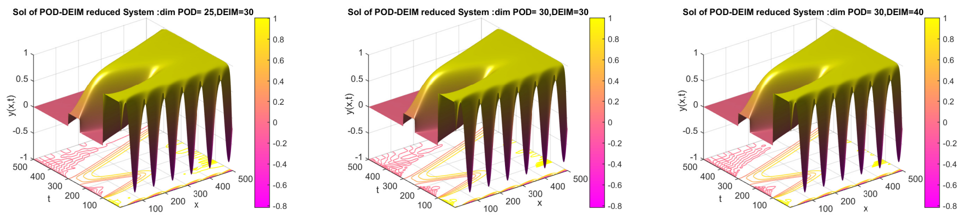
Figure 18 shows the solutions of POD reduced systems for Fisher's equation with the different POD dimensions. Notice that they are similar to the solution of full-order system in Figure 15. The numerical experiment shows that POD reduced system can be used to compute the solutions with less time than the full-order system while still maintaining the accuracy. Figure 19 shows the solutions of POD-DEIM reduced systems for Fisher's equation with different POD and DEIM dimensions. Notice that they are similar to the solution of full-order system in Figure 15. The POD-DEIM reduced system can be used to compute the solutions with less CPU time than both the full-order system and the POD reduced system. The average errors and CPU time are given in Table 3.

Notice that, although the solutions of Fisher's equation with boundary conditions $y(-25,t) = \sin(\frac{4\pi}{5}t), y(50,t) = 0$ are not used in the construction of POD basis, the POD and POD-DEIM reduced systems give very accurate approximations as shown in Figure 18 and 19. From Table 3, CPU time of POD reduced system reduce by a factor of approximately 12 and CPU time of POD-DEIM reduced system decreases by a factor of approximately 30. That is POD-DEIM reduced system uses less the computational time than POD reduced system, as shown in the previous two numerical tests.



(a) POD dim = 40. (b) POD dim = 30. (c) POD dim = 20.

FIGURE 18. [Application 3] Solution of POD reduced system for Fisher's equation with the 3 different POD dimensions where boundary conditions are $y(-25,t) = \sin(\frac{4\pi}{5}t)$, and $y(50,t) = 0$.



(a) POD = 25, DEIM = 30. (b) POD = 30, DEIM = 30. (c) POD = 30, DEIM = 40.

FIGURE 19. [Application 3] Solution of POD-DEIM reduced system for Fisher's equation with the 3 different dimensions of POD and DEIM where boundary conditions are $y(-25,t) = \sin(\frac{4\pi}{5}t)$, and $y(50,t) = 0$.

Dimension	Error (Average)	CPU Time (sec)	Ratio CPU Time
Full 500 (FD)	–	6.7724	1
POD 25	1.2600×10^{-3}	4.5149×10^{-1}	1/15
POD 30	8.5056×10^{-4}	6.7724×10^{-1}	1/10
POD 25/DEIM 30	7.9945×10^{-2}	1.7822×10^{-1}	1/38
POD 30/DEIM 30	2.0557×10^{-2}	2.1846×10^{-1}	1/31
POD 30/DEIM 40	7.3983×10^{-3}	2.3353×10^{-1}	1/29

TABLE 3. [Application 3] Average error of solutions and CPU time (with its corresponding ratio) of the full system, POD reduced system, and POD-DEIM reduced system.

6. Conclusion

This work focused on reducing the computational complexity of Fisher's equation, which had been used as the main model problem for nonlinear partial differential equations (PDEs). Finite difference (FD) method was used to discretize the model problem, which could result in a nonlinear ordinary differential equations (ODEs) with large dimension. This system was called full-order system and it was solved by using semi-implicit Euler method to obtain the solution snapshots. We used SVD to compute the POD basis set, and it was then combined with the Galerkin projection, which was called POD-Galerkin approach, to construct a POD reduced system with small dimension. Even if POD reduced system had a small dimension, it could not reduce the computational complexity of nonlinear term. Therefore, we combined POD with DEIM. The resulting POD-DEIM reduced system with small dimension could reduce the complexity in computing the nonlinear term. The accuracy of the reduced model was shown numerically to be in the same order when POD was extended by DEIM. An a-priori error bound was proposed for the POD-DEIM reduced-order solutions from the semi-implicit Euler method by extending the derivation in [59]. This work also considered combining the original snapshots with additional finite difference of the adjacent snapshots, which was called "POD basis with snapshot difference quotients." Adding the snapshot difference quotients could increase the accuracy of the POD reduced system as shown in [33], but it did not necessary increase the accuracy of the POD-DEIM reduced system. This work demonstrated the efficiency of the POD-DEIM technique for Fisher's equation with varying boundary conditions. We used two different boundary conditions that were sampled to generate snapshot matrix for computing POD basis sets used in the Galerkin projection and DEIM approximation. These basis sets were used to construct POD-DEIM reduced systems that could accurately approximate the solutions of the original systems with different boundary conditions, although these systems were not used in generating the snapshot matrix for POD basis.

Some possible extensions of this work are discussed as follows. The framework presented in this work can be directly extended to the 2D fisher's equation. The result in Section 5 shows that the snapshot selection procedure use for computing POD basis can be improved to give more accurate results for both POD and POD-DEIM reduced system. Algorithm for selecting DEIM indices can also be improved in order to have smaller error. In addition, the solutions of the full-order system, POD and POD-DEIM reduced systems which are computed by using semi-implicit Euler method in this work can be improved by using higher order numerical scheme. It is also important to investigate how to obtain the minimum dimensions that is still enough to maintain the stability and accuracy of the original system as much as possible. Since POD basis used in model reduction techniques is computed from the snapshots, the choice of snapshots is a very important factor for achieving good accuracy. We can also add the different snapshots from the original snapshots to increase the efficiency and accuracy of the reduced system. Some systems with external factors may not be predicted for their solution precisely. In many simulations, we are only able to describe the probability of that solutions by using the probability theory. Hence, a possible future research is to combine stochastic notion with model reduction frameworks that can maintain stability and can decrease computational time for simulating the large-scale systems. Finally, this POD-DEIM approach can be used to study sensitively analysis when other parameter values in the system are changed.

Acknowledgements

We would like to thank the referees for their comments and suggestions on the manuscript. This work was supported by the Faculty of Science and Technology, Thammasat University Grant No. 7/2558.

References

[1] D. Xiao A, F. Fang A, A. G. Buchan A, C. C. Pain A, I. M. Navon C, J. Du D, and G. Hu B. Non-linear model reduction for the navier-stokes equations using the residual deim method.

[2] FISHER R. A. The wave of advance of advantageous genes. *Annals of Eugenics*, 7(4):355–369, 1937.

[3] I. Petrovsky A. Kolomogoroff and 1 N. Piscounoff, Moscow Univ. Bull. Math. 1. 1937.

[4] Kamel Al-Khaled. Numerical study of fisher's reactiondiffusion equation by the sinc collocation method. *Journal of Computational and Applied Mathematics*, 137(2):245 – 255, 2001.

[5] S R Arridge, J P Kaipio, V Kolehmainen, M Schweiger, E Somersalo, T Tarvainen, and M Vauhkonen. Approximation errors and model reduction with an application in optical diffusion tomography. *Inverse Problems*, 22(1):175, 2006.

[6] M Barrault. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[7] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667 – 672, 2004.

[8] Gal Berkooz, Philip Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Rev. Fluid Mech*, pages 539–575, 1993.

[9] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42:1505–1516, 2004.

[10] A. Cammilleri, F. Gueniat, J. Carlier, L. Pastur, E. Memin, F. Lusseyran, and G. Artana. Pod-spectral decomposition for fluid flow analysis and model reduction. *Theoretical and Computational Fluid Dynamics*, 27(6):787–815, 2013.

[11] Kevin Carlberg, Charbel Farhat, Julien Cortial, and David Amsallem. The {GNAT} method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242(0):623 – 647, 2013.

[12] Dominique Chapelle, Asven Gariah, and Jacques Sainte-Marie. Galerkin approximation with Proper Orthogonal Decomposition: new error estimates and illustrative examples. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(4):731–757, 2012.

[13] Saifon Chaturantabut and Danny C. Sorensen. Discrete Empirical Interpolation for nonlinear model reduction. In *Conference on Decision and Control*, pages 4316–4321, 2009.

[14] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, September 2010.

[15] Saifon Chaturantabut and Danny C. Sorensen. Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media. *Mathematical and Computer Modelling of Dynamical Systems*, 17:337–353, 2011.

[16] Saifon Chaturantabut and Danny C. Sorensen. A state space error estimate for pod-deim nonlinear model reduction. *SIAM J. Numer. Anal.*, 50(1):46–63, January 2012.

[17] Laurent Cordier, Manuel Girault, and Daniel Petit. Reduced order modeling by modal identification method and pod-galerkin approach of the heated circular cylinder wake in mixed convection. *Journal of Physics: Conference Series*, 395(1):012102, 2012.

[18] R. Ştefănescu and I. M. Navon. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. *Journal of Computational Physics*, 237:95–114, March 2013.

[19] J. Duffin, O. Sobczyk, A.P. Crawley, J. Poublanc, D.J. Mikulis, and J.A. Fisher. The dynamics of cerebrovascular reactivity shown with transfer function analysis. *NeuroImage*, 114(0):207 – 216, 2015.

[20] Freddy Dumortier, Nikola Popovi, and Tasso J Kaper. The critical wave speed for the fisherkolmogorovpetrowskiipiscounov equation with cut-off. *Nonlinearity*, 20(4):855, 2007.

[21] Kristine Embree. Fisher-KPP equation nonlinear traveling waves, 2001. [Online; accessed 30 August 2001].

[22] F. Fang, C.C. Pain, I.M. Navon, A.H. Elsheikh, J. Du, and D. Xiao. Non-linear petrovgalerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. *Journal of Computational Physics*, 234(0):540 – 559, 2013.

[23] Zhengkun Feng and Azzeddine Soulaimani. Reduced order modelling based on pod method for 3d nonlinear aeroelasticity. In *The 18th IASTED International Conference on Modelling and Simulation*, MS '07, pages 489–494, Anaheim, CA, USA, 2007. ACTA Press.

[24] Kimberly J. Fink and Laura Ray. Individualization of head related transfer functions using principal component analysis. *Applied Acoustics*, 87(0):162 – 173, 2015.

[25] R. A. FISHER. The wave of advance of advantageous genes. *Annals of Eugenics*, 7(4):355–369, 1937.

[26] Brian A. Freno and Paul G.A. Cizmas. A proper orthogonal decomposition method for nonlinear flows with deforming meshes. *International Journal of Heat and Fluid Flow*, 50(0):145 – 159, 2014.

[27] Sudipta Ghosh and Nilanjan Senroy. Balanced truncation based reduced order modeling of wind farm. *International Journal of Electrical Power and Energy Systems*, 53(0):649 – 655, 2013.

[28] G.H. Golub and C.F. Van Loan. *Matrix Computations.*

[29] M.D. Gunzburger, L.S. Hou, and W. Zhu. Fully discrete finite element approximations of the forced fisher equation. *Journal of Mathematical Analysis and Applications*, 313(2):419 – 440, 2006.

[30] Roi Gurka, Alexander Liberzon, and Gad Hetsroni. {POD} of vorticity fields: A method for spatial characterization of coherent structures. *International Journal of Heat and Fluid Flow*, 27(3):416 – 423, 2006.

[31] Michael Hinze and Stefan Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In Peter Benner, DannyC. Sorensen, and Volker Mehrmann, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 261–306. Springer Berlin Heidelberg, 2005.

[32] Amit Hochman, Bradley N. Bond, and Jacob K. White. A stabilized discrete empirical interpolation method for model reduction of electrical, thermal, and microelectromechanical systems. In *Proceedings of the 48th Design Automation Conference*, DAC '11, pages 540–545, New York, NY, USA, 2011. ACM.

[33] Traian Iliescu and Zhu Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM Journal on Scientific Computing*, 36(3):A1221–A1250, 2014.

[34] G.B. Jasmon and L.H.C.C. Lee. Distribution network reduction for voltage stability analysis and loadflow calculations. *International Journal of Electrical Power and Energy Systems*, 13(1):9 – 13, 1991.

[35] Eileen Kammann, Fredi Trltzsch, and Stefan Volkwein. A method of a-posteriori error estimation with application to proper orthogonal decomposition. 2012.

[36] Wei Kang, Jia-Zhong Zhang, Sheng Ren, and Peng-Fei Lei. Nonlinear galerkin method for low-dimensional modeling of fluid dynamic system using {POD} modes. *Communications in Nonlinear Science and Numerical Simulation*, 22(13):943 – 952, 2015.

[37] Anthony R. Kellems, Saifon Chaturantabut, and Steven J. Cox. Morphologically accurate reduced order modeling of spiking neurons. *Journal of Computational Neuroscience*, 28:477–494, 2010.

[38] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.

[39] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, February 2002.

[40] F Lanata and A Del Grosso. Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals. *Smart Materials and Structures*, 15(6):1811, 2006.

[41] Norman Lang, Jens Saak, and Tatjana Stykel. Towards practical implementations of balanced truncation for {LTV} systems. *IFAC-PapersOnLine*, 48(1):7 – 8, 2015.

[42] Alex Liberzon, Roi Gurka, Iztok Tiselj, and Gad Hetsroni. Spatial characterization of the numerically simulated vorticity fields of a flow in a flume. *Theoretical and Computational Fluid Dynamics*, 19(2):115–125, 2005.

[43] J. L. Lumley. Stochastic Tools in Turbulence. *Academic Press, New York*, 1970.

[44] J.L. Lumley. The structure of inhomogeneous turbulent flows. *in Atmospheric Turbulence and Radio Wave Propagation (A. M. Yaglom and V. I. Tararsky, eds.), (Nauka, Moscow)*, 1967.

[45] Zhendong Luo, Jiang Zhu, Ruiwen Wang, and I.M. Navon. Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical pacific ocean reduced gravity model. *Computer Methods in Applied Mechanics and Engineering*, 196(4144):4184 – 4195, 2007.

[46] Hung V. Ly and Hien T. Tran. Proper orthogonal decomposition for flow calculations and optimal control in a horizontal cvd reactor. Technical report, 1998.

[47] Vicenc Mendez, Sergei Fedotov, and Werner Horsthemke. *Reaction-Transport Systems: Mesoscopic Foundations, Fronts, and Spatial Instabilities (Springer Series in Synergetics)*. Springer, 2010 edition, June 2010.

[48] Ha Binh Minh, Carles Batlle, and Enric Fossas. A new estimation of the lower error bound in balanced truncation method. *Automatica*, 50(8):2196 – 2198, 2014.

[49] Thien Duy Nguyen, John Craig Wells, Paritosh Mokhasi, and Dietmar Rempfer. Proper orthogonal decomposition-based estimations of the flow field from particle image velocimetry wall-gradient measurements in the backward-facing step flow. *Measurement Science and Technology*, 21(11):115406, 2010.

[50] Daniel Olmos and Bernie D. Shizgal. A pseudospectral method of solution of fisher's equation. *J. Comput. Appl. Math.*, 193(1):219–242, August 2006.

[51] Casian Alexandru Pantea. *Mathematical and Computational Analysis of Biochemical Reaction Networks*. PhD thesis, University of Wisconsin at Madison, Madison, WI, USA, 2010. AAI3437121.

[52] A. Placzek, D.-M. Tran, and R. Ohayon. A nonlinear pod-galerkin reduced-order model for compressible flows taking into account rigid body motions. *Computer Methods in Applied Mechanics and Engineering*, 200(4952):3497 – 3514, 2011.

[53] Michael Presho, Anastasiya Protasov, and Eduardo Gildin. Localglobal model reduction of parameter-dependent, single-phase flow models via balanced truncation. *Journal of Computational and Applied Mathematics*, 271(0):163 – 179, 2014.

[54] Y. Qiu and D.M. Sloan. Numerical solution of fisher's equation using a moving mesh method. *Journal of Computational Physics*, 146(2):726 – 746, 1998.

[55] Shodhan Rao, Arjan van der Schaft, Karen van Eunen, Barbara M. Bakker, and Bayu Jayawardhana. Model-order reduction of biochemical reaction networks. *CoRR*, abs/1212.2438, 2012.

[56] Micha Rewieski and Jacob White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415(23):426 – 454, 2006. Special Issue on Order Reduction of Large-Scale Systems.

[57] P.R. Runkle, M.A. Blommer, and G.H. Wakefield. A comparison of head related transfer function interpolation methods. In *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, pages 88–91, Oct 1995.

[58] S. QIN S. TANG and R. O. WEBER. NUMERICAL STUDIES ON 2-DIMENSIONAL REACTION-DIFFUSION EQUATIONS. *J. Austral. Math. Soc. Sen B*, 35:223–243, 1993.

[59] K. Kunisch S. Volkwein, M. Kahlbacher and F. Troltzsch. *Proper Orthogonal Decomposition: Applications in Optimization and Control.*

[60] Elisa Schenone. *Reduced Order Models, Forward and Inverse Problems in Cardiac Electrophysiology.* Theses, Université Pierre et Marie Curie - Paris VI, November 2014.

[61] John R. Singler. Optimality of balanced proper orthogonal decomposition for data reconstruction ii: Further approximation results. *Journal of Mathematical Analysis and Applications*, 421(2):1006 – 1020, 2015.

[62] Angela Slavova. Cellular neural network models of some equations from biology, physics and ecology. *FUNCTIONAL DIFFERENTIAL EQUATIONS*, 10:579–591, 2003.

[63] D. Stan and J. L. Vázquez. The Fisher-KPP equation with nonlinear fractional diffusion. *ArXiv e-prints*, March 2013.

[64] Razvan Stefanescu, Adrian Sandu, and Ionel Michael Navon. POD/DEIM strategies for reduced data assimilation systems. *CoRR*, abs/1402.5992, 2014.

[65] Kyle Chand William Arrighi Tanya Kostova, Geoffrey Oxberry. Error bounds and analysis of proper orthogonal decomposition model reduction methods using snapshots from the solution and the time derivatives.
[66] Jonathan H. Tu and Clarence W. Rowley. An improved algorithm for balanced {POD} through an analytic treatment of impulse response tails. *Journal of Computational Physics*, 231(16):5317 – 5333, 2012.
[67] Alexander Vendl and Heike Fabender. Projection-based model order reduction for steady aerodynamics. In Norbert Kroll, Rolf Radespiel, Jan Willem Burg, and Kaare Srensen, editors, *Computational Flight Testing*, volume 123 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 151–166. Springer Berlin Heidelberg, 2013.
[68] S. Volkwein, editor. *Model Reduction using proper Orthogonal Decomposition*. 2008.
[69] T. Vo, R. Pulch, E.J.W. ter Maten, and A. El Guennouni. Trajectory piecewise linear approach for nonlinear differential-algebraic equations in circuit simulation. In Gabriela Ciuprina and Daniel Ioan, editors, *Scientific Computing in Electrical Engineering*, volume 11 of *Mathematics in Industry*, pages 167–173. Springer Berlin Heidelberg, 2007.
[70] A-xia Wang and Yi-Chen Ma. An error estimate of the proper orthogonal decomposition in model reduction and data compression. *Numerical Methods for Partial Differential Equations*, 25(4):972–989, 2009.
[71] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, pages 2323–2330, 2002.
[72] John Williamson. Note on a principal axis transformation for non-hermitian matrices. *Bull. Amer. Math. Soc.*, 45:920–922, 12 1939.
[73] Daniel Wirtz, Danny C. Sorensen, and Bernard Haasdonk. A-posteriori error estimation for deim reduced nonlinear dynamical systems. *Submitted to SIAM Journal on Scientific Computing*, 2012.
[74] D. Xiao, F. Fang, J. Du, C.C. Pain, I.M. Navon, A.G. Buchan, A.H. ElSheikh, and G. Hu. Non-linear petrovgalerkin methods for reduced order modelling of the navierstokes equations using a mixed finite element pair. *Computer Methods in Applied Mechanics and Engineering*, 255(0):147 – 157, 2013.
[75] Xunnian Yang and Jianmin Zheng. Curvature tensor computation by piecewise surface interpolation. *Computer-Aided Design*, 45(12):1639 – 1650, 2013.
[76] Xiaoji Ye, Peng Li, Min Zhao, R. Panda, and Jiang Hu. Scalable analysis of mesh-based clock distribution networks using application-specific reduced order modeling. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(9):1342–1353, Sept 2010.
[77] Wojciech Zalewski and Piotr Antoni Gryglaszewski. Mathematical model of heat and mass transfer processes in evaporative fluid coolers. *Chemical Engineering and Processing: Process Intensification*, 36(4):271 – 280, 1997.
[78] Huiyan Zhang, Ligang Wu, Peng Shi, and Yuxin Zhao. Balanced truncation approach to model reduction of markovian jump time-varying delay systems. *Journal of the Franklin Institute*, (0):–, 2015.