



# Reduced-Order Modeling of a Local Discontinuous Galerkin Method for Burgers-Poisson Equations

Nattapol Ploymaklam<sup>1,2</sup> and Saifon Chaturantabut<sup>3,\*</sup>

<sup>1</sup> Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand  
e-mail : [nattapol.p@cmu.ac.th](mailto:nattapol.p@cmu.ac.th)

<sup>2</sup> Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand

<sup>3</sup> Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani 12121, Thailand  
e-mail : [saifon@mathstat.sci.tu.ac.th](mailto:saifon@mathstat.sci.tu.ac.th)

**Abstract** In this work, we apply model reduction techniques to efficiently approximate the solution of the Burgers-Poisson equation. The proper orthogonal decomposition (POD) framework is first used with the Galerkin projection to reduce the number of unknowns in the discretized system obtained from a local Discontinuous Galerkin (LDG) method. Due to nonlinearity of Burgers-Poisson equation, the complexity in computing the resulting POD reduced system may still depend on the original discretized dimension. The discrete empirical interpolation method (DEIM) is therefore used to solve this complexity issue. Numerical experiments demonstrate that the combination of POD and DEIM approaches can provide accurate approximate solution of the Burgers-Poisson equation with much less computational cost.

**MSC:** 65F30

**Keywords:** model order reduction; Burgers-Poisson equation; differential equations; proper orthogonal decomposition; discontinuous Galerkin

---

Submission date: 10.03.2020 / Acceptance date: 13.11.2020

## 1. INTRODUCTION

In this work, we are interested in reducing the computational cost for the numerical approximation of the Burgers-Poisson system

$$u_t + uu_x = \phi_x \tag{1.1a}$$

$$\phi_{xx} - \phi = u \tag{1.1b}$$

derived from the conservation law

$$\partial_t u + \partial_x f(u) + \partial_x [G * u] = 0, \tag{1.2}$$

---

\*Corresponding author.

presented in [1]. When the kernel  $G$  is symmetry, i.e.  $G(x) = G(-x)$ , it can be shown that the solution of (1.2) admits the invariant properties

$$\int u(x, t) dx = \int u(x, 0) dx, \quad (1.3)$$

$$\int u^2(x, t) dx = \int u^2(x, 0) dx. \quad (1.4)$$

One example of such kernels  $G$  is

$$G(x) = \frac{1}{2\pi} \int \left( \frac{\tanh k}{k} \right)^{1/2} e^{ikx} dk, \quad (1.5)$$

which yields Whitham equation. Another simple form of  $G$  is

$$G(x) = \frac{1}{2} e^{-|x|}, \quad (1.6)$$

which, together with  $f(u) = u^2/2$ , yields the system (1.1). Under appropriate conditions, it can be shown that the weak solution of the system (1.1) exists [2].

In [3], a local Discontinuous Galerkin (LDG) method was proposed to solve the system (1.1). It was proved and verified that the optimal convergence is achieved when polynomials of even degree are used in the approximation. The method involves rewriting the Burgers-Poisson system into a system of first-order PDE, then into the weak form. The numerical approximation is then obtained from finding the solution from the set of piecewise continuous functions which are discontinuous across the computational cells. This process is part of the discontinuous Galerkin (DG) framework. The DG idea was used for the first time by Reed and Hill in 1973 to solve the neutron transport equations [4]. Since then, the DG method has been made popular by Shu and Cockburn to solve the first-order hyperbolic conservative equation. (See [5] for introduction on general framework.) On the other hand, the process of rewriting PDE's of higher order before applying DG belongs to LDG framework. (See [6] for a complete introductory to LDG method.)

This paper is organized as follow. In Section 2, we summarize the LDG scheme for solving the Burgers-Poisson equations proposed in [3] and [7]. In Section 3, we describe the model reduction process on the LDG scheme. The results and discussion are presented in Section 4. Finally, some conclusion will be made in Section 5.

## 2. STUDY PROBLEM

In this section, we summarize the local discontinuous Galerkin schemes for the Burgers-Poisson system (subject to initial data  $u_0(x)$  and periodic boundary conditions) proposed in [3] and in [7]. First of all, the system (1.1) is extended to the form

$$u_t + uu_x - \phi_x = \epsilon u_{xx}, \quad (2.1a)$$

$$\phi_{xx} - \phi = u, \quad (2.1b)$$

on the domain  $(x, t) \in [0, L] \times (0, T)$  with the periodic boundary conditions

$$u(0, t) = u(L, t), \quad u_x(0, t) = u_x(L, t), \quad t \in [0, T], \quad (2.2)$$

and initial condition

$$u(x, 0) = u_0(x), \quad x \in [0, L]. \quad (2.3)$$

First, the interval  $I = [0, L]$  is partitioned into  $N$  equal subintervals  $I_j = [x_{j-1/2}, x_{j+1/2}]$ ,  $j = 1, \dots, N$ . The center of the cell is  $x_j = \frac{1}{2} (x_{j-1/2} + x_{j+1/2})$ . The solution is from the piecewise polynomial space  $V_h^k$  defined by

$$V_h^k = \{v : v|_{I_j} \in P^k(I_j), j = 1, 2, \dots, N\}. \tag{2.4}$$

For the LDG framework, auxiliary variable  $w = \sqrt{\epsilon}u_x$  and  $p = \phi_x$  are introduced so we can rewrite (1.1a)-(1.1b) as:

$$u_t + \left(\frac{u^2}{2}\right)_x - p - \sqrt{\epsilon}w_x = 0, \tag{2.5a}$$

$$w - \sqrt{\epsilon}u_x = 0 \tag{2.5b}$$

$$p - \phi_x = 0, \tag{2.5c}$$

$$p_x - \phi = u. \tag{2.5d}$$

Then, the scheme is defined as follows: find  $u_h, p_h, \phi_h, w_h \in V_h^k$  such that

$$\int_{I_j} (u_h)_t v \, dx - \int_{I_j} \frac{u_h^2}{2} v_x \, dx + \frac{\widehat{u_h^2}}{2} v|_{\partial I_j} - \int_{I_j} p_h v \, dx + \sqrt{\epsilon} \int_{I_j} w_h v_x \, dx - \sqrt{\epsilon} \widehat{w_h} v|_{\partial I_j} = 0, \tag{2.6a}$$

$$\int_{I_j} w_h z \, dx + \int_{I_j} \sqrt{\epsilon} u_h z_x \, dx - \sqrt{\epsilon} \widehat{u_h} z|_{\partial I_j} = 0, \tag{2.6b}$$

$$\int_{I_j} p_h \psi \, dx + \int_{I_j} \phi_h \psi_x \, dx - \widehat{\phi_h} \psi|_{\partial I_j} = 0, \tag{2.6c}$$

$$- \int_{I_j} p_h q_x \, dx + \widehat{p_h} q|_{\partial I_j} - \int_{I_j} (\phi_h + u_h) q \, dx = 0, \tag{2.6d}$$

$$\int_{I_j} (u_h - u)|_{t=0} v \, dx = 0, \tag{2.6e}$$

for all test functions  $v, z, \psi, q$  in the finite element space  $V_h^k$ . The choice for numerical fluxes  $\widehat{u_h^2}, \widehat{\phi_h}, \widehat{p_h}$  is given by

$$\widehat{u_h^2} = \frac{1}{3} ((u_h^+)^2 + u_h^+ u_h^- + (u_h^-)^2), \tag{2.7a}$$

$$\widehat{\phi_h} = \theta \phi_h^+ + (1 - \theta) \phi_h^-, \tag{2.7b}$$

$$\widehat{p_h} = (1 - \theta) p_h^+ + \theta p_h^-, \tag{2.7c}$$

where  $\theta \in [0, 1/2]$ . Here, the numerical fluxes at the endpoints of  $I$  can be defined using  $U_{1/2}^- := U_{N+1/2}^-$  and  $U_{N+1/2}^+ := U_{1/2}^+$  where  $U$  represents  $u_h^2, \phi_h$ , or  $p_h$ .

Let  $\vec{u}$  be a vector consisting of unknown coefficients for  $u_h$ . The scheme (2.6) with flux (2.7) can be written as system of ordinary differential equations of the form

$$\frac{d}{dt} \mathbf{u} = \mathfrak{L}(\mathbf{u}), \tag{2.8}$$

where the operator  $\mathfrak{L}$  is nonlinear. To further approximate the solution of the system (2.8), one may use any ODE solver available in the literature. In [3], the third order TVD Runge-Kutta scheme [8] is used.

When  $\epsilon = 0$ , it is shown in [2] that one of the stationary solutions of (2.1) is given by

$$u(x) = \frac{4}{3} \left( \frac{\cosh(x/2)}{\cosh(p/2) - 1} \right),$$

which is periodic on the interval  $[-p, p]$ . When  $\epsilon$  is present, we use the non-homogeneous example in [7] with sinusoidal solution.

### 3. MODEL REDUCTION

The main concept of model order reduction is to construct a low-dimensional system that can provide accurate approximate solutions for the original high-dimensional system. Since many nonlinear partial differential equations in practical applications are often required to use high dimensional discretized system for obtaining accurate numerical solutions, the simulation can be extremely long. This motivates us to apply model reduction techniques to decrease the computational complexity.

Projection-based techniques, such as piecewise tangential interpolation [9], balanced truncation [10–14], and transfer function interpolation [15–17] are commonly used to obtain reduced-order systems. Most of these techniques are required to use a low-dimensional basis, which can be optimally constructed by an approach called proper orthogonal decomposition (POD). Due to its high efficiency in reducing the number of variables in the original systems, POD has been used extensively with the Galerkin projection in a number of works, such as in the analysis of network modeling [18, 19], biochemical reaction networks [20, 21], and flow dynamics [22, 23]. However, with the nonlinearity in the original system, POD may not truly reduce the computational complexity during the simulation of the resulting projected reduced system.

There are some existing model reduction techniques for nonlinear systems, such as Empirical Interpolation Method (EIM) [24], Trajectory Piecewise-Linear approach [25], Missing Point Estimation [26], and Discrete Empirical Interpolation Method (DEIM) [27]. This work focuses on POD and DEIM approaches, since it can reduce the computational complexity of general nonlinear term, including the one appearing in the discretized systems obtained from the local Discontinuous Galerkin method. DEIM 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 which selects the interpolation indices by heuristically minimizing the approximation error. The approach that combines POD and DEIM techniques has been used in many applications, such as non-linear miscible viscous fingering in porous media [28, 29], morphological structure spiking neurons [30], shallow-water equations [31], four-dimensional variational data assimilation [32], and nonlinear aeroelasticity model [33]. Most of these existing applications employed POD and DEIM with either finite element, finite difference or finite volume discretization of the corresponding dynamical systems. This work investigates the applicability of POD-DEIM approach on the discretized systems obtained from the local Discontinuous Galerkin method for Burgers-Poisson equation.

The details for each of POD and DEIM approaches, as well as the procedures for applying them to the local discontinuous Galerkin method are provided next.

#### 3.1. PROPER ORTHOGONAL DECOMPOSITION (POD)

POD is also known by other names, for example, Karhunen-Loève decomposition (KLD), principal component analysis (PCA), or singular value decomposition (SVD).

POD has been used with the Galerkin projection in many applications to construct a low dimensional subspace that captures dominant behavior in various applications, e.g. [34–36]. 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  $k$ . In general, POD basis is obtained by using singular value decomposition or the *method of snapshots* as discussed next.

Consider a set of snapshots  $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_s}\}$  where  $\mathbf{y}_j \in \mathbb{R}^N$ ,  $j = 1, 2, \dots, n_s$ . In general,  $\mathbf{y}_j$  may depend on certain parameter value or time instance. Suppose we want to approximate a snapshot  $\mathbf{y}_j$  by using a set of orthonormal vectors  $\{\phi_1, \phi_2, \dots, \phi_k\} \subset \mathbb{R}^N$ , which has rank  $k < N$ . Then the approximation is in the form

$$\mathbf{y}_j \approx \sum_{i=1}^k c_i \phi_i, \tag{3.1}$$

for some constant  $c_i$ ,  $i = 1, 2, \dots, k$ . Alternatively, we can write this approximation in matrix form as follow:

$$\mathbf{y}_j \approx \Phi_k \mathbf{c}, \tag{3.2}$$

where  $\Phi_k = [\phi_1, \phi_2, \dots, \phi_k] \in \mathbb{R}^{N \times k}$  is a matrix of basis vectors and  $\mathbf{c} = [c_1, c_2, \dots, c_k]^T \in \mathbb{R}^k$  is the vector of unknown coefficients. To find  $\mathbf{c}$ , we use the fact that  $\Phi$  has orthonormal columns, i.e.  $\Phi^T \Phi = \mathbf{I}$ , and the minimum error occurs when the residual is orthogonal to the column span of  $\Phi_k$ , i.e.

$$\Phi_k^T (\mathbf{y}_j - \Phi_k \mathbf{c}) = 0, \tag{3.3}$$

which implies that  $\mathbf{c} = \Phi_k^T \mathbf{y}_j$  and the approximation becomes

$$\mathbf{y}_j \approx \Phi_k \Phi_k^T \mathbf{y}_j. \tag{3.4}$$

Proper Orthogonal Decomposition provides an orthonormal basis that minimizes this approximation error in 2-norm for a given basis rank  $k \leq \text{rank}(\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_s}\})$ . In particular, POD basis is the optimal solution to the minimization problem (3.5) given in the definition below.

**Definition 3.1** (POD basis, [37]). Let  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_{n_s}] \in \mathbb{R}^{N \times n_s}$  be a snapshot matrix with rank  $r \leq \min\{N, n_s\}$ . POD basis of dimension  $k$ , where  $k \leq r$ , is the solution to the following optimization problem:

$$\min_{\Phi_k \in \mathbb{R}^{N \times k}} \sum_{j=1}^{n_s} \|\mathbf{y}_j - \Phi_k \Phi_k^T \mathbf{y}_j\|_2^2 \quad \text{such that} \quad \Phi_k^T \Phi_k = \mathbf{I}_k, \tag{3.5}$$

where  $\mathbf{I}_k \in \mathbb{R}^{k \times k}$  is the identity matrix.

It can be shown [37] that POD basis defined above can be obtained from the left singular vector of the snapshot matrix  $\mathbf{Y}$ . Let  $\mathbf{Y} = \mathbf{V} \Sigma \mathbf{W}^T$  be the singular value decomposition of  $\mathbf{Y}$ , where matrices  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{N \times r}$  and  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{n_s \times r}$  are matrices with orthonormal columns and  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  is a diagonal matrix with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ . Then the POD basis matrix of dimension  $k$  is  $\mathbf{V}_k = [\mathbf{v}_1, \dots, \mathbf{v}_k] \in \mathbb{R}^{N \times k}$ ,  $k \leq r$ , i.e.

$$\mathbf{V}_k = \arg \min_{\Phi_k \in \mathbb{R}^{N \times k}} \sum_{j=1}^{n_s} \|\mathbf{y}_j - \Phi_k \Phi_k^T \mathbf{y}_j\|_2^2 \quad \text{with} \quad \mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}_k. \tag{3.6}$$

It is well-known [37] that this minimum error is given by

$$\sum_{j=1}^{n_s} \|\mathbf{y}_j - \mathbf{V}_k \mathbf{V}_k^T \mathbf{y}_j\|_2^2 = \sum_{\ell=k+1}^r \sigma_\ell^2, \tag{3.7}$$

which is the sum of the neglected singular values  $\sigma_{k+1}, \dots, \sigma_r$  from the SVD of  $\mathbf{Y}$ . The algorithm for constructing a POD basis matrix by using SVD is summarized below.

---

**Algorithm 1** Algorithm for constructing POD basis

---

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

### 3.2. METHOD OF SNAPSHOTS

When the dimension  $N$  of snapshots is not too large, we can directly obtain the POD basis from the SVD of the snapshot matrix as shown in Algorithm 1. However, in practice,  $N$  can be extremely large and computing POD basis through SVD might not be efficient. In this case, many previous works have used a technique called **method of snapshots** described below.

Consider a snapshot matrix  $\mathbf{Y} \in \mathbb{R}^{N \times n_s}$  where  $N \gg n_s$ . As in the previous section, suppose the SVD of  $\mathbf{Y}$  is given by

$$\mathbf{Y} = \mathbf{V}\Sigma\mathbf{W}^T,$$

where matrices  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{N \times r}$  and  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{n_s \times r}$  are matrices with orthogonal columns and  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  is a diagonal matrix with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ . Consider also  $\mathbf{Y}^T \mathbf{Y} \in \mathbb{R}^{n_s \times n_s}$ :

$$\mathbf{Y}^T \mathbf{Y} = (\mathbf{V}\Sigma\mathbf{W}^T)^T (\mathbf{V}\Sigma\mathbf{W}^T) = \mathbf{W}\Sigma^2\mathbf{W}^T, \tag{3.8}$$

which implies that  $\mathbf{Y}^T \mathbf{Y} \mathbf{w}_i = \sigma_i^2 \mathbf{w}_i$ . i.e.  $(\sigma_i^2, \mathbf{w}_i)$  is an eigenpair of  $\mathbf{Y}^T \mathbf{Y}$  for  $i = 1, \dots, r$ . As a result, if we compute eigendecomposition of  $\mathbf{Y}^T \mathbf{Y}$  in the form

$$\mathbf{Y}^T \mathbf{Y} \tilde{\mathbf{Z}} = \tilde{\mathbf{Z}} \tilde{\mathbf{D}} \tag{3.9}$$

and use the fact that  $\mathbf{Y}^T \mathbf{Y}$  has rank  $r \leq n_s$ , we can set  $\tilde{\mathbf{Z}} = [\mathbf{W}, \tilde{\mathbf{W}}] \in \mathbb{R}^{n_s \times n_s}$ ,  $\tilde{\mathbf{D}} = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2, 0, \dots, 0) \in \mathbb{R}^{n_s \times n_s}$ , where  $\tilde{\mathbf{W}}$  is a matrix of size  $n_s \times (n_s - r)$  whose columns contain the eigenvectors of  $\mathbf{Y}^T \mathbf{Y}$  which are corresponding to zero eigenvalue. Using eigenpairs of  $\mathbf{Y}^T \mathbf{Y}$  with nonzero eigenvalues gives

$$\mathbf{Y}^T \mathbf{Y} = \mathbf{W} \mathbf{D} \mathbf{W}^T, \tag{3.10}$$

where  $\mathbf{D} = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2)$ . Note that  $\mathbf{Y}^T \mathbf{Y}$  is symmetric and any symmetric matrix has real eigenvalues and orthogonal eigenvectors. Notice that  $\Sigma = \mathbf{D}^{1/2}$ . From  $\mathbf{Y} = \mathbf{V}\Sigma\mathbf{W}^T$ , we can compute the POD basis  $\mathbf{V}$  by

$$\mathbf{V} = \mathbf{Y} \mathbf{W} \Sigma^{-1} = \mathbf{Y} \mathbf{W} \mathbf{D}^{-1/2}. \tag{3.11}$$

Algorithm 2 below is used to construct a POD basis matrix by using the method of snapshots.

---

**Algorithm 2** Algorithm for constructing POD basis: Method of Snapshots

---

- INPUT : Snapshots  $\{\mathbf{y}_j\}_{j=1}^{n_s} \subset \mathbb{R}^N$
  - OUTPUT : POD basis matrix  $\mathbf{V}_k \in \mathbb{R}^{N \times k}$
1. Create snapshot matrix :  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_{n_s}] \in \mathbb{R}^{N \times n_s}$  and let  $r = \text{rank}(\mathbf{Y})$
  2. Compute Eigendecomposition  $\mathbf{Y}^T \mathbf{Y}$  and form :  $\mathbf{Y}^T \mathbf{Y} = \mathbf{W} \mathbf{D} \mathbf{W}^T$  given in (3.10).
  3. Compute  $\mathbf{V} = \mathbf{Y} \mathbf{W} \mathbf{D}^{-1/2}$
  4. Choose dimension  $k \leq r$
  5. POD basis of rank  $k$  :  $\mathbf{V}_k = [\mathbf{v}_1, \dots, \mathbf{v}_k] = \mathbf{V}(:, 1 : k)$
- 

**Remark:** To specify the reduced dimension  $k$ , we can consider the following ratio

$$R := \frac{\sum_{\ell=1}^k \sigma_\ell^2}{\sum_{\ell=1}^r \sigma_\ell^2}. \tag{3.12}$$

In particular, if  $k = r$  is used, we have this ratio  $R = 1$ , which can be interpreted as capturing 100% of the characteristic of snapshot set. In many applications, we can use  $k$  that is much smaller than  $r$  to get  $R$  around 0.9 to 0.95, i.e. to get 90% to 95% of the overall features of the snapshot set.

### 3.3. AN APPLICATION OF POD TO THE LOCAL DISCONTINUOUS GALERKIN METHOD

Recall from the dynamical system obtained from the local discontinuous Galerkin method in (2.8):

$$\frac{d}{dt} \mathbf{u} = \mathfrak{L}(\mathbf{u}). \tag{3.13}$$

In practice, the variable  $\mathbf{u}$  may be in a high-dimensional subspace and solving for  $\mathbf{u}$  numerically could be time consuming. Proper orthogonal decomposition can be applied to above system of differential equations to reduce the number of unknowns as follows. Suppose  $\mathbf{V}_k$  is the POD basis of dimension  $k$  constructed from the snapshot solutions. Then we can approximate the solution  $\mathbf{u}$  by projecting on the subspace spanned by the  $k$ -dimensional POD basis vectors in  $\mathbf{V}_k$ , i.e.  $\mathbf{u} \approx \mathbf{V}_k \tilde{\mathbf{u}}$ . By applying the Galerkin projection to (2.8) on the space spanned by columns of  $\mathbf{V}_k$ , we obtain a POD reduced system:

$$\frac{d}{dt} \tilde{\mathbf{u}} = \mathbf{V}_k^T \mathfrak{L}(\mathbf{V}_k \tilde{\mathbf{u}}), \tag{3.14}$$

Note that POD basis is orthonormal and therefore  $\mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}$ .

Notice that, since  $\mathfrak{L}$  is nonlinear, computing the discretization of the term  $\mathbf{V}_k^T \mathfrak{L}(\mathbf{V}_k \tilde{\mathbf{u}})$  still depends on the dimension  $N$  of the original variable  $\mathbf{u}$ , although the number of variables in  $\tilde{\mathbf{u}}$  is now reduced to  $k$ . To overcome this inefficiency, we will apply an additional nonlinear model reduction approach, called Discrete empirical interpolation method.

### 3.4. DISCRETE EMPIRICAL INTERPOLATION METHOD (DEIM)

DEIM [27] is an efficient approach to reduce the complexity for evaluating the nonlinear term. To illustrate this issue, we consider again the nonlinearity in (3.14)

$$\mathbf{F}(t) = \mathfrak{L}(\mathbf{u}(t))$$

Let  $\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m\} \subset \mathbb{R}^n$  be the set of the nonlinear snapshots  $\mathbf{f}_j = \mathbf{F}(t_j) = \mathfrak{L}(\mathbf{u}(t_j))$  for all  $j = 1, 2, \dots, m$ , where  $\mathbf{u}(t_j)$  is already computed from (3.13). Suppose that  $\mathcal{F} = \text{span}\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m\}$  with  $\dim(\mathcal{F}) = r_f$ . We denote the nonlinear snapshot matrix with  $\bar{\mathbf{F}} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m] \in \mathbb{R}^{n \times m}$ . The SVD is then used on  $\bar{\mathbf{F}}$  to find the POD basis of rank  $l < r_f$  of the nonlinear term. In particular, assume that the SVD of  $\bar{\mathbf{F}}$  is  $\bar{\mathbf{F}} = \bar{\mathbf{V}}\bar{\Sigma}\bar{\mathbf{W}}^T$ , where  $\bar{\mathbf{V}} = [\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \dots, \bar{\mathbf{v}}_{r_f}] \in \mathbb{R}^{n \times r_f}$ ,  $\bar{\mathbf{W}} = [\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \dots, \bar{\mathbf{w}}_{r_f}] \in \mathbb{R}^{m \times r_f}$  and  $\bar{\Sigma} = \text{diag}(\bar{\sigma}_1, \bar{\sigma}_2, \dots, \bar{\sigma}_{r_f}) \in \mathbb{R}^{r_f \times r_f}$ . Thus, the POD basis of rank  $l$  of the nonlinear term is the first  $l$  columns of the matrix  $\bar{\mathbf{V}}$ , denoted by  $\bar{\mathbf{V}}_l$ . Then the nonlinear function  $\mathbf{F}(t)$  can be approximated by a subspace spanned by the basis  $\{\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \dots, \bar{\mathbf{v}}_l\}$ , which is of the form

$$\mathbf{F}(t) \approx \bar{\mathbf{V}}_l \mathbf{c}(t), \tag{3.15}$$

where  $\mathbf{c} : \mathcal{D} \rightarrow \mathbb{R}^l$  and  $\mathbf{c}(t)$  is the corresponding coefficient vector at the time  $t \in \mathcal{D}$ . The DEIM technique is applied here to specify  $\mathbf{c}(t)$  by selecting the  $l$  rows of (3.15). Let  $\mathbf{P}$  be a matrix used in the interpolation defined as  $\mathbf{P} = [\mathbf{e}_{\varphi_1}, \mathbf{e}_{\varphi_2}, \dots, \mathbf{e}_{\varphi_l}] \in \mathbb{R}^{n \times l}$ , where  $\mathbf{e}_{\varphi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T$  is the  $\varphi_i$  column of the identity matrix  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  for all  $i = 1, 2, \dots, l$ . By multiplying  $\mathbf{P}^T$  both sides of Eq. (3.15), the selection of components in the nonlinear term is made as follows

$$\mathbf{P}^T \mathbf{F}(t) \approx \underbrace{\mathbf{P}^T \bar{\mathbf{V}}_l}_{l \times l} \mathbf{c}(t). \tag{3.16}$$

Assume that  $\mathbf{P}^T \bar{\mathbf{V}}_l$  is a nonsingular matrix. Then  $\mathbf{c}(t)$  can be determined uniquely as  $\mathbf{c}(t) \approx (\mathbf{P}^T \bar{\mathbf{V}}_l)^{-1} \mathbf{P}^T \mathbf{F}(t)$ . As a result, the final approximation of (3.15) becomes

$$\mathbf{F}(t) \approx \bar{\mathbf{V}}_l (\mathbf{P}^T \bar{\mathbf{V}}_l)^{-1} \mathbf{P}^T \mathbf{F}(t). \tag{3.17}$$

The interpolation indices  $\varphi_1, \varphi_2, \dots, \varphi_l$  are generated by the DEIM algorithm shown in Algorithm 3.

---

#### Algorithm 3 DEIM

---

- INPUT :  $l < r_f$ ,  $\{\bar{\mathbf{v}}\}_{j=1}^{r_f} = \{\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \dots, \bar{\mathbf{v}}_{r_f}\} \subset \mathbb{R}^n$
  - OUTPUT :  $\mathbf{P} \in \mathbb{R}^{n \times l}$ ,  $\bar{\varphi}_l = [\varphi_1, \varphi_2, \dots, \varphi_l]^T \in \mathbb{R}^l$
  - 1.  $[\rho, \varphi_1] = \max\{|\bar{\mathbf{v}}_1|\}$
  - 2.  $\bar{\mathbf{V}} = [\bar{\mathbf{v}}_1]$ ,  $\bar{\mathbf{P}} = [\mathbf{e}_{\varphi_1}]$ ,  $\bar{\varphi} = [\varphi_1]$
  - 3. **for**  $j = 2 : r_f$  **do**
    - $\mathbf{c} = (\bar{\mathbf{P}}^T \bar{\mathbf{V}})^{-1} \bar{\mathbf{P}}^T \bar{\mathbf{v}}_j$
    - $\mathbf{r} = \bar{\mathbf{v}}_j - \bar{\mathbf{V}} \mathbf{c}$
    - $[\rho, \varphi_j] = \max\{|\mathbf{r}|\}$
    - $\bar{\mathbf{V}} \leftarrow [\bar{\mathbf{V}} \ \bar{\mathbf{v}}_j]$ ,  $\bar{\mathbf{P}} \leftarrow [\bar{\mathbf{P}} \ \mathbf{e}_{\varphi_j}]$ ,  $\bar{\varphi} \leftarrow \begin{bmatrix} \bar{\varphi} \\ \varphi_j \end{bmatrix}$
  - end for**
  - 4.  $\mathbf{P} = \bar{\mathbf{P}}(:, 1 : l)$ ,  $\bar{\varphi}_l = \bar{\varphi}(1 : l)$
-

From Algorithm 3, the procedure constructs a set of indices inductively on the input basis. The process starts from selecting the first interpolation index  $\wp_1 \in \{1, 2, \dots, n\}$  corresponding to the first input basis  $\bar{\mathbf{v}}_1$  entry which has the largest magnitude. The remaining indices  $\wp_j$  for  $j = 2, 3, \dots, l$  are selected from the entry of the residual  $\mathbf{r} = \bar{\mathbf{v}}_j - \bar{\mathbf{V}}\mathbf{c}$  with the largest magnitude. This algorithm also guarantees the invertibility of  $\mathbf{P}^T \bar{\mathbf{V}}_l$ . More details on this procedure can be found in [27].

The output matrix  $\mathbf{P}$  is employed to construct a low-dimensional approximation of the nonlinear term. Then the POD technique described in Section 3.1 is used in conjunction with the DEIM technique to construct a reduced-order system that is completely independent of the full dimension as shown below

$$\frac{d}{dt} \tilde{\mathbf{u}}(t) = \underbrace{\mathbf{V}_k^T \bar{\mathbf{V}}_l}_{k \times l} \underbrace{(\mathbf{P}^T \bar{\mathbf{V}}_l)^{-1}}_{l \times l} \underbrace{\mathbf{P}^T \mathcal{L}(\mathbf{V}_k \tilde{\mathbf{u}}(t))}_{l \times 1}. \tag{3.18}$$

Note that, the above equation uses the DEIM approximation in (3.17) by setting  $\mathbf{F}(t) = \mathcal{L}(\mathbf{V}_k \tilde{\mathbf{u}}(t))$ . An error bound of the DEIM approximation can be found in [27]. Next section demonstrates the numerical results obtained from the POD and POD-DEIM reduced order systems.

#### 4. RESULTS AND DISCUSSION

The accuracy of the solution from the reduced systems is measured through the average absolute and average relative errors computed, respectively, from the formulas:

$$\frac{1}{n_t} \sum_{j=1}^{n_t} \|\mathbf{u}_j - \mathbf{u}_j^{red}\|_2 \tag{4.1}$$

and

$$\frac{1}{n_t} \frac{\sum_{j=1}^{n_t} \|\mathbf{u}_j - \mathbf{u}_j^{red}\|_2}{\sum_{j=1}^{n_t} \|\mathbf{u}_j\|_2}, \tag{4.2}$$

where  $\mathbf{u}_j$  is the solution from the original discretized system and  $\mathbf{u}_j^{red}$  is the solution of the reduced-order system at time  $t_j$ , for  $j = 1, \dots, n_t$ . This section considers two numerical tests: non-parametrized and parametrized systems. The first one demonstrates the accuracy and efficiency of the POD and POD-DEIM approaches when compared with directly solving the original LDG discretized system. The second numerical test illustrates the applicability of the model reduction approaches when many parameter values have to be used to obtain numerical solutions.

##### 4.1. NON-PARAMETRIZED SYSTEM

In the following numerical tests, the original dimension of LDG discretized system is  $N = 500$ , i.e.  $\mathbf{u} \in \mathbb{R}^{500}$ .

Figure 1 illustrates the singular values of the solution snapshots. Notice from the decay of the singular values that using only first 10 POD basis vectors should capture most of the dynamics of the original system. Similar observation can be noticed from the decay of the singular values of the nonlinear snapshots shown in Figure 4.

Average absolute error (4.1) and relative error (4.2) of the solutions from POD reduced systems with different dimensions are shown in Figures 2. Notice that after using POD of dimension 10, there is no significant accuracy improvement, i.e. the errors seem to

be constant. This exactly follows from the decay of the singular values in Figure 1. Figure 3 shows the absolute error at the each grid points of the solutions from POD basis of dimensions 1 and 5. From Figure 3, the errors approximately decrease from order of  $10^{-11}$  to  $10^{-14}$  when the dimension of POD is increase from 1 to 5.

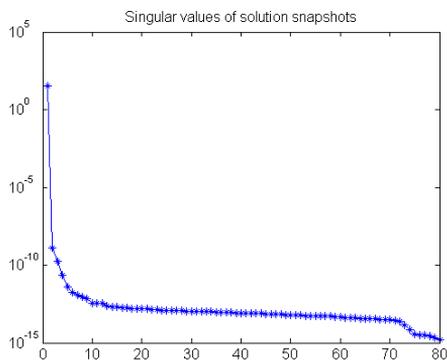


FIGURE 1. [Non-Parametrized System] Singular values corresponding to the POD basis constructed from 80 solution snapshots.

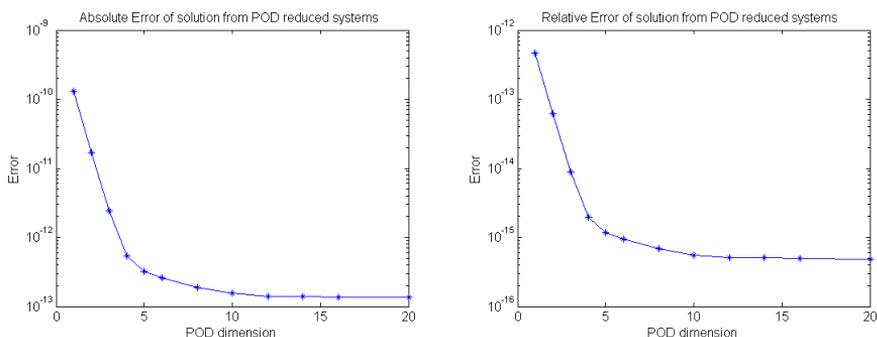


FIGURE 2. [Non-Parametrized System] Absolute and relative errors of the solutions from POD reduced system with different dimensions between 1 and 20.

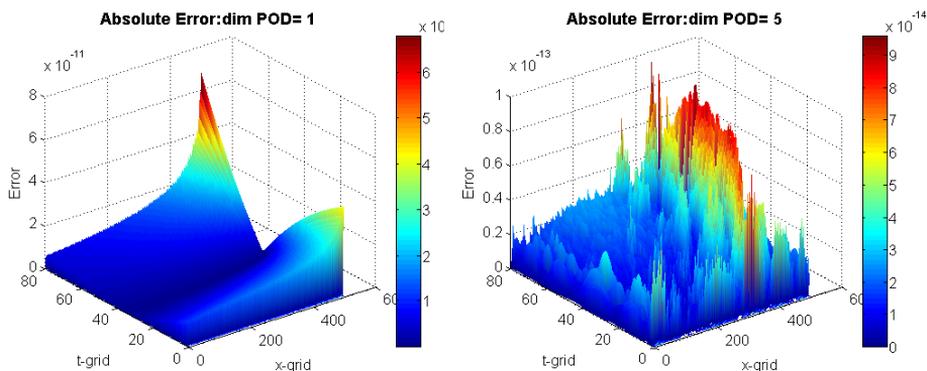


FIGURE 3. [Non-Parametrized System] Absolute error of the solutions from POD reduced system with dimension 5.

To further reduce the computational cost, DEIM is next applied on the nonlinear terms of the POD reduced system. Figure 5 shows the absolute errors at the grid points of the solutions from POD-DEIM reduced systems with dimension 1 and 5. To compare the accuracy of the solutions from POD and POD-DEIM reduced systems, we consider the average relative errors in Figure 6 and Table 1, which show the trends of errors as the reduced dimensions of the bases for POD and DEIM increase. Notice that the errors decay very fast at the beginning when the reduced dimensions are ranging from 1 to 5, and after that they become constant and converging to  $\mathcal{O}(10^{-13})$  and  $\mathcal{O}(10^{-12})$ , respectively, for the POD reduced system and the POD-DEIM reduced system. Notice also that, for small reduced dimensions, the accuracy of both POD and POD-DEIM reduced systems seems to be equivalent. For reduced dimensions larger than 3, the POD reduced system starts to be slightly more accurate than POD-DEIM reduced system. Table 1 also shows the CPU times for the POD and POD-DEIM reduced systems, which are scaled with the simulation time used for solving the full-order system. Notice that, for the error of order  $\mathcal{O}(10^{-12})$ , using the POD reduced system can reduce the computational time for solving the original systems by roughly a factor of 10, while using the POD-DEIM reduced system can reduce by roughly a factor of 300. Figure 7 considers 3 different fixed dimensions of POD with various DEIM dimensions. When the dimension of POD is fixed to be 1, increasing dimension of DEIM has no effect on the accuracy. However, when the dimension of POD is fixed to be 5 or 9, increasing dimension of DEIM can improve the approximate solution to some certain accuracy.

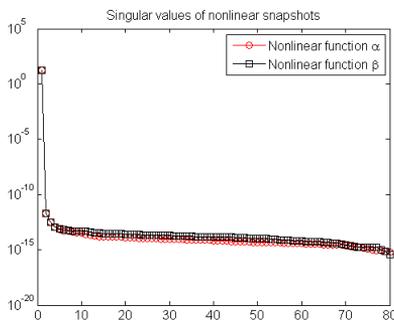


FIGURE 4. [Non-Parametrized System] Singular values corresponding to the POD basis (used in DEIM) constructed from 80 nonlinear solution snapshots.

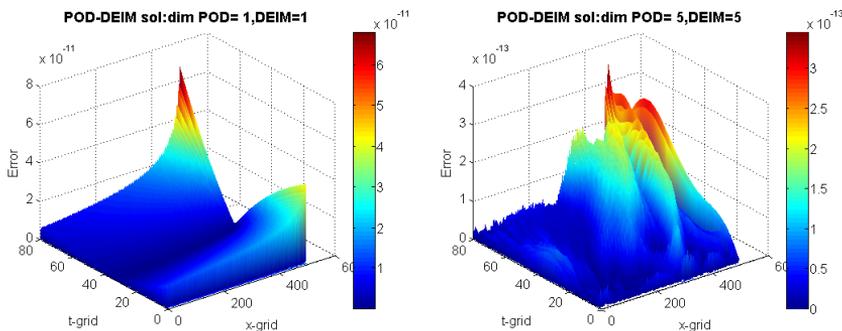


FIGURE 5. Absolute error of the solutions from POD-DEIM reduced system with dimension 5.

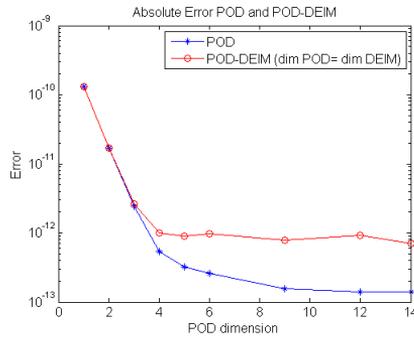


FIGURE 6. [Non-Parametrized System] Average absolute error of the solutions from POD-DEIM reduced system with different dimensions of POD and DEIM. Dimensions of POD and DEIM are the same for each reduced system.

Dimension of POD	Dimension of DEIM	Avg. Abs. Error POD system	Avg. Abs. Error POD-DEIM system	CPU time POD (scaled)	CPU time POD-DEIM (scaled)
1	1	$1.2927 \times 10^{-10}$	$1.2927 \times 10^{-10}$	1/11	1/321
2	2	$1.7047 \times 10^{-11}$	$1.7047 \times 10^{-11}$	1/10	1/318
3	3	$2.4411 \times 10^{-12}$	$2.6057 \times 10^{-12}$	1/9	1/313
4	4	$5.3830 \times 10^{-13}$	$9.8880 \times 10^{-13}$	1/9	1/308
5	5	$3.2666 \times 10^{-13}$	$9.0758 \times 10^{-13}$	1/8	1/291
9	9	$1.5765 \times 10^{-13}$	$7.8899 \times 10^{-13}$	1/7	1/283
14	14	$1.4003 \times 10^{-13}$	$7.0226 \times 10^{-13}$	1/5	1/281

TABLE 1. [Non-Parametrized System] Average absolute error of the solutions from POD-DEIM reduced system with different dimensions of POD and DEIM. The CPU times for both POD and POD-DEIM reduced systems are scaled with the simulation time used for solving the full-order system.

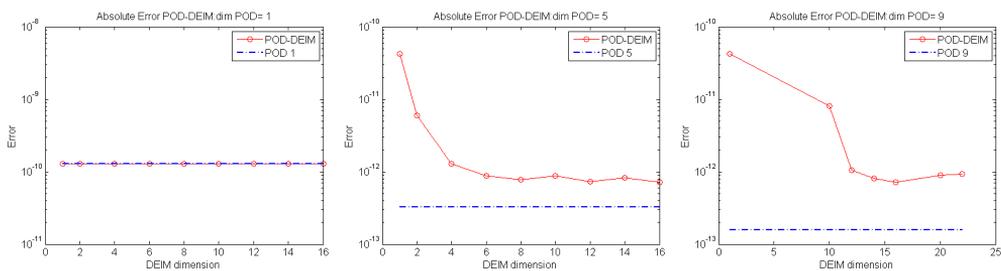


FIGURE 7. [Non-Parametrized System] Average absolute error of the solutions from POD-DEIM reduced system with different dimensions of DEIM with fixed POD dimension 1, 5, 9.

#### 4.2. MODEL REDUCTION FOR PARAMETRIZED SYSTEMS

The goal of this section is to generate one set of POD basis for state variable and one set of basis for each of nonlinear terms, which can be used for constructing many reduced order systems for different parameter values. In this case, we consider parameter  $\epsilon$  in the interval  $[0, 1]$ . We use solutions of the original full-order systems of dimension 120 with parameters  $\epsilon = 0, 0.5, 1$  to construct bases for POD and DEIM approximations.

The singular values of snapshots from the solution and the three nonlinear terms in the full-order discretized systems are shown in Figure 8. Figure 9 illustrates some of the corresponding POD basis vectors. The solution and the corresponding absolute error of the system with parameter  $\epsilon = 0$ , which is used as a training parameter value, are shown in Figure 10 from both POD and POD-DEIM reduced systems. Notice that the solutions of these reduced systems are visibly indistinguishable when compared with the original full-order discretized system. Therefore, the corresponding absolute errors at grid points are also provided in Figure 10, which shows that the POD approach is more accurate than POD-DEIM approach.

Figure 11 considers parameters values  $\epsilon = 0.01, 0.3, 0.7$ , which are not used in a training set of snapshots to generate reduced-order bases. In particular, Figure 11 shows the absolute errors at the grid points for the solutions corresponding to the systems with  $\epsilon = 0.01, 0.3, 0.7$  when using POD approach (with POD dimension equals to 3) and POD-DEIM approach (with POD dimension and DEIM dimension are 3). In these cases, it can be seen that errors for POD approach is roughly  $\mathcal{O}(10)$  times smaller than the error from POD-DEIM approach. However, as the dimensions of POD and DEIM increase, the errors for both POD and POD-DEIM approaches becomes equivalent as shown in Figure 12. In particular, Figure 12 displays average absolute errors computed from (4.1) for the solutions of the POD and POD-DEIM reduced systems with parameter  $\epsilon = 0, 0.3, 0.7$  when different dimension of POD and DEIM are used. These plots not only confirm the convergence of the approximate reduced-order solution, but also demonstrates that DEIM approach can further reduce the computational complexity of the nonlinear term without scarifying the accuracy of the numerical solutions.

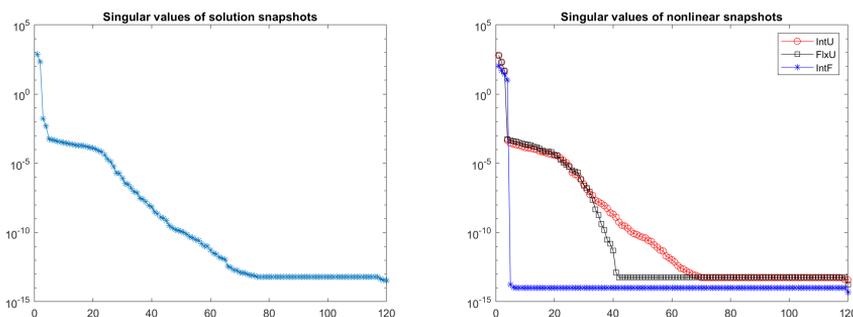


FIGURE 8. [Parametrized System] Singular values of solution snapshots and nonlinear snapshots from the original full-order systems with parameters  $\epsilon = 0, 0.5, 1$ .

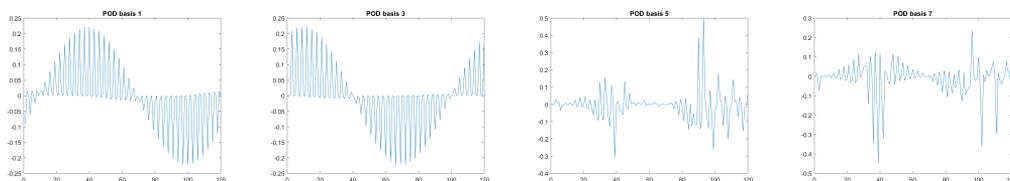


FIGURE 9. [Parametrized System] The first, third, fifth, and seventh POD basis vectors of solution snapshots.

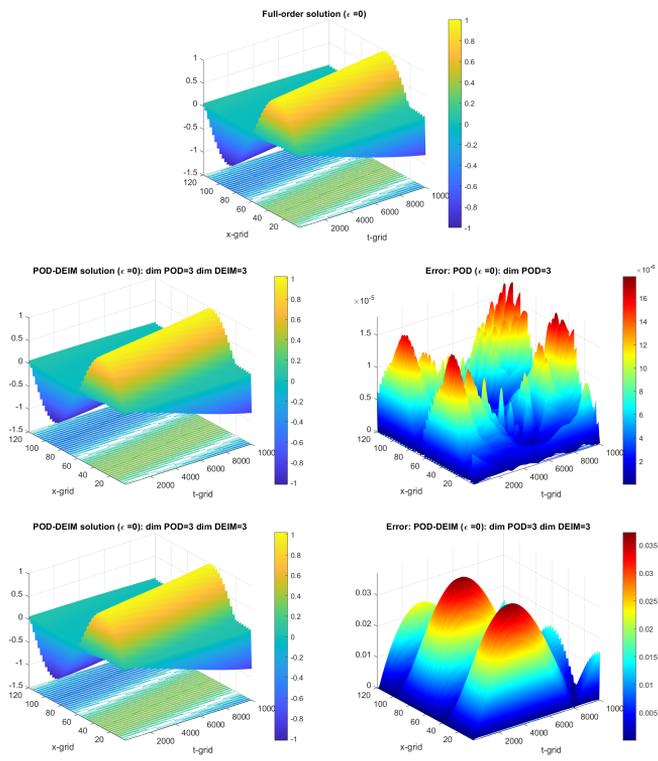


FIGURE 10. [Parametrized System] Approximate solutions Of the original system with parameter  $\epsilon = 0$  (which is used as a training parameter value) by using POD reduced system with POD dimension = 3 and the POD-DEIM reduced system with POD dimension =3 and DEIM dimension = 3.

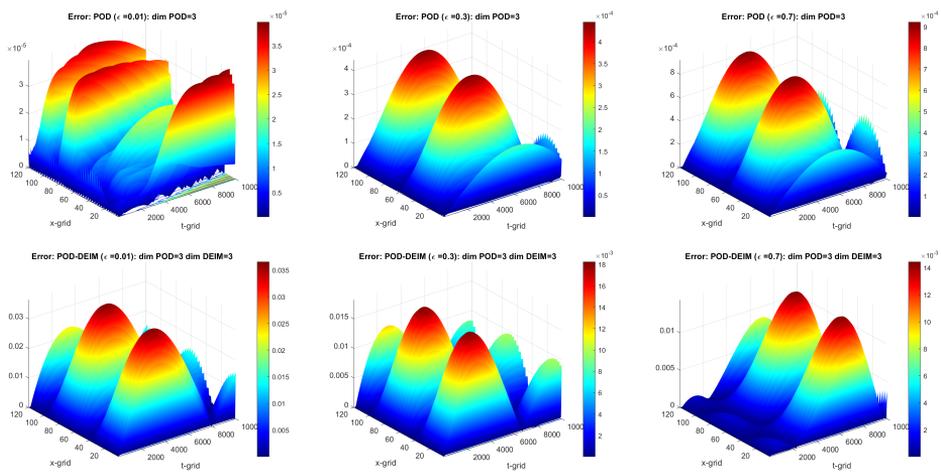


FIGURE 11. [Parametrized System] Absolute errors at the grid points of the POD reduced system ( POD dimension = 3 ) and POD-DEIM reduced system (POD dimension =3 and DEIM dimension = 3) with parameter  $\epsilon = 0.01, 0.3, 0.7$ .

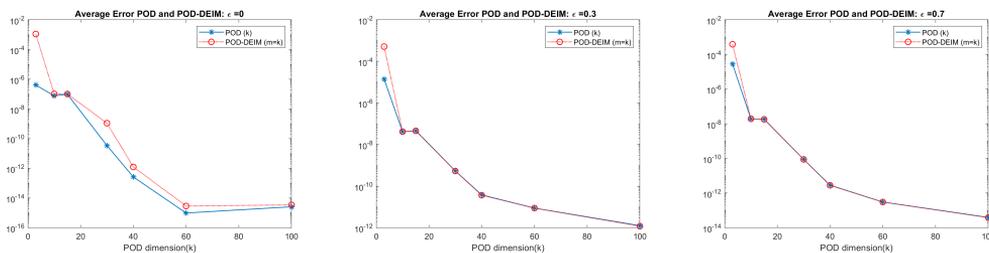


FIGURE 12. [Parametrized System] Average absolute error (4.1) of the POD and POD-DEIM reduced systems for parameter  $\epsilon = 0, 0.3, 0.7$  using different dimension of POD basis. The dimension of DEIM used in POD-DEIM approach is the same as the dimension of POD for each of these cases (i.e.  $m = k$ ).

### 5. CONCLUSIONS

This work demonstrates how to construct a cost-efficient reduced system for approximating the solutions of Burgers-Poisson equation. A local discontinuous Galerkin method is used as a starting point to generate a semi-discretized system. POD is then used with DEIM to decrease both the number of variables and the computational complexity. The numerical experiments illustrate that the resulting reduced systems can accurately approximate the solutions of Burgers-Poisson equation with substantially smaller dimension. In particular, when the dimension is reduced by a factor of 1/100, the corresponding error is shown to be less than the order of  $\mathcal{O}(10^{-12})$ . These results reflect the possibility of using POD and DEIM to reduce computational cost for more complicated dynamical systems that require discontinuous Galerkin method to obtain realistic numerical solutions.

### ACKNOWLEDGEMENTS

We would like to thank the referees for their comments and suggestions on the manuscript. The first author is supported by Chiang Mai University and the Centre of Excellence in Mathematics, The Commission on Higher Education, Thailand. The second author is support by Thammasat University Research Fund, Contract No. TUGR 2/12/2562 and The Thailand Research Fund (TRF) grant for new researcher: Grant No. TRG5880216.

### REFERENCES

- [1] G. Whitham, *Linear and nonlinear waves*. Modern Book Incorporated, 1975.
- [2] C. Schmeiser and K. Fellner, “Burgers–poisson: A nonlinear dispersive model equation,” *SIAM Journal on Applied Mathematics*, vol. 64, no. 5, pp. 1509–1525, 2004.
- [3] H. Liu and N. Ploymaklam, “A local discontinuous galerkin method for the burgers–poisson equation,” *Numerische Mathematik*, vol. 129, no. 2, pp. 321–351, 2015.
- [4] W. H. Reed and T. Hill, “Triangular mesh methods for the neutron transport equation,” tech. rep., Los Alamos Scientific Lab., N. Mex.(USA), 1973.
- [5] B. Cockburn and C.-W. Shu, “Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework,” *Mathematics of computation*, vol. 52, no. 186, pp. 411–435, 1989.

- [6] B. Cockburn and C.-W. Shu, “The local discontinuous galerkin method for time-dependent convection-diffusion systems,” *SIAM Journal on Numerical Analysis*, vol. 35, no. 6, pp. 2440–2463, 1998.
- [7] N. Ploymaklam, P. M. Kumbhar, and A. K. Pani, “A priori error analysis of the local discontinuous galerkin method for the viscous burgers-poisson system,” *Int. J. Numer. Anal. Model*, vol. 14, no. 4-5, pp. 784–807, 2017.
- [8] S. Gottlieb and C.-W. Shu, “Total variation diminishing runge-kutta schemes,” *Mathematics of computation of the American Mathematical Society*, vol. 67, no. 221, pp. 73–85, 1998.
- [9] X. Yang and J. Zheng, “Curvature tensor computation by piecewise surface interpolation,” *Computer-Aided Design*, vol. 45, no. 12, pp. 1639 – 1650, 2013.
- [10] H. Zhang, L. Wu, P. Shi, and Y. Zhao, “Balanced truncation approach to model reduction of markovian jump time-varying delay systems,” *Journal of the Franklin Institute*, no. 0, pp. –, 2015.
- [11] N. Lang, J. Saak, and T. Stykel, “Towards practical implementations of balanced truncation for {LTV} systems,” *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 7 – 8, 2015.
- [12] M. Presho, A. Protasov, and E. Gildin, “Localglobal model reduction of parameter-dependent, single-phase flow models via balanced truncation,” *Journal of Computational and Applied Mathematics*, vol. 271, no. 0, pp. 163 – 179, 2014.
- [13] H. B. Minh, C. Batlle, and E. Fossas, “A new estimation of the lower error bound in balanced truncation method,” *Automatica*, vol. 50, no. 8, pp. 2196 – 2198, 2014.
- [14] S. Ghosh and N. Senroy, “Balanced truncation based reduced order modeling of wind farm,” *International Journal of Electrical Power and Energy Systems*, vol. 53, no. 0, pp. 649 – 655, 2013.
- [15] P. Runkle, M. Blommer, and G. Wakefield, “A comparison of head related transfer function interpolation methods,” in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, pp. 88–91, Oct 1995.
- [16] J. Duffin, O. Sobczyk, A. Crawley, J. Poublanc, D. Mikulis, and J. Fisher, “The dynamics of cerebrovascular reactivity shown with transfer function analysis,” *NeuroImage*, vol. 114, no. 0, pp. 207 – 216, 2015.
- [17] K. J. Fink and L. Ray, “Individualization of head related transfer functions using principal component analysis,” *Applied Acoustics*, vol. 87, no. 0, pp. 162 – 173, 2015.
- [18] X. Ye, P. Li, M. Zhao, R. Panda, and J. 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*, vol. 29, pp. 1342–1353, Sept 2010.
- [19] G. Jasmon and L. Lee, “Distribution network reduction for voltage stability analysis and loadflow calculations,” *International Journal of Electrical Power and Energy Systems*, vol. 13, no. 1, pp. 9 – 13, 1991.
- [20] S. Rao, A. van der Schaft, K. van Eunen, B. M. Bakker, and B. Jayawardhana, “Model-order reduction of biochemical reaction networks,” *CoRR*, vol. abs/1212.2438, 2012.
- [21] C. A. Pantea, *Mathematical and Computational Analysis of Biochemical Reaction Networks*. PhD thesis, University of Wisconsin at Madison, Madison, WI, USA, 2010. AAI3437121.
- [22] 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*, vol. 27, no. 6, pp. 787–815, 2013.
- [23] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, “The {GNAT} method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows,” *Journal of Computational Physics*, vol. 242, no. 0, pp. 623 – 647, 2013.
- [24] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, “An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations,” *Comptes Rendus Mathématique*, vol. 339, no. 9, pp. 667 – 672, 2004.
- [25] T. Vob, R. Pulch, E. ter Maten, and A. El Guennoui, “Trajectory piecewise linear approach for nonlinear differential-algebraic equations in circuit simulation,” in *Scientific Computing in Electrical Engineering* (G. Ciuprina and D. Ioan, eds.), vol. 11 of *Mathematics in Industry*, pp. 167–173, Springer Berlin Heidelberg, 2007.
- [26] A. Vendl and H. Fabender, “Projection-based model order reduction for steady aerodynamics,” in *Computational Flight Testing* (N. Kroll, R. Radespiel, J. W. Burg, and K. Srensen, eds.), vol. 123 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pp. 151–166, Springer Berlin Heidelberg, 2013.
- [27] S. Chaturantabut and D. C. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM J. Sci. Comput.*, vol. 32, pp. 2737–2764, Sept. 2010.
- [28] S. Chaturantabut and D. 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*, vol. 17, pp. 337–353, 2011.
- [29] N. Sukuntee and S. Chaturantabut, “Parametric nonlinear model reduction using-means clustering for miscible flow simulation,” *Journal of Applied Mathematics*, vol. 2020.
- [30] A. R. Kellems, S. Chaturantabut, and S. J. Cox, “Morphologically accurate reduced order modeling of spiking neurons,” *Journal of Computational Neuroscience*, vol. 28, pp. 477–494, 2010.
- [31] R. Ștefănescu and I. M. Navon, “POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model,” *Journal of Computational Physics*, vol. 237, pp. 95–114, Mar. 2013.
- [32] R. Stefanescu, A. Sandu, and I. M. Navon, “POD/DEIM strategies for reduced data assimilation systems,” *CoRR*, vol. abs/1402.5992, 2014.
- [33] Z. Feng and A. Soulaïmani, “Reduced order modelling based on pod method for 3d nonlinear aeroelasticity,” in *The 18th IASTED International Conference on Modelling and Simulation*, MS ’07, (Anaheim, CA, USA), pp. 489–494, ACTA Press, 2007.
- [34] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual Rev. Fluid Mech*, pp. 539–575, 1993.
- [35] E. Schenone, *Reduced Order Models, Forward and Inverse Problems in Cardiac Electrophysiology*. Theses, Université Pierre et Marie Curie - Paris VI, Nov. 2014.
- [36] R. Gurka, A. Liberzon, and G. Hetsroni, “{POD} of vorticity fields: A method for spatial characterization of coherent structures,” *International Journal of Heat and Fluid Flow*, vol. 27, no. 3, pp. 416 – 423, 2006.
- [37] S. Volkwein, *Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling*. Lecture notes, University of Konstanz.