

## Research Article

# Parametric Nonlinear Model Reduction Using $K$ -Means Clustering for Miscible Flow Simulation

Norapon Sukuntee and Saifon Chaturantabut 

*Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani 12120, Thailand*

Correspondence should be addressed to Saifon Chaturantabut; saifon@mathstat.sci.tu.ac.th

Received 14 April 2020; Revised 7 August 2020; Accepted 12 August 2020; Published 4 September 2020

Academic Editor: Yansheng Liu

Copyright © 2020 Norapon Sukuntee and Saifon Chaturantabut. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work considers the model order reduction approach for parametrized viscous fingering in a horizontal flow through a 2D porous media domain. A technique for constructing an optimal low-dimensional basis for a multidimensional parameter domain is introduced by combining  $K$ -means clustering with proper orthogonal decomposition (POD). In particular, we first randomly generate parameter vectors in multidimensional parameter domain of interest. Next, we perform the  $K$ -means clustering algorithm on these parameter vectors to find the centroids. POD basis is then generated from the solutions of the parametrized systems corresponding to these parameter centroids. The resulting POD basis is then used with Galerkin projection to construct reduced-order systems for various parameter vectors in the given domain together with applying the discrete empirical interpolation method (DEIM) to further reduce the computational complexity in nonlinear terms of the miscible flow model. The numerical results with varying different parameters are demonstrated to be efficient in decreasing simulation time while maintaining accuracy compared to the full-order model for various parameter values.

## 1. Introduction

Viscous fingering is the formation of patterns in a morphologically unstable interface between two fluids in a porous medium. It generally refers to the onset and evolution of instabilities that occur in the displacement of fluids [1]. The mathematical description of viscous fingering can be given in terms of several partial differential equations. Many numerical methods are used to find the approximation of pressure, velocity, concentration, or temperature. In [2], the coupled set of partial differential equations is approximated by the semidiscrete SUPG stabilized finite element formulation plus a discontinuity capturing technique to improve stability around the moving sharp fronts. The pressure equation is discretized by the standard Galerkin method, and a post-processing scheme is used to improve the numerical evaluation of Darcy's velocity. Branston et al. developed a finite volume numerical simulator with a level of certainty where fingering perturbations in stochastic fields do not result into

severe channeling as discussed in [3]. The simulator development involves discretization of convection–diffusion and pressure equations with total variation diminishing and central differencing schemes, respectively. Correlated stochastic fields for simulation runs were created using convolution with a Gaussian filter achieved through 2D fast and inverse Fourier transform, Dykstra–Parsons coefficient, random porosity generator, and autocorrelation lengths. In [4], the Oldroyd model has been studied using a generalized incompressible Navier-Stokes equation via the interpolating stabilized element free Galerkin technique. In this work, the Crank-Nicolson method is first utilized to discrete the temporal direction, and then, the meshless element free Galerkin technique is employed to approximate the spatial direction. The equation contains the integral term which is approximated by using a difference scheme. Although the numerical methods discussed earlier are effective and accurate, the resulting simulations generally require enormous memory storage and consume excessively computational time. In

order to manage this problem, model order reduction techniques have been proposed in many previous works for various applications.

Proper orthogonal decomposition (POD) is a commonly used technique for model order reduction. POD is used to generate the optimal basis that can describe the dominant features of a dynamical system. This basis is utilized with the Galerkin projection to construct a reduced-order model. As in [5], a surrogate model based on the proper orthogonal decomposition is developed in order to enable fast and reliable evaluations of aerodynamic fields. In 2017, Chang et al. introduced a reduced-order model for wall shear stress in abdominal aortic aneurysms by proper orthogonal decomposition [6]. A more interesting detail on POD can be found in [7–9].

Using POD still has a limitation when the model has complicated nonlinear property. POD cannot handle the complexity issue arising in computing nonlinear terms. A powerful technique called the discrete empirical interpolation method (DEIM) is used here to overcome this problem. In particular, the DEIM is employed to construct selected interpolation indices, which will allow much less expensive evaluation for the nonlinear terms. The DEIM can be used to combine with POD to improve the efficiency of the nonlinear reduced-order model. In [10], the POD-DEIM approach has been applied to the groundwater equation and the time-dependent incompressible Navier-Stokes equation with variable density based on the meshfree RBF-FD technique. In [11], POD-DEIM is applied to a dynamic 2D catalytic reactor model. There are also many other applications of POD-DEIM on various problems, which can be found, for example, in [12–14].

In this paper, we apply the POD-DEIM technique on a finite difference discretization of a nonlinear miscible flow with viscous fingering in porous media. This model contains many parameters which can be changed in many different situations. Our additional effort is to construct a reduced-order model, which can be used to approximate the solution with any given parameter in the domain of interest. This issue in model reduction is crucial for various parametrized dynamical systems. There are a number of works that have been done for parametric model order reduction (PMOR). In [15], the PMOR method for laminated composite structures is presented. The parametrization is carried out for the angle between the layers of structure and the thickness of the structure. In [16], this work proposed the PMOR scheme for second-order systems that does not require a priori sampling of the parameter space. For the flow model, PMOR for probabilistic analysis of unsteady aerodynamic applications is discussed in [17]. Also in [18], the PMOR approach is applied to the fluid flow governed by the Navier-Stokes equations at varying Reynolds number. In this work, we use  $K$ -means clustering to efficiently construct a reduced system for a parametrized miscible flow model.

This paper is arranged as follows. We first formulate the model equations in Section 2. In Section 3, a general setup for constructing POD-DEIM reduced systems is discussed. Section 4 introduces a parametrized nonlinear model reduction using  $K$ -means with the POD-DEIM approach for the

miscible flow model. In Section 5, the numerical experiments are used to illustrate the efficiency of the proposed approach. We finally conclude the results and discuss some future research directions in Section 6.

## 2. Problem Formulation

We consider a viscous fingering in a horizontal flow through a 2D porous media domain [19] with a constant permeability  $K$  given by

$$\nabla \cdot \mathbf{u} = 1, \quad (1)$$

$$\nabla P = -\frac{\mu}{K} \mathbf{u}, \quad (2)$$

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = D \nabla^2 c + f(c), \quad (3)$$

$$\rho c_p \left[ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right] = D_T \nabla^2 T + (-\Delta H) f(c), \quad (4)$$

where  $\mathbf{u} = [u_1, u_2] \in \mathbb{R}^2$  is the velocity with components in  $x$  and  $y$  coordinates,  $P$  is the global pressure,  $c$  is the concentration of the injected fluid,  $T$  is the temperature,  $\mu$  is the viscosity depending on  $c$  and  $T$ ,  $f(c)$  is the reaction rate, and  $\rho$ ,  $c_p$ ,  $D$ ,  $D_T$ , and  $\Delta H$  are assumed to be constant. To solve (1)–(4), a general procedure nondimensionalizes the system which will be converted into the equations in terms of stream function  $\psi$  and vorticity  $\omega$  as follows:

$$\nabla^2 \psi = -\omega, \quad (5)$$

$$\begin{aligned} \omega = & -R_c \left[ \psi_x c_x + \psi_y c_y + c_y \right] \\ & + R_T \left[ \psi_x T_x + \psi_y T_y + T_y \right], \end{aligned} \quad (6)$$

$$\frac{\partial c}{\partial t} + \psi_y c_x - \psi_x c_y = \nabla^2 c + \text{Daf}(c), \quad (7)$$

$$\frac{\partial T}{\partial t} + \psi_y T_x - \psi_x T_y = \text{Le} \nabla^2 T + \text{sgn}(\phi) \text{Daf}(c), \quad (8)$$

where  $R_c$  and  $R_T$  are constant parameters determining the effects of concentration and temperature to the viscosity,  $\text{Da}$  (Damköhler number) and  $\text{Le}$  (Lewis number) are dimensionless parameters,

$$\text{sgn}(\phi) = \begin{cases} 1; & \text{exothermic,} \\ -1; & \text{endothermic,} \end{cases} \quad (9)$$

and  $f(c) = -c(c-1)(c+d)$ . These equations (5)–(8) are defined on the spatial domain  $(x, y) \in \Omega = [0, \alpha \text{Pe}] \times [0, \text{Pe}] \subset \mathbb{R}^2$ , where  $\text{Pe}$  (Péclet number) is a dimensionless constant aspect ratio and time domain  $t \in [0, t_f] \subset \mathbb{R}$ . For this system, the boundary conditions are

$$\begin{aligned}
\psi(x, 0, t) &= \psi(x, \text{Pe}, t), \\
\psi_y(x, 0, t) &= \psi_y(x, \text{Pe}, t), \\
c(x, 0, t) &= c(x, \text{Pe}, t), \\
c_y(x, 0, t) &= c_y(x, \text{Pe}, t), \\
T(x, 0, t) &= T(x, \text{Pe}, t), \\
T_y(x, 0, t) &= T_y(x, \text{Pe}, t),
\end{aligned} \tag{10}$$

$$\mathbf{A}(\mathbf{p}) = \begin{bmatrix} -A & -\mathbf{I}_n & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\mathbf{I}_n & -R_c A_y & R_T A_y \\ \mathbf{O} & \mathbf{O} & A & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \text{Le}A \end{bmatrix} \in \mathbb{R}^{4n \times 4n} \tag{14}$$

along top and bottom sides. The boundary conditions are

$$\begin{aligned}
\psi(0, y, t) &= 0, \\
\psi(\alpha \text{Pe}, y, t) &= 0, \\
c(0, y, t) &= 1, \\
c(\alpha \text{Pe}, y, t) &= 0, \\
T(0, y, t) &= \begin{cases} 1; & \text{exothermic,} \\ 0; & \text{endothermic,} \end{cases} \\
T(\alpha \text{Pe}, y, t) &= \begin{cases} 0; & \text{exothermic,} \\ 1; & \text{endothermic,} \end{cases}
\end{aligned} \tag{11}$$

along the remaining two sides. The initial conditions are

$$c(x, y, 0) = T(x, y, 0) = \begin{cases} 1; & x \leq \hat{x} \\ 0; & x > \hat{x} \end{cases} \text{ for all } y \in [0, \text{Pe}], \tag{12}$$

where  $\hat{x} = \alpha \text{Pe}/2$  is the interface location. Let  $0 = x_0 < x_1 < \dots < x_{n_x} < x_{n_x+1} = \alpha \text{Pe}$  and  $0 = y_0 < y_1 < \dots < y_{n_y} < y_{n_y+1} = \text{Pe}$  be equally spaced grid points on  $x$ -axis and  $y$ -axis with  $\Delta x = \alpha \text{Pe}/(n_x + 1)$  and  $\Delta y = \text{Pe}/(n_y + 1)$ , respectively. Define  $\boldsymbol{\psi}(t)$ ,  $\boldsymbol{\omega}(t)$ ,  $\mathbf{c}(t)$ , and  $\mathbf{T}(t)$  as a vector of dimension  $n = n_x n_y$  containing approximate solutions  $\psi(x_i, y_j, t)$ ,  $\omega(x_i, y_j, t)$ ,  $c(x_i, y_j, t)$ , and  $T(x_i, y_j, t)$ , respectively, for  $i = 1, 2, \dots, n_x$  and  $j = 1, 2, \dots, n_y$ . The central finite difference is applied to construct the spatial discretization of (5)–(8), and it becomes a system of nonlinear ODEs in the form of

$$\mathbf{M}(\mathbf{p}) \frac{d\mathbf{z}(t)}{dt} = \mathbf{A}(\mathbf{p})\mathbf{z}(t) + \mathbf{d}(\mathbf{p}) + \mathbf{E}(\mathbf{p}; \mathbf{z}(t), t), \tag{13}$$

where

$$\mathbf{M}(\mathbf{p}) = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}_n & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}_n \end{bmatrix},$$

are parametrized constant matrices;

$$\mathbf{d}(\mathbf{p}) = \begin{bmatrix} \mathbf{O} \\ \mathbf{O} \\ \mathbf{b} \\ \text{Le}\mathbf{b} \end{bmatrix} \in \mathbb{R}^{4n} \tag{15}$$

is a parametrized constant vector;  $\mathbf{E}(\mathbf{p}; \cdot, \cdot): \mathbb{R}^{4n} \times \mathbb{R} \rightarrow \mathbb{R}^{4n}$  is a nonlinear function with

$$\mathbf{E}(\mathbf{p}; \mathbf{z}(t), t) = \begin{bmatrix} \mathbf{O} \\ -R_c \mathbf{N}(\boldsymbol{\psi}(t), \mathbf{c}(t)) + R_T \mathbf{N}(\boldsymbol{\psi}(t), \mathbf{T}(t)) \\ -\mathbf{F}(\boldsymbol{\psi}(t), \mathbf{c}(t)) + \text{Daf}(\mathbf{c}(t)) \\ -\mathbf{F}(\boldsymbol{\psi}(t), \mathbf{T}(t)) + \text{sign}(\phi) \text{Daf}(\mathbf{c}(t)) \end{bmatrix}; \tag{16}$$

$$\mathbf{z}(t) = [\boldsymbol{\psi}(t); \boldsymbol{\omega}(t); \mathbf{c}(t); \mathbf{T}(t)] \in \mathbb{R}^{4n}$$

is the vector of state variables, and  $\mathbf{p} = [R_c; R_T; \text{Da}; \text{Le}; \text{Pe}]$  is the vector of parameters. Note that  $A, A_x, A_y \in \mathbb{R}^{n \times n}$  are constant coefficient matrices,  $\mathbf{O} \in \mathbb{R}^{n \times n}$  is the zero matrix,  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  is the identity matrix,  $\mathbf{b}, \mathbf{b}_x \in \mathbb{R}^n$  are vectors indicating the boundary conditions, and  $\mathbf{N}, \mathbf{F}$  are nonlinear functions that maps some state variables to  $\mathbb{R}^n$ . We strongly recommend that the reader should read more details in the derivation of (13) in Appendix A. The solution  $\mathbf{z}$  will generally depend on the parameter vector  $\mathbf{p}$ , and therefore, we will also denote  $\mathbf{z}(t)$  as  $\mathbf{z}(\mathbf{p}; t) = [\boldsymbol{\psi}(\mathbf{p}; t); \boldsymbol{\omega}(\mathbf{p}; t); \mathbf{c}(\mathbf{p}; t); \mathbf{T}(\mathbf{p}; t)]$ . More details on discretization and numerical schemes can be found in [19, 20].

### 3. Reduced-Order Modeling by the POD-DEIM Approach

This section provides a brief overview of the general procedure for constructing a reduced-order model of a differential equation using proper orthogonal decomposition (POD) and a discrete empirical interpolation method (DEIM).

**3.1. Subspace from Proper Orthogonal Decomposition.** Consider a matrix of snapshots  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{n_s}] \in \mathbb{R}^{n \times n_s}$ , where  $\mathbf{z}_j \in \mathbb{R}^n$ ,  $j = 1, 2, \dots, n_s$ . Suppose we want to approximate a snapshot  $\mathbf{z}_j$  by using a set of orthonormal vectors  $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2,$

$\dots, \Phi_k\} \subset \mathbb{R}^n$ , which has rank  $k < n$ . Then, the approximation is in the form

$$\mathbf{z}_j \approx \Phi_k \mathbf{q}, \quad (17)$$

where  $\Phi_k = [\Phi_1, \Phi_2, \dots, \Phi_k] \in \mathbb{R}^{n \times k}$  is a matrix of basis vectors and  $\mathbf{q} = [q_1, q_2, \dots, q_k]^T \in \mathbb{R}^k$  is the vector of unknown coefficients. To find  $\mathbf{q}$ , we use the fact that  $\Phi_k$  has orthonormal columns, i.e.,  $\Phi_k^T \Phi_k = \mathbf{I}_k$ , and the minimum error occurs when the residual is orthogonal to the column span of  $\Phi_k$ , i.e.,  $\Phi_k^T (\mathbf{z}_j - \Phi_k \mathbf{q}) = 0$ , which implies that  $\mathbf{q} = \Phi_k^T \mathbf{z}_j$  and the approximation becomes

$$\mathbf{z}_j \approx \Phi_k \Phi_k^T \mathbf{z}_j. \quad (18)$$

Proper orthogonal decomposition (POD) provides an orthonormal basis that minimizes this approximation error in 2-norm for a given basis rank  $k \leq k_* := \text{rank}(\mathbf{Z})$ ; i.e., POD basis is the optimal solution to the minimization problem

$$\min_{\Phi_k \in \mathbb{R}^{n \times k}} \sum_{j=1}^{n_s} \|\mathbf{z}_j - \Phi_k \Phi_k^T \mathbf{z}_j\|_2^2 \text{ such that } \Phi_k^T \Phi_k = \mathbf{I}_k, \quad (19)$$

where  $\mathbf{I}_k \in \mathbb{R}^{k \times k}$  is the identity matrix. It can be shown [21] that the POD basis defined above can be obtained from the left singular vector of the snapshot matrix  $\mathbf{Z}$ . Let  $\mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}^T$  be the singular value decomposition of  $\mathbf{Z}$ , where matrices  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{k_*}] \in \mathbb{R}^{n \times k_*}$  and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{k_*}] \in \mathbb{R}^{n_s \times k_*}$  are matrices with orthonormal columns and  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{k_*}) \in \mathbb{R}^{k_* \times k_*}$  is a diagonal matrix with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{k_*} > 0$ . Then, the POD basis matrix of dimension  $k$  is  $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{n \times k}$ ,  $k \leq k_*$ . Moreover, this minimum error is given by [21] the sum of neglected singular values squared, i.e.,

$$\min_{\Phi_k \in \mathbb{R}^{n \times k}} \sum_{j=1}^{n_s} \|\mathbf{z}_j - \Phi_k \Phi_k^T \mathbf{z}_j\|_2^2 = \sum_{j=1}^{n_s} \|\mathbf{z}_j - \mathbf{U}_k \mathbf{U}_k^T \mathbf{z}_j\|_2^2 = \sum_{\ell=k+1}^{k_*} \sigma_\ell^2. \quad (20)$$

As a result, the decay of singular values can be used to specify the dimension  $k$  of the subspace of the POD basis used in the approximation. In most existing applications, POD can provide a basis that explains the overall dynamics of a given snapshot set and we can use reduced dimension  $k$  that is much less than the original dimension  $n$ , while maintaining high accuracy in the approximation.

**3.2. Discrete Empirical Interpolation Method (DEIM).** In the context of nonlinear model reduction for dynamical systems, the DEIM [22] is generally used for reducing the complexity for evaluating the projected nonlinear term in the POD-Galerkin reduced system. To illustrate this issue, we consider

the projected nonlinear function  $\tilde{\mathbf{E}}(\cdot, \cdot): \mathbb{R}^k \times \mathbb{R} \rightarrow \mathbb{R}^k$  of  $\mathbf{E}(\cdot, \cdot): \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  defined by

$$\tilde{\mathbf{E}}(\tilde{\mathbf{z}}(t), t) := \mathbf{U}_k^T \mathbf{E}(\mathbf{U}_k \tilde{\mathbf{z}}(t), t), \quad \tilde{\mathbf{z}}: \mathbb{R} \rightarrow \mathbb{R}^k. \quad (21)$$

For convenience, the notation will be simplified by leaving the dependence on time parameter  $t$ . Notice that, for a given  $\tilde{\mathbf{z}}$ , the computational complexity for computing the above nonlinear term depends on the dimension  $n$  when multiplying  $\mathbf{U}_k^T$  with  $\mathbf{E}$ . This can be avoided by using the DEIM approximation of the form

$$\tilde{\mathbf{E}}(\tilde{\mathbf{z}}) \approx \mathbf{U}_k^T \bar{\mathbf{U}}_r (\mathbf{P}^T \bar{\mathbf{U}}_r)^{-1} \mathbf{P}^T \mathbf{E}, \quad (22)$$

where the matrix  $\bar{\mathbf{U}}_r$  is the POD basis matrix of rank  $r$  of the nonlinear snapshot matrix  $\mathbf{E}_{\text{mat}} = [\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_{n_s}]$ , for  $\mathbf{E}_j = \mathbf{E}(\mathbf{z}_j, t_j)$  with  $\mathbf{z}_j$  coming from the available snapshot in  $\mathbb{R}^n$  ( $\mathbf{z}_j \approx \mathbf{U}_k \tilde{\mathbf{z}}$  from POD approximation),  $j = 1, \dots, n_s$ , and the matrix  $\mathbf{P}$  is constructed from the interpolation indices  $\varrho_1, \varrho_2, \dots, \varrho_r$ , which are generated by the DEIM algorithm shown in Algorithm 1. In particular,  $\mathbf{P} = [\mathbf{e}_{\varrho_1}, \mathbf{e}_{\varrho_2}, \dots, \mathbf{e}_{\varrho_r}] \in \mathbb{R}^{n \times r}$ , where  $\mathbf{e}_{\varrho_i} = [0, \dots, 0, 1, 0, \dots, 0]^T$  is the  $\varrho_i$  column of the identity matrix  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  for all  $i = 1, 2, \dots, r$  for  $r \leq \text{Rank}(\mathbf{E}_{\text{mat}})$ . Note that the complexity for evaluating (22) can depend only on the reduced-order dimension  $r$  and  $k$  because the term  $\mathbf{U}_k^T \bar{\mathbf{U}}_r (\mathbf{P}^T \bar{\mathbf{U}}_r)^{-1}$  can be precomputed for any given input vector  $\mathbf{z}$  and the term  $\mathbf{P}^T \mathbf{E}$  can be considered as selecting  $r$  components corresponding to the indices  $\varrho_1, \varrho_2, \dots, \varrho_r$ .

From Algorithm 1, the procedure initializes the first interpolation index  $\varrho_1 \in \{1, 2, \dots, n\}$  by using the first input basis  $\bar{\mathbf{u}}_1$  entry which has the largest magnitude. The rest of the indices  $\varrho_j$  for  $j = 2, 3, \dots, r$  are selected from the residual  $\bar{\mathbf{r}} = \bar{\mathbf{v}}_j - \mathbf{U}\mathbf{c}$  with the largest magnitude. More details on this procedure as well as the proof for invertibility of  $\mathbf{P}^T \mathbf{U}_r$  can be found in [22].

**3.3. The POD-DEIM Reduced System.** Consider an initial value problem of the form

$$\mathbf{M} \frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{d} + \mathbf{E}(\mathbf{z}), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \quad (23)$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{d} \in \mathbb{R}^n$  are constant matrices;  $\mathbf{E}(\mathbf{z})$  is the nonlinear function; and  $\mathbf{z} = \mathbf{z}(t) \in \mathbb{R}^n$  is the state variable. The POD reduced system can be obtained by applying the Galerkin projection with the POD basis  $\mathbf{U}_k$  of dimension  $k$  (see Section 3.1), and this system can be written in the form

$$\tilde{\mathbf{M}} \frac{d\tilde{\mathbf{z}}}{dt} = \tilde{\mathbf{A}}\tilde{\mathbf{z}} + \tilde{\mathbf{d}} + \tilde{\mathbf{E}}(\tilde{\mathbf{z}}), \quad \tilde{\mathbf{z}}(t_0) = \mathbf{U}_k^T \mathbf{z}_0, \quad (24)$$

where  $\tilde{\mathbf{M}} = \mathbf{U}_k^T \mathbf{M} \mathbf{U}_k \in \mathbb{R}^{k \times k}$ ,  $\tilde{\mathbf{A}} = \mathbf{U}_k^T \mathbf{A} \mathbf{U}_k \in \mathbb{R}^{k \times k}$ , and  $\tilde{\mathbf{d}} = \mathbf{U}_k^T \mathbf{d} \in \mathbb{R}^k$  are constant matrices;  $\tilde{\mathbf{E}}(\tilde{\mathbf{z}}) = \mathbf{U}_k^T \mathbf{E}(\mathbf{U}_k \tilde{\mathbf{z}})$  is the projected nonlinear function; and  $\tilde{\mathbf{z}} = \tilde{\mathbf{z}}(t) \in \mathbb{R}^k$  is the state

**Require:** Inputs (i) dimension  $r < \text{Rank}(\mathbf{E}_{mat})$ , (ii) basis set  $\{\bar{\mathbf{u}}\}_{j=1}^r = \{\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \dots, \bar{\mathbf{u}}_r\} \subset \mathbb{R}^n$ .

**Ensure:** Outputs  $\mathbf{P} \in \mathbb{R}^{n \times r}$ ,  $\vec{\rho}_r = [\rho_1, \rho_2, \dots, \rho_r]^T \in \mathbb{R}^r$

- 1:  $[\rho_1, \rho_1] = \max \{|\bar{\mathbf{u}}_1|\}$
- 2:  $\bar{\mathbf{U}} = [\bar{\mathbf{u}}_1]$ ,  $\bar{\mathbf{P}} = [\mathbf{e}_{\rho_1}]$ ,  $\vec{\rho} = [\rho_1]$
- 3: **for**  $j = 2 : r$  **do**
- 4:      $\mathbf{c} = (\bar{\mathbf{P}}^T \bar{\mathbf{U}})^{-1} \bar{\mathbf{P}}^T \bar{\mathbf{u}}_j$
- 5:      $\bar{\mathbf{r}} = \bar{\mathbf{u}}_j - \bar{\mathbf{U}} \mathbf{c}$
- 6:      $[\rho_j, \rho_j] = \max \{|\bar{\mathbf{r}}|\}$
- 7:      $\bar{\mathbf{U}} \leftarrow [\bar{\mathbf{U}} \bar{\mathbf{u}}_j]$ ,  $\bar{\mathbf{P}} \leftarrow [\bar{\mathbf{P}} \mathbf{e}_{\rho_j}]$ ,  $\vec{\rho} \leftarrow \begin{bmatrix} \vec{\rho} \\ \rho_j \end{bmatrix}$
- 8: **end for**
- 9:  $\mathbf{P} = \bar{\mathbf{P}}(:, 1 : r)$ ,  $\vec{\rho}_r = \vec{\rho}(1 : r)$

ALGORITHM 1: DEIM.

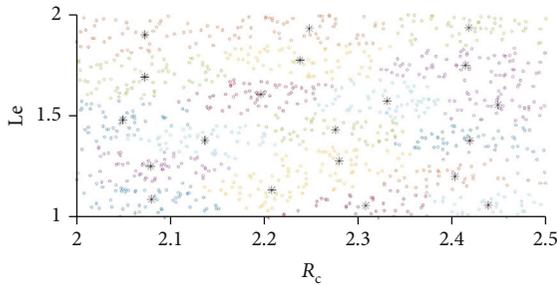


FIGURE 1: Plot of 1000 random samples for parameters  $(R_c, Le) \in [2, 2.5] \times [1, 2] \subset \mathbb{R}^2$  is clustered into  $K = 20$  centroids using the  $K$ -means clustering algorithm. The “\*” denotes the centroid of each cluster.

variable that can be used to approximate the original state variable  $\mathbf{z}$  by  $\mathbf{z} \approx \mathbf{U}_k \tilde{\mathbf{z}}$ . Finally, after approximating the nonlinear term as shown in Section 3.2, we obtain the POD-DEIM reduced system in the following form:

$$\tilde{\mathbf{M}} \frac{d\tilde{\mathbf{z}}}{dt} = \tilde{\mathbf{A}} \tilde{\mathbf{z}} + \tilde{\mathbf{d}} + \mathbf{Q} \tilde{\mathbf{E}}_p(\tilde{\mathbf{z}}), \quad \tilde{\mathbf{z}}(t_0) = \mathbf{U}_k^T \mathbf{z}_0, \quad (25)$$

where  $\mathbf{Q} = \mathbf{U}_k^T \bar{\mathbf{U}}_r (\mathbf{P}^T \bar{\mathbf{U}}_r)^{-1}$  and  $\tilde{\mathbf{E}}_p(\tilde{\mathbf{z}}) = \mathbf{P}^T \mathbf{E}(\mathbf{U}_k \tilde{\mathbf{z}})$  as discussed in (22). The next section extends the previous setup for reduced-order modeling for the parametrized miscible flow model.

#### 4. Model Reduction for the Parametrized Miscible Flow Model by Using the POD-DEIM Approach with $K$ -Means Clustering

This section applies  $K$ -means clustering with the POD-DEIM reduced-order modeling described in Section 3 on the parametrized miscible flow model given in Section 2. In this work, the numbers of spatial grid points are  $n_x$  and  $n_y = 100$ , i.e., the dimension of the full-order model is  $n = n_x n_y$ . The computation is terminated at the time  $t_f$  with time stepsize  $\Delta t$ . Moreover, we are interested in an exothermic reaction, which implies that  $\text{sgn}(\phi) = 1$ . The parameters

$\alpha$ ,  $d$ ,  $Da$ , and  $Pe$  are fixed to be 2, 0.1, 0.01, and 250, respectively. Furthermore, the random noise between 0 and 1 is added to the initial condition for concentration at the collocation points ( $\hat{x} = \alpha Pe/2$  for all  $y \in [0, Pe]$ ) to generate the instability which employs the viscous fingering.

**4.1.  $K$ -Means Clustering.** Let  $I_{R_c} \subset \mathbb{R}$  and  $I_{Le} \subset \mathbb{R}$  be the closed intervals for the values of  $R_c$  and  $Le$ , respectively. Two parameters  $(R_c, Le)$  are generated randomly in  $I_{R_c} \times I_{Le} \subset \mathbb{R}^2$  for 1000 samples and clustered into  $K$  centroids using the  $K$ -means clustering algorithm as shown in Figure 1.

The  $K$ -means clustering algorithm for generating centroids is shown in Algorithm 2 as follows. From Algorithm 2, the requirements are number of desire cluster  $K$ , number of iteration iter, and the data set which is the set  $RcLe$  that contains 1000 training samples of parameters  $(R_c, Le) \in [2, 2.5] \times [1, 2] \subset \mathbb{R}^2$  in this work. The procedure starts with first randomly picking  $K$  initial centroids,  $p_1, p_2, \dots, p_K$ . Then, every point in training set measured the distance between each centroid. The points that have a minimum distance to centroid  $p_k$  are gathered together in the  $k^{\text{th}}$  cluster. The new centroids will be updated by averaging the whole samples in each cluster. This procedure continues and terminates at the  $l = \text{iter}$ . As discussed previously, we use  $K = 20$  in this work.

**4.2. The POD-DEIM Parametrized Reduced Systems.** The snapshots for constructing the POD basis are obtained from the full-order models corresponding to all selected centroids from the  $K$ -means clustering algorithm. Let  $\mathbf{z}_s^\ell = [\psi_s^\ell, \omega_s^\ell, c_s^\ell, T_s^\ell]$  be the numerical solution of  $\mathbf{z}(\mathbf{p}_\ell; t_s)$  at time  $t_s$  from the discretized system of (26) with  $\mathbf{p} = \mathbf{p}_\ell$ , which is the  $\ell^{\text{th}}$  parameter centroid from  $K$ -means clustering, i.e., we obtain the snapshots from solving the following 20 full-order systems:

$$\mathbf{M}(\mathbf{p}_\ell) \frac{d\mathbf{z}}{dt} = \mathbf{A}(\mathbf{p}_\ell) \mathbf{z} + \mathbf{d}(\mathbf{p}_\ell) + \mathbf{E}(\mathbf{p}_\ell; \mathbf{z}, t), \quad \ell = 1, \dots, 20. \quad (26)$$

The snapshots  $\psi_s^\ell, \omega_s^\ell, c_s^\ell$ , and  $T_s^\ell$  are the numerical solutions of (26) that approximate  $\Psi(\mathbf{p}_\ell; t_s)$ ,  $\omega(\mathbf{p}_\ell; t_s)$ ,  $\mathbf{c}(\mathbf{p}_\ell; t_s)$ , and  $\mathbf{T}(\mathbf{p}_\ell; t_s)$ , respectively, for  $s = 1, \dots, n_s$  and  $\ell = 1, \dots, K$ .

**Require:** Inputs (i)  $K \in \mathbb{N}$ , (ii)  $iter \in \mathbb{N}$ , (iii)  $RcLe = \{[R_c^{(i)}, Le^{(i)}]^T : R_c^{(i)} \in I_{R_c} \subset \mathbb{R}^n \text{ and } Le^{(i)} \in I_{Le} \subset \mathbb{R}^n\}_{i=1}^{1000} \subset \mathbb{R}^2$

**Ensure:** Outputs  $p_1, p_2, \dots, p_K$

- 1: Choose randomly  $K$  initial centroids,  $p_1, p_2, \dots, p_K$  from RcLe
- 2:  $D = \text{zeros}(1000, K + 1)$
- 3: **for**  $l = 1 : iter$  **do**
- 4:   **for**  $i = 1 : 1000$  **do**
- 5:     **for**  $j = 1 : K$  **do**
- 6:        $D(i, j) = \|RcLe(:, i) - p_j\|_2$
- 7:     **end for**
- 8:      $[m, \text{label}] = \min(D(i, 1 : K))$
- 9:      $D(i, K + 1) = \text{label}$
- 10:   **end for**
- 11:   **for**  $k = 1 : K$  **do**
- 12:      $p_k \leftarrow \text{mean}(RcLe(:, D(:, K + 1) == k))$
- 13:   **end for**
- 14: **end for**

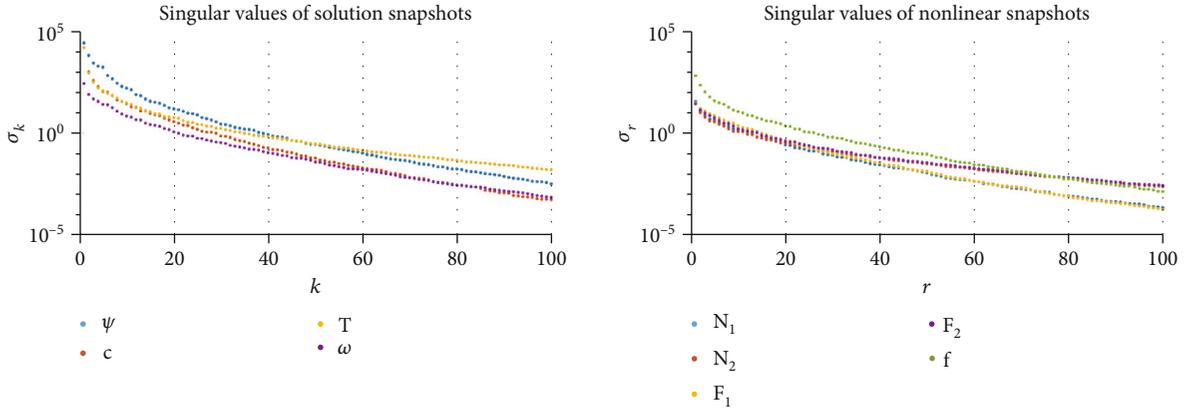
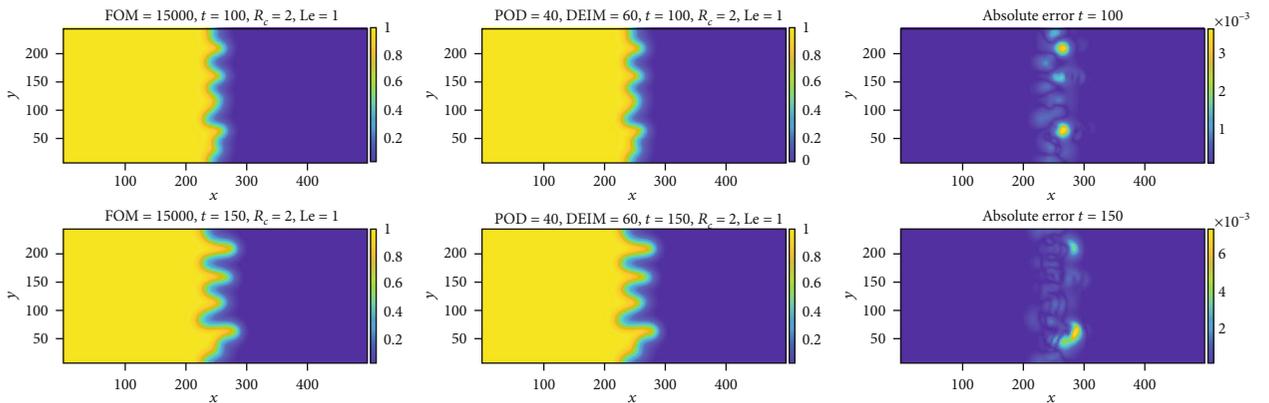
ALGORITHM 2:  $K$ -means clustering algorithm.

FIGURE 2: Plots of the first 100 singular values for solution snapshots and nonlinear snapshots.

FIGURE 3: Comparison of approximate concentration for fixed parameters  $(R_c, Le) = (2, 1)$  compared to the full-order model. The results are illustrated at  $t = 100$  or  $s = 1000$  (top row) and  $t = 150$  or  $s = 1500$  (bottom row).

Let  $\mathbf{C} = [c_1^1, c_2^1, \dots, c_{n_s}^1 \mid c_1^2, c_2^2, \dots, c_{n_s}^2 \mid \dots \mid c_1^K, c_2^K, \dots, c_{n_s}^K] \in \mathbb{R}^{n \times K n_s}$  be the snapshot matrix for concentration. In our numerical experiment,  $K = 20$  and  $n_s = 1500$ . To obtain the POD basis, the singular value decomposition (SVD) of  $\mathbf{C}$  is computed, i.e.,  $\mathbf{C} = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$ . Then, the POD basis of dimen-

sion  $k \ll \text{rank}(\mathbf{C})$  is the first  $k$  columns of the left singular matrix  $\mathbf{U}_c$  of  $\mathbf{C}$  denoted by  $\mathbf{U}_c^k \in \mathbb{R}^{n \times k}$ . Similarly, let  $\mathbf{U}_T^k$ ,  $\mathbf{U}_\psi^k$ , and  $\mathbf{U}_\omega^k$  be the POD basis of dimension  $k$  for temperature, stream function, and vorticity, respectively. The POD reduced system is constructed by applying the Galerkin

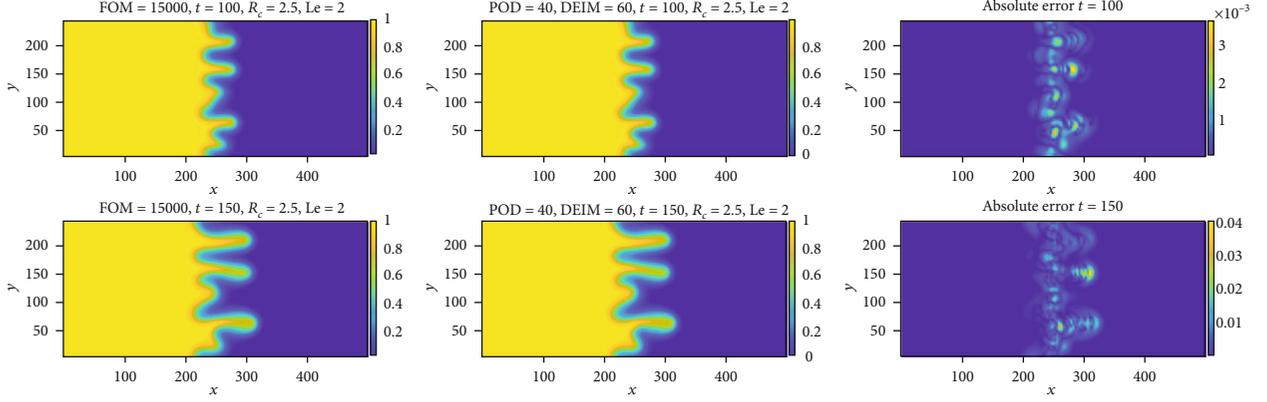


FIGURE 4: Comparison of approximate concentration for fixed parameters  $(R_c, Le) = (2.5, 2)$  compared to the full-order model. The results are illustrated at  $t = 100$  or  $s = 1000$  (top row) and  $t = 150$  or  $s = 1500$  (bottom row).

TABLE 1: Error at specific time and global error.

$(R_c, Le) = (2, 1)$	$k$	$r$	$E_{100}$	$E_{150}$	$E$	$\hat{E}$
FOM (15,000)	—	—	—	—	—	—
POD	40	—	0.0223	0.0489	0.8283	$5.5220 \times 10^{-4}$
POD	60	—	0.0021	0.0051	0.0888	$5.9200 \times 10^{-5}$
POD-DEIM	40	40	0.0973	0.1650	3.4607	$2.3071 \times 10^{-3}$
POD-DEIM	40	60	0.0334	0.0651	1.2382	$8.2547 \times 10^{-4}$

TABLE 2: Error at specific time and global error.

$(R_c, Le) = (2.5, 2)$	$k$	$r$	$E_{100}$	$E_{150}$	$E$	$\hat{E}$
FOM (15,000)	—	—	—	—	—	—
POD	40	—	0.0795	0.3174	2.8859	$1.9239 \times 10^{-3}$
POD	60	—	0.0083	0.1092	0.5732	$3.8213 \times 10^{-4}$
POD-DEIM	40	40	0.2491	0.5100	9.5423	$6.3621 \times 10^{-3}$
POD-DEIM	40	60	0.0395	0.3554	2.3611	$1.5741 \times 10^{-3}$

TABLE 3: Average CPU time usage and average global error.

	$k$	$r$	Avg. CPU time	Avg. $E$	Avg. $\hat{E}$
FOM (15,000)	—	—	112.8980	—	—
POD	40	—	49.2821	0.7933	$5.2887 \times 10^{-4}$
POD	60	—	75.5342	0.1349	$8.9933 \times 10^{-4}$
POD-DEIM	40	40	1.0009	2.7002	$1.8001 \times 10^{-4}$
POD-DEIM	40	60	1.3127	0.9024	$6.0160 \times 10^{-4}$

projection on (26) with the basis  $\mathbf{U}^k = \text{diag}(\mathbf{U}_\psi^k, \mathbf{U}_\omega^k, \mathbf{U}_c^k, \mathbf{U}_T^k) \in \mathbb{R}^{4n \times 4k}$ . The state variable is approximated by  $\mathbf{z}(t) \approx \mathbf{U}^k \tilde{\mathbf{z}}(t)$ , where  $\tilde{\mathbf{z}}(t) = [\tilde{\psi}(t), \tilde{\omega}(t), \tilde{c}(t), \tilde{T}(t)] \in \mathbb{R}^{4k}$  is the state variable of the POD reduced-order system, i.e.,  $\mathbf{c}(t) \approx \mathbf{U}_c^k \tilde{c}(t)$ ,  $\mathbf{T}(t) \approx \mathbf{U}_T^k \tilde{T}(t)$ ,  $\boldsymbol{\psi}(t) \approx \mathbf{U}_\psi^k \tilde{\psi}(t)$ , and  $\boldsymbol{\omega}(t) \approx \mathbf{U}_\omega^k \tilde{\omega}(t)$ . The resulting POD reduced system is in the form of

$$\tilde{\mathbf{M}}(\mathbf{p}) \frac{d\tilde{\mathbf{z}}}{dt} = \tilde{\mathbf{A}}(\mathbf{p})\tilde{\mathbf{z}} + \tilde{\mathbf{d}}(\mathbf{p}) + \tilde{\mathbf{E}}(\mathbf{p}; \tilde{\mathbf{z}}, t), \quad (27)$$

where  $\tilde{\mathbf{M}}(\mathbf{p}) = \mathbf{U}^{kT} \mathbf{M}(\mathbf{p}) \mathbf{U}^k$ ,  $\tilde{\mathbf{A}}(\mathbf{p}) = \mathbf{U}^{kT} \mathbf{A}(\mathbf{p}) \mathbf{U}^k \in \mathbb{R}^{4k \times 4k}$ , and  $\tilde{\mathbf{d}}(\mathbf{p}) = \mathbf{U}^{kT} \mathbf{d}(\mathbf{p}) \in \mathbb{R}^{4k}$  are constant matrices for a given parameter vector  $\mathbf{p}$ , and  $\tilde{\mathbf{E}}(\mathbf{p}; \tilde{\mathbf{z}}, t) = \mathbf{U}^{kT} \mathbf{E}(\mathbf{p}; \mathbf{U}^k \tilde{\mathbf{z}}, t)$  is the projected nonlinear term. As described in Section 3, we can obtain a POD-DEIM reduced system from the POD reduced system (27) as shown below:

$$\tilde{\mathbf{M}}(\mathbf{p}) \frac{d\tilde{\mathbf{z}}}{dt} = \tilde{\mathbf{A}}(\mathbf{p})\tilde{\mathbf{z}} + \tilde{\mathbf{d}}(\mathbf{p}) + \hat{\mathbf{E}}(\mathbf{p}; \tilde{\mathbf{z}}, t), \quad (28)$$

where  $\hat{\mathbf{E}}(\mathbf{p}; \tilde{\mathbf{z}}, t)$  is the nonlinear term obtained from using the DEIM approximation for each nonlinear term in  $\tilde{\mathbf{E}}$  in (27). Note that  $\hat{\mathbf{E}}(\mathbf{p}; \tilde{\mathbf{z}}, t)$  is not directly constructed as explained in Section 3, since it consists of many nonlinear functions  $\mathbf{N}(\boldsymbol{\psi}(t), \mathbf{c}(t)), \mathbf{N}(\boldsymbol{\psi}(t), \mathbf{T}(t)), \mathbf{F}(\boldsymbol{\psi}(t), \mathbf{c}(t)), \mathbf{F}(\boldsymbol{\psi}(t),$

$\mathbf{T}(t)$ ), and  $\mathbf{f}(\mathbf{c}(t))$  given in (13) that depends on different state variables. Therefore, we separately apply this approximation to each nonlinear term. For example, let us first construct the nonlinear snapshot matrix

$$\mathbf{N}_c = \left[ \mathbf{N}_1^1, \mathbf{N}_2^1, \dots, \mathbf{N}_{n_s}^1 \mid \mathbf{N}_1^2, \mathbf{N}_2^2, \dots, \mathbf{N}_{n_s}^2 \mid \dots \mid \mathbf{N}_1^K, \mathbf{N}_2^K, \dots, \mathbf{N}_{n_s}^K \right] \in \mathbb{R}^{n \times Kn_s}, \quad (29)$$

for  $\mathbf{N}(\boldsymbol{\psi}(t), \mathbf{c}(t))$ , where  $\mathbf{N}_s^\ell = \mathbf{N}(\boldsymbol{\psi}_s^\ell, \mathbf{c}_s^\ell)$ . The DEIM is then applied on the nonlinear term  $\mathbf{N}(\boldsymbol{\psi}(t), \mathbf{c}(t))$ . Other nonlinear terms are done in the same manner. Details on the construction of each nonlinear term by the DEIM can be found in Appendix A which follows from [20]. Finally, the construction of  $\hat{\mathbf{E}}$  is just the combination of the DEIM results in each nonlinear term. The numerical simulations based on the techniques described in this section are illustrated and analyzed in Section 5.

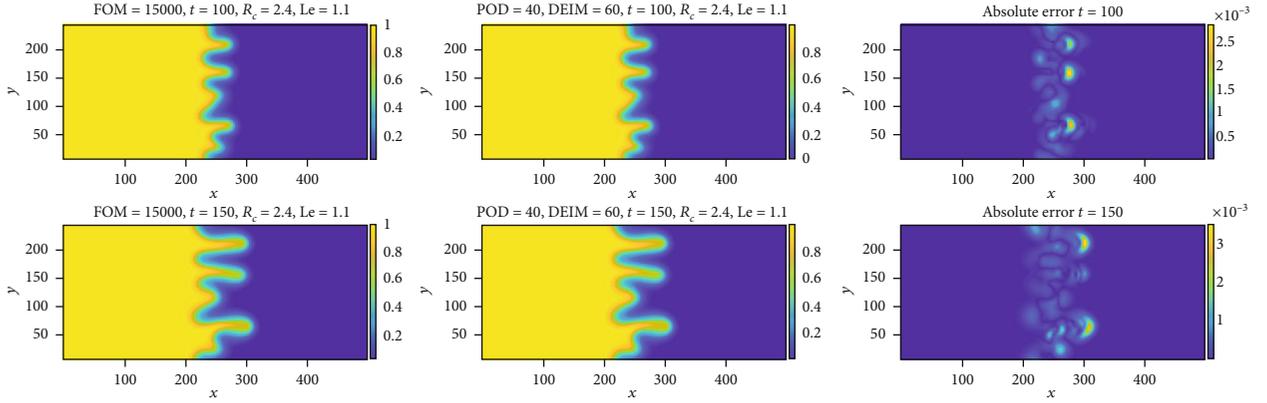


FIGURE 5: Comparison of approximate concentration for fixed parameters  $(R_c, Le) = (2.4, 1.1)$  compared to the full-order model. The results are illustrated at  $t = 100$  or  $s = 1000$  (top row) and  $t = 150$  or  $s = 1500$  (bottom row).

TABLE 4: Error at specific time and global error.

$(R_c, Le) = (2.4, 1.1)$	$k$	$r$	$E_{100}$	$E_{150}$	$E$	$\hat{E}$
FOM (15,000)	—	—	—	—	—	—
POD	40	—	0.0329	0.0517	1.0691	$7.1273 \times 10^{-4}$
POD	60	—	0.0046	0.0078	0.1573	$1.0487 \times 10^{-4}$
POD-DEIM	40	40	0.0852	0.1530	2.9759	$1.9839 \times 10^{-3}$
POD-DEIM	40	60	0.0240	0.0326	0.8189	$5.4593 \times 10^{-4}$

## 5. Numerical Experiments

In this section, there are two main numerical tests that consider accuracy and efficiency in terms of reducing the simulation time of the model reduction framework presented in this paper. In Section 5.1, the numerical solutions of the proposed method and the original full-order system are compared for different parameter values that are not in the training set. In Section 5.2, the overall computational time reduction and average errors are considered for the solutions from the proposed method when using various randomly selected parameter values in the interested domain. The relevant MATLAB code contributed in this work is available at <https://bit.ly/3airQ2u>.

In this work, the number of spatial grid points is  $n_x = 150$  and  $n_y = 100$ , i.e., the dimension of the full-order model is  $n = n_x n_y = 15000$ . The computation is terminated at the time  $t_f = 150$  with  $\Delta t = 0.1$  (1500 time steps). The system temperature is set to be constant (isothermal case), i.e., it is equivalent to set  $R_T = 0$ .

To investigate the possibility of reducing the dimension, we first consider the singular values of both solution snapshots and nonlinear snapshots as depicted in Figure 2. Note that this work illustrates only the approximations of concentration. Other state variables have similar accuracy results since they are all coupled in the differential equations.

**5.1. Numerical Test I.** In this section, the POD and POD-DEIM reduced-order model are compared to the full-order

model for different given parameters  $(R_c, Le) \in [2, 2.5] \times [1, 2] \subset \mathbb{R}^2$ . To demonstrate the efficiency of the reduced-order model, the error at a certain time  $t_s$  and the global error and its average are defined in terms of the Frobenius norm as follows:

$$\begin{aligned}
 E_{t_s} &= \left\| \mathbf{c}(t_s) - \mathbf{U}_c^k \mathbf{c}(t_s) \right\|_F, \\
 E &= \sum_{s=1}^{n_s} \left\| \mathbf{c}(t_s) - \mathbf{U}_c^k \mathbf{c}(t_s) \right\|_F, \\
 \hat{E} &= \frac{E}{n_s},
 \end{aligned} \tag{30}$$

where  $\mathbf{c}(t_s)$  and  $\tilde{\mathbf{c}}(t_s)$  are the concentrations obtained from the full-order and reduced-order models, respectively, at the time step  $s$  or the time  $t_s$ , and  $n_s$  is the number of snapshots. In this experiment,  $n_s = 1500$ . The approximation  $\tilde{\mathbf{c}}(t_s)$  can be the solution from the POD or POD-DEIM reduced-order model. Some cases of parameters  $(R_c, Le) \in [2, 2.5] \times [1, 2]$  are chosen arbitrarily, and the corresponding numerical results are shown below. In particular, we will consider the cases of  $(R_c, Le) = (2, 1)$ ,  $(2.5, 2)$ , and  $(2.4, 1.1)$ , in Sections 5.1.1, 5.1.2, and 5.1.3, respectively.

In Sections 5.1.1 and 5.1.2, the parameters are chosen from different corners of our interested 2-dimensional parameter domain, i.e.,  $(R_c, Le) = (2, 1)$  and  $(R_c, Le) = (2.5, 2)$ . These parameters are not in the set of training parameters. As shown in Figures 3 and 4, the resulting POD-DEIM

approximation with dimension  $\text{POD} = 40/\text{DEIM} = 60$  of concentration for both cases is compared to the full-order model at the time  $t = 100$  ( $s = 1000$ ) and the final time  $t = 150$  ( $s = 1500$ ). Since these solutions are visually indistinguishable, the absolute errors at all grid points are also provided in Figures 3 and 4. These errors are shown to be in the order of  $\mathcal{O}(10^{-3})$  or  $\mathcal{O}(10^{-2})$ . More details on the error at specific time  $t = 100$ ,  $t = 150$ , and average global error are given in Tables 1 and 2, respectively, for Sections 5.1.1 and 5.1.2. As expected, the increasing number of POD basis  $k$  provides more accuracy as the error seems to be decreasing. However, this can increase the simulation time as shown in Table 3, which will be discussed in more detail later in Section 5.2.

Similarly, in Section 5.1.3, all results are demonstrated in the same manner as seen in Figure 5 and Table 4. The difference is the parameters that are chosen inside the parameter domain, i.e.,  $(R_c, \text{Le}) = (2.4, 1.1)$ . Table 4 shows that the error at a specific time and the average global error are smaller compared to the previous cases.

$$5.1.1. (R_c, \text{Le}) = (2, 1)$$

$$5.1.2. (R_c, \text{Le}) = (2.5, 2)$$

$$5.1.3. (R_c, \text{Le}) = (2.4, 1.1)$$

5.2. *Numerical Test II.* In this numerical test, we randomly generate 10 distinct parameters  $(R_c, \text{Le})$  in our interested parameter domain  $[2, 2.5] \times [1, 2]$ . The solutions are obtained from solving the reduced-order model corresponding to those parameters. CPU time and global error are recorded and averaged as shown in Table 3.

From Table 3, applying the POD method can make the simulation faster, but it is only approximately twice or three times speedup due to the inefficiency in computing the nonlinear term. After the DEIM is applied, the average CPU time speedups by roughly the order of  $\mathcal{O}(100)$  at equivalent accuracy. In particular, the POD method with  $k = 40$  uses approximately 1.3 hours (49,2821 seconds), while the POD-DEIM method with  $k = 40$ ,  $r = 60$  uses roughly only 1.3 seconds. In this case, the average error for both POD and POD-DEIM approach is approximately the same. As shown in the previous numerical test, an increasing dimension can improve the accuracy with a trade-off in computational time.

The results in this section suggest that the POD-DEIM reduced-order model with the basis constructed from the  $K$ -means approach is reliable in approximating the solution for any given parameter in the interested parameter domain  $[2, 2.5] \times [1, 2]$ , even though they are not in the training set.

## 6. Conclusions

This work provides a model order reduction technique for simulating the parametrized viscous fingering model in a horizontal flow through a porous media domain by using several techniques. The  $K$ -means clustering algorithm is first used to cluster parameters equipped in the model. The snapshots are collected from solving the solution from each cen-

teroid. Then, the POD is applied in conjunction with the DEIM to construct a reduced-order model. The resulting reduced-order model can be calculated entirely in the reduced dimension without dependence on the original dimension for various parameter values that are not used in the training set obtained from the  $K$ -means algorithm. The approximation admits faster evaluation while trading a small amount of accuracy. The error estimates of the proposed approach for this nonlinear miscible flow model can be obtained by incorporating the accuracy of  $K$ -means clustering with the error analysis from [23] and from [24] for, respectively, a posteriori error and a priori error estimates. These theoretical results are beyond the scope of this work and left to be considered in the future.

## Appendix

### A. Details on Discretized Systems

A.1. *Discretized Full-Order System.* The central finite difference is applied to construct the spatial discretization of (5)–(8), and it becomes a system of nonlinear ODEs as follows:

$$A\boldsymbol{\Psi}(t) = -\boldsymbol{\omega}(t), \quad (\text{A.1})$$

$$\begin{aligned} \boldsymbol{\omega}(t) = & -R_c [\mathbf{N}(\boldsymbol{\Psi}(t), \mathbf{c}(t)) + A_x \mathbf{c}(t)] \\ & + R_T [\mathbf{N}(\boldsymbol{\Psi}(t), \mathbf{T}(t)) + A_y \mathbf{T}(t)], \end{aligned} \quad (\text{A.2})$$

$$\frac{d\mathbf{c}(t)}{dt} = -\mathbf{F}(\boldsymbol{\Psi}(t), \mathbf{c}(t)) + [\mathbf{Ac}(t) + \mathbf{b}] + \text{Daf}(\mathbf{c}(t)), \quad (\text{A.3})$$

$$\begin{aligned} \frac{d\mathbf{T}(t)}{dt} = & -\mathbf{F}(\boldsymbol{\Psi}(t), \mathbf{T}(t)) + \text{Le}[\mathbf{Ac}(t) + \mathbf{b}] \\ & + \text{sgn}(\phi) \text{Daf}(\mathbf{c}(t)), \end{aligned} \quad (\text{A.4})$$

where the nonlinear terms are defined as

$$\begin{aligned} \mathbf{N}(\boldsymbol{\Psi}(t), \mathbf{v}(t)) = & [(A_x \boldsymbol{\Psi}(t)) .* (A_x \mathbf{v}(t) + \mathbf{b}_x)] \\ & + [(A_y \boldsymbol{\Psi}(t)) .* (A_y \mathbf{v}(t))], \\ \mathbf{F}(\boldsymbol{\Psi}(t), \mathbf{v}(t)) = & [(A_y \boldsymbol{\Psi}(t)) .* (A_x \mathbf{v}(t) + \mathbf{b}_x)] \\ & - [(A_x \boldsymbol{\Psi}(t)) .* (A_y \mathbf{v}(t))], \\ \mathbf{f}(\mathbf{v}(t)) = & -\mathbf{v}(t) .* (\mathbf{v}(t) - 1) .* (\mathbf{v}(t) + d), \end{aligned} \quad (\text{A.5})$$

with  $.*$  denotes the componentwise multiplication in MATLAB,  $A, A_x, A_y \in \mathbb{R}^{n \times n}$  are constant coefficient matrices, and  $\mathbf{b}, \mathbf{b}_x \in \mathbb{R}^n$  are vectors indicating the boundary conditions. Then, the forward time integration with the predictor-corrector scheme is applied to (A.1)–(A.4) to obtain the approximation at each time step. However, the dimension of system  $n$  can be very large to obtain an accurate numerical result which demands tremendous memory storage and over-abundant computational time.

*A.2. POD Reduced System.* The POD reduced system is constructed by applying the Galerkin projection on (A.1)–(A.4) by first substituting their approximations  $\mathbf{c}(t) \approx \mathbf{U}_c^k \tilde{c}(t)$ ,  $\mathbf{T}(t) \approx \mathbf{U}_T^k \tilde{T}(t)$ ,  $\boldsymbol{\psi}(t) \approx \mathbf{U}_\psi^k \tilde{\psi}(t)$ , and  $\boldsymbol{\omega}(t) \approx \mathbf{U}_\omega^k \tilde{\omega}(t)$  and then pre-multiplying (A.1) and (A.2) by  $\mathbf{U}_\omega^{kT}$ , (A.3) by  $\mathbf{U}_c^{kT}$ , and (A.4) by  $\mathbf{U}_T^{kT}$ . The POD reduced-order model is of the form

$$A_1 \tilde{\psi}(t) = -\tilde{\omega}(t), \quad (\text{A.6})$$

$$\begin{aligned} \tilde{\omega}(t) = & -R_c \left[ \mathbf{U}_\omega^{kT} \mathbf{N}_1(\tilde{\psi}(t), \tilde{c}(t)) + A_2 \tilde{c}(t) \right] \\ & + R_T \left[ \mathbf{U}_\omega^{kT} \mathbf{N}_2(\tilde{\psi}(t), \tilde{T}(t)) + A_3 \tilde{T}(t) \right], \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \frac{d\tilde{c}(t)}{dt} = & -\mathbf{U}_c^{kT} \mathbf{F}_1(\tilde{\psi}(t), \tilde{c}(t)) + \left[ A_4 \tilde{c}(t) + \tilde{b}_1 \right] \\ & + \text{Da} \mathbf{U}_c^{kT} \mathbf{f}(\mathbf{U}_c^k \tilde{c}(t)), \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned} \frac{d\tilde{T}(t)}{dt} = & -\mathbf{U}_T^{kT} \mathbf{F}_2(\tilde{\psi}(t), \tilde{T}(t)) + \text{Le} \left[ A_5 \tilde{T}(t) + \tilde{b}_2 \right] \\ & + \text{sgn}(\phi) \text{Da} \mathbf{U}_T^{kT} \mathbf{f}(\mathbf{U}_c^k \tilde{c}(t)), \end{aligned} \quad (\text{A.9})$$

where  $A_1 = \mathbf{U}_\omega^{kT} A \mathbf{U}_\psi^k$ ,  $A_2 = \mathbf{U}_\omega^{kT} A_y \mathbf{U}_c^k$ ,  $A_3 = \mathbf{U}_\omega^{kT} A_y \mathbf{U}_T^k$ ,  $A_4 = \mathbf{U}_c^{kT} A \mathbf{U}_c^k$ ,  $A_5 = \mathbf{U}_T^{kT} A \mathbf{U}_T^k$ ,  $\tilde{b}_1 = \mathbf{U}_c^{kT} \mathbf{b}$ ,  $\tilde{b}_2 = \mathbf{U}_T^{kT} \mathbf{b}$ , and the nonlinear becomes

$$\begin{aligned} \mathbf{N}_1(\tilde{\psi}(t), \tilde{c}(t)) = & \left[ \left( A_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_x \mathbf{U}_c^k \tilde{c}(t) + \mathbf{b}_x \right) \right] \\ & + \left[ \left( A_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_y \mathbf{U}_c^k \tilde{c}(t) \right) \right], \end{aligned} \quad (\text{A.10})$$

$$\begin{aligned} \mathbf{N}_2(\tilde{\psi}(t), \tilde{T}(t)) = & \left[ \left( A_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_x \mathbf{U}_T^k \tilde{T}(t) + \mathbf{b}_x \right) \right] \\ & + \left[ \left( A_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_y \mathbf{U}_T^k \tilde{T}(t) \right) \right], \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} \mathbf{F}_1(\tilde{\psi}(t), \tilde{c}(t)) = & \left[ \left( A_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_x \mathbf{U}_c^k \tilde{c}(t) + \mathbf{b}_x \right) \right] \\ & - \left[ \left( A_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_y \mathbf{U}_c^k \tilde{c}(t) \right) \right], \end{aligned} \quad (\text{A.12})$$

$$\begin{aligned} \mathbf{F}_2(\tilde{\psi}(t), \tilde{T}(t)) = & \left[ \left( A_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_x \mathbf{U}_T^k \tilde{T}(t) + \mathbf{b}_x \right) \right] \\ & - \left[ \left( A_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) .* \left( A_y \mathbf{U}_T^k \tilde{T}(t) \right) \right], \end{aligned} \quad (\text{A.13})$$

$$\mathbf{f}(\mathbf{U}_c^k \tilde{c}(t)) = -\mathbf{U}_c^k \tilde{c}(t) .* \left( \mathbf{U}_c^k \tilde{c}(t) - 1 \right) .* \left( \mathbf{U}_c^k \tilde{c}(t) + d \right). \quad (\text{A.14})$$

The coefficient matrices and boundary vectors are reduced to be a very smaller size  $\mathbb{R}^{k \times k}$  and  $\mathbb{R}^k$ . Nevertheless, the projected nonlinear terms which occur in (A.6)–(A.9)

$$\mathbf{U}_\omega^{kT} \mathbf{N}_1(\tilde{\psi}(t), \tilde{c}(t)) \mathbf{U}_\omega^{kT} \mathbf{N}_2(\tilde{\psi}(t), \tilde{T}(t)), \quad (\text{A.15})$$

$$\mathbf{U}_c^{kT} \mathbf{F}_1(\tilde{\psi}(t), \tilde{c}(t)) \mathbf{U}_T^{kT} \mathbf{F}_2(\tilde{\psi}(t), \tilde{T}(t)), \quad (\text{A.16})$$

$$\mathbf{U}_c^{kT} \mathbf{f}(\mathbf{U}_c^k \tilde{c}(t)) \mathbf{U}_T^{kT} \mathbf{f}(\mathbf{U}_c^k \tilde{c}(t)), \quad (\text{A.17})$$

still have computational complexities depending on the original dimension  $n$ . In the next section, we introduce the prevalent technique to overcome this problem.

*A.3. Discrete Empirical Interpolation Method (DEIM).* The DEIM is an efficient way to reduce the complexities for evaluating nonlinear terms (A.10)–(A.14) as previously discussed. The DEIM provides the evaluation in a nonlinear term at the selected interpolation indices. Suppose that we already have nonlinear snapshots matrices  $\mathbf{N}_c$ ,  $\mathbf{N}_T$ ,  $\mathbf{F}_c$ ,  $\mathbf{F}_T$ , and  $\mathbf{f}_c$  for  $\mathbf{N}_1$ ,  $\mathbf{N}_2$ ,  $\mathbf{F}_1$ ,  $\mathbf{F}_2$ , and  $\mathbf{f}$ , respectively. Note that these snapshots are by-product obtaining from the full-order model calculation. The SVD is computed for each nonlinear snapshot matrix, and the first  $r$  columns of left singular matrices are chosen. For convenience, we will show only  $\mathbf{N}_1$  case; the other cases will be done in the same manner. Let  $\mathbf{U}_{N_1}^r \in \mathbb{R}^{n \times r}$  be the POD basis of dimension  $r$  for  $\mathbf{N}_1$  which is obtained from the SVD of  $\mathbf{N}_c$ . This follows that

$$\mathbf{N}_1 \approx \mathbf{U}_{N_1}^r \tilde{\mathbf{N}}_1. \quad (\text{A.18})$$

Consider a matrix

$$\mathbf{P}_{N_1} = \left[ \mathbf{e}_{\varrho_1}, \mathbf{e}_{\varrho_2}, \dots, \mathbf{e}_{\varrho_r} \right] \in \mathbb{R}^{n \times r}, \quad (\text{A.19})$$

where  $\mathbf{e}_{\varrho_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$  is the  $\varrho_i$  column of the identity matrix  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ . By premultiplying  $\mathbf{P}_{N_1}^T$  both sides of (A.18), the selection of components in the nonlinear terms  $\mathbf{N}_1$  is made as follows:

$$\mathbf{P}_{N_1}^T \mathbf{N}_1 \approx \mathbf{P}_{N_1}^T \mathbf{U}_{N_1}^r \tilde{\mathbf{N}}_1. \quad (\text{A.20})$$

Assume that  $\mathbf{P}_{N_1}^T \mathbf{N}_1$  is nonsingular. Then,  $\tilde{\mathbf{N}}_1 \approx (\mathbf{P}_{N_1}^T \mathbf{U}_{N_1}^r)^{-1} \mathbf{P}_{N_1}^T \mathbf{N}_1$ . Hence, the final approximation form of (A.18) becomes

$$\mathbf{N}_1 \approx \mathbf{U}_{N_1}^r \left( \mathbf{P}_{N_1}^T \mathbf{U}_{N_1}^r \right)^{-1} \mathbf{P}_{N_1}^T \mathbf{N}_1. \quad (\text{A.21})$$

If the DEIM is applied for all cases of nonlinear terms, (A.15)–(A.17) can be expressed as

$$\begin{aligned}
\mathbf{U}_\omega^k \mathbf{N}_1 &\approx \underbrace{\mathbf{U}_\omega^k \mathbf{U}_{N_1}^r \left( \mathbf{P}_{N_1}^T \mathbf{U}_{N_1}^r \right)^{-1}}_{\mathbf{E}_1} \underbrace{\mathbf{P}_{N_1}^T \mathbf{N}_1}_{\tilde{\mathbf{N}}_1}, \\
\mathbf{U}_\omega^k \mathbf{N}_2 &\approx \underbrace{\mathbf{U}_\omega^k \mathbf{U}_{N_2}^r \left( \mathbf{P}_{N_2}^T \mathbf{U}_{N_2}^r \right)^{-1}}_{\mathbf{E}_2} \underbrace{\mathbf{P}_{N_2}^T \mathbf{N}_2}_{\tilde{\mathbf{N}}_2}, \\
\mathbf{U}_c^k \mathbf{F}_1 &\approx \underbrace{\mathbf{U}_c^k \mathbf{U}_{F_1}^r \left( \mathbf{P}_{F_1}^T \mathbf{U}_{F_1}^r \right)^{-1}}_{\mathbf{E}_3} \underbrace{\mathbf{P}_{F_1}^T \mathbf{F}_1}_{\tilde{\mathbf{F}}_1}, \\
\mathbf{U}_T^k \mathbf{F}_2 &\approx \underbrace{\mathbf{U}_T^k \mathbf{U}_{F_2}^r \left( \mathbf{P}_{F_2}^T \mathbf{U}_{F_2}^r \right)^{-1}}_{\mathbf{E}_4} \underbrace{\mathbf{P}_{F_2}^T \mathbf{F}_2}_{\tilde{\mathbf{F}}_2}, \\
\mathbf{U}_c^k \mathbf{f} &\approx \underbrace{\mathbf{U}_c^k \mathbf{U}_f^r \left( \mathbf{P}_f^T \mathbf{U}_f^r \right)^{-1}}_{\mathbf{E}_5} \underbrace{\mathbf{P}_f^T \mathbf{f}}_{\tilde{\mathbf{f}}}, \\
\mathbf{U}_T^k \mathbf{f} &\approx \underbrace{\mathbf{U}_T^k \mathbf{U}_f^r \left( \mathbf{P}_f^T \mathbf{U}_f^r \right)^{-1}}_{\mathbf{E}_6} \underbrace{\mathbf{P}_f^T \mathbf{f}}_{\tilde{\mathbf{f}}},
\end{aligned} \tag{A.22}$$

where the nonlinear terms are reduced to be

$$\begin{aligned}
\tilde{\mathbf{N}}_1 &= \left[ \left( \mathbf{P}_{N_1}^T \mathbf{A}_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{N_1}^T \mathbf{A}_x \mathbf{U}_c^k \tilde{c}(t) + \mathbf{P}_{N_1}^T \mathbf{b}_x \right) \right] \\
&+ \left[ \left( \mathbf{P}_{N_1}^T \mathbf{A}_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{N_1}^T \mathbf{A}_y \mathbf{U}_c^k \tilde{c}(t) \right) \right],
\end{aligned} \tag{A.23}$$

$$\begin{aligned}
\tilde{\mathbf{N}}_2 &= \left[ \left( \mathbf{P}_{N_2}^T \mathbf{A}_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{N_2}^T \mathbf{A}_x \mathbf{U}_T^k \tilde{T}(t) + \mathbf{P}_{N_2}^T \mathbf{b}_x \right) \right] \\
&+ \left[ \left( \mathbf{P}_{N_2}^T \mathbf{A}_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{N_2}^T \mathbf{A}_y \mathbf{U}_T^k \tilde{T}(t) \right) \right],
\end{aligned} \tag{A.24}$$

$$\begin{aligned}
\tilde{\mathbf{F}}_1 &= \left[ \left( \mathbf{P}_{F_1}^T \mathbf{A}_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{F_1}^T \mathbf{A}_x \mathbf{U}_c^k \tilde{c}(t) + \mathbf{P}_{F_1}^T \mathbf{b}_x \right) \right] \\
&- \left[ \left( \mathbf{P}_{F_1}^T \mathbf{A}_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{F_1}^T \mathbf{A}_y \mathbf{U}_c^k \tilde{c}(t) \right) \right],
\end{aligned} \tag{A.25}$$

$$\begin{aligned}
\tilde{\mathbf{F}}_2 &= \left[ \left( \mathbf{P}_{F_2}^T \mathbf{A}_y \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{F_2}^T \mathbf{A}_x \mathbf{U}_T^k \tilde{T}(t) + \mathbf{P}_{F_2}^T \mathbf{b}_x \right) \right] \\
&- \left[ \left( \mathbf{P}_{F_2}^T \mathbf{A}_x \mathbf{U}_\psi^k \tilde{\psi}(t) \right) * \left( \mathbf{P}_{F_2}^T \mathbf{A}_y \mathbf{U}_T^k \tilde{T}(t) \right) \right],
\end{aligned} \tag{A.26}$$

$$\tilde{\mathbf{f}} = -\mathbf{P}_f^T \mathbf{U}_c^k \tilde{c}(t) * \left( \mathbf{P}_f^T \mathbf{U}_c^k \tilde{c}(t) - 1 \right) * \left( \mathbf{P}_f^T \mathbf{U}_c^k \tilde{c}(t) + d \right). \tag{A.27}$$

By substituting (A.23)–(A.27) into the POD reduced-order model, the POD-DEIM reduced-order model takes the form

$$\mathbf{A}_1 \tilde{\psi}(t) = -\tilde{\omega}(t), \tag{A.28}$$

$$\begin{aligned}
\tilde{\omega}(t) &= -\mathbf{R}_c \left[ \mathbf{E}_1 \tilde{\mathbf{N}}_1 \left( \tilde{\psi}(t), \tilde{c}(t) \right) + \mathbf{A}_2 \tilde{c}(t) \right] \\
&+ \mathbf{R}_T \left[ \mathbf{E}_2 \tilde{\mathbf{N}}_2 \left( \tilde{\psi}(t), \tilde{T}(t) \right) + \mathbf{A}_3 \tilde{T}(t) \right],
\end{aligned} \tag{A.29}$$

$$\begin{aligned}
\frac{d\tilde{c}(t)}{dt} &= -\mathbf{E}_3 \tilde{\mathbf{F}}_1 \left( \tilde{\psi}(t), \tilde{c}(t) \right) + \left[ \mathbf{A}_4 \tilde{c}(t) + \tilde{b}_1 \right] \\
&+ \text{Da} \mathbf{E}_5 f \left( \mathbf{P}_f^T \mathbf{U}_c^k \tilde{c}(t) \right),
\end{aligned} \tag{A.30}$$

$$\begin{aligned}
\frac{d\tilde{T}(t)}{dt} &= -\mathbf{E}_4 \tilde{\mathbf{F}}_2 \left( \tilde{\psi}(t), \tilde{T}(t) \right) + \text{Le} \left[ \mathbf{A}_5 \tilde{T}(t) + \tilde{b}_2 \right] \\
&+ \text{sgn}(\phi) \text{Da} \mathbf{E}_6 f \left( \mathbf{P}_f^T \mathbf{U}_c^k \tilde{c}(t) \right).
\end{aligned} \tag{A.31}$$

Finally, the POD-DEIM approximation is obtained by solving (A.28)–(A.31) and projecting them onto the original dimension.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by Thammasat University Research Fund, Contract No. TUGR 2/12/2562.

## References

- [1] G. Homsy, “Viscous fingering in porous media,” *Annual Review of Fluid Mechanics*, vol. 19, pp. 271–311, 2003.
- [2] P. Sesini, D. A. F. de Souza, and A. L. G. A. Coutinho, “Finite element simulation of viscous fingering in miscible displacements at high mobility-ratios,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 32, no. 3, pp. 292–299, 2010.
- [3] E. T. Brantson, B. Ju, and D. Wu, “Numerical simulation of viscous fingering and flow channeling phenomena in perturbed stochastic fields: finite volume approach with tracer injection tests,” *Arabian Journal for Science and Engineering*, vol. 43, no. 11, pp. 6333–6353, 2018.
- [4] M. Abbaszadeh and M. Dehghan, “Investigation of the Oldroyd model as a generalized incompressible Navier–Stokes equation via the interpolating stabilized element free Galerkin technique,” *Applied Numerical Mathematics*, vol. 150, pp. 274–294, 2020.
- [5] V. Dolci and R. Arina, “Proper orthogonal decomposition as surrogate model for aerodynamic optimization,” *International Journal of Aerospace Engineering*, vol. 2016, Article ID 8092824, 15 pages, 2016.
- [6] G. H. Chang, C. M. Schirmer, and Y. Modarres-Sadeghi, “A reduced-order model for wall shear stress in abdominal aortic aneurysms by proper orthogonal decomposition,” *Journal of Biomechanics*, vol. 54, pp. 33–43, 2017.
- [7] P. Christ and T. Sattelmayer, “Reduced order modelling of flow and mixing in an automobile hvac system using proper orthogonal decomposition,” *Applied Thermal Engineering*, vol. 133, pp. 211–223, 2018.
- [8] B. Karasözen, C. Akkoyunlu, and M. Uzunca, “Model order reduction for nonlinear Schrödinger equation,” *Applied Mathematics and Computation*, vol. 258, pp. 509–519, 2015.

- [9] E. Liberge and A. Hamdouni, "Reduced order modelling method via proper orthogonal decomposition (pod) for flow around an oscillating cylinder," *Journal of Fluids and Structures*, vol. 26, no. 2, pp. 292–311, 2010.
- [10] M. Dehghan and M. Abbaszadeh, "A combination of proper orthogonal decomposition–discrete empirical interpolation method (pod–deim) and meshless local rbf-dq approach for prevention of groundwater contamination," *Computers and Mathematics with Applications*, vol. 75, no. 4, pp. 1390–1412, 2018.
- [11] J. Bremer, P. Goyal, L. Feng, P. Benner, and K. Sundmacher, "Pod-deim for efficient reduction of a dynamic 2d catalytic reactor model," *Computers and Chemical Engineering*, vol. 106, pp. 777–784, 2017.
- [12] F. Ghavamian, P. Tiso, and A. Simone, "Pod–deim model order reduction for strain-softening viscoplasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 458–479, 2017.
- [13] M. Mordhorst, T. Strecker, D. Wirtz, T. Heidlauf, and O. Röhrle, "Pod-deim reduction of computational emg models," *Journal of Computational Science*, vol. 19, pp. 86–96, 2017.
- [14] R. Ștefănescu, A. Sandu, and I. M. Navon, "Pod/deim reduced-order strategies for efficient four dimensional variational data assimilation," *Journal of Computational Physics*, vol. 295, pp. 569–595, 2015.
- [15] W. Bova, E. Lappano, P. G. Catera, and D. Mundo, "Development of a parametric model order reduction method for laminated composite structures," *Composite Structures*, vol. 243, article 112219, 2020.
- [16] S. van Ophem, E. Deckers, and W. Desmet, "Parametric model order reduction without a priori sampling for low rank changes in vibro-acoustic systems," *Mechanical Systems and Signal Processing*, vol. 130, pp. 597–609, 2019.
- [17] T. Bui-Thanh, K. Willcox, and O. Ghattas, "Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications," *AIAA Journal*, vol. 46, no. 10, pp. 2520–2529, 2008.
- [18] C. Poussot-Vassal and D. Sipp, "Parametric reduced order dynamical model construction of a fluid flow control problem," *IFAC-Papers OnLine*, vol. 48, no. 26, pp. 133–138, 2015.
- [19] S. Swernath and S. Pushpavanam, "Viscous fingering in a horizontal flow through a porous medium induced by chemical reactions under isothermal and adiabatic conditions," *The Journal of Chemical Physics*, vol. 127, no. 20, article 204701, 2007.
- [20] S. Chaturantabut and D. 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, no. 4, pp. 337–353, 2011.
- [21] S. Volkwein, "Proper orthogonal decomposition: theory and reduced-order modelling," in *Lecture notes*, University of Konstanz, 2013, <http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf>.
- [22] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.
- [23] D. Wirtz, D. C. Sorensen, and B. Haasdonk, "A posteriori error estimation for deim reduced nonlinear dynamical systems," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. A311–A338, 2014.
- [24] S. Chaturantabut and D. C. Sorensen, "A state space error estimate for pod-deim nonlinear model reduction," *SIAM Journal on Numerical Analysis*, vol. 50, no. 1, pp. 46–63, 2012.