A Discrete Empirical Interpolation Method (DEIM) is applied in conjunction with Proper Orthogonal Decomposition (POD) to construct a nonlinear reduced-order model of finite difference discretized system used in the simulation of nonlinear miscible viscous fingering in a 2-D porous medium. POD is first applied to extract a low-dimensional basis that optimally captures the dominant characteristics of the system trajectory. This basis is then used in a Galerkin projection scheme to construct a reduced-order system. DEIM is then applied to greatly improve the efficiency in computing the projected nonlinear terms in the POD reduced system. DEIM achieves a complexity reduction of the nonlinearities which is proportional to the number of reduced variables while POD retains a complexity proportional to the original number of variables. Numerical results demonstrate that the dynamics of the viscous fingering in the full-order system of dimension 15000 can be captured accurately by the POD-DEIM reduced system of dimension 40 with the computational time reduced by factor of $O(1000)$.

Keywords: Nonlinear Model Reduction, Proper Orthogonal Decomposition, Empirical Interpolation Methods, Nonlinear Partial Differential Equations, Miscible Viscous Fingering

AMS Subject Classification: 65P99, 65M99, 37M05

1. Introduction

This paper describes application of a very efficient method for model order reduction to nonlinear miscible flow in porous media. We use a Discrete Empirical Interpolation Method (DEIM) developed in [1] for model reduction of general nonlinear systems of ordinary differential equations (ODEs). Here, we demonstrate that this method can provide a vast reduction in complexity arising from nonlinearities as compared to that of Proper Orthogonal Decomposition (POD). As a result, simulation times can be decreased by as much as three orders of magnitude.

POD is an efficient method for constructing a problem specific set of basis functions with global support that capture the dominant characteristics of a dynamical system. When the dynamical system is described by partial differential equations (PDEs), POD generates an optimal global basis generally from a set of samples of states (the trajectories) which are numerical solutions of the finite dimensional system obtained with virtually any standard high fidelity grid-based discretization scheme. Fine scale details at grid points are encoded in this global basis. It is therefore possible to use only few dominant POD basis functions in the Galerkin projection method to obtain an accurate representation in reduced-order system. POD has been successfully used for model reduction in various applications such as compressible flow [2], fluid dynamics [3], aerodynamics [4], and optimal control [5].
Unfortunately, for general nonlinear and/or non-affine parametrized PDEs, the efficiency in solving the reduced-order systems constructed from standard Galerkin projection with any reduced globally supported basis set, including the one from POD, is limited to the linear or bilinear part, as indicated in [6] for a finite element (FE) scheme, and in [1] for a finite difference (FD) scheme. In particular, the complexity for computing a projected nonlinear term in this reduced-order system still depends on the dimension of the original full-order system. This paper is concerned with dimension reduction for FD discretized system of PDEs. The Discrete Empirical Interpolation Method (DEIM) from [1] provides a dimension reduction of the nonlinear term that has complexity proportional to the number of reduced variables.

DEIM [1] is a discrete variation of Empirical Interpolation Method (EIM) proposed by Barrault, Maday, Nguyen and Patera in [6]. A considerable reduction in complexity is achieved by DEIM because evaluating the approximate nonlinear term does not require a prolongation of the reduced state variables back to the original high dimensional state approximation required to evaluate the nonlinearity in the POD approximation. DEIM therefore improves the efficiency of the POD approximation and achieves a complexity reduction of the nonlinear term with a complexity proportional to the number of reduced variables. An error bound for the DEIM approximation is given in [1] which shows it is nearly as good as the optimal POD approximation.

As mentioned above, we shall describe an application of DEIM combined with POD to provide dimension reduction of a nonlinear miscible flow with viscous fingering (VF) in porous media. VF instability occurs when a less viscous fluid moves through a porous medium occupied with another more viscous fluid, which leads further to the development of finger-shaped intrusions flowing between the two fluids. This flow process is commonly used to describe many important physical phenomena such as oil recovery process, chromatographic separation, and pollutant dispersion.

An extensive number of studies have been done both experimentally and numerically to observe, investigate, and predict the flow displacement behavior as well as the fingering mechanisms, such as spreading, shielding, tip splitting, and coalescence (see, e.g. [7–14] for reviews and more details). Numerical simulations have been carried out by various discretization schemes such as finite difference, finite element, discontinuous Galerkin and Pseudo-Fourier spectral methods. The dimension of the discretized system is determined by the number of grid points in the flow domain. Usually finer grids and smaller time steps are required to capture the fine structure of the viscous fingers to obtain numerical solutions with higher accuracy. This results in a significant increase in the computational time and data storage requirements. Dimension reduction can overcome this difficulty. However, to our knowledge, there are only few relevant articles on dimension reduction for VF simulation such as in [15] and [16], which is primarily based on POD combining with artificial neuron network for constructing a model for predicting fluid displacement.

The formulation of the governing equations describing the nonlinear miscible VF in a 2-D porous medium, presented here in Section 2, as well as the FD discretization scheme are taken from [13]. The matrix form of the full-order system and its
corresponding reduced-order systems both from POD and POD with DEIM are given in Section 3 and Section 4. A practical method for computing POD basis of a sampled set from high-dimensional subspace is discussed in Section 5. The numerical results are presented in Section 6. It is shown that the VF dynamics in the full-order system of dimension 15000 can be captured accurately by the POD-DEIM reduced system of dimension 40 with the computational time reduced by factor of $O(1000)$. The numerical comparison of the POD and POD-DEIM reduced systems is also presented and indicates a significant reduction in computational time when DEIM is further applied to the POD reduced system. To illustrate a potential usefulness of dimension reduction for parametrized systems, POD-DEIM approach is also used to construct a single reduced-order model that can provide an accurate representation of the original full-order system over the entire specified range of parameter values. Further applications of POD-DEIM approach is also shown on a similar problem of miscible flow with VF induced by chemical reactions. Finally, our conclusions and possible extensions are discussed in Section 7.

2. Governing Equations

The equations of motion given in [13] are used here to describe the viscous fingering in horizontal flow of an incompressible fluid through a 2-D homogeneous porous medium of length $L_x$ (horizontal) and width $L_y$ (vertical) with a constant permeability $K$. The fluid is assumed to be injected horizontally from the left boundary with a uniform velocity $U$. Assume also that the porous medium is already occupied by another fluid with higher viscosity than the injected fluid and that the two fluids are miscible. The flow evolution is then governed by:

\[ \nabla \cdot \mathbf{u} = 0 \]  
\[ \nabla P = -\frac{\mu}{K} \mathbf{u} \]  
\[ \frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = D \nabla^2 c + f(c), \]  
\[ \rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = D_T \nabla^2 T + (-\Delta H)f(c), \]

with the rate of autocatalytic reaction defined as $f(c) = -c(k_a + k_r c)(c - c_1)$, where $k_a$, $k_r$, $c_1$, $\rho$, $c_p$ are constant parameters; $\mathbf{u} = [u, w]^T \in \mathbb{R}^2$ is the velocity with components in $x$ and $y$ coordinates; $P$ is the pressure; $c$ is the concentration of the injected fluid; $T$ is the temperature; $\mu$ is the viscosity depending on $c$ and $T$; $D$, $D_T$ and $\Delta H$ denote diffusion coefficients and enthalpy, which are assumed to be constant. Note that (1) is the continuity equation of an incompressible fluid, (2) is the momentum equation in the form of Darcy’s law, (3) and (4) are convection-diffusion-reaction equations for the concentration and the temperature of the injected fluid. A common procedure for solving the system of equations (1)-(4) is to first nondimensionalize the system and then convert it into the form of streamfunction and vorticity which is finally solved numerically by a discretization scheme. Define the nondimensional variables and parameters to be:

\[ \tilde{x} = \left[ \frac{x}{L_x}, \frac{y}{L_y} \right]^T = \frac{1}{D/U} [x, y]^T - [t, 0]^T, \quad \tilde{u} = \left[ \frac{u}{U}, \frac{w}{U} \right]^T = \frac{1}{U} [u, w]^T - [1, 0]^T, \quad \tilde{t} = \frac{t}{D/U}, \quad \tilde{T} = \frac{T - T_1}{T_2 - T_1}, \quad \tilde{c} = \frac{c}{c_1}, \quad \tilde{\mu} = \frac{\mu(c,T)}{\mu_1}, \]

\[ P = \frac{P}{\mu D/K}, \quad \phi = \frac{-\Delta H c_1}{pc_p T_1}, \]
where \( c_1, T_1, \) and \( \mu_1 \) are the initial concentration, temperature, and viscosity of the injected fluid. The nondimensionalized equations governing this system in the moving reference frame with constant velocity \( U \) in horizontal direction are given by

\[
\begin{align*}
\frac{\partial \bar{u}}{\partial \bar{x}} + \frac{\partial \bar{w}}{\partial \bar{y}} &= 0 \quad (5) \\
\frac{\partial \bar{P}}{\partial \bar{x}} &= -\bar{\mu} (\bar{u} + 1) \quad (6) \\
\frac{\partial \bar{P}}{\partial \bar{y}} &= -\bar{\mu} \bar{w} \quad (7) \\
\frac{\partial \bar{c}}{\partial \bar{t}} + \bar{u} \frac{\partial \bar{c}}{\partial \bar{x}} + \bar{w} \frac{\partial \bar{c}}{\partial \bar{y}} &= \frac{\partial^2 \bar{c}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{c}}{\partial \bar{y}^2} + Da \bar{f}(\bar{c}), \quad (8) \\
\frac{\partial \bar{T}}{\partial \bar{t}} + \bar{u} \frac{\partial \bar{T}}{\partial \bar{x}} + \bar{w} \frac{\partial \bar{T}}{\partial \bar{y}} &= Le \left[ \frac{\partial^2 \bar{T}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{T}}{\partial \bar{y}^2} \right] + \text{sgn}(\phi) Da \bar{f}(\bar{c}), \quad (9)
\end{align*}
\]

where \( \bar{f} = -\bar{c}(\bar{c} - 1)(\bar{c} + d), \) \( d = k_a/k_c c_1; \) Da (Damköhler number) and Le (Lewis number) are constant dimensionless parameters; \( \text{sgn}(\phi) = 1 \) for exothermic reactions and \( \text{sgn}(\phi) = -1 \) for endothermic reactions. The variables with the overbar symbol are dimensionless quantities. For notational convenience, the bar symbol will be dropped and all variables mentioned in the remaining parts of this paper are assumed to be dimensionless. The viscosity of the fluid is assumed to vary with concentration and temperature as

\[
\mu = \mu_0 e^{-R_c c + R_T T}, \quad \text{or} \quad \frac{1}{\mu} \frac{\partial \mu}{\partial c} = -R_c, \quad \frac{1}{\mu} \frac{\partial \mu}{\partial T} = R_T,
\]

where \( R_c \) and \( R_T \) are given constant parameters determining the effects of concentration and temperature to the viscosity. Define a streamfunction \( \psi(x, y) \) so that

\[
\begin{align*}
u &= \frac{\partial \psi}{\partial y}, \quad w = -\frac{\partial \psi}{\partial x},
\end{align*}
\]

and define the vorticity \( \omega(x, y) \) as

\[
\omega = (\nabla \times \mathbf{u}) \cdot \mathbf{k} = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial y},
\]

where \( \mathbf{k} = [0, 0, 1]^T. \) The equations (5)-(9) are then converted to the equations
described in terms of streamfunction $\psi$ and vorticity $\omega$ as:

$$\nabla^2 \psi = \omega \quad (10)$$

$$\omega = -R_c \left( \psi_x c_x + \psi_y c_y \right) + R_T \left( \psi_x T_x + \psi_y T_y \right) \quad (11)$$

$$\frac{\partial c}{\partial t} + \psi_y c_x - \psi_x c_y = \nabla^2 c + Da f(c), \quad (12)$$

$$\frac{\partial T}{\partial t} + \psi_y T_x - \psi_x T_y = Le \nabla^2 T + \text{sgn}(\phi) Da f(c), \quad (13)$$

where $\psi_x = \frac{\partial \psi}{\partial x}, \psi_y = \frac{\partial \psi}{\partial y}, c_x = \frac{\partial c}{\partial x}, c_y = \frac{\partial c}{\partial y}, T_x = \frac{\partial T}{\partial x}, T_y = \frac{\partial T}{\partial y}$. The unknowns of these transformed equations (10)-(13) are $c(x,y,t), T(x,y,t), \psi(x,y,t)$, and $\omega(x,y,t)$ with spatial points $(x,y) \in \Omega$, and the dimensionless domain $\Omega = [0, \alpha Pe] \times [0, Pe] \subset \mathbb{R}^2$, $\alpha := L_x/L_y$ is a constant aspect ratio and time $t \in [0, t_f]$ with (dimensionless) final simulation time $t_f$. Note that the dimensionless parameter Péclet number $Pe$, defined as $Pe := UL_x/D$, determines the ratio of the rate of convective transport to the rate of diffusive transport and it also represents the length of the dimensionless flow domain.

The nonlinear functions in this system (10)-(13) are:

$$N(\psi, v) = \psi_x v_x + \psi_y v_y \quad (14)$$

$$F(\psi, v) = \psi_y v_x - \psi_x v_y \quad (15)$$

$$f(c) = -c(c-1)(c+d). \quad (16)$$

In the system of equations (10)-(13), periodic boundary conditions are imposed along top-bottom boundaries for $c$, $T$, $\psi$ and Dirichlet boundary conditions are imposed along left-right boundaries for $c$, $T$, $\psi$. That is, for $x \in [0, \alpha Pe], t \in [0, t_f],$

$$c(x,0,t) = c(x,Pe,t), \quad c_y(x,0,t) = c_y(x,Pe,t) \quad (17)$$

$$T(x,0,t) = T(x,Pe,t), \quad T_y(x,0,t) = T_y(x,Pe,t) \quad (18)$$

$$\psi(x,0,t) = \psi(x,Pe,t), \quad \psi_y(x,0,t) = \psi_y(x,Pe,t), \quad (19)$$

for $y \in [0, Pe], t \in [0, t_f]$, at the boundaries $x = 0$ and $x = \alpha Pe$, we have

$$c(0,y,t) = 1, \quad c(\alpha Pe, y, t) = 0 \quad (20)$$

$$\psi(0,y,t) = 0, \quad \psi(\alpha Pe, y, t) = 0. \quad (21)$$

exothermic : $T(0,y,t) = 1, T(\alpha Pe, y, t) = 0 \quad (22)$

endothermic : $T(0,y,t) = 0, T(\alpha Pe, y, t) = 1. \quad (23)$

No boundary conditions are required for the vorticity $\omega$, since it is defined by an
algebraic expression. The initial conditions are:
\[
c(x, y, 0) = T(x, y, 0) = \begin{cases} 1 & x \leq \hat{x} \\ 0 & x > \hat{x} \end{cases},
\]
for all \( y \in [0, Pe] \), where \( \hat{x} \) is the interface location (this paper sets \( \hat{x} = \alpha Pe/2 \)) and \( \psi(x, y, 0) = 0 \) for all \( (x, y) \in \Omega \).

3. Discretized System by Finite Difference (FD) Method

Central finite differences are used to construct a spatial discretization of equations (10)-(13) to obtain a system of nonlinear ODEs (29)-(32). Then the forward time integration with predictor-corrector scheme introduced in [13] is applied to (29)-(32) to obtain FD solution at each time step.

Let \( 0 = x_0 < x_1 < \cdots < x_{n_x} < x_{n_x+1} = \alpha Pe \) and \( 0 = y_0 < y_1 < \cdots < y_{n_y} < y_{n_y+1} = Pe \) be equally spaced points on \( x \)-axis and \( y \)-axis for generating the grid points on the dimensionless domain \( \Omega = [0, \alpha Pe] \times [0, Pe] \) with \( dx = \alpha Pe/(n_x + 1) \) and \( dy = Pe/(n_y + 1) \). Let \( c_{ij}(t) \), \( T_{ij}(t) \), \( \psi_{ij}(t) \), and \( \omega_{ij}(t) \) denote the finite difference approximations of \( c(x_i, y_j, t) \), \( T(x_i, y_j, t) \), \( \psi(x_i, y_j, t) \), and \( \omega(x_i, y_j, t) \). Define vectors of unknown variables of dimension \( n := n_y n_x \) as
\[
\mathbf{c}(t) = [c_{11}(t), c_{21}(t), \ldots, c_{n_x1}(t), \ldots, c_{n_y1}(t), c_{12n}(t), \ldots, c_{n_xn}(t)]^T \in \mathbb{R}^n \quad (25)
\]
\[
\mathbf{T}(t) = [T_{11}(t), T_{21}(t), \ldots, T_{n_x1}(t), \ldots, T_{n_y1}(t), T_{12n}(t), \ldots, T_{n_xn}(t)]^T \in \mathbb{R}^n \quad (26)
\]
\[
\mathbf{\psi}(t) = [\psi_{11}(t), \psi_{21}(t), \ldots, \psi_{n_x1}(t), \ldots, \psi_{n_y1}(t), \psi_{12n}(t), \ldots, \psi_{n_xn}(t)]^T \in \mathbb{R}^n \quad (27)
\]
\[
\mathbf{\omega}(t) = [\omega_{11}(t), \omega_{21}(t), \ldots, \omega_{n_x1}(t), \ldots, \omega_{n_y1}(t), \omega_{12n}(t), \ldots, \omega_{n_xn}(t)]^T \in \mathbb{R}^n \quad (28)
\]

The corresponding spatial finite difference discretized system of (10)-(13) then becomes a system of nonlinear ODEs coupled with algebraic equations which can be written in matrix form as follows. For \( t \in [0, t_f] \),
\[
\frac{d\mathbf{c}(t)}{dt} = -\mathbf{F}(\mathbf{\psi}(t), \mathbf{c}(t)) + [\mathbf{A}\mathbf{c}(t)] + \text{Daf}(\mathbf{c}(t)) \quad (29)
\]
\[
\frac{d\mathbf{T}(t)}{dt} = -\mathbf{F}(\mathbf{\psi}(t), \mathbf{T}(t)) + \text{Le}[\mathbf{A}\mathbf{T}(t)] + \text{sgn}(\phi)\text{Daf}(\mathbf{c}(t)) \quad (30)
\]
\[
\omega(t) = -R_c[\mathbf{N}(\mathbf{\psi}(t), \mathbf{c}(t)) + \mathbf{A}_y\mathbf{c}(t)] + R_T[\mathbf{N}(\mathbf{\psi}(t), \mathbf{T}(t)) + \mathbf{A}_y\mathbf{T}(t)] \quad (31)
\]
\[
\mathbf{A}\mathbf{\psi}(t) = -\mathbf{\omega}(t), \quad (32)
\]

where the nonlinear functions \( \mathbf{F}, \mathbf{N}, \mathbf{f} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \) are defined as
\[
\mathbf{F}(\mathbf{\psi}, \mathbf{c}) = (\mathbf{A}_y\mathbf{\psi}) \ast (\mathbf{A}_x \mathbf{c} + \mathbf{b}_x) - (\mathbf{A}_x \mathbf{\psi}) \ast (\mathbf{A}_y \mathbf{c}), \quad (33)
\]
\[
\mathbf{N}(\mathbf{\psi}, \mathbf{c}) = (\mathbf{A}_x \mathbf{\psi}) \ast (\mathbf{A}_x \mathbf{c} + \mathbf{b}_x) + (\mathbf{A}_y \mathbf{\psi}) \ast (\mathbf{A}_y \mathbf{c}), \quad (34)
\]
\[
\mathbf{f}(\mathbf{c}) = -\mathbf{c} \ast (\mathbf{c} - 1) \ast (\mathbf{c} + d), \quad (35)
\]
with ‘.*’ denoting componentwise multiplication as used in MATLAB; for \( n = n_x n_y \),

\[
A = \begin{pmatrix}
\frac{1}{2}B & B & \frac{1}{2}B \\
B & E & B \\
B & E & B \\
\cdots & \cdots & \cdots \\
B & E & B \\
\cdots & \cdots & \cdots \\
\frac{1}{2}B & B & \frac{1}{2}B
\end{pmatrix} \in \mathbb{R}^{n \times n},
\]

(36)

with

\[
E = \begin{pmatrix}
\alpha & \beta & \beta & \cdots & \beta \\
\beta & \alpha & \beta & \cdots & \beta \\
\beta & \beta & \alpha & \cdots & \beta \\
\beta & \beta & \beta & \cdots & \alpha \\
\beta & \beta & \beta & \cdots & \beta
\end{pmatrix} \in \mathbb{R}^{n \times n_x}.
\]

(37)

\[
B = \begin{pmatrix}
\gamma \\
\gamma \\
\gamma \\
\gamma \\
\gamma
\end{pmatrix} \in \mathbb{R}^{n \times n_x}.
\]

(37)

for \( \alpha = -\frac{2}{dx^2} - \frac{2}{dy^2} \), \( \beta = \frac{1}{dx^2} \), \( \gamma = \frac{1}{dy^2} \).

(38)

\[
b = \begin{pmatrix}
\beta, 0, 0, \ldots, 0 \\
\beta, 0, 0, \ldots, 0 \\
\beta, 0, 0, \ldots, 0 \\
\beta, 0, 0, \ldots, 0 \\
\beta, 0, 0, \ldots, 0
\end{pmatrix}^T \in \mathbb{R}^n.
\]

(39)

\[
A_x = \frac{1}{2dx} \begin{pmatrix}
K \\
\vdots \\
K
\end{pmatrix} \in \mathbb{R}^{n \times n}, \quad K = \begin{pmatrix}
0 & 1 \\
-1 & 0 & 1 \\
\ddots & \ddots & \ddots \\
-1 & 0 & 1 \\
-1 & 0
\end{pmatrix} \in \mathbb{R}^{n_x \times n_x},
\]

(40)

\[
b_x = \begin{pmatrix}
\frac{1}{2dx}, 0, 0, \ldots, 0 \\
\frac{1}{2dx}, 0, 0, \ldots, 0 \\
\frac{1}{2dx}, 0, 0, \ldots, 0 \\
\frac{1}{2dx}, 0, 0, \ldots, 0 \\
\frac{1}{2dx}, 0, 0, \ldots, 0
\end{pmatrix}^T \in \mathbb{R}^n,
\]

(41)

\[
A_y = \frac{1}{2dy} \begin{pmatrix}
-\frac{1}{2}I & I & -\frac{1}{2}I \\
-\frac{1}{2}I & 0 & I \\
-\frac{1}{2}I & 0 & I \\
\cdots & \cdots & \cdots \\
-\frac{1}{2}I & 0 & I \\
\frac{1}{2}I & -1 & \frac{1}{2}I
\end{pmatrix} \in \mathbb{R}^{n \times n}, \quad I = \begin{pmatrix}
1 \\
\ddots \\
1
\end{pmatrix} \in \mathbb{R}^{n_x \times n_x}.
\]

(42)

The details for constructing these coefficient matrices are provided in Appendix B. The system of equations (29)-(32) can be solve by using the algorithm in [13] which is reviewed in Appendix C. In general, the discretized system for this nonlinear VF has to be very large to capture the fine details of fingers flowing through the domain, especially for high Péclet number. This therefore causes substantially increases in computational time and memory storage which may further make it impossible to perform the simulation in a reasonable computational time. Next section introduces efficient model reduction techniques that can be used to overcome this difficulty.
4. The Reduced-Order System

As in [1], the Proper Orthogonal Decomposition (POD) and Discrete Empirical Interpolation Method (DEIM) are applied to construct a reduced-order system of the full-order system (29)-(32) described in the previous section. Sections 4.1 and 4.2 show the details of constructing this reduced-order system.

4.1. POD reduced system

POD is an efficient method for extracting orthonormal basis elements that contain characteristics of the space of expected solutions which is defined as the span of the snapshots. In this setting, snapshots are the sampled (numerical) solutions at particular time steps or at particular parameter values. POD gives an optimal set of basis vectors minimizing the mean square error from approximating these snapshots. Formal definition of POD is reviewed in Appendix A. In this finite dimensional setting, POD is in fact just the singular value decomposition (SVD).

Let \( \hat{C} = [c^1, \ldots, c^{n_s}] \), \( \hat{T} = [T^1, \ldots, T^{n_s}] \), \( \hat{M} = [\omega^1, \ldots, \omega^{n_s}] \), \( \hat{\Psi} = [\psi^1, \ldots, \psi^{n_s}] \) \( \in \mathbb{R}^{n \times n_s} \) be the snapshot matrices. E.g. \( c^j \) corresponds to the solution of the FD discretized system at time \( t_j \) and similarly for \( T^j \), \( \omega^j \), and \( \psi^j \). Let \( r_c = \text{rank}(\hat{C}) \), \( r_T = \text{rank}(\hat{T}) \), \( r_\omega = \text{rank}(\hat{M}) \), \( r_\psi = \text{rank}(\hat{\Psi}) \). Let \( k \leq \min\{r_c, r_T, r_\omega, r_\psi\} \). The POD basis of dimension \( k \) of the snapshots \( \{c^j\}_{j=1}^{n_s} \) consists of the leading \( k \) columns of \( \hat{U} \) so that \( U = \hat{U}(::, 1 : k) \in \mathbb{R}^{n \times k} \), is the matrix whose columns corresponding to the POD basis of dimension \( k \) of the snapshots \( \{c^j\}_{j=1}^{n_s} \).

Let \( \hat{C} = [c^1, \ldots, c^{n_s}] \), \( \hat{T} = [T^1, \ldots, T^{n_s}] \), \( \hat{M} = [\omega^1, \ldots, \omega^{n_s}] \), \( \hat{\Psi} = [\psi^1, \ldots, \psi^{n_s}] \) \( \in \mathbb{R}^{n \times n_s} \) be the snapshot matrices. E.g. \( c^j \) corresponds to the solution of the FD discretized system at time \( t_j \) and similarly for \( T^j \), \( \omega^j \), and \( \psi^j \). Let \( r_c = \text{rank}(\hat{C}) \), \( r_T = \text{rank}(\hat{T}) \), \( r_\omega = \text{rank}(\hat{M}) \), \( r_\psi = \text{rank}(\hat{\Psi}) \). Let \( k \leq \min\{r_c, r_T, r_\omega, r_\psi\} \). The POD basis of dimension \( k \) of the snapshots \( \{c^j\}_{j=1}^{n_s} \) consists of the leading \( k \) columns of \( \hat{U} \) so that \( U = \hat{U}(::, 1 : k) \in \mathbb{R}^{n \times k} \), is the matrix whose columns corresponding to the POD basis of dimension \( k \) of the snapshots \( \{c^j\}_{j=1}^{n_s} \), where

\[
\hat{C} = \hat{U} \Sigma^c (Z^c)^T
\]

is the SVD of \( \hat{C} \) with \( \Sigma^c = \text{diag}(\sigma^c_1, \ldots, \sigma^c_{r_c}) \in \mathbb{R}^{r_c \times r_c}; \sigma^c_1 \geq \cdots \geq \sigma^c_{r_c} > 0 \) and \( \hat{U} \in \mathbb{R}^{n \times r_c}, Z^c \in \mathbb{R}^{n \times r_c} \) having orthonormal columns. Similarly, let \( Q, V, W \in \mathbb{R}^{n \times k} \) be matrices whose columns corresponding to the POD basis of dimension \( k \) of the snapshots \( \{T^j\}_{j=1}^{n_s}, \{\omega^j\}_{j=1}^{n_s}, \{\psi^j\}_{j=1}^{n_s} \). Hence, the POD basis of the snapshots \( \{c^j\}_{j=1}^{n_s} \) consists of the leading \( k \) columns of \( \hat{U} \) so that \( U = \hat{U}(::, 1 : k) \in \mathbb{R}^{n \times k} \), is the matrix whose columns corresponding to the POD basis of dimension \( k \) of the snapshots \( \{c^j\}_{j=1}^{n_s} \), where

\[
\begin{align*}
c &\leftarrow U\tilde{c}, \\
T &\leftarrow QT, \\
\omega &\leftarrow V\tilde{\omega}, \\
\psi &\leftarrow W\tilde{\psi},
\end{align*}
\]

with reduced variables \( \tilde{c}, \tilde{T}, \tilde{\omega}, \tilde{\psi} \in \mathbb{R}^k \), and then enforcing the Galerkin orthogonality condition of the residual to the approximation space which is equivalent to premultiplying equation (29) by \( U^T \), equation (30) by \( Q^T \), and equations (31) and
Now, the coefficient matrices of the linear term of the POD reduced system can be
(32) by $V^T$. The resulting reduced-order system is of the form
\[
\frac{d\tilde{c}(t)}{dt} = -U^T\tilde{F}_1(\tilde{\psi}(t), \tilde{c}(t)) + [U^TAU \tilde{c}(t) + U^Tb] + DaU^Tf(U\tilde{c}(t))
\]  
(44)
\[
\frac{d\tilde{T}(t)}{dt} = -Q^T\tilde{F}_2(\tilde{\psi}(t), \tilde{T}(t)) + [Q^TAQ \tilde{T}(t) + Q^Tb] + \text{sgn}(\psi)DaQ^Tf(U\tilde{c}(t))
\]  
(45)
\[
\tilde{\omega}(t) = -Re \left[ V^T\tilde{N}_1(\tilde{\psi}(t), \tilde{c}(t)) + V^TA_2U \tilde{c}(t) \right] + RT \left[ V^T\tilde{N}_2(\tilde{\psi}(t), \tilde{T}(t)) + V^TA_3Q \tilde{T}(t) \right]
\]  
(46)
\[
V^TAW \tilde{\psi}(t) = -\tilde{\omega}(t).
\]  
(47)

where $\tilde{F}_1, \tilde{F}_2, \tilde{N}_1, \tilde{N}_2: \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}^n$,
\[
\tilde{F}_1(\tilde{\psi}, \tilde{c}) = (A_yW \tilde{\psi}) \ast (A_xU \tilde{c} + b_x) - (A_xW \tilde{\psi}) \ast (A_yU \tilde{c}),
\]  
(48)
\[
\tilde{F}_2(\tilde{\psi}, \tilde{T}) = (A_yW \tilde{\psi}) \ast (A_yQ \tilde{T} + b_x) - (A_xW \tilde{\psi}) \ast (A_yQ \tilde{T}),
\]  
(49)
\[
\tilde{N}_1(\tilde{\psi}, \tilde{c}) = (A_xW \tilde{\psi}) \ast (A_xU \tilde{c} + b_x) + (A_yW \tilde{\psi}) \ast (A_yU \tilde{c}),
\]  
(50)
\[
\tilde{N}_2(\tilde{\psi}, \tilde{T}) = (A_xW \tilde{\psi}) \ast (A_yQ \tilde{T} + b_x) + (A_yW \tilde{\psi}) \ast (A_yQ \tilde{T}),
\]  
(51)
\[
f(U\tilde{c}) = -U\tilde{c} \ast (U\tilde{c} - 1) \ast (U\tilde{c} + d).
\]  
(52)

Now, the coefficient matrices of the linear term of the POD reduced system can be
precomputed in formulas (44)-(47) to give:
\[
A_1 = U^TAU, \quad A_2 = Q^TAQ, \quad A_3 = V^TA_yU, \quad A_4 = V^TA_yQ, \quad A_5 = V^TAW,
\]  
(53)
\[
b_1 = U^Tb, \quad b_2 = Q^Tb.
\]

Also, the coefficient matrices in the nonlinear functions in (48)-(52) can be pre-
computed to give:
\[
A_yW, \quad A_xU, \quad A_xW, \quad A_yU, \quad A_xQ, \quad A_yQ.
\]  
(54)

These precomputed matrices are then retained and re-used in all time steps. How-
ever, the nonlinear terms in equations (44)-(47):
\[
U^T\tilde{F}_1(\tilde{\psi}, \tilde{c}), \quad U^Tf(U\tilde{c}), \quad Q^T\tilde{F}_2(\tilde{\psi}, \tilde{T}), \quad Q^Tf(U\tilde{c}), \quad V^T\tilde{N}_1(\tilde{\psi}, \tilde{c}), \quad V^T\tilde{N}_2(\tilde{\psi}, \tilde{T})
\]  
(55)

still have computational complexities depending on the dimension $n = n_xn_y$ of the
original system. The Discrete Empirical Interpolation Method (DEIM) is used to
remove this dependency.

4.2. POD-DEIM reduced-order system

The projected nonlinear function in (55) can be approximated by DEIM in the form
that enables precomputation so that evaluating the approximate nonlinear term
from DEIM does not require a prolongation of the reduced state variables back to
the original high dimensional state approximation as is required to evaluate the
nonlinearity in the original POD approximation. Only a few entries of the original

nonlinear term corresponding to the specially selected interpolation indices from DEIM must be evaluated at each time step. The DEIM approximation is given formally in Definition 4.1 and the procedure for selecting DEIM indices is shown in Algorithm 4.1. It has been shown in [1] that each DEIM index is selected to limit growth of a global error bound relating the DEIM approximation to the full optimal POD approximation.

**Definition 4.1:** Let \( f : \mathcal{D} \rightarrow \mathbb{R}^n \) be a nonlinear vector-valued function with \( \mathcal{D} \subset \mathbb{R}^d \), for some positive integer \( d \). Let \( \{ \mathbf{u}_\ell \}_{\ell=1}^m \subset \mathbb{R}^n \) be a linearly independent set, for \( m = 1, \ldots, n \). For \( \tau \in \mathcal{D} \), the DEIM approximation of order \( m \) for \( f(\tau) \) in the space spanned by \( \{ \mathbf{u}_\ell \}_{\ell=1}^m \) is given by

\[
\hat{f}(\tau) := \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}(\tau),
\]

where \( \mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n \times m} \) and \( \mathbf{P} = [\mathbf{e}_{\varphi_1}, \ldots, \mathbf{e}_{\varphi_m}] \in \mathbb{R}^{n \times m} \) with \( \{ \varphi_1, \ldots, \varphi_m \} \) being the output from Algorithm 4.1 with the input basis \( \{ \mathbf{u}_\ell \}_{\ell=1}^m \).

**Algorithm 4.1 : DEIM**

**INPUT:** \( \{ \mathbf{u}_\ell \}_{\ell=1}^m \subset \mathbb{R}^n \) linearly independent  
**OUTPUT:** \( \hat{\varphi} = [\varphi_1, \ldots, \varphi_m]^T \in \mathbb{R}^m 

1: \([|\rho| \varphi_1] = \max\{|\mathbf{u}_1|\} 
2: \mathbf{U} = [\mathbf{u}_1], \mathbf{P} = [\mathbf{e}_{\varphi_1}], \hat{\varphi} = [\varphi_1] 
3: \text{for } \ell = 2 \text{ to } m \text{ do} 
4: \text{ Solve } (\mathbf{P}^T \mathbf{U})\mathbf{c} = \mathbf{P}^T \mathbf{u}_\ell \text{ for } \mathbf{c} 
5: \mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c} 
6: \([|\rho| \varphi_\ell] = \max\{|\mathbf{r}|\} 
7: \mathbf{U} \leftarrow [\mathbf{U} \ \mathbf{u}_\ell], \mathbf{P} \leftarrow [\mathbf{P} \ \mathbf{e}_{\varphi_\ell}], \hat{\varphi} \leftarrow [\hat{\varphi} \ \varphi_\ell] 
8: \text{end for} 

The notation \( \max \) in Algorithm 4.1 is the same as the function \( \max \) in MATLAB. Thus, \([|\rho| \varphi_\ell] = \max\{|\mathbf{r}|\} \) implies \(|\rho| = |r_{\varphi_\ell}| = \max_{i=1,\ldots,n}\{|r_i|\} \), with the smallest index taken in case of a tie.

From Algorithm 4.1, the DEIM procedure constructs a set of indices inductively on the input basis in such a way that, at each iteration, the current selected index captures the maximum variation of the input basis vectors. The order of the input basis \( \{ \mathbf{u}_\ell \}_{\ell=1}^m \) according to the dominant singular values is important and an error analysis in [1] indicates that the POD basis is a suitable choice for this algorithm.

Let \( \mathbf{U}^{F_1}, \mathbf{U}^{F_2}, \mathbf{U}^{N_1}, \mathbf{U}^{N_2}, \mathbf{U}^f \in \mathbb{R}^{n \times m}, m \leq n \) be the matrices whose columns containing the POD bases of the nonlinear functions \( F_1, F_2, N_1, N_2 \) and \( f \) defined in (48)-(52). These POD bases are used to select the sets of \( m \) interpolation indices from DEIM algorithm. Let \( \hat{\varphi}^{F_1}, \hat{\varphi}^{F_2}, \hat{\varphi}^{N_1}, \hat{\varphi}^{N_2}, \hat{\varphi}^f \) be the DEIM interpolation indices of the nonlinear functions defined in (48)-(52). Let \( \mathbf{P}_{F_1} \in \mathbb{R}^{n \times m} \) be the matrix whose \( j \)-th column is the \( \varphi_j^{F_1} \)-th column of the identity matrix, i.e., it is the vector \([0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^n \), having all zero entries except one at the entry \( \varphi_j^{F_1} \), for \( j = 1, \ldots, m \). Define \( \mathbf{P}_{F_2}, \mathbf{P}_{N_1}, \mathbf{P}_{N_2}, \mathbf{P}_f \in \mathbb{R}^{n \times m} \) in a similar way as \( \mathbf{P}_{F_1} \). Then the DEIM approximation of the nonlinear functions in (48)-(52) is of
the form:
\[
\begin{align*}
\hat{F}_1 &\approx U^{F_1}(P_{F_1}^T U^{F_1})^{-1}\hat{F}_1^m, \\
\bar{N}_1 &\approx U^{N_1}(P_{N_1}^T U^{N_1})^{-1}\bar{N}_1^m, \\
\hat{f} &\approx U/(P_f^T U)^{-1}\hat{f}_m,
\end{align*}
\]

and the nonlinear terms for the POD reduced system can then be approximated as
\[
\begin{align*}
U^T\hat{F}_1(\hat{\psi}, \hat{c}) &\approx U^T U^{F_1}(U_{\hat{\psi}}^m)^{-1}\hat{F}_1^m, \\
Q^T\hat{F}_2(\hat{\psi}, \hat{T}) &\approx Q^T U^{F_2}(U_{\hat{\psi}}^m)^{-1}\hat{F}_2^m, \\
V^T\bar{N}_1(\hat{\psi}, \hat{c}) &\approx V^T U^{N_1}(U_{\hat{\psi}}^m)^{-1}\bar{N}_1^m,
\end{align*}
\]

where the nonlinear functions \(\hat{F}_1^m, \hat{F}_2^m, \bar{N}_1^m, \bar{N}_2^m, R^k \times R^k \rightarrow R^m\) are defined as:
\[
\begin{align*}
\hat{F}_1^m(\hat{\psi}, \hat{c}) &= P_{F_1}^T \hat{F}_1(\hat{\psi}, \hat{c}), \\
\hat{F}_2^m(\hat{\psi}, \hat{T}) &= P_{F_2}^T \hat{F}_2(\hat{\psi}, \hat{T}), \\
\bar{N}_1^m(\hat{\psi}, \hat{c}) &= P_{N_1}^T \bar{N}_1(\hat{\psi}, \hat{c}), \\
\bar{N}_2^m(\hat{\psi}, \hat{T}) &= P_{N_2}^T \bar{N}_2(\hat{\psi}, \hat{T}), \\
\hat{f}_m(\hat{c}) &= P_f^T f(U\hat{c}) = f(P_f^T U\hat{c}).
\end{align*}
\]

The last equation of (59) is valid because \(f_j(c) = f(c_j)\) is a pointwise function. The explicit form of \(\hat{F}_1^m = \hat{F}_1^m(\hat{\psi}, \hat{c})\) is given by
\[
P_{F_1}^T \hat{F}_1(\hat{\psi}, \hat{c}) = (P_{F_1}^T A_w W \hat{\psi}) + (P_{F_1}^T A_x U \hat{c} + P_{F_1}^T b_x) - (P_{F_1}^T A_x W \hat{\psi}) \ast (P_{F_1}^T A_y U \hat{c}) (60)
\]
and similarly for \(\hat{F}_2^m, \bar{N}_1^m, \bar{N}_2^m\). Note that, from (58), the \(k\)-by-\(m\) matrices
\[
\begin{align*}
E_1 &= U^T U^{F_1}(U_{\hat{\psi}}^m)^{-1}, \\
E_2 &= U^T U^{F_2}(U_{\hat{\psi}}^m)^{-1}, \\
E_3 &= Q^T U^{F_2}(U_{\hat{\psi}}^m)^{-1}, \\
E_4 &= Q^T U^{F_2}(U_{\hat{\psi}}^m)^{-1}, \\
E_5 &= V^T U^{N_1}(U_{\hat{\psi}}^m)^{-1}, \\
E_6 &= V^T U^{N_2}(U_{\hat{\psi}}^m)^{-1}
\end{align*}
\]

can be precomputed and reused at each time step. Also, each of the \(m\)-by-\(k\) coefficient matrices in (60) grouped by the curly brackets are precomputed so that the computational complexity of each nonlinear function is independent of the dimension \(n\) of the original full-order system. Finally, the POD-DEIM reduced system is of the form:
\[
\begin{align*}
\frac{d\hat{c}(t)}{dt} &= -E_1 \hat{F}_1^m(\hat{\psi}(t), \hat{c}(t)) + [\hat{A}_1 \hat{c}(t) + \hat{b}_1] + Da E_2 f(P_f^T \hat{U}\hat{c}(t)) \\
\frac{d\hat{T}(t)}{dt} &= -E_2 \hat{F}_2^m(\hat{\psi}(t), \hat{T}(t)) + \text{Le}[\hat{A}_2 \hat{T}(t) + \hat{b}_2] + \text{sgn}(\hat{\phi}) Da E_4 f(P_f^T \hat{U}\hat{c}(t)) \\
\hat{\omega}(t) &= -R \left[ E_5 \bar{N}_1^m(\hat{\psi}(t), \hat{c}(t)) + \hat{A}_3 \hat{c}(t) \right] + R_T \left[ E_6 \bar{N}_2^m(\hat{\psi}(t), \hat{T}(t)) + \hat{A}_4 \hat{T}(t) \right] \\
\hat{A}_5 \hat{\psi}(t) &= -\hat{\omega}(t).
\end{align*}
\]
Before presenting the numerical results from POD-DEIM reduced system constructed in this section, we shall discuss an important issue on the computation of POD basis which is a crucial ingredient for constructing the subspace of the approximate low-rank solution as well as the DEIM approximation. The following section presents an efficient approach for computing POD of snapshot solutions from high dimensional full-order system.
5. Computation of POD Basis

This section discusses a practical implementation for computing POD basis of sampled snapshots. We present here an algorithm that is particularly efficient when the spatial dimension $n$ of the FD discretization is much larger than the number $n_s$ of the sampled snapshots (this situation often arises in systems with 2-D or 3-D spatial domain). This algorithm automatically detects and eliminates the redundant snapshots as well as prevents the loss of orthogonality of the POD basis due to the numerical instability.

Let $Y$ be an $n$-by-$n_s$ matrix whose columns contain $n_s$ sampled solution snapshots. Assume here that $n \gg n_s$. As noted, the POD basis can be obtained from the left singular vectors of the SVD of the snapshot matrix $Y$. Since $n \gg n_s$, an efficient procedure for computing the SVD of $Y$, as suggested in [17], is to first perform the QR factorization of $Y$, and then compute the SVD of the $n_s$-by-$n_s$ matrix $R$. In particular, let $Y = QR$ be the QR decomposition of $Y$ where $Q \in \mathbb{R}^{n \times n_s}$ is a matrix with orthonormal columns and $R \in \mathbb{R}^{n_s \times n_s}$ is an upper triangular matrix and let $R = U \Sigma V^T$ be the SVD of $R$, then the SVD of $Y$ is finally given by $Y = (QU) \Sigma V^T$ and the POD basis can be obtained from the columns of $QU$.

We apply a Gram-Schmidt process with reorthogonalization to construct the QR factorization of $Y$ (which is more efficient than the Householder orthogonalization) to allow the elimination of the redundant snapshot vectors during the computation process as well as to employ the available reorthogonalization algorithm [18] for preserving the numerical stability as explained next. Each iteration of the Gram-Schmidt process is essentially an algorithm for constructing an orthogonal vector from each newly appending column vector and therefore it can be used to detect the linearly dependence of the snapshots. That is, in each iteration of Algorithm 5.1, the appending snapshot vector $y_\ell$ would be discarded once it is detected to be in the space spanned by the previously constructed orthonormal columns of $Q$.

The reorthogonalization is applied to ensure the orthogonality of the columns in $Q$ computed from the Gram-Schmidt process which in turn affect the orthogonality of the POD basis. In the finite machine precision, when the column rank of the snapshot matrix $Y$ is nearly deficient, it is well-known [18] that the columns of $Q$ computed from the Gram-Schmidt process can deviated arbitrarily far from orthogonality due to the presence of rounding error. Hence, this numerical instability can in fact often occur in practice when the snapshot vectors are sampled randomly or uniformly without using any prior information which makes it possible that some of the snapshot vectors are close to be linearly dependent. The reorthogonalization algorithm [18] based on a simple modification of Gram-Schmidt process is proved to be stable and efficient method to prevent this loss of orthogonality (see [18] for details). In principle, the reorthogonalization can be applied as many times as we want, but it was shown in [19] that one reorthogonalization is enough for maintaining the orthogonality (to machine precision level).

Algorithm 5.1 summarizes the procedure for computing the POD basis from snapshots $\{y_\ell\}_{\ell=1}^{n_s}$ as discussed above.
Algorithm 5.1: POD (QR-SVD)

INPUT: \( \{y_\ell\}_{\ell=1}^{n_s} \subset \mathbb{R}^n \) (not necessary linearly independent)
OUTPUT: \( U=\) POD basis; \( S=\) Singular values; \( J_{\text{out}}=\) redundant snapshots’ indices:

1: \( \rho = \|y_1\|_2 \), \( j_0 = 1 \), \( J_{\text{out}} = [ ] \)
2: while \( \rho \neq 0 \) do
3: \( J_{\text{out}} \leftarrow [J_{\text{out}} \ j_0] \), \( j_0 = j_0 + 1 \), \( \rho = \|y_{j_0}\|_2 \)
4: end while
5: \( q_1 = y_{j_0}/\rho \)
6: \( Q = [q_1] \)
7: \( j_s = j_0 + 1 \)
8: for \( \ell = j_0 + 1 \) to \( n \) do
9: \( r = Q^T y_\ell \)
10: \( q = y_\ell - Qr \)
11: \{Reorthogonalization\}
12: \( c = Q^T q \)
13: \( q = q - Qc \)
14: \( r = r + c \)
15: \( \rho = \|q\|_2 \)
16: if \( \rho \geq \epsilon \|r\|_2 \) then
17: \( Q \leftarrow [Q \ q] \)
18: \( R \leftarrow [R \ r; zeros(1, j_s - 1) \ \rho] \)
19: \( j_s \leftarrow j_s + 1 \)
20: else
21: \( J_{\text{out}} \leftarrow [J_{\text{out}} \ \ell] \) \{Linearly dependence found\}
22: end if
23: end for
24: \( [U \ S \ V] = \text{svd}(R) \)
25: \( U \leftarrow UQ \)

6. Numerical Results of POD-DEIM Reduced System

We shall present three numerical experiments. The first one considers the POD-DEIM reduced system for a set of fixed parameters. The second one considers the reduced system that can be used for various values of parameter Péclet number in a certain range. The last one considers miscible flow with viscous fingering induced by a simple chemical reaction. For all these cases, in addition to the initial condition for \( c \) given in (24), the random noise between 0 and 1 is added at each grid point on the interface to trigger the instability on reasonable computing time as done in many literatures such as \([8, 9, 13]\).

6.1. Fixed Parameters

The system (10)-(13) is solved numerically using finite difference scheme from [13] as mentioned earlier. This section considers the isothermal case (constant temperature: \( R_T = 0 \)). The parameters used here are \( R_c = 3; R_T = 0; a = 2; Pe = 250; Le = 1; Da = 0.01; d = 0.1 \). The number of spatial grid points is 150 on the \( x \)-axis and 100 on the \( y \)-axis. The dimension of the full-order system is then 15000. The singular values of 250 solution snapshots and nonlinear snapshots are shown in Fig-
ure 1. The plots of corresponding POD bases are shown in Appendix D. In Figure 2, the solutions for concentration from POD-DEIM reduced system (62)-(65), with POD and DEIM of dimension 40, are shown with the corresponding ones from the full-order system as well as the corresponding absolute error at the grid points. It shows that POD-DEIM reduces more than 300 times in dimension and reduces the computational time by factor of $O(10^3)$ with $O(10^{-3})$ error as shown in Table 1. From Figure 3, the plot of the CPU time used in computing POD reduced system clearly reflects the dependency on the dimension of the original full-order system. Figure 3 and Table 1 show a significant improvement in computational time of the POD-DEIM reduced system from both POD reduced system and full-order system.

6.2. Varying Péclet number: $Pe \in [110, 120]$

Consider the same numerical setup as for the previous case in Section 6.1 except that we are now interested in the parameter $Pe$ in the interval $[110, 120]$. The POD basis used for approximating the solution space is constructed from 398 snapshots.
Table 1. Average relative error (2-norm) of the solution for the concentration $c$ and CPU time of full-order system, POD reduced system, and POD-DEIM reduced system with Pe = 250 (fixed parameters).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Avg Rel Error of $c$</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full 15000 (FD)</td>
<td>-</td>
<td>$2.138 \times 10^3$</td>
</tr>
<tr>
<td>POD20</td>
<td>$5.597 \times 10^{-3}$</td>
<td>$1.206 \times 10^2$</td>
</tr>
<tr>
<td>POD20/DEIM20</td>
<td>$2.041 \times 10^{-2}$</td>
<td>0.923</td>
</tr>
<tr>
<td>POD40</td>
<td>$4.066 \times 10^{-4}$</td>
<td>$2.442 \times 10^2$</td>
</tr>
<tr>
<td>POD40/DEIM40</td>
<td>$2.045 \times 10^{-3}$</td>
<td>1.275</td>
</tr>
</tbody>
</table>

Figure 3. (a) Average relative errors of $y = [c; \psi; \omega]$ from the POD-DEIM reduced system compared with the ones from POD reduced system. (b) CPU time of the full system, POD reduced system, and POD-DEIM reduced system.

taken from two full-order FD systems corresponding to Pe = 110 and 120 (199 snapshots are uniformly selected in time $t \in [0, 200]$ from each system). The vector values of the nonlinear function evaluated at these snapshots are used to construct the POD basis for the DEIM approximation of each nonlinear term. The resulting POD-DEIM reduced system can be used to approximate the systems with arbitrarily parameter Pe in the interval [110, 120]. To demonstrate the effectiveness of this reduced system, we consider the solutions of the VF system with parameter Pe = 115 which was not used in constructing the POD bases of this POD-DEIM reduced system. With Pe = 115, Figure 4 illustrates the solutions for concentration from the full-order system of dimension 15000 compared with the ones from the POD-DEIM reduced system with POD of dimension 30 and DEIM of dimension 50 as well as the corresponding absolute error at the grid points. This shows that POD-DEIM can reduce 300 times in dimension with average relative error $O(10^{-3})$ as also shown in Table 2. Notice from Table 2 that, while the POD system of dimension 30 reduces roughly by factor of 5 in computational time of the full-order system, the POD-DEIM system (POD=30, DEIM=50) reduces roughly by factor of 700. Table 2 also illustrates that for the fixed dimension 30 of POD, as the dimension of DEIM increases from 30 to 50, the POD-DEIM reduced system becomes more accurate (i.e., the average relative error decreases).

6.3. Miscible Viscous Fingering Induced by Chemical Reaction

This section considers a system from [9] that describes miscible flow with viscous fingering induced by a simple chemical reaction

$$A + B \rightarrow C,$$
Figure 4. Concentration plots of the injected fluid at time $t = 50, 100, 200$ from the full-order system of dimension 15000 and from the POD-DEIM reduced system with POD and DEIM having dimensions 30 and 50, with the corresponding absolute error at the grid points (Péclet number $Pe = 115$).

Table 2. Average relative error (2-norm) of the concentration $c$ and CPU time of full-order system, POD reduced system, and POD-DEIM reduced system with Péclet number $Pe = 115$ which is arbitrary chosen from the interval $[110, 120]$.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Avg Rel Error of $c$</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full 15000 (FD)</td>
<td>-</td>
<td>$7.384 \times 10^2$</td>
</tr>
<tr>
<td>POD30</td>
<td>$5.907 \times 10^{-3}$</td>
<td>$1.338 \times 10^2$</td>
</tr>
<tr>
<td>POD30/DEIM30</td>
<td>$3.133 \times 10^{-2}$</td>
<td>$0.843$</td>
</tr>
<tr>
<td>POD30/DEIM50</td>
<td>$7.905 \times 10^{-3}$</td>
<td>$0.909$</td>
</tr>
<tr>
<td>POD50</td>
<td>$5.910 \times 10^{-3}$</td>
<td>$2.434 \times 10^2$</td>
</tr>
<tr>
<td>POD50/DEIM50</td>
<td>$8.579 \times 10^{-3}$</td>
<td>$1.150$</td>
</tr>
</tbody>
</table>

which occurs at the interface of the reactants $A$ and $B$ producing a product $C$. The system of governing equations is in a similar form to the one presented in Section 2 and given by the convection-diffusion-reaction equations as:

$$\nabla \cdot \mathbf{u} = 0$$  \hspace{1cm} (66)

$$\nabla P = -\frac{\mu}{K} \mathbf{u}$$  \hspace{1cm} (67)

$$\frac{\partial a}{\partial t} + \mathbf{u} \cdot \nabla a = D_A \nabla^2 a - \kappa ab$$  \hspace{1cm} (68)

$$\frac{\partial b}{\partial t} + \mathbf{u} \cdot \nabla b = D_B \nabla^2 b - \kappa ab$$  \hspace{1cm} (69)

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = D_C \nabla^2 c + \kappa ab,$$  \hspace{1cm} (70)
where \(a, b, c\) are the concentrations of the two reactants \(A\) and \(B\) and of the product \(C\); \(\kappa\) is the kinetic constant; \(P\) is the pressure; \(u = [u, w] \in \mathbb{R}^2\) is the velocity field; \(D_A, D_B, D_C\) are constant diffusion coefficients of \(A, B, C\); \(K\) is the constant permeability; \(\mu = \mu(c)\) is the viscosity as a function of the local concentration \(c = c(x, y, t)\) of the product \(C\) only (when \(c = 0\), \(\mu(0) = \mu_0\) is the viscosity of the reactant). Assume that the viscosity varies exponentially with the concentration \(c\) as:

\[
\mu(c) = \mu_0 e^{R(c/c_0)},
\]

where \(R\) is the log-mobility ratio defined as \(R = \ln \left[ \frac{\mu(c_0)}{\mu_0} \right] \). If \(R > 0\), then the chemical reaction produces a more viscous product \(C\) at the interface and the less viscous reactant pushes the more viscous product as shown later in Figure 5. The same technique as presented in Section 2 is used for transforming (66)-(70) to the nondimensionalized system and then to the streamfunction formulation as:

\[
\nabla^2 \psi = R(\psi_x c_x + \psi_y c_y + c_y)
\]

\[
\frac{\partial a}{\partial t} + \psi_y a_x - \psi_x a_y = \delta_A \nabla^2 a - D_a a b,
\]

\[
\frac{\partial b}{\partial t} + \psi_y b_x - \psi_x b_y = \delta_B \nabla^2 b - D_a a b,
\]

\[
\frac{\partial c}{\partial t} + \psi_y c_x - \psi_x c_y = \nabla^2 c + D_a a b,
\]

which are finally solved by the same discretization scheme as the previous cases. Note that the parameters that arise during the nondimensionalization process are the ratios of the diffusion coefficients of \(A\) and \(B\): \(\delta_A = D_A/D_C\), \(\delta_B = D_B/D_C\), and the Damköhler number \(Da = D_C \kappa a_0/U^2\), where \(a_0\) is the initial concentration of reactant \(A\) and \(U\) is the uniform injecting velocity.

The numerical results presented here are generated from the nondimensionalized equations (71)-(74) with parameters: \(R = 3, \ Pe = 250, \ Le = 1, \ Da = 1, \ d = 0.1, \ \delta_A = 1, \ \delta_B = 5, \) and with aspect ratio of 3. The periodic boundary conditions are used in both \(x\) and \(y\) coordinates. Initially, the reactant \(B\) is sandwiched between the reactant \(A\) (see Figure 5 for the flow domain). This initial setting allows us to observe the behavior of viscous fingering in both cases when \(A\) is injected to \(B\) and when \(B\) is injected to \(A\) at the same time. Figure 5 illustrates the concentrations of reactants \(A\), \(B\) and the product \(C\) in a 2-D homogeneous porous medium at time \(t = 100, 250, 500\). Similar to the previous two numerical cases, it shows that the POD-DEIM reduced model with POD and DEIM of dimension 30 and 40, can accurately capture the VF dynamics of the full-order system having dimension 15000 with substantially less CPU time, i.e. \(O(1000)\) reduction, as shown in Table 3.

Note that this type of nonlinear system is influenced by various parameters (e.g. \(Pe, \ \delta_A, \ \delta_B, \ Da\)) and the parametric study therefore becomes an important tool and a common method for analyzing the dynamics of this system as done in [9]. Hence, POD-DEIM approach is a promising technique that can be used to improve the efficiency of the simulation for this parametric study.
Figure 5. Concentration plots in the flow domain of reactants $A$, $B$ and the product $C$ from the reaction $A + B \rightarrow C$ at time $t = 100, 250, 500$ from the full-order system of dimension 15000 and from the POD-DEIM reduced system with POD and DEIM having dimensions 30 and 40, with the corresponding absolute error at the grid points (fixed parameters).
Table 3. Average relative error (2-norm) of the solution for the concentrations \(a, b, c\) of the reactants \(A, B,\) and the product \(C\) and CPU time of full-order system, POD reduced system, and POD-DEIM reduced system (fixed parameters).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Avg Rel Error of concentrations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full 15000 (FD)</td>
<td>-</td>
<td>1.699 \times 10^3</td>
</tr>
<tr>
<td>POD10</td>
<td>4.561 \times 10^{-3}</td>
<td>1.757 \times 10^2</td>
</tr>
<tr>
<td>POD10/DEIM10</td>
<td>8.255 \times 10^{-3}</td>
<td>1.612</td>
</tr>
<tr>
<td>POD20</td>
<td>9.131 \times 10^{-4}</td>
<td>3.057 \times 10^2</td>
</tr>
<tr>
<td>POD20/DEIM20</td>
<td>3.267 \times 10^{-3}</td>
<td>1.970</td>
</tr>
<tr>
<td>POD30</td>
<td>4.006 \times 10^{-4}</td>
<td>4.435 \times 10^2</td>
</tr>
<tr>
<td>POD30/DEIM40</td>
<td>8.382 \times 10^{-4}</td>
<td>2.567</td>
</tr>
<tr>
<td>POD40</td>
<td>3.162 \times 10^{-4}</td>
<td>6.325 \times 10^2</td>
</tr>
<tr>
<td>POD40/DEIM40</td>
<td>4.867 \times 10^{-4}</td>
<td>2.791</td>
</tr>
</tbody>
</table>

7. Conclusions and Future Work

The model reduction technique combining POD with DEIM has been shown to be efficient for capturing the dynamics in the VF simulation with substantial reduction in dimension and computational time. The failure to decrease complexity with the standard POD technique was clearly demonstrated by the comparative computational times shown in, e.g., the plot of CPU time in Figure 3. DEIM was shown to be very effective in overcoming the deficiencies of POD with respect to general nonlinearities in VF simulation. Moreover, the preliminary numerical results in the previous section suggest the extension of the POD-DEIM approach to the VF parametric study. In particular, by using the analogous setup presented in this paper, the POD-DEIM technique can be applied to speed up the VF simulations for predicting and analyzing the flow displacement and other VF mechanisms, as the parameter of interest is varied in a specified range. Another possible extension is to incorporate the POD-DEIM approach with other higher-order FD schemes to improve the overall accuracy. For these extensions, the quality of the sampled snapshots and the choice of the numerical scheme are the unavoidable issues that would affect the efficiency of the POD-DEIM approximation. However, these two issues still remain as the challenging research topics and will be left for future work.

Acknowledgement

We wish to thank Prof. Beatrice Riviere for suggesting this miscible flow problem, and for giving helpful advice and comments throughout the course of this work. This work was supported in part by AFOSR grant FA9550-09-1-0225 and by NSF grant DMS-0914021.

References

Appendices

Appendix A. Proper Orthogonal Decomposition (POD)

POD is a method for constructing a low-dimensional approximation representation of a subspace in Hilbert space. It is essentially the same as the singular value decomposition (SVD) in a finite dimensional space or in Euclidean space. It efficiently extracts the basis elements that contain characteristics of the space of expected solutions of the PDE. The POD basis in Euclidean space is defined formally as follows.

Given a set of snapshots \( \{ y^1, \ldots, y^{n_s} \} \subset \mathbb{R}^n \) (recall snapshots are samples of trajectories), let \( \mathcal{Y} = \text{span}\{ y^1, \ldots, y^{n_s} \} \subset \mathbb{R}^n \) and \( r = \text{dim}(\mathcal{Y}) \). A POD basis of dimension \( k < r \) is a set of orthonormal vectors \( \{ \phi_i \}_{i=1}^k \subset \mathbb{R}^n \) whose linear span best approximates the space \( \mathcal{Y} \). The basis set \( \{ \phi_i \}_{i=1}^k \) solves the minimization problem:

\[
\min_{\{\phi_i\}_{i=1}^k} \sum_{j=1}^{n_s} \left\| y_j - \sum_{i=1}^k (y_j^T \phi_i) \phi_i \right\|^2_2
\]

\[
\phi_i^T \phi_j = \delta_{ij} = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \quad i, j = 1, \ldots, k.
\]

It is well known that the solution to (A1) is provided by the set of the left singular vectors of the snapshot matrix \( Y = [y^1, \ldots, y^{n_s}] \in \mathbb{R}^{n \times n_s} \). In particular, suppose that the singular value decomposition (SVD) of \( Y \) is

\[
Y = V \Sigma W^T,
\]
where \( \mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r] \in \mathbb{R}^{n \times r} \) and \( \mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_r] \in \mathbb{R}^{n \times r} \) are orthogonal and \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r} \) with \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 \). The rank of \( \mathbf{Y} \) is \( r \leq \min(n, n_s) \). Then the POD basis or the optimal solution of (A1) is \( \{ \mathbf{v}_1 \}_{j=1}^k \). The minimum 2-norm error from approximating the snapshots using the POD basis is then given by

\[
\sum_{j=1}^{n_s} \| \mathbf{y}_i - \sum_{i=1}^{k} (\mathbf{y}_j^T \mathbf{v}_i) \mathbf{v}_i \|_2^2 = \sum_{i=k+1}^{r} \sigma_i^2.
\]

We refer to [3] for more details on the POD basis in general Hilbert space.

The choice of the snapshot ensemble is a crucial factor in constructing a POD basis. This choice can greatly affect the approximation of the original solution space, but it is a separate issue and will not be discussed here.

### Appendix B. Details for Constructing a Finite Difference (FD) Discretized System

This section considers the FD discretization of the system (10)-(13). Let \( 0 = x_0 < x_1 < \cdots < x_{n_x-1} = aPe \) and \( 0 = y_0 < y_1 < \cdots < y_{n_y-1} = Pe \) be equally spaced points on \( x \)-axis and \( y \)-axis for generating the grid points on the domain \( \Omega \) with \( dx = aPe/(n_{nx}+1) \) and \( dy = Pe/(n_{ny}+1) \). Let \( c_{ij}^\ell, T_{ij}^\ell, \psi_{ij}^\ell, \) and \( \omega_{ij}^\ell \) denote the finite difference approximations of \( c(x_i, y_j, t), T(x_i, y_j, t), \psi(x_i, y_j, t), \) and \( \omega(x_i, y_j, t) \). The system (10)-(13) can be written as: for \( i = 1, \ldots, n_x, j = 1, \ldots, n_y, \) and \( \ell = 0, \ldots, n_t \)

\[
\frac{1}{dt} (c_{ij}^{\ell+1} - c_{ij}^\ell) = - \left( \nabla_y \psi_{ij}^\ell \nabla_x c_{ij}^\ell - \nabla_x \psi_{ij}^\ell \nabla_y c_{ij}^\ell \right) + \nabla^2 \alpha_{ij}^\ell + \text{Da}(c_{ij}^\ell) \label{eq:B1}
\]

\[
\frac{1}{dt} (T_{ij}^{\ell+1} - T_{ij}^\ell) = - \left( \nabla_y \psi_{ij}^\ell \nabla_x T_{ij}^\ell - \nabla_x \psi_{ij}^\ell \nabla_y T_{ij}^\ell \right) + \text{Le}(\nabla \omega_{ij}^\ell) + \text{sgn}(\phi) \text{Da}(c_{ij}^\ell) \label{eq:B2}
\]

\[
\omega_{ij}^\ell = - R_c \left[ \nabla_x \psi_{ij}^\ell \nabla_x c_{ij}^\ell + \nabla_y \psi_{ij}^\ell \nabla_y c_{ij}^\ell \right] + \text{RT} \left[ \nabla_x \psi_{ij}^\ell \nabla_x T_{ij}^\ell + \nabla_y \psi_{ij}^\ell \nabla_y T_{ij}^\ell \right] \label{eq:B3}
\]

\[
\nabla^2 \psi_{ij}^\ell = - \omega_{ij}^\ell, \label{eq:B4}
\]

where

\[
\nabla_x \psi_{ij}^\ell = \frac{\psi_{i+1,j}^\ell - \psi_{i-1,j}^\ell}{2dx}, \quad \nabla_x c_{ij}^\ell = \frac{c_{i+1,j}^\ell - c_{i-1,j}^\ell}{2dx}, \quad \nabla_x T_{ij}^\ell = \frac{T_{i+1,j}^\ell - T_{i-1,j}^\ell}{2dx}. \label{eq:B5}
\]

For \( i = 1 \) and \( i = n_x \), the boundary conditions \( \psi_{0,j}^\ell = 0 \) and \( \psi_{n_x+1,j}^\ell = 0 \) \( c_{0,j}^\ell = 1 \) and \( c_{n_x+1,j}^\ell = 0 \) \( T_{0,j}^\ell = 1 \) and \( T_{n_x+1,j}^\ell = 0 \) for exothermic case \( (T_{0,j}^\ell = 0 \) and \( T_{n_x+1,j}^\ell = 1 \) for endothermic case) are used, so that

\[
\nabla_x \psi_{1,j}^\ell = \frac{\psi_{2,j}^\ell - \psi_{0,j}^\ell}{2dx}, \quad \nabla_x \psi_{n_x,j}^\ell = \frac{\psi_{n_x+1,j}^\ell - \psi_{n_x-1,j}^\ell}{2dx}, \label{eq:B6}
\]

\[
\nabla_x c_{1,j}^\ell = \frac{c_{2,j}^\ell - c_{0,j}^\ell}{2dx}, \quad \nabla_x c_{n_x,j}^\ell = \frac{c_{n_x+1,j}^\ell - c_{n_x-1,j}^\ell}{2dx}, \label{eq:B7}
\]

\[
\nabla_x T_{1,j}^\ell = \frac{T_{2,j}^\ell - T_{0,j}^\ell}{2dx}, \quad \nabla_x T_{n_x,j}^\ell = - \frac{T_{n_x+1,j}^\ell - T_{n_x-1,j}^\ell}{2dx}, \label{eq:B8}
\]

\[
\nabla_y \psi_{ij}^\ell = \frac{\psi_{i,j+1}^\ell - \psi_{i,j-1}^\ell}{2dy}, \quad \nabla_y c_{ij}^\ell = \frac{c_{i+1,j}^\ell - c_{i-1,j}^\ell}{2dy}, \quad \nabla_y T_{ij}^\ell = \frac{T_{i+1,j}^\ell - T_{i-1,j}^\ell}{2dy}. \label{eq:B9}
\]
For \( i = 1 \) and \( i = n_y \), the periodic boundary conditions for \( \psi, c, T \) are used. That is, for all \( i = 1, \ldots, n_x \), \( \psi^\ell_{i,0} = \psi^\ell_{i,n_y+1} \) and \( \frac{\psi^\ell_{i,1} - \psi^\ell_{i,0}}{dy} = \frac{\psi^\ell_{i,n_y+1} - \psi^\ell_{i,n_y}}{dy} \). I.e.,

\[
\psi^\ell_{i,0} = \psi^\ell_{i,n_y+1} = \frac{\psi^\ell_{i,1} + \psi^\ell_{i,n_y}}{2},
\]

(B10)

Hence

\[
\nabla_y \psi^\ell_{i,1} = \frac{1}{2dy} \left( \psi^\ell_{i,2} - \psi^\ell_{i,0} \right) = \frac{1}{2dy} \left( -\frac{1}{2} \psi^\ell_{i,1} + \psi^\ell_{i,2} - \frac{1}{2} \psi^\ell_{i,n_y} \right),
\]

(B11)

\[
\nabla_y \psi^\ell_{i,n_y} = \frac{1}{2dy} \left( \psi^\ell_{i,1} + \psi^\ell_{i,n_y} - \psi^\ell_{i,n_y-1} \right) = \frac{1}{2dy} \left( \frac{1}{2} \psi^\ell_{i,1} - \psi^\ell_{i,n_y-1} + \frac{1}{2} \psi^\ell_{i,n_y} \right),
\]

(B12)

and similarly for \( \nabla_y c^\ell_{i,1}, \nabla_y c^\ell_{i,n_y}, \nabla_y T^\ell_{i,1}, \) and \( \nabla_y T^\ell_{i,n_y} \).

\[
\nabla^2 \psi^\ell_{i,j} = \frac{1}{dx^2} \left[ \psi^\ell_{i-1,j} - 2 \psi^\ell_{i,j} + \psi^\ell_{i+1,j} \right] + \frac{1}{dy^2} \left[ \psi^\ell_{i,j-1} - 2 \psi^\ell_{i,j} + \psi^\ell_{i,j+1} \right]
\]

(B13)

\[
= \frac{1}{dx^2} \psi^\ell_{i-1,j} + \frac{1}{dx^2} \psi^\ell_{i+1,j} + \frac{1}{dy^2} \psi^\ell_{i,j-1} + \frac{1}{dy^2} \psi^\ell_{i,j+1},
\]

(B14)

and similarly for \( \nabla^2 c^\ell_{i,j} \) and \( \nabla^2 T^\ell_{i,j} \). The boundary conditions are needed to be used for \( i = 1, i = n_x, j = 1, \) and \( j = n_y \).

For \( i = 1 \) and \( i = n_x \), by using \( \psi^\ell_{0,j} = 0 \) and \( \psi^\ell_{n_x+1,j} = 0 \), for \( j = 2, \ldots, n_y - 1 \),

\[
\nabla^2 \psi^\ell_{1,j} = \frac{1}{dx^2} \psi^\ell_{2,j} + \frac{1}{dy^2} \psi^\ell_{1,j-1} + \frac{1}{dy^2} \psi^\ell_{1,j+1},
\]

(B15)

\[
\nabla^2 \psi^\ell_{n_x,j} = \frac{1}{dx^2} \psi^\ell_{n_x-1,j} + \frac{1}{dy^2} \psi^\ell_{n_x,j-1} + \frac{1}{dy^2} \psi^\ell_{n_x,j+1},
\]

(B16)

and by using \( c^\ell_{0,j} = T^\ell_{0,j} = 1 \) and \( c^\ell_{n_x+1,j} = T^\ell_{n_x+1,j} = 0 \)

\[
\nabla^2 c^\ell_{1,j} = \frac{1}{dx^2} + \frac{1}{dx^2} c^\ell_{2,j} + \frac{1}{dy^2} \psi^\ell_{1,j-1} + \frac{1}{dy^2} \psi^\ell_{1,j+1},
\]

(B17)

\[
\nabla^2 c^\ell_{n_x,j} = \frac{1}{dx^2} c^\ell_{n_x-1,j} + \frac{1}{dx^2} c^\ell_{n_x,j-1} + \frac{1}{dy^2} c^\ell_{n_x,j+1},
\]

(B18)

and similarly for \( \nabla^2 T^\ell_{1,j} \).

For \( j = 1 \) and \( j = n_y \), the periodic boundary conditions are used by applying (B10) to (B14):

\[
\nabla^2 \psi^\ell_{1,1} = \frac{1}{dx^2} \psi^\ell_{1,1} + \frac{1}{dx^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{2,1} \]

(B19)

\[
= \frac{1}{dx^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{1,1} + \frac{1}{dy^2} \psi^\ell_{2,1} \]

(B20)

\[
\nabla^2 \psi^\ell_{1,n_y} = \frac{1}{dx^2} \psi^\ell_{1,n_y} + \frac{1}{dx^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} \]

(B21)

\[
= \frac{1}{dx^2} \psi^\ell_{1,n_y} + \frac{1}{dx^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} + \frac{1}{dy^2} \psi^\ell_{1,n_y} \]

(B22)

and similarly for \( \nabla^2 c^\ell_{1,1} \) and \( \nabla^2 T^\ell_{1,n_y} \). In the case when (B19)-(B21) with \( i = 1 \) or \( i = n_x \), the boundary conditions \( \psi^\ell_{0,1} = 0, \psi^\ell_{n_x+1,1} = 0, c^\ell_{0,1} = 1, c^\ell_{n_x+1,1} = 0, T^\ell_{0,1} = 1, T^\ell_{n_x+1,1} = 0 \) are used in (B19)-(B21).
Appendix C. Details of the scheme for time integration [13]

Let $c^\ell_{ij}$, $T^\ell_{ij}$, $\psi^\ell_{ij}$, and $\omega^\ell_{ij}$ denote the finite difference approximations of $c(x_i, y_j, t^\ell)$, $T(x_i, y_j, t^\ell)$, $\psi(x_i, y_j, t^\ell)$, and $\omega(x_i, y_j, t^\ell)$, for $n = n_x n_y$, define

$$c^\ell = [c^\ell_{11}, c^\ell_{12}, \ldots, c^\ell_{1n_y}, c^\ell_{21}, c^\ell_{22}, \ldots, c^\ell_{2n_y}, \ldots, c^\ell_{n_x1}, c^\ell_{n_x2}, \ldots, c^\ell_{n_xn_y}]^T \in \mathbb{R}^{n_y} \quad (C1)$$

$$T^\ell = [T^\ell_{11}, T^\ell_{12}, \ldots, T^\ell_{1n_y}, T^\ell_{21}, T^\ell_{22}, \ldots, T^\ell_{2n_y}, \ldots, T^\ell_{n_x1}, T^\ell_{n_x2}, \ldots, T^\ell_{n_xn_y}]^T \in \mathbb{R}^{n_y} \quad (C2)$$

$$\psi^\ell = [\psi^\ell_{11}, \psi^\ell_{12}, \ldots, \psi^\ell_{1n_y}, \psi^\ell_{21}, \psi^\ell_{22}, \ldots, \psi^\ell_{2n_y}, \ldots, \psi^\ell_{n_x1}, \psi^\ell_{n_x2}, \ldots, \psi^\ell_{n_xn_y}]^T \in \mathbb{R}^{n_y} \quad (C3)$$

$$\omega^\ell = [\omega^\ell_{11}, \omega^\ell_{21}, \ldots, \omega^\ell_{n_x1}, \omega^\ell_{12}, \omega^\ell_{22}, \ldots, \omega^\ell_{n_x2}, \ldots, \omega^\ell_{1n_y}, \omega^\ell_{2n_y}, \ldots, \omega^\ell_{n_xn_y}]^T \in \mathbb{R}^{n_y} \quad (C4)$$

The system (B1)-(B4) can be written in matrix form as: $\ell = 0, \ldots, n_t$

$$\frac{1}{dt}(c^{\ell+1} - c^\ell) = -F(\psi^\ell, c^\ell) + [Ac^\ell + ] + Daf(c^\ell) \quad (C5)$$

$$\frac{1}{dt}(T^{\ell+1} - T^\ell) = -F(\psi^\ell, T^\ell) + Le[AT^\ell + ] + sgn(\phi)Daf(c^\ell) \quad (C6)$$

$$\omega^\ell = -Re \left[ N(\psi^\ell, c^\ell) + A_y c^\ell \right] + RT \left[ N(\psi^\ell, T^\ell) + A_y T^\ell \right] \quad (C7)$$

$$A \psi^\ell = -\omega^\ell, \quad (C8)$$

where

$$F(\psi^\ell, c^\ell) = (A_y \psi^\ell) \ast (A_x c^\ell + b_x) - (A_x \psi^\ell) \ast (A_y c^\ell), \quad (C9)$$

$$N(\psi^\ell, c^\ell) = (A_x \psi^\ell) \ast (A_x c^\ell + b_x) + (A_y \psi^\ell) \ast (A_y c^\ell), \quad (C10)$$

$$f(c^\ell) = -c^\ell \ast (c^\ell - 1) \ast (c^\ell + d), \quad (C11)$$

with ‘$\ast$’ denoting componentwise multiplication as used in Matlab; for $n = n_x n_y$.

I.e., Given $c^0$, $T^0$, $\psi^0$,

(1) Solve the convection-diffusion equations (C5) and (C6) to get a new concentration $c^{\ell+1}$ and a new temperature $T^{\ell+1}$.

$$c^{\ell+1} = c^\ell + dt \left[ -F(\psi^\ell, c^\ell) + [Ac^\ell + ] + Daf(c^\ell) \right]$$

$$T^{\ell+1} = T^\ell + dt \left[ -F(\psi^\ell, T^\ell) + Le[AT^\ell + ] + sgn(\phi)Daf(c^\ell) \right]$$

(2) Update $\omega^{\ell+1}$ from (C7) using $c^{\ell+1}$, $T^{\ell+1}$, and $\psi^\ell$:

$$\omega^{\ell+1} = -Re \left[ N(\psi^\ell, c^{\ell+1}) + A_y c^{\ell+1} \right] + RT \left[ N(\psi^\ell, T^{\ell+1}) + A_y T^{\ell+1} \right]$$
(3) Solve $\tilde{\psi}^{\ell+1}$ from (C8) using $\bar{\omega}^{\ell+1}$:

$$A\tilde{\psi}^{\ell+1} = -\bar{\omega}^{\ell+1},$$

(4) Update again $\omega^{\ell+1}$ from (C7) using $\bar{c}^{\ell+1}$, $\bar{T}^{\ell+1}$, and $\tilde{\psi}^{\ell+1}$:

$$\omega^{\ell+1} = -R_c\left[ N(\tilde{\psi}^{\ell+1}, \bar{c}^{\ell+1}) + A_y\bar{c}^{\ell+1} \right] + R_T\left[ N(\tilde{\psi}^{\ell+1}, \bar{T}^{\ell+1}) + A_y\bar{T}^{\ell+1} \right]$$

(5) Solve again for $\psi^{\ell+1}$ from (C8) using the newly updated $\omega^{\ell+1}$:

$$A\psi^{\ell+1} = -\omega^{\ell+1},$$

(6) Solve the convection-diffusion equations (C5) and (C6) to get a new concentration $c^{\ell+1}$ and a new temperature $T^{\ell+1}$:

$$c^{\ell+1} = c^{\ell} + dt\left[ -F(\psi^{\ell+1}, c^{\ell}) + [Ac^{\ell} + b] + Daf(c^{\ell}) \right]$$

$$T^{\ell+1} = T^{\ell} + dt\left[ -F(\psi^{\ell+1}, T^{\ell}) + Le[AT^{\ell} + b] + sgn(\phi)Daf(c^{\ell}) \right].$$
Appendix D. Plots of POD Basis

Figure D1. The first 6 POD basis functions of the solution snapshots $c$, $\psi$, and $\omega$; Peclet numbers $Pe = 250$. 
Figure D2. The first 6 POD basis functions of the snapshots of nonlinear functions $F$, $N$, and $f$: Peclet numbers $Pe = 250$. 