computation

Article

Online Adaptive Local-Global Model Reduction for Flows in Heterogeneous Porous Media

Yalchin Efendiev [1,2,*], Eduardo Gildin [3] and Yanfang Yang [1]

[1] Department of Mathematics, Texas A&M University, College Station, TX 77843, USA; yyfang.y@gmail.com

[2] Numerical Porous Media (NumPor), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

[3] Department of Petroleum Engineering, Texas A&M University, College Station, TX 77843, USA; egildin@tamu.edu

[*] Correspondence: efendiev@math.tamu.edu; Tel.: +1-979-458-0836

Academic Editor: Qinjun Kang
Received: 8 March 2016; Accepted: 23 May 2016; Published: 7 June 2016

Abstract: We propose an online adaptive local-global POD-DEIM model reduction method for flows in heterogeneous porous media. The main idea of the proposed method is to use local online indicators to decide on the global update, which is performed via reduced cost local multiscale basis functions. This unique local-global online combination allows (1) developing local indicators that are used for both local and global updates (2) computing global online modes via local multiscale basis functions. The multiscale basis functions consist of offline and some online local basis functions. The approach used for constructing a global reduced system is based on Proper Orthogonal Decomposition (POD) Galerkin projection. The nonlinearities are approximated by the Discrete Empirical Interpolation Method (DEIM). The online adaption is performed by incorporating new data, which become available at the online stage. Once the criterion for updates is satisfied, we adapt the reduced system online by changing the POD subspace and the DEIM approximation of the nonlinear functions. The main contribution of the paper is that the criterion for adaption and the construction of the global online modes are based on local error indicators and local multiscale basis function which can be cheaply computed. Since the adaption is performed infrequently, the new methodology does not add significant computational overhead associated with when and how to adapt the reduced basis. Our approach is particularly useful for situations where it is desired to solve the reduced system for inputs or controls that result in a solution outside the span of the snapshots generated in the offline stage. Our method also offers an alternative of constructing a robust reduced system even if a potential initial poor choice of snapshots is used. Applications to single-phase and two-phase flow problems demonstrate the efficiency of our method.

Keywords: online adaptive model reduction; local model reduction; POD global model reduction; discrete empirical interpolation method; flows in heterogeneous media

1. Introduction

Reduced order models (ROM) aims at reducing the computational complexity and, in turn, the simulation time of large-scale dynamical systems by approximating the state-space to much lower dimensions. This yields reduced models that can produce nearly the same input/output response characteristics as the original ones. Approaches to construct ROMs by projection include the Proper Orthogonal Decomposition (POD) method [1], the balanced truncation (BT) [2], the Krylov subspace projection methods, the trajectory-piecewise linear (TPWL) techniques [3] and variants of them.

Among these approaches, the POD method is perhaps the most popular one. For example, the POD was successfully applied to a number of areas [4–11].

For POD projection type model reduction, the process typically consists of an offline stage followed by an online stage. In the offline stage, the low dimension solution space, also called as the POD subspace, is generated and a reduced system for the problem at hand is constructed by projecting onto the low dimension solution space; in the online stage, on the other hand, an approximated solution is obtained by integrating the reduced dynamical system. The online solution relies only on the pre-computed quantities from the offline stage. Therefore, a good approximation from the reduced model can be expected only if the offline information is a good representation of the problem with initial input parameters.

Much progress on adaptivity has been made in the context of model reduction. Offline adaptive methods [12] seek to provide a better reduce solution space while the reduced system is constructed in the offline stage. However, once the reduced system is derived, it stays fixed and is kept unchanged in the online stage. Therefore, the online solution relies only on the pre-computed information from the offline stage which is not incorporated at the online phase. Unlike the offline adaptation, online adaptive methods modify the reduced system during the computations in the integration stage. Most of the existing global online adaptive methods [13–15] rely only on precomputed quantities from the offline stage. We consider here a different approach whereby online adaptivity is performed by incorporating new data that becomes available in the online stage.

We use the POD Galerkin method for global model reduction. However, for a nonlinear system, projection alone is not sufficient to deliver a computationally efficient model, since the nonlinear terms require computations that often render solving the reduced system almost as expensive as solving the full order system. To deal with nonlinearities, we apply the Discrete Empirical Interpolation Method (DEIM) proposed in [16]. The DEIM samples the nonlinear terms at selected interpolation points and then combines interpolation and projection to obtain an approximation in a low-dimensional DEIM space.

In the linear system setting, one can compute the projection matrices based on the fixed operators offline, that is, outside of the time dependent loop, and perform the reduction by projecting directly onto a much smaller subspace as in Equation (2). However, for nonlinear systems, as in the case of flow in heterogeneous porous media, one cannot perform such projection without having to recompute the operators for every time step. In this case, the best choice is to compute the projection matrices offline and perform the projection and integration of the reduced model online. This setting is particularly useful for the POD-DEIM-based model reduction, whereby the projection matrices are computed offline and are based solely on runs of the fine scale model. In summary, in the offline stage, we construct the reduced order model; in the online stage, given particular parameters or inputs, the reduced model can be approximated efficiently and integrated in time. Online computations have costs that may not depend on the size of the fine scale model. For this reason, we perform the online adaptivity infrequently.

In the online stage, we propose a criterion to decide when updates are required. Once the criterion is satisfied, we adapt the POD subspaces, the DEIM space and the DEIM interpolation points for the nonlinear functions by incorporating new data that becomes available online. For the adaption of the POD subspace, we propose a local-global strategy. First, instead of global error indicators based on the residual which are expensive to compute, we use local error indicators to monitor when global POD basis adaption is needed. By using local error indicators, the computational time can be reduced as one avoids computing global error indicators. Secondly, we employ a local model order reduction method, *i.e.*, the generalized multiscale finite element method (GMsFEM) [17–20], to solve the global residual problem to get the new POD basis function. Based on local error indicators (see [21–24] for local error indicators for GMsFEM), some local multiscale basis functions are updated and used to compute the online global modes inexpensively. For the adaption of DEIM, we employ a modified low rank updates method investigated in [25], which introduces computational costs that scale linearly in the number of unknowns of the full-order system. We remark that the offline local-global approaches have been studied in the literature [26–29]. In the proposed method, we develop online multiscale basis functions for both local and global updates.

Our particular interest here is subsurface flow problems. POD techniques were first used by Vermeulen *et al.* [30] in the context of subsurface flow problems, e.g., that was a groundwater flow problem with a single water component. Jansen *et al.* [9,31] applied POD to a two-phase oil-water flow. In [32] TPWL was introduced to solve an oil-water flow system. A local-global multi-scale model reduction was proposed in [33]. A constraint reduction based on POD-TPWL was present in [34]. In [11,35], POD-DEIM model reduction was used to solve a two-phase flow problem with gravity. However, these methods can only deal with cases that the variation in inputs is within a certain range, *i.e.*, the offline procedure should be able to anticipate almost all the the dynamics of the online system. In many applications, such as optimization and history matching, the final solution trajectory may be hard to predict before the problem is solved. Therefore, a good POD subspace just from using the offline information may not yield robust approximation solutions. Here, we propose to adapt the reduced system by using online data in the online stage. Our numerical results show that one can achieve a good approximation of the solution at a reduced cost.

This paper is organized as follows. In Section 2, the concept of model reduction based on POD-DEIM is given; a short review of POD and DEIM is presented. In Section 3, online adaptive POD is described in Section 3.1, and a discussion of online adaptive DEIM approximation is described in Section 3.2. Applications to single-phase flow and two-phase flow problems are given in Sections 4.1 and 4.2 respectively. The paper ends with our conclusion in Section 5.

2. Preliminaries

In this section, we will first review the concept of POD-DEIM based model reduction method, which can reduce the dimension of large-scale ODE systems regardless of their origin. A large source of such systems is the semidiscretization of time dependent PDEs. After spatial discretization for a nonlinear PDE, we get a system of nonlinear ODEs of the form

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{B}\mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t)) \tag{1}$$

where $\mathbf{y}(t) \in \mathbb{R}^n$, $\mathbf{B} \in \mathbb{R}^{n\times n}$ is a constant matrix, and $\mathbf{f}(\mathbf{y}(t))$ is a nonlinear function.

We start our exposition by briefly reviewing the global model reduction framework. Let $\mathbf{V}_k \in \mathbb{R}^{n\times k}(k \ll n)$ be an orthonormal matrix consisting of reduced basis. Replacing $\mathbf{y}(t)$ in Equation (1) by $\mathbf{V}_k\tilde{\mathbf{y}}(t)(\tilde{\mathbf{y}} \in \mathbb{R}^k)$, and by Galerkin projection of the system (1) onto $\mathbf{V}_k$, we get the reduced system of (1) (the super index T means the transpose of a matrix):

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \mathbf{V}_k^T\mathbf{B}\mathbf{V}_k\tilde{\mathbf{y}}(t) + \mathbf{V}_k^T\mathbf{f}(\mathbf{V}_k\tilde{\mathbf{y}}(t)). \tag{2}$$

The quality of the above POD Galerkin approximation depends on the choice of reduced basis. Snapshot based POD constructs a set of global basis functions from a singular value decomposition (SVD) of snapshots, which are discrete samples of trajectories associated with a particular set of boundary conditions and inputs. It is expected that the samples will be representative of the solution manifold of the problem of interest. Among the various techniques for obtaining a reduced basis, POD constructs a reduced basis that is optimal in the sense that a certain approximation error concerning the snapshots is minimized. The details of POD are presented in Section 2.1.

To evaluate the nonlinear term $\mathbf{V}_k^T\mathbf{f}(\mathbf{V}_k\tilde{\mathbf{y}}(t))$ in Equation (2), we first need to project the solution from the reduced space to the fine solution space, then evaluate the nonlinear function $\mathbf{f}$ at all the n fine components, finally we project the n fine components back onto $\mathbf{V}_k$. The process results in a cost that is as expensive as solving the original system. To handle this inefficiency, we use DEIM [16], which is reviewed in Section 2.2.

The POD-DEIM model reduction method consists of two stages. In the offline stage, given some inputs, which can be boundary conditions, or control parameters, we run several simulations for the system described by Equation (1) to produce snapshots for the solution states (pressure, saturation or concentration in subsurface flow simulation context) and nonlinear functions. POD is performed to construct the reduced subspaces for the states, and interpolants for the nonlinear functions. In the online stage, the approximate solution of the problem is obtained by writing the solution as a linear combination of the reduced basis, which was fixed at the offline phase. Next, we describe in more details the aforementioned process.

2.1. Proper Orthogonal Decomposition

The POD basis in Euclidean space is specified formally in this section. We refer to [36] for more details on the POD basis in general Hilbert space.

Given a set of snapshots $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{n_s}\} \in \mathbb{R}^n$, let $\mathbb{Y} = \text{span}\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{n_s}\}$ and $r = \dim(\mathbb{Y})$. A POD basis of dimension $k < r$ is a set of orthonormal vectors $\{\boldsymbol{\phi}_i\}_{i=1}^k \in \mathbb{R}^n$ whose linear span best approximates the space $\mathbb{Y}$. The basis set $\{\boldsymbol{\phi}_i\}_{i=1}^k \in \mathbb{R}^n$ solves the minimization problem

$$\min_{\{\boldsymbol{\phi}_i\}_{i=1}^k} \sum_j^{n_s} \left\| \mathbf{y}_j - \sum_{i=1}^k (\mathbf{y}_j^T \boldsymbol{\phi}_i) \boldsymbol{\phi}_i \right\|_2^2 \tag{3}$$

with $\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j = \delta_{ij}$.

It is well known that the solution to Equation (3) is provided by the set of the left singular vectors of the *snapshot matrix* $\mathbb{S}_y = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{n_s}] \in \mathbb{R}^{n \times n_s}$. Applying SVD to $\mathbb{S}_y$, we get

$$\mathbb{S}_y = \mathbf{V}\,\boldsymbol{\Lambda}\,\mathbf{W}^T \tag{4}$$

where $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_r] \in \mathbb{R}^{n_s \times r}$ are the left and right projection matrices respectively, and $\boldsymbol{\Lambda} = \text{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r \times r}$ with $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > 0$. The rank of $\mathbb{S}_y$ is $r \leq \min(n, n_s)$. The first k column vectors $\{\mathbf{v}_i\}_{i=1}^k$ of $\mathbf{V}$ are selected as the POD basis (the criterion to decide k is explained later). The minimum l^2-norm error from approximating the snapshots using the POD basis is then given by

$$\sum_j^{n_s} \left\| \mathbf{y}_j - \sum_{i=1}^k (\mathbf{y}_j^T \mathbf{v}_i) \mathbf{v_i} \right\|_2^2 = \sum_{i=k+1}^r \sigma_i^2. \tag{5}$$

The appropriate number of POD basis to be retained for a prescribed accuracy can be determined, for example, by means of the fractional energy (see discussions in [37] for selecting modes):

$$E = \frac{\sqrt{\sum_{i=1}^k \sigma_i^2}}{\sqrt{\sum_{i=1}^{n_s} \sigma_i^2}}. \tag{6}$$

A small number of POD basis is retained if the singular values decay fast. This decay depends on the intrinsic dynamics of the system and the selection of the snapshots. In the next section, we review DEIM, which is applied to reduce the nonlinear terms.

2.2. Discrete Empirical Interpolation Method (DEIM)

In the offline stage, we get the snapshots of the nonlinear function $\mathbf{f}(\mathbf{y}(t))$ as:

$$\{\mathbf{f}(\mathbf{y}(t_1)), \cdots, \mathbf{f}(\mathbf{y}(t_M))\} \in \mathbb{R}^n.$$

By applying POD to the above snapshot matrix and selecting the first m eigen-vectors corresponding to the dominant eigen-values, we obtain the DEIM basis $\mathbf{U} \in \mathbb{R}^{n \times m}$. DEIM [16] selects

m distinct interpolation points $\{p_1, \cdots, p_m\} \in \{1, \cdots, n\}$ and assemble the DEIM interpolation points matrix $\mathbf{P} = [e_{p_1}, \cdots, e_{p_m}] \in \mathbb{R}^{n\times m}$, where $e_i \in \{0,1\}^n$ is the i-th canonical unit vector. The DEIM interpolant of $\mathbf{f}$ is defined by $(\mathbf{U}, \mathbf{P})$. The DEIM approximation of $\mathbf{f}$ is derived, based on $(\mathbf{U}, \mathbf{P})$, as

$$\mathbf{f}(\mathbf{y}(t)) \approx \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\mathbf{y}(t))$$

where $\mathbf{P}^T\mathbf{f}(\mathbf{y}(t))$ samples $\mathbf{f}$ at m components only. In what follows, local model reduction will be reviewed.

2.3. Local Model Order Reduction via GMsFEM

We will give a brief overview of the local model reduction using the GMsFEM [17–20]. Let $\mathcal{T}^{\mathcal{H}}$ be a usual conforming partition of the domain Ω into finite elements called coarse elements, and H is the coarse mesh size. Then, each coarse element is divided into a connected union of fine elements, which are conforming across coarse element faces. This fine grid partition is denoted by $\mathcal{T}^h$, with h as fine mesh size . The fine grid is used to construct the locally supported multiscale basis functions, and compute the reference (fine) solution in the numerical examples. The reference solution is computed on the fine grid. Let $\{x_i | 1 \leq i \leq N_c\}$ be the set of nodes on the coarse grid with N_c being the number of nodes in $\mathcal{T}^{\mathcal{H}}$. For each coarse node x_i, a coarse neighborhood is defined as $\omega_i = \bigcup\{K_j \in \mathcal{T}^{\mathcal{H}}; x_i \in \bar{K}_j\}$. An illustration of the definitions is given in Figure 1. The multiscale basis functions are nodal based and supported on coarse neighborhoods. Specifically, for each coarse node x_i, we construct a set of basis functions $\{\boldsymbol{\psi}_k^{\omega_i} | k = 1, 2, \cdots, L_i\}$ such that each $\boldsymbol{\psi}_k^{\omega_i}$ is supported on the coarse neighborhood ω_i. Note that L_i can be different for different nodes.

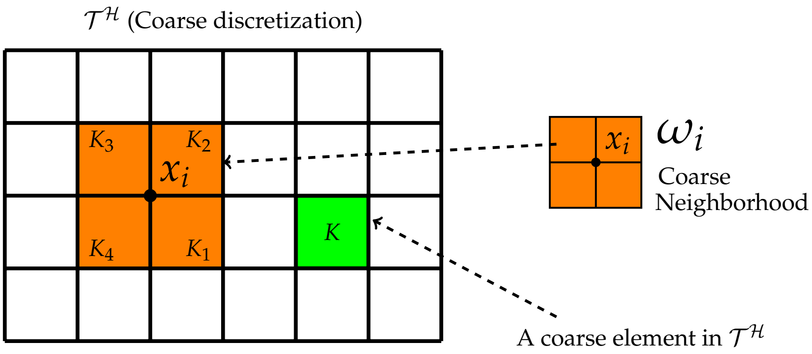
$\mathcal{T}^{\mathcal{H}}$ (Coarse discretization)
K_3
K_2
x_i
K_4
K_1
K
x_i
ω_i
Coarse Neighborhood
A coarse element in $\mathcal{T}^{\mathcal{H}}$

Figure 1. Illustration of a coarse neighborhood and a coarse element.

Let ω be a given coarse neighborhood, to construct the multiscale basis functions supported on ω, first we construct a snapshot space V_{snap}^{ω}. V_{snap}^{ω} is a set of functions defined on ω and contains all or most necessary information of the fine scale solution restricted to ω. A typical choice of V_{snap}^{ω} is local flow solutions, which are constructed by solving local problems with all possible boundary conditions. This can be expensive and for this reason, one can use randomized boundary conditions (see [38,39]) and construct only a few more snapshots than the number of desired local basis functions. Next, we construct the offline space V_{off}^{ω} for ω by performing a spectral decomposition of V_{snap}^{ω}, and choosing the eigenvectors corresponding to the largest eigenvalues. The offline space is defined as $\Phi_{\text{off}}^L = \text{span}\{\boldsymbol{\psi}_k^i : 1 \leq k \leq L_i, 1 \leq i \leq N_c\}$. We refer to [17,40] for details. The above computations are performed in the offline stage. For time dependent problems, as time advancing, Φ_{off}^L may not be good enough to represent the behavior of the full resolved solution at some time instants. We use an adaptive enrichment algorithm which allows one to use more basis functions in regions with large errors. We refer to [40] for details.

3. Online Adaptive POD-DEIM Model Reduction

In this section, we discuss the construction of the online adaptive POD-DEIM model reduction framework. We start with the online adaptive POD and later introduce the adaptive POD-DEIM algorithm.

3.1. Online Adaptive Local-Global Proper Orthogonal Decomposition

We adapt the POD subspaces at some particular instants as time proceeds. Two issues are addressed here: at which instants to adapt, and how to adapt the POD subspaces. The updates are performed online, therefore the goal is to keep the computational cost as low as possible.

The main idea of adaptive local-global POD model order reduction method is summarized in Algorithm 1. The superindex G stands for global, superindex L stands for local, and superindex T means transpose of a matrix. Φ^G_{k-1} is the initial global POD matrix for time step $k-1$. Similarly, Φ^L_{k-1} is the initial local multiscale basis matrix for time step $k-1$. In Figure 2, a flow chat corresponding to Algorithm 1 is given. In Figure 3, we present an schematic description of Adaptive-POD-1. We note that there is no global fine problem computation in the method.

Algorithm 1 Adaptive Local-Global POD Model Order Reduction Method

```
   OFFLINE STAGE:
1: Construction of snapshots for states, local off-line space (consists of local ms basis) by GMsFEM
2: Construction of POD subspaces (POD projection matrices)
   ONLINE STAGE : for step $k$ adaption
3: INPUT : Global POD basis matrix $\Phi^G_{k-1}$, local off-line space $\Phi^L_{k-1}$
4: Solve the reduced system: $(\Phi^G_{k-1})^T \mathbf{A} \Phi^G_{k-1} \mathbf{X}^{k-1}_r = (\Phi^G_{k-1})^T \mathbf{F}_k$ (Global reduced-order model)
5: Compute local error indicators, and decide if adaption is needed. If yes, go to 6. Otherwise, go to
   next time step $k+1$
6: Solve the global residual problem $\mathbf{A}\boldsymbol{\phi}^{k,\text{online}} = \mathbf{F}_k - \mathbf{A}\Phi^G_{k-1}\mathbf{X}^{k-1}_r$ for $\boldsymbol{\phi}^{k,\text{online}}$ by adaptive local
   method with initial local off-line space $\Phi^L_{k-1}$
7: Update the POD subspace by Adaptive-POD-1 or Adaptive-POD-2
8: OUTPUT : Global POD basis matrix $\Phi^G_k$, local offline space $\Phi^L_k$
```

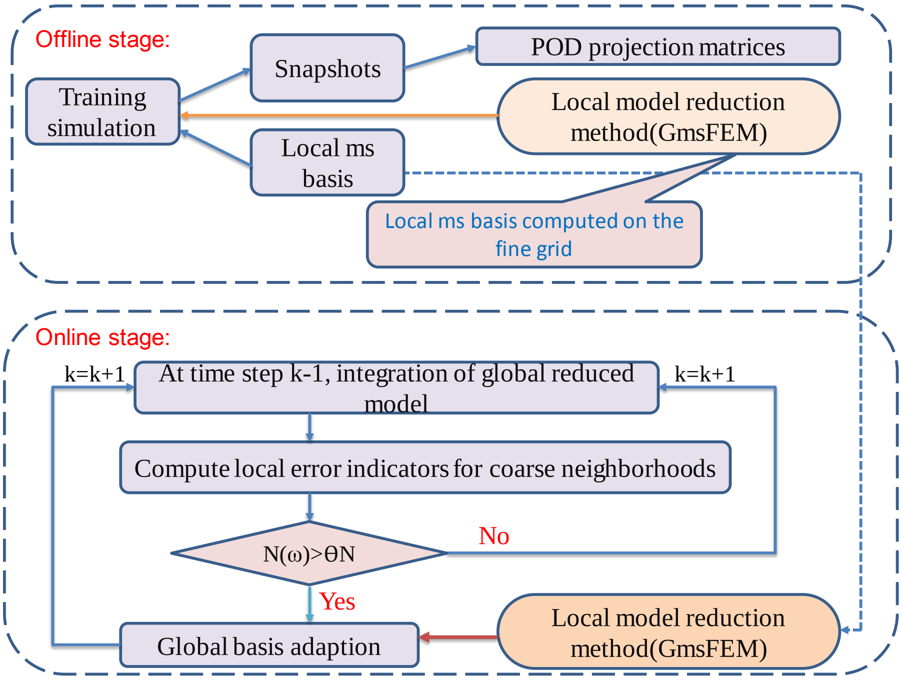
Offline stage:
Snapshots
POD projection matrices
Training simulation
Local model reduction method(GmsFEM)
Local ms basis
Local ms basis computed on the fine grid
Online stage:
k=k+1
At time step k-1, integration of global reduced model
k=k+1
Compute local error indicators for coarse neighborhoods
N(ω)>θN
No
Yes
Global basis adaption
Local model reduction method(GmsFEM)

Figure 2. Flowchat of the online adaptive local-global POD method.

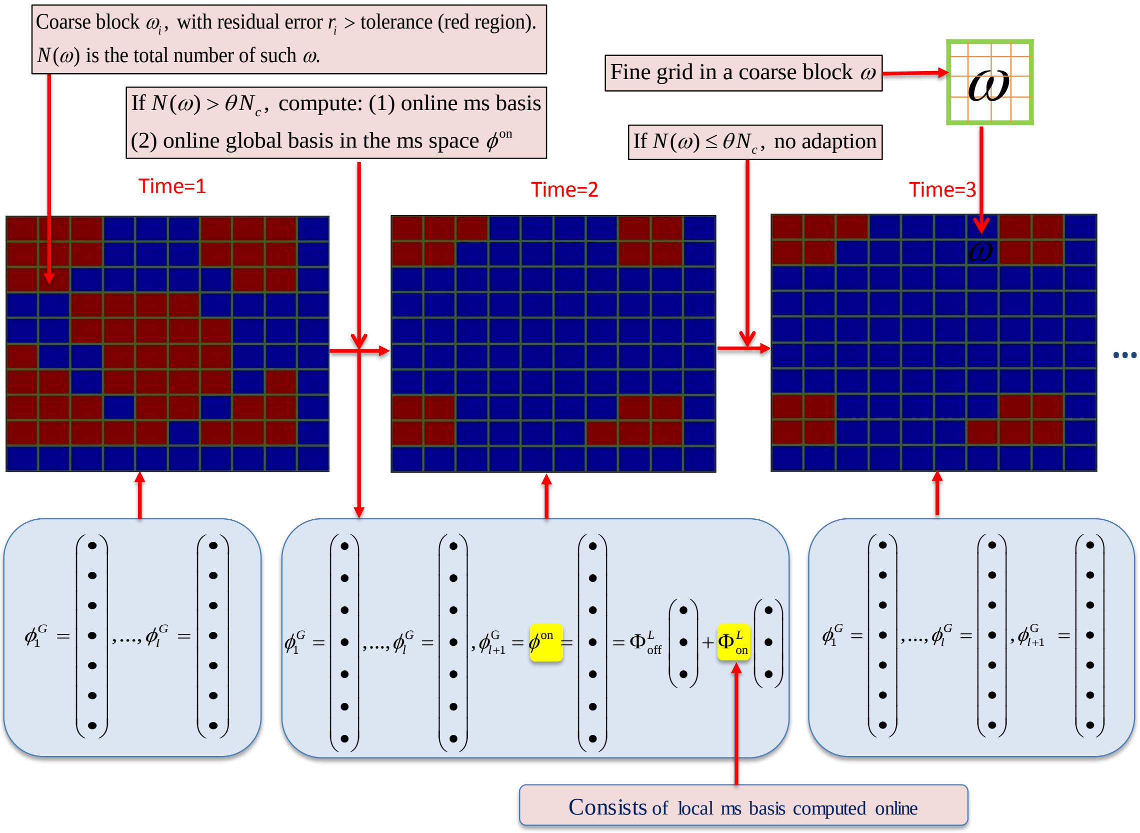
Coarse block ω_i, with residual error r_i > tolerance (red region).
$N(\omega)$ is the total number of such ω.
If $N(\omega) > \theta N_c$, compute: (1) online ms basis
(2) online global basis in the ms space ϕ^{on}
Fine grid in a coarse block ω
ω
If $N(\omega) \le \theta N_c$, no adaption
Time=1
Time=2
Time=3
ω
...
$\phi_1^G =$, ..., $\phi_l^G =$
$\phi_1^G =$, ..., $\phi_l^G =$, $\phi_{l+1}^G = \phi^{on} =$ $= \Phi_{off}^L$ $+ \Phi_{on}^L$
$\phi_1^G =$, ..., $\phi_l^G =$, $\phi_{l+1}^G =$
Consists of local ms basis computed online

Figure 3. Schematic description of the online adaptive local-global POD method.

In the offline stage, a training simulation is run by the GMsFEM. Local multiscale bases are constructed, and snapshots for states at some time steps are saved. Next, POD is performed on the snapshots to construct the POD projection matrices. In the online stage, at every time step we solve a global reduced system, which is usually very small, for example, in Section 4.1, the largest dimension is 8. Next we use our criterion to decide if global basis adaption is necessary. The criterion is specified in the next paragraph. If adaption is needed, a new global basis is obtained by solving a residual problem with GMsFEM. Again, in Section 4.1, a residual problem is of size about 10^4; in Table 1, the size of the reduced problem of the residual problem by GMsFEM is about 600.

It is important to monitor when the approximation from the reduced system is no longer sufficiently accurate according to a particular criterion, leading to adaptation of the POD subspaces. This adaption must be performed when the system dynamics has deviated from the current POD subspace as reflected by the residual evaluation. In [41], some residual based POD modes were added to the old POD subspace for Navier-Stokes equations. In [42], a second error estimate based on a second Galerkin system to account for situations in which qualitative changes in the dynamics occur was introduced. In this case, two reduced systems must be solved at a time, thus incurring higher computational cost. In [43], a residual based indicator was used in nonlinear dissipative systems. Here, we propose to use local error indicators based on the multiscale basis, which is cheaper to be computed. The main idea is as follows:

- For coarse blocks $\omega_1, \omega_2, \cdots, \omega_N$, compute their corresponding H^{-1} norm of the residual, and denote as $r_1, r_2, \cdots, r_{N_c}$.
- Count the total number $N(\omega)$ of coarse blocks with $r_i >$ a certain error tolerance. Here a large error means the current POD modes cannot give a good representation of the features in that coarse block.
- If $N(\omega) > \theta N_c$, then adaption is needed, θ is a fraction of the total number of coarse blocks.

In general, one can use a threshold for the sum of the local indicators. The choice of the threshold is typically based on experimental results and analysis. The larger the threshold is, more updates are performed. Next, we describe how to update the POD subspace.

For the POD subspace adaption at step k, suppose N_{k-1} number of initial POD modes $\{\boldsymbol{\phi}_1^G, \cdots, \boldsymbol{\phi}_{N_{k-1}}^G\}$ are given. Φ^G is the POD matrix with $\{\boldsymbol{\phi}_1^G, \cdots, \boldsymbol{\phi}_{N_{k-1}}^G\}$ as column vectors. We obtain the online snapshot vector $\boldsymbol{\phi}^{k,\text{online}}$ by solving a residual problem (for linear system), or the high fidelity system (for nonlinear system). We illustrate how to get the new online global basis for the linear case. Suppose we are solving Equation (1). At every time step, we need to solve a linear system of the form

$$\mathbf{A}\mathbf{X} = \mathbf{F}_k. \tag{7}$$

The POD reduced order system for Equation (7) is

$$(\Phi^G)^T \mathbf{A} \Phi^G \mathbf{X}_r = (\Phi^G)^T \mathbf{F}_k. \tag{8}$$

The residual is $\mathbf{res} = \mathbf{F}_k - \mathbf{A}\Phi^G \mathbf{X}_r$. If update is needed, then we solve the residual problem

$$\mathbf{A}\boldsymbol{\phi}^{k,\text{online}} = \mathbf{res} \tag{9}$$

to obtain the online basis $\boldsymbol{\phi}^{k,\text{online}}$.

Note that Equation (9) is expensive to solve because it is a fine-grid problem. We propose to use the local model reduction method as described in Section 2.3 to solve it. In the following, we introduce two methods to adapt the POD subspace after $\boldsymbol{\phi}^{k,\text{online}}$ is obtained. One way is to add $\boldsymbol{\phi}^{k,\text{online}}$ directly to $\{\boldsymbol{\phi}_1^G, \cdots, \boldsymbol{\phi}_{N_{k-1}}^G\}$, therefore the new POD subspace is the span of $\{\boldsymbol{\phi}_1^G, \cdots, \boldsymbol{\phi}_{N_{k-1}}^G, \boldsymbol{\phi}^{k,\text{online}}\}$. In this way, the number of POD modes increases by one after every adaption. We denote this method as Adaptive-POD-1.

It is possible that as time proceeds, some modes in the POD subspace become negligible (redundant) in representing the near future dynamics, therefore it can be removed from the POD subspace. This can be achieved in various ways. In our simulations. we follow the following procedure. Apply POD to the set of vectors, see [43]

$$\{w_1\boldsymbol{\phi}_1^G, \cdots, w_{N_{k-1}}\boldsymbol{\phi}_{N_{k-1}}^G, \boldsymbol{\phi}^{k,\text{online}}\}$$

where the weights w_i are defined as

$$w_i = \min\left\{\frac{\sigma_i}{\sqrt{\sum_{j=1}^{N_{k-1}} \sigma_j^2}}, \frac{\langle |C_i| \rangle}{\sqrt{\sum_{j=1}^{N_{k-1}} \langle |C_j| \rangle^2}}\right\}$$

where σ_i are singular values associated with $\boldsymbol{\phi}_i$, C_i is the amplitude of $\boldsymbol{\phi}_i^G$, *i.e.*, the coefficient of mode $\boldsymbol{\phi}_i^G$, and $\langle |C_i| \rangle$ is the temporal mean value of $|C_i|$ obtained from the reduced system starting from the first instant since the last update until $k-1$. In this way, the weights w_i eliminate those modes whose average energy decreased considerably since the last update. Therefore only a small number of POD modes is retained through the whole simulation. We denote this method as Adaptive-POD-2.

3.2. Online Adaptive DEIM

In this section, we summarize the idea of online adaptive DEIM as in [25]. First, we adapt the DEIM space by rank one updates. Secondly, we adapt the DEIM interpolation points. Its algorithm is given in Algorithm 2. It is shown in [25] that the run time cost of the adaption scales linearly with the number of degrees of freedom of the full-order system.

Algorithm 2 Online Adaptive DEIM [25]

```
**INPUT :** $(\mathbf{U}_{k-1}, \mathbf{P}_{k-1})$, number of sampling points $m_s$, window size $w$
Select $m_s$ points from $\{1, 2, \cdots, n\}$ that are not points in $\mathbf{P}_{k-1}$.
Assemble sampling points matrix $\mathbf{S}_k$ by combining $\mathbf{P}_{k-1}$ and the selected points.
Compute the coefficient matrix $\mathbf{C}_k$ for $[\hat{\mathbf{y}}(t_k), \hat{\mathbf{y}}(t_{k-1}), \cdots, \hat{\mathbf{y}}(t_{k-w+1})]$ with interpolant $(\mathbf{U}_{k-1}, \mathbf{S}_k)$
Solve the minimization problem (10) for $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k$
**Update basis** $\mathbf{U}_k = \mathbf{U}_{k-1} + \boldsymbol{\alpha}_k \boldsymbol{\beta}_k^T$
Compute $\mathrm{diag}(\mathbf{U}_k^T \mathbf{U}_{k-1})$
Find the index $i$ of the pair of basis vectors which are nearest to orthogonal
Let $\mathbf{u}_k$ be the $i-$ th column of the adapted basis $\mathbf{U}_k$
Store all other $m-1$ columns of $\mathbf{U}_k$ in $\hat{\mathbf{U}}_k \in \mathbb{R}^{n \times (m-1)}$
Store all other $m-1$ columns of $\mathbf{P}_{k-1}$ in $\hat{\mathbf{P}}_k \in \mathbb{R}^{n \times (m-1)}$
Approximate $\mathbf{u}_k$ with the DEIM interpolant $(\hat{\mathbf{U}}_k, \hat{\mathbf{P}}_k)$ as $\hat{\mathbf{u}}_k = \hat{\mathbf{U}}_k (\hat{\mathbf{P}}_k^T \hat{\mathbf{U}}_k)^{-1} \hat{\mathbf{P}}_k^T \mathbf{u}_k$
Find the index $p_k^i$ that corresponds to largest residual of $|\hat{\mathbf{u}}_k - \mathbf{u}_k|$
**Update interpolation matrix** $\mathbf{P}_k$: if $\mathbf{e}_{p_k^i}$ is not a column of $\mathbf{P}_{k-1}$, then for the $i-$ th column of $\mathbf{P}_{k-1}$,
    replace interpolation point $\mathbf{e}_{p_{k-1}^i}$ with $\mathbf{e}_{p_k^i}$
**OUTPUT :** $(\mathbf{U}_k, \mathbf{P}_k)$
```

For the basis adaption for $\mathbf{f}(\mathbf{y}(t))$ at step k, suppose an initial DEIM interpolant $(\mathbf{U}_{k-1}, \mathbf{P}_{k-1})$ is given. We extend the interpolation points matrix $\mathbf{P}_{k-1}$ as

$$\mathbf{S}_k = [\mathbf{e}_{p_1}, \cdots, \mathbf{e}_{p_m}, \mathbf{e}_{p_{m+1}}, \cdots, \mathbf{e}_{p_{m+m_s}}].$$

The extra m_s points are chosen from $\{1, \cdots, n\} \backslash \{p_1, \cdots, p_m\}$ randomly. Therefore, we get a new DEIM interpolant $(\mathbf{U}_{k-1}, \mathbf{S}_k)$. Note that $\mathbf{U}_{k-1}$ is still the same. Next we define a window of size w that contains the vectors of $\hat{\mathbf{y}}(t_k), \hat{\mathbf{y}}(t_{k-1}), \cdots, \hat{\mathbf{y}}(t_{k-w+1})$ obtained in the previous online computations; an alternative is to also include intermediate states from the Newton-Raphson iterations of the previous online computations. We then construct DEIM approximations of the nonlinear function $\mathbf{f}$ at the vectors $\hat{\mathbf{y}}(t_k), \hat{\mathbf{y}}(t_{k-1}), \cdots, \hat{\mathbf{y}}(t_{k-w+1})$ with the DEIM interpolant $(\mathbf{U}_{k-1}, \mathbf{S}_k)$. For each $\hat{\mathbf{y}}(t_{k_i})$, the coefficient is derived as

$$\mathbf{c}_k(\mathbf{y}(t_{k_i})) = (\mathbf{S}_k^T \mathbf{U}_{k-1})^+ \mathbf{S}_k^T \mathbf{f}(\hat{\mathbf{y}}(t_{k_i}))$$

which are put as columns in the coefficient matrix $\mathbf{C}_k \in \mathbb{R}^{m \times w}$. $(\mathbf{S}_k^T \mathbf{U}_{k-1})^+ \in \mathbb{R}^{m \times (m+m_s)}$ is the Moore-Penrose pseudo-inverse.

We then derive two vectors $\boldsymbol{\alpha}_k \in \mathbb{R}^n$ and $\boldsymbol{\beta}_k \in \mathbb{R}^m$ such that the adapted DEIM basis $\mathbf{U}_k = \mathbf{U}_{k-1} + \boldsymbol{\alpha}_k \boldsymbol{\beta}_k^T$ minimizes the Frobenius norm of the residual at the sampling points given by $\mathbf{S}_k$

$$\|\mathbf{S}_k^T (\mathbf{U}_k \mathbf{C}_k - \mathbf{F}_k)\|_F^2 \tag{10}$$

where $\mathbf{F}_k = [\mathbf{f}(\hat{\mathbf{y}}(t_k)), \cdots, \mathbf{f}(\hat{\mathbf{y}}(t_{k-w+1}))]$. The vectors $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k$ are obtained by solving a generalized eigenvalue problem, we refer to [25] for details.

Note that the sampling points are only used for the DEIM basis adaptation whereas ultimately the DEIM interpolation points are used for the DEIM approximation.

With the adapted DEIM basis $\mathbf{U}_k$ at step k, we also adapt the DEIM interpolation points. The standard DEIM greedy algorithm is computationally expensive to apply in the online stage, since it recomputes all m interpolation points. It is unnecessary to change all interpolation points after a rank-one update to the DEIM basis. Therefore, we adapt the DEIM interpolation points as follows. First, we compute the dot product between the old and the adapted DEIM basis

$$\mathrm{diag}(\mathbf{U}_k^T \mathbf{U}_{k-1}). \tag{11}$$

If the dot product of two normalized vectors is one, then they are colinear and the adapted basis vector has not been rotated with respect to the previous vector at step $k-1$. If it is zero, they are orthogonal. We select the basis vector $\mathbf{u}_k$ of $\mathbf{U}_k$ that corresponds to the component of Equation (11) with the lowest absolute value. The new interpolation point p_k^i is derived from $\mathbf{u}_k$ following the standard DEIM algorithm. It then replaces the interpolation point p_{k-1}^i. Other interpolation points are unchanged.

4. Applications

We present numerical examples to demonstrate the efficiency of the online adaptive POD-DEIM model reduction method. In Section 4.1, we will apply the approach to a time-dependent linear single-phase problem, therefore only adaptive POD is needed. In Section 4.2, we consider a two-phase flow model driven by viscosity. In all examples, we use time step units.

4.1. Single-Phase Flow

In this section, we consider the single-phase flow problem:

$$\frac{\partial \tilde{h}}{\partial t} + \nabla \cdot a \nabla \tilde{h} = q \quad \text{in} \quad \Omega \tag{12}$$

$$\tilde{h} = 0 \quad \text{on} \quad \partial\Omega \quad \text{(boundary condition)} \tag{13}$$

$$\tilde{h}(t=0) = \tilde{h}_0 \quad \text{in} \quad \Omega \quad \text{(initial condition).} \tag{14}$$

Discretizing in a finite element space, for example Q_1 which consists of piecewise linear functions on Ω, for space and using backward Euler method for time discretization, we obtain the following algebraic system:

$$(\mathbf{M} + \Delta t \mathbf{S})\tilde{\mathbf{h}}^n = \mathbf{M}\tilde{\mathbf{h}}^{n-1} + \Delta t \mathbf{Q}^n, \quad \text{in} \quad \Omega \tag{15}$$

where $\mathbf{M}_{ij} = \int_\Omega \psi_i \psi_j dx$ is the mass matrix, $\mathbf{S}_{ij} = \int_\Omega a \nabla \psi_i \cdot \nabla \psi_j dx$ is the stiffness matrix, $\psi_i, \psi_j \in \mathbf{Q}_1$, Δt is the time step size.

In this example, the computational domain is $\Omega = (0,1)^2$, a uniform fine mesh with $n = 100$ is used. Our proposed approach can also be used for solving problems on unstructured grids. The size of the full-order system at every time step is 10, 000. For the coarse mesh, we use the mesh size 1/10. The final simulation time is 10, time-stepping size is $\Delta t = 0.1$, therefore there are 100 snapshots in total. The logarithm of the permeability field is shown in Figure 4. In the offline stage, the right-side hand function q for the training simulation is only nonzero on the five blue regions in Figure 5. The values of q with respect to time is given in the left figure of Figure 6. In the test simulation, q is given in the right figure of Figure 6, two nonzero regions are added, one is nonzero through 60–80, the other is nonzero through 90–100, which are the red ones in Figure 5. There is a large decrease in the value $q1$ at time instant 35. In Table 2, the average L^2, H^1 errors with different number of POD basis are presented. We can see that even if we use all the 100 POD basis, the average L^2 error is still about 18%. In Table 1, we start with 5 initial global POD basis. The column G^{off} means the number of initial global POD basis for each time instants, while the column L^{off} means the number of initial local basis functions, and the column L^{add} means the number of local basis added to solve the residual problem (9). For instance, for the third row, the first column shows that we need to update at time instant 2; the second column means for time instant 2, the number of initial global POD basis is 6; the third column means for time instant 2, the number of initial local basis functions is 484; the forth column means we add 126 local basis to the multiscale solution space, from which the global basis is obtained. For the whole test simulation, we add basis at time instants 1, 2 and 35, then we get about 1% average L^2 error. The relative L^2, H^1 errors with respect to time are shown in Figure 7, which show a good agreement of our numerical solution with the reference solution. The reference solution is computed on the fine grid. Here, since the number of POD basis is already very small, we only use the Adaptive-POD-1 method.

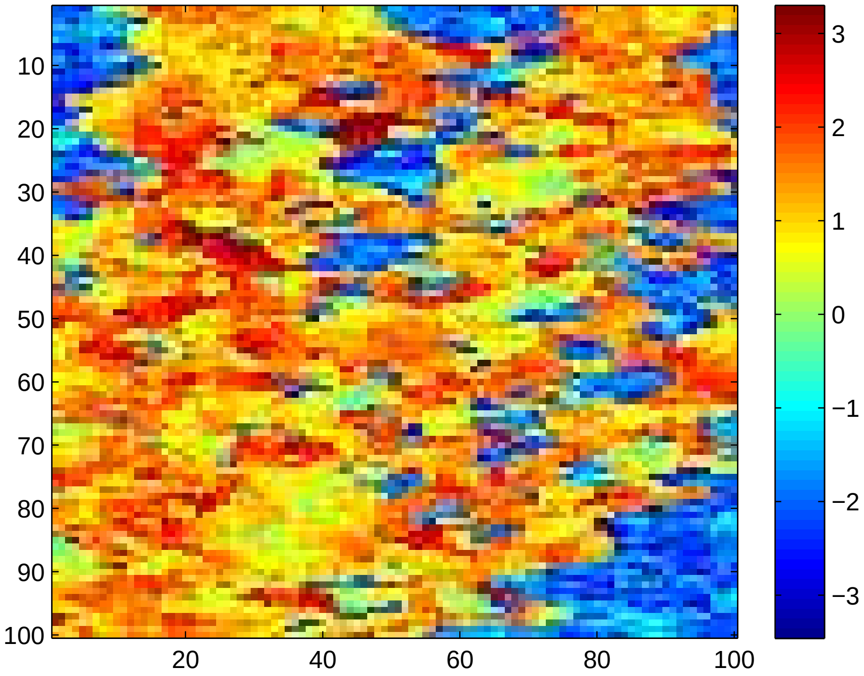

Figure 4. Logarithm of permeability field.

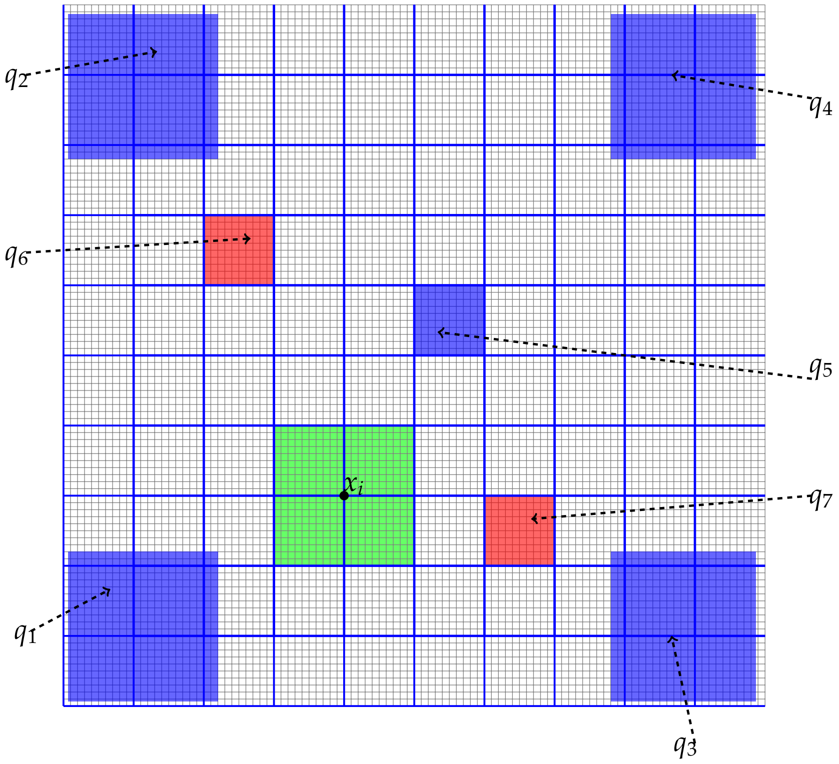

Figure 5. Grid information and q.

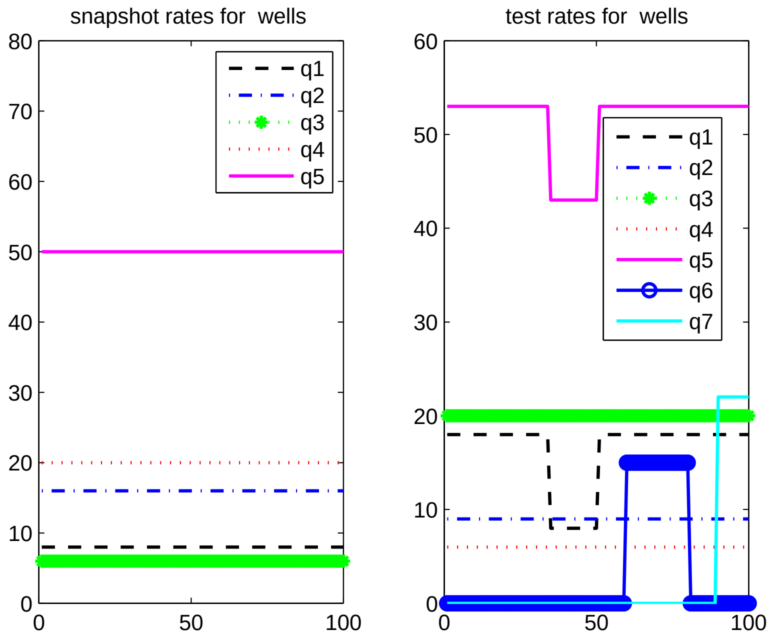

Figure 6. Injection rate schedules for training and test models.

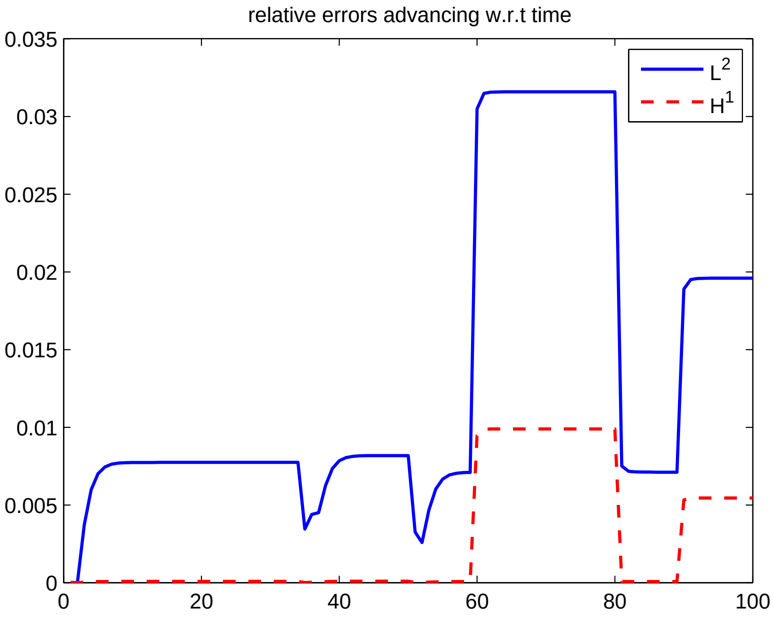

Figure 7. Relative errors advancing in time for the adaptive POD.

Table 1. Adaptive-POD-1 with 5 initial global POD basis.

Time Instant to Add	G^{off}	L^{off}	L^{add}
1	5	484	119
2	6	484	116
35	7	484	126
Average error	L^2	H^1	
	0.0135	0.0027	

Table 2. Average POD errors.

Number of POD Basis	L^2	H^1
2	0.6114	0.3254
5	0.3470	0.1676
8	0.2956	0.1378
100	0.1830	0.0796

4.2. Two-Phase Flow

4.2.1. An Incompressible Two-Phase Flow Model

In this section, we summarize the underlying partial differential equations to simulate porous media flows. In particular, we consider two-phase flow in a reservoir domain (denoted by Ω) under the assumption that the displacement is dominated by viscous effects; *i.e.*, we neglect the effects of gravity, compressibility, and capillary pressure. This assumption is a useful simplification, but the overall methodology can be applied to more general cases. The two phases are water and oil, and they are assumed to be immiscible. We write Darcy's law for each phase as follows:

$$\mathbf{u}_l = -\frac{k_{rl}(s_l)}{\mu_l} K \nabla \tilde{p} \tag{16}$$

where $\mathbf{u}_l$ is the phase velocity, K is the permeability tensor, k_{rl} is the relative permeability to phase l ($l = o, w$), s_l is saturation, and $\tilde{p}$ is pressure. Throughout the paper, we use a single set of relative permeability.

The mass conservation law for the two phases is:

$$\phi \frac{\partial s_l}{\partial t} + \nabla \cdot \mathbf{u}_l = q_l. \tag{17}$$

Combining Darcy's law, mass conservation, and the property $s_w + s_o = 1$, we derive the following coupled system of pressure and saturation equations (we use s instead of s_w for simplicity):

$$\begin{aligned}
\nabla \cdot \mathbf{u} &= q_w + q_o \quad \text{in} \quad \Omega & (18)\\
\phi \frac{\partial s}{\partial t} + \nabla \cdot (f_w(s)\mathbf{u}) &= \frac{q_w}{\rho_w} \quad \text{in} \quad \Omega & (19)\\
\mathbf{u} \cdot n &= 0 \quad \text{on} \quad \partial\Omega \quad \text{(no flow at boundary)} & (20)\\
s(t=0) &= s_0 \quad \text{in} \quad \Omega \quad \text{(initial known saturation)} & (21)
\end{aligned}$$

where ϕ is the porosity, λ is the total mobility defined as

$$\lambda(s) = \lambda_w(s) + \lambda_o(s) = \frac{k_{rw}(s)}{\mu_w} + \frac{k_{ro}(s)}{\mu_o} \tag{22}$$

$f_w(s)$ is the flux function,

$$f_w(s) = \frac{\lambda_w(s)}{\lambda(s)} = \frac{k_{rw}(s)}{k_{rw}(s) + \frac{\mu_w}{\mu_o} k_{ro}(s)} \tag{23}$$

and $\mathbf{u} = \mathbf{u}_w + \mathbf{u}_o = -\lambda(s) K \nabla \tilde{p}$ is the total flux. Moreover, q_w and q_o are volumetric source terms for water and oil.

Here, we follow the sequential formulation, that is at each time step one solves for pressure first and then uses the result to solve for the saturation. The pressure equation is solved by mixed finite element method, see [35,44] which produces mass conservative velocity field, and the saturation equation is solved by finite volume method.

4.2.2. Numerical Example

We apply the online adaptive model reduction method to an oil-water flow model with the structure of a 5-spot. A 5-spot pattern is a well configuration that involves 1 producing well (or 1 injector well) in the center and 4 injecting wells (or 4 producers) in the corners of a Cartesian grid. It can be seen as a five wells (point source) in the whole domain. It is designed to maximize the reservoir sweep as water is being injected into the reservoir. Here we set four injection wells in the corners and one production well in the middle of the domain. All the wells are rate controlled. The computational domain is $\Omega = (0,1)^2$, a uniform fine mesh with $h = 1/220$ is used. The fluid viscosity ratio is $\mu_w/\mu_o = 0.2$. To get the snapshots, we simulate the flow with end of simulation time 800, time stepping size $\Delta t = 4$, with well rates as shown in the left in Figure 8. From this forward (training) simulation, we save the snapshots of the states and the nonlinear functions at every 2 units in time. Therefore, 100 snapshots are saved for the states and nonlinear functions. Each snapshot is reshaped to a column vector and is stacked in a snapshot matrix. After applying POD to each matrix, we get the POD subspace. For the test simulation, total simulation time is 400, $\Delta t = 4$, and the well rates are given in the right of Figure 8.

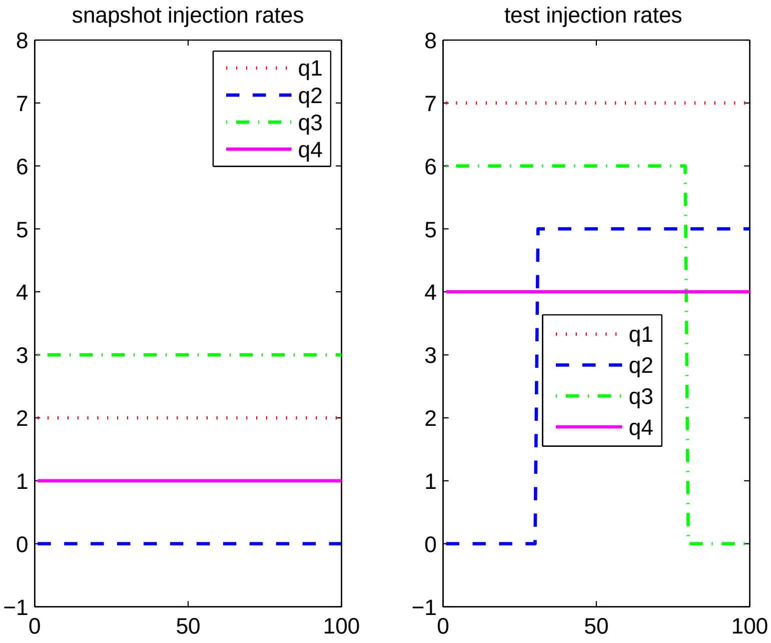

Figure 8. Injection rate schedules for training and test models.

From Figure 8 we see that the rates are constant for the training simulation, and well 2 is shut off all the time; while for the test simulation, well 2 is open at the time instant 31 and well 3 is closed from 80 to the end of simulation time. Large deviations in the rates of the training and test simulation start at three points: the beginning since different constants are used for rates; time instant 31; time instant 80. These deviations are expected to result in requirement of online adaption. We use the L^2 norm of the residual as indicators for adaption. To better understand the impact of injection rates on the whole system, we will investigate the method for the pressure and saturation equations separately first. For the adaptive DEIM application, number of sampling points is $m_s = 800$, and window size is $w = 25$.

In Case 1, we apply POD reduction to the pressure equation only, while solve the saturation equation on the fine grid. Note that only POD reduction is used, no DEIM is involved since the pressure equation is linearly dependent on the pressure and velocity. The selection criterion (6) for choosing the number of initial POD basis is to capture 99.99% of the energy of the snapshots.

In Figure 9, relative saturation errors for 3 scenarios with 5 initial POD basis for flux are given. The red line is for static POD without updating, the errors are greater than 10% all the time, and there are jumps at time instants {31,80}. The blue line is for adaptive POD with adaption at time instants {1,31}. The dot purple line is for adaptive POD with adaption at time instants {1,31,80}, which gives a good accuracy. Figure 10 presents the comparison of water-cut (water flux fractional function $f_w(s)$) corresponding to these 3 scenarios with the fine water-cut. Both figures show that adaption is necessary in order to obtain a good accuracy.

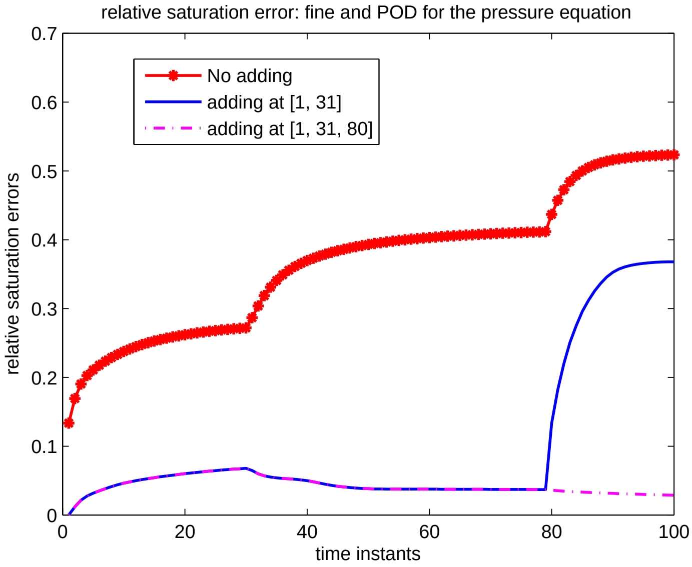

Figure 9. Relative saturation errors advancing in time: POD for the pressure equation.

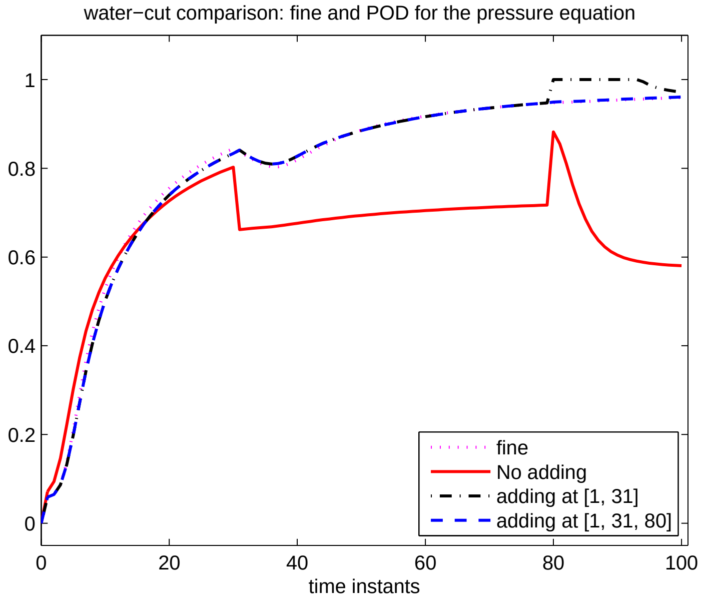

Figure 10. Water-cut of online adptive POD for the pressure equation.

In Case 2, we apply model reduction to the saturation equation only while solving the pressure equation on the fine grid. First, we consider POD without DEIM. As pointed out in Section 2, due to the nonlinearity of the saturation equation, no improvement in efficiency will be achieved if no DEIM is used. We only consider this case to compare with the case for POD-DEIM in the next example. We will see that the POD-DEIM needs more updates than POD. The selection criterion (6) for choosing the number of initial saturation POD basis is to capture 99% of the energy of the snapshots. 20 initial saturation POD basis is selected. Updates are needed at $\{1, 4, 31, 32, 33, 35, 36, 38, 40, 42, 43\}$. In Table 3, the numbers of saturation POD basis after updating for Adaptive-POD-1, Adaptive-POD-2 are presented. For example, the number of basis for Adaptive-POD-1 after updating at time instant 33 is 25, the number of basis for Adaptive-POD-2 after 4 is 16. The relative saturation errors for both Adaptive-POD-1 and Adaptive-POD-2 are given in Figure 11, we can see that Adaptive-POD-2 delivers almost the same accuracy while using much less number of basis compared with Adaptive-POD-1. The comparison of water-cut between the solutions of Adaptive-POD-1, Adaptive-POD-2 and the fine solution is shown in Figure 12, which shows a good agreement. Second, we apply POD-DEIM to the saturation equation. We use the adaptive DEIM technique as described in Section 3.2 to approximate f_w. 15 initial DEIM basis functions are selected for f_w. The comparison of water-cut between the solution of Adaptive-POD-2-DEIM, Adaptive-POD-2-DEIM and the fine solution is shown in Figure 13, they are in good agreement in general. The saturation errors for both Adaptive-POD-1-DEIM and Adaptive-POD-2-DEIM are given in Figure 14. We can see that the error of Adaptive-POD-2-DEIM is less than Adaptive-POD-1-DEIM, while the number of basis for Adaptive-POD-2 is less than Adaptive-POD-1.

Table 3. Number of POD basis for saturation after each update of POD subspace for Equation (19).

	1	**4**	**31**	**32**	**33**	**35**	**36**	**38**	**40**	**42**	**43**
Adaptive-POD-1	21	22	23	24	25	26	27	28	29	30	31
Adaptive-POD-2	22	16	8	8	7	7	7	7	7	7	7

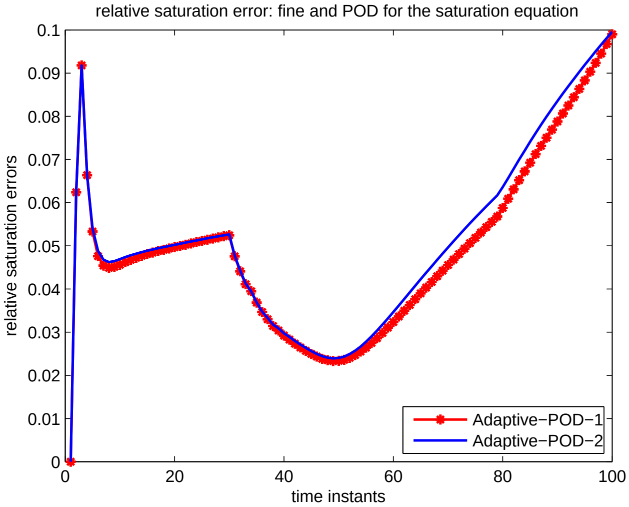

Figure 11. Relative saturation errors advancing in time: POD for the saturation equation.

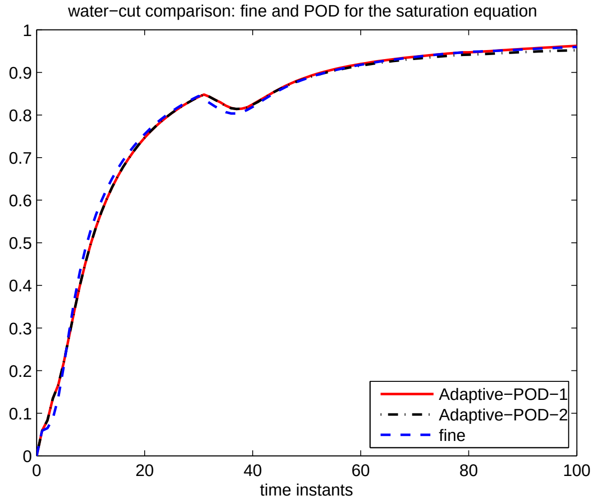

Figure 12. Water-cut of online adptive POD for the saturation equation.

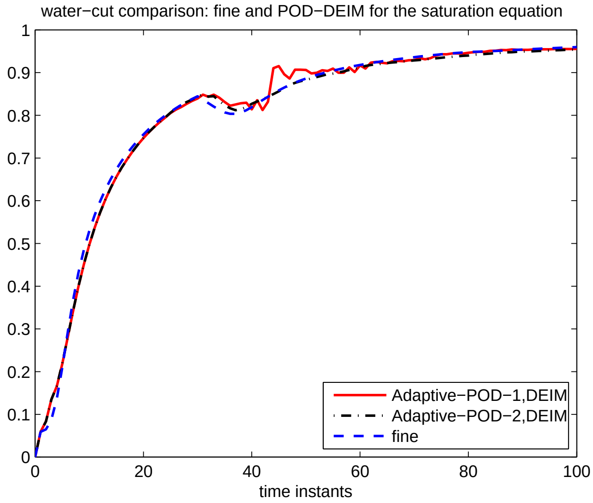

Figure 13. Water-cut of online adaptive POD-DEIM for the saturation equation.

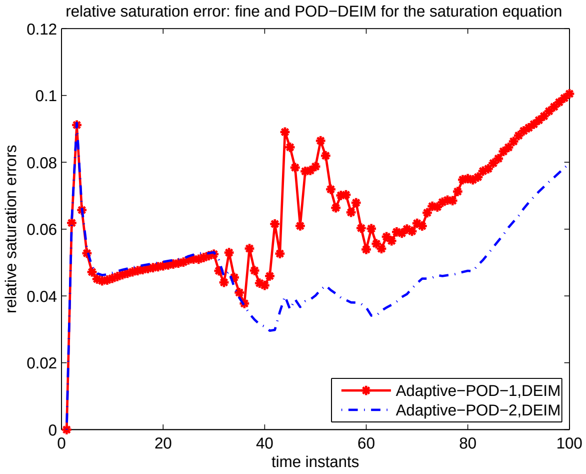

Figure 14. Relative saturation errors advancing in time: POD-DEIM for the saturation equation.

In Case 3, apply our model reduction method to the coupled system, *i.e.*, POD for the pressure equation and POD-DEIM for the saturation equation. Velocity needs updates at $\{1, 31, 36, 48, 80\}$, and saturation needs 13 updates at instants as listed in the Table 4. The numbers of saturation POD basis after updating for Adaptive-POD-1, Adaptive-POD-2 are shown in Table 4. The saturation error is given in Figure 15, from there we see that Adaptive-POD-2 has better performance than Adaptive-POD-1. The comparison of water-cut between the solution of Adaptive-POD-1, Adaptive-POD-2 and the fine solution is shown in Figure 16. We observe good agreement.

Table 4. Number of POD basis for saturation after each update of POD subspace for the coupled system.

	1	**4**	**31**	**32**	**34**	**35**	**36**	**38**	**39**	**47**	**53**	**54**	**60**
Adaptive-POD-1	21	22	23	24	25	26	27	28	29	30	31	32	33
Adaptive-POD-2	21	16	8	8	7	7	7	7	7	7	7	7	7

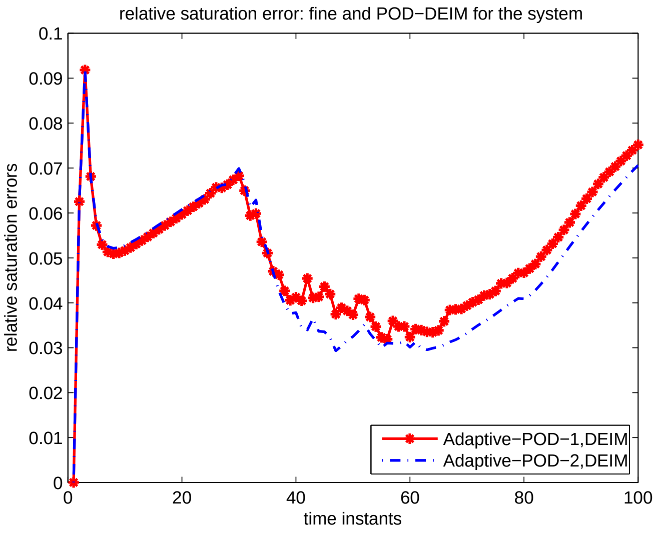

Figure 15. Relative saturation errors advancing in time: POD-DEIM for the coupled system.

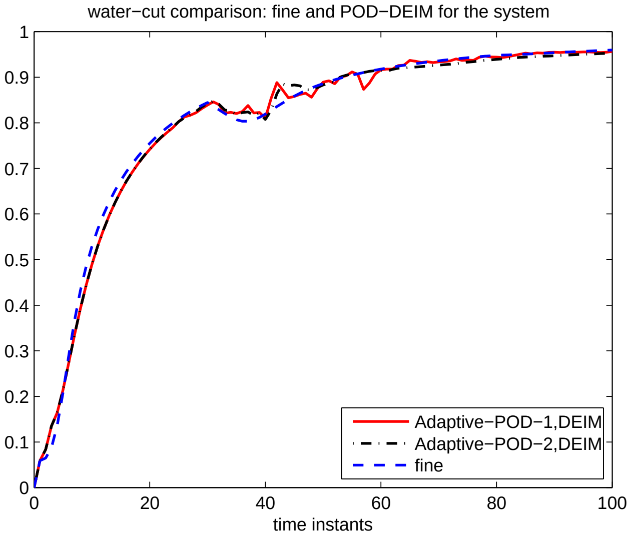

Figure 16. Water-cut of online adaptive POD-DEIM for the coupled system.

5. Conclusion

We presented an online adaptive local-global POD-DEIM model reduction method for nonlinear systems where both POD subspaces and DEIM interpolants are adapted during the online stage by incorporating online information locally. In the global adaptive framework, we employ local techniques to realize adaption leading to computational savings in evaluating the residual. Our approach is particularly useful for situations where it is desired to solve the reduced system for inputs or controls that result in a solution outside the span of the snapshots generated in the offline stage. This happens in many applications, such as inverse problem and optimization, where the final solution path may be difficult to predict before the problem is solved. Our method also offers an alternative to constructing a robust reduced system even if a potential initial poor choice of snapshots is used, since it can absorb representative information into the POD solution space along the online simulation process. Future work includes finding an efficient way to obtain the online basis for a nonlinear system.

Acknowledgments: This publication also was made possible by a National Priorities Research Program grant NPRP grant 7-1482-1278 from the Qatar National Research Fund (a member of The Qatar Foundation). Y.E. and E.G. would like to thank the partial support from the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-FG02-13ER26165 and the DoD Army ARO Project.

Author Contributions: All of the authors conceived and designed the method and experiments. Yanfang Yang performed the experiments. All of the authors contributed to the analysis of the data and writing of the tmanuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lumley, J. Atmospheric turbulence and radio wave propagation. *J. Comput. Chem.* **1967**, *23*, 1236–1243.
2. Gugercin, S.; Antoulas, A. A survey of model reduction by balanced truncation and some new results. *Int. J. Control* **2004**, *77*, 748–766.
3. Rewienski, M.; White, J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Comput.-Aided Des. Intergrated Circ. Syst.* **2003**, *22*, 155–170.
4. Hall, K.C.; Thomas, J.P.; Dowell, E.H. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA J.* **2000**, *38*, 1853–1862.
5. Kunisch, K.; Volkwein, S. Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optim. Theory Appl.* **1999**, *102*, 345–371.
6. Rowley, C.W.; Colonius, T.; Murray, R.M. Model reduction for compressible flows using POD and Galerkin projection. *Phys. D* **2004**, *189*, 115–129.
7. Bui-Thanh, T.; Damodaran, M.; Willcox, K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J.* **2004**, *42*, 1505–1516.
8. Meyer, M.; Matthies, H. Efficient model reduction in non-linear dynamics using the karhunen-loève expansion and dual-weighted-residual methods. *Comput. Mech.* **2003**, *31*, 179–191.
9. Heijn, T.; Markovinovic, R.; Jansen, J. Generation of low-order reservoir models using system-theoretical concepts. *SPE J.* **2004**, *9*, 202–218.
10. Cardoso, M.; Durlofsky, L.; Sarma, P. Development and application of reduced-order modeling procedures for subsurface flow simulation. *Int. J. Numer. Method. Eng.* **2009**, *77*, 1322–1350.
11. Ghasemi, M.; Yang, Y.; Gildin, E.; Efendiev, Y.; Calo, V. Fast multiscale reservoir simulations using POD-DEIM model reduction. In Proceedings of the SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Houston, TX, USA, 23–25 February 2015; doi:10.2118/173271-MS.
12. Haasdonk, B.; Ohlberger, M. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In *Finite Volumes for Complex Applications V*; Wiley-ISTE: Hoboken, NJ, USA; London, UK, 2008; pp. 471–479.
13. Peherstorfer, B.; Butnaru, D.; Willcox, K.; Bungartz, H.-J. Localized discrete empirical interpolation method. *SIAM J. Sci. Comput.* **2014**, *36*, A168–A192.

14. Maday, Y.; Stamm, B. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J. Sci. Comput.* **2013**, *35*, A2417–A2411.
15. Amsallem, D.; Farhat, C. An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.* **2011**, *33*, 2169–2198.
16. Chaturantabut, S.; Sorensen, D.C. Discrete empirical interpolation for nonlinear model reduction. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764.
17. Efendiev, Y.; Galvis, J.; Hou, T.Y. Generalized multiscale finite element methods (gmsfem). *J. Comput. Phys.* **2013**, *251*, 116–135.
18. Efendiev, Y.; Hou, T.Y. *Multiscale Finite Element Methods: Theory And Applications*; Springer Science & Business Media: New York, NY, USA, 2009; Volume 4.
19. Chung, E.; Efendiev, Y.; Hou, T.Y. Adaptive multiscale model reduction with generalized multiscale finite element methods. *J. Comput. Phys.* **2016**, doi:10.1016/j.jcp.2016.04.054.
20. Efendiev, Y.; Galvis, J.; Wu, X.-H. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *J. Comput. Phys.* **2011**, *230*, 937–955.
21. Chung, E.T.; Efendiev, Y.; Li, G. An adaptive gmsfem for high-contrast flow problems. *J. Comput. Phys.* **2014**, *273*, 54–76.
22. Chung, E.T.; Efendiev, Y.; Leung, W.T. Residual-driven online generalized multiscale finite element methods. *J. Comput. Phys.* **2015**, *302*, 176–190.
23. Chung, E.T.; Leung, W.T.; Pollock, S. Goal-oriented adaptivity for gmsfem. *J. Comput. Appl. Math.* **2016**, *296*, 625–637.
24. Chan, H.Y.; Chung, E.T.; Efendiev, Y. Adaptive mixed gmsfem for flows in heterogeneous media. *arXiv preprint* **2015**, arXiv:1507.01659.
25. Peherstorfer, B.; Willcox, K. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM J. Sci. Comput.* **2015**, *37*, 2123–2150.
26. Alotaibi, M.; Calo, V.M.; Efendiev, Y.; Galvis, J.; Ghommem, M. Global–local nonlinear model reduction for flows in heterogeneous porous media. *Comput. Method. Appl. Mech. Eng.* **2015**, *292*, 122–137.
27. Efendiev, Y.; Galvis, J.; Gildin, E. Local–global multiscale model reduction for flows in high-contrast heterogeneous media. *J. Comput. Phys.* **2012**, *231*, 8100–8113.
28. Durlofsky, L.; Efendiev, Y.; Ginting, V. An adaptive local–global multiscale finite volume element method for two-phase flow simulations. *Adv. Water Resour.* **2007**, *30*, 576–588.
29. Presho, M.; Ye, S. Reduced-order multiscale modeling of nonlinear p-laplacian flows in high-contrast media. *Comput. Geosci.* **2015**, *19*, 921–932.
30. Vermeulen, P.; Heemink, A.; Stroet, C. Reduced models for linear groundwater flow models using empirical orthogonal functions. *Adv. Water Resour.* **2004**, *27*, 57–69.
31. Van Doren, J.; Markovinovic, R.; Jansen, J. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comput. Geosci.* **2006**, *10*, 137–158.
32. Cardoso, M.; Durlofsky, L. Linearized reduced-order models for subsurface flow simulation. *J. Comput. Phys.* **2010**, *229*, 681–700.
33. Efendiev, Y.; Galivs, J.; Gildin, E. Local-global multiscale model reduction for flows in high-contrast heterogeneous media. *J. Comput. Phys.* **2012**, *231*, 8100–8113.
34. He, J.; Durlofsky, L.J. Constraint reduction procedures for reduced-order subsurface flow models based on POD-TPWL. *Int. J. Numer. Meth. Eng.* **2015**, doi:10.1002/nme.4874.
35. Yang, Y.; Ghasemi, M.; Gildin, E.; Efendiev, Y.; Calo, V. Fast multi-scale reservoir simulations using POD-DEIM model reduction. In Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, USA, 23–25 February 2015; doi:10.2118/173271-MS.
36. Kunisch, K.; Volkwein, S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.* **2002**, *40*, 492–515.
37. Volkwein, S.; Hinze, M. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In *Reduction of Large-Scale Systems*; Benner, P., Mehrmann, V., Sorensen, D.C., Eds.; Lecture Notes in Computational Science and Engineering, Vol. 45; Elsevier: Amsterdam, The Netherlands, 2005; pp. 261–306.
38. Calo, V.M.; Efendiev, Y.; Galvis, J.; Li, G. Randomized oversampling for Generalized Multiscale Finite Element Methods. *Multiscale Model. Simul.* **2016**, *14*, 482–501, doi:10.1137/140988826.

39. Chung, E.T.; Efendiev, Y.; Lee, C.S. Mixed generalized multiscale finite element methods and applications. *Multiscale Model. Simul.* **2015**, *13*, 338–366.
40. Chung, E.T.; Efendiev, Y.; Leung, T. Residual-driven online generalized multiscale finite element methods. *J. Comput. Phys.* **2015**, *302*, 176–190.
41. Bergmann, M.; Bruneau, C.H.; Iollo, A. Enablers for robust POD models. *J. Comput. Phys.* **2009**, *228*, 516–538.
42. Rapún, M.L.; Vega, J.M. Reduced order models based on local POD plus Galerkin projection. *J. Comput. Phys.* **2010**, *229*, 3046–3063.
43. Rapún, M.L.; Terragni, F.; Vega, J.M. Adaptive POD-based low-dimensional modeling supported by residual estimates. *Int. J. Numer. Meth. Eng.* **2015**, doi:10.1002/nme.4947.
44. Boffi, D.; Brezzi, F.; Fortin, M. *Mixed Finite Element Methods and Applications*; Springer-Verlag Berlin-Heidelberg: Heidelberg, Germany, 2013.