

Localized Model Reduction in Porous Media Flow^{*}

Mohamadreza Ghasemi^{*} Eduardo Gildin^{**}

^{*} *Petroleum Engineering Department, Texas A&M University, College Station, USA, (e-mail: torezaghasemi@gmail.com)*

^{**} *Petroleum Engineering Department, Texas A&M University, College Station, USA, (e-mail: eduardo.gildin@pe.tamu.edu)*

Abstract: This paper introduces a new localized approach to construct an efficient reduced order model for fluid flow simulation and optimization in porous media flow. For nonlinear systems, one of the most common methodology used is the proper orthogonal decomposition (POD) combined with discrete empirical interpolation method (DEIM) due to its computational efficiency and good approximation. Whereas regular POD-DEIM approximates the fine scale model with just one single reduced subspace, the localized POD (LPOD) and localized DEIM (LDEIM) that are introduced in this work compute several local subspaces. Each subspace characterize a region of solutions and all together they not only can approximate the high fidelity model better, but also reduced the computational cost of simulation. LPOD and LDEIM use classification approach to find these regions in the offline computational phase. After obtaining each class, POD and DEIM is applied to construct the basis of the reduced space. In the online phase, at each time step, the reduced states and functions will be used to find the most representative basis for POD and DEIM without requiring fine scale information. The advantages of LPOD and LDEIM are shown in a numerical example of two phase flow in porous media.

Keywords: model order reduction, localized POD, localized DEIM, classification, fluid dynamics, reservoir simulation, production optimization.

1. INTRODUCTION

Reservoir management enables one to obtain the most favorable production scenario given the current information of the reservoir, such as reservoir characterization parameters (permeability, porosity) and production data. Very often, one is directed to computational methods (reservoir simulation) to assist on the search for the optimal control strategy by sweeping all the spectrum of reservoir uncertainties. This would be true except for two main challenges: the large amount of computational infrastructure necessary to perform such calculations; and the fact that there is no certainty in the range of parameters. In this paper our primary focus is to tackle the first problem by means of reduced order models. Note that the latter issue can be investigated by uncertainty quantification and parameter estimation.

In many cases, reduced-order modeling techniques have shown to be a viable way of mitigating computational complexity in simulation of large-scale reservoirs, while maintaining high level of accuracy. The options range from non-intrusive methods that do not depend on modifications of a reservoir simulation code Cardoso and Durlofsky (2010); Ghommem et al. (2015), to a more intrusive and sophisticated methods that depend on several modifications of legacy code or the development of new simulator

codes, see Heijn et al. (2004); Gildin et al. (2013); Gildin and Ghasemi (2014)

POD is one of the most efficient methodology used in model order reduction context due to its computational simplicity and good approximation (Van Doren et al., 2004). However, computational savings are not always attained because in order to evaluate nonlinear terms, the reduced state needs to be projected back to fine scale state yielding similar computational cost as the original system. There are different techniques to alleviate this problem. One approach is to linearize these nonlinear functions (trajectories) around several known states and use these piecewise linear solutions, see Cardoso and Durlofsky (2010). Here, we use DEIM, where one construct another subspace for reducing the nonlinear function evaluations (Chaturantabut and Sorensen, 2010). In reservoir simulation and optimization, these methodologies have shown to yield reduction in the size of the approximated models of several orders of magnitude (Ghasemi et al., 2015). However, if the system exhibits a very dynamic state with a wide range of changes, many POD and DEIM basis are required to accurately approximate the state of the system as well as nonlinear terms. A remedy to this problem is to search for not just a single global subspace, but rather multiple local subspaces and basis to capture nonlinear dynamics. This process is called localization in this work.

With localization, the dimension of the reduced space can be decreased more and consequently the computational

^{*} The authors would like to thanks the US DoD - Army Research Office and Foundation CMG for supporting this project.

cost is reduced even further. In this technique rather than just one single subspace, multiple local subspaces are constructed for the reduced system. In the online phase, based on the state of system at each time step, the proper subspace that has the best approximation will be selected.

The localization idea was introduced for POD in (Am-sallem et al., 2012) and was verified in reducing the pressure state in dynamical equations describing a MEMS switch device. They suggested using auxiliary quantities to ensure an online selection procedure is independent of the dimension of the original system. However, the number of these auxiliary quantities, which are computed in the offline phase, scales cubically with the number of clusters. This can be an issue especially in a optimization workflow, whereby one updates basis of reduced model periodically to obtain more stable and accurate solution. Following localized POD (LPOD), localized DEIM (LDEIM) was proposed in (Peherstorfer et al., 2014) based on machine learning techniques and using efficient classification algorithm. They discussed both parameter and state based LDEIM approaches and applied it to a steady reacting flow simulation.

In this paper, we extend LPOD and LDEIM to reduce the computational cost of porous media fluid flow simulation. We present a new approach to resolve the existing issues in LPOD by reducing the number of required auxiliary parameters. Also, an efficient reduced order model for the fluid transport equation is presented by applying LDEIM on the nonlinear function of fractional flow.

2. TWO-PHASE FLOW MODEL

In this section, we summarize the underlying partial differential equations related to porous media flow simulation. In particular, we briefly discuss two-phase oil-water systems, see Aarnes et al. (2009) for more details.

We consider two-phase flow in a reservoir domain under the assumption that the displacement is dominated by viscous effects; i.e., we neglect the effects of gravity, compressibility, and capillary pressure. The two phases are water and oil, and they are assumed to be immiscible. The Darcy's law for each phase is as follows,

$$v_l = -\frac{k_{rl}(s)}{\mu_l} K \nabla p, \quad (1)$$

where v_l is the phase velocity, K is the permeability tensor, k_{rl} is the relative permeability to phase l ($l = o, w$), s is the water saturation (we use s instead of s_w for simplicity) and p is pressure. Throughout the paper, we will assume that a single set of relative permeability curves is used. Combining Darcy's law with conservation of mass allows us to express the governing equations in terms of the so-called pressure and saturation equations as,

$$-\nabla \cdot (\lambda(s) K \nabla p) = q_w + q_o, \quad (2)$$

$$\phi \frac{\partial s}{\partial t} + \nabla \cdot (f_w(s) v) = \frac{q_w}{\rho_w}, \quad (3)$$

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

$$\lambda(s) = \lambda_w(s) + \lambda_o(s) = \frac{k_{rw}(s)}{\mu_w} + \frac{k_{ro}(s)}{\mu_o}, \quad (4)$$

$f_w(s)$ is the fractional flow function,

$$f_w(s) = \frac{\lambda_w(s)}{\lambda(s)} = \frac{k_{rw}(s)}{k_{rw}(s) + \frac{\mu_w}{\mu_o} k_{ro}(s)}, \quad (5)$$

and v is the total velocity defined as,

$$v = v_w + v_o \quad (6)$$

In this paper, we follow a sequential formulation; at each time step one solves for pressure and flux first and then use these results to solve for saturation. We employ mixed finite element methods to discretize the pressure equation in order to preserve the conservative velocity field, see (Ghasemi et al., 2015). One can write the pressure equation after spatial discretization of the problem as the following system of equations

$$\begin{pmatrix} B(\lambda(s)) & -C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} v \\ p \end{pmatrix} = \begin{pmatrix} 0 \\ g \end{pmatrix} \quad (7)$$

where matrix B and C are derived from FEM discretization and g results from the sink/source terms.

Generally, an implicit time (backward Euler) discretization will be followed to solve for saturation profile, while a mass conservative finite volume is used for the spatial derivative discretization. Consider a cell Ω_i with edges γ_{ij} and associated normal vectors n_{ij} pointing out of Ω_i , the saturation Eq. (3) will be discretized as,

$$\frac{\phi_i}{\Delta t} (s_i^k - s_i^{k-1}) + \frac{1}{|\Omega_i|} \sum_{j \neq i} F_{ij} (s_i^k) = \frac{q_w (s_i^k)}{\rho} \quad (8)$$

where s_i^k is the cell-average of the water saturation at time t_k , and F_{ij} is the numerical approximation of the flux over edge γ_{ij} ,

$$F_{ij} \approx \int_{\gamma_{ij}} f_w(s)_{ij} [v_{ij}(s)] \cdot n_{ij} dv \quad (9)$$

Note that to find $f_w(s)_{ij}$ over each edge, upwinding method is usually applied as discussed in Aarnes et al. (2009).

3. MODEL REDUCTION

In the subsequent analysis, we propose different approaches to reduce the computational complexity associated with simulation of two-phase flows. Throughout the paper, we refer to fully-resolved solution as the high fidelity results. The main approach in this paper is to use POD-Galerkin method to project the fine scale states to reduced subspace and solve for fewer number of unknowns.

3.1 Proper Orthogonal Decomposition (POD)

As already stated, POD is typically performed on the premise that the state solutions belong to a subspace with dimension much smaller than a large scale model. Assume the solution states are stored at each time step as,

$$\mathbb{S}_{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{ns}] \in \mathbb{R}^{n \times ns} \quad (10)$$

where n is the total number of grid-blocks, ns is the number of snapshots, \mathbf{x}_i can be a pressure, saturation, or nonlinear function snapshot that is vectorized to be stacked in the snapshots matrix \mathbb{S} . These snapshots are assembled into different matrices, \mathbb{S}_p , \mathbb{S}_v , \mathbb{S}_s respectively. After applying a singular value decomposition on these

matrices, we select the projection matrix, Φ , by indicating the fraction of total energy to be captured (Volkwein and Kunisch, 2008).

After obtaining the projection matrix, the pressure, velocity and saturation are re-parametrized as,

$$p(t) = \Phi_p p_r(t), \quad v(t) = \Phi_v v_r(t), \quad s(t) = \Phi_s s_r(t) \quad (11)$$

The pressure and saturation equations are now solved in the reduced space by substituting variables of (11) into (7) and (8). The reduced pressure equation is,

$$\begin{pmatrix} \Phi_v^T B \Phi_v - \Phi_v^T C^T \Phi_p \\ \Phi_p^T C \Phi_v & 0 \end{pmatrix} \begin{pmatrix} v_r \\ p_r \end{pmatrix} = \begin{pmatrix} 0 \\ \Phi_p^T g \end{pmatrix} \quad (12)$$

Thus, the original system with (number of edges + number of grid blocks) degrees of freedom is reduced to a dynamical system with few basis for velocity and pressure combined.

Furthermore, the nonlinear saturation equation is solved iteratively in reduced space as follows,

$$s_r^k = s_r^{k-1} + \Phi_s^T (A(v^k) f(\Phi_s s_r^k) + q^+) \quad (13)$$

Note that due to the presence of a nonlinear function f in (13), at each iteration the reduced saturation should be projected back to fine scale nonlinear function evaluation. This issue increases computational cost, but can be mitigated by applying DEIM, as it is explained in the next section.

3.2 Discrete empirical interpolation method (DEIM)

We briefly review the DEIM as presented in Chaturantabut and Sorensen (2010). Let $f(\tau) \in \mathbb{R}^n$ denotes a nonlinear function where τ refers to time or a parameter.

We approximate the function f by projecting it into a subspace spanned by the basis functions $\Psi = (\psi_1, \dots, \psi_m) \in \mathbb{R}^{n \times m}$ as

$$f(\tau) \approx \Psi c(\tau) \quad (14)$$

The projection basis Ψ is determined very similar to the POD basis for states. The function evaluations is assembled in a matrix $\mathbb{S}_f \in \mathbb{R}^{n \times n_s}$ and *svd* is employed to compute the m modes. These modes are used as the projection basis in the approximation given by (14).

To compute the coefficient vector c , we consider a matrix

$$P = [e_{\varphi_1}, \dots, e_{\varphi_m}] \in \mathbb{R}^{n \times m}$$

where $e_{\varphi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$ is the φ_i^{th} column of the identity matrix $I_n \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, m$. Multiplying Equation (14) by P^T and assuming that the matrix $P^T \Psi$ is nonsingular, we obtain

$$f(\tau) \approx \Psi c(\tau) = \Psi (P^T \Psi)^{-1} P^T f(\tau) \quad (15)$$

As for the interpolation indices $\{\varphi_1, \dots, \varphi_m\}$, they are selected using greedy algorithm as given in Chaturantabut and Sorensen (2010) to have a nonsingular matrix.

Although POD-DEIM will reduce the computational runtime, for some problems a large number of basis and interpolation points are required to reduce the error and have a reliable approximation. To improve basis selection, one can compute several local subspaces with similar dynamical features and find their corresponding basis. We will elaborate more on this idea in the next section.

3.3 Localized POD

The main idea is to divide the snapshots into different subgroups and apply POD to each domain to obtain local POD basis. This will require less number of basis for each region compared to global POD basis. Also, it will give us a better approximation of the solution trajectory in each subdomain, because more representative basis will be used. Two main questions are how to cluster the snapshots into subsets based on a feature and how to efficiently select the proper set in the online phase.

There are different techniques to split the solution snapshots space into different regions. In most of these methods, the domain is split into subdomains recursively (Drohmann et al., 2011), but this method might results in a large number of subdomains in practice. Here, we classify the snapshots into different clusters with machine learning techniques and for each cluster a local reduced-order model is constructed as suggested in (Peherstorfer et al., 2014). A local reduced-order model is then selected with respect to the current state of the system.

This localization approach was introduced for POD method in (Amsallem et al., 2012). However, they proposed unsupervised learning methods that can results in unstable clustering behavior if the clustering method and its parameters are not carefully chosen (Von Luxburg, 2007). Furthermore, the given procedure in Amsallem et al. (2012) requires precomputing auxiliary quantities to ensure an online selection procedure is independent of the large scale system; however, the number of auxiliary quantities scales cubically with the number of clusters. This can be an issue especially in optimization workflows, where one updates basis of reduced model periodically to obtain more stable and accurate solution. Thus, in this work we propose a modified algorithm for localized POD that only uses few number of indexes for classification in the offline phase and also using these few indexes in the online phase to find the corresponding cluster. This also reduces the storage required for the auxiliary parameters.

In the offline (preprocessing) phase of this approach, Algorithm 1 is applied to cluster the snapshot matrix. Here LPOD is used only in saturation equation but it will be extended to pressure equation in future work. The number of clusters k and the number of POD basis in each cluster r are also provided for this algorithm. The subscript s refers to saturation.

In the first step of this algorithm *deim* function is used in order to select grid points that are most representative in the solution space based on the greedy algorithm. At Line 2 and 3 of this algorithm, a small sets of indices, instead of fine scale dimension, will be used for clustering in the *k-means* algorithm. This not only accelerates clustering, but also helps finding the proper set of basis in the online phase to become independent of fine scale solution. Also, the auxiliary variables are smaller and need less storage. It should be clear that this will not introduce any extra error in the LPOD, if enough indices are selected based on the greedy algorithm, because the selected grid points are only used for clustering and classification.

The output of the *k-means* algorithm is the centroid of each cluster and the corresponding index. These centroids

and the indecies are used to define a classifier function as explained in Line 4. This classifier will be used in the online phase to distinguish the proper index of a cluster where the current state belongs to. After clustering all the snapshots one can find the POD basis for each group as explained in Lines 5-8. Note that in the Line 7 of this algorithm again a few rows of these basis are selected to be used in the classifier in the online phase. The POD basis for each group, the classifier and the small subset of these basis are return as the output of this algorithm.

Algorithm 1 Classification procedure of LPOD

```

1: procedure LPOD( $\mathbb{S}_s, r, k$ )
2:    $P_g = \text{deim}(\mathbb{S}_s)$ 
3:    $(\text{centroids}, \{S_1, \dots, S_k\}) \leftarrow k\text{means}(\mathbb{S}_s^T P_g, k)$ 
4:    $c^s(x) = \text{knnclassify}(x, \text{centroids}, 1 : k)$ 
5:   for  $i = 1 : k$  do
6:      $\Phi_i \leftarrow \text{svd}(S_i, r)$ 
7:      $w_i^s \leftarrow P_g^T \Phi_i$ 
8:   end for
9:   return  $c^s(x), \Phi_i, w_i^s \quad i = 1, \dots, k$ 
10: end procedure

```

If LPOD is applied to the saturation equation, in the online phase at each Newton-Raphson iteration, the index of a cluster that current state is most likely belongs to can be found as follows,

$$i \leftarrow c^s(w_i^s s_r) \quad (16)$$

where $c^s(\cdot)$ and w^s are the output of the Algorithm 1, and s_r is the reduced saturation state. After obtaining the cluster index, one can choose the corresponding POD basis and solve the problem in proper reduced subspace.

3.4 Localized DEIM

In this section we discuss the general description of LDEIM and an efficient method to select each subspace in the online phase without fine scale calculation. LDEIM constructs several local DEIM subspaces as, $(\Psi_1, P_1), \dots, (\Psi_k, P_k)$ in the offline phase and then select the best one in the online phase. We follow the approach suggested in Peherstorfer et al. (2014) regarding clustering the snapshots and selecting the best basis in the online phase as explained in Algorithm 2. The inputs are a snapshots matrix of nonlinear functions \mathbb{S}_f , the number of clusters k , and the number of DEIM basis in each cluster m . At the first step a subset of all snapshots, which consists a few gridblocks is selected to be used in k -means clusterings as shown in Line 2 and 3 of this algorithm. These steps, similar to LPOD, accelerate the clustering and also make the online classification independent of fine scale calculations.

After clustering all the snapshots one can find the DEIM basis for each group as explained in Lines 5-9. Note that in the Line 8 of this algorithm a small subset of this basis are selected to be used in the classifier in the online phase Peherstorfer et al. (2014). The DEIM basis for each group, the classifier and the small subset of these basis are return from this algorithm.

If LDEIM is applied to the saturation equation, in the online phase at each Newton-Raphson iteration, the index

Algorithm 2 LDEIM Procedure Peherstorfer et al. (2014)

```

1: procedure LDEIM( $\mathbb{S}_f, m, k$ )
2:    $P_g = \text{deim}(\mathbb{S}_f)$ 
3:    $(\text{centroids}, \{S_1, \dots, S_k\}) \leftarrow k\text{means}(\mathbb{S}_f^T P_g, k)$ 
4:    $c^f(x) = \text{knnclassify}(x, \text{centroids}, 1 : k)$ 
5:   for  $i = 1 : k$  do
6:      $\Psi_i \leftarrow \text{svd}(S_i, m)$ 
7:      $P_i \leftarrow \text{deim}(\Psi_i, m)$ 
8:      $w_i^f \leftarrow P_g^T \Psi_i (P_i^T \Psi_i)^{-1}$ 
9:   end for
10:  return  $c^f(x), \Psi_i, w_i^f \quad i = 1, \dots, k$ 
11: end procedure

```

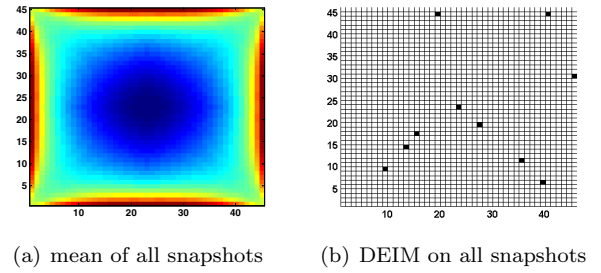


Fig. 1. Illustration of global DEIM on all snapshots and selected grid blocks in a homogeneous model

of a cluster that current state is most likely belongs to can be found as follows,

$$i \leftarrow c^f(w_i^f \tilde{f}) \quad (17)$$

where $c^f(\cdot)$ and w^f are the output of the Algorithm 2, and \tilde{f} is the approximated nonlinear function based on DEIM approach. Here we select following point-based feature extraction,

$$\tilde{f} = f(P_i^T s) \in \mathbb{R}^m \quad (18)$$

After obtaining the cluster index, one can choose the corresponding DEIM basis and solve the problem in the reduced subspace.

Here we compare LDEIM and a regular DEIM on 100 snapshots of fractional flow that were obtained by running a 45×45 homogeneous model with 5-spot pattern. Primarily, regular DEIM was applied to all the snapshots to obtain 10 basis and select the corresponding grid blocks. The mean of all the snapshots and the selected grid blocks from regular DEIM is shown in Fig. 1(b).

These snapshots were also used in Algorithm 2 for classifying 3 clusters and 10 basis for each cluster. The mean of each cluster and the selected grid blocks of the reservoir model based on are shown in Fig. 2. Comparing the global selected grid blocks and the local ones in each cluster reveals that the clustering allows LDEIM to concentrate the interpolation points in only a certain parts of the reservoir, i.e. close to the front of water saturation. This is one of the reasons that LDEIM can give us better results compared to regular DEIM, because the selected points are more into dynamic part of the model.

One can do similar analysis for LPOD and compares the resulted basis with regular POD. However, we skip it for the sake of space limitation.

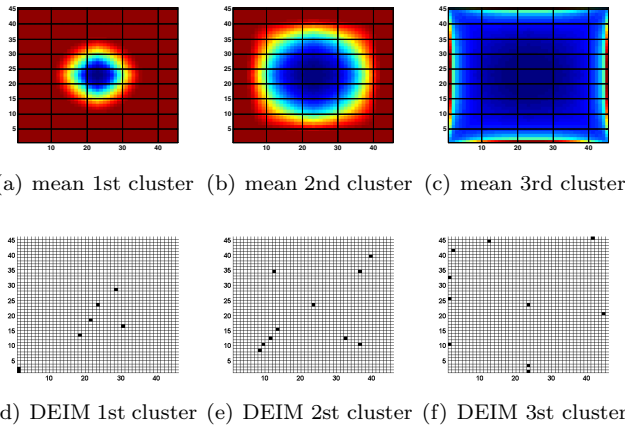


Fig. 2. Illustration of localized DEIM on each cluster and selected grid blocks in a homogeneous model

4. NUMERICAL EXAMPLE

In this example the localized model reduction is applied to a two-phase flow (oil-water) reservoir model under the water flooding recovery process with the structure of a 5-spot as shown in Fig. 3(a). The permeability of the reservoir is taken from SPE10 comparative model (Christie and Blunt, 2001) (layer 10th). The reservoir model is discretized using Cartesian grid of size $20\text{ ft} \times 10\text{ ft} \times 2\text{ ft}$. Overall the reservoir model has $60 \times 220 \times 1 = 13200$ active cells. The fluid viscosity ratio is $\mu_w/\mu_o = 0.1$. The relative permeability curves is depicted in Fig. 3(b). We assumed a constant porosity of 0.2 for entire model.

The producers are controlled by bottom hole pressure and the injector by rate. The input schedule is changed every 200 days as shown in Figs. 4. This figure includes both the training and the test schedule. Note that injection volume is at least one pore volume throughout simulation time (1000 days). The initial water saturation and pressure are assumed to be 0.0 and 2500 psia, respectively.

In order to apply the POD-DEIM methods, the reservoir is simulated with training schedule for 1000 days and the snapshots of pressures, velocity, water saturations and the nonlinear fractional function are saved every 10 days. Thus, we have 100 snapshots for each variables. One can find the basis and construct the reduced model as explained in previous sections. Next, the reduced model is verified with the test schedule.

Regular (global) POD is applied to velocity and pressure, and POD/LPOD and DEIM/LDEIM to saturation equation as it is a nonlinear equation. The selection criteria for pressure and velocity basis was to capture at least 99% of the energy of snapshots. Throughout this example LPOD and POD as well as DEIM and LDEIM are compared in the saturation equation.

The number of basis is compared between different reduced order models and the original fine scale one in Table 1. The snapshots were classified into 4 clusters. The overall speedup is around 5.5 because this is still relatively small model. Also, even though localized method has more overhead during the online phase, it does not have significantly higher computational time compared to regular POD-DEIM.

Table 1. Compare fine and reduced model.

Num # of	Fine Scale	POD/LPOD	DEIM/LDEIM
pressure basis	13200	3	–
velocity basis	26120	12	–
saturation basis	13200	7/7	–
fractional basis	13200	–	9/9
iterations	712	260/281	320/335
Total elapsed time	108.7 (s)	19.6/20.2 (s)	18.7/19.2 (s)

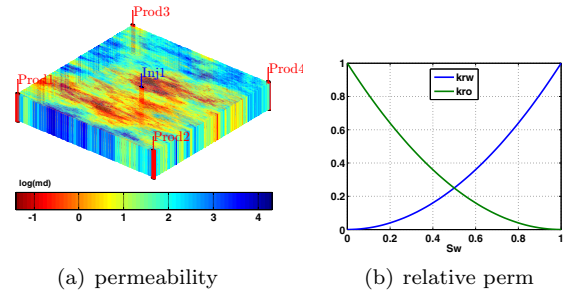


Fig. 3. Layer 10th of SPE10

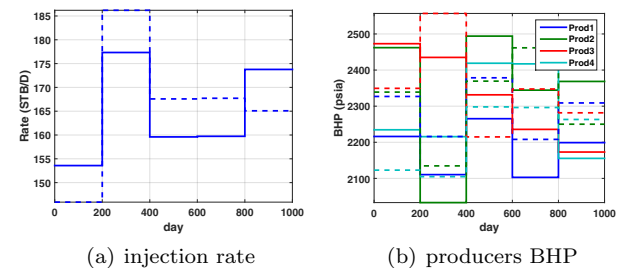


Fig. 4. Training schedule (solid line) and Test schedule (dashed line) with $\pm 5\%$ variation

Note that, when we apply DEIM or LDEIM, the saturation assumed to be projected to reduced subspace by regular (global) POD basis. Combining LPOD-LDEIM will be considered in future work.

The final water saturation and the water cut at all the producers are shown in Figs. 5(a) and 5(b) for POD and LPOD, respectively. Note that the water cut results from high fidelity model is shown as dashed line in these figures. As illustrated in these figures the LPOD approximates the original fine scale model much better, with almost exact water cut. Figs. 7(a) and 7(a) compare the same results for DEIM and LDEIM, whereas LDEIM suppresses regular DEIM. The final water saturation error is also compared between POD and LPOD in Fig. 6 and between DEIM and LDEIM in Fig. 8. The temporal saturation error is compared for POD and LPOD in Fig. 9(a) and for DEIM and LDEIM in Fig. 9(b).

All of these results confirm the superiority of local model reduction over global one.

5. CONCLUSION

A localized POD and DEIM model reduction scheme has been introduced for the solution of the two-phase flow in heterogeneous porous media. It has been our experience that performing localization with multiple snapshot yield a more accurate and stable reduced order models than a

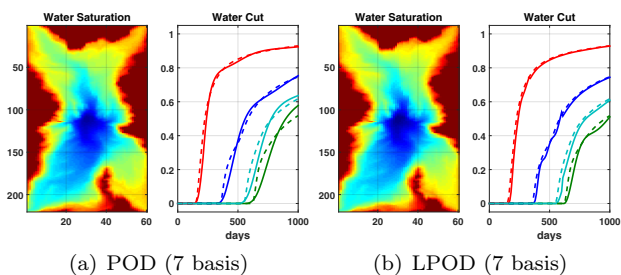


Fig. 5. Final water saturation and water cut at producers

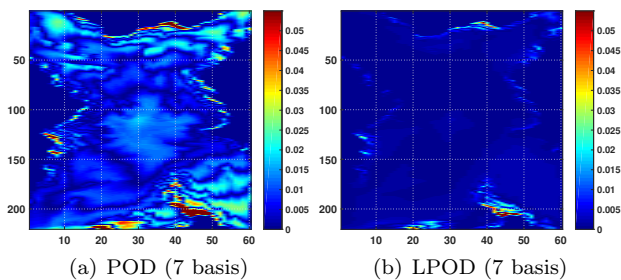


Fig. 6. Final water saturation error

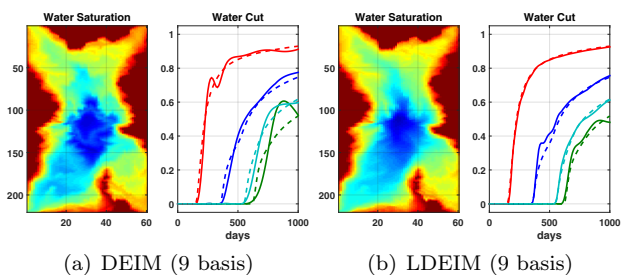


Fig. 7. Final water saturation and water cut at producers

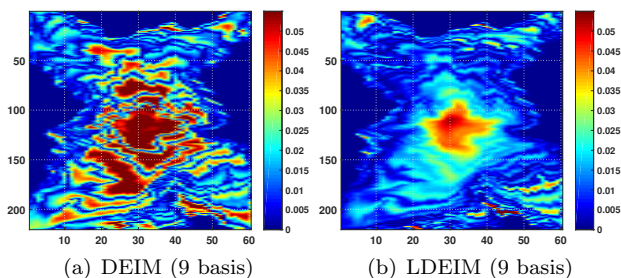


Fig. 8. Final water saturation error

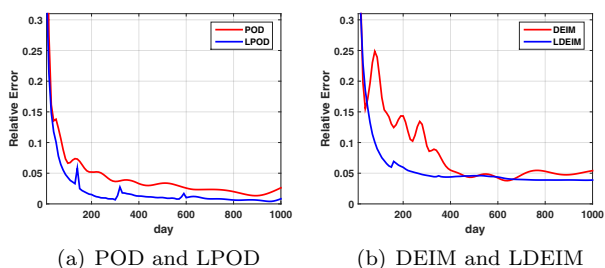


Fig. 9. Temporal saturation error

single reduced subspace. More work still need to be done in selecting the appropriate number of clusters and the specific clustering technique used. Moreover, applications of the proposed method in reservoir production optimization should be investigated.

REFERENCES

Aarnes, J.E., Lie, K.A., Kippe, V., and Krogstad, S. (2009). Multiscale methods for subsurface flow. In *Multiscale Modeling and Simulation in Science*, volume 66, 3–48. Springer Verlag.

Amsallem, D., Zahr, M.J., and Farhat, C. (2012). Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10), 891–916.

Cardoso, M. and Durlofsky, L. (2010). Use of reduced-order modeling procedures for production optimization. *SPE Journal*, 15(2), 426–435.

Chaturantabut, S. and Sorensen, D.C. (2010). Discrete empirical interpolation for nonlinear model reduction. *SIAM J. Sci. Comput.*, 32(5), 2737–2764.

Christie, M. and Blunt, M. (2001). Tenth spe comparative solution project: a comparison of upscaling techniques. spe-72469. *SPEE*, 4, 308–317.

Drohmann, M., Haasdonk, B., and Ohlberger, M. (2011). Adaptive reduced basis methods for nonlinear convection–diffusion equations. In *Finite Volumes for Complex Applications VI Problems & Perspectives*, 369–377. Springer.

Ghasemi, M., Gildin, E., Yang, Y., Efendiev, Y., and Calo, V. (2015). Fast multiscale reservoir simulations using pod-deim model reduction. In *SPE Reservoir Simulation Symposium*. Houston, Texas. SPE 173271.

Ghommem, M., Gildin, E., and Ghasemi, M. (2015). Complexity reduction of multi-phase flows in heterogeneous porous media. *SPE Journal*. SPE 167295.

Gildin, E. and Ghasemi, M. (2014). A new model reduction technique applied to reservoir simulation. In *14th European conference on the mathematics of oil recovery*. Sicily, Italy.

Gildin, E., Ghasemi, M., Romanovskay, A., and Efendiev, Y. (2013). Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media. In *SPE Reservoir Simulation Symposium*. The Woodlands, Texas. SPE 163618.

Heijn, T., Markovinovic, R., and Jansen, J. (2004). Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal*, 9(2).

Peherstorfer, B., Butnaru, D., Willcox, K., and Bungartz, H.J. (2014). Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1), A168–A192.

Van Doren, J., Markovinovic, R., and Jansen, J.D. (2004). Reduced-order optimal control of waterflooding using pod. In *9th European Conference on the Mathematics of Oil Recovery*.

Volkwein, S. and Kunisch, K. (2008). Proper orthogonal decomposition for optimality systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42, 1–23.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4), 395–416.