



## Article

# Data-Driven Model Reduction for Coupled Flow and Geomechanics Based on DMD Methods

Anqi Bao <sup>1</sup>, Eduardo Gildin <sup>1,\*</sup>, Abhinav Narasingam <sup>2</sup> and Joseph S. Kwon <sup>2</sup><sup>1</sup> Harold Vance Department of Petroleum Engineering, Texas A&M University, College Station, TX 77843, USA<sup>2</sup> Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77843, USA

\* Correspondence: egildin@tamu.edu; Tel.: +1-979-862-4578

Received: 3 May 2019; Accepted: 12 July 2019; Published: 19 July 2019



**Abstract:** Learning reservoir flow dynamics is of primary importance in creating robust predictive models for reservoir management including hydraulic fracturing processes. Physics-based models are to a certain extent exact, but they entail heavy computational infrastructure for simulating a wide variety of parameters and production scenarios. Reduced-order models offer computational advantages without compromising solution accuracy, especially if they can assimilate large volumes of production data without having to reconstruct the original model (data-driven models). Dynamic mode decomposition (DMD) entails the extraction of relevant spatial structure (modes) based on data (snapshots) that can be used to predict the behavior of reservoir fluid flow in porous media. In this paper, we will further enhance the application of the DMD, by introducing sparse DMD and local DMD. The former is particularly useful when there is a limited number of sparse measurements as in the case of reservoir simulation, and the latter can improve the accuracy of developed DMD models when the process dynamics show a moving boundary behavior like hydraulic fracturing. For demonstration purposes, we first show the methodology applied to (flow only) single- and two-phase reservoir models using the SPE10 benchmark. Both online and offline processes will be used for evaluation. We observe that we only require a few DMD modes, which are determined by the sparse DMD structure, to capture the behavior of the reservoir models. Then, we applied the local DMDc for creating a proxy for application in a hydraulic fracturing process. We also assessed the trade-offs between problem size and computational time for each reservoir model. The novelty of our method is the application of sparse DMD and local DMDc, which is a data-driven technique for fast and accurate simulations.

**Keywords:** sparsity promoting; dynamic mode decomposition; model order reduction; reservoir simulation; hydraulic fracturing

## 1. Introduction

Discovering the dynamics of the fluid flow through a reservoir based on production data is paramount to the development of fast predictive models for reservoir management. A data-driven surrogate model with less computational effort is needed to simulate the underlying physics of the reservoir and perform accurate future prediction. Reservoir management workflows usually require repeated simulations for production optimization, enhancing oil recovery, history matching and uncertainty quantification processes [1–4]. Especially, when solving the production optimization with nonlinear constraints [5,6], even with an advanced algorithm it takes thousands of forward simulations to achieve an optimal solution. Those scenarios can benefit tremendously from the rapid turnarounds that data-driven surrogate models can deliver. However, the complex physics of the multiphase fluid flow in porous media and the multiscale nature of the fluid and rock properties pose challenges in obtaining good predictive surrogate models.

Although data-driven surrogate models are being considered in many scenarios (drilling, production, reservoir and operations) [7,8], understanding (i.e., learning) the physics of the process is still of great value in designing optimal strategies for reservoir development [9–11]. In this paper, we are particularly interested in reduced-order modeling, which may be categorized as a physics-based surrogate model. Reduced complexity models are usually obtained by projecting the underlying equations of the high fidelity model to a smaller subspace [12].

Fluid flow in porous media is governed by infinite-dimensional nonlinear partial differential equations (PDEs), which is, in general, spatially discretized into a large dimensional set of nonlinear ordinary differential equations (ODEs) [13–15]. In order to represent consistently the geology and flow dynamics, millions of grid blocks are needed and time-consuming algorithms are used for their spatio-temporal solution. Reduced-order models, in this case, can yield faster, but accurate approximate solutions if designed properly.

Hydraulic fracturing is one of the most important well stimulation techniques that has made the recovery of shale oil and gas economically viable. Achieving uniform proppant concentration inside the fracture at the end of pumping is paramount in maximizing oil and gas extraction from the rock formation. However, hydraulic fracturing is a complex moving boundary problem characterized by simultaneous fracture propagation and proppant transport. The physics-based model for such a flow can be obtained by combining lubrication theory, elasticity equations and conservation laws which results in a complex set of partial differential equations. Additionally, since the spatial domain changes with time, the number of discretized nonlinear algebraic equations to be solved grows with time. These significant computational challenges associated with fully resolving the evolution of key variables in the hydraulic fracturing process can be addressed by accurate reduced-order models.

There has been a plethora of model reduction methodologies applied to porous media flow simulation for both single-phase and multiphase flow [16–19]. In particular, projection-based frameworks have been successfully implemented in reservoir simulation and hydraulic fracturing control. We can single out, the proper orthogonal decomposition (POD) method [17], trajectory piecewise linearization (TPWL) techniques [20,21], discrete empirical interpolation method (DEIM) [22], multivariable output error state space (MOESP) [23], among others. The newly emerging dynamic mode decomposition (DMD) method has been widely used recently to identify the coherent structure in reservoir simulation in both conventional reservoir [16,18,24] and unconventional reservoirs [9,25].

Model reduction by projection using POD modes is easy to implement. The spatial orthogonal POD modes are computed from snapshot data collected from numerical simulator or experiments [18,26], which is called “training step” in reservoir simulation and depends on several runs of a high fidelity model. The method selects the dominant modes that have strong impact on the solution by the rank of energy level. For DMD, the aim is to identify the dynamical system underlying the reservoir model by using only reservoir data. In this way, the numerical simulator is not necessarily to be used when approximating the solution. DMD methods have been gaining popularity in fluid flow modeling industry given its practical use and a solid theoretical background connected to nonlinear dynamics, namely, the spectral analysis of the Koopman operator [27,28], Fourier analysis [29] and iterative numerical methods, such as the Arnoldi method [30]. The Koopman operator, as the heart of the DMD method, is a linear operator whose modes and eigenvalues could be used to completely describe the nonlinear dynamic system. The modes will identify the coherent structure of the dynamics, while the eigenvalues will specify the decay rate and frequency of each corresponding mode. In fact, it is just because DMD is a numerical approximation to the Koopman spectral analysis, it could be applied to nonlinear systems. As the recent evolution of artificial intelligence, Koopman operator, as a crucial but hard to approximate infinite-dimensional operator, has been re-investigated by several researchers using deep learning neural network, dictionary learning and autoencoders [31–33]. In addition, the family of DMD methods has been growing to accommodate different situations, for example, the extended DMD [33] was developed to approximate the Koopman operator with dictionary learning,

DMD with control (DMDc) [34,35] for including the effect of control and disambiguate the underlying dynamics and the effect of actuation.

Both the POD and DMD methods are snapshot-based approaches, which means they rely on a two-step process called off- and online. In the offline step, one deals with gathering snapshots and computing the projection basis or modes, which are computed through an SVD decomposition. In the online step, one performs dimensionality-reduction of the states of the system, based on the highly ranked modes, and implement different boundary conditions as used in the offline step. However, unlike the POD method, it is hard for the standard DMD method to choose an optimal subset of the modes since the modes are not ranked by energy and it is even harder considering the modes are non-orthogonal. We cannot simply select modes with the largest amplitude or largest singular value. Several methods including optimal DMD [29] and sparse DMD [27] have been proposed to overcome this issue.

In this paper, we focus on the development of reduced-order models based on the DMD family of methods. In particular, we investigate two fixes for the selection of proper modes. We start with the sparsity-promoting DMD method, which was first proposed by Jovanovic et al. [27] for choosing appropriate DMD modes. Sparsity-promoting DMD uses a least square deviation norm plus an extra  $l_1$  norm that penalizes the number of non-zero basis vectors as a convex optimization function, to which the solution consists of the sparse structures of DMD modes. Note that  $l_1$  regularization is widely used and proved efficient in other fields of study such as statistics and geophysical inversion [36]. The purpose is to seek a trade-off between the number of non-zero DMD basis and the accuracy of the approximation.  $l_1$  norm instead of  $l_0$  norm (cardinality function) is used in order to keep the objective function convex [36].

Next, we investigate the use of DMDc. In many dynamical systems, the data often comes with associated interventions or exogenous inputs. In such cases, the developed reduced-order models must be able to extract the underlying dynamics by decoupling it from the effect of applied inputs. This is particularly important in the case when we are controlling a system. DMDc addresses this problem by providing a reduced-order description of the input-to-output behavior of the original high-dimensional system. At its heart, DMDc is a regression method that solves a least-squares problem to determine the associated system matrices that represent a discrete-time linear system. However, in the case of complex moving boundary problems since different portions of the solution trajectory can be associated with distinct regimes, a global linear model may fail to capture the dominant dynamics effectively. In this context, recently the concept of local model reduction was proposed to better capture the local dynamics of the system by integrating model reduction with temporal clustering of time-series data [37]. This idea has been extended to develop the local DMDc method [34], resulting in reduced-order models that are tailored to accurately approximate the system dynamics within specific regimes (clusters) [38,39]. There has also been related work where the authors propose a systematic method to make the local DMDc models valid on a larger domain [40].

In what follows, we assess the benefits of the reduced-order models derived from the DMD family of methods using two distinct applications. First, we apply the sparsity promoting DMD method to reservoir simulation using single- and two-phase models. We compare the relative errors and efficiency of the sparse DMD with two other reduced-order modeling methods, namely POD and standard DMD. In the second application, we assess the benefit of local DMDc to approximate the process dynamics in hydraulic fracturing, i.e., the evolution of proppant concentration at different positions with time. We start by stating the PDE equations used in this paper, both for the reservoir and the simplified hydraulic fracturing process. We, then provide the four algorithms considered here for constructing the projection basis, namely, POD, standard DMD, sparse DMD, and local DMDc. Finally, we demonstrate the benefits of the DMD family by applying the methods to a 3D reservoir simulation based on the SPE10 benchmark and with a hydraulic fracturing process. We follow the discussions by concluding our work. We should point out that although we use different reservoir model types (simplified and fully discretized) in both applications, the simulation of coupled flow and

geomechanics can be implemented in a fully coupled manner. As this is out of the scope of this paper, we will generalize these results in a forthcoming paper.

Although many of the tools used in this paper have been developed in different settings, the novelty of this work is threefold: (1) the application of sparsity-promoting DMD to multiphase flow in porous media and its comparison to POD-based methods; (2) the application of Local DMDc to coupled flow and geomechanics (PKN) with added complexities accounting for hydraulic fracturing dynamics, and to the author's knowledge, this is the first publication in this area; (3) in addition, in this paper, we proposed a new GOS clustering algorithm which facilitates optimal clustering (both the number of clusters and the cluster configuration) of the data based on their dynamical behavior.

## 2. Problem Formulation and Methodology

### 2.1. Multi-Phase Flow Problem

Flow and transport models in porous media can be described by a set of partial differential equations representing conservation of mass, momentum and energy as functions of pressure, saturation and temperature, and reconciled by the equation of state. In what follows, we describe briefly the single and two-phase flows models used in this paper.

As discussed in References [13,15], to simplify the problem we neglect the inertial effects and assume the flow to be isothermal. Hence, one can assume the black-oil formulation, where there are two components (oil-water) and there are two phases of the hydrocarbon substance (oil and gas) present in the reservoir.

For the case of a two-phase oil-water system in a reservoir domain (denoted by  $\Omega$ ), the mass balance equation for each phase is

$$\nabla \cdot (\rho_\chi u_\chi) + \frac{\partial(\rho_\chi \varphi s_\chi)}{\partial t} - \rho_\chi q_\chi = 0 \text{ in } \Omega, \chi \in \{o, w\}, \quad (1)$$

where  $\rho_\chi$  is the fluid phase density,  $u_\chi$  is the fluid phase superficial velocity,  $t$  is time,  $\nabla$  denotes the divergence operator,  $\varphi$  is the porosity of the rock,  $s_\chi$  denotes fluid phase saturation,  $q_\chi$  is the volumetric source/sink term and subscript  $\chi \in \{o, w\}$  indicates the oil and water phases, respectively. We apply Darcy's law:

$$u_\chi = \frac{k_{r\chi}}{\eta_\chi} \mathbf{K} \cdot (\nabla p_\chi - \rho_\chi g \nabla h) \text{ in } \Omega \quad (2)$$

where  $\nabla$  is the gradient operator,  $\mathbf{K}$  denotes the permeability tensor,  $\eta_\chi$  is the fluid phase viscosity,  $k_{r\chi}$  denotes the relative permeability of each phase (which is a function of water saturation),  $P_\chi$  is the phase pressure,  $g$  is the constant of gravity acceleration and  $h$  denotes depth. Combining Equations (1) and (2) yields

$$\nabla \cdot \left[ -\frac{\rho_\chi k_{r\chi}}{\eta_\chi} \mathbf{K} \cdot (\nabla P_\chi - \rho_\chi g \nabla h) \right] + \frac{\partial(\rho_\chi \varphi s_\chi)}{\partial t} - \rho_\chi q_\chi = 0. \quad (3)$$

The general oil-water model is completed by enforcing the saturation constraint  $s_o + s_w = 1$  and by specifying a capillary pressure relationship  $P_c(s_w) = P_o - P_w$ . As can be seen, Equation (3) is nonlinear. We take  $P_o$  and  $s_w$  to be the primary unknown variables, from which  $P_w$  and  $s_o$  can be easily computed. Equation (3) can be solved numerically using a fully-implicit finite volume procedure. More details can be found in Reference [13].

With the fully-implicit procedure, each grid block is characterized by two primary state variables, oil pressure  $p_o$  and water saturation  $S_w$ . Defining  $\mathbf{x} = [P_o, s_w]$  as the state vector for all reservoir cells and  $\mathbf{v}$  as the well control parameters, one can write Equation (3) in the fully implicit form as

$$R(\mathbf{x}, \mathbf{x}^n, \mathbf{v}^{n+1}) = F(\mathbf{x}) + A(\mathbf{x}, \mathbf{x}^n) + Q(\mathbf{x}, \mathbf{v}^{n+1}) = 0, \quad (4)$$

where  $R$  is the residual vector, superscripts  $n$  and  $n + 1$  indicate time levels,  $F(x) = T(x)x$  represents the transmissibility term, with  $T(x)$  being the transmissibility matrix,  $A(x, x^n) = -D(x)(x - x^n)$  is the accumulation term, and  $Q(x, v^{n+1})$  is the source/sink term. The source/sink term corresponding to injection/production from a well is modeled by using a well equation as described in Reference [41]:

$$-q_\chi = WI\lambda_\chi(P_\chi - P_{wb}), \quad (5)$$

where  $q_\chi$  is the well flow rate of phase  $\chi$ ,  $\lambda_\chi = \rho_\chi k_{r\chi} / \eta_\chi$  is the mobility of phase  $\chi$ ,  $P_\chi$  is the computed pressure of phase  $\chi$  in the reservoir cell,  $P_{wb}$  is the wellbore pressure, and  $WI$  is the well index which in this paper is based on the Peaceman model [41]. In this paper, bottom hole pressure (BHP) is chosen as the well control parameter  $v$ . The solution of Equation (4) is to find the spatio-temporal evolution of the state  $x^{n+1}$  such that

$$R(x^{n+1}, x^n, v^{n+1}) = 0. \quad (6)$$

Equation (6) is nonlinear. Typically, the Newton-Raphson iteration method is used to solve Equation (6), where  $x^{n+1}$  is obtained iteratively as follows:

$$x^{n+1,k+1} = x^{n+1,k} - J(x^{n+1,k})^{-1} R(x^{n+1,k}, x^n, v^{n+1}). \quad (7)$$

In Equation (7),  $J$  is the Jacobian matrix and is defined as  $J(x) = \partial R / \partial x$ . Iteration for solving Equation (7) will be stopped when the relative norm of the residual is close to zero. Given the size of the Jacobian matrix and the computational cost associated with solving the linear system in the Newton-Raphson algorithm, model order-reduction described in the next sections can alleviate some of these bottlenecks by reducing the size of these matrices. Before we describe the POD and DMD methods and their application to reservoir simulation, it is worth noting two relevant aspects of our particular implementation.

**Single-phase flow case.** In the case of single-phase flow, the equations simplify and the mass balance equation can be given by a single equation in the only unknown pressure.

**Mixed formulation for the two-phase flow.** In this paper, we use for the two-phase flow system a mixed formulation of the equation [26]. Previous works have demonstrated that it gives more stable computations, especially after applying the projection matrices. To this end, the system of discretized equations is of the form

$$\begin{bmatrix} B & -C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ D \end{bmatrix}, \quad (8)$$

where  $B$ ,  $C$  and  $D$  are matrices and vectors defined using the flux and pressure equations. The reader can refer to [14] for more details of the formulation.

## 2.2. Hydraulic Fracturing Process

In this work, we consider a nonlinear hydraulic fracturing process characterized by simultaneous fracture propagation and fluid transport. A brief description of the equations governing the fracture dynamics is presented below.

**Fracture propagation.** The fracture propagation is modeled using the Perkins-Kern-Nordgren (PKN) model [42,43]. It is the most common 2D hydraulic fracturing model which assumes that the fracture length (x-axis shown in Figure 1) is much greater than the fracture height (y-axis), as is the case in low permeability formations. Additionally, the following assumptions are also considered: (1) the fractures are confined vertically (constant height), (2) the rock properties remain constant with respect to time and space and (3) because the fracture length is much greater than its width (z-axis), the fracture propagation and proppant advection are neglected along the z-axis. In the PKN model, the fracture cross-sectional area is considered to be elliptical and the fracture shape is rectangular [44].

By taking into account the fracture volume changes and fluid leak-off into the surrounding reservoir, the local mass conservation of an incompressible fluid inside the fracture is given by the following form

$$\frac{\partial A}{\partial t} + \frac{\partial q_x}{\partial x} + HU = 0, \quad (9)$$

where  $A = \pi WH/4$  is the cross-sectional area of the elliptic fracture,  $t$  is the elapsed time,  $H$  is the fracture height,  $U$  is the fluid-leak off into the formation and  $q_x$  is the local flow rate along the  $x$ -axis which can be obtained using lubrication theory and the elasticity equation as follows

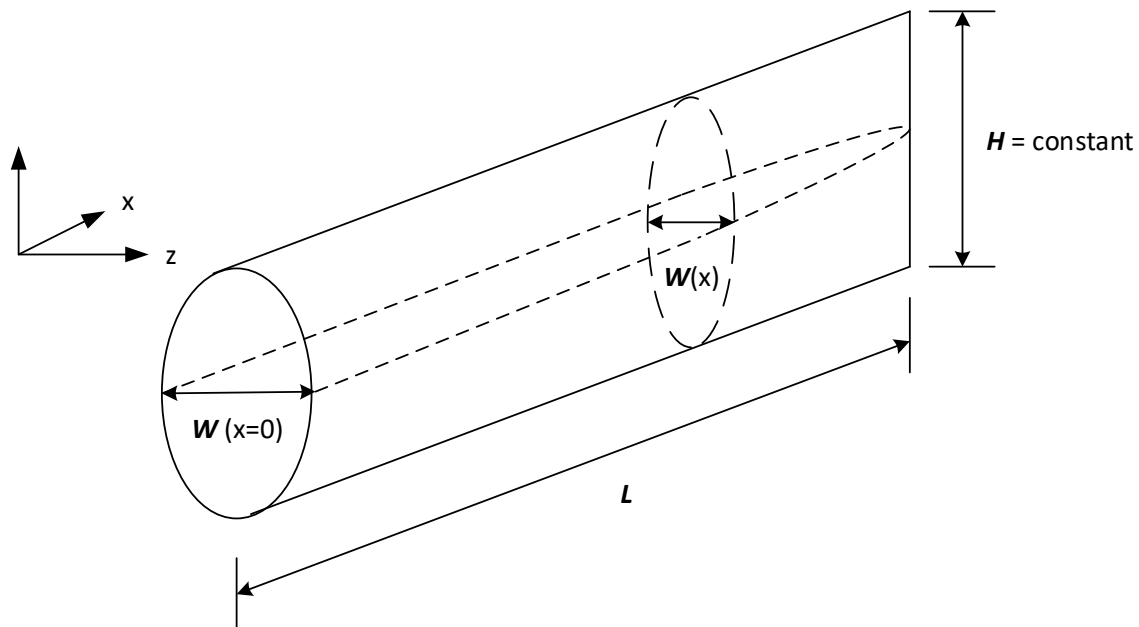
$$q_x = \frac{\pi HEW^3}{128\mu(1-\nu^2)} \cdot \frac{\partial W}{\partial x}, \quad (10)$$

where  $E$  is the Young's modulus of the formation,  $\mu$  is the fracturing fluid viscosity,  $\nu$  is the Poisson ratio of the formation, and  $W$  is the fracture width. The rate of fluid leak-off into the surrounding rock formation per unit height normal to the elliptic fracture plane is given by

$$U = \frac{2C_{\text{leak}}}{\sqrt{t - \tau(x)}}, \quad (11)$$

where  $C_{\text{leak}}$  is the overall fluid leak-off coefficient, and  $\tau(x)$  is the time at which the fracture propagation has arrived at the location  $x$  for the first time. By substituting Equation (10) into Equation (9) we get the following parabolic PDE model with a moving boundary that describes fracture propagation in a hydraulic fracturing process.

$$\frac{\pi H}{4} \frac{\partial W}{\partial t} - \frac{\pi E}{128\mu(1-\nu^2)} \left[ 3W^2 \left( \frac{\partial W}{\partial x} \right)^2 + W^3 \frac{\partial^2 W}{\partial x^2} \right] + HU = 0. \quad (12)$$



**Figure 1.** Perkins-Kern-Nordgren (PKN) hydraulic fracture model.

The above PDE is associated with the following boundary and initial conditions.

$$q_x(0, t) = q_0 \quad \text{and} \quad W(L(t), t) = 0, \quad (13a)$$

$$W(x, 0) = 0, \quad (13b)$$

where  $q_0$  is the injection rate of fracturing fluid at the wellbore and  $L(t)$  is the fracture length.



**Proppant transport.** In this model, the proppant transport is modeled assuming simultaneous advection and settling at the fracture bottom, at the velocity equal to the carrier fluid. Moreover, it is assumed that the proppant particles are large enough to neglect the diffusivity of the solid phase. Therefore, the convective flux can be computed as follows:

$$\frac{\partial(WC)}{\partial t} + \frac{\partial}{\partial x}(WCV_p) = 0, \quad (14)$$

$$C(0, t) = C_0(t) \quad \text{and} \quad C(x, 0) = 0, \quad (15)$$

where  $C(x, t)$  is the proppant concentration inside the fracture,  $C_0(t)$  is the injected concentration at the wellbore (manipulated input),  $V_p$  is the velocity of individual proppant particles which is related to the velocity of the fluid, and  $V$ , and the gravitational settling velocity,  $V_s$ , as follows [45]:

$$V_p = V - (1 - C)V_s, \quad (16)$$

The gravitational settling velocity in the above equation can be computed as follows [46]:

$$V_s = \frac{(1 - C)^2 (\rho_{sd} - \rho_f) g d^2}{10^{1.82C} 18\mu}, \quad (17)$$

where  $\rho_{sd}$  is the proppant particle density,  $\rho_f$  is the pure fluid density,  $g$  is the gravitational acceleration constant, and  $d$  is the proppant diameter. The fracturing fluid viscosity is related to the suspended proppant concentration by the following empirical model

$$\mu(C) = \mu_0 \left(1 - \frac{C}{C_{\max}}\right)^{-\alpha}, \quad (18)$$

with  $\mu_0$  being the pure fluid viscosity at  $C = 0$ ,  $\alpha$  is an exponent usually in the range of 1.2 – 1.8, and  $C_{\max}$  is the maximum theoretical concentration determined as  $C_{\max} = (1 - \phi)\rho_{sd}$  where  $\phi$  is the proppant bank porosity.

### 2.3. Model Reduction by Projection

We introduced the Galerkin projection theory, as in previous works, to reduce the dimensionality of the system. By using  $P(t) = \Phi P_r(t)$ , for instance for single-phase [12], where  $\Phi^* \Phi = I_k$  and  $\Phi = \{\phi_i\}_{i=1}^r \in \mathbb{R}^{nc \times r}$  is the matrix that contains basis vectors. Note that for the POD method, the column vectors are orthogonal to each other. Thus the reduced system is of the form

$$R_r = \Phi^* R, \quad (19)$$

$$J_r = \frac{\partial R_r}{\partial P_r} = \Phi^* \frac{\partial R}{\partial P} \frac{\partial P}{\partial P_r} = \Phi^* J \Phi. \quad (20)$$

In the case of the mixed formulation, we can assume we have basis  $\Phi_{u,p} \in \mathbb{C}^{(nu+nc) \times r}$  for velocity and pressures, and thus, the Galerkin projection on to the subspace spanned by the basis  $\Phi_{u,p}$  is given by

$$\Phi_{u,p}^T \begin{bmatrix} B & -C^T \\ C & 0 \end{bmatrix} \Phi_{u,p} \begin{bmatrix} u_r \\ P_r \end{bmatrix} = \Phi_{u,p}^T \begin{bmatrix} 0 \\ D \end{bmatrix}. \quad (21)$$

### 2.4. The POD Method

The POD method is derived from constructing a lower-dimensional approximation in Hilbert space [47]. Basically, it can be viewed as an application of singular value decomposition in a finite-dimensional space (usually Euclidean space). We will briefly review the method here.

Consider given a set of pressure snapshots at different time steps  $\mathbb{P} = [P^1, \dots, P^{ns}]^T \in \mathbb{R}^{nc \times ns}$ ,  $P^j = P(t_j)$ ,  $d = \dim(\text{span}\{P^1, \dots, P^{ns}\})$ , where  $ns$  is the number of time steps and  $nc$  is the number of grid blocks. Let  $\{\phi_i\}_{i=1}^d$  denotes the orthonormal basis, then each of the snapshot columns can be expressed as

$$P(t_j) = \sum_{i=1}^d \langle P^j, \phi_i \rangle \phi_i. \quad (22)$$

The goal is to find the corresponding basis  $\{\phi_i\}_{i=1}^l$  with  $l \ll d$  to describe the snapshot data by solving the minimization problem

$$\min_{\{\phi_i\}_{i=1}^l} \sum_{j=0}^{ns} \xi_j \|P^j - \sum_{i=1}^l \langle P^j, \phi_i \rangle \phi_i\|_H, \quad (23)$$

$$\text{Subject to } \langle \phi_j, \phi_i \rangle = \delta_{ij}$$

where  $\{\xi_j\}_{j=0}^{ns}$  are positive coefficients,  $H$  denotes a proper Hilbert space. The solution of the above minimization problem is obtained from singular value decomposition of the snapshot matrix [47]. For simplification purposes, we perform economy-sized SVD in Euclidean space as

$$\mathbb{P} = V \Sigma W^*, \quad (24)$$

where  $V = \{v_i\}_{i=1}^d \in \mathbb{R}^{nc \times d}$  and  $W = \{w_i\}_{i=1}^d \in \mathbb{R}^{ns \times d}$  are unitary matrices,  $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_d\} \in \mathbb{R}^{d \times d}$  contains singular values of snapshot matrix. The solution for optimal POD basis is simply the first  $l$  columns in  $V$ , which is  $\{v_i\}_{i=1}^l$ .

The error in corresponding space is

$$\min_{\{\phi_i\}_{i=1}^l} \sum_{j=0}^{ns} \xi_j \left\| P^j - \sum_{i=1}^l ((P^j)^T \phi_i) \phi_i \right\|_2^2 = \sum_{i=l+1}^d \sigma_i^2. \quad (25)$$

This error formula indicates a factor called energy fraction Equation (26) that can be used to improve the performance of POD basis by choosing the truncation parameter  $l$  so that  $EF$ , defined below, is close to one.

$$EF = \frac{\sum_{i=1}^l \sigma_i^2}{\sum_{i=1}^d \sigma_i^2}. \quad (26)$$

## 2.5. Dynamic Mode Decomposition (DMD)

Dynamic mode decomposition (DMD) has been proposed as a non-intrusive method for discovering system dynamics [30]. It can be a non-intrusive method from the simulator-run point of view. In this case, we call it an offline process. The major assumption of the method is that there exists a linear mapping matrix  $A$  that connects the temporal evolution of the state of the system. Without elaborating further on the theory of DMD, we provide the steps used in the DMD algorithm. The reader can refer to Schmid [30] for more details.

1. Suppose we have two snapshot matrices comprised of the states of the system. They can be pressures for the single-phase flow case, or pressures, velocities and saturations for the two-phase flow. The columns of the matrix data are captured equal-spaced in time, with a time step  $\Delta t$ .

$$\mathbb{P}_2 = \begin{bmatrix} | & | & \cdots & | \\ P^2 & P^3 & \cdots & P^{ns} \\ | & | & \cdots & | \end{bmatrix}$$



$$\mathbb{P}_1 = \begin{bmatrix} | & | & \cdots & | \\ P^1 & P^2 & \cdots & P^{ns-1} \\ | & | & \cdots & | \end{bmatrix}.$$

Each  $P^i = P(i\Delta t)$  is a vector with  $nc$  components ( $nc$  gridlocks), and thus  $\mathbb{P}_1, \mathbb{P}_2 \in \mathbb{R}^{nc \times (ns-1)}$ . Ideally, we assume there is a time-independent matrix  $A$  that satisfies

$$P^{i+1} = AP^i, \quad (27)$$

and thus

$$\mathbb{P}_2 = A\mathbb{P}_1. \quad (28)$$

In addition, notice that this assumption potentially gives us a Krylov subsequence since

$$\mathbb{P}_1 = \{P^1, AP^1, A^2P^1 \dots A^{ns-2}P^1\}.$$

Next, to get an approximation of the matrix  $A$ , we perform economy-sized SVD (singular value decomposition) on snapshot matrix  $\mathbb{P}_1$ .

$$\mathbb{P}_1 = YS_\mu Z^*. \quad (29)$$

Here  $Y = \{y_i\}_{i=1}^r \in \mathbb{R}^{nc \times r}$  and  $Z = \{z_i\}_{i=1}^r \in \mathbb{R}^{ns \times r}$  are unitary matrix, and  $S_\mu = \text{diag}\{\mu_1, \dots, \mu_d\} \in \mathbb{R}^{r \times r}$ ,  $r \leq \min\{n, ns-1\}$  is the rank of matrix  $\mathbb{P}_1$ .

2. Define matrix  $\tilde{A}$  as

$$\tilde{A} = Y^*AY = Y^*\mathbb{P}_2ZS_\mu^{-1}. \quad (30)$$

3. Perform eigenvalue decomposition of matrix  $\tilde{A}$

$$\tilde{A}X = X\Lambda, \quad (31)$$

where  $X = [x_1 \dots x_r] \in \mathbb{R}^{r \times r}$  is eigenvector matrix and  $\Lambda = \text{diag}\{\lambda_1 \dots \lambda_r\} \in \mathbb{R}^{r \times r}$  is eigenvalue diagonal matrix.

4. The dynamic mode can be calculated as

$$\phi_j = Ux_j, \quad \Phi = \{\phi_i\}_{i=1}^r. \quad (32)$$

It is usually preferred to normalize the DMD mode  $\phi_j$  as  $\overline{\phi_j}$  according to the first snapshot so that it preserves the correct scale [48]. Essentially,  $d_{\text{DMD}} = \Phi^*P^1$ ,  $\overline{\phi_j} = \phi_j d_{\text{DMD}}$ .

5. Finally, the pressure value at a particular time step  $k$  can be approximated as

$$P^k \approx \sum_{j=1}^r (\lambda_j)^k \overline{\phi_j}. \quad (33)$$

## 2.6. Local DMDc

Local DMDc was proposed as an extension to DMDc to tackle complex dynamics such as moving boundary problems [34]. For the most part, the formulation of local DMDc follows that of DMDc except it has an additional post-processing step of clustering time-series data. Local DMDc uses a novel clustering technique called Global optimum search (GOS) [49] that has proven to outperform heuristic algorithms like k-means. It partitions data based on their similarity in dynamic behavior and has an added advantage of auto-computing the optimal number of clusters. Once the clustering step is performed, the local DMDc algorithm then proceeds by applying DMDc to determine local reduced-order models within each cluster. Below are the steps to implement the local DMDc algorithm. For more details please refer to the article [34].

1. Similar to DMD, we first collect the snapshot data for the states of the system at  $ns$  time steps. With regards to the hydraulic fracturing process, these can be the time-series proppant concentration or width profiles at different positions. We also collect the input data and construct the state and input snapshot matrices as below:

$$\mathbb{C}_1 = [\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{ns-1}], \mathbb{C}_2 = [\mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{ns}], \mathbb{C}_0 = [\mathbf{C}_{0,1}, \mathbf{C}_{0,2}, \dots, \mathbf{C}_{0,ns-1}], \quad (34)$$

where each  $\mathbf{C}_i$  is a vector of proppant concentrations with  $nc$  spatial nodes obtained from discretizing the PDEs, so that  $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{nc \times (ns-1)}$  and the manipulated input for the system is the injected proppant concentration at the wellbore  $\mathbf{C}_0 \in \mathbb{R}^{1 \times (ns-1)}$ .

2. Apply the GOS algorithm to cluster the entire snapshot data, i.e.,  $\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2 = [\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{ns}]$  into  $m$  optimal clusters such that

$$\mathbb{C}^k = \{\mathbf{C}_1^k, \mathbf{C}_2^k, \dots, \mathbf{C}_{n_k}^k\}, \quad \text{for } k = 1, \dots, m, \quad (35)$$

where  $\mathbb{C}^k$  denotes the  $k$ -th cluster configuration and  $n_k$  denotes the number of snapshots in the  $k$ -th cluster. Please note that the following relations will be satisfied.

$$\bigcup_{k=1}^m \mathbb{C}^k = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{ns}\}, \quad \sum_{k=1}^m n_k = ns, \quad (36)$$

3. For every cluster, form the augmented state matrix,  $\mathbf{\Omega}^k \in \mathbb{R}^{(nc+1) \times (ns-1)}$  by stacking the state and input matrices as shown below and compute the reduced SVD of  $\mathbf{\Omega}^k$  along with the shifted snapshot matrix  $\mathbb{C}_2^k$

$$\mathbf{\Omega}^k = \begin{bmatrix} \mathbb{C}_1^k \\ \mathbb{C}_0^k \end{bmatrix}, \quad (37)$$

$$\mathbf{\Omega}^k = \mathbf{U}^k \mathbf{\Sigma}^k \mathbf{V}^{k*}, \quad (38)$$

$$\mathbf{\Omega}^k = \mathbf{U}^k \mathbf{\Sigma}^k \mathbf{V}^{k*}, \quad (39)$$

where  $\mathbf{U}^k \in \mathbb{R}^{(nc+1) \times r}$ ,  $\mathbf{V}^k \in \mathbb{R}^{n_k \times r}$ ,  $\mathbf{\Sigma}^k \in \mathbb{R}^{r \times r}$  are the matrices comprising the left singular vectors, right singular vectors and singular values of  $\mathbf{\Omega}^k$  and similarly  $\mathbf{W}^k \in \mathbb{R}^{nc \times r}$ ,  $\mathbf{R}^k \in \mathbb{R}^{n_k \times r}$  and  $\mathbf{\Gamma}^k \in \mathbb{R}^{r \times r}$  are the matrices comprising the left singular vectors, right singular vectors and singular values of  $\mathbb{C}_2^k$ .

4. Using the above matrices, compute the local reduced-order models as follows

$$\mathbf{A}^k = \mathbf{W}^{k*} \mathbb{C}_2^k \mathbf{V}^k \mathbf{\Sigma}^{k-1} \mathbf{U}_1^{k*} \mathbf{W}^k, \quad (40)$$

$$\mathbf{B}^k = \mathbf{W}^{k*} \mathbb{C}_2^k \mathbf{V}^k \mathbf{\Sigma}^{k-1} \mathbf{U}_2^{k*}, \quad (41)$$

where  $\mathbf{U}_1^k, \mathbf{U}_2^k$  are obtained by decomposing  $\mathbf{U}^k$  as  $\mathbf{U}^k = \begin{bmatrix} \mathbf{U}_1^k \\ \mathbf{U}_2^k \end{bmatrix}$  such that  $\mathbf{U}_1^k \in \mathbb{R}^{nc \times r}$ ,  $\mathbf{U}_2^k \in \mathbb{R}^{1 \times r}$  and  $\mathbf{A}^k \in \mathbb{R}^{r \times r}$ ,  $\mathbf{B}^k \in \mathbb{R}^{r \times 1}$  are the reduced-order representations of the system matrices.

## 2.7. Sparsity Promoting Dynamic Mode Decomposition

Unlike the POD method that selects the subset of the mode according to the rank of the energy level from singular values, there are no natural ways for the DMD method to select modes. Furthermore, the DMD modes are not orthogonal, which makes the projection behave in a different way as the POD modes. One of the approaches proposed is the sparsity-promoting DMD method that includes the sparsity structure by augmenting an objective function,  $I$  as in the Equation (44), with an additional term that penalizes the cardinality of non-zero elements. For detailed proof and procedure, we direct to the paper [30].

We return to the classical DMD method and introduce a vector of amplitude  $\alpha$  to Equation (33),

$$P^k \approx \sum_{j=1}^r (\lambda_j)^k \overline{\phi_j} \alpha_j, \quad (42)$$

where  $\alpha = [\alpha_1 \dots \alpha_r]$ .

In the matrix form:

$$\begin{bmatrix} P^1 & P^2 & \dots & P^{n_s-1} \end{bmatrix} \approx \begin{bmatrix} \phi_1 \phi_2 & \dots & \phi_r \end{bmatrix} \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_r \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \dots & (\lambda_1)^{n_s-1} \\ 1 & \lambda_2 & \dots & (\lambda_2)^{n_s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_r & \dots & (\lambda_r)^{n_s-1} \end{bmatrix} = \Phi \alpha V_{\text{and}}. \quad (43)$$

The object is to determine the initial value of  $\alpha$  and furthermore seek a sparsity structure with an optimal value of  $\alpha$ . The initial value of  $\alpha$  is determined through the following optimization problem

$$\min_{\alpha} I(\alpha) = \|\mathbb{P}_1 - \Phi \alpha V_{\text{and}}\|_F^2. \quad (44)$$

To determine the sparsity structure of  $\alpha$ , an extra penalty term of  $l_1$  norm instead of the  $l_0$  norm (cardinality function) is used to preserve the convex structure of the optimization problem.

$$\min_{\alpha} I(\alpha) + \gamma \sum_{i=1}^r |\alpha_i|. \quad (45)$$

The process could be illustrated with Figure 2. After the sparsity structure is fixed for  $\alpha$ , the problem reduces to determining only the nonzero amplitude element from the optimization problem in Equation (46), with the constraint  $E^T \alpha = 0$ , where the matrix  $E$  contains the sparsity structure of  $\alpha$ . For example, if we prefer a structure of alpha as  $\alpha = [0 \ \alpha_2 \ 0 \ \alpha_4]^T$ , the matrix  $E$  could be  $E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T$ . Finally, polish the results to get the optimal amplitude by using the method of Lagrange multiplier again:

$$L_p(\alpha, \tau) = I(\alpha) + \tau^* E^T \alpha + (E^T \alpha)^* \tau, \quad (46)$$

where  $\tau$  is the vector of Lagrange multiplier.

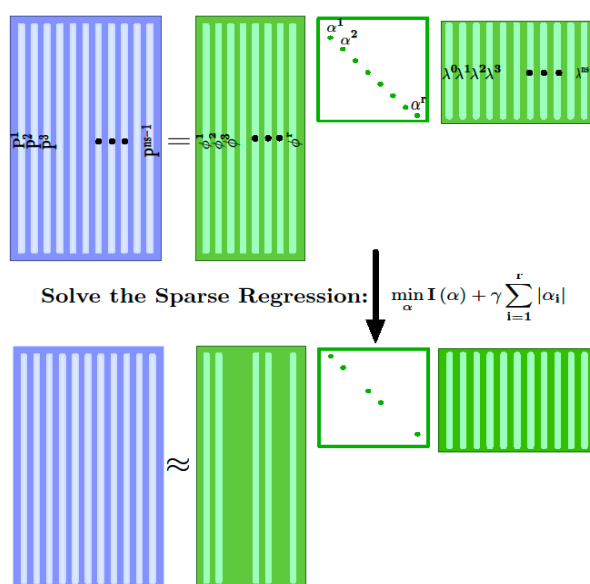


Figure 2. Step of solving for sparsity structure (Bao and Gildin, 2017).

### 3. Results

In this section, we apply the sparse DMD and local DMDc techniques discussed above to compare them with the standard DMD and POD approaches. We formulate the model reduction workflow by using single and two-phase flow examples.

#### 3.1. Single Phase Flow Example

In this example, we will demonstrate how sparsity DMD modes are applied to the single-phase flow model. Consider a reservoir discretized into  $15 \times 15 \times 3$  Cartesian grids of size 30 ft  $\times$  30 ft  $\times$  30 ft, the field is equipped with one producer at the center of the reservoir with constant bottom-hole pressure of 2900 psia. We assume the well is only perforated at the bottom layer (layer1). The permeability and porosity of the three layers are shown in Figure 3. The initial pressure is set to be 3100 psia.

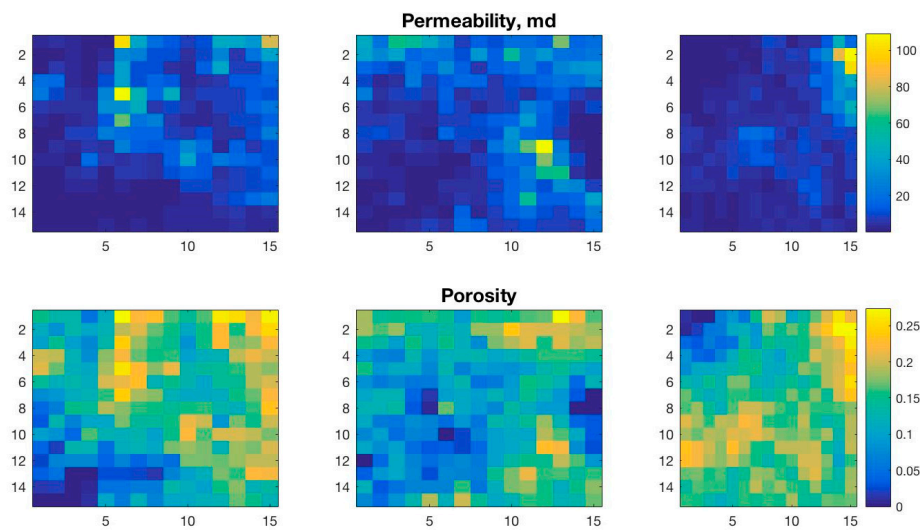


Figure 3. Permeability and porosity of single-phase reservoir model.

We simulate the reservoir for 365 days to get the pressure snapshot matrix of dimension  $675 \times 365$ .

Through the sparsity DMD algorithm, we obtain the basis with the sparsity structure. The level of the sparsity of structure will depend on the value of the penalty factor gamma ( $\gamma$ ) in Equation (45). The larger the gamma value, the higher the sparsity level will be (Figure 4). The sparsity pattern can be seen further from Figure 5. Out of a total 26 DMD modes (the rank of the singular value matrix is 26), the sparsity DMD is able to select several of the important modes with low level frequency (smaller value for the imaginary part of the eigenvalue).

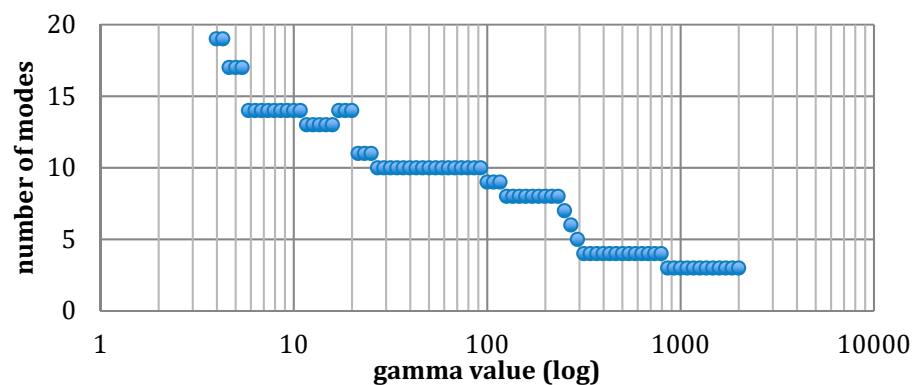
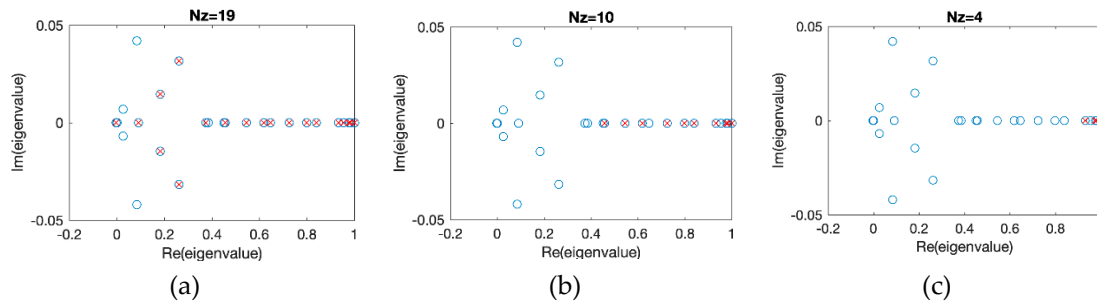


Figure 4. Number of sparsity promoting dynamic mode decomposition (DMD) modes changes with penalty factor gamma.



**Figure 5.** Eigenvalues of  $\tilde{A}$  (the matrix that contains coherent structure for the DMD method as in Equation (20)). The crosses are from for sparsity promoting DMD algorithm with  $N_z$  number of DMD modes. The circles are for the standard DMD algorithm. (a)  $N_z = 19$ ; (b)  $N_z = 10$ ; (c)  $N_z = 4$ .

The basis can be used either in the offline process or the online process. Note that the offline process does not require the use of a simulator; instead, it only involves a superposition on the basis of Equation (43). Thus, with affordable effort to compute the sparse DMD modes (see Figure 6), we could obtain a computationally more efficient prediction, especially with the offline process. The online process utilizes Galerkin projection to project the system onto subspaces spanned by the basis. The final results shown in Figure 7 implies good agreement between fine-scale model and reduced model (with sparse DMD). In all examples, the overall relative error is computed with the Frobenius norm:

$$\text{error} = \frac{\|\mathbb{P}_{ref} - \mathbb{P}_{app}\|_F}{\|\mathbb{P}_{ref}\|_F} \times 100\%$$

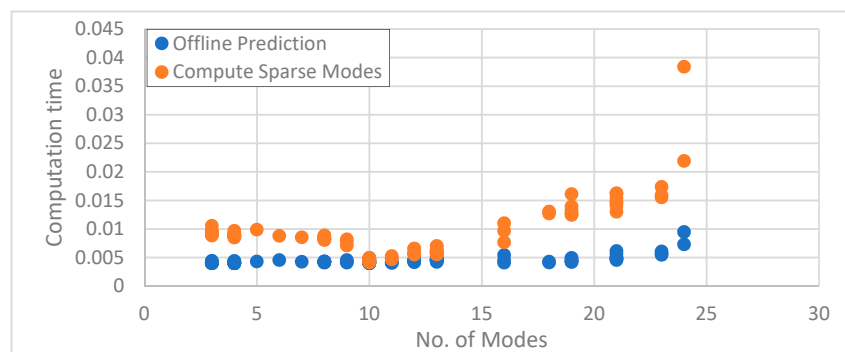
where  $\mathbb{P}_{ref}$  is the snapshot matrix from the simulator,  $\mathbb{P}_{app}$  is approximated solution from model reduction.

The relative error in terms of days is computed with  $l_2$  norm:

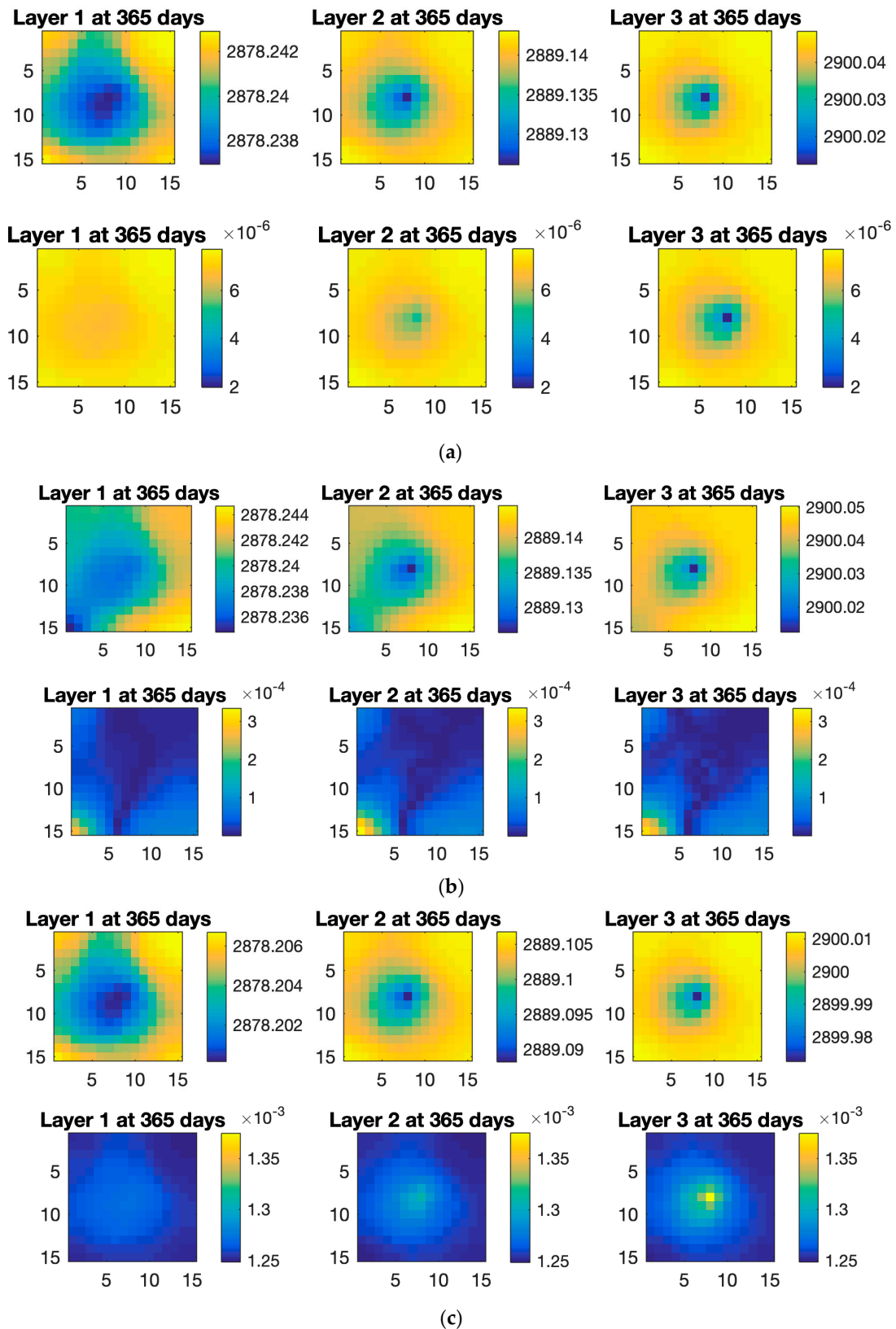
$$P_{err}(t_i) = \frac{\|\mathbb{P}_{ref}(t_i) - \mathbb{P}_{app}(t_i)\|_{l_2}}{\|\mathbb{P}_{ref}(t_i)\|_{l_2}} \times 100\%$$

where  $\mathbb{P}_{ref}(t_i)$  is a vector of pressure results from simulator at time  $t_i$ , while  $\mathbb{P}_{app}(t_i)$  is the approximated solution from model reduction.

As we already noticed before [50], the overall error of sparse DMD compared with the POD method suggests unstable behavior of POD, the error is not always decreasing as the number of modes decreased. When the number of modes reduced to a certain amount (in this case five) for the POD method, as seen in Figure 6, the pressure map shows incorrect, non-physical behavior: lower-left corner of layer one has pressure even lower than the actual production well. We further investigate the computation efficiency of sparsity DMD by showing the reduced solver time with fewer basis used, as seen in Table 1.



**Figure 6.** Computation time for sparse DMD modes and offline prediction.



**Figure 7.** Results of reduced-order model with different methods, within each group of (a–c) the upper one is pressure map and the lower one is error map. (a) Online sparsity DMD with five basis (first row = pressure maps; second row = error maps); (b) Online proper orthogonal decomposition (POD) with 5 basis (first row = pressure maps; second row = error maps); (c) Offline sparsity DMD with 24 basis (first row = pressure maps; second row = error maps).

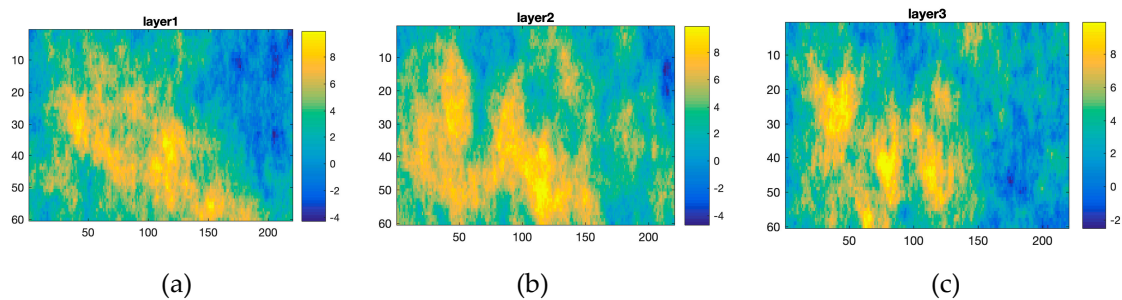


**Table 1.** Computation time of solver with different model order reduction (MOR) method.

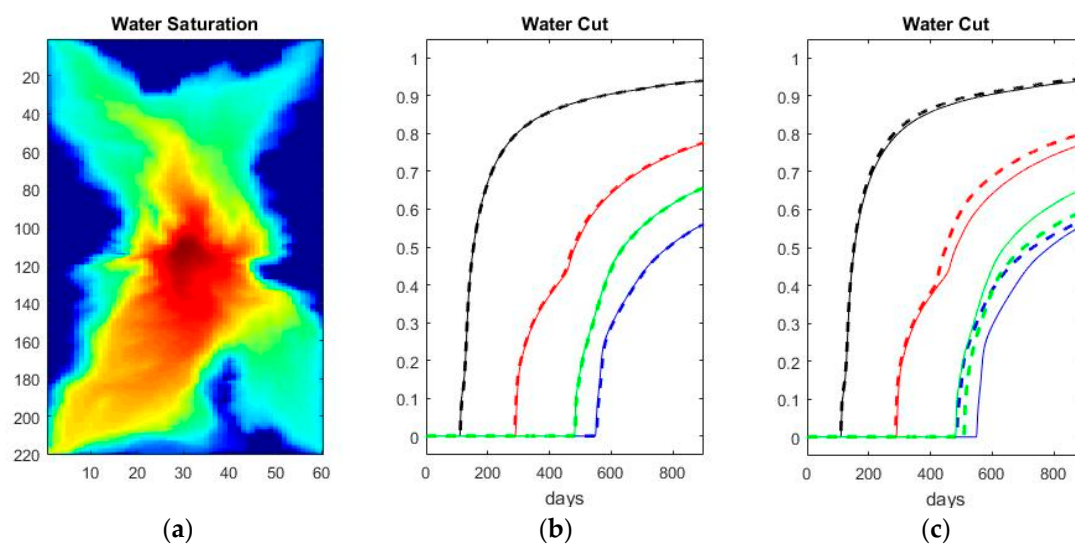
Method	Jacobian Size	Computation Time (s)	Relative Error (%)
Fine-scale	$675 \times 675$	9.89 (solver time)	-
Standard DMD, online	$50 \times 50$	1.51 (solver time)	$1.77 \times 10^{-4}$
Sparse DMD, online	$10 \times 10$	0.09 (solver time)	0.0056
Sparse DMD, online	$5 \times 5$	0.0748 (solver time)	0.011
Sparse DMD, offline	$23 \times 23$	0.042 (total time)	0.010

### 3.2. Two-Phase Flow Example

In this section, we will present the results of sparse DMD method when it's applied to 2D and 3D heterogeneous reservoir with two-phase flow, five-spot pattern: four producers at the corner and one injector at the center. For 3D reservoir, the production well is only perforated in the bottom layer and the injection well is only perforated in the top layer. We use the SPE10 benchmark [38] with one layer and three layers cases to show our method is capable to apply to both 2D and 3D reservoir. The absolute permeability for each of the three layers is shown in Figure 8. The relative permeability is approximated with the Corey correlation, which is essentially a power-law function of saturation. The 2D case ( $60 \times 220 \times 1$  grid blocks) results have been discussed in a previous paper [50] and we will briefly show here (Figure 9). For the 3D case, the detail of the reservoir is shown in the following Table 2.



**Figure 8.** Logarithmic of permeability field in the 3D two-phase flow reservoir model. (a) First layer; (b) Second layer; (c) Third Layer.



**Figure 9.** Results comparison in 2D reservoir (a) Final water saturation (b) water cut for the reduced model (78 sparse DMD modes) (dashed lines) and high-fidelity (solid lines) (c) water cut for the reduced model (78 POD modes) (dashed lines) and high-fidelity (solid lines).

**Table 2.** Reservoir configuration.

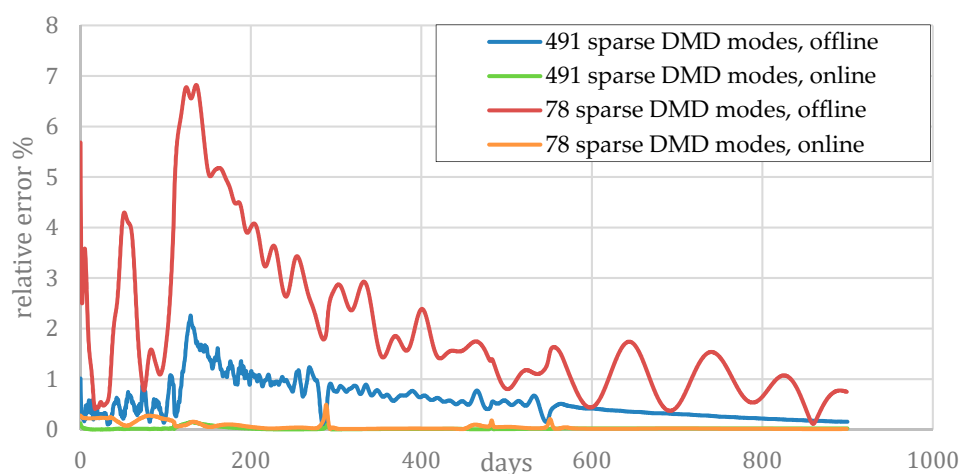
Grid Block	Grid Block Size	Porosity	Water Viscosity	Oil Viscosity	Injection Rate
220 × 60 × 3 BHP	10 ft × 20 ft × 2 ft	0.2	0.3	3	235 stb/day
2500 pisa	Total Injection 1 Pore volume	Initial Sw 0.25	Initial Pressure 3100 psia	Time Step Size 0.25 days	Total Time 1000 days

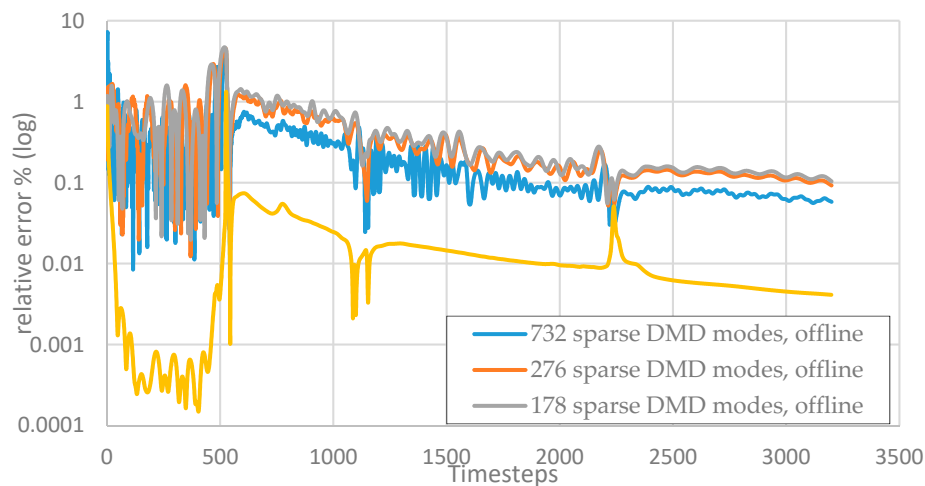
To use the sparse DMD method, first, we collect 1000 days of simulation results of pressure and velocity to build the snapshot matrix with dimensions of  $1,444,360 \times 4000$  (time interval is 0.25 days, velocity with dimension 104,760 and pressure with dimension 39,600). We then apply the sparsity-promoting DMD algorithm to obtain an optimal set of DMD basis. The basis could be used to reproduce pressure and velocity using offline process as in Equation (43). Total time for this offline process is often several seconds (for example the 3D with 178 basis takes 40 s) since it only involves with one step matrix multiplication, which is more efficient compared with the high-fidelity simulator run time. In Table 3, we present the overall relative error between the results from the offline model and fine-scale-model. The error is rather small, with a magnitude of  $10^{-3}$ .

**Table 3.** Relative error of pressure and velocity between high fidelity and reduced model.

Model	Solver Time (sec)	Relative Online Error%	Relative Offline Error%	Energy Factor for POD
211 Sparse DMD modes, 2D	9.64	0.093	2.3112	-
78 Sparse DMD modes, 2D	4.01	0.1277	3.0878	-
211 POD modes, 2D	10.14	0.1043	-	0.9999987
78 POD modes, 2D	6.28	0.497	-	0.9999646
Fine-scale simulator run, 2D	2013.00	-	-	-
178 sparse DMD modes, 3D	5.1364	0.0593	0.8268	-
Fine-scale simulator run, 3D	$1.7047 \times 10^4$	-	-	-

The online process using sparse DMD modes produces more accurate results than the POD method when using less number of modes as seen in the water cut comparison (Figure 9). With as few as 78 modes, sparse DMD method keeps high consistency with the reference solution, while POD modes lose some accuracy especially after the water breakthrough of the third well at around 480 days. Compared with the POD method, in general, the sparse DMD method could result in a smaller relative error (Table 2). Furthermore, the online case gives a better match than the offline process (Figures 10 and 11) in terms of relative error, however, the offline process is highly efficient in computation.

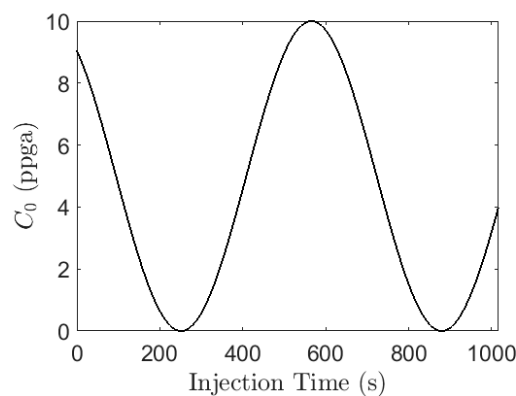
**Figure 10.** Relative error of pressure and velocity as the change of time in 2D reservoir.



**Figure 11.** Relative error of pressure and velocity as the change of time in 3D reservoir.

### 3.3. Hydraulic Fracturing Example

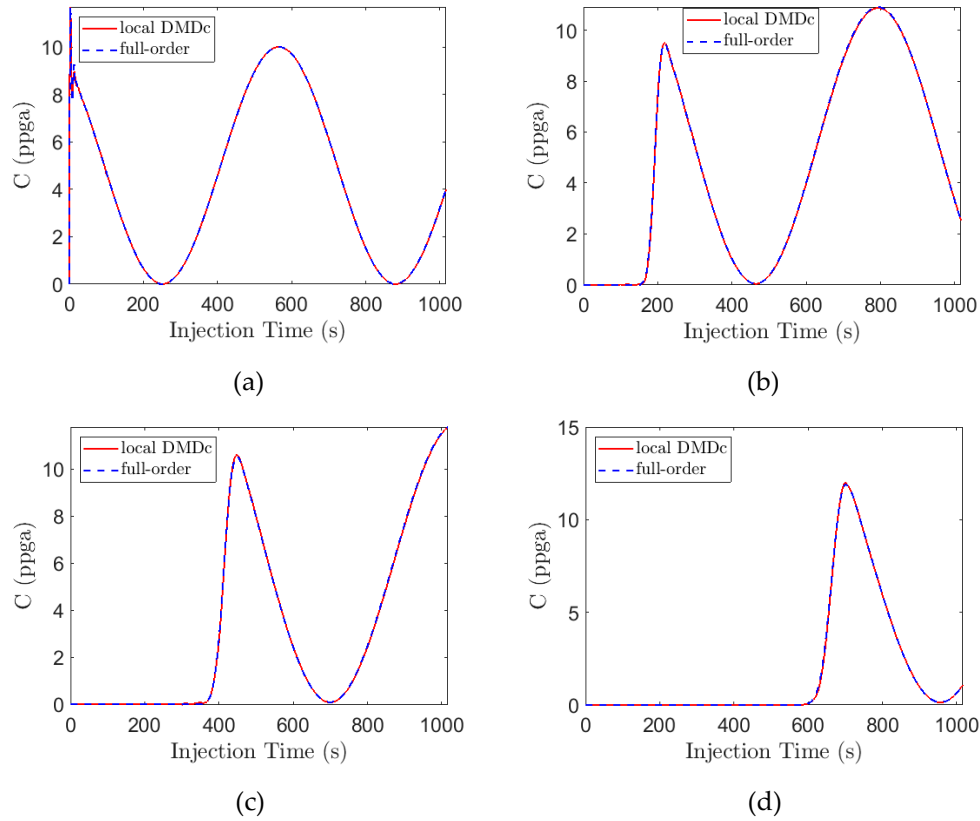
In this section, we show the results of local DMDc as applied to the hydraulic fracturing process. To solve the high-fidelity model, Equations (9)–(18), an adaptive finite-discretization method originally developed in Reference [44] and further improved in Reference [51] is used. The values of process parameters used in the calculations are obtained from Reference [52]. The injection flow rate of  $q_0 = 0.03 \text{ m}^3/\text{s}$  is used throughout the process. A sinusoidal profile shown in Figure 12 was used for the injected proppant concentration at the wellbore  $C_0(t)$ . The process was terminated when the fracture length reached 150 m. The full-order discretization resulted in a total of 501 spatial nodes and 12,201 snapshots.



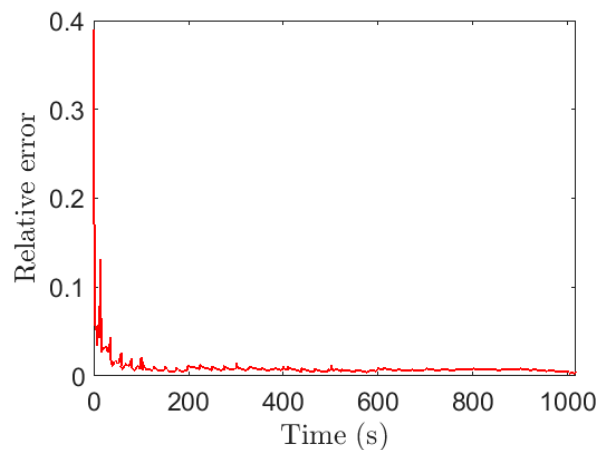
**Figure 12.** A sinusoidal input profile for injected proppant concentration.

We collect this data and construct the concentration matrices  $\mathbb{C}_1$ ,  $\mathbb{C}_2$  of size  $501 \times 12,200$  along with the input matrix  $\mathbb{C}_0$  which is of size  $1 \times 12,200$ . We then apply the GOS algorithm to partition the snapshots based on the similarity in their dynamical behavior. The GOS algorithm was implemented using GAMS which was interfaced with the CPLEX solver to cluster the data into  $m = 100$  optimal clusters. Within each cluster, we then apply the DMDc algorithm to compute the system matrices as in Equations (30) and (31) which could be used to approximate the high-fidelity proppant concentration data. The results of the model approximation are illustrated in Figure 13 which shows a comparison between the high-fidelity solution and the local DMDc solution computed using only 10 modes in each cluster. As we already noticed earlier [34], the local DMDc produces very accurate results in approximating the concentration dynamics in the hydraulic fracturing example; both the solution profiles almost overlap as can be seen in Figure 13. To quantify this approximation performance we

use relative error with time  $C_{err}(t_i)$  which can be computed in a similar manner as that of  $P_{err}(t_i)$  shown in the preceding section. The relative error plot is shown in Figure 14 which further illustrates the superior performance of local DMDc in producing accurate reduced-order representations of the hydraulic fracturing process.



**Figure 13.** Results comparing local DMDc and full-order solutions of concentration at 4 different locations inside the fracture. (a)  $C(t)$  at  $x = 18$  m; (b)  $C(t)$  at  $x = 36$  m; (c)  $C(t)$  at  $x = 72$  m; (d)  $C(t)$  at  $x = 108$  m.



**Figure 14.** Relative error of proppant concentration with time.

#### 4. Discussion on Data-Driven DMD

As presented before, the DMD method is a data-driven approach that post-processes the global snapshot matrix to identify the coherent structure associated with the flow behavior and approximates solution with the obtained modes. The overall relative error, as we can see in the tables, is less than

5% for as few as 78 DMD modes. The online process bears smaller error, however, it is thousands of times slower than the offline process (hours vs. seconds), which only involves one step of matrix multiplication Equation (43). Overall, the challenge of this data-driven approach mostly lies in the snapshot selection and the complex number involved in DMD algorithm Equation (31). The complex number involved in the DMD modes and eigenvalue indicates the frequency or decay of the modes, usually the imaginary part and real part of the eigenvalue needs to stay in the unit circle for stability purpose [27]. The imaginary part in DMD modes is not an issue for the offline process. In online DMD, when it comes to an upwinding strategy that involves the comparison of values, we need to make sure only the real part is utilized. In terms of a snapshot, one of the observations is that the column number of the snapshot, which is also the time step number, should be smaller than the number of the grid for the purpose of preserving the spatial information as much as possible. After the SVD of snapshot matrix Equation (29), we need to truncate the three matrix components  $Y$ ,  $S_\mu$ ,  $Z$  according to the rank of snapshot matrix. If the number of rows is less than the number of columns, part of the columns of  $Y$  that contains spatial information will be lost. However, this is usually not a problem if we deal with large systems. Another observation is that data in snapshot matrix should be on the same scale, otherwise the algorithm has a hard time converging. For instance, if one put velocity ( $10^{-8}$ ) and pressure ( $10^8$ ) in the same snapshot, they have to be properly scaled to be used in the algorithm. Other minor issue includes choosing a proper rank number that is relatively small but keeps enough dynamic information, avoid choosing too many snapshots after the system is at steady-state (e.g., after breakthrough).

With regards to local DMDc, although the reduced-order models obtained can be used for very fast online computations, the temporal clustering step, which is performed offline, usually requires a high computational cost. In the example presented, the total time taken by the GOS algorithm is almost the same as that of solving the high-fidelity model. However, this offline computation is performed only once and the resulting reduced-order models are in general more accurate; so there is a trade-off. Unlike in the sparsity DMD example, the online computation in local DMDc does not consider a Galerkin projection but rather use a discrete-time linear controlled type system which is described by the system matrices in Equations (40) and (41). This type of equation is more suitable for control applications for which the local DMDc was initially developed.

## 5. Conclusions

In this work, we presented applications of the sparsity-promoting DMD method to single and two-phase fluid flow modeling. Compared with standard DMD, the sparsity-promoting DMD can further reduce the number of modes required with a sparse structure solved by augmented Lagrange optimization with the objective of minimizing the difference between the actual snapshot and formulated results. The relative error is steadily reduced corresponding to more modes used. Within a reasonable range of time period, the offline stage with DMD or sparse DMD is faster than full-order or reduced order online modeling without compromise on the accuracy. In the online process, the most time-consuming solver part will speed up with the aid of Galerkin projection and calculated basis from model reduction algorithms. Furthermore, from the example, sparse DMD mode preserves more accuracy than POD modes especially when the number of modes is small. In the future, we will investigate the snapshot selection and complex number involved in the DMD method as well as seek the possibility of applying the methodology on a fully implicit framework.

**Author Contributions:** Conceptualization, E.G., A.B., J.S.K. and A.N.; writing—original draft preparation, A.B. and A.N.; writing—review and editing, E.G., J.S.K.

**Funding:** This research was funded by National Science Foundation (CBET-1804407).

**Acknowledgments:** E.G. and A.B. acknowledge financial support from Energi Simulation through the Energi Simulation Research Chair at Texas A and M. J.S.K. and A.N. acknowledge financial support from the National Science Foundation (CBET-1804407).

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

## References

- Guo, Z.; Reynolds, A.C.; Zhao, H. A physics-based data-driven model for history matching, prediction, and characterization of waterflooding performance. *SPE J.* **2018**, *23*, 367–395. [\[CrossRef\]](#)
- Guo, Z.; Reynolds, A.C.; Zhao, H. Waterflooding optimization with the insim-ft data-driven model. *Comput. Geosci.* **2018**, *22*, 745–761. [\[CrossRef\]](#)
- Lerlertpakdee, P.; Jafarpour, B.; Gildin, E. Efficient production optimization with flow-network models. *SPE J.* **2014**, *19*. [\[CrossRef\]](#)
- Salehi, A.; Hetz, G.; Olalotiti, F.; Sorek, N.; Darabi, H.; Castineira, D. A Comprehensive Adaptive Forecasting Framework for Optimum Field Development Planning. In *Proceedings of the SPE Reservoir Simulation Conference, Galveston, TX, USA, 10–11 April 2019*; Society of Petroleum Engineers: Galveston, TX, USA, 2019. [\[CrossRef\]](#)
- Liu, Z.; Forouzanfar, F.; Zhao, Y. Comparison of sqp and al algorithms for deterministic constrained production optimization of hydrocarbon reservoirs. *J. Pet. Sci. Eng.* **2018**, *171*, 542–557. [\[CrossRef\]](#)
- Liu, Z.; Reynolds, A.C. An sqp-filter algorithm with an improved stochastic gradient for robust life-cycle optimization problems with nonlinear constraints. In *Proceedings of the SPE Reservoir Simulation Conference, Galveston, TX, USA, 10–11 April 2019*; Society of Petroleum Engineers: Galveston, TX, USA, 2019. [\[CrossRef\]](#)
- Klie, H. Physics-based and data-driven surrogates for production forecasting. In *Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, USA, 23–25 February 2015*; Society of Petroleum Engineers: Houston, TX, USA, 2015; p. 18.
- Mohaghegh, S.D. *Shale Analytics: Data-Driven Analytics in Unconventional Resources*; Springer: Berlin, Germany, 2017.
- Klie, H.; Florez, H. Data-driven discovery of unconventional shale reservoir dynamics. In *Proceedings of the SPE Reservoir Simulation Conference, Galveston, TX, USA, 10–11 April 2019*; Society of Petroleum Engineers: Galveston, TX, USA, 2019; p. 24.
- Bao, A.; Gildin, E.; Zalavadia, H. Development of proxy models for reservoir simulation by sparsity promoting methods and machine learning techniques. In *Proceedings of the ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery, Barcelona, Spain, 3–6 September 2018*. [\[CrossRef\]](#)
- Narasingam, A.; Sang-Il Kwon, J. Data-driven identification of interpretable reduced-order models using sparse regression. *Comput. Chem. Eng.* **2018**, *119*, 101–111. [\[CrossRef\]](#)
- Antoulas, A.C. *Approximation of Large-Scale Dynamical Systems*; Siam: Philadelphia, PA, USA, 2005; Volume 6.
- Aziz, K. *Petroleum Reservoir Simulation*; Applied Science Publishers Ltd.: London, UK, 1979; Volume 476.
- Aarnes, J.; Krogstad, S.; Lie, K. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.* **2006**, *5*, 337–363. [\[CrossRef\]](#)
- Chen, Z.; Huan, G.; Ma, Y. *Computational Methods for Multiphase Flows in Porous Media*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006.
- Ghommam, M.; Gildin, E.; Ghasemi, M. Complexity reduction of multiphase flows in heterogeneous porous media. *SPE J.* **2016**, *21*, 144–151. [\[CrossRef\]](#)
- Van Doren, J.F.M.; Markovinović, R.; Jansen, J.-D. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comput. Geosci.* **2006**, *10*, 137–158. [\[CrossRef\]](#)
- Gildin, E.; Ghasemi, M.; Romanovskay, A.; Efendiev, Y. Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media. In *Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 18–20 February 2013*; Society of Petroleum Engineers: The Woodlands, TX, USA, 2013; p. 14.
- Tan, X.; Gildin, E.; Trehan, S.; Yang, Y.; Hoda, N. Trajectory-based deim tdeim model reduction applied to reservoir simulation. In *Proceedings of the SPE Reservoir Simulation Conference, Montgomery, TX, USA, 20–22 February 2017*; Society of Petroleum Engineers: Montgomery, TX, USA, 2017; p. 19.
- Cardoso, M.A.; Durlafsky, L.J. Use of reduced-order modeling procedures for production optimization. *SPE J.* **2010**, *15*, 426–435. [\[CrossRef\]](#)
- Trehan, S.; Durlafsky, L.J. Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to pde-constrained optimization. *J. Comput. Phys.* **2016**, *326*, 446–473. [\[CrossRef\]](#)
- Chaturantabut, S.; Sorensen, D.C. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764. [\[CrossRef\]](#)



23. Van Overschee, P.; De Moor, B. *Subspace Identification for Linear Systems: Theory—Implementation—Applications*; Springer: New York, NY, USA, 1996.
24. Ghommam, M.; Presho, M.; Calo, V.M.; Efendiev, Y. Mode decomposition methods for flows in high-contrast porous media. Global–local approach. *J. Comput. Phys.* **2013**, *253*, 226–238. [[CrossRef](#)]
25. Florez, H.; Gildin, E. Model order reduction applied to coupled flow and geomechanics. In Proceedings of the ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery, Barcelona, Spain, 3–6 September 2018.
26. Yang, Y.; Ghasemi, M.; Gildin, E.; Efendiev, Y.; Calo, V. Fast multiscale reservoir simulations with pod-deim model reduction. *SPE J.* **2016**, *21*, 2141–2154. [[CrossRef](#)]
27. Jovanović, M.R.; Schmid, P.J.; Nichols, J.W. Sparsity-promoting dynamic mode decomposition. *Phys. Fluids* **2014**, *26*, 024103. [[CrossRef](#)]
28. Rowley, C.W.; Mezic, I.; Bagheri, S.; Schlatter, P.; Henningson, D.S. Spectral analysis of nonlinear flows. *J. Fluid Mech.* **2009**, *641*, 115–127. [[CrossRef](#)]
29. Chen, K.K.; Tu, J.H.; Rowley, C.W. Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses. *J. Nonlinear Sci.* **2012**, *22*, 887–915. [[CrossRef](#)]
30. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [[CrossRef](#)]
31. Lusch, B.; Kutz, J.N.; Brunton, S.L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **2018**, *9*, 4950. [[CrossRef](#)]
32. Takeishi, N.; Kawahara, Y.; Yairi, T. Learning koopman invariant subspaces for dynamic mode decomposition. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Guangzhou, China, 14–18 November 2017*; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 1130–1140.
33. Williams, M.O.; Kevrekidis, I.G.; Rowley, C.W. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.* **2015**, *25*, 1307–1346. [[CrossRef](#)]
34. Narasingam, A.; Kwon, J.S.-I. Development of local dynamic mode decomposition with control: Application to model predictive control of hydraulic fracturing. *Comput. Chem. Eng.* **2017**, *106*, 501–511. [[CrossRef](#)]
35. Proctor, J.L.; Brunton, S.L.; Kutz, J.N. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* **2016**, *15*, 142–161. [[CrossRef](#)]
36. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
37. Narasingam, A.; Siddhamshetty, P.; Sang-Il Kwon, J. Temporal clustering for order reduction of nonlinear parabolic pde systems with time-dependent spatial domains: Application to a hydraulic fracturing process. *AIChE J.* **2017**, *63*, 3818–3831. [[CrossRef](#)]
38. Christie, M.A.; Blunt, M.J. Tenth spe comparative solution project: A comparison of upscaling techniques. *SPE Reserv. Eval. Eng.* **2001**, *4*, 308–317. [[CrossRef](#)]
39. Narasingam, A.; Siddhamshetty, P.; Kwon, J.S.-I. Handling spatial heterogeneity in reservoir parameters using proper orthogonal decomposition based ensemble kalman filter for model-based feedback control of hydraulic fracturing. *Ind. Eng. Chem. Res.* **2018**, *57*, 3977–3989. [[CrossRef](#)]
40. Bangi, M.S.F.; Narasingam, A.; Siddhamshetty, P.; Kwon, J.S.-I. Enlarging the domain of attraction of the local dynamic mode decomposition with control technique: Application to hydraulic fracturing. *Ind. Eng. Chem. Res.* **2019**, *58*, 5588–5601. [[CrossRef](#)]
41. Peaceman, D.W. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability. *Soc. Pet. Eng. J.* **1983**, *23*, 531–543. [[CrossRef](#)]
42. Nordgren, R.P. Propagation of a vertical hydraulic fracture. *Soc. Pet. Eng. J.* **1972**, *12*, 306–314. [[CrossRef](#)]
43. Perkins, T.K.; Kern, L.R. Widths of hydraulic fractures. *J. Pet. Technol.* **1961**, *13*, 937–949. [[CrossRef](#)]
44. Yang, S.; Siddhamshetty, P.; Kwon, J.S.-I. Optimal pumping schedule design to achieve a uniform proppant concentration level in hydraulic fracturing. *Comput. Chem. Eng.* **2017**, *101*, 138–147. [[CrossRef](#)]
45. Adachi, J.; Siebrits, E.; Peirce, A.; Desroches, J. Computer simulation of hydraulic fractures. *Int. J. Rock Mech. Min. Sci.* **2007**, *44*, 739–757. [[CrossRef](#)]
46. Daneshy, A.A. Numerical solution of sand transport in hydraulic fracturing. *J. Pet. Technol.* **1978**, *30*, 132–140. [[CrossRef](#)]
47. Kunisch, K.; Volkwein, S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.* **2002**, *40*, 492–515. [[CrossRef](#)]

48. Tu, J.H.; Rowley, C.W.; Luchtenburg, D.M.; Brunton, S.L.; Kutz, J.N. On dynamic mode decomposition: Theory and applications. *arXiv* **2013**, arXiv:1312.0041.
49. Tan, M.P.; Broach, J.R.; Floudas, C.A. A novel clustering approach and prediction of optimal number of clusters: Global optimum search with enhanced positioning. *J. Glob. Optim.* **2007**, *39*, 323–346. [[CrossRef](#)]
50. Bao, A.; Gildin, E. Data-driven model reduction based on sparsity-promoting methods for multiphase flow in porous media. In *Proceedings of the SPE Latin America and Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, 17–19 May 2017*; Society of Petroleum Engineers: Buenos Aires, Argentina, 2017; p. 16.
51. Siddhamshetty, P.; Yang, S.; Kwon, J.S.-I. Modeling of hydraulic fracturing and designing of online pumping schedules to achieve uniform proppant concentration in conventional oil reservoirs. *Comput. Chem. Eng.* **2018**, *114*, 306–317. [[CrossRef](#)]
52. Gu, Q.; Hoo, K.A. Evaluating the performance of a fracturing treatment design. *Ind. Eng. Chem. Res.* **2014**, *53*, 10491–10503. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).