J Petrol Explor Prod Technol (2014) 4:315–325
DOI 10.1007/s13202-013-0084-8

ORIGINAL PAPE - PRODUCTION ENGINEERING

Development of a surrogate simulator for two-phase subsurface flow simulation using trajectory piecewise linearization

Esmail Ansari

Received: 17 July 2013 / Accepted: 6 October 2013 / Published online: 22 October 2013

Abstract Reduced-order modeling (ROM) is a novel approach in all realms of computational science including reservoir simulation. Among various ROM methods, trajectory piecewise linearization (TPWL) is evolving for reservoir engineering applications. Previous investigations reflect promising future for incorporating TPWL into the next generations of enhanced reservoir simulators. In this work, we employ this method to examine the claimed efficiency, robustness and accuracy of it as a surrogate simulator. The self-construction of the used simulator gives us the opportunity to explore this method and to examine previous assertions on the subject. The efficiency of TPWL is primarily due to direct calculation of new saturation and pressure states using a linearized expansion around previously simulated states instead of traditionally solving the flow equations. For further efficiency and reduction of the required memory, TPWL method needs to accompany a space reducing scheme, through which the captured dynamic of the reservoir is projected into a lower-order space. The projection matrix is conventionally constructed through proper orthogonal decomposition (POD) of converged time stepping solutions known as 'snapshots' which are obtained during a serious of preprocessing runs called 'training' runs. In this work, we apply TPWL method to a hypothetical three-dimensional heterogeneous reservoir consisting of a compressible rock type. We assume an inverted five-spot production–injection pattern and present the results for a two-phase (oil–water) reservoir model under water flooding scenario, in which the injection well is controlled by injection rate. Achieved results demonstrate that use of TPWL leads to significantly faster simulation compared to high fidelity model. We achieved speedup of a factor of 120 while preserving accuracy and reliability of the results. This study suggests that TPWL methodology will be particularly attractive when many solutions of similar simulation models with different well settings are required for history matching or optimization problems. Future research should focus to assess the applicability of TPWL to conditions with strongly compressible flow or capillary pressure effects.

Keywords Trajectory piecewise linearization · TPWL · Proper orthogonal decomposition · POD · Surrogate simulator · Model order reduction · Reduced-order model · Two-phase flow

E. Ansari (✉)
Louisiana State University, Baton Rouge, USA
e-mail: eansar2@lsu.edu

Introduction

Development of reduced order models which accurately and efficiently represent the original model is a very crucial part of reservoir management. Optimization, history matching and optimal design of the reservoir require to run the forward model for many times. Though, parallel computing has proved to be effective, it still cannot provide adequate efficiency for reservoir management studies which deal with incorporating the real time data into the reservoir model.

Literature prospers from many cases on the application of proper orthogonal decomposition (POD) scheme to the various science disciplines (Antoulas and Sorensen 2005). For subsurface flow, however, its application is relevantly new (Markovinovic and Jansen 2006; Vermeulen et al. 2005; Van Doren et al. 2006) and culminates in studies done by Cardoso et al. (2009) and Cardoso and Durlofsky (2009, 2010).

POD method, also known as principal component analysis (PCA), is designed to capture the most dominant features of a dynamical system and to identify its coherent structure (Astrid and Papaioannou 2011). POD is a very popular method for reducing nonlinear large-scale systems. This method is designed to project the system to a subspace spanned by a small number of vectors known as 'basis functions'. The projection subspace should be small and reflect the most dominant and relevant features of the high fidelity system. The 'basis' functions should be able to represent the most probable realizations of the input. To attain this objective, the target boundary conditions (input well pressures and rates) should be in the domain of the training boundary conditions which should also be 'exited' to comprise a variety of input scenarios. We note that He et al. (2011) have recently been successful to address some schemes to mitigate the severity of these limitations, though we did not implemented them here.

Trajectory piecewise linearization (TPWL) was introduced by Rewienski and White (2003) who applied it for a nonlinear transmission line circuit model. In the reservoir engineering context, Cardoso et al. (2009) and Cardoso and Durlofsky (2010) pioneered publishing the development of TPWL formulation for two-phase reservoir flow and later He et al. (2011) addressed some of its foibles and enhanced the basic TPWL method to meet a wider range of conditions.

TPWL provides a means to calculate the new simulation results by directly using previously saved results, which are projected to a lower space. In this method, first some high fidelity simulations known as 'training runs' are performed, from which the time stepping converged states (solution of saturation and pressure known as 'snapshots') and Jacobian matrices are saved. Then, POD method is applied to the saved states to construct a matrix called 'basis' matrix. Using this matrix, the saved results are projected to a lower space; therefore, the reduced states and reduced Jacobians are formed. For subsequent simulations, a linear expansion around previously saved states is carried out and new states are directly calculated using those states.

This paper proceeds as follows. We first concisely introduce the flow equation and POD procedure, and then we shortly describe linearization of the flow equations and concisely review incorporation of the POD method into TPWL. We refer to cited publications for more detailed description. Finally, we represent our case study and attained solutions.

Problem formulation

The governing equations for oil–water flow and TPWL procedure are formulated in detail in Cardoso et al. (2009). For completeness and because of some differences in our formulation and implementation to that of Cardoso et al. (2009), Cardoso and Durlofsky (2010) and He et al. (2011), we have included required formulas here too.

Flow equations

By combining mass conservation and Darcy's law, the equation of two-phase (oil and water) flow in porous media is expressed as Eq. (1), in which subscript j is used to designate the phase ($j = o$ for oil and w for water):

$$\nabla.[\lambda_i \mathbf{k}(\nabla p_j - \rho_j g \nabla D)] = \frac{\partial}{\partial t}\left(\frac{\phi S_j}{B_j}\right) + q_j^w \tag{1}$$

In the Eq. (1), $\mathbf{k}$ is the absolute permeability tensor, $\lambda_j = k_{rj}/\mu_j$ is the phase mobility, with k_{rj} the relative permeability to phase j and μ_j is the phase viscosity, g is gravitational acceleration, p_j is phase pressure, ρ_j is the phase density, D is depth, t is time, S_j is saturation and q_j^w is the source/sink term. Equation (1) is coupled by the saturation constraint $(S_o + S_w) = 1$ and by specifying a capillary pressure relationship $p_c\ (S_w) = p_o + p_w$.

The fully implicit formulation of flow equations in the three-dimensional space can be summarized as Eq. (2).

$$\mathbf{g}(\mathbf{x}^{n+1}, \mathbf{u}^{n+1}) = \mathbf{F}(\mathbf{x}^{n+1}) + \mathbf{A}(\mathbf{x}^{n+1}, \mathbf{x}^n) + \mathbf{Q}(\mathbf{x}^{n+1}, \mathbf{u}^{n+1}), \tag{2}$$

where $\mathbf{g}$ is the residual vector, $\mathbf{x}$ represents the state of the system (pressure and saturation), $\mathbf{u}$ represents the system controls (bottom hole pressure and injection rates) and $\mathbf{F}, \mathbf{A}$ and $\mathbf{Q}$, respectively stand for convection flow, accumulation rate and injection/production rates.

POD procedure

For developing a POD matrix and to project the model into low-dimensional space using this matrix, the high fidelity model should be run at least once (training run), during which snapshots of the converged time stepping answers (pressure and saturation) of all the grid blocks will be saved. We represent the number of snapshots as S, the collection of pressure snapshots as $\mathbf{X}_p$ and the collection of saturation snapshots as $\mathbf{X}_s$. It is also conventional to subtract the time-averaged value of snapshots from the data, although we did not implement this subtraction (Astrid and Papaioannou 2011).

$$\mathbf{X}_p = [x_p^1, x_p^2, , x_p^S]_{N_c * S} \tag{3}$$

$$\mathbf{X}_s = [x_s^1, x_s^2, , x_s^S]_{N_c * S} \tag{4}$$

The columns of the above equations are in the form of x_s^i, where i indicates the time step. The POD method is then

applied to each of these matrices to construct a basis for projection. POD basis is a solution to an optimization problem subject to orthonormality constraints of the basis vectors. It minimizes the reconstruction of state vectors in a least squares sense with the minimum required basis vectors for constructing the reduced subspace. It is proved that this is equivalent to solving the eigenvalue problem (5), where $\mathbf{C}$ is the covariance matrix of the snapshot which is calculated using Eq. (5). Equation (5) can be solved using a singular value decomposition (SVD) scheme which is the main part of POD methodology.

$$\mathbf{C} = \mathbf{X}^T\mathbf{X} \tag{5}$$

$$\mathbf{C}\Psi = \lambda\Psi, \tag{6}$$

where Ψ stands for the eigenvectors and λ for eigenvalues of the covariance matrix. However, we never build a covariance matrix, and for calculating the eigenvectors and eigenvalues of $\mathbf{C}$, a Singular Value Decomposition of the snapshots is performed because the eigenvalues of $\mathbf{X}(\sigma_i)$ are related to the eigenvectors of the covariance matrix through $\sigma_i = \lambda_i^{1/2}$. The magnitude of every eigenvalue associated with each eigenvector determines its corresponding impact on construction of the basis of the lower space. This magnitude is sometimes referred to as the 'energy' of the eigenvector. Hence by eliminating the eigenvectors that have less impact on the lower space construction (their eigenvalues are low) and by projecting the model into the lower space using the remained eigenvectors, we are reducing the energy of the system. Then, the remaining energy of the system can be calculated using $E_l = \sum_{i=1}^{l}\lambda_i/E_t$ where λ_i shows the energy of its corresponding eigenvectors and $E_t = \sum_{i=1}^{S}\lambda_i$ is the total energy of the system that corresponds to l largest eigenvalues.

After reducing the energy of the system, the columns of attained matrix Φ_l are called basis functions which are used to project the variables into the lower space, using Eq. (7).

$$\mathbf{x} \approx \Phi_l\mathbf{z}. \tag{7}$$

Note that this process is done separately for saturation and pressure states since they are physically independent variables. The reduced state $\mathbf{z}$ can now be used for reducing the number of variables in the flow equation, so that we have only $l = l_p + l_s$ variables instead of $2N_c$.

TPWL procedure

The principal concept in TPWL procedure is to use a Taylor expansion around previously saved states during preprocessing runs (or so-called training runs) and to find a linear model which best represents the original model. Hence it would be possible to directly (without iteration) find the solution of the nonlinear Eq. (2) using the linear model for any given controlling parameter quite different from that of training runs. The expansion of the Eq. (1) around the saved states $(\mathbf{x}^{i+1}, \mathbf{x}^i)$ and controlling parameters $\mathbf{u}^i$ gives Eq. (8).

$$\mathbf{g}^{n+1} = \mathbf{g}^{i+1} + \frac{\partial \mathbf{g}^{i+1}}{\partial \mathbf{x}^{i+1}}(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}) + \frac{\partial \mathbf{g}^{i+1}}{\partial \mathbf{x}^{i}}(\mathbf{x}^{n} - \mathbf{x}^{i}) + \frac{\partial \mathbf{g}^{i+1}}{\partial \mathbf{u}^{i+1}}(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}) \tag{8}$$

In Eq. (8), $\mathbf{x}^{n+1}$ stands for the state that is supposed to be found for the controlling parameter $\mathbf{u}^{n+1}$ and $\mathbf{x}^{i+1}, \mathbf{u}^{i+1}$, respectively, showing the nearest saved state and controlling parameter to $\mathbf{x}^{n+1}, \mathbf{u}^{n+1}$ during the training runs. $\mathbf{x}^n$ is the current state and the nearest state to it is designated by $\mathbf{x}^i$. Also in this equation $\mathbf{g}^{n+1}$ and $\mathbf{g}^{i+1}$ are, respectively, abbreviated forms for $\mathbf{g}(\mathbf{x}^{n+1}, \mathbf{x}^n, \mathbf{u}^{n+1})$ and $\mathbf{g}(\mathbf{x}^{i+1}, \mathbf{x}^i, \mathbf{u}^{i+1})$. After expanding all terms of Eq. (1) around the saved points and substituting them into Eq. (8), the final equation of linearization in the high dimension becomes as Eq. (9):

$$\mathbf{J}^{i+1}(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}) = -[\mathbf{F}^{i+1} + \mathbf{A}^{i+1} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}(\mathbf{x}^n - \mathbf{x}^i) + \mathbf{Q}(\mathbf{x}^{i+1}, \mathbf{u}^{n+1})] \tag{9}$$

in which the Jacobian matrix and other abbreviations are defined by:

$$\mathbf{J}^{i+1} = \frac{\partial \mathbf{F}^{i+1}}{\partial \mathbf{x}^{i+1}} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i+1}} + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{x}^{i+1}} \tag{10}$$

$$\mathbf{A}^{i+1} = \mathbf{A}(\mathbf{x}^{i+1}, \mathbf{x}^i), \mathbf{F}^{i+1} = \mathbf{F}(\mathbf{x}^{i+1}), \mathbf{Q}^{i+1} = \mathbf{Q}(\mathbf{x}^{i+1}, \mathbf{u}^{i+1}). \tag{11}$$

This is still suboptimal, because for saving the Jacobian and other expansion matrices, a large amount of memory is required. Hence, in practice, for having a much higher speed and reducing the required memory, TPWL is always coupled with a space reducing scheme. Here, we use POD to attain this objective. By incorporating Eq. (7) into Eq. (9), and multiplying both sides into Φ^T, we can approximate $\mathbf{x}$ by projecting it into the lower space and reduce the model equations from $2N_c$ to l.

$$\Phi^T\mathbf{J}^{i+1}\Phi(\mathbf{z}^{n+1} - \mathbf{z}^{i+1}) = -\Phi^T[\mathbf{F}^{i+1} + \mathbf{A}^{i+1} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\Phi(\mathbf{z}^n - \mathbf{z}^i) + \mathbf{Q}(\mathbf{x}^{i+1}, \mathbf{u}^{n+1})] \tag{12}$$

In Eq. (12), $\mathbf{z}^{n+1}$ is the state that we seek to find, $\mathbf{z}^n$ is its previous state that we have found using TPWL and, $\mathbf{x}^{i+1}$ and $\mathbf{x}^i$ are, respectively, the nearest states to $\mathbf{x}^{n+1}$ and $\mathbf{x}^n$. After finding the states in the lower space, Eq. (7) can again be used to project the solution back into the original

high dimensional space. This concludes our description of TPWL and POD. We note that our implementation here differs from that of previous investigators Cardoso et al. (2009), Cardoso and Durlofsky (2009a, 2010) and He et al. (2011) in using Eq. (12) for calculating the next state in TPWL procedure. This implementation was easier and more straightforward for our solver. Also, we have set the controlling parameter of the injection well on the rate of injection instead of pressure as this is the case in real problems.

Application: three-dimensional water flooding scenario

A hypothetical reservoir consisting of a compressible rock type and operating through inverse five-spot pattern is considered. The assumed model is three-dimensional and consists of 147 grid blocks with $N_x = 7$, $N_y = 7$ and $N_z = 3$, where N_x, N_y and N_z represent the number of grid blocks in each spatial direction. Figure 1 shows the position of the injection and production wells. The model consists of four production wells on the edges of the reservoir (designated as well 1–4) and one injection well at the center (designated as well 5). The wells are perforated only in the first layer and considered to be Peaceman vertical type with the radius of 0.1 *ft* (0.03048 *m*). Permeability field is taken to be isotropic and heterogeneous and vary between $1 - 5\text{md}(9.869 \times 10^{-16} - 4.934 \times 10^{-15}\text{m}^2)$ (Fig. 2). The permeability in each layer is separate from other layers and we use a Sequential Gaussian Simulation methodology for generating them. The initial oil and water saturations are considered 0.8 and 0.2, respectively, and the residual oil saturation (S_{or}) and residual water saturation

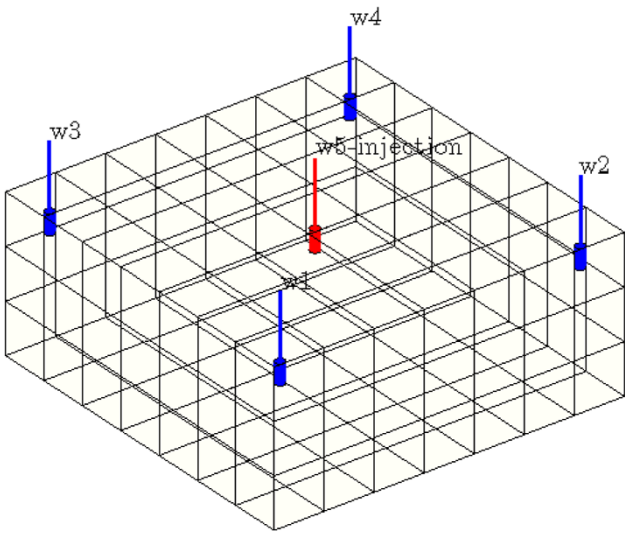

Fig. 1 Hypothetical reservoir and its wells

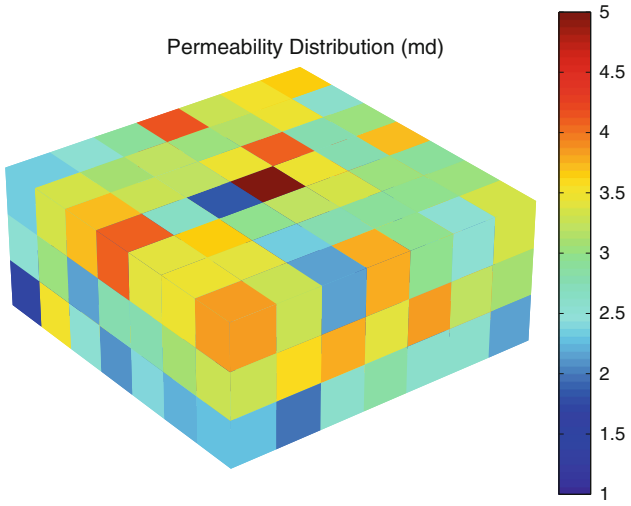

Fig. 2 Gaussian permeability field

(S_{wr}) are 0.2. The oil and water densities are $\rho_o = 45$ lb/ft^3 (721 kg/m^3) and $\rho_w = 60$ lb/ft^3 (961 kg/m^3). Capillary forces are disregarded and the relative permeabilities for the oil and water phases are determined using Corey equations:

$$k_{ro}(S_w) = k_{ro}^0 \left(\frac{1 - S_w - S_{or}}{1 - S_{wr} - S_{or}} \right)^a \tag{13}$$

$$k_{rw}(S_w) = k_{rw}^0 \left(\frac{S_w - S_{wr}}{1 - S_{wr} - S_{or}} \right)^a \tag{14}$$

in which k_{ro}^0 and k_{rw}^0 are the endpoint relative permeabilities. Here, we set $k_{ro}^0 = k_{rw}^0 = 1$ and $a = b = 2$. In addition, the rock compressibility is assumed to be 10^{-5}Psi^{-1} $(1.45 \times 10^{-9}\text{Pa}^{-1})$.

Training run

For establishing the training runs, we constructed a simple fully implicit reservoir simulator for representing the high fidelity model. In the next step, we validated our 3D two-phase simulator with two commercial reservoir simulators (CMG IMEX and ECLIPSE 100).

For the case of our study, the high fidelity model is run only once for both determining the reduced-order subspace and saving the required data for the linearized model. The snapshots were collected by simulating the reservoir while dynamically exiting it by frequently changing the well controllers. Our controlling parameters are set to bottom hole pressure (BHP) for production wells and water injection rate for injection well. The controlling parameters are set in a quite random way. These parameters are generated and updated every 100 days using a random function in MATLAB. For cases that target controllers are

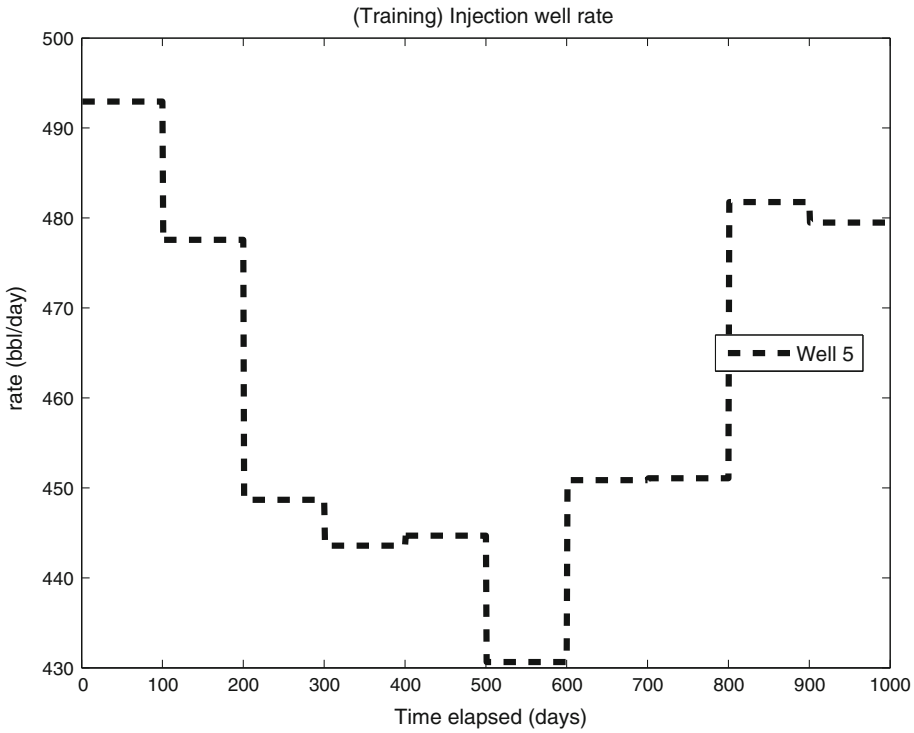

Fig. 3 Injection well's rate schedule for training run

constructed by linearly disturbing the training controllers, TPWL method leads to very identical results to high fidelity simulation. However, unlike previous investigators, we have implemented random methods here instead of heuristically making the input of training and target runs. Hence quite different and independent controllers for the training and targeting runs are used. The bottom hole pressure of the production wells is changed in the domain 3,000–4,500 psi (20.684–31.026 MPa) and the injection rate of the water is altered in the domain 400–500 bbl/day (63.59–79.49 m^3/day). Note that the simulator is coupled with a random generator function and hence completely different scenarios happen for the training and target runs. We used our personal computer for running the model and we simulated the performance of the reservoir up to 1,000 days, with the time step of 1 day.

Figures 3 and 4 show the training run schedules used for calculating basis functions. This elaborate schedule may seem an extreme assumption to happen in practice, but our reasons for making this scenario are twofold. Firstly, it tests the performance of TPWL methodology in case of multiple transients and secondly changing production and injection controllers plays an important role in determining viable production scenarios which typically arise in flooding optimization problems. The reduced energy from the system is 10^{-12} % of the total energy of the system. This is a heuristic measure and currently there is no established predictive method to guarantee measuring how much the corresponding reduced-order model would deviate from high fidelity model, beforehand. In Figs. 5 and 6, the circles highlight the selected eigenvectors. As the figures show, even this very little reduction in the energy of the system results in the elimination of a large number of eigenvalues in the SVD scheme. Larger eigenvalues are associated with eigenvectors that have a greater impact in capturing the dynamic of the reservoir. The eigenvectors corresponding to these remained eigenvalues are used for calculating basis functions by which the model is projected into the lower space. The number of required basis functions depends on the complexity of the reservoir and its controllers. The degree of nonlinearity of the model, which may originate from relative permeability correlations or compressibility of the reservoir fluid, has a great impact on the complexity of the model. When the reservoir fluids have small or no compressibility (which is the case in water flooding scenarios), the number of the selected eigenvalues is small. This directly depends on how much the pressure and saturation equations can be decoupled.

For this example, the number of selected eigenvalues for saturation and pressure is 51 and 31 out of 147, respectively; therefore, the model unknowns are reduced from 294 to 81. For more complicated reservoirs, two or more training runs may be employed but in our studies, the choice of one training run was resulting in reasonable accuracy and efficiency, so we used only one training run here.

Figures 7 and 8 depict the production and injection schedules for the target run. Similar to training run, the controllers are changed quite randomly and every 100

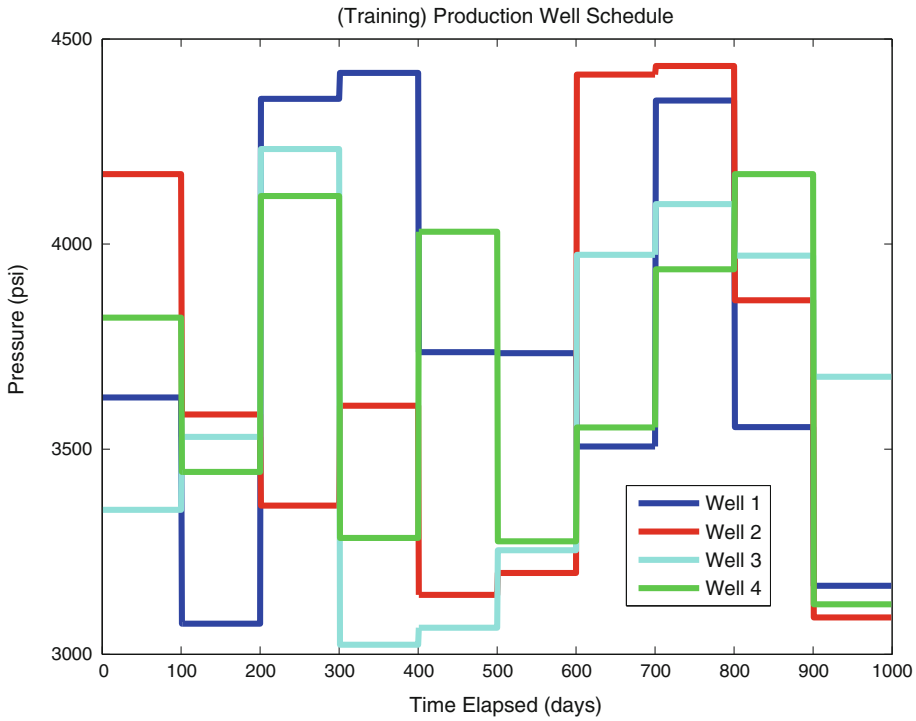

Fig. 4 Production wells' pressure schedule for training run

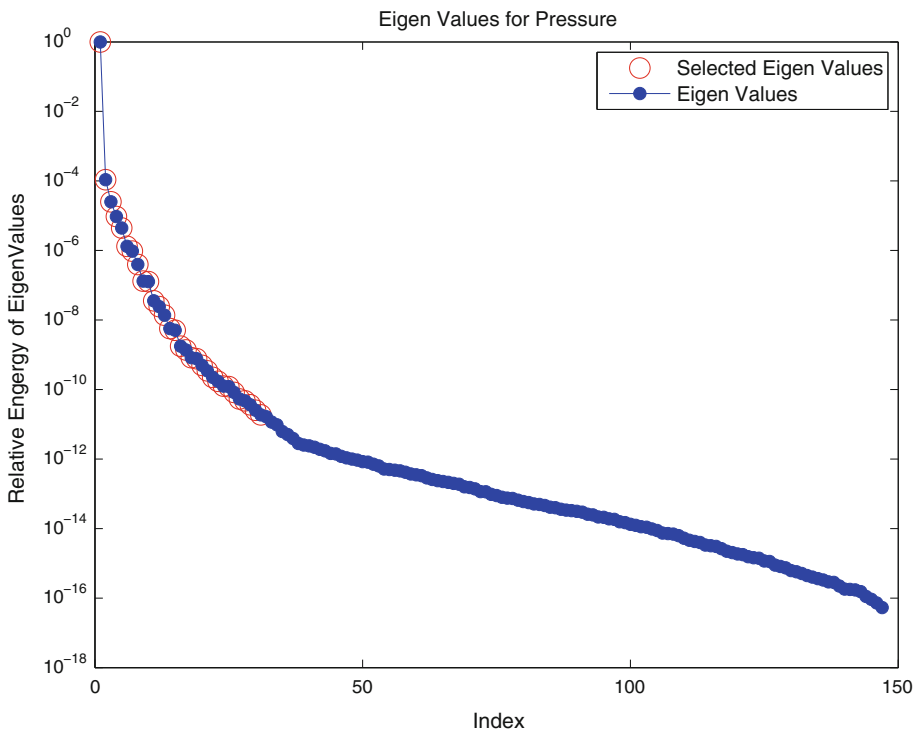

Fig. 5 Selected eigenvalues for pressure

days; therefore, we expect different scenarios for training and target schedules.

Figures 9, 10, 11 and 12 illustrate the robustness and accuracy of TPWL and POD methods in simulating the production rates for target schedule. Oil and water production rates of TPWL and high fidelity method show close agreements. We note that we can observe discrepancies around breakthrough time in each well. This discrepancy sometimes makes the reduced order model to blow up. We have found that the accuracy of Jacobian matrix and other linear model parameters around breakthrough time has a great impact on mitigating this problem. We believe time

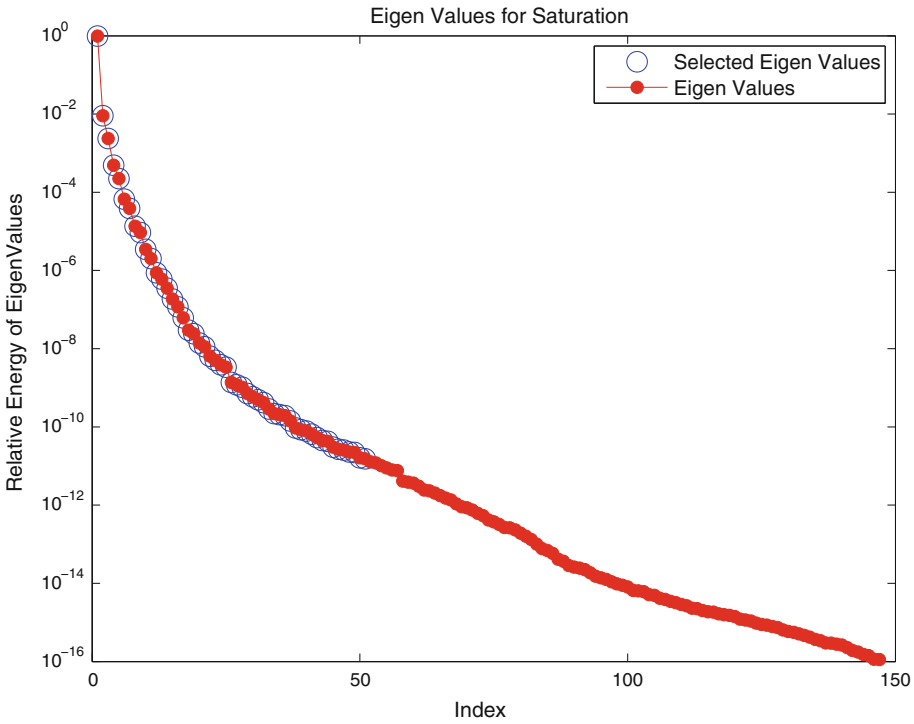

Fig. 6 Selected eigenvalues for saturation

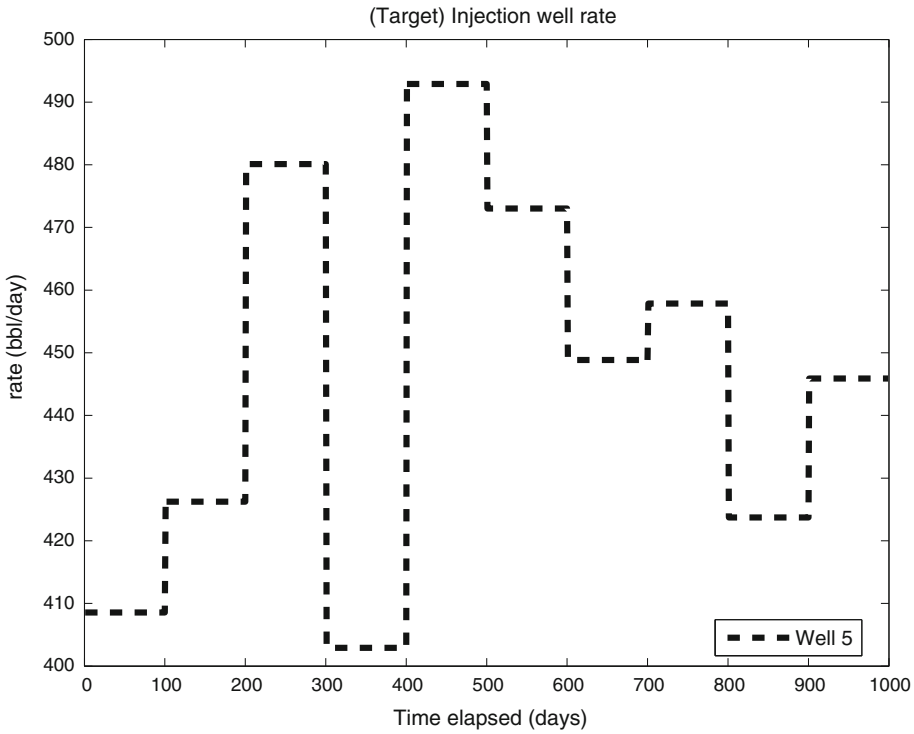

Fig. 7 Injection well's rate schedule for target run

step cutting, which is a feature of commercial reservoir simulators when the residuals are oscillating, would be effective in overcoming this problem. In our developed simulator, this feature was not available and we had to employ small time steps for obtaining accurate Jacobians and other linear model parameters around breakthrough time. The reason of this discrepancy is not completely clear to us. We attributed this to changes in production wells equations, which take place at breakthrough time. This encourages for more accurate estimation of Jacobian matrixes around breakthrough time. We also observed that using TPWL before breakthrough time or using it after

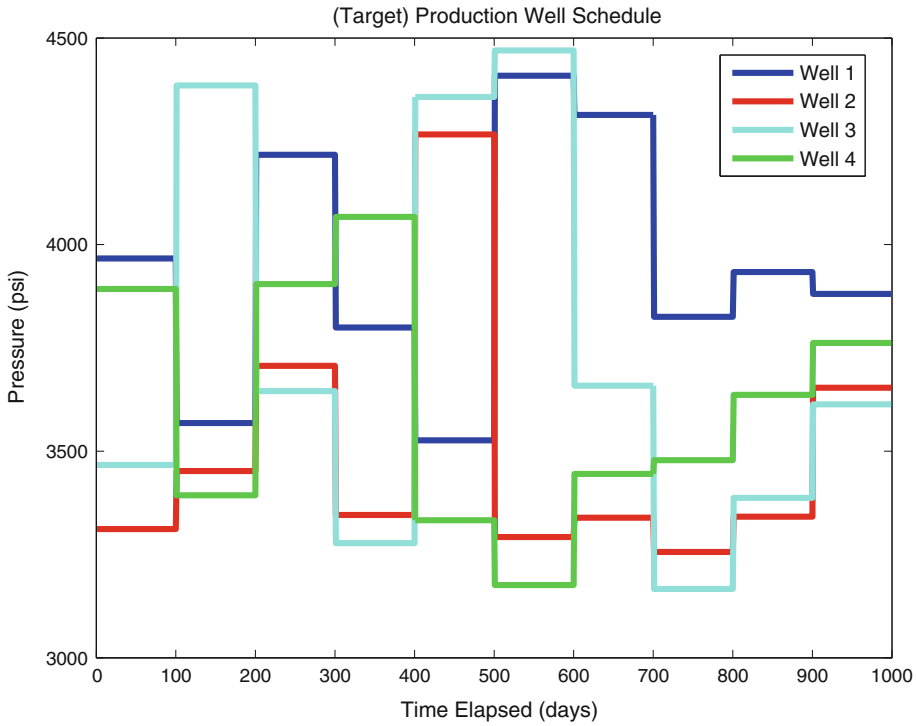

Fig. 8 Production wells' pressure schedule for target run

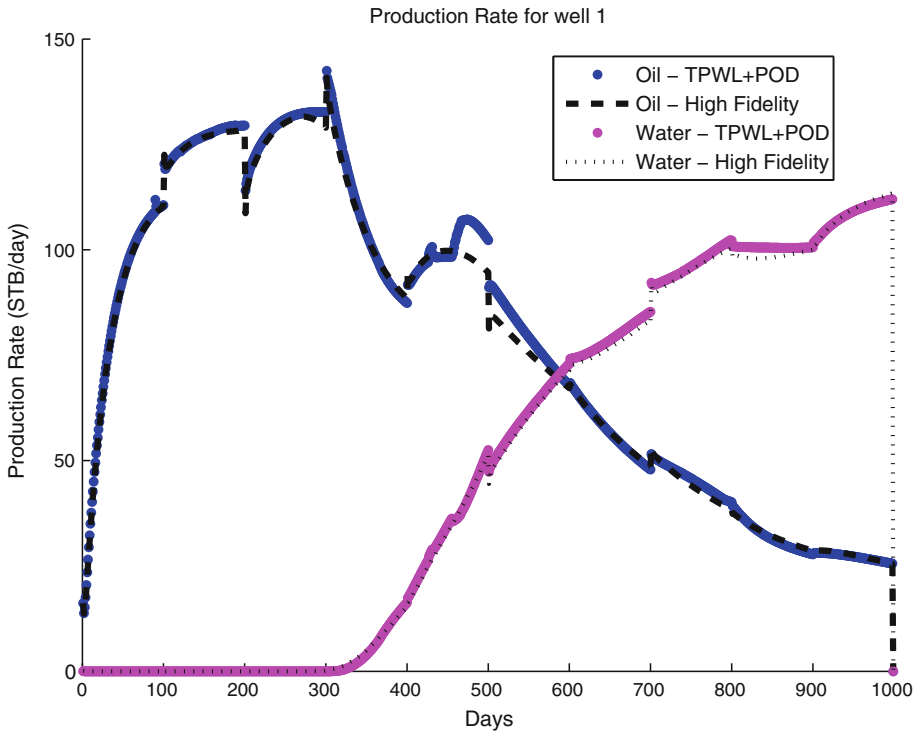

Fig. 9 Production rate for well 1

breakthrough time results in completely identical results to that of high fidelity model, therefore, a more realistic and challenging scenario was presented here and solved.

Figure 13 shows bottom hole pressure for the injection well. The result of high fidelity and TPWL methods is comparable, though local significant errors can, again, be clearly observed. These errors in water injection BHP take place at around production wells breakthrough times.

$$E_p rd^m = \frac{1}{n_t \tilde{Q}^m_{o,\mathrm{hf}}} \sum_{i=1}^{n_t} \left| Q^{m,i}_{o,\mathrm{hf}} - Q^{m,i}_{o,\mathrm{tpwl}} \right| \tag{15}$$

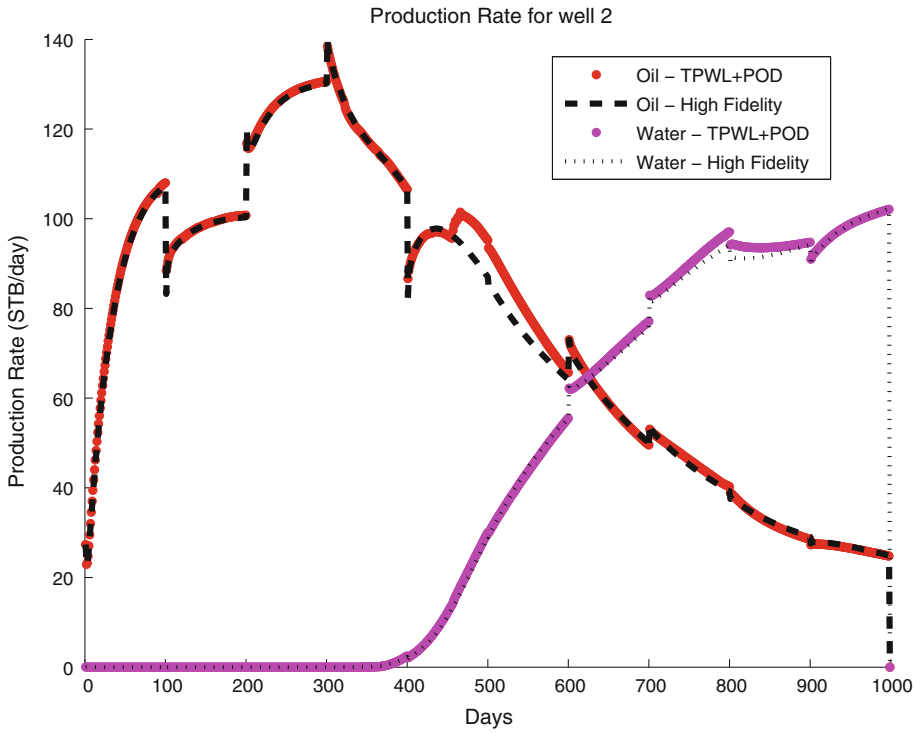

Fig. 10 Production rate for well 2

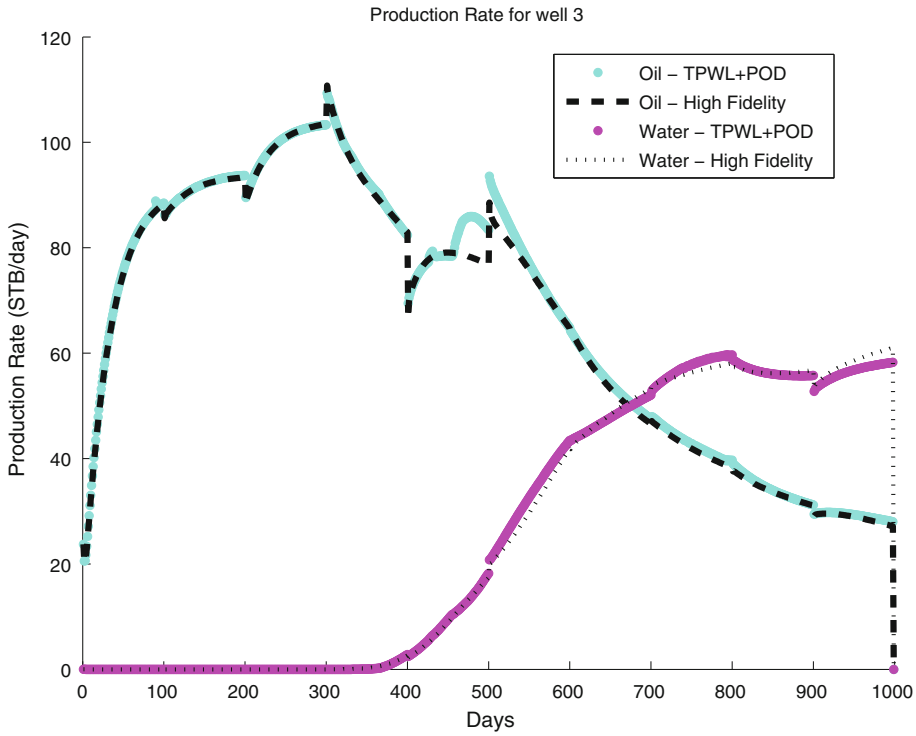

Fig. 11 Production rate for well 3

$$E_p = \frac{1}{n_w}\sum_{m=1}^{n_w} E_{prd}^{m} \tag{16}$$

The accuracy of TPWL can be evaluated visually by comparing the production rate or injection pressure figures. Also, we can compute the average production error using formula 15 and 16 (Cardoso et al. 2009).

In Eq. (15), i stands for time step, m stands for the number of production well, o represents the producing oil phase and subscripts hf and tpwl show the results for

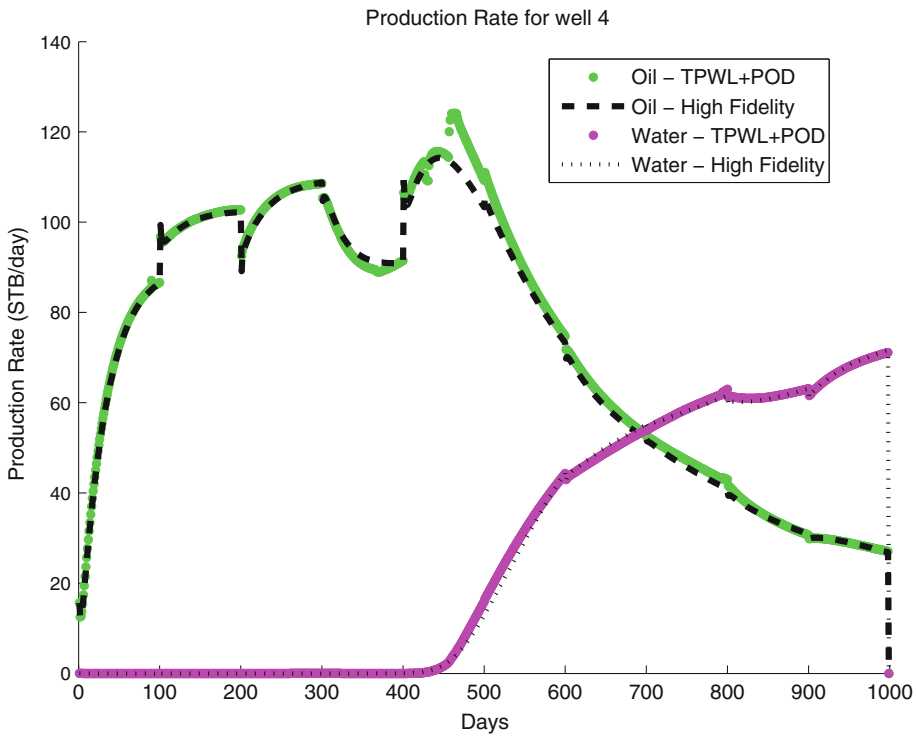

Fig. 12 Production rate for well 4

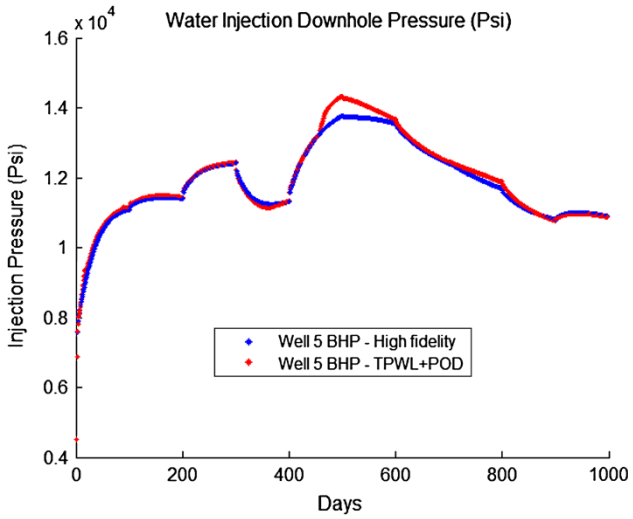

Fig. 13 Water injection BHP

high fidelity and TPWL, respectively. Also $\tilde{Q}$ indicates the average production for each production well in the high fidelity model. The overall average error of all wells is then computed by Eq. (16). The same procedure can be applied for water phase or injection well pressure. For calculating the error of the injection pressure, the bottom hole pressure of the injection well should be substituted instead of production rate in Eq. (15). Table 1 shows calculated error for important parameters using Eq. (16).

Table 1 Calculated errors for important parameters

Parameter	Oil production	Water production	Injection pressure
Error	0.017181	0.037288	0.0091

We further note that the high fidelity recovery factor of the reservoir after 1,000 days of water flooding is 0.4559 and the reduced model recovery factor is 0.4621.

Figure 14 shows the efficiency of TPWL which implements POD as its space reducing scheme. The required time for training run includes the time needed for a high fidelity run and the time required for determining the reduced space basis functions. From the above figure, it can be deduced that the time required for computing basis functions is about 80 s.

The reader may surprise to see a significant high fidelity simulation time for the small case study presented here; the reason for this, at least partially, is that our implementation is not generally efficient compared to commercial simulators which utilize efficient solvers particularly linear solvers.

Although we have not tested TPWL for realistic reservoirs containing numerous grids because of our limitations, but our tests for various small cases indicate that high fidelity simulation is largely affected by number of grid blocks but TPWL representation is less sensitive to the dimension of the model. This is because the linear model consists of multiplication and summation of matrices which takes only a few seconds for calculation (inverse of

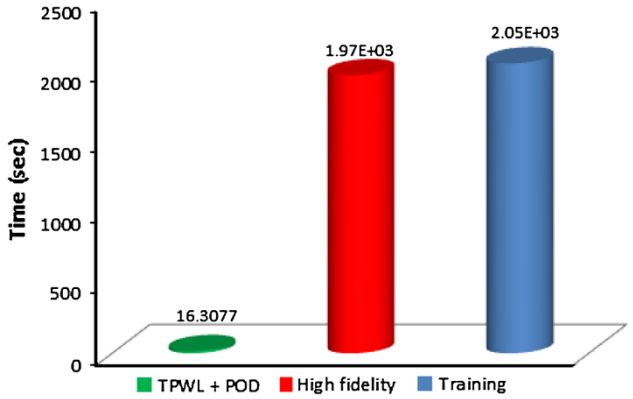

Fig. 14 Comparing the time for various methods

converged Jacobian matrix can be directly saved during training run).

Conclusion

1. We developed a surrogate simulator with trajectory piecewise linearization and used it for efficiently simulating hypothetical oil reservoirs under water flooding scenarios. In this method, the traditional fully implicit discretization of flow equations is replaced by a linear model, which eliminates the demand for using solvers for subsequent runs.
2. For further efficiency, TPWL has to accompany a space-reducing scheme, which is used for reducing the number of model unknowns by projecting it into a lower space. For achieving this objective, POD was successfully used in this study.
3. Achieved results from TPWL and high fidelity model show close agreement and robustness of TPWL model.
4. TPWL method needs to be developed and further tested for situations under capillary forces, strong gravity differences between phases or compressible reservoir fluids which encompass several state variables because in these cases, the stability of TPWL deteriorates and requires prescribing enhanced mathematical methods.

Acknowledgments I gratefully thank Jincong He and Marco Cardoso for E-mail communications regarding the implementation of TPWL and POD. Further, I would like to express my gratitude to Mahmoud Reza Pishvaie and Ramin Bozorgmehri for providing excellent advice and facilitating my professional growth during work on this subject.

References

Antoulas A, Sorensen D (2005) Approximation of Large-scale dynamical systems. SIAM publication, Philadelphia

Astrid P, Papaioannou G (2011) Pressure preconditioning using proper orthogonal decomposition. In: SPE reservoir simulation symposium. The Woodlands, Texas, pp 1–12

Cardoso MA, Durlofsky LJ (2009) Linearized reduced-order models for subsurface flow simulation. J Comput Phys 229:681–700

Cardoso MA, Durlofsky LJ (2010) Use of reduced-order modeling procedures for production optimization. SPEJ 15(2):426–435. doi:10.2118/119057-PA

Cardoso M, Durlofsky LJ, Sarma P (2009) Development and application of reduced-order modeling procedures for subsurface flow simulation. Int J Numer Methods Eng 77:1322–1350

He J, Saetrom J, Durlofsky LJ (2011) Enhanced linearized reduced-order models for subsurface flow simulation. J Comput Phys 230:8313–8341

Markovinovic R, Jansen J (2006) Accelerating iterative solution methods using reduced-order models as solution predictors. Int J Numer Methods Eng 68:525–541

Rewienski M, White JA (2003) A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol 22, pp 155–170

van Doren JF, Markovinovic R, Jansen J-D (2006) Reduced-order optimal control of water flooding using proper orthogonal decomposition. Comput Geosci 10:137–158

Vermeulen P, Heemink A, Valstar J (2005) Inverse modeling of groundwater flow using model reduction. Water Resour Res 41, W06003