

Machine Learning Applied to 3-D Reservoir Simulation

Marco A. Cardoso

1 Introduction

The optimization of subsurface flow processes is important for many applications including oil field operations and the geological storage of carbon dioxide. These optimizations are very demanding computationally due to the large number of flow simulations that must be performed and the typically large dimension of the simulation models.

In this work, reduced-order modeling (ROM) techniques are applied to reduce the simulation time of complex large-scale subsurface flow models. The procedures all entail proper orthogonal decomposition (POD), in which a high fidelity training simulation is run, solution snapshots are stored and an eigen-decomposition is performed on the resulting data matrix. A clustering procedure to reduce the size of the eigen-decomposition problem and the resulting number of degrees of freedom is also implemented. Extensive flow simulations involving water injection into a geologically complex 3D oil reservoir model containing 60,000 grid blocks are presented.

2 Modeling procedure

2.1 Reservoir simulation model

Oil reservoir simulation involves the solution of equations governing the flow of reservoir fluids (oil, gas and water) through porous subsurface formations. More general formulations track individual components (e.g., methane, ethane, etc.) and/or include thermal effects. For simplicity, here is considered oil-water flows in the absence of gravity. The simulator into which the ROM is implemented is quite general, so more complex systems could be considered with relatively slight modifications.

The governing equation for flow of oil, gas and water in a porous media is determined incorporating Darcy's law and mass balance [1]. The fully implicit solution of the reservoir simulation model is given by:

$$\mathbf{T}^{n+1}\mathbf{x}^{n+1} - \mathbf{D}(\mathbf{x}^{n+1} - \mathbf{x}^n) - \mathbf{Q} = \mathbf{R} \quad (1)$$

where \mathbf{T} is a block pentadiagonal matrix for two-dimensional grids and a block heptadiagonal matrix for three-dimensional grids, \mathbf{D} is a block diagonal matrix, \mathbf{Q} represents the source/sink terms and \mathbf{R} is the residual vector. The time level is designated by the superscript n or $n + 1$. The $\mathbf{T}^{n+1}\mathbf{x}^{n+1}$ term represents convective effects while the $\mathbf{D}(\mathbf{x}^{n+1} - \mathbf{x}^n)$ term represents accumulation. The matrices \mathbf{T} , \mathbf{D} and \mathbf{Q} depend on \mathbf{x} and must be updated at each iteration of every time step.

Equation (1) is nonlinear and is solved by applying Newton's method, $\mathbf{J}\delta = -\mathbf{R}$, to drive the residuals to zero where \mathbf{J} is the Jacobian matrix given by $J_{ij} = \partial R_i / \partial x_j$ and $\delta_i = x_i^{n+1,k+1} - x_i^{n+1,k}$ with k and $k + 1$ indicating iteration level.

2.2 Proper orthogonal decomposition

Proper orthogonal decomposition (POD) was developed to identify coherent structures in dynamical systems. The basic idea of the POD technique is to project a high dimensional model onto a low dimensional model. To generate a POD reduced-order basis a time simulation of the high dimensional flow model is performed and the states of the system are saved by recording \mathcal{S} snapshots for the primary variables oil pressure and water saturation for all n_c grid blocks.

The reduced order basis generated for the pressure state will be different from the one generated for the saturation state because each represent unique physical processes. Therefore, the ROM must be applied once for the pressure state and once for the saturation state. After the snapshots are obtained, the mean of the snapshots is computed as $\bar{\mathbf{x}} = \frac{1}{S} \sum_{i=1}^S \mathbf{x}_i$ and the data matrix as $\mathbf{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_S - \bar{\mathbf{x}}]_{n_c \times S}$.

Then, a correlation matrix \mathbf{C} is determined applying the *method of snapshots* [4]. With this method, instead of computing a matrix of dimensions $n_c \times n_c$, a much smaller matrix of dimensions $S \times S$ is computed as $\mathbf{C} = \mathbf{X}^T \mathbf{X}$ and a small eigen-decomposition problem can be solved as $\mathbf{C} \Psi = \lambda \Psi$ where Ψ represents the eigenvectors and λ the eigenvalues of \mathbf{C} , respectively.

Next, the POD basis vectors can be written as a linear combination of the snapshots $\varphi_j = \sum_{i=1}^S \Psi_{i,j} \mathbf{X}_i$. After S basis vectors are computed the basis matrix of dimension $n_c \times S$ is given by $\Phi = [\varphi_1, \dots, \varphi_S]_{n_c \times S}$.

Moreover, the eigenvalues are related to the *energy* of the system and, after arranging the eigenvalues in decreasing order, this energy can be used to classify the eigenvectors; consequently, some of them can be removed from the basis matrix because they have very low energy and do not have significant information about the model under study. λ_i represents the energy of each basis vector, $E_t = \sum_{i=1}^S \lambda_i$ and α is the fraction of the total energy E_t that will be retained. Also, α can be different for the oil pressure and water saturation states. The number of basis functions retained, designated l , is simply the first l eigenvalues necessary to provide $E \geq \alpha E_t$.

Now the reduced state vector for the oil pressure or the water saturation is given by the transformation equation $\mathbf{z} \simeq \Phi_l^T (\mathbf{x} - \bar{\mathbf{x}})$.

As indicated above, for an oil-water system two transformation matrices will be created, Φ_{l_p} and Φ_{l_s} for the oil pressure and water saturation states, respectively, where l_p represents the number of retained basis vectors for the pressure state and l_s for the saturation state.

A key aspect of the reduced-order modeling process is the generation of snapshots. The information embodied in the snapshots is used to compute the POD basis and therefore impacts strongly the ability of the reduced-order basis to represent the states of the system. The predictive capabilities of the model are important because our target application is production optimization. This means that we require the ROM to be predictive for a variety of different flow scenarios, and these scenarios are not known in detail when the snapshots are generated.

2.3 Reduced-order reservoir model

The intent is now to introduce the reduced state \mathbf{z} into the discretized flow equations. This will allow to solve for only $l_p + l_s$ unknowns rather than $2n_c$ unknowns. By inserting the transformation equation into the discrete model equation 1 and premultiplying by Φ_l^T , the reduced form of the residual equation for the fully implicit solution is given by:

$$\Phi_l^T \mathbf{T}^{n+1} (\Phi_l \mathbf{z}^{n+1} + \bar{\mathbf{x}}) - \Phi_l^T \mathbf{D} \Phi_l (\mathbf{z}^{n+1} - \mathbf{z}^n) - \Phi_l^T \mathbf{Q} = \Phi_l^T \mathbf{R} = \mathbf{R}_r \quad (2)$$

Similarly, the reduced Jacobian can be computed as $\mathbf{J}_r = \Phi_l^T \mathbf{J} \Phi_l$. Finally, Newton's method can be applied to the reduced system of equations as $\mathbf{J}_r \delta_r = -\mathbf{R}_r$ where $(\delta_r)_i = z_i^{n+1, k+1} - z_i^{n+1, k}$.

2.4 Clustering snapshots

Although the POD basis is optimal in the sense that it captures the most information in an average sense, the number of snapshots needed to capture all of the necessary information to construct a reliable reduced-order basis can be substantial for complex reservoir simulation models. As a consequence, the eigen-decomposition problem can become very time-consuming. In addition, although large snapshot sets may contain the information needed to construct the reduced-order basis, they also contain redundant information that could be removed without affecting the quality of the ROM.

One option to address this problem is to apply a technique that clusters a large set of snapshots $\{\mathbf{x}^1, \dots, \mathbf{x}^S\}$ into a smaller number of clusters. In our approach, the target number of clusters \mathcal{C} is specified and the cluster centroids $\xi_1, \dots, \xi_{\mathcal{C}}$ are initialized. This can be accomplished by choosing randomly \mathcal{C} snapshots. The algorithm then computes the distance between every \mathbf{x}^i and all ξ_j as $d^i = \arg \min_j \|\mathbf{x}^i - \xi_j\|$ and assigns each snapshot \mathbf{x}^i to the closest cluster centroid ξ_j . The cluster centroids are then shifted so that

each centroid is the mean of all of the snapshots assigned to it given by $\xi_j = \frac{1}{\mathcal{S}_j} \sum_{k \in \mathcal{S}_j} \mathbf{x}_k$, where \mathcal{S}_j is the number of snapshots in cluster j .

This algorithm is not guaranteed to converge to the global minimum. The clustering procedure is therefore repeated several times using a different set of initial cluster centroids. The solution with the lowest $\sum_{i=1}^{\mathcal{S}} \|\mathbf{x}^i - \xi_d^i\|$ is then used. Applying this procedure, \mathcal{S} snapshots are reduced to \mathcal{C} centroids. Then the POD procedure described earlier is applied to the \mathcal{C} centroids, resulting in a smaller eigen-decomposition problem.

2.5 Implementation in General Purpose Research Simulator

Stanford’s general purpose research simulator (GPRS), originally formulated by Cao [2], has evolved into a modeling package containing many advanced capabilities and features. All of the simulations presented in the following sections were performed using a new version of GPRS which is able to handle the POD procedure described above.

The ROM described previously can be separated into off-line (pre-processing) and in-line portions. The off-line portion, executed just once, contains all of the computations needed to construct the ROM. This entails first running a training simulation and recording snapshot sets for pressure and saturation. Then, following the procedure given in section 2.4, the snapshots can be clustered and the basis functions generated (section 2.2).

The in-line portion of the ROM can be applied for a variety of different simulation scenarios. The basis functions, as determined from the off-line procedure, are inputs. Then, within the Newton loop, the standard Jacobian matrix and residual vector are generated, after which the reduced Jacobian and residual are formed. Then the reduced vector of unknowns (\mathbf{z}) is determined by applying Newton’s method, after which the full state vector (\mathbf{x}) is reconstructed. The model initialization and time stepping are exactly as in the standard simulator. The reduced-order models generate small but full matrices. We therefore apply direct solution techniques for these linear systems.

In the following section, the performance and timing of the ROM will be compared to full GPRS simulations, so some discussion of the linear solvers applied in GPRS is appropriate. The linear system of equations arising in the full simulation model is very sparse. For this solution, GPRS employs the iterative generalized minimum residual (GMRES) solver along with various preconditioners. The constrained pressure residual (CPR) preconditioner is the most advanced option. This preconditioner is specially designed for the reservoir simulation equations (it takes full advantage of the elliptic character of the underlying pressure equation) and is therefore highly efficient. Most comparisons will be against full simulation models that use CPR, so the comparison will be against what is believed to be (essentially) the best current procedure. More generic preconditioners, such as ILU(0), are also available within GPRS and will be considered.

2.6 Reservoir simulation results using reduced-order modeling (ROM)

The application of ROM to a realistic reservoir simulation model is illustrated and the ability of the ROM to provide accurate predictions for cases that differ from the initial training simulation will be demonstrated.

The simulation model, shown in Figure 1, is a portion of a very large geological model developed by Castro [3]. This model was developed based on a fluvial channel system. The model is three-dimensional and contains a total of 60,000 grid blocks (with $n_x=75$, $n_y=100$ and $n_z=8$, where n_x , n_y and n_z designate the number of grid blocks in the corresponding coordinate direction). Five production wells and four water injection wells drive the flow.

To extract the information needed to reproduce the behavior of the system, a full run (referred to as the training simulation) was performed. As indicated earlier, the conditions applied for this training simulation impact the quality of the reduced basis, so they should be selected with care. Here we apply a heuristic procedure in which we vary the bottom hole pressures (BHPs) of the production wells randomly and independently over their expected ranges (between 4,150 and 4,850 psia). These BHPs are changed every 100 days and the resulting schedule is shown in Figure 2. The injector well BHPs are held constant at 6,500 psia (this BHP is maintained in all runs). A total of 173 snapshots for the oil pressure and water saturation states were recorded.

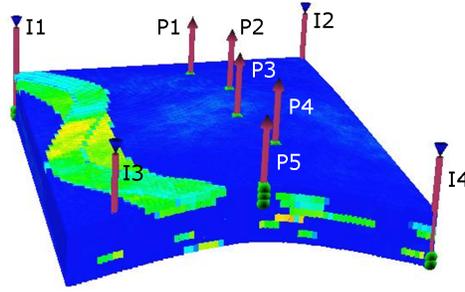


Figure 1: Reservoir

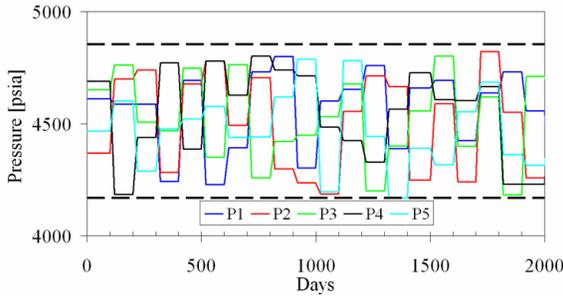


Figure 2: BHP for the producer wells

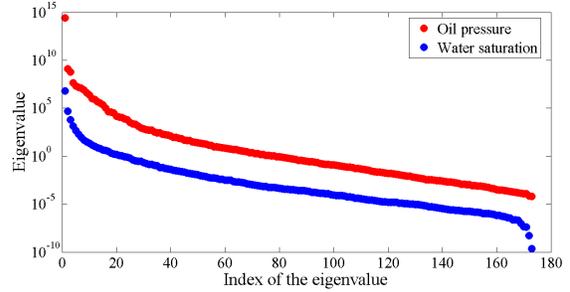


Figure 3: Eigenvalue variation for pressure and saturation matrices

Applying the POD approach provides the eigenvalue spectrums for pressure and saturation shown in Figure 3. It is evident that the maximum eigenvalue for the pressure state is almost 10^{15} while the minimum is around 10^{-5} , which means that the pressure eigenvalues vary over around 20 orders of magnitude. For the water saturation state this variation is also substantial, about 16 orders of magnitude.

Then a basis is formed such the fraction of energy in the pressure state ignored is very low, around 10^{-10} , and the fraction ignored in the saturation state is around 10^{-7} . This requires that the first 23 eigenvalues for the oil pressure state and the first 35 eigenvalues for the water saturation state are retained. Therefore, the transformation matrix $\Phi_l \in \mathbb{R}^{n_c \times l}$, where $l = l_p + l_s$ has $23 + 35 = 58$ basis vectors, meaning that while the standard reservoir simulation model needs to solve $\mathbf{J}\delta = -\mathbf{R}$ for $2n_c = 120,000$ unknowns, the reduced-order reservoir simulation model needs to solve $\mathbf{J}_r\delta_r = -\mathbf{R}_r$ for only 58 unknowns.

Implementing the clustering technique presented in section 2.4, the number of snapshots for each state was reduced from 173 to 50. Ignoring the same amounts of energy as before, the number of basis functions required for the oil pressure state decreases from 23 to 14, while for the water saturation state it decreases from 35 to 25. Thus the reduced-order reservoir simulation model now requires only 39 unknowns.

The ability of the ROM to reproduce the training simulation will be tested using both ROMs (POD and clusters + POD). Next four different flow scenarios are used to evaluate the predictive capability of both ROMs. The results for the training simulation and schedules I, II, III and IV are presented in appendix A.

An important point is the quantification of error using ROMs. Errors arise from the ROM simulations for two main reasons. These are (1) a large number of basis vectors are discarded which, although less important than the basis vectors retained, do carry some information, and (2) the ROM is in general applied to models that differ from the training simulation used to generate the reduced basis.

This error quantification could be accomplished in a variety of manners. Here a very simple procedure focus on error in oil rate, which is the most important output for this application. For each simulated time step i , the oil production rate in the reference simulation ($Q_{o,full}^i$) and in each ROM simulation ($Q_{o,ROM}^i$) is taken for each well w . The error is then computed as the average of the absolute differences, normalized

by the time-average flow rate $\overline{Q}_{o,full}$ of each well:

$$\text{Error}_w = \frac{1}{n_t \overline{Q}_{o,full_w}} \sum_{i=1}^{n_t} |Q_{o,full_w}^i - Q_{o,ROM_w}^i| \quad (3)$$

where n_t is the total number of time steps. The average error computed using this equation for the training simulation and the four schedules are presented in Table 1.

It is evident that the training simulation case results in the smallest errors, which is not surprising because the basis vectors were generated using snapshots from this case. In addition, the ROM generated with only POD has the smallest error for all examples, presumably because a greater number of basis functions are used in these cases. While the other ROM perform well, it can be observed that, although clustering snapshots decreases the number of basis functions, it increases the error by a small amount. For all simulations the errors are small, less than 3% (when quantified in this way). We note that there is a tradeoff between speedup and accuracy, as further speedup could be achieved by decreasing the dimension of Φ , though this will result in increased error.

Table 1: Errors for the various ROMs

	Training data set	Schedule I	Schedule II	Schedule III	Schedule IV
POD	0.0031	0.0247	0.0064	0.0111	0.0085
Clusters + POD	0.0078	0.0280	0.0084	0.0153	0.0114

Computational speedups for the ROMs using POD and clusters + POD are evaluated relative to full-order GPRS simulations using a specialized solver and preconditioner combination (GMRES solver and CPR preconditioner). As shows on Table 2 the maximum speedup obtained was 2.85 for schedule I. However, the comparison to a full-order GPRS simulation using a much simpler preconditioner (ILU(0)) showed a speedup of about 130. This highlights the fact that the ROM procedures considered here are very well suited for situations where the linear solver occupies the great majority of the computational time. Otherwise, the maximum speedup attainable is more limited.

Table 2: Total simulation time for the various ROMs

	Training data set	Schedule I	Schedule II	Schedule III	Schedule IV
GPRS (CPR)	1561s/100%	638s/100%	480s/100%	659s/100%	400s/100%
POD	1017s/65%	332s/52%	336s/52%	367s/56%	286s/71%
Clusters + POD	774s/49%	220s/35%	215s/45%	295s/45%	228s/56%

2.6.1 Acknowledgements

I would like to acknowledge the valuable comments provided by Prof. Louis Durlofsky and the implementation of the ROMs in the GPRS by Pallav Sarma.

References

- [1] K. Aziz and A. Settari. *Fundamentals of reservoir simulation*. Elsevier Applied Science Publishers, 1986.
- [2] H. Cao. *Development of Techniques for General Purpose Simulators*. PhD thesis, Stanford University, 2002.
- [3] S. A. Castro. *A Probabilistic Approach to Jointly Integrate 3D/4D Seismic Production Data and Geological Information for Building Reservoir Models*. PhD thesis, Stanford University, 2007.
- [4] L. Sirovich. Turbulence and the dynamics of coherent structures part I - III. *Quarterly of Applied Mathematics*, 45(3):561–590, Oct. 1987.

A Additional results

Beside the bottom hole pressures for the producer wells on the training simulation given by Figure 2 the ROM will be applied to four different schedules as shown of Figure 4

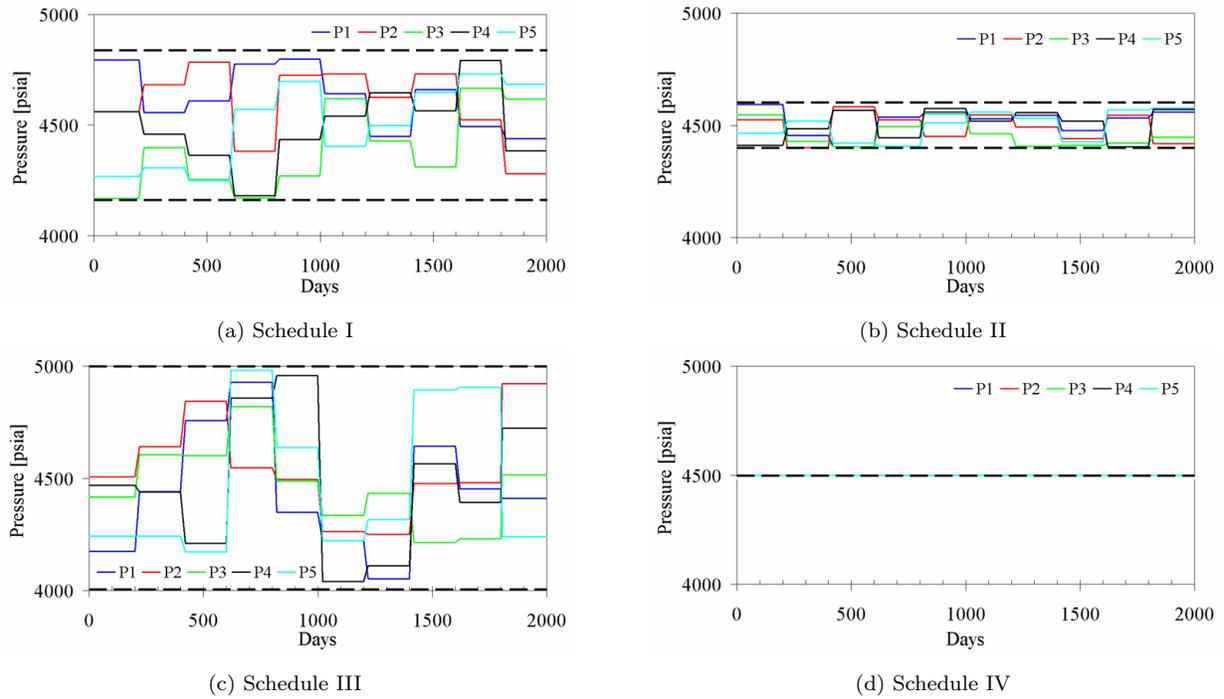


Figure 4: Bottom hole pressure for the producer wells

A.1 Training data set

Figure 5 compares the oil flow rate for all producer wells using both procedures. The reference flow rates from the training simulation (blue circles) are very well reproduced using POD only (red line) and clustered snapshots + POD (green line). These results are very encouraging and indicate that the snapshot set contains sufficient information to reproduce the training simulation. As a consequence, the POD method selects the most important directions and ignores the least important. Also can be observed that the clustering approach reduces the number of snapshots without losing significant information.

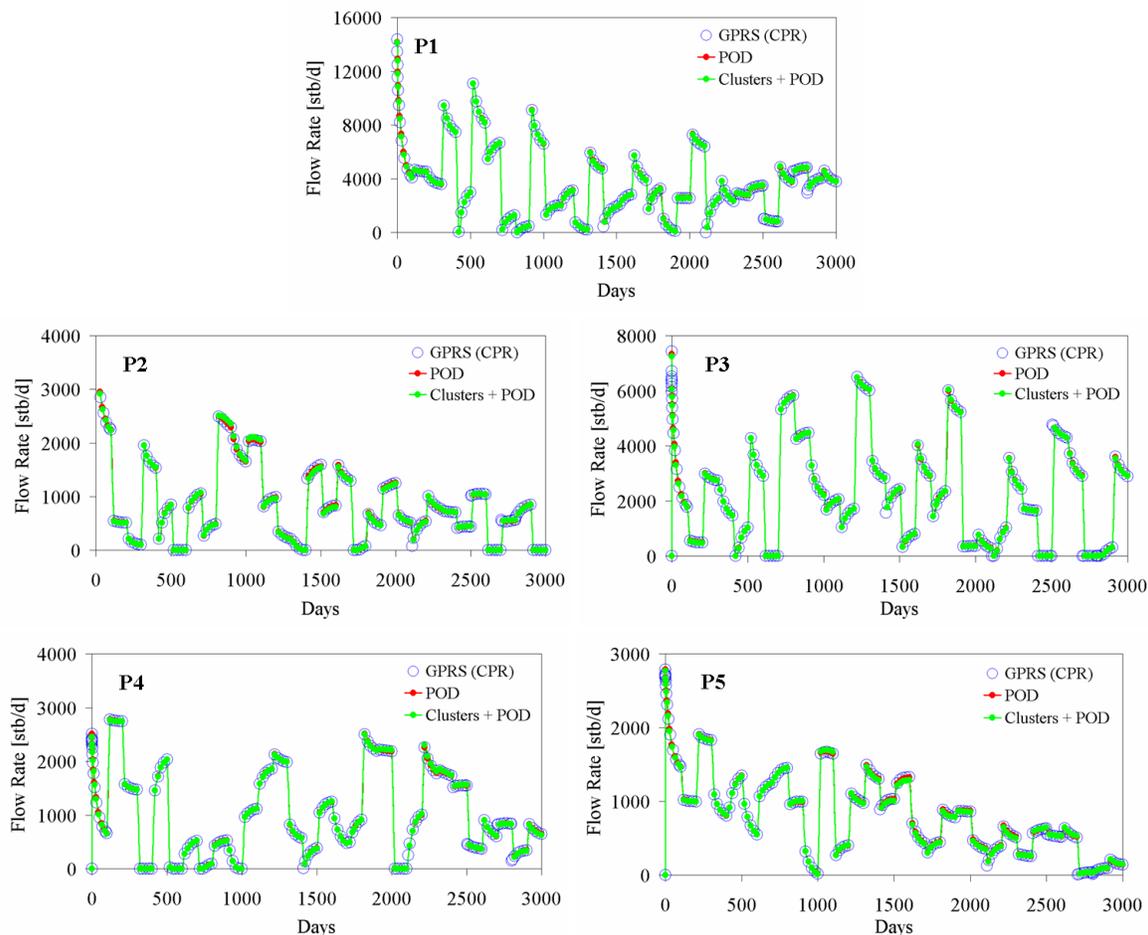


Figure 5: Oil flow rate for training simulation using full-order GPRS and two ROMs

A.2 Schedule I

The schedule for the bottom hole pressures (BHPs) of the production wells using schedule I is shown in Figure 4(a). The overall range is the same as was used in the training simulation, though the detailed schedule is quite different. In this example the BHP is changed every 200 days, while in the training simulation it was changed every 100 days. As in the training simulation, the injection wells are kept at a constant pressure of 6,500 psia.

Figure 6 shows the oil flow rate for all producer wells using both basis matrices. The flow rates from the standard (reference) solution (blue circles) are, in general, well reproduced using both ROMs. Relatively small mismatches can be observed, however, in some of the results (e.g., producer 1 between 1,000 and 1,800 days). Taken in total, these results are very encouraging as they demonstrate the ability of the ROM to provide flow results for scenarios that differ substantially from the training simulations. This is important if the ROM is to be used within the context of optimization.

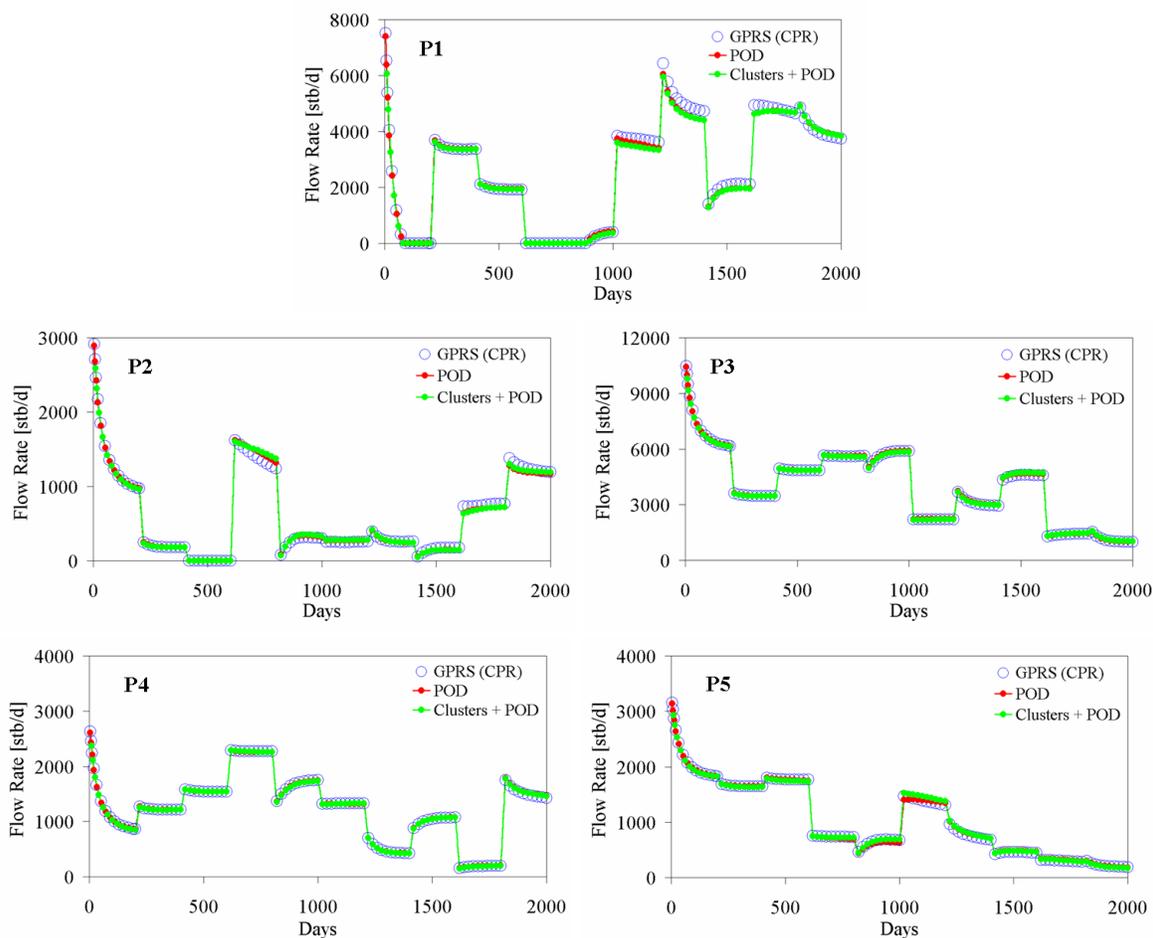


Figure 6: Oil flow rate for schedule I using full-order GPRS and two ROMs

A.3 Schedule II

Figure 4(b) shows the bottom hole pressures for the production wells for schedule II. For this example the overall BHP range (between 4,400 and 4,600 psia) and the frequency for changing the BHP (every 200 days) are different than in the training simulation. Figure 7 shows the oil flow rate for all producer wells using both basis matrices. As was the case for schedule I, the reference flow rates are again closely matched by the two ROMs. The BHP variation in this case is over a smaller range than in the training simulation, so this example demonstrates the ability of the ROMs to predict the reservoir dynamics for ‘interpolating’ conditions.

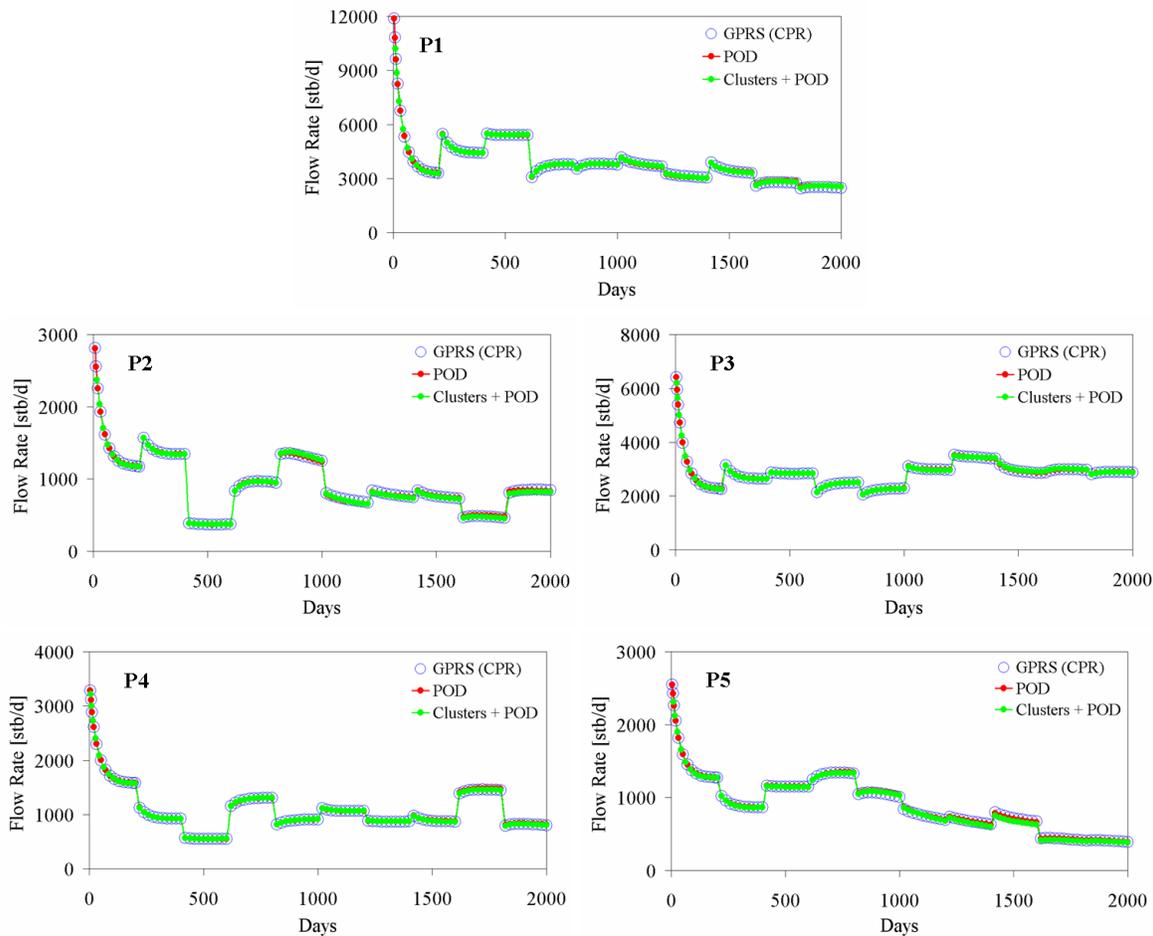


Figure 7: Oil flow rate for schedule II using full-order GPRS and two ROMs

A.4 Schedule III

The producer BHPs for schedule III are shown in Figure 4(c). In this case the overall BHP range is between 4,000 psia and 5,000 psia (a slightly greater range than in the training simulation) and the frequency for changing the BHP is every 200 days. Figure 8 shows the oil flow rate for all production wells for both simulation runs. Consistent with the previous results, the flow rates are closely matched by the ROMs. We can, however, observe some slight mismatches (e.g., well P2 at around 800 days). Overall, this example demonstrates the ability of the ROMs to ‘extrapolate’ (at least to some degree) outside of the range of the training simulation. This is a useful capability, as it suggests a degree of robustness with respect to the settings used in the training simulation.

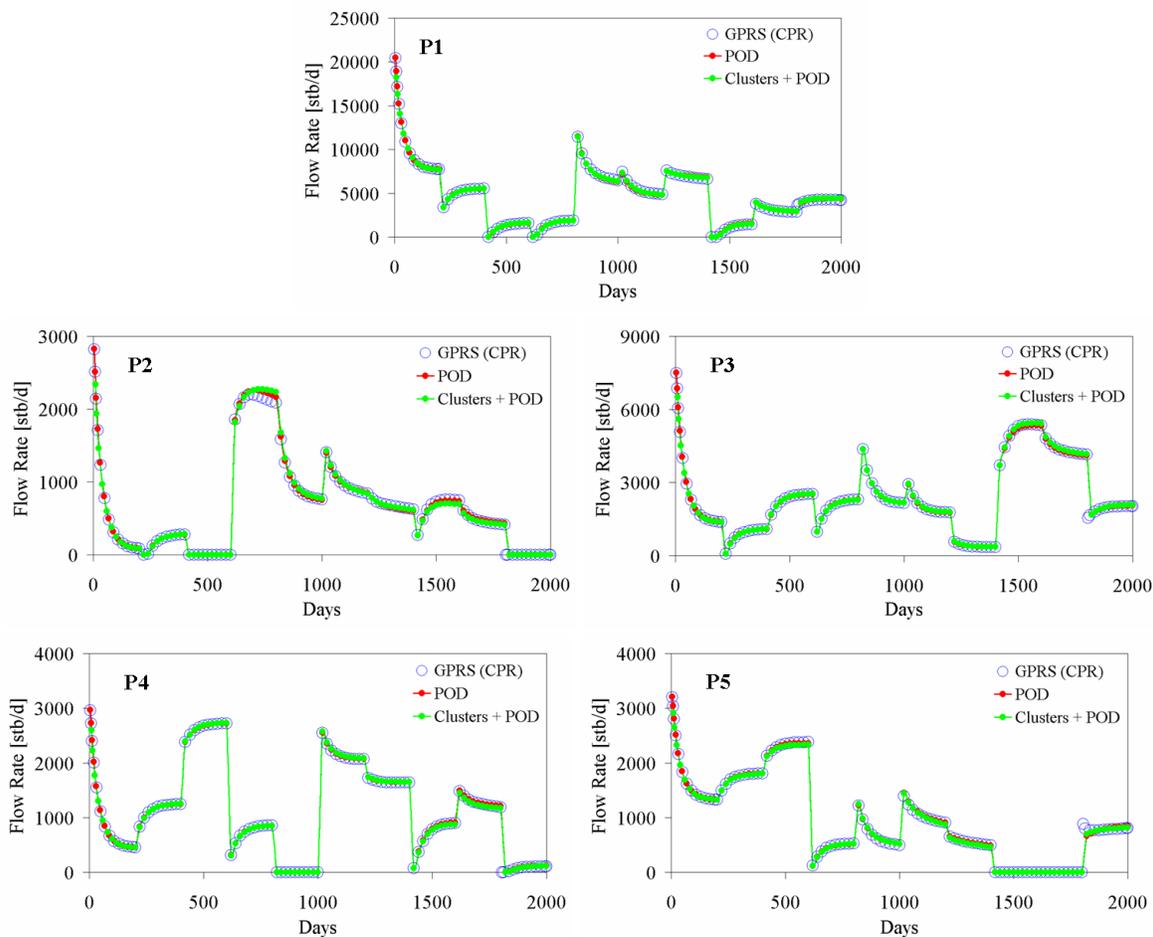


Figure 8: Oil flow rate for schedule III using full-order GPRS and two ROMs

A.5 Schedule IV

The goal here is to demonstrate that the ROMs can also reproduce much simpler reservoir dynamics. Thus now the BHPs are specified for all production wells to be constant (4,500 psia) over the entire simulation, Figure 4(d). Flow results for the five production wells are shown in Figure 9. The performance of both ROMs is quite good. This again demonstrates the ability of the ROMs to simulate the dynamics of the reservoir model for conditions very different than those used in the training simulation.

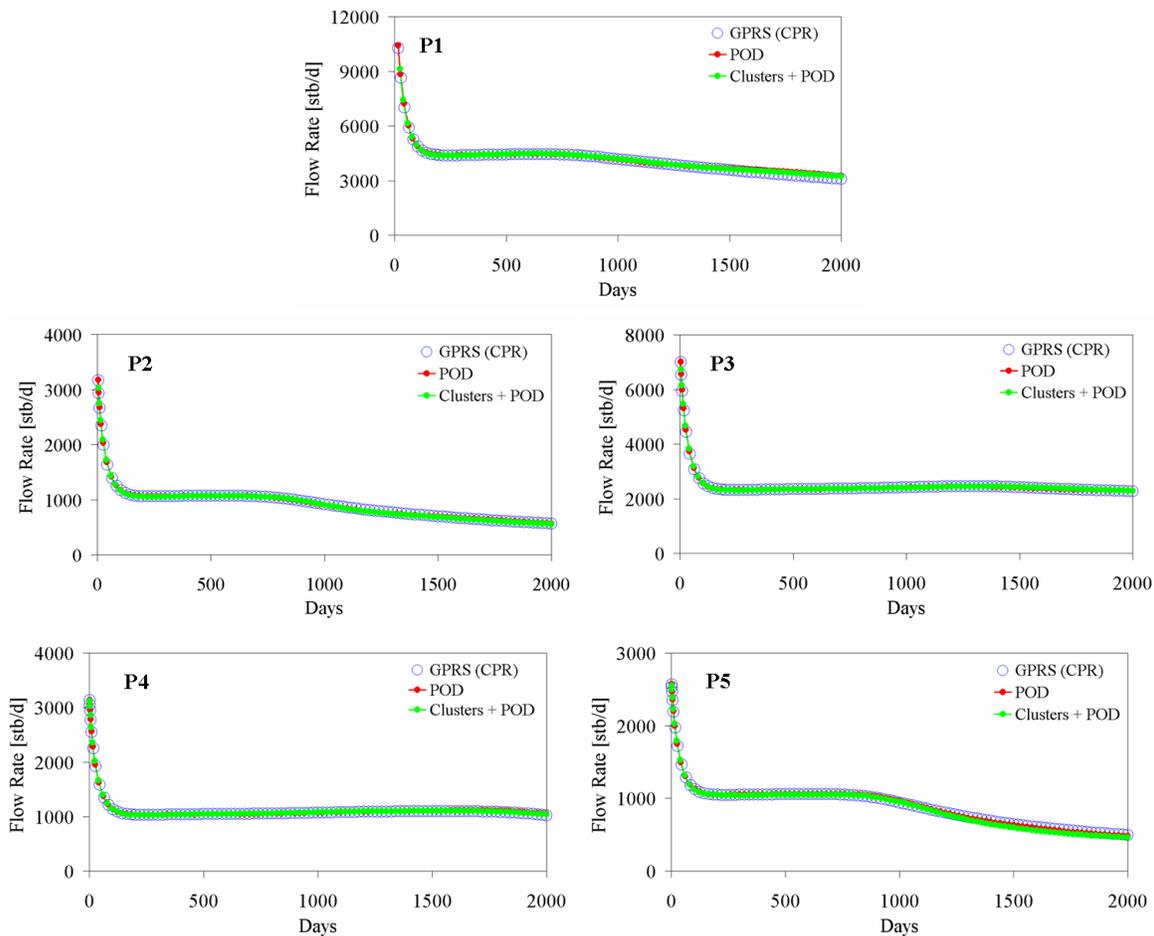


Figure 9: Oil flow rate for schedule IV using full-order GPRS and two ROMs