SPE 141922

Pressure Preconditioning Using Proper Orthogonal Decomposition

P. Astrid, SPE, G.V. Papaioannou, SPE, J.C. Vink, SPE, Shell, and J.D. Jansen, SPE, Delft University of Technology

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA, 21–23 February 2011.

Abstract

We developed and implemented a new first-stage preconditioning method for large-scale reservoir simulation as an alternative to the popular Algebraic Multi Grid (AMG) method. We used Proper Orthogonal Decomposition (POD) to derive a reduced-order model for the linearized pressure equation in a proprietary reservoir simulator. A small set of pre-computed pressure solutions are used to transform the equation into a lower-order system that can be solved economically yet provides a relatively accurate estimation of the solution of the original high-order system. We present results for a two-phase (oil-water) reservoir model under water flooding conditions. Use of the POD-based preconditioner lead to a significantly faster convergence of the overall simulation compared to AMG, and reduced the linear solver times between approximately 70 % and 80%, i.e. we obtained accelerations with factors between three and five. The highest speed up was achieved for a case in which the flow rates in the eight injection wells were changed frequently during the entire production period, using ten POD basis functions obtained from short training simulations in which the injectors were operated only once each. These first results are very encouraging, especially because there is room to optimize the trade-off between the number of basis functions used in the POD method and the required number of linear iterations. However, further research is required to assess the applicability of the POD-based preconditioner to more complex cases including e.g. strongly compressible flow or compositional effects. The overhead required to pre-compute the POD solutions implies that the new method will be particularly attractive when many solutions of near-similar simulation models are required such as in computer-assisted history matching or flooding optimization.

Introduction

The primary driver of this work is to improve simulation speed, in particular for situations where a large number of simulations are required as e.g. during computer-assisted history matching or flooding optimization using iterative procedures. We employed reduced-order modeling (ROM) concepts to develop a new preconditioner that reduces the time spent on the linear solver by speeding-up convergence and reducing operation count while preserving the accuracy and reliability of the results. In particular, we used Proper Orthogonal Decomposition (POD) to speed up the pressure solution during a two-stage preconditioning procedure in a fully implicit time integration scheme. The method is also applicable to Implicit Pressure Explicit Saturation (IMPES) schemes, either with finite difference or with streamline simulation of the saturations, Implicit Pressure Explicit Accumulation (IMPEC) schemes, as well as Adaptive Implicit Methods (AIM). We implemented the POD-based preconditioning method in a proprietary dynamic reservoir simulator which employs several of the aforementioned schemes, as an alternative to the Algebraic-MultiGrid (AMG) method which is typically used to solve the pressure equations in these schemes. A peculiarity of the method compared to the AMG method is the requirement to compute a number of basis functions based on ‘snapshots’, i.e. converged time stepping solutions, of the pressure field during a number of short simulations as a precursor to the full simulation(s). Obtaining a set of basis vectors that would be ‘sufficiently accurate’ for the full simulation(s) is key to the success of the method. Several possible workflows have been identified where that condition is likely to be met, in particular for water flooding cases with multiple injectors and producers:

- Running a short pre-simulation during which each of the wells is subsequently activated with the aim to capture the main flow characteristics of the reservoir. In the absence of compressibility and capillary effects this is expected to accurately capture the reservoir dynamics based on linear superposition arguments.
- Running many simulations with slightly modified parameters or well inputs such as required for assisted history matching or flooding optimization studies using iterative procedures or for Monte-Carlo simulations to quantify the effects of reservoir parameter uncertainty. In this case a base run is usually preceding a large number of simulations. Basis functions obtained from pressure snapshots in the base run could then be used to speed-up the subsequent simulations.

In this paper we will demonstrate results for the first workflow. We will consider 1) the total simulation time, 2) the linear solver time, and 3) and the number of linear solver iterations as metrics to assess the suitability of the method in comparison to the AMG method.

Model reduction as an alternative solution method for the first stage preconditioner

ROM is a rapidly developing activity in various application domains of large-scale numerical simulation, such as computational fluid dynamics, structural dynamics, and the simulation of integrated circuits; see Antoulas (2005) and Schilders et al. (2008) for recent overviews. Most ROM methods are based on a projection-framework. In this framework, the large-scale system is transformed into a low-order (small-scale) system by projecting the high-order model onto the space spanned by a set of orthonormal basis functions. In a projection-based model reduction method, a high dimensional variable, say, $\mathbf{x} \in \mathbb{R}^n$ is approximated by a linear combination of ℓ orthonormal basis functions where $\ell << n$:

$$\mathbf{x} \approx \sum_{i=1}^{\ell} z_i \boldsymbol{\varphi}_i , \tag{1}$$

where $\boldsymbol{\varphi}_i \in \mathbb{R}^n$ is a basis function and z_i is its corresponding coefficient. Equation (1) can be expressed in matrix notation as:

$$\mathbf{x} \approx \boldsymbol{\Phi}\mathbf{z} , \tag{2}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2 \quad \cdots \quad \boldsymbol{\varphi}_\ell]$, $\boldsymbol{\Phi} \in \mathbb{R}^{n \times \ell}$ is the basis or matrix of the basis functions, and $\mathbf{z} = [z_1 \quad z_2 \quad \cdots \quad z_\ell]^T$, $\mathbf{z} \in \mathbb{R}^\ell$ is the vector of basis function coefficients. In our application, ROM is applied to speed-up the first stage of a Constraint Pressure Residual (CPR)-type preconditioning (Wallis et al., 1985) in the linear solver, which requires solving of a (Laplacian-type) equation

$$\mathbf{A}\mathbf{x} = \mathbf{b} , \tag{3}$$

where $\mathbf{x}$ now represents a vector of grid block pressures. The reduced-order model is obtained by projecting the linearized pressure equation (3) onto the subspace spanned by the basis $\boldsymbol{\Phi}$:

$$\underbrace{\boldsymbol{\Phi}^T\mathbf{A}\boldsymbol{\Phi}}_{\mathbf{A}_r}\mathbf{z} = \underbrace{\boldsymbol{\Phi}^T\mathbf{b}}_{\mathbf{b}_r} . \tag{4}$$

We can thus write the reduced-order model as:

$$\mathbf{A}_r\mathbf{z} = \mathbf{b}_r . \tag{5}$$

Using the projection matrix $\boldsymbol{\Phi}$, the original large scale, sparse matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is transformed into a much smaller, dense matrix $\mathbf{A}_r \in \mathbb{R}^{\ell \times \ell}$. Even though the reduced-order matrix is dense, it is sufficiently small that it can be solved efficiently using direct inversion methods. In practice an LU decomposition is calculated once every time $\mathbf{A}$ is updated; for a fully implicit time integration this is once per Newton-Raphson iteration. Then a back-substitution is done once every time the preconditioner is applied in the linear solver iterations. The solution $\mathbf{z}$ of the reduced-order model, equation (5), is then used to compute an approximate solution $\tilde{\mathbf{x}} = \boldsymbol{\Phi}\mathbf{z} \approx \mathbf{x}$ to equation (3) with the aid of equation (2). The ROM-based projection method is expected to be computationally advantageous over the AMG method for solving equation (3) if it can yield an approximation $\tilde{\mathbf{x}}$ to $\mathbf{x}$ of a similar accuracy, with less computational effort than the AMG method. The two approaches are different in that the reduced system in the ROM approach is solved exactly and the fine scale pressure solution is approximated using the prolongation method, i.e. equation (2), while the AMG method is an iterative method terminating at a requested tolerance. Another reason to expect that ROM approach may be more efficient is that the coarsening procedure in the AMG method is recalculated for every linear system within a Newton-Raphson iteration, while in the ROM method the basis vectors are pre-calculated off-line but are nonetheless expected to capture the dominant characteristics of the individual

linear systems throughout the simulation. However, there is no guarantee that a pre-computed ROM basis will remain valid throughout the simulation. Generally, when ROMs are used to reduce high dimensional variables resulting from the discretization of the spatial domain, the bases are less sensitive to temporal variations than spatial variations. This is particularly true for our application of ROM to a predominantly elliptical equation. For example, a ROM basis generated from training simulations that employed a specific well configuration will not be valid to simulate cases when new wells are added to the configuration. A ROM basis generated from training simulations for given (transient) well rates may still be very well applicable to simulate cases with different well rates as long as the transient effects are within the range explored during the training simulations. The validity of a ROM subspace remains an open problem for nonlinear partial differential equations (PDEs). Analytical solutions do not exist for nonlinear PDEs; therefore it is very difficult to determine ROM approximation qualities a-priori for all possible variations of model parameters. There are several ways to improve ROM robustness:

- Start with an initial set of simulation studies, derive an initial reduced model basis based on these simulation studies and update this basis on-the-fly while carrying out simulations for other conditions. This approach will require the original model to be run in parallel with the reduced-order model.
- Use a solution-independent basis such as a Fourier basis which would make the method resemble a traditional 2-level multi-grid method. This approach will result in a more predictable, solution-independent basis, but it is to be expected that this reduced model space will need to be larger in order to capture dominant model features sufficiently accurately. If the model space is too small, this will result in slower convergence, if it needs to be very big, the gain of model reduction is too small.
- Carry out an extensive simulation study (e.g. using the Monte-Carlo method) to sample widely in the model parameter space and derive a basis that spans the solutions from this simulation study. Obviously, this approach will be computationally tedious.

In this study, basis functions are not updated during the validation simulations. Consequently, the reduced-order model may not be able to capture variations, especially spatial variations that are not encountered during the training simulations. However, we have deliberately chosen well configurations in the training simulations that cover all situations encountered during the validation simulations.

Proper Orthogonal Decomposition

Literature review

To speed up convergence, the ROM subspace must be small (spanned by a small number of basis functions) and contain the dominant or most relevant features of the solution of the fine-scale system. The latter is difficult to guarantee in general and we shall simply derive a ROM basis based on representative training simulations. In this work, we used POD, also known as Karhuenen-Loève expansion, principal component analysis, or the method of empirical orthogonal functions, to create the ROM basis for the pressure preconditioning step. POD is one of the most popular model reduction methods for nonlinear large-scale systems. POD has its origins in the statistical analysis of data, and is frequently used in e.g. computational fluid dynamics and other fields that employ large-scale numerical simulation. An early application to analyze the flow through porous media has been reported by Gharbi et al. (1997). A first attempt to speed up the simulation of single-phase (groundwater) flow was described by Vermeulen et al. (2004), and of two-phase (oil-water) flow by Heijn et al. (2004), and Van Doren et al. (2006). These papers indicated that POD is an extremely efficient method for speeding up (linear) single-phase flow simulations but less so for accelerating the simulation of (nonlinear) two-phase flow simulations. In the latter case, the need to reassemble the reduced order system (5) at every iteration reduces the computational efficiency. Moreover, POD is much more efficient to reduce the (near-elliptic parabolic) pressure equation, than the (near-hyperbolic parabolic) saturation equation. Markovinović and Jansen (2006) proposed to use POD to improve the initial guess for linear solver iterations for the pressure equation by computing a time-varying POD basis ‘on the fly’ using recent pressure solutions. Another application of POD to porous media flow was reported by Kaleta et al. (2010) who used the method to obtain tangent linear approximations of the simulation equations without the need to compute Jacobian matrices, which were subsequently used for gradient-based flooding optimization. Cardoso et al. (2009) improved the computational efficiency of POD for two-phase flow simulation using various techniques, among which ‘missing point estimation’, building on the work of Astrid et al. (2008) and achieved speed ups up to a factor of 10 compared to an optimized conventional simulation. Moreover, they analyzed the performance of POD for various reservoir drive mechanisms, and found that the method works well for flow driven by injection and production wells, but may lead to difficulties in case of gravity-driven flow. Also the presence of strongly nonlinear relative permeabilities was shown to reduce the effectiveness of POD. More recently, Cardoso et al. (2010a, 2010b) introduced the ‘trajectory piecewise linearized’ (TPWL) method in reservoir simulation, in which POD forms an essential component. The results are promising, and speed up factors between 500 and 2000 were reported.

Overview

We will briefly review the POD method and refer to the publications mentioned above for more detailed descriptions. The POD method yields the optimal basis for a given dataset. By optimal we mean that the POD approximation yields an error that is minimum in a least squares sense, and that the number of POD basis functions required to approximate the high dimensional variables is the lowest amongst all other orthonormal bases (e.g. those related to a Fourier expansion). POD basis functions can be visually understood as spanning the dominant spatial or temporal features of the original solutions. The POD basis is derived by solving an optimization problem. Given a collection of high dimensional state vectors $\mathbf{x}_i$, say from m time samples collected in a 'snapshot' matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ according to

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_m) \end{bmatrix}, \tag{6}$$

the POD basis $\mathbf{\Phi}$ is the solution of an L_2 optimization problem:

$$\min_{\mathbf{\Phi}} \left\| \mathbf{X} - \tilde{\mathbf{X}} \right\|_2, \tag{7}$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times m}$ is a matrix of reconstructed state vectors $\tilde{\mathbf{x}} = \mathbf{\Phi z}$, subject to orthonormality constraints

$$\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j = \delta_{ij}, \tag{8}$$

where δ_{ij} denotes the Kronecker delta. The solutions of problem (6) are obtained by solving the eigenvalue problem:

$$\mathbf{C\Phi} = \mathbf{\Phi \Lambda}, \tag{9}$$

where $\mathbf{C}$ is a generalized covariance matrix of the snapshot data:

$$\mathbf{C} = \mathbf{X}\mathbf{X}^T. \tag{10}$$

This result is known as the Schmidt-Eckart-Young-Mirsky theorem named after the four authors who derived proofs of different degrees of generality; see e.g. Antoulas (2005). It is also common practice to subtract the time-averaged value of snapshot data from the snapshot data, although we did not implement this subtraction in this study:

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{x}(t_1) - \bar{\mathbf{x}} & \mathbf{x}(t_2) - \bar{\mathbf{x}} & \cdots & \mathbf{x}(t_m) - \bar{\mathbf{x}} \end{bmatrix}, \tag{11}$$

where

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}(t_i). \tag{12}$$

The corresponding 'true' covariance matrix for the mean-subtracted snapshot data is then

$$\bar{\mathbf{C}} = \bar{\mathbf{X}}\bar{\mathbf{X}}^T. \tag{13}$$

The eigenvectors of equation (10) with large eigenvalues represent the most dominant dynamics of the system. Therefore one can restrict the low-order model to the modes with the largest eigenvalues or 'energies'. The order of the reduced-order model, i.e. the number of eigenvectors, can be determined by requiring that the 'energy' content E_ℓ of the reduced model

$$E_\ell = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{14}$$

is close to one. Note that the term 'energy' is used loosely here, motivated by the fact that energy can often be written as a quadratic form. The desired energy content is typically chosen to be higher than 0.90. This suggests that more than 90% of the dynamics contained in the set of training images obtained from the original model is retained in the reduced-order model. Of course this is a heuristic measure and there is no guarantee that the corresponding reduced-order model will only deviate 10% from the original model. To summarize: the POD basis functions are the eigenvectors of the generalized covariance matrix, as defined in equation (10), that correspond to ℓ largest eigenvalues. By construction matrix $\mathbf{C}$ is symmetric non-negative definite and thus diagonalizable by an orthogonal matrix containing the eigenvectors of $\mathbf{C}$. The eigenvectors of $\mathbf{C}$ are the left-singular vectors $\mathbf{U}$ of the non-square matrix $\mathbf{X}$. An efficient way to calculate them is through the eigenvalue decomposition of the much smaller matrix $\mathbf{D} = \mathbf{X}^T\mathbf{X}$:

$$\mathbf{D} = \mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{15}$$

and calculation of $\mathbf{U}$ from:

$$\mathbf{U} = \mathbf{X}\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}. \tag{16}$$

Case Study: Three-dimensional oil-water model

Simulation set up

The reservoir model that we consider in this case study is a three-dimensional, two phase (oil-water) model, developed by Van Essen et al. (2009). The spatial domain is divided into 25200 grid blocks (60×60×7 in x, y and z directions; grid sizes are 8 m in x and y-directions and 4 m in z-direction.). Out of 25200 grid blocks, 18553 grid cells are active. There are 8 injection wells (indicated by blue lines) and 4 production wells (indicated by red lines) as depicted in **Fig. 1**.

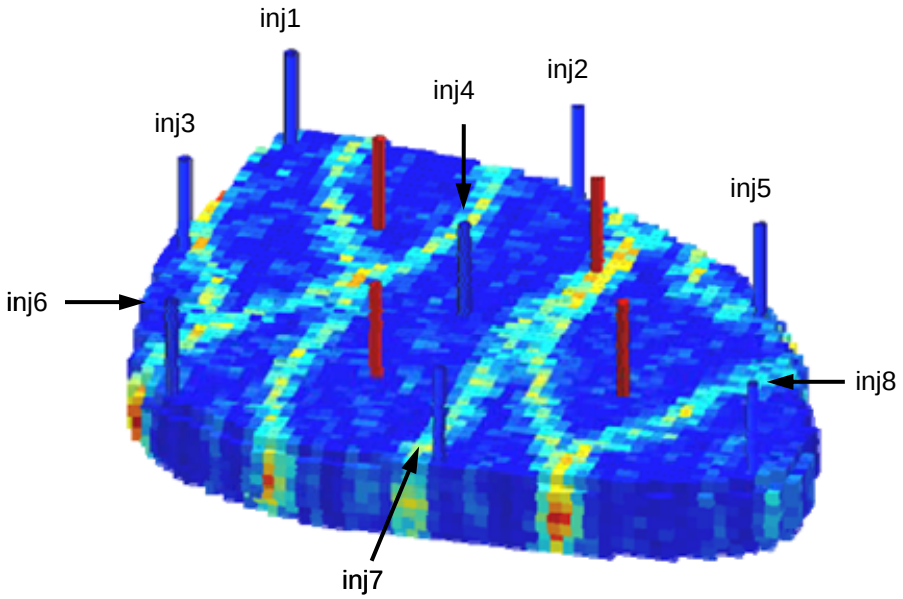

Fig. 1. Schematic overview of the reservoir model used in the case study; after Van Essen et al. (2009).

The oil and water densities are 800 and 1000 kg/m^3 respectively. The viscosities and fluid compressibilities are equal for both phases and have magnitudes 10^{-3} Pa s and 10^{-10} Pa^{-1}. The rock compressibility is negligible. The absolute permeabilities are isotropic and vary between 5 and 7000 mD, but typically the background and channel permeabilities are around a few hundred and a few thousand mD respectively. The relative permeability curves have been plotted in **Fig. 2**. Capillary forces are disregarded. The wells are perforated over the entire reservoir height and the well models are standard Peaceman-type. The positions of the injection and production wells, along with the high permeability streaks have been indicated in Fig. 1.

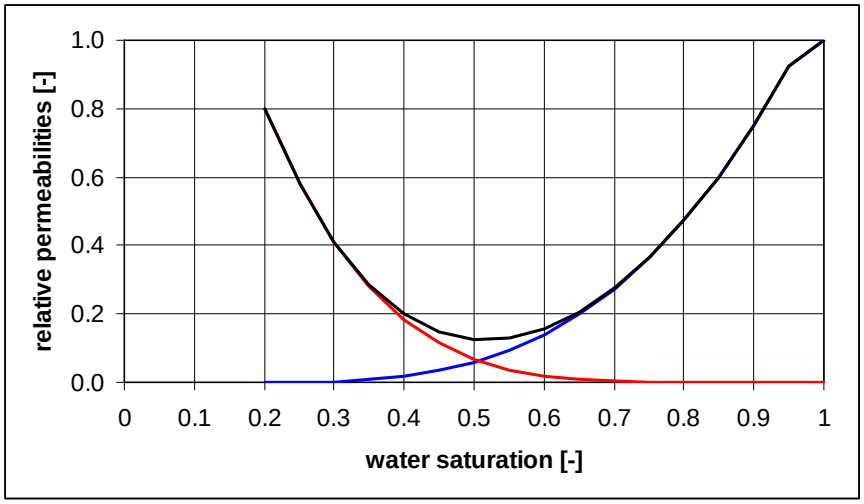

Fig. 2. Relative permeability curves; red - oil, blue - water, black - total.

Snapshot collection

Training simulation data (i.e. snapshots of the pressure field) are collected by simulating the reservoir for a short period of time while dynamically 'exciting' the reservoir by turning on and off the wells one by one. Assuming that all well control settings may change during the 'full' simulation, the POD basis functions should at least contain the solutions obtained by flowing each of the wells separately (or, more generally, by flowing them in at least as many linear combinations of well control settings as there are wells). Transient pressure effects can be captured by taking snapshots during the transient period of pressure increase or decrease after operating a well. The training simulation set up is based on the observations that for waterflooding, the fluids are almost incompressible (also in field cases) and hence the transient pressure effects can be captured in a relatively short time. This is an approximate approach because relative permeability effects may influence the pressure distribution in the reservoir as a function of the oil-water front position, but we assume that these effects are small enough to be neglected. (After all we aim only at an approximate pressure solution as a precursor to a full solution of the system equations.) In that case the pressure field generated by any combination of operating conditions of the wells is a linear combination of the pressure fields generated by each injector or production well. Note that we only need to excite those wells of which the well settings (i.e. prescribed bottom hole pressures, tubing head pressures, flow rates, or choke settings) will be changed during the subsequent simulations(s). Producers and injectors have similarly important effects on the pressures in the reservoir. It was shown by Zandvliet et al. (2008) that for slightly compressible flow the controllability of the reservoir pressures by manipulating the well controls is very limited, and that the reservoir dynamics can be captured accurately by a number of basis functions that is at most twice the number of wells. If compressibility is very small, such that the pressure equations can be approximated as being elliptic, and if relative permeabilities can be approximated as straight lines to decouple pressure from saturation, the number of required basis functions reduces to at most the number of wells. The snapshots can then simply be obtained by computing steady-state responses for a given saturation distribution, and the number of required snapshots will just be equal to the number of to-be-controlled wells. In case of strong rel-perm effects additional snapshots could, in theory, be collected for increasingly progressing water-oil front, but this would increase the 'overhead' time of the procedure. Similarly, for assisted history matching applications, snapshots could be collected for a wide range of reservoir parameter combinations, e.g. by simulating the steady state response at a fixed well control setting for an ensemble of reservoir models. Alternatively, the basis can be computed 'on the fly', using a small number of the most recently converged results; see Markovinović and Jansen (2006). In the current example, we will consider a single realization and compute the basis in advance. Moreover, the producers will remain operating under constant conditions during the full simulations, so we only excite the injector wells in the training simulations. We will consider three test cases (validation simulations) using combinations of two different training simulations and two different validation simulations. In the first training simulation only the outer left injectors (inj1, inj3 and inj6) are excited, to illustrate the impact of spatial excitations on the POD approximation quality, whereas in the second training simulation all eight injectors are excited. In the first test simulation we introduce a single transient by switching on all wells simultaneously and subsequently producing the reservoir for 1000 days with fixed well settings. In the second validation simulation, we continuously vary the injection rates in all injectors to create situation of continuous transient response during the 1000 days of production.

Training simulations with three injectors at the outer left (inj1, inj3, and inj6)

For this training simulation, three datasets are created. In each data set, an injector is turned on at a rate of 400 bbl/day, and the simulation period is set to 90 days. The converged pressure data sets are then collected in a composite snapshot matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1' & \mathbf{X}_2' & \mathbf{X}_3' \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{74} \end{bmatrix}, \tag{17}$$

where each matrix $\mathbf{X}_i', i = 1,\ldots,3$, corresponds to the simulation data set when injector *i* is turned on. The snapshot matrix in this example has a dimension of 18553 rows (corresponding to 18553 active grid cells) and 74 columns (corresponding to 74 time steps collected in the snapshot matrix **X**). The corresponding spatial generalized covariance matrix of equation (16) is $\mathbf{C} = \mathbf{X}\mathbf{X}^T$, $\mathbf{C} \in \mathbb{R}^{18553\times 18553}$. The eigenvectors of this spatial covariance matrix will be the POD basis, but computing an eigenvalue decomposition of a matrix of size 18553×18553 is computationally very intensive. Fortunately, the basis can be calculated by forming the temporal covariance matrix as described in equations (16) and (17) which only involves calculating the eigenvalues and eigenvectors of a 74×74 matrix. Note that the reduced-order model is going to be used to simulate conditions different than the ones under which the basis functions are derived. In this case it is assumed that the pressure dynamics due to the operation of several injection wells can be approximated as a (linear) combination of pressure dynamics induced by individual wells. Eigenvalues of the covariance matrix absolute pressure fields are plotted in **Fig. 3**.The first eigenvalue of the pressure covariance matrix is very large and (based on equation (14)) already amounts to 0.9999 of the total 'energy' captured in the snapshots. The first eigenvector that corresponds to the first eigenvalue is the mean of the pressure datasets. Taking the first eigenvector only will not suffice as we will only capture the mean dynamics. In practice, the cut-off point is usually determined from the point where the eigenvalue plot starts to be asymptotic. Based on Fig. 3, the eigenvalue plot does not seem to reach an asymptotic value, but the relative changes in the eigenvalues start to be smaller after the 20th eigenvalue. We took 25 eigenvectors corresponding to 25 largest eigenvalues, as the pressure field induced by each injector set tends to settle after about 8 time steps, so we expect that in total at least $3 \times 8 = 24$ eigenvalues will be needed to cover the pressure transients induced by 3 injectors. Because we took 25 eigenvectors, the order of the reduced-order model will be at most 25.

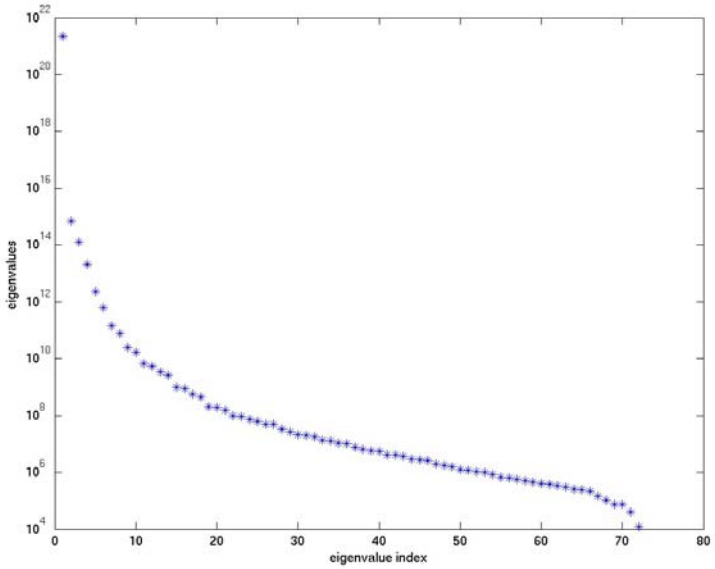

Fig. 3. Eigenvalues of the pressure snapshot matrix corresponding to subsequent activation of injectors inj1, inj3, and inj6.

Training simulations with all injectors subsequently activated

In this case snapshots are collected with injectors 1 to 8 subsequently activated, each for a period of 90 days. There is only one injector injecting at a time. **Fig. 4** shows the spectrum of the matrix composed by the pressure snapshots collected from this training simulation.

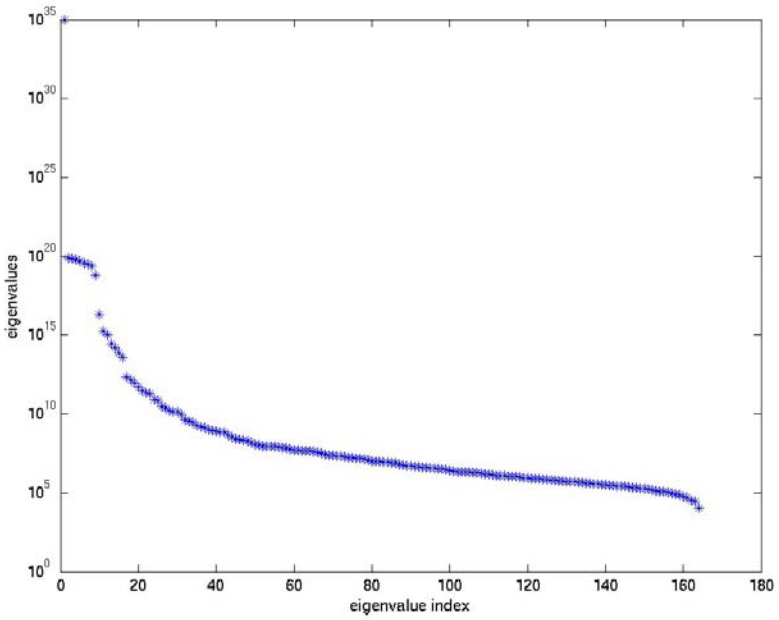

Fig. 4. Eigenvalues of the pressure snapshot matrix corresponding to subsequent activation of all injectors (inj1 to inj8).

Validation simulation with a single transient

As a first step, the ROM is tested in a simulation case where all injectors are turned on simultaneously at a rate of 600 bbl/day, using the training simulation with three injectors only. A total of 1000 days of production are modeled. To have a fair comparison, ROM results are benchmarked against simulations results using AMG with a default accuracy setting (tolerance = 0.1). A higher accuracy setting will require extremely long simulation times and is not practical for most simulation purposes.

TABLE 1 – TEST RESULTS FOR THREE CASES

	3 injectors, single transient	all injectors, single transient	all injectors, multiple transients
Total solver time reduction [%]	58	61	71
Linear solver time reduction [%]	69	72	79
Linear solver steps reduction [%]	15	42	42

The second column in **Table 1** shows that for this example, ROM-based preconditioning reduces the linear solver time by 69 %, i.e. with more than a factor three compared to AMG preconditioning. The number of linear solver steps is also reduced and the total solver time is reduced by 58 %. Any speed-up comparison requires a quality check on the attained results. Since the method is only a different preconditioner the results should be accurate up to the tolerances of the linear solver accumulated over the time interval. This was confirmed in all cases. We note that the results in Table 1 indicate that the AMG preconditioner did not perform optimally in this case. In a typical CPR/AMG application we would expect the linear solver time to be roughly half the total solver time. Moreover, we are considering a two-phase model were only one additional state variable (besides the pressure) needs to be solved during the second stage of the solver. In more complex, compositional simulations, where several additional state variables need to be solved during the second stage, the benefit of accelerated pressure solves will be relatively reduced.

TABLE 2 – TEST RESULTS WITH VARYING NUMBER OF BASIS FUNCTIONS (BF) DERIVED FROM TRAINING SIMULATION WITH 3 INJECTORS; SINGLE-TRANSIENT TEST

	AMG	3 BF	**5 BF**	10 BF	15 BF	20 BF	25 BF
Total solver time [s]	327	166	**137**	142	141	146	152
Linear solver time [s]	277	115	**86**	90	89	94	100
CPU time in solver setup [s]	0.11	0.22	**0.31**	2.83	2.73	2.98	3.05
# Time steps	256	256	**256**	256	256	256	256
# Newton steps	652	657	**658**	658	655	655	655
# Linear solver steps	5837	6687	**4982**	4804	4404	4379	4395
Max. # linear solver iterations	115	48	**43**	42	40	39	39
Fraction of spectrum	N/A	0.9998232	**0.9999516**	0.9999885	0.9999948	0.9999970	0.9999980

In **Table 2** we show results and comparisons for the presented model using different basis sizes, i.e. different numbers of basis vectors. As expected, the number of linear solver iterations decreases as the number of basis vectors increases. However, there seems to be an optimum number of basis vectors beyond which the benefit of the more accurate representation is offset by the increase in operations associated with the larger basis size. The last row of Table 2 shows the fraction of the spectrum represented by the basis according to equation (14). The shaded column highlights the case that displayed the best performance among the ones tested. Next the ROM is tested with the same simulation case where all injectors are turned on simultaneously at a rate of 600 bbl/day, but now using the training simulation with all injectors. The results for different sizes of basis have been displayed in **Table 3**. An even higher reduction in total solver time of 61 % is achieved in this case, while the linear solver time is reduced by 72 % or equivalently a speed-up by a factor of over 3.6; see also the third column of Table 1. Comparing the results of Table 2 with those of Table 3 we see that in the latter case the best performance was achieved with 20 basis vectors as opposed to just 5 basis vectors in the first case. A comparison of the corresponding spectra of Fig. 3 and Fig. 4 shows a flat region in the high eigenvalue end for the latter case indicating a more even distribution of the 'energy' among those modes. The fact that the shortest simulation time is achieved with the second more elaborate training schedule should be expected. It is important to note however that the more snapshots we include in the matrix the higher the maximum eigenvalue is, as can be observed in Fig. 3 and 4.With the maximum eigenvalue reaching so high values the quality of the basis calculation using the 'minor' product **D** could become questionable.

TABLE 3 – TEST RESULTS WITH VARYING NUMBER OF BASIS FUNCTIONS (BF) DERIVED FROM TRAINING SIMULATION WITH ALL INJECTORS; SINGLE-TRANSIENT TEST

	AMG	5 BF	10 BF	15 BF	**20 BF**	25 BF	30 BF
Total solver time [s]	327	184	226	154	**128**	141	135
Linear solver time [s]	277	123	146	93	**77**	81	83
CPU time in solver setup [s]	0.11	0.39	0.39	1.87	**3.19**	3.40	3.52
# Time steps	256	256	256	256	**256**	256	256
# Newton steps	652	654	655	655	**652**	654	654
# Linear solver steps	5837	6791	5222	3898	**3363**	3041	2999
Max. # linear solver iterations	115	51	43	35	**31**	34	33
Fraction of spectrum	N/A	0.9999997	0.9999998	0.9999998	**0.9999998**	0.9999998	0.9999998

Validation simulation with multiple transients

In flooding optimization studies, reservoir engineers typically change injection and production rates to come up with viable production scenarios. In our second validation simulation we therefore simulated the 1000 days of production with continuously varying injection rates in all eight injectors. We note that this is an extreme scenario that would not be used in practice, but it does test the performance of the POD based preconditioner in case of multiple transients. **Fig. 5** depicts an example of the rates used in four of the injectors, switching randomly between values of 0, 300 and 600 bbl/day at 10-day intervals. **Table 4** presents the results of the multiple-transient test, omitting some details compared to the previous tables. For this test we only used the training simulation in which all injection wells were excited. In this case the reduction in total and linear solver times further increases to 70 % and almost 80 % respectively; see also Table 1, column 4. As illustrated in Table 4 a number of 10 basis functions resulted in the minimal solver time, although the difference between using 5, 10, 15 or 20 basis functions was not very large. Interestingly, the results obtained with only five basis functions were still good, although in theory a number of nine functions (corresponding to eight injectors plus the average pressure field) is required to describe all possible linear combinations of the injectors for arbitrary production settings. An explanation is that solving correctly for the first average-pressure during the preconditioning is the dominating requirement to obtain good performance in solving the subsequent full system equations. Further research into this aspect is necessary.

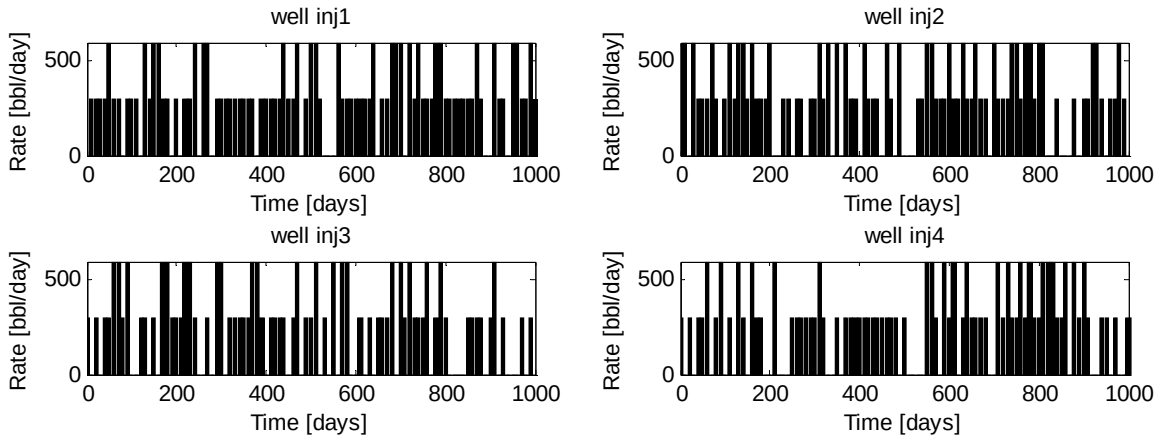

Fig. 5. Randomly fluctuating injection rates for four of the injectors.

TABLE 4 – TEST RESULTS WITH VARYING NUMBER OF BASIS FUNCTIONS (BF) DERIVED FROM TRAINING SIMULATION WITH ALL INJECTORS; MULTIPLE-TRANSIENT TEST

	AMG	5 BF	**10 BF**	15 BF	20 BF	25 BF	30 BF
Total solver time [s]	247	73	**72**	73	74	81	80
Linear solver time [s]	221	46	**45**	47	47	51	53
# Time steps	306	306	**308**	308	306	308	308
# Newton steps	747	748	**757**	758	750	755	755
# Linear solver steps	9989	6334	**5826**	5713	5419	5457	5480

Conclusions

- We developed a new method for the solution of the pressure equation in reservoir simulation and embedded it in a proprietary reservoir simulator. The method is based on the use of concepts from reduced-order modeling (ROM), and in particular the use of the proper orthogonal decomposition (POD). It is applicable to IMPES or IMPEC solvers, and to two-stage preconditioning of a fully implicit solution scheme as an alternative to the Algebraic MultiGrid (AMG) method.
- The underlying idea is that the complexity of the dynamics of a reservoir model is primarily related to the number of wells and to a much lesser extent to the number of grid blocks. This knowledge can be used to

construct a basis consisting of a limited number of pressure field (the number of which is primarily related to the number of wells). Any pressure field can then be expressed in terms of this limited number of basis function.

- For cases where the compressibility of the reservoir fluids is very small and saturation approximately decouples from pressure, such as often is the case during water flooding, the number of required basis functions reduces to at most the number of wells. The basis can then simply be obtained by computing steady-state responses for a given saturation distribution, and the number of required basis functions will just be equal to the number of to-be-controlled wells.
- We performed an initial test of the method by simulating a small (approximately 18500 active grid blocks) waterflooding scenario with multiple injectors and producers. We obtained a basis by subsequently switching on and off the injectors and simulating the resulting transient response for short periods of time. We subsequently used this basis to obtain the pressure solutions in two-stage preconditioned full simulations, one in which the injectors were switched-on only once, and one in which the injectors were switched on and off at random for the entire production period.
- Results of the method used as a first-stage preconditioner for the pressure indicated reductions of the linear solver time up to almost 80 %, i.e. a speed-up factor of 5, compared to the use of AMG to compute the pressures. The simulation results, in terms of wellbore pressure response and oil and water production, were nearly identical.
- The method requires a certain amount of pre-computation in the form of short 'training simulations' and is therefore believed to be particularly useful for situations were it is required to run many reservoir simulations with small changes in the input variables (well control settings or reservoir parameters) as occur during iterative flooding optimization, assisted history matching or parameter uncertainty assessment. In that case the overhead of the short training simulations is shared by a large number of full simulations. Alternatively, the basis can be computed 'on the fly', using a small number of the most recently converged results.
- Further testing of the method is required. This concerns in particular testing against optimized AMG performance under situations of strong compressibility, gravity-driven flow, compositional effects, or strong capillary forces, for which cases the performance of the POD-based method is uncertain. However, for situations where compressibility is low and where the flow is primarily driven by injection and production from wells, such as is often the case in water flooding, the initial test results are encouraging.

Nomenclature

$\mathbf{A}$ = coefficient matrix in linear system of equations
$\mathbf{b}$ = right-hand side in linear system of equations
$\mathbf{C}$ = (generalized) covariance matrix
$\mathbf{D}$ = auxiliary matrix
E = energy content
i = counter
ℓ = number of coefficients
m = number of snap shots
t = time
$\mathbf{U}$ = matrix of left singular vectors
$\mathbf{V}$ = matrix of right singular vectors
$\mathbf{x}$ = vector of state variables
$\mathbf{X}$ = snap shot matrix
z = coefficient
$\mathbf{z}$ = vector of coefficients
λ = eigenvalue
$\mathbf{\Lambda}$ = matrix of eigenvalues
$\boldsymbol{\varphi}$ = basis function
$\mathbf{\Phi}$ = matrix of basis functions

Subscripts

r = reduced

Superscripts

T = transposed

References

Antoulas, A.C., 2005: *Approximation of large-scale dynamical systems*, SIAM, Philadelphia.

Astrid, P., Weiland, S., Willcox, K., Backx, T. 2008. Missing point estimation in models described by proper orthogonal decomposition IEEE Trans. Aut. Control **53** (10) 2237-2251. DOI: 10.1109/TAC.2008.2006102.

Cardoso, M.A., Durlofsky, L.J. and Sarma, P. 2009. Development and application of reduced-order modeling procedures for subsurface flow simulation. *Int. J. for Num. Meth. in Eng.* **77** (9) 1322-1350. DOI: 10.1002/nme.2453.

Cardoso, M.A., and Durlofsky, L.J.. 2010a. Linearized reduced-order models for subsurface flow simulation. *J. of Comp. Phys.* **229** (3) 681-700. DOI: 10.1016/j.jcp.2009.10.004.

Cardoso, M.A., and Durlofsky, L.J., 2010b: Use of reduced-order modeling procedures for production optimization. *SPEJ* **15** (2) 426-435. DOI: 10.2118/119057-PA.

Gharbi, R., Smaoui, N. and Peters, E.J. 1997. Prediction of unstable fluid displacements in porous media using the Karhunen-Loève decomposition. *In Situ* **21** 331-356.

Heijn, T., Markovinović, R. and Jansen, J.D. 2004. Generation of low-order reservoir models using system-theoretical concepts. *SPEJ* **9** (2) 202-218. DOI: 10.2118/88361-PA.

Kaleta, M.P., Hanea, R.G., Heemink, A.W. and Jansen, J.D. 2010. Model-reduced gradient-based history matching. *Computational Geosciences*. Published online. DOI: 10.1007/s10596-010-9203-5.

Markovinović, R. and Jansen, J.D. 2006. Accelerating iterative solution methods using reduced-order models as solution predictors. *Int. J. for Num. Meth. in Eng.* **68** 525-541. DOI: 10.1002/nme.1721.

Schilders, W,H.A., Van der Vorst, H.A. and Rommes, J. (eds.) 2008. *Model order reduction: theory, research aspects and applications.* Springer, Berlin.

Van Doren, J.F.M., Markovinović, R. and Jansen, J.D. 2006. Reduced-order optimal control of waterflooding using POD. *Computational Geosciences* **10** (1) 137-158. DOI: 10.1007/s10596-005-9014-2.

Van Essen, G.M., Zandvliet, M.J., Van den Hof, P.M.J., Bosgra, O.H. and Jansen, J.D. 2009. Robust water flooding optimization of multiple geological scenarios. *SPEJ* **14** (1) 202-210. DOI: 10.2118/102913-PA.

Vermeulen, P.T.M., Heemink, A.W. and Te Stroet, C.B.M.. 2004. Reduced models for linear groundwater flow models using empirical orthogonal functions. *Advances in water resources* **27** 57-69. DOI: 10.1016/j.advwatres.2003.09.008.

Wallis, J.R., Kendall, R.P. and Little, T.E. 1985. Constrained residual acceleration of conjugate residual methods. Paper SPE13536 presented at the Reservoir Simulation Symposium Dallas, USA, February 10-13. DOI: 10.2118/13536-MS.

Zandvliet, M.J., van Doren, J.F.M., Bosgra, O.H., Jansen, J.D. and van den Hof, P.M.J. 2008. Controllability, observability and identifiability in single-phase porous media flow. *Computational Geosciences* **12** (4) 605-622. DOI: 10.1007/s10596-008-9100-3.