

Use of reduced-order models in well control optimization

Jan Dirk Jansen¹ · Louis J. Durlafsky²

Received: 7 May 2015 / Revised: 8 December 2015 / Accepted: 2 January 2016 /

Published online: 22 February 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Many aspects of reservoir management can be expected to benefit from the application of computational optimization procedures. The focus of this review paper is on well control optimization, which entails the determination of well settings, such as flow rates or bottom hole pressures, that maximize a particular objective function. As is the case with most reservoir-related optimizations, this problem is in general computationally demanding since function evaluations require reservoir simulation runs. Here we describe reduced-order modeling procedures, which act to accelerate these simulation runs, and discuss their use within the context of well control optimization. The techniques considered apply proper orthogonal decomposition (POD), which enables the representation of reservoir states (e.g., pressure and saturation in every grid block) in terms of a highly reduced set of variables. Two basic approaches are described—the direct application of POD-based reduction at each Newton iteration, and a trajectory piecewise linearization (POD-TPWL) procedure that applies POD to a linearized representation of the governing equations. Both procedures require one or more pre-processing ‘training’ simulation runs using the original full-order model. The use of both gradient-based optimization methods (including adjoint procedures) and direct search approaches with reduced-order models is described. Several concepts relevant to the general topic, including adjoint formulations and controllability, are also reviewed. Numerical results are presented for both approaches. In particular, the

✉ Jan Dirk Jansen
j.d.jansen@tudelft.nl

Louis J. Durlafsky
lou@stanford.edu

¹ Department of Geoscience and Engineering, Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands

² Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305-2220, USA

POD-TPWL procedure is applied to a computationally demanding bi-objective optimization problem, where it is shown to provide reasonable accuracy and a high degree of speedup.

Keywords Production optimization · Reduced-order models · Reservoir simulation · Controllability · Proper orthogonal decomposition · Trajectory piecewise linearization

1 Introduction

A wide variety of reservoir management issues can be addressed through the application of formal optimization procedures. In planning new wells, key questions include the determination of the optimum well location, well type (injector or producer; vertical, horizontal or multilateral) and drilling schedule, given operational and economic constraints. For existing wells, significant benefit can be achieved through optimization of well settings (well controls) such as bottom hole pressures (BHPs) or flow rates, as a function of time. By optimizing these settings, oil production can be increased and the time at which injected fluids appear at production wells (referred to as breakthrough time) can be delayed. The use of optimization for different aspects of reservoir characterization and management is discussed in, e.g., Echeverría Ciaurri et al. (2011b), Jansen (2011), Oliver and Chen (2011) and Velez-Langs (2005).

In this work, our focus will be on the well control problem. This is a form of model-based optimization, also referred to as flooding optimization, recovery optimization, sweep optimization, life-cycle optimization, or production optimization. We note that the latter name may be confusing because it is traditionally used for short-term optimization of well rates (without large simulation models, and on a time scale of days to months), rather than for optimization over the entire life of the field (simulation-based, and on a time scale of years to decades as considered in this paper). The determination of optimal well settings generally requires that the forward flow simulation problem be run many times (depending on the underlying optimization algorithm and the size of the search space, tens to thousands of simulations may be needed). If the forward flow simulation is itself computationally demanding, which will be the case when the model contains a large number of grid blocks and/or complicated physics, the computational demands for the optimization can be excessive. This has motivated the development of reduced-order models, or more generally surrogate models. Ideally, surrogate models should run much faster than the full-order representation while providing a sufficiently accurate approximation of the model outputs required for the optimization.

Our goal in this paper is to provide a review and synthesis of approaches for model-order reduction that are applicable within the context of well control optimization. The approaches considered involve the application of proper orthogonal decomposition (POD) to solutions, or ‘snapshots,’ generated during one or more ‘training’ simulations. This enables subsequent solutions of the forward model, which are required during optimization, to be represented as linear

combinations of a relatively small number of basis functions. These basis functions are simply the columns of the POD basis matrix, and the unknowns are the coefficients multiplying each basis function. Depending on the specific way in which POD is used with the simulation equations, a range of speedups (and model accuracies) can be achieved. POD-based reduced-order models have been used in a wide variety of application areas, and the general method is known under various names such as the Karhunen–Loève decomposition or the method of empirical orthogonal functions. The use of POD for various applications involving subsurface flow simulation is described in, e.g., Astrid et al. (2011), Cardoso et al. (2009), Gharbi et al. (1997), Gildin et al. (2006), Heijn et al. (2004), Kaleta et al. (2011), Krogstad (2011), Markovinović and Jansen (2006), van Doren et al. (2006) and Vermeulen et al. (2004).

One reduced-order modeling procedure, referred to as POD-based trajectory piecewise linearization or POD-TPWL (Rewiński and White 2003), entails the combined use of the POD representation with linearization around saved states. These saved states are solutions of the full-order model generated during one or more training runs. The linearization also requires that the Jacobian matrices associated with the converged solutions be saved. The linearized representation is projected into a low-dimensional subspace using POD, which enables subsequent solutions to be computed very efficiently. POD-TPWL has been applied for reservoir simulation in Cardoso and Durlofsky (2010a, b), He et al. (2011), He and Durlofsky (2014, 2015).

The general problem of well control optimization has been addressed using both gradient-based and derivative-free procedures, and we will now discuss some of the relevant literature. The most efficient gradient-based approaches apply adjoint techniques to provide gradients of the cost function with respect to the control parameters. These methods are effective and very efficient but, because they are invasive with respect to the flow simulator, can only be applied when full access to the simulator exists. Adjoint-based approaches were first applied for well control optimization by Ramirez (1987). More recently, these techniques have been incorporated by, e.g., Brouwer and Jansen (2004), Sarma et al. (2008, 2006), van Essen et al. (2011), Kourounis et al. (2014). For further references, see the review paper Jansen (2011). Prior to their use for production optimization, adjoint-based techniques were applied for history matching; for a detailed review, see Oliver et al. (2008). Other techniques originally developed for history matching were also later applied to the well optimization problem. Examples include streamline-based optimization of well rates, as described in Alhuthali et al. (2007) and Thiele and Batycky (2006), or the use of ensemble methods (EnOpt), in which an approximate gradient is computed without adjoints (Su and Oliver 2010). Another way to obtain approximate gradient information is through the use of simultaneous perturbation stochastic approximation (SPSA); see Spall (1998) for details on this method, and Wang et al. (2009) for a comparison of SPSA, EnOpt and adjoint-based approaches in well control optimization.

Derivative-free methods have also been studied for this problem. The techniques considered can be classified as stochastic global search methods and local

deterministic approaches. The stochastic search algorithms applied to this problem include genetic algorithms (Goldberg 1989) and particle swarm optimization (Kennedy and Eberhart 1995). These algorithms are noninvasive and parallelize readily, though they are population-based and thus require a large number of function evaluations (reservoir simulation runs in this context). Applications for well control optimization appear in Almeida et al. (2007), Matott et al. (2006), Echeverría Ciaurri et al. (2011a), Isebor and Durlofsky (2014) (note that, in Matott et al. 2006; Isebor and Durlofsky 2014, well location was also optimized). Direct-search methods that have been applied for well control optimization include generalized pattern search (Audet and Dennis Jr 2002; Kolda et al. 2003; Conn et al. 2009), mesh adaptive direct search (Audet and Dennis Jr 2006; Conn et al. 2009) and Hooke-Jeeves direct search, Hooke and Jeeves (1961). Optimization results using derivative-free methods, for a variety of well control optimization problems including some with general (nonlinear) constraints, are presented in Echeverría Ciaurri et al. (2011a, b).

The use of reduced-order models for well control optimization has been considered in a few previous studies. The first such effort within the context of reservoir simulation appears to be that of van Doren et al. (2006), where reduced-order models were incorporated into an adjoint-based optimization. Occasional retraining (i.e., regeneration of the POD basis) was applied in this work. The speedups achieved relative to using full-order models in the optimization were relatively modest, though this work did demonstrate the viability of using ROMs for this application.

The POD-TPWL procedure has also been used for well control optimization (Cardoso and Durlofsky 2010a, b; He et al. 2011; He and Durlofsky 2014). This approach has been coupled with both gradient-based and derivative-free (generalized pattern search) optimization techniques. In Cardoso and Durlofsky (2010a, b) the gradients were computed using numerical differentiation, which is viable with POD-TPWL because forward simulations are very inexpensive. The method has also been applied to perform bi-objective optimization (Cardoso and Durlofsky 2010a).

This paper proceeds as follows. In Sect. 2, we present the optimization problem and provide brief overviews of optimization procedures that are well suited for use with reduced-order modeling. Section 3 presents the basic POD procedure and discusses the general concept of controllability. The POD-based and TPWL reduced-order modeling (ROM) procedures, and their use in the context of well control optimization, are described in Section 4. Optimization results for an example problem appear in Sect. 5. Concluding remarks are provided in Sect. 6.

2 Optimization problem and algorithms

In this section we introduce the basic optimization problem and then briefly describe two optimization procedures that have been used with reduced-order models – the adjoint (gradient-based) approach and the use of the derivative-free generalized

pattern search. We begin by discussing the underlying reservoir simulation equations.

2.1 Reservoir simulation

All optimization methods are used in combination with a reservoir simulator, which computes the time evolution of the reservoir state using some form of time and spatial discretization of the underlying partial differential equations (Aziz and Settari 1979). In an abstract form the simulator can be represented as

$$\mathbf{g}_{n+1}(\mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{u}_{n+1}) = \mathbf{0}, \quad n = 0, \dots, N - 1, \quad (1)$$

where subscripts refer to discrete instants n of time, with N the total number of time steps. Furthermore $\mathbf{g} : \mathbb{R}^k \rightarrow \mathbb{R}^k$ is a vector-valued nonlinear function, $\mathbf{x} \in K \subset \mathbb{R}^k$ is a vector of reservoir state variables (pressures, saturations or component concentrations), and $\mathbf{u} \in M \subset \mathbb{R}^m$ is a vector of control variables in the wells. In practice \mathbf{u} could represent any combination of well flow rates, bottom hole pressures (BHPs), or tubing head pressures (THPs). The sets K and M are subsets of the set of real numbers because their elements are constrained to stay within physical limits; e.g., pressures are always positive and saturations have values between zero and one. Starting from given initial conditions \mathbf{x}_0 , the implicit recursive Eq. (1) is typically solved using a time stepping algorithm with Newton iteration to reduce the residual at each time step to a preset tolerance.

In the following we will frequently use an even more compact notation to represent the simulator as

$$\mathbf{g}(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \quad (2)$$

where \mathbf{g} , \mathbf{x} and \mathbf{u} (without subscripts) can be interpreted as ‘concatenated vectors’

$$\mathbf{g} = [\mathbf{g}_1^T, \mathbf{g}_2^T, \dots, \mathbf{g}_N^T]^T, \quad \mathbf{g} : \mathbb{R}^p \rightarrow \mathbb{R}^p, \quad (3)$$

$$\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_N^T]^T, \quad \mathbf{x} \in P \subset \mathbb{R}^p, \quad (4)$$

$$\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_N^T]^T, \quad \mathbf{u} \in Q \subset \mathbb{R}^q. \quad (5)$$

In these expressions $p = k \times N$ and $q \leq m \times N$, where the inequality holds if the number of control time steps is smaller than the number of simulation time steps N .

2.2 Well control optimization problem

The production optimization problem can be expressed as

$$\min_{\mathbf{u} \in Q} J(\mathbf{u}, \mathbf{x}(\mathbf{u})) \quad \text{subject to} \quad \mathbf{c}(\mathbf{u}, \mathbf{x}(\mathbf{u})) \leq \mathbf{0}, \quad (6)$$

where $J \in \mathbb{R}$ is the objective function we seek to minimize and $\mathbf{c} : \mathbb{R}^{p+q} \rightarrow \mathbb{R}^r$ represents the nonlinear constraints. Linear input constraints and bounds on the input are included in the set $Q \subset \mathbb{R}^q$. In the remainder of this paper we will tacitly

assume that physical limitations or other constraints on \mathbf{u} and \mathbf{x} are present and simply write $\mathbf{u} \in \mathbb{R}^q$ etc. Nonlinear state constraints appear if we specify, e.g., maximum water-oil ratio or minimum oil rate. Typical objective functions could be the negative of cumulative oil produced or the negative of net present value (NPV). The latter is computed as:

$$J_{NPV}(\mathbf{u}, \mathbf{x}(\mathbf{u})) = - \sum_{i=1}^{N_t} \frac{r_o \times q_{o,i}(\mathbf{x}(\mathbf{u})) - c_{wp} \times q_{wp,i}(\mathbf{x}(\mathbf{u})) - c_{wi} \times q_{wi,i}(\mathbf{u})}{(1 + \beta)^i}, \quad (7)$$

where N_t is the total production time (in years), r_o is oil price, c_{wp} and c_{wi} are the water production and injection costs, β is the yearly discount rate, and $q_{o,i}$, $q_{wp,i}$ and $q_{wi,i}$ are the oil and water production rates and the water injection rate, respectively, for year i . Here it is assumed that we can directly control the water injection rates, which are therefore a direct function of \mathbf{u} , whereas the oil and water production rates are functions of the states \mathbf{x} (more specifically, of the well-block pressures and saturations) and are therefore an indirect function of \mathbf{u} . Prices and costs are in units of \$/STB ('STB' refers to stock tank barrel; 1 STB = 0.1590 m³) and flow rates are in units of STB/year. For a given \mathbf{u} , the flow rate quantities, as well as the degree of constraint violation (if any), are computed by performing a reservoir simulation.

Both gradient-based and derivative-free algorithms will be described. Within the context of derivative-free approaches, the POD-based reduced-order models developed here are best suited for use with local deterministic (rather than global stochastic) search techniques. This is because these ROMs can be expected to retain accuracy only for pressure and saturation states that lie within some neighborhood of the training simulation. Well control optimization is an iterative process because the reservoir flow equations are too complex to solve for the optimal control vector $\hat{\mathbf{u}}$ in one step. Using local search methods, the vector of controls \mathbf{u}^{k+1} at iteration $k + 1$ differs relatively slightly from \mathbf{u}^k at the previous iteration (assuming algorithm parameters are specified appropriately), which suggests that the states associated with iterations k and $k + 1$ will also resemble one another. Thus, the POD-based ROM can be expected to maintain accuracy for some number of iterations. With a global stochastic search method, by contrast, \mathbf{u}^{k+1} and \mathbf{u}^k can differ substantially (i.e., they may correspond to entirely different regions of the search space), so we cannot expect the POD representation to necessarily maintain accuracy.

2.3 Adjoint-based optimization

Within the wide class of optimization methods applicable to large-scale problems, the most efficient approaches use the gradient of the objective function with respect to the controls:

$$\nabla J = \left(\frac{dJ}{d\mathbf{u}} \right)^T. \quad (8)$$

Because the objective function is typically a complicated function of \mathbf{u} , through its dependence on \mathbf{u} via \mathbf{x} , computation of the derivative $dJ/d\mathbf{u}$ is not straightforward. Setting aside for now the nonlinear constraints \mathbf{c} , the relationship between \mathbf{x} and \mathbf{u} is

governed by the system equations given in Eq. (2). The derivative can therefore be computed formally through implicit differentiation (Bryson and Ho 1975):

$$\frac{dJ}{d\mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}} + \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}} - \frac{\partial J}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{u}}. \quad (9)$$

Introducing the auxiliary variable

$$\boldsymbol{\lambda}^T = \frac{\partial J}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1}, \quad (10)$$

Eq. (9) can be rewritten as

$$\frac{dJ}{d\mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}}{\partial \mathbf{u}}. \quad (11)$$

Here $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a ‘concatenated vector’ of so called co-state variables or Lagrange multipliers, and $\partial \mathbf{g} / \partial \mathbf{x} \in \mathbb{R}^{p \times p}$ is a ‘concatenated Jacobian matrix’ with a special sparse structure. A detailed analysis shows that after expansion of the ‘concatenated quantities,’ the co-state vectors $\boldsymbol{\lambda}_n$ can be computed from a linear implicit recursive relationship solved backward in time (see Kraaijevanger et al. 2007). Computation of the derivative $dJ/d\mathbf{u}$ with the adjoint method therefore requires a forward nonlinear simulation to compute the state variables \mathbf{x} , and a backward linear simulation to compute the co-state variables $\boldsymbol{\lambda}$, after which the derivative follows from Eq. (11). This is computationally a very efficient procedure in comparison with a finite difference approach, which would require one forward base simulation plus q forward perturbed simulations to compute $dJ/d\mathbf{u}$. We note that the backward computation may require an adapted version of the pre-conditioning scheme used for the forward computation (see Han et al. 2013). Moreover, it is generally necessary to compute the forward simulations with tight nonlinear solver tolerances in order to obtain accurate adjoint gradients.

The adjoint method can also be derived by considering a constrained optimization problem

$$\min_{\mathbf{u}} \bar{J}(\mathbf{u}, \mathbf{x}, \boldsymbol{\lambda}) = \min_{\mathbf{u}} [J(\mathbf{u}, \mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{u}, \mathbf{x})], \quad (12)$$

where the Lagrange multiplier vector $\boldsymbol{\lambda}$ serves to adjoin the ‘system constraints’ \mathbf{g} to the objective function J . For an optimal control $\hat{\mathbf{u}}$, the first derivatives of the augmented objective function \bar{J} with respect to \mathbf{u} , \mathbf{x} and $\boldsymbol{\lambda}$ must vanish. For a near-optimal control \mathbf{u} we again find Eq. (11). Nonlinear equality constraints can be incorporated in a similar manner. Nonlinear inequality constraints are generally much more difficult to implement and various approaches have been explored. We refer to Jansen (2011), Kourounis et al. (2014), Sarma et al. (2008) for further discussion.

After computation of the gradient, various gradient-based algorithms are available to search for the next improved control vector \mathbf{u} ; see e.g., Nocedal and Wright (2006). In the simplest case, a steepest descent is used to take a step along the gradient direction, either with a fixed step size, or with a line search algorithm.

More advanced methods search along a conjugate direction, obtained with the aid of an approximation to the Hessian matrix d^2J/du^2 . Among such ‘quasi-Newton’ methods the limited-memory Broyden-Fletcher-Goldfarb-Shannon (LBFGS) method is particularly suited for very large problems; see Oliver et al. (2008). Other approaches involve interior point or trust region methods.

2.4 Optimization using generalized pattern search

Generalized pattern search (GPS) algorithms are described in detail in Audet and Dennis Jr (2002), Kolda et al. (2003), Conn et al. (2009). Their use for well control optimization is discussed in Echeverría Ciaurri et al. (2011a, b), and their use with reduced-order models is presented in He et al. (2011). The algorithm used in He et al. (2011) applies ‘polling’ within the search space. Designating the vector of controls at iteration k as \mathbf{u}^k , during iteration $k + 1$ the algorithm evaluates $2q$ new solutions, where q is the dimension of the search space. The set of controls for each new (trial) solution is prescribed by perturbing one component of \mathbf{u}^k by a specified amount Δ . Because both positive and negative perturbations are applied, and because all search directions are considered, $2q$ new solutions must be computed. Of these $2q$ new solutions, the set of controls \mathbf{u} that minimizes J is taken as \mathbf{u}^{k+1} . The algorithm proceeds in this way until no improvement in J is achieved, at which point the stencil size Δ is reduced (different stencil sizes and stencil-size reductions can be used for different variables). The algorithm terminates once a minimum stencil size is reached, or when a maximum number of iterations has been performed.

There are several variants of this GPS approach that have been shown to be effective. Within the context of production optimization, mesh adaptive direct search (MADS), in which the stencil is selected randomly from an asymptotically dense set of directions, has been applied. MADS is expected to have some advantages in cases with noisy objective functions. In existing implementations of GPS and MADS for production optimization, general (nonlinear) constraint satisfaction has been accomplished using both penalty functions and filters (Echeverría Ciaurri et al. 2011a; Isebor and Durlofsky 2014). Filter-based methods share some similarities with bi-objective optimization, as the optimization seeks to minimize both the objective function and the constraint violation. Both the standard GPS and MADS procedures parallelize very naturally since the $O(q)$ function evaluations required at each iteration can be performed on different processors.

3 Controllability of reservoir flow

This section describes some system-theoretical aspects of subsurface flow control as an introduction to ROM. Although this material is not essential to our description of reduced-order modeling procedures, it does provide insight into the often remarkable level of reduction that can be achieved using ROMs. The optimization of oil recovery through manipulating well controls is strongly related to the capacity to influence subsurface flow behavior. In particular, the control of fluid front positions in the

reservoir and phase arrival times at the wells is strongly related to our ability to control streamlines (Alhuthali et al. 2007; Jansen 2011). Streamlines are governed by the spatial derivatives of the pressure field, and the controllability of streamlines is therefore closely related to the controllability of pressures in the reservoir. In the theoretical case of a homogeneous reservoir, the controllability of a fluid front at some distance from a well is very limited (Fyrozjaee and Yortsos 2006; Jansen et al. 2009; Ramakrishnan 2007). However, in the case of strong heterogeneities there is often a significant scope to influence streamlines, and hence front positions, because high-permeable regions can act in some sense as ‘extensions’ to the wells (Jansen 2011).

3.1 Linear systems theory

The concept of controllability originates from linear systems theory (Kailath 1980). A discrete dynamic system is fully controllable if it is possible to bring it from any state to any other state in a finite time by manipulating the controls. Although the reservoir flow equations are typically nonlinear, useful insight can be obtained by first considering a simplified case in which the implicit nonlinear Eq. (1) is replaced by an explicit linear expression:

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{u}_n, \quad n = 0, \dots, N - 1, \tag{13}$$

where $\mathbf{A} \in \mathbb{R}^{k \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times m}$ are called the system matrix and the input matrix respectively. Such a simplified description gives a reasonable description of slightly compressible single-phase flow if \mathbf{x} is taken to represent the reservoir pressure. The input vector \mathbf{u} consists of prescribed flow rates or bottom hole pressures in the wells.

It can be shown that such a system is fully controllable if the controllability Gramian

$$\mathcal{P} \in \mathbb{R}^{k \times k} \triangleq \sum_{i=0}^{k-1} \mathbf{A}^i \mathbf{B} \mathbf{B}^T (\mathbf{A}^T)^i = \mathcal{C} \mathcal{C}^T \tag{14}$$

has full rank (Kailath 1980). This is equivalent to the requirement that the controllability matrix

$$\mathcal{C} \in \mathbb{R}^{k \times k} \triangleq [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{k-1}\mathbf{B}] \tag{15}$$

has full rank. Note that the superscripts of \mathbf{A} are exponents. A physical interpretation of Eqs. (14) and (15) can be obtained by rewriting Eq. (13) as a recursive sequence:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0, \\ \mathbf{x}_2 &= \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1, \\ \mathbf{x}_3 &= \mathbf{A}^3\mathbf{x}_0 + \mathbf{A}^2\mathbf{B}\mathbf{u}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_1 + \mathbf{B}\mathbf{u}_2, \\ &\vdots \\ \mathbf{x}_k &= \mathbf{A}^k\mathbf{x}_0 + \mathbf{A}^{k-1}\mathbf{B}\mathbf{u}_0 + \dots + \mathbf{A}\mathbf{B}\mathbf{u}_{k-2} + \mathbf{B}\mathbf{u}_{k-1}. \end{aligned} \tag{16}$$

Starting from an initial condition $\mathbf{x}_0 = \mathbf{0}$, and a unit impulse input at time zero, i.e., $\mathbf{u}_0 = \mathbf{1}, \mathbf{u}_1 = \mathbf{u}_2 = \dots = \mathbf{u}_{k-1} = \mathbf{0}$, it follows that

$$\dot{\mathbf{x}}_k = \mathbf{A}^{k-1} \mathbf{B}, \tag{17}$$

where the accent indicates that the response results from an impulse input. Eq. (14) can therefore be rewritten as

$$\mathcal{P} = \sum_{i=1}^k \dot{\mathbf{x}}_i (\dot{\mathbf{x}}_i)^T = [\dot{\mathbf{x}}_1 \quad \dot{\mathbf{x}}_2 \quad \dots \quad \dot{\mathbf{x}}_k] \begin{bmatrix} \dot{\mathbf{x}}_1^T \\ \dot{\mathbf{x}}_2^T \\ \vdots \\ \dot{\mathbf{x}}_k^T \end{bmatrix} = \mathcal{C} \mathcal{C}^T. \tag{18}$$

It is thus evident that the controllability matrix $\mathcal{C} = [\dot{\mathbf{x}}_1 \quad \dot{\mathbf{x}}_2 \quad \dots \quad \dot{\mathbf{x}}_k]$ can be interpreted as an impulse response, i.e., a series of subsequent state vectors resulting from a unit impulse input at time zero.

3.2 Spatial covariance

Another interpretation is obtained by considering the definition of (an estimator of) the spatial covariance between elements of the state vector \mathbf{x} :

$$\text{Cov}(\mathbf{x}, \mathbf{x}) \in \mathbb{R}^{k \times k} \triangleq \frac{1}{s-1} [\tilde{\mathbf{x}}_1 \quad \tilde{\mathbf{x}}_2 \quad \dots \quad \tilde{\mathbf{x}}_s] \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \tilde{\mathbf{x}}_2^T \\ \vdots \\ \tilde{\mathbf{x}}_s^T \end{bmatrix}, \tag{19}$$

where s is the sample size, i.e., the number of state vector ‘snapshots’ in time used to estimate the covariance, and $\tilde{\mathbf{x}}$ indicates a centered snapshot (estimated mean subtracted):

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \frac{1}{s} \sum_{j=1}^s \mathbf{x}_j. \tag{20}$$

Comparison of Eqs. (18) and (19) shows that the controllability Gramian \mathcal{P} can be interpreted as a generalized spatial covariance matrix.

3.3 Quantitative controllability

Rank deficiency of either \mathcal{C} or \mathcal{P} implies that not all states can be influenced by manipulating the inputs. A rank test of \mathcal{C} or \mathcal{P} therefore gives a qualitative (yes/no) answer to the question if a reservoir system is fully controllable. However, this definition of controllability does not put any restrictions on the magnitude of the inputs, i.e., it assumes that it is possible to exert positive or negative pressures or flow rates of arbitrary magnitude in the wells, a condition that is clearly unrealistic. A more meaningful, quantitative, measure of controllability is obtained by performing an eigenvalue decomposition of \mathcal{P} , or a singular value decomposition

(SVD) of \mathcal{C} , which allows for determination of the controllable subspace Zandvliet et al. (2008). Choosing the SVD of \mathcal{C} we obtain

$$\Xi \Sigma \Psi^T = \mathcal{C}, \tag{21}$$

where the columns of $\Xi \in \mathbb{R}^{k \times k}$, i.e., the left singular vectors, represent an ordered set of linear combinations of the states \mathbf{x} (i.e., for single-phase flow, linear combinations of the grid block pressures). These linear combinations can be interpreted as spatial pressure patterns that are decreasingly controllable. The corresponding singular values lie on the main diagonal of $\Sigma \in \mathbb{R}^{k \times l}$, where l represents the number of nonzero singular values (those singular values with a magnitude above machine precision).

The span of the first l left singular vectors is just the controllable subspace (i.e. all possible controllable linear combinations of states). In reservoir simulation models we have typically $l \ll k$, with k equal to the number of grid blocks and l on the order of twice the number of wells (Zandvliet et al. 2008). Not surprisingly the most controllable states (grid-block pressures) are those close to the wells and in high-permeability areas. We note that the concept of controllability is a system property and not a model property, and that, although we have presented these ideas in a spatially discretized setting, the same conclusions can be reached by considering a continuous representation. A similar analysis for the spatial derivatives of the pressure field for single-phase flow leads to analogous conclusions: the controllable subspace is of much smaller dimension than the state space, and the most controllable pressure gradients are those in the vicinity of the wells (Jansen et al. 2009). These results are only locally valid in the state space, meaning they refer to controllability of the states close to a reference state (also referred to as a ‘reference trajectory’ in the state space).

3.4 Empirical controllability Gramian

A similar restriction to local controllability around a reference trajectory is unavoidable in the nonlinear analysis of two-phase flow. In that case the results for controllability of the pressure field (although slowly time-varying) are similar to those for single-phase flow, whereas the controllability of saturations is, logically, restricted to regions close to the fluid fronts (van Doren et al. 2013). Among several approximate techniques to analyze the controllability of nonlinear systems, there is one which is closely related to ROM. This concerns the use of an ‘empirical controllability Gramian’ constructed from a large number of system state ‘snapshots’ along a number of reference trajectories, each for a slightly different input (control vector) (Lall et al. 2002). In a simplified form this can be expressed as

$$\mathcal{P}_{emp} = \sum_{j=1}^t \sum_{i=1}^s \mathbf{x}_i^j (\mathbf{x}_i^j)^T = \sum_{j=1}^t [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_s]^j \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_s^T \end{bmatrix}^j, \tag{22}$$

where s is the number of snapshots and t is the number of different control vectors. For an application to reservoir simulation, see van Doren et al. (2013). In contrast to Eq. (18), the controls \mathbf{u}^j , $j = 1, \dots, t$ here are typically not unit impulses, and the ‘snapshot matrices’ $[\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_s]^j$ are therefore not unit impulse responses. However, as in the linear case, it is possible to compute a (locally) controllable subspace by taking an SVD of the snapshot matrix to determine the l singular vectors corresponding to the l nonzero singular values. As will be discussed in the next section, this is the same procedure that is followed to compute reduced-order models with the aid of POD. Indeed, the fact that we have in general $l \ll k$ means that the controllable subspace is of much lower order than the state space. This is precisely the reason why there is such a large scope for model-order reduction in reservoir flow modeling and optimization. Note that because the empirical Gramians are computed on the basis of an $s \times k$ snapshot matrix (and not on the basis of a $k \times k$ system matrix as is the case for ordinary Grammians), their computation is feasible, even for large models.

4 Optimization using ROM

In this section we describe the use of POD in reservoir simulation and gradient-based optimization. We also introduce the POD-based trajectory piecewise linearization (TPWL) procedure, which has been used with both gradient-based and direct search (derivative-free) optimizers.

4.1 Basic POD-based ROM

Consider a sequence of s state vector snapshots collected in a matrix

$$\mathbf{X} \in \mathbb{R}^{k \times s} \triangleq [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_s]. \quad (23)$$

Here we assume that the snapshots have been generated by numerically solving the system of nonlinear equations defined in (1). The number of snapshots s is at most the number of simulation steps N , and nearly always $s \ll k$, so \mathbf{X} is a tall, thin matrix. Moreover, because the state vector typically contains elements of a completely different physical nature (e.g., pressures and saturations) and with large differences in magnitude, it is customary to work with separate snapshot matrices for the different parts of the state vector. For example, in the case of two-phase (oil-water) flow it is customary to generate separate $N_c \times s$ matrices, designated \mathbf{X}_p and \mathbf{X}_s , for the pressures and the saturations, where N_c is the number of grid blocks in the reservoir model (Cardoso et al. 2009; van Doren et al. 2006). In the following we will tacitly assume the use of separate snapshots, write N_c instead of k , and drop the subscripts p and s .

As discussed above, $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{N_c \times N_c}$ may be interpreted as a generalized spatial covariance between the elements of the state vector. Because the rank of a tall matrix can never be larger than the number of columns, it is clear that $\mathbf{X}\mathbf{X}^T$ is a

(very) low-rank approximation of the generalized covariance. This effect can be quantified by performing an SVD of \mathbf{X} :

$$\mathbf{X} = \mathbf{\Xi}\mathbf{\Sigma}\mathbf{\Psi}^T, \quad (24)$$

in which case we typically find a number κ of nonzero singular values that is never larger than s , and always much smaller than N_c . The corresponding singular vectors constitute the first κ columns of the $N_c \times N_c$ matrix $\mathbf{\Xi}$ and can therefore be represented as an $N_c \times \kappa$ matrix $\mathbf{\Phi}$. A further reduction is often obtained by only maintaining the first l columns of $\mathbf{\Phi}$, where $l < \kappa$, based on an ‘energy’ criterion in which the total amount of energy present in the snapshots is defined as $E_t = \sum_{i=1}^{\kappa} \sigma_i^2$, where σ_i are the singular values. The number of columns to be kept is then computed as the largest number $l \in \{1, \dots, \kappa\}$ that satisfies

$$E = \frac{\sum_{i=1}^l \sigma_i^2}{E_t} \leq \alpha, \quad (25)$$

where α denotes the fraction of energy that should be maintained. Typical values of α are between 0.95 and 1.

Apparently the information in the large covariance matrix $\mathbf{X}\mathbf{X}^T$ is limited and can be represented with a small number of singular values and singular vectors. As discussed in Sect. 3.4 this is equivalent to the statement that the local controllability is very limited. ROM aims at using this physical effect to reduce the number of state variables in the mathematical description of the reservoir. The basic idea is to express the full state vector \mathbf{x}_n as a linear combination of a small number of basis functions:

$$\mathbf{x}_n = \mathbf{\Phi}\mathbf{z}_n + \mathbf{e}_n, \quad (26)$$

where \mathbf{z}_n is a time-varying vector of coefficients, \mathbf{e}_n is a time-varying error term, and the columns of $\mathbf{\Phi}$ are the time-independent basis functions. Because $\mathbf{\Phi}$ results from an SVD it is an orthogonal matrix. Therefore the error \mathbf{e}_n is zero for any value of \mathbf{x} that is in the span of $\mathbf{\Phi}$, which is only the case if \mathbf{x} forms part of a full rank snapshot matrix \mathbf{X} . For the more general case of a vector \mathbf{x} that is not in the span of the snapshot matrix the error will not be zero and we have

$$\mathbf{x}_n \approx \mathbf{\Phi}\mathbf{z}_n. \quad (27)$$

We refer to Antoulas (2005) for a detailed discussion of the error in ROM in general, and to Cardoso et al. (2009) for strategies to optimize the snapshot selection to minimize the error in a reservoir simulation setting. Under the assumption that the error is small enough to be neglected, we can now rewrite the explicit linear equation 13 as

$$\mathbf{\Phi}\mathbf{z}_{n+1} = \mathbf{A}\mathbf{\Phi}\mathbf{z}_n + \mathbf{B}\mathbf{u}_n. \quad (28)$$

Because $\mathbf{\Phi}$ is orthogonal, it follows that

$$\mathbf{z}_{n+1} = \mathbf{\Phi}^T \mathbf{A} \mathbf{\Phi} \mathbf{z}_n + \mathbf{\Phi}^T \mathbf{B} \mathbf{u}_n. \quad (29)$$

This premultiplication by $\mathbf{\Phi}^T$ is known as a Galerkin projection.

Eq. 29 is a low-order dynamic equation in terms of the short ($l \times 1$) vector \mathbf{z} with an $l \times l$ system matrix $\bar{\mathbf{A}} \triangleq \Phi^T \mathbf{A} \Phi$ and an $l \times m$ input matrix $\bar{\mathbf{B}} \triangleq \Phi^T \mathbf{B}$. Unfortunately, $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are full matrices, whereas \mathbf{A} and \mathbf{B} are typically very sparse. However, for linear subsurface flow systems the order reduction is usually significant, say from $O(10^5)$ to $O(10^2)$. Moreover, the reduced-size matrix $\bar{\mathbf{A}}$ needs to be constructed only once, as a preprocessing step. As a result, evaluation of the (full) low-order system can be orders of magnitude faster than evaluation of the (sparse) high-order system. A similar increase in speed is obtained in case of a linear implicit formulation, which involves solving a system of equations at every time step. POD is therefore a very effective model-order reduction technique for linear flow problems (Vermeulen et al. 2004).

To apply a similar model-order reduction approach to the nonlinear equation (1), we need to consider that reservoir simulation involves solving a linear system of equations at every Newton iteration k . This procedure can be represented as

$$\mathbf{J}_{n+1}^k \mathbf{r}_{n+1}^{k+1} = \mathbf{g}_{n+1}^k, \quad (30)$$

$$\mathbf{x}_{n+1}^{k+1} = \mathbf{x}_{n+1}^k + \mathbf{r}_{n+1}^{k+1}, \quad (31)$$

where \mathbf{r}_{n+1}^{k+1} is the update vector and $\mathbf{J}_{n+1}^k = \partial \mathbf{g}_{n+1}^k / \partial \mathbf{x}_{n+1}^k$ is the Jacobian matrix. A reduced-order version of Eq. 30 can now be constructed:

$$\Phi^T \mathbf{J}_{n+1}^k \Phi \mathbf{r}_{n+1}^{k+1} = \Phi^T \mathbf{g}_{n+1}^k. \quad (32)$$

As in the linear case, the reduced Jacobian $\bar{\mathbf{J}} \triangleq \Phi^T \mathbf{J} \Phi$ is now a reduced-size but full matrix, whereas \mathbf{J} is sparse. Moreover, in contrast to the linear case, it is now necessary to assemble the full-order Jacobian matrix \mathbf{J} at every Newton iteration and then to perform the multiplications required in Eq. 32. As a result, the computational gains from using POD for nonlinear systems are much smaller than those for linear cases. Specifically, typical speedup factors reported for subsurface flow applications (relative to solutions of the full-order equations obtained with optimized solvers) are between 3 and 10 (Cardoso et al. 2009; Heijn et al. 2004; Markovinović and Jansen 2006).

We note that with current linear solvers and preconditioners, the total time spent solving the linear system for large scale models typically does not exceed 90 %, in which case the theoretical limit of speedup with POD would be about 10 times. Over the past years, attempts have been made to improve the performance of POD for nonlinear systems, in particular with the aid of discrete empirical interpolation (DEIM), a technique that aims to approximate the nonlinear function at a limited number of points in space. For applications to reservoir flow, see Ghasemi et al. (2015), Sampaio Pinto et al. (2015). Moreover, it has been argued that higher-quality basis functions (better capturing spatial continuity) are obtained by performing the SVD using a higher-order tensor formalism rather than the snapshot matrix as described above. In that case the state snapshots are not stored as vectors and combined in a matrix, but are instead stored as matrices (2D or 3D, depending

on the simulation performed) and combined in 3D or 4D tensors. For a reservoir application, see Insuasty et al. (2015).

4.2 Example of POD-based ROM

As an example of the use of POD, consider the two-dimensional reservoir model depicted in Fig. 1. The model contains 10,201 cells and 61 wells: 36 injectors and 25 producers arranged in a regular five-spot pattern. The constant porosity ϕ is 0.3, and the isotropic permeability field represents a channel structure with values between 7 and 1700 mD. The rock, oil and water compressibilities are $c_r = c_o = c_w = 7 \times 10^{-5} \text{psi}^{-1}$, and the oil and water viscosities are $\mu_o = 0.5 \text{ cp}$ and $\mu_w = 1.0 \text{ cp}$, respectively. Relative permeabilities are represented using a Corey model with exponents $n_o = n_w = 3$, end points $k_{ro}^0 = 0.9$ and $k_{rw}^0 = 0.6$, and residual saturations $S_{or} = S_{wc} = 0.2$. The mid-field injectors operate at fixed rates of 109 bpd, and the injectors at the edges and corners operate at half and quarter values of the full rates, respectively. The producers are operated at fixed bottom hole pressures that are 145 psi below the initial reservoir pressure. No further well constraints are imposed. We used an in-house Matlab simulator with a fully-implicit time integration with Newton iterations and an automatic time-stepping scheme. Simulation of the model for a period of 22.8 years, corresponding to the injection of one moveable pore volume of water, results in water breakthrough in all producers.

For the purpose of constructing the basis matrix Φ , we used two sets of 290 unevenly spaced snapshots of the simulated pressure and saturation fields. We applied the highest snapshot density just after start-up of the wells to capture the transient behaviour of the pressures. Some representative examples of the saturation snapshots are shown in Fig. 2. After constructing the snapshot matrices \mathbf{X}_p and \mathbf{X}_s , performing SVDs (without subtraction of the means), and using a cut-off value $\alpha = 0.9999$, we obtained the two matrices Φ_p and Φ_s , containing 120 and 121 columns respectively. The rapid drop of the singular values of the snapshot matrices is evident in Fig. 3. Figure 4 depicts the first two basis functions for pressure and saturation. Unlike predefined basis functions, as used in, e.g., a Fourier expansion,

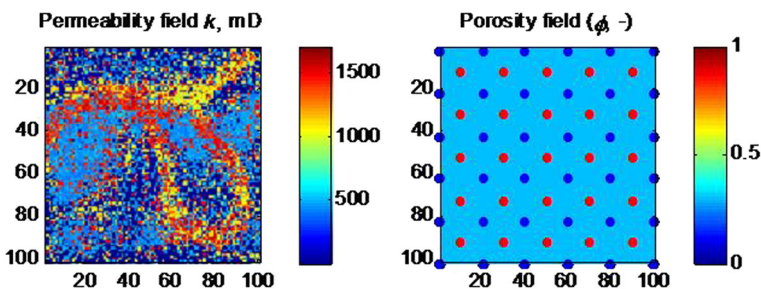


Fig. 1 Permeability field (left) and constant porosity field with wells (right); blue dots indicate injectors and red dots indicate producers. (Color figure online)

the data-driven basis functions of the POD method display spatial features representing the dominant dynamics.

Repetition of the forward simulation with the aid of POD, using the same input as was used to generate the snapshots, resulted in solutions with accurate pressure and saturation fields; see Fig. 5. The gain in computational speed was considerable for this example: the reduced-order run required only 5 % of the time needed for the full-order run (33 % including the pre-processing of the basis functions). We note that the linear solver used in the full-order solution is not highly specialized for the reservoir flow equations and was not optimized, so the speedups due to POD are somewhat greater than would be achieved using a state-of-the-art simulator. Moreover, we did optimize the basis functions (by varying the number of snapshots, the snapshot spacing, and the values of α), which took several trial runs. Repetition of the simulation with well specifications that are different from those used to generate the snapshots will lead to larger errors. A priori determination of these errors is in general not feasible, but clearly the errors will grow with increasing deviation of the inputs, and therefore of the corresponding states, from their reference trajectories. See, e.g., Cardoso et al. (2009), Heijn et al. (2004) for POD-based simulation results where the controls differ significantly from those used in the training runs.

4.3 Use of POD-based ROM in optimization

As noted above, in most cases it is not possible to quantify a priori the error between the solutions of a POD-based reduced-order model and the corresponding full-order model. Moreover, reducing the error to acceptable levels may require significant

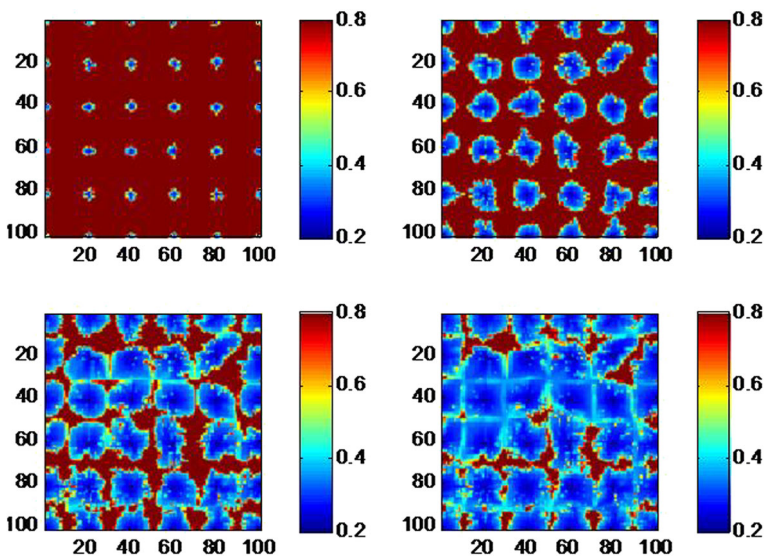


Fig. 2 Oil saturation snapshots at four moments in time

Fig. 3 Singular values versus number of columns in Φ for pressure (*left*) and saturation (*right*)

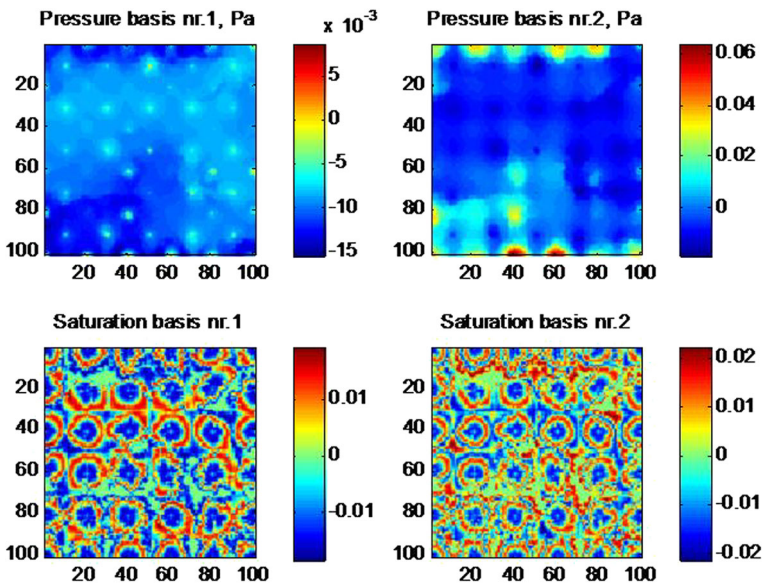
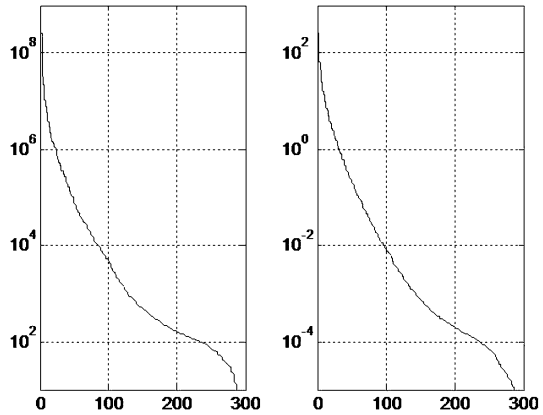


Fig. 4 First two basis functions for pressure (*top*) and saturation (*bottom*)

tuning which increases the pre-processing time, thus reducing the computational benefits. This suggests the use of POD to create approximate solutions for situations where the availability of such a low-order approximation can help to subsequently obtain a detailed high-order solution in a significantly reduced computational time. One way to accomplish this is to use POD solutions as an initial guess or as a preprocessor in the iterative solution of linear systems of equations (Astrid et al. 2011; Markovinoić and Jansen 2006). Another possible use of these solutions is in well control optimization, in which the same model is simulated many times for a

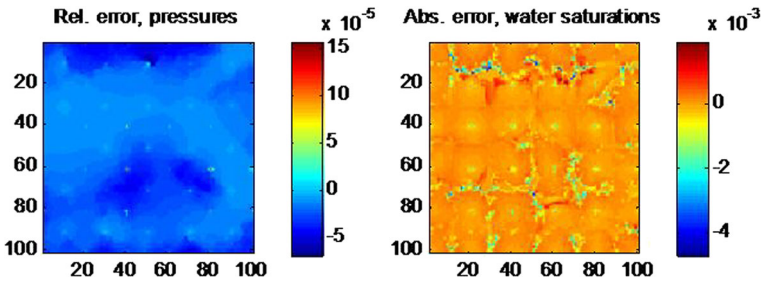


Fig. 5 Relative differences in pressure (*left*) and absolute differences in saturation (*right*)

large number of slightly different inputs in order to iteratively compute the optimal well control. In that case it is attractive to first perform a full-order simulation and to use this to construct a POD-based reduced-order model, then iterate to the optimal control using reduced-order simulations, and finally perform a full-order simulation to verify the results. If necessary these steps may be repeated, leading to a nested sequence of full-order and low-order simulations. This procedure is illustrated in the flow chart in Fig. 6.

Van Doren et al. (2006) applied this approach to adjoint-based optimization of water flooding. To compute the adjoint solutions they applied a similar projection to the ‘backward’ adjoint equations as to the ‘forward’ simulation equations. This

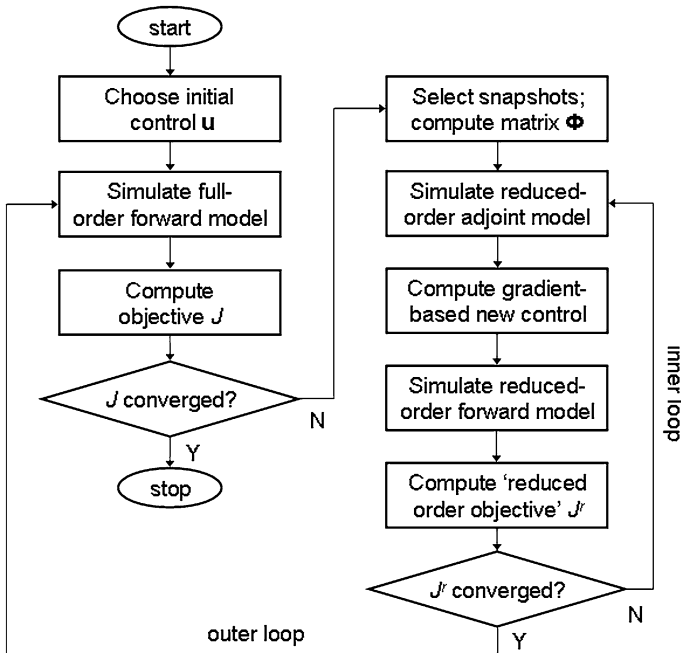


Fig. 6 Flow chart for reduced-order water flooding optimization (after van Doren et al. 2006)

provided reduced-order state vectors \mathbf{z} and reduced-order Lagrange multiplier vectors $\boldsymbol{\mu}$. As discussed above, the nonlinear nature of the two-phase flow equations requires that we perform the matrix multiplication $\boldsymbol{\Phi}^T \mathbf{J} \boldsymbol{\Phi}$ every Newton iteration during the forward simulation. Moreover, it was necessary to frequently recompute the $\boldsymbol{\Phi}$ matrix during the outer iterations of the optimization process. As a result the reduction in computing time was quite modest and never exceeded 35 % of the time needed for optimization using only the full-order model.

A different way to use POD in gradient-based well control optimization was suggested by Kaleta et al. (2011). In this approach an approximate adjoint is obtained by using the transpose of the reduced-order Jacobian matrix. This alleviates the programming of an adjoint code, although it still requires that Jacobian matrices from the forward solution be available to the user. Moreover, it requires a significant number of forward simulations with slightly perturbed well control settings as a preprocessing step. The method was not fully tested, but numerical results from a related approach involving gradient-based optimization for history matching indicated a limited numerical efficiency; i.e., a speedup of about a factor of 2 compared to a full-order numerical perturbation approach (Kaleta et al. 2011).

4.4 POD-TPWL procedure

The POD-TPWL procedure is in general more approximate than the POD-based ROM described above, but it can provide more substantial speedups. The POD-TPWL method is described in detail in Cardoso and Durlofsky (2010a), He et al. (2011), He and Durlofsky (2014, 2015); our more concise discussion here follows the development in He and Durlofsky (2010). The basic idea is to represent new solutions as linearizations around previously simulated solutions, and to project this linearized representation into a low-order subspace. We begin by expressing the discretized flow equations (1) as:

$$\mathbf{g}_{n+1}(\mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{u}_{n+1}) = \mathbf{a}_{n+1}(\mathbf{x}_{n+1}, \mathbf{x}_n) + \mathbf{f}_{n+1}(\mathbf{x}_{n+1}) + \mathbf{q}_{n+1}(\mathbf{x}_{n+1}, \mathbf{u}_{n+1}) = \mathbf{0}, \quad (33)$$

where \mathbf{a} , \mathbf{f} and \mathbf{q} are the discretized accumulation, flux and injection/production terms, respectively.

Like the ROM described in Sect. 4.1, the POD-TPWL method requires one or more full-order training simulations in which particular sequences of controls (BHPs in this case) for each well are specified. From these training simulations, the states and Jacobian matrices (for the converged states) are saved at each time step. Then, for subsequent simulations, the solution is expressed as a linear expansion around a saved state. Thus we write:

$$\begin{aligned} \mathbf{g}(\mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{u}_{n+1}) &\approx \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{u}_{i+1}) + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_{i+1} (\mathbf{x}_{n+1} - \mathbf{x}_{i+1}) \\ &\quad + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_i (\mathbf{x}_n - \mathbf{x}_i) + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right)_{i+1} (\mathbf{u}_{n+1} - \mathbf{u}_{i+1}), \end{aligned} \quad (34)$$

where \mathbf{u}_{n+1} is the new set of controls (which are specified), \mathbf{x}_{n+1} is the new state we wish to determine, \mathbf{x}_n is the (known) previous state, \mathbf{x}_i and \mathbf{x}_{i+1} designate two sequential saved states (generated during the training runs), and \mathbf{u}_{i+1} are the controls used in the training run. Note that, in our description here, subscripts i and $i + 1$ always designate saved information. The detailed matrices $(\partial \mathbf{g} / \partial \mathbf{x})$, $(\partial \mathbf{g} / \partial \mathbf{x})_{i+1}$ and $(\partial \mathbf{g} / \partial \mathbf{u})_{i+1}$ are saved, along with the states, upon convergence at each time step of the training runs.

Combining Eqs. (33) and (34), and recalling that the Jacobian matrix is given by $\mathbf{J}_{i+1} = \partial \mathbf{g}_{i+1} / \partial \mathbf{x}_{i+1}$, enables us to represent new solutions (\mathbf{x}_{n+1}) as (see Cardoso and Durlofsky 2010a for details):

$$\mathbf{J}_{i+1}(\mathbf{x}_{n+1} - \mathbf{x}_{i+1}) = - \left[\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_i}(\mathbf{x}_n - \mathbf{x}_i) + \frac{\partial \mathbf{q}_{i+1}}{\partial \mathbf{u}_{i+1}}(\mathbf{u}_{n+1} - \mathbf{u}_{i+1}) \right]. \tag{35}$$

Because Eq. (35) is linear, its solution does not require iteration. This equation is, however, still in the original high-dimensional (full-order) space.

We thus introduce the POD representation, which maps from the high-dimensional space to a low-dimensional subspace; i.e., we apply $\mathbf{x} \approx \mathbf{\Phi} \mathbf{z}$, where $\mathbf{\Phi}$ is computed as described in Sect. 4.1. The $\mathbf{\Phi}$ matrix is of dimensions $2N_c \times l$, where N_c is the number of grid blocks in the full-order (high-fidelity) model and $l = l_p + l_s$ is the total number of basis vectors, with l_p and l_s the number of basis vectors associated with pressure and saturation states respectively. Significant reduction is achieved because $l \ll N_c$. A major contribution to the speedup seen with POD-TPWL results from the application of POD and left-projection to linearized equations rather than to nonlinear equations. Thus, it is possible to precompute the reduced-order system matrices, whereas for the nonlinear case these matrices must be recomputed at every Newton iteration.

Representing \mathbf{x} in Eq. (35) as $\mathbf{x} \approx \mathbf{\Phi} \mathbf{z}$, and then performing a Galerkin projection (premultiplying both sides of the equation by $\mathbf{\Phi}^T$) gives, after some manipulation,

$$\mathbf{z}_{n+1} = \mathbf{z}_{i+1} - (\mathbf{J}_{i+1}^r)^{-1} \left[\left(\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_i} \right)^r (\mathbf{z}_n - \mathbf{z}_i) + \left(\frac{\partial \mathbf{q}_{i+1}}{\partial \mathbf{u}_{i+1}} \right)^r (\mathbf{u}_{n+1} - \mathbf{u}_{i+1}) \right], \tag{36}$$

where the superscript r indicates reduced and

$$\mathbf{J}_{i+1}^r = \mathbf{\Phi}^T \mathbf{J}_{i+1} \mathbf{\Phi}, \left(\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_i} \right)^r = \mathbf{\Phi}^T \left(\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_i} \right) \mathbf{\Phi}, \left(\frac{\partial \mathbf{q}_{i+1}}{\partial \mathbf{u}_{i+1}} \right)^r = \mathbf{\Phi}^T \left(\frac{\partial \mathbf{q}_{i+1}}{\partial \mathbf{u}_{i+1}} \right). \tag{37}$$

The vectors and matrices appearing in Eq. (36) are of dimensions l and $l \times l$ respectively, so this equation can be solved very efficiently. Evaluation of Eq. (36) requires particular (reduced) saved states \mathbf{z}_i and \mathbf{z}_{i+1} around which the expansion is performed. See Cardoso and Durlofsky (2010a) and He and Durlofsky (2014) for the specific treatments used for this determination.

The POD-TPWL model given by Eq. (36) was shown to provide accurate results for a variety of examples in Cardoso and Durlofsky (2010a), and this is the approach used in Sect. 5 below. It was observed, however, that the model can encounter numerical stability problems for some cases, such as those in which the densities of

the oil and water phases differ significantly. This motivated the development of procedures to stabilize the POD-TPWL representation. One such stabilization approach entails the selection of an optimized basis (i.e., the determination of a particular l_p, l_s combination) such that the stability properties of the resulting model are improved (He et al. 2011). This is accomplished by minimizing the spectral radius of an appropriately defined amplification matrix, as described in He et al. (2011) and He and Durlofsky (2015). This method cannot guarantee numerical stability, but it does provide stable POD-TPWL models in some cases. An alternative approach, which appears to perform more reliably than the basis optimization procedure, is to apply a Petrov–Galerkin projection instead of a Galerkin projection procedure. In this case, rather than premultiplying the linearized equation by Φ^T , we premultiply by $(\mathbf{J}_{i+1}\Phi)^T$. The definitions of the reduced matrices in Eq. (37) change accordingly. See He and Durlofsky (2014, 2015) for details and for examples demonstrating the enhanced stability, for both oil-water and oil-gas compositional POD-TPWL models, achieved through use of this Petrov–Galerkin approach.

In the example below involving optimization with POD-TPWL, the full-order runs needed to generate the saved states, Jacobian matrices and Φ matrix are performed using Stanford’s General Purpose Research Simulator, GPRS (Cao 2002; Jiang 2007). The simulator has been modified to output the arrays required by POD-TPWL.

4.5 Use of POD-TPWL in optimization

In the example presented in Sect. 5 below, we use two POD-TPWL training runs and prescribe the well controls for these runs using a heuristic procedure. No retraining is applied during the course of the optimization. This approach will be seen to perform well for the case considered, but more sophisticated treatments have been found to be advantageous in other cases. We now describe two such approaches.

POD-TPWL was used within the context of a generalized pattern search (GPS) procedure by He et al. (2011). The approach used in that work is depicted in Fig. 7. A training simulation with well BHPs defined by the initial guess is first performed. The states and Jacobian matrices are saved and the POD-TPWL model is constructed as described in Sect. 4.4. Then the GPS optimization is started, and the POD-TPWL representation is used for function evaluations. After a specified number of function evaluations are performed, GPS is paused and a training simulation is run at the current best point (the specified number of function evaluations can vary during the course of the optimization). POD-TPWL is then retrained at this point and GPS is resumed.

It is occasionally observed that, upon retraining, the objective function of the current point, evaluated using the full-order model, is suboptimal relative to that of the previous full-order solution. This inconsistency can occur when the POD-TPWL solution loses accuracy because it is too far from the most recent training case. When this problem is detected, the search is restarted from the previous retraining

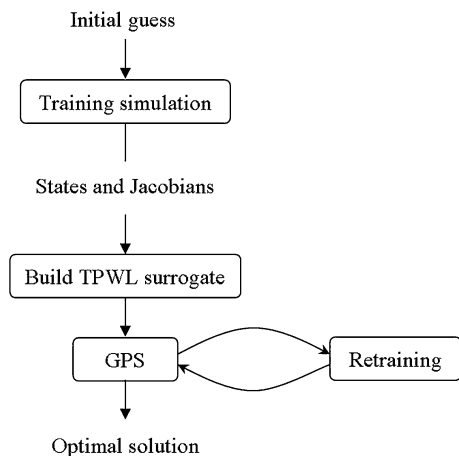
point and the number of function evaluations until the next retraining is reduced. The size of the GPS mesh may also be reduced. We note that it should be possible to incorporate more sophisticated criteria, possibly based on mass balance errors in the POD-TPWL model (which are straightforward to compute), for retraining. Such procedures will be considered in future work.

Another approach, which entails a two-stage training and optimization procedure, was applied by Cardoso and Durlofsky (2010b). In that work, a gradient-based technique, with gradients computed by numerical finite difference (using POD-TPWL for function evaluations), was applied for the optimization. The first stage of the procedure entails two initial (heuristic) training runs and construction of the POD-TPWL model, followed by the full POD-TPWL-based optimization. After the convergence of this (first-stage) optimization, an additional full-order training run, using the final controls from the first-stage optimization, is simulated. A new Φ matrix, based on the results from all three training runs, is then constructed. Using this new Φ matrix along with the additional saved states, a second-stage POD-TPWL-based optimization is performed. This procedure (full-order simulation using controls from the previous optimization stage, followed by POD-TPWL model construction, followed by POD-TPWL-based optimization) can be repeated if necessary.

5 Optimization example using POD-TPWL

Because of the substantial speedups achievable using the POD-TPWL procedure, the method is well suited for use in computationally demanding optimization problems. Here we consider a bi-objective optimization, where we seek to maximize cumulative oil produced and minimize cumulative water injected. These objectives are in conflict, since additional water injection generally leads to increased oil production.

Fig. 7 Flowchart for use of POD-TPWL with generalized pattern search (from He et al. 2011)



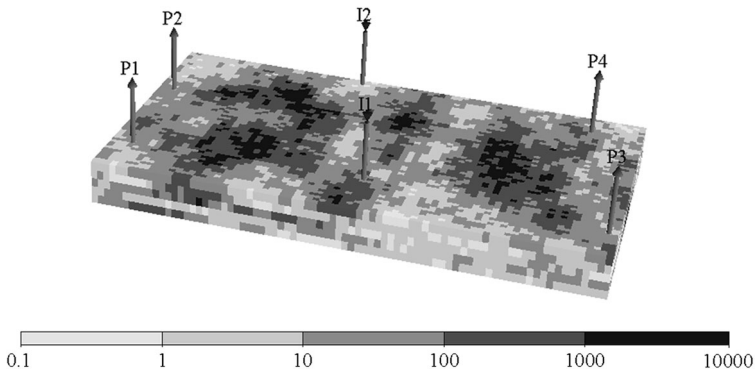


Fig. 8 Reservoir model (24,000 grid blocks) with four production wells and two injection wells used for bi-objective optimization. Permeability in x -direction (in mD) is shown (from Cardoso and Durlofsky 2010a)

This example is from Cardoso and Durlofsky (2010a), and that paper should be consulted for further details. To construct the Pareto front, we perform a large number of single-objective optimizations, with the objective function modified in each optimization run. Specifically, we maximize the net present value of the process using a sequence of oil prices (from \$10/stb to \$150/stb), in increments of \$2/stb. Water injection and production costs are \$10/stb in all cases. By varying the ratio of the oil price to water costs, we vary the relative weightings of the two objectives, which enables us to generate the convex portion of the Pareto front.

The problem set up is as follows. The reservoir model is a modified portion of a channelized system presented in Christie and Blunt (2001). The model, shown in Fig. 8, contains 24,000 grid blocks. There are four production wells and two injection wells. Permeability is a diagonal tensor, with $k_x = k_y$ in all blocks. Vertical permeability is prescribed as $k_z = 0.3k_x$ in the channels and $k_z = 10^{-3}k_x$ elsewhere. The mean k_x is 418 mD and the mean porosity is 0.203. Oil and water densities and viscosities are $\rho_o = 45 \text{ lb/ft}^3$, $\rho_w = 60 \text{ lb/ft}^3$, $\mu_o = 3.0 \text{ cp}$, and $\mu_w = 0.5 \text{ cp}$, respectively. The system is incompressible.

The simulation time frame is 3000 days. We optimize only the BHPs of the four production wells; injectors are prescribed to operate continuously at 10,000 psi. The producer BHP bounds are 1000 and 4000 psia. These BHPs are updated every 250 days during the optimization, and the maximum change allowed from one control step to the next is 250 psia. There are thus a total of 48 control variables (12 for each production well). The optimizations are performed using “fmincon” in Matlab. We compute the required gradients using numerical finite differences, which is feasible when POD-TPWL is applied. Two training runs are used to generate the POD-TPWL model. In one training run constant BHPs are applied, while in the second training run a random sequence of BHPs are used (see Cardoso and Durlofsky 2010a for the detailed well specifications). The POD-TPWL basis Φ is constructed with $\ell_p = 250$ and $\ell_s = 150$.

Fig. 9 Pareto front showing tradeoff between optimized cumulative oil produced and cumulative water injected (from Cardoso and Durlofsky 2010a)

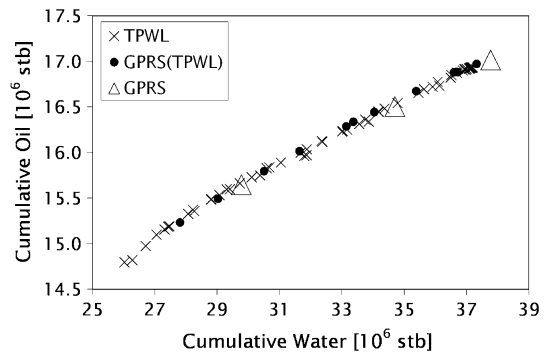


Figure 9 displays a plot of cumulative oil produced versus cumulative water injected for the optimized solutions. This curve represents the Pareto front for the bi-objective optimization. Results using three different approaches are displayed in this figure. The \times 's indicate optimizations performed using the POD-TPWL model (71 such points were generated, meaning 71 single-objective optimization runs were performed). The solid circles represent high-fidelity (GPRS) results computed for some of the points on the Pareto front using the well settings determined from the POD-TPWL optimizations. The fact that the \times 's and solid circles nearly coincide indicates the accuracy of key quantities in the POD-TPWL model relative to those computed by GPRS. For three cases we also performed the full optimization using the high-fidelity model (these results are indicated by triangles). The correspondence of these results to the other two sets of points further demonstrates the applicability of POD-TPWL-based optimization for this problem.

The 71 optimizations performed using POD-TPWL required a total of nearly 14,000 runs. The runtime speedup using POD-TPWL (relative to GPRS) for this case was about a factor of 500. Including overhead, POD-TPWL required less than 4 h for these computations. We estimate that the use of the high-fidelity GPRS model for this case would have required about 69 days of computation. Thus the benefit of POD-TPWL for this example is very substantial. Note, however, that the use of an adjoint procedure (rather than numerical finite difference) for computing gradients would result in many fewer simulations. Thus, the computational requirements using the high-fidelity GPRS model could be greatly reduced in this case, though they would still exceed that required by POD-TPWL.

6 Concluding remarks

In this paper we described several concepts and methods relevant to the use of reduced-order models within the context of production optimization. The reduced-order modeling techniques considered applied proper orthogonal decomposition (POD) to provide a low-order representation of the state vector. In one set of approaches, the POD-based reduction is applied at each Newton iteration at every time step. Although this leads to a much smaller linear system of equations than in the full-order system, speedups using this technique are limited because the

construction of the reduced Jacobian matrix is itself time consuming. Another approach, trajectory piecewise linearization, combines a linearized representation with POD. This leads to a more approximate reduced-order model, but speedups are in general much more substantial than those achieved using POD alone.

Reduced-order models are well suited for use with local optimization techniques since the states encountered during the optimization, at least at early iterations, can be expected to resemble those generated during the pre-processing training runs. Strategies for using POD-based reduced-order models within the context of an adjoint procedure were described, as were approaches for using POD-TPWL models with direct search (GPS) and gradient-based (numerical finite difference) methods. Numerical results for a computationally challenging bi-objective optimization problem demonstrate the potential advantages offered by these approaches.

There are a number of issues that must be addressed before these or other related reduced-order modeling procedures can be used reliably for practical optimization problems. Areas deserving of further investigation include the extension and testing of reduced-order modeling techniques for large realistic problems, the development of procedures to determine the range of validity of a particular reduced-order model (such techniques, within the context of an optimization procedure, will indicate when retraining is necessary), and the development of approaches for further accelerating POD-based techniques that do not apply linearization. Topics along these lines will be addressed in future work.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Alhuthali AH, Oyerinde D, Datta-Gupta A (2007) Optimal waterflood management using rate control. *SPE Reserv Eval Eng* 10(5):539–551
- Almeida LF, Tupac YJ, Lazo JG, Pacheco MA, Vellasco MMBR (2007) Evolutionary optimization of smart-wells control under technical uncertainties. SPE paper 107872 presented at the Latin American & Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, 15–18 April 2007
- Antoulas AC (2005) Approximation of large-scale dynamical systems. SIAM, Philadelphia
- Astrid P, Papaioannou GV, Vink JC, Jansen JD (2011) Pressure preconditioning using proper orthogonal decomposition. Paper SPE 141922 presented at the SPE Reservoir Simulation Symposium, The Woodlands, USA, 21–23 February 2011
- Audet C, Dennis JE Jr (2002) Analysis of generalized pattern searches. *SIAM J Optim* 13(3):889–903
- Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. *SIAM J Optim* 17(1):188–217
- Aziz K, Settari A (1979) Petroleum reservoir simulation. Kluwer Academic Publishers, Dordrecht
- Bryson AE, Ho Y-C (1975) Applied optimal control. Taylor and Francis, Levittown
- Brouwer DR, Jansen JD (2004) Dynamic optimization of waterflooding with smart wells using optimal control theory. *SPE J* 9(4):391–402
- Cao H (2002) Development of Techniques for General Purpose Simulators. PhD thesis, Department of Petroleum Engineering, Stanford University, Stanford, USA

- Cardoso MA, Durlofsky LJ (2010) 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. *SPE J* 15:426–435
- Cardoso MA, Durlofsky LJ, Sarma P (2009) Development and application of reduced-order modeling procedures for subsurface flow simulation. *Int J Numer Methods Eng* 77(9):1322–1350
- Christie MA, Blunt MJ (2001) Tenth SPE comparative solution project: a comparison of upscaling techniques. *SPE Reserv Eval Eng* 4:308–317
- Conn AR, Scheinberg K, Vicente LN (2009) Introduction to derivative-free optimization. MPS-SIAM Series on Optimization, MPS-SIAM
- Echeverría Ciaurri D, Isebor OJ, Durlofsky LJ (2011) Application of derivative-free methodologies for generally constrained oil production optimization problems. *Int J Math Model Numer Optim* 2:134–161
- Echeverría Ciaurri D, Mukerji T, Durlofsky LJ (2011) Derivative-free optimization for oil field operations. In: Yang X-S, Koziel S (eds) *Computational optimization and applications in engineering and industry*, SCI 359. Springer, Berlin, pp 19–55
- Fyrozjaee MH, Yortsos Y (2006) Control of a displacement front in potential flow using flow-rate partition. Paper SPE 99524 presented at the SPE Intelligent Energy Conference, Amsterdam, The Netherlands, 11–13 April 2006
- Gharbi R, Smaoui N, Peters EJ (1997) Prediction of unstable fluid displacements in porous media using the Karhunen–Loève decomposition. *Situ* 21(4):331–356
- Ghasemi M, Yang Y, Gildin E, Efendiev Y, Calo V (2015) Fast multiscale reservoir simulations using POD-DEIM model reduction. Paper 173271-MS presented at the SPE Reservoir Simulation Symposium, Houston, USA, 22–25 February 2015
- Gildin E, Klie H, Rodriguez A, Wheeler MF, Bishop RH (2006) Development of low-order controllers for high-order reservoir models and smart wells. Paper SPE 102214 presented at the SPE Annual Technical Conference and Exhibition, San Antonio, 24–27 September 2006
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Boston
- Han C, Wallis J, Sarma P et al (2013) Adaptation of the CPR preconditioner for efficient solution of the adjoint equation. *SPE J* 18(2):207–213
- He J, Durlofsky LJ (2010) Use of linearized reduced-order modeling and pattern search methods for optimization of oil production. In: *Proceedings of the 2nd International Conference on Engineering Optimization*, Lisbon, Portugal, 6–9 September 2010
- He J, Saetrom J, Durlofsky LJ (2011) Enhanced linearized reduced-order models for subsurface flow simulation. *J Comput Phys* 230:8313–8341
- He J, Durlofsky LJ (2014) Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization. *SPE J* 19:858–872
- He J, Durlofsky LJ (2015) Constraint reduction procedures for reduced-order subsurface flow models based on POD-TPWL. *Int J Numer Methods Eng* 103:1–30
- Heijn T, Markovinović R, Jansen JD (2004) Generation of low-order reservoir models using system-theoretical concepts. *SPE J* 9(2):202–218
- Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. *J ACM* 8(2):212–229
- Isebor OJ, Durlofsky LJ, Echeverría Ciaurri D (2014) A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Comput Geosci* 18:463–482
- Insuasty E, Van den Hof PMJ, Weiland S, Jansen JD (2015) Spatial-temporal tensor decompositions for characterizing control-relevant flow profiles in reservoir models. Paper 173235-MS presented at the SPE Reservoir Simulation Symposium, Houston, USA, 22–25 February 2015
- Jansen JD (2011) Adjoint-based optimization of multiphase flow through porous media—a review. *Comput Fluids* 46(1):40–51
- Jansen JD, van Doren JFM, Heidary-Fyrozjaee M, Yortsos YC (2009) Front controllability in two-phase porous media flow. In: Van den Hof PMJ, Scherer C, Heuberger PSC (eds) *Model-based control: bridging rigorous theory and advanced control*. Springer, Berlin, pp 203–219
- Jiang Y (2007) *Techniques for modeling complex reservoirs and advanced wells*. PhD thesis, Department of Energy Resources Engineering, Stanford University, Stanford, USA
- Kailath T (1980) *Linear systems*. Prentice-Hall, Englewood Cliffs

- Kaleta MP (2011) Model-reduced gradient-based history matching, PhD thesis, Delft University of Technology, The Netherlands
- Kaleta MP, Hanea RG, Heemink AW, Jansen JD (2011) Model-reduced gradient-based history matching. *Comput Geosci* 15(1):135–153
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp 1942–1948
- Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev* 45(3):385–482
- Kourounis D, Durlafsky LJ, Jansen JD, Aziz K (2014) Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Comput Geosci* 18(2):117–137
- Kraaijevanger JFBM, Egberts PJP, Valstar JR, Buurman HW (2007) Optimal waterflood design using the adjoint method. Paper SPE 105764 presented at the SPE Reservoir Simulation Symposium, Houston, USA, 26–28 February 2007
- Krogstad S (2011) A sparse basis POD for model reduction of multiphase compressible flow. Paper SPE 141973 presented at the SPE Reservoir Simulation Symposium, The Woodlands, USA, 21–23 February 2011
- Lall S, Marsden JE, Glavaski S (2002) A subspace approach to balanced truncation for model reduction of nonlinear control systems. *Int J Robust Nonlinear Control* 12:514–535
- Markovinović R, Jansen JD (2006) Accelerating iterative solution methods using reduced-order models as solution predictors. *Int J Numer Methods Eng* 68(5):525–541
- Matott LS, Rabideau AJ, Craig JR (2006) Pump-and-treat optimization using analytic element method flow models. *Adv Water Resour* 29:760–775
- Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York
- Oliver DS, Chen Y (2011) Recent progress on reservoir history matching: a review. *Comput Geosci* 15(1):185–221
- Oliver DS, Reynolds AC, Liu N (2008) Inverse theory for petroleum reservoir characterization and history matching. Cambridge University Press, Cambridge
- Ramakrishnan TS (2007) On reservoir fluid-flow control with smart completions. *SPE Prod Oper* 22(1):4–12
- Ramirez WF (1987) Application of optimal control theory to enhanced oil recovery. Elsevier, Amsterdam
- Rewienski M, White J (2003) A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans Comput Aided Des Integr Circuits Syst* 22:155–170
- Sampaio Pinto MA, Ghasemi M, Sorek N, Gildin E, Schiozer J (2015) Hybrid optimization for closed-loop reservoir management. Paper 173278-MS presented at the SPE Reservoir Simulation Symposium, Houston, USA, 22–25 February 2015
- Sarma P, Chen WH, Durlafsky LJ, Aziz K (2008) Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reserv Eval Eng* 11:326–339
- Sarma P, Durlafsky LJ, Aziz K, Chen WH (2006) Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Comput Geosci* 10:3–36
- Spall JC (1998) Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans Aerospace Electron Syst* 34(3):817–823
- Su H-J, Oliver DS (2010) Smart-well production optimization using an ensemble-based method. *SPE Reserv Eval Eng* 13(6):884–892
- Theile MR, Batycky RP (2006) Using streamline-derived injection efficiencies for improved waterflood management. *SPE Reserv Eval Eng* 9(2):187–196
- van Doren JFM, Markovinović R, Jansen JD (2006) Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comput Geosci* 10(1):137–158
- van Doren JFM, Van den Hof PMJ, Bosgra OH, Jansen JD (2013) Controllability and observability in two-phase porous media flow. *Comput Geosci* 17(5):773–788
- van Essen GM, van den Hof PMJ, Jansen JD (2011) Hierarchical long-term and short-term production optimization. *SPE J* 16(1):191–199
- Velez-Langs O (2005) Genetic algorithms in oil industry: an overview. *J Pet Sci Eng* 47:15–22
- Vermeulen PTM, Heemink AW, Stroet CBMT (2004) Reduced models for linear groundwater flow models using empirical orthogonal functions. *Adv Water Resour* 27:57–69

-
- Wang C, Li G, Reynolds AC (2009) Production optimization in closed-loop reservoir management. *SPE J* 14(3):506–523
- Zandvliet MJ, van Doren JFM, Bosgra OH, Jansen JD, van den Hof PMJ (2008) Controllability, observability and identifiability in single-phase porous media flow. *Comput Geosci* 12(4):605–622