Generation of Low-Order Reservoir Models Using System-Theoretical Concepts

T. Heijn,* SPE, and **R. Markovinović,** Delft U. of Technology, and **J.-D. Jansen,** SPE,
Delft U. of Technology and Shell Intl. E&P

Summary

We present five methods to derive low-order numerical models of two-phase (oil/water) reservoir flow, and illustrate their features with numerical examples. Starting from a known high-order model, these methods apply system-theoretical concepts to reduce the model size. Using a simple but heterogeneous reservoir model, we illustrate that the essential information of the model can be captured by a limited number of state variables (pressures and saturations). Ultimately, we aim at developing computationally efficient algorithms for history matching, optimization, and the design of control strategies for smart wells. In this study we applied (1) modal decomposition, (2) balanced realization, (3) a combination of these two methods, (4) subspace identification, and (5) proper orthogonal decomposition (POD), also known as principal component analysis, Karhunen-Loève decomposition, or the method of empirical orthogonal functions. Methods 1 through 4 result in linear low-order models, which are only valid during a limited time span. However, the POD results in a nonlinear model that remains valid over a much longer period. Methods that result in linear low-order models are not very promising for speeding up reservoir simulation. POD, however, has the potential to improve computational efficiency in the case of multiple simulations of the same reservoir for different well operating strategies, but further research is required to quantify this scope. The potential benefit of low-order models is therefore mainly in the development of low-order control algorithms, and in history matching, where the use of reduced models may form an alternative to classical regularization methods.

Introduction

Smart wells have the potential to increase oil recovery through controlling the pressures or flow rates in the smart well segments. Optimization techniques for reservoir models containing smart wells have been developed to investigate this potential[1,2] and **Fig. 1** represents a waterflooding example of a simple 2D heterogeneous reservoir taken from Ref. 1. At one side of the reservoir, a horizontal smart injection well is installed, and at the opposite side, a horizontal smart production well; the optimization problem involves maximizing oil recovery or net present value over a given time interval by adjusting the flow rates in the smart well segments.

A reservoir model is called a high-order model if it consists of a large number (typically 10^3 to 10^6) of variables (pressures and saturations). Optimization of high-order reservoir models is computationally very intensive and thus time-consuming and expensive. Therefore, we are looking for methods to reduce high-order models to low-order models (typically 10^1 to 10^3 variables) before optimization. The dynamics of high-order reservoir models are usually captured in a smaller degree space than the models initially may imply. Therefore low-order models, found by, for instance, projecting the original state dynamics onto lower-order subspaces, are often sufficiently accurate to describe reservoir dynamics. Based on these low-order models, which contain the most relevant features, low-order controllers can be constructed. Note that controllers also need to be of relatively low order to be of practical value.

In addition, low-order models are of relevance for updating of reservoir model parameters based on measured data from, for example, production tests or time-lapse seismic. This inverse problem, also known as history matching or data assimilation, is well known to be ill-posed because high-order models typically contain many more parameters than can be uniquely determined from the measurements. The inverse problem usually involves minimizing an objective function that represents the difference between modeled and measured data, and a classic way to overcome the ill-posedness is to impose constraints on the solution space for the model parameters through the addition of regularization terms to the objective function. The use of low-order models provides an alternative to classic regularization.

There are two main approaches for deriving low-order models: mathematical reduction of high-order white-box models, and the identification of low-order black-box models directly; respectively, they are illustrated in the upper and the lower branch of **Fig. 2.** White-box models explicitly take the physics of the system into account, whereas black-box models are based on measured input/output behavior only. We will discuss mathematical reduction of a white-box model using modal decomposition, balanced realization, a combination of the two, and POD. While the first three methods result in linear low-order models, the model obtained by the latter method remains nonlinear. Afterward, we will discuss identification of a black-box model. Early attempts to use black-box models in reservoir engineering have been reported by Rowan and Clegg[3] and Chierici.[4] We will use a more recently developed identification method, which is one of the many methods that are available in the measurement and control community at present. Although identification is typically a black-box modeling method, it can also be applied to input/output data of a white-box high-order model. We are not always able to access and to derive low-order models from the mathematical high-order models used in (commercial) reservoir simulators. Therefore, identification can be seen as a useful fifth method of mathematical reduction. Reduction and identification have already successfully been applied to single-phase linear 2D reservoir models.[5] The use of POD to derive low-order proxies of reservoir models was described by Gharbi *et al.,*[6,7] while the use in groundwater flow modeling was described by Vermeulen *et al.*[8] We briefly reported a comparison of the various methods in an earlier publication.[9]

Reservoir Model

The two-phase reservoir model consists of one horizontal layer of $\overline{m}\times\overline{n}$ grid cells. At one side of the reservoir, a horizontal smart injection well is installed, and at the opposite side, a horizontal smart production well (see Fig. 1). In the grid cells containing a well segment, the total flow rate is imposed per timestep. For every grid cell, porosity, permeability, initial pressure, and initial saturation are defined (see **Table 1**). The model is based on the assumptions that the displacement of oil by water is immiscible, porosity is constant over time, the process is isothermal, and that

* Now with Shell E&P U.K.

This paper (SPE 88361) was revised for publication from paper SPE 79674, presented at the 2003 SPE Reservoir Simulation Symposium, Houston, 3–5 February. Original manuscript received for review 3 February 2003. Revised manuscript received 28 January 2004. Manuscript peer approved 18 February 2004.

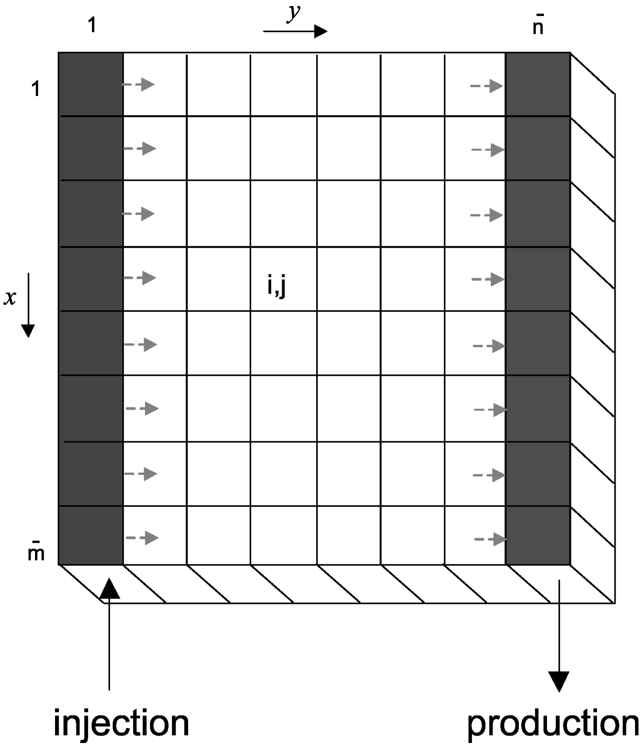

Fig. 1—Top view of a 2D horizontal reservoir model of $\overline{m}\times\overline{n}$ grid cells with an eight-segment horizontal smart well injector installed at the left side and an eight-segment horizontal smart well producer at the right side.

there are no gravity effects. The model has no-flow boundaries at all sides. The behavior of the reservoir in time is modeled by a set of differential equations derived from material balances for oil and water, Darcy's law, and two closure equations for saturation and pressure: $S_w+S_o=1$ (where S_w = water saturation, and S_o = oil saturation) and $p_c=p_o-p_w$ (where p_c = capillary pressure, p_o = pressure in the oil phase, and p_w = pressure in the water phase[10,11]). Spatial finite-difference discretization using a five-point block-centered scheme leads to the following matrix equation in continuous time:

$$\begin{bmatrix}\dot{\mathbf{p}}_o\ (t)\\ \dot{\mathbf{s}}_w\ (t)\end{bmatrix}=\begin{bmatrix}\mathbf{\Psi}_{11} & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_{22}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{V}_{11} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{22}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{T}_{11} & \mathbf{T}_{12}\\ \mathbf{T}_{21} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{p}_o\ (t)\\ \mathbf{s}_w\ (t)\end{bmatrix}$$
$$+\begin{bmatrix}\mathbf{\Psi}_{11} & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_{22}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{q}_o\ (t)\\ \mathbf{q}_w\ (t)\end{bmatrix}=\mathbf{f}(\mathbf{p}_o,\mathbf{s}_w,\mathbf{q}_o,\mathbf{q}_w).\ \ \ldots\ldots (1)$$

The matrix $\Psi_{2\overline{mn}\times 2\overline{mn}}$ accounts for the pore-fluid compressibility and porosity, $\mathbf{V}_{2\overline{mn}\times 2\overline{mn}}$ is a mass matrix (grid-cell volume times density of the fluids), and $\mathbf{T}_{2\overline{mm}\times 2\overline{mn}}$ is formed by the transmissibilities (containing permeabilities and fluid viscosities) at the gridblock boundaries. The vector with the oil pressures p_o and water saturations S_w is called the state vector and has length $2\overline{mn}$. The vector containing the imposed oil and water flow rates, q_o and q_w, in the well segments can be written as $\mathbf{K}_{2\overline{mn}\times 2\overline{m}}\mathbf{u}_{2\overline{m}\times 1}$. $\mathbf{K}$ is a matrix selecting only the elements that contain a well segment; $\mathbf{u}$ consists of total flow rates $q_t=q_w+q_o$ and is called the input vector. In our reservoir model, we implemented the state vector and the input vector as alternating vectors. Because the matrices $\mathbf{V}$, $\mathbf{\Psi}$, $\mathbf{T}$ and the flow rates $\mathbf{q}_o$ and $\mathbf{q}_w$ are functions of $\mathbf{p}_o$ and $\mathbf{s}_w$, Eq. 1 is nonlinear. We can write Eq. 1 in general state space notation as

$$\dot{\mathbf{x}}(t)=\mathbf{f}[\mathbf{x}(t),\mathbf{u}(t)]=\mathbf{A}_c\mathbf{x}(t)+\mathbf{B}_c\mathbf{u}(t),\ \mathbf{y}(t)=\mathbf{C}_c\mathbf{x}(t)\ \ \ldots\ldots\ldots (2)$$

It should be noted that the parameters of the system matrix $\mathbf{A}_c$ and the input matrix $\mathbf{B}_c$ are still functions of the state $\mathbf{x}$. Vector $\mathbf{y}$ is called the output vector. The output matrix $\mathbf{C}_c$ is equal to $\mathbf{K}^T$. The order n of the state space system in Eq. 2 is equal to the total number of state variables (i.e., the pressures and saturations in all gridblocks). Thus, in our reservoir model $n=2\overline{mn}$. From now on we will discuss the theory in the general state space notation of Eq. 2.

Reduction of a High-Order Model to a Linear Low-Order Model

Overview. Most mathematical reduction methods (upper branch of Fig. 2) have been developed for linear systems, so it is reasonable to start with linearizing the nonlinear high-order model. The linearized high-order model is subsequently reorganized to be able to select the most important, "dominant" information. We will discuss reorganization through modal decomposition, balanced realization, and a combination of the two. Finally, the reservoir model is reduced to a low-order model for instance by truncation (i.e., selecting only the dominant information and neglecting the less important information; see **Fig. 3**). The reservoir model, the linearization, and the reduction routines have numerically been implemented in MATLAB, a high-level programming language for technical computing which is frequently used in the measurement and control community.[12]

Linearization. We start with linearizing equation Eq. 2 around $\mathbf{x}=\mathbf{x}^*$ and $\mathbf{u}=\mathbf{u}^*$:

$$\dot{\mathbf{x}}(t)=[\mathbf{f}]_{\mathbf{x}^*,\mathbf{u}^*}+\left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right]_{\mathbf{x}^*,\mathbf{u}^*}[\mathbf{x}(t)-\mathbf{x}^*]+\left[\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right]_{\mathbf{x}^*,\mathbf{u}^*}[\mathbf{u}(t)-\mathbf{u}^*]$$
$$=(\mathbf{A}_c^*-\overline{\mathbf{A}}_c)\mathbf{x}^*+(\mathbf{B}_c^*-\overline{\mathbf{B}}_c)\mathbf{u}^*+\overline{\mathbf{A}}_c\mathbf{x}(t)+\overline{\mathbf{B}}_c\mathbf{u}(t),\ \ \ldots\ldots\ldots (3)$$

$$\mathbf{y}(t)=\mathbf{C}_c\mathbf{x}^*+\mathbf{C}_c[\mathbf{x}(t)-\mathbf{x}^*],\ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

where the higher-order terms have been neglected, $\mathbf{A}_c^*=[\mathbf{A}_c]_{\mathbf{x}^\bullet}$, $\mathbf{B}_c^*=[\mathbf{B}_c]_{\mathrm{x}^\bullet}$, $\overline{\mathbf{A}}_c=[\boldsymbol{\partial}\mathbf{f}/\boldsymbol{\partial}\mathbf{x}]_{\mathrm{x}^\bullet\mathrm{u}^\bullet}$, and $\overline{\mathbf{B}}_c=[\boldsymbol{\partial}\mathbf{f}/\boldsymbol{\partial}\mathbf{u}]_{\mathbf{x}^\bullet\mathbf{u}^\bullet}$. After a change of variables defined by $\bar{\mathbf{x}}=\mathbf{x}+\overline{\mathbf{A}}_c^{-1}(\mathbf{A}_c^*-\overline{\mathbf{A}}_c)\mathbf{x}^*$, and $\bar{\mathbf{y}}=\mathbf{y}+\mathbf{C}_c\overline{\mathbf{A}}_c^{-1}(\mathbf{A}_c^*-\overline{\mathbf{A}}_c)\mathbf{x}^*$, under the condition that the inverse of $\overline{\mathbf{A}}_{\mathrm{c}}$ exists, and noting that $\overline{\mathbf{B}}_c=\mathbf{B}_c^*$, Eq. 3 can be written in the general state space notation in continuous time:

$$\dot{\overline{\mathbf{x}}}(t)=\overline{\mathbf{A}}_c\overline{\mathbf{x}}(t)+\overline{\mathbf{B}}_c\overline{\mathbf{u}}(t),\ \overline{\mathbf{y}}(t)=\overline{\mathbf{C}}_c\mathbf{x}(t),\ \ \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where $\overline{\mathbf{C}}_c=\mathbf{C}_c=\mathbf{K}^T$. Semi-implicit Euler discretization of Eq. 5 by treating the state vector $\mathbf{x}(t)$ implicitly and the input vector $\mathbf{u}(t)$ explicitly results in

$$\overline{\mathbf{x}}(k+1)=[\mathbf{I}-\Delta t\overline{\mathbf{A}}_c]^{-1}[\overline{\mathbf{x}}(k)+\Delta t\overline{\mathbf{B}}_c\mathbf{u}(k)],\ \overline{\mathbf{y}}(k)=\overline{\mathbf{C}}_c\overline{\mathbf{x}}(k).\ \ \ldots (6)$$

Defining $\overline{\mathbf{A}}_d=[\mathbf{I}-\Delta t\overline{\mathbf{A}}_c]^{-1}$, $\overline{\mathbf{B}}_d=[\mathbf{I}-\Delta t\overline{\mathbf{A}}_c]^{-1}\Delta t\overline{\mathbf{B}}_c$, and $\overline{\mathbf{C}}_d=\overline{\mathbf{C}}_c$, allows us to write the general state space system in discrete time:

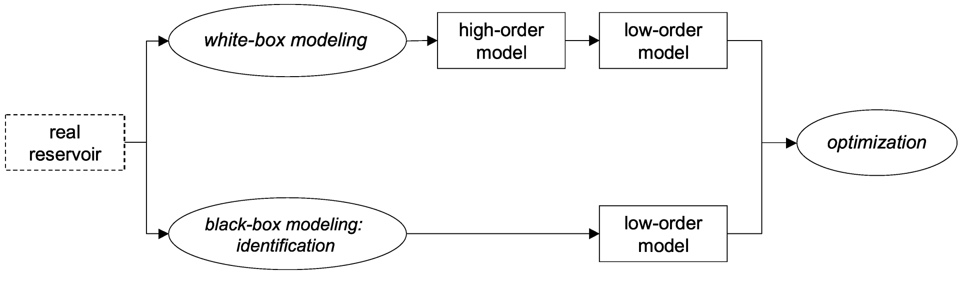

Fig. 2—Two basic approaches for deriving a low-order reservoir model that can be used for optimization.

TABLE 1—RESERVOIR AND FLUID PROPERTIES

Variable	Description	Value	Units
$\bar{m}, \bar{n}$	number of gridblocks in x and y direction	8	—
h	grid cell height	100	m
Δx, Δy	gridblock width	10	m
$S_{w,\text{ini}}$	initial water saturation	0.2	—
$p_{o,\text{ini}}$	initial pressure	$40*10^{6}$	Pa
ϕ	porosity	0.06–0.29	—
k	permeability	$1.7*10^{-14}$ – $6.7*10^{-13}$	m^2
c	compressibility	$1*10^{-9}$	Pa^{-1}
μ	oil and water viscosity	$1*10^{-3}$	Pa s
ρ_o	oil density	800	kg m^{-3}
ρ_w	water density	1000	kg m^{-3}

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}_d\bar{\mathbf{x}}(k) + \bar{\mathbf{B}}_d\mathbf{u}(k), \bar{\mathbf{y}}(k) = \bar{\mathbf{C}}_d\bar{\mathbf{x}}(k). \quad \text{(7)}$$

To simplify the notation, we will omit the overbar in the matrices in the remainder of the article.

Order Reduction Based on Modal Decomposition. Modal decomposition can be applied to a system defined in continuous time or in discrete time. We consider the continuous system in Eq. 5 of order n. We assume that the system matrix $\mathbf{A}_c$ has a complete set of n linearly independent eigenvectors, $\mathbf{m}_1$, $\mathbf{m}_2$, …, $\mathbf{m}_n$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. (In theory, the system may possess "degenerate" eigenvalues, in which case the diagonalization and the subsequent reduction can still be obtained in a block-Jordan form[13].) These eigenvectors and eigenvalues can be used to diagonalize the system into n uncoupled first-order equations. We start with writing the state vector $\mathbf{x}(t)$ as a linear combination of the eigenvectors[13]:

$$\mathbf{x}(t) = z_1(t)\mathbf{m}_1 + z_2(t)\mathbf{m}_2 + \cdots + z_n(t)\mathbf{m}_n = \mathbf{M}\mathbf{z}(t). \quad \text{(8)}$$

The elements of $\mathbf{z}(t)$ are called scalar mode participation factors. The columns of matrix $\mathbf{M}$ are the eigenvectors of $\mathbf{A}_c$. If we apply Eq. 8 to the system in Eq. 5, we can diagonalize the model:

$$\dot{\mathbf{z}}(t) = \Lambda\mathbf{z}(t) + \mathbf{M}^{-1}\mathbf{B}_c\mathbf{u}(t), \mathbf{y}(t) = \mathbf{C}_c\mathbf{M}\mathbf{z}(t), \quad \text{(9)}$$

where

$$\Lambda = \mathbf{M}^{-1}\mathbf{A}_c\mathbf{M} = diag\,(\lambda_1, \lambda_2, \cdots, \lambda_n), \quad \text{(10)}$$

is a diagonal matrix that contains the n eigenvalues of $\mathbf{A}_c$. If we define the combination $\mathbf{M}^{-1}\mathbf{B}_c\mathbf{u}(t)$ as the vector $\hat{\mathbf{u}}(t)$ of length n, we can write Eq. 9 as a set of n uncoupled first-order differential equations for the mode participation factors z_i:

$$\begin{aligned} \dot{z}_1(t) &= \lambda_1 z_1(t) + \hat{u}_1(t) \\ \dot{z}_2(t) &= \lambda_2 z_2(t) + \hat{u}_2(t) \quad \text{(11)} \\ &\vdots \\ z_n(t) &= \lambda_n z_n(t) + \hat{u}_n(t). \end{aligned}$$

In the specific case of flow through a porous medium, the eigenvalues are all real and either negative or close to zero. (In our case, one zero-valued eigenvalue occurs because all wells are operated on rate control, which implies that the system has a "constant pressure mode," analogous to the occurrence of "embedded statics" in mechanical system models.[13] If at least one of the wells would have been operated on a pressure constraint, all eigenvalues would have been nonzero.) The fact that none of the eigenvalues has an imaginary part indicates that the response is not oscillatory. The negative eigenvalues correspond to a damped response of the associated eigenvectors, and the more negative an eigenvalue, the faster the associated mode damps out. We are now able to reduce the state vector by truncation to length l based on the magnitude of the eigenvectors, $0 \leq |\lambda_1| \leq \ldots \leq |\lambda_l| << |\lambda_{l+1}| \leq \ldots \leq |\lambda_n|$:

$$\mathbf{x}(t) \approx z_1(t)\mathbf{m}_1 + z_2(t)\mathbf{m}_2 + \ldots + z_1(t)\mathbf{m}_l = \mathbf{M}_l\mathbf{z}_l, \quad \text{(12)}$$

where we assumed that the highly damped modes can be disregarded without a significant effect on the time response of our dynamic system.

Order Reduction Based on Balanced Realization. Modal decomposition is based on properties of the system matrix $\mathbf{A}_c$ only. Balanced realization also takes into account the effects of the input and output matrices $\mathbf{B}_c$ and $\mathbf{C}_c$ on the system response. It can be shown for the (asymptotically stable) continuous-time system (Eq. 5) that the minimum (2-norm) input energy L_{con} required to bring the system state from $\mathbf{x}(-\infty) = \mathbf{0}$ to $\mathbf{x}(0) = \mathbf{x}_0$ is given by

$$L_{\text{con}}(\mathbf{x}_0) = \frac{1}{2}\mathbf{x}_0^T\mathbf{W}_{\text{con}}^{-1}\mathbf{x}_0, \quad \text{(13)}$$

while the output 2-norm energy L_{obs} produced by the state $\mathbf{x}_0$ for $\mathbf{u=0}$ equals

$$L_{\text{obs}}(\mathbf{x}_0) = \frac{1}{2}\mathbf{x}_0^T\mathbf{W}_{\text{obs}}\mathbf{x}_0. \quad \text{(14)}$$

Here, the positive (semi-)definite matrices $\mathbf{W}_{\text{con}}$ and $\mathbf{W}_{\text{obs}}$ denote the controllability and observability Gramians of the system, defined as[14,15]:

$$\mathbf{W}_{\text{con}} \overset{\Delta}{=} \int_0^\infty e^{A_c t}\mathbf{B}_c\mathbf{B}_c^T e^{A_c^T t}dt, \mathbf{W}_{\text{obs}} \overset{\Delta}{=} \int_0^\infty e^{A_c^T t}\mathbf{C}_c^T\mathbf{C}e^{A_c t}dt. \quad \text{(15)}$$

Herein, $e^{\mathbf{A}_c t}\mathbf{B}_c \equiv \mathbf{x}_\delta(t)$ is the impulse state-response of system in Eq. 5, while $e^{\mathbf{A}_c^T t}\mathbf{C}_c^T \equiv \mathbf{x}'_\delta(t)$ is the impulse state-response of the dual of the system in Eq. 5 [i.e., of (omitting the overbars)]

$$\mathbf{x}'(t) = \mathbf{A}_c^T\mathbf{x}'(t) + \mathbf{C}^T\mathbf{u}'(t), \mathbf{y}'(t) = \mathbf{B}^T\mathbf{x}'(t). \quad \text{(16)}$$

Numerically, the Gramians can be found as solutions of the Lyapunov equations

$$\mathbf{A}_c\mathbf{W}_{\text{con}} + \mathbf{W}_{\text{con}}\mathbf{A}_c^T = -\mathbf{B}_c\mathbf{B}_c^T, \quad \text{(17)}$$

$$\mathbf{A}_c^T\mathbf{W}_{\text{obs}} + \mathbf{W}_{\text{obs}}\mathbf{A}_c = -\mathbf{C}_c^T\mathbf{C}_c. \quad \text{(18)}$$

Now a transformation, $\mathbf{x} = \mathbf{\Phi}\mathbf{z}$ is sought such that the state space representation is "balanced" in the sense that

$$\overline{\mathbf{W}}_{\text{con}} = \overline{\mathbf{W}}_{\text{obs}} = \sum = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n), \quad \text{(19)}$$

where $\overline{\mathbf{W}}_{\text{con}} = \mathbf{\Phi}^{-1}\mathbf{W}_{\text{con}}\mathbf{\Phi}^{-T}$ and $\overline{\mathbf{W}}_{\text{obs}} = \mathbf{\Phi}^T\mathbf{W}_{\text{obs}}\mathbf{\Phi}$. The values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$ are called the Hankel singular values of the system in Eq. 5 and are the square roots of the eigenvalues of $\mathbf{W}_{\text{con}}\mathbf{W}_{\text{obs}}$. These values are similarity invariants (i.e., not depending on the choice of coordinates). This means that $\mathbf{W}_{\text{con}}\mathbf{W}_{\text{obs}}$ and $\overline{\mathbf{W}}_{\text{con}}\overline{\mathbf{W}}_{\text{obs}}$ have the same eigenspectrum.

Following Moore,[14] the formal computation of $\mathbf{\Phi}$ involves an eigenvector/eigenvalue decomposition of $\mathbf{W}_{\text{con}}$ with which we can compute the intermediate transformation

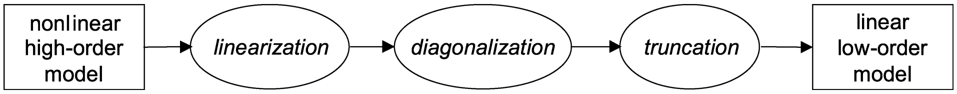

Fig. 3—Different steps in deriving a linear low-order model from a nonlinear high-order model.

$$\mathbf{\Phi}_1 = \mathbf{M}_{\text{con}} \sum\nolimits_{\text{con}}, \qquad (20)$$

where Σ_{con} is a diagonal matrix with the square roots of the eigenvalues ordered in nonincreasing magnitude and $\mathbf{M}_{\text{con}}$ a matrix with the corresponding eigenvalues as columns. Next, we compute

$$\tilde{\mathbf{W}}_{\text{obs}} = \mathbf{\Phi}_1^{-1} \mathbf{W}_{\text{obs}} \mathbf{\Phi}_1, \qquad (21)$$

and determine the corresponding eigenvalue and eigenvector matrices and $\tilde{\Sigma}^2_{\text{obs}}$ and $\tilde{\mathbf{M}}_{\text{obs}}$ with which we can determine

$$\tilde{\mathbf{\Phi}} = \tilde{\mathbf{M}}_{\text{obs}} \tilde{\Sigma}_{\text{obs}}^{-1/2}, \qquad (22)$$

and subsequently

$$\mathbf{\Phi} = \mathbf{\Phi}_1 \tilde{\mathbf{\Phi}}_2. \qquad (23)$$

After the initial publication of Moore, computationally more efficient algorithms have been developed; see, for instance, Laub *et al.*,[16] who describe the algorithm that is implemented in the MATLAB control toolbox. In the transformed coordinates $\mathbf{z}=\mathbf{\Phi}^{-1}\mathbf{x}$, the controllability and observability energy functions (Eqs. 13 and 14) can be written in the following form:

$$L_{\text{con}}(\mathbf{z}_0) = \frac{1}{2}\mathbf{z}_0^T \Sigma^{-1} \mathbf{z}_0 = \frac{1}{2}\sum_{i=1}^{n} \frac{1}{\sigma_i} z_{0,i}^2, \qquad (24)$$

$$L_{\text{obs}}(\mathbf{z}_0) = \frac{1}{2}\mathbf{z}_0^T \Sigma \mathbf{z}_0 = \frac{1}{2}\sum_{i=1}^{n} \sigma_i z_{0,i}^2. \qquad (25)$$

If σ_i is large, it takes very little energy to move the system in the associated direction; the value for the controllability energy function L_{con} is small. However, at the same time, the effect on the output of a movement in that direction is very large; the value of the observability energy function L_{obs} is large. Based on this energy consideration, we can understand the dominance of a singular value and reduction can be achieved. We can leave out the information associated with the smallest singular values. Because these small values of σ_i are positioned in the right lower corner of matrix $\mathbf{\Sigma}$, this can be done by truncation at the point where $\sigma_l >> \sigma_{l+1}$.

Order Reduction Based on Modal Decomposition and Balanced Realization. The state vector has a twofold character, because the pressure and the saturation are behaving very differently.[10,11] The pressure transient is governed by a parabolic equation and displays a diffusive fast-moving signal that rapidly decays because it is highly damped. The saturation on the other hand is governed by a diffusive/convective equation and forms a slow-moving signal that is only very lightly damped. In the absence of capillary pressures, there is no internal driving force, and the saturation changes are completely governed by convection. The difference in behavior is also recognizable in the eigenvalues of the system matrix $\mathbf{A}_c$ of Eq. 5. The heavily damped "pressure modes" are associated with negative eigenvalues very large in magnitude, whereas the almost undamped saturation modes are associated with the eigenvalues that are all close to zero. (For zero capillary pressure, they would become completely zero). The "pressure eigenvalues" show a clear difference in magnitude, and it is therefore possible to select a few dominant pressure modes. The magnitudes of the "saturation eigenvalues," on the other hand, are almost identical, which makes selecting only a few saturation modes disputable. To overcome this difficulty, we can combine reduction based on modal decomposition with reduction based on balanced realization as follows. First, we remove a certain number of pressure modes from the model described by Eq. 9. Subsequently, we find a balanced realization for this initial reduced model. Based on the Hankel singular values of the product of the observability and the controllability Gramian, we then further reduce the linear model (see **Fig. 4**).

Reduction of a High-Order Model to a Nonlinear Low-Order Model

POD is a technique developed in the fluid mechanics community to describe "coherent structures" that represent low-order dynamics of turbulent flow.[17,18] A transformation $\mathbf{\Phi}$ based on the covariance data of the nonlinear high-order model is found and used to reduce the nonlinear model. We apply a transformation, which can be defined in three steps (see **Fig. 5**). First, the state-sequences of the high-order model $\mathbf{x}(i)$, referred to as snapshots, are collected by means of simulation of the discrete-time model in Eq. 7 for κ timesteps in a data matrix $\mathbf{X}$:

$$\mathbf{X} = [\mathbf{x}(1) \quad \mathbf{x}(2) \ldots \mathbf{x}(\kappa)]. \qquad (26)$$

Typically, $\kappa << n$ and $\mathbf{X}$ is a tall $n \times \kappa$ matrix. If the state sequences are generated by sufficiently exciting input signals, they incorporate to a certain extent the nonlinear behavior of the high-order model. The vectors $\mathbf{x}(i)$ can be normalized by subtraction of the mean $\bar{\mathbf{x}} = (1/\kappa)\Sigma_{i=1}^{\kappa}\mathbf{x}(i)$:

$$\mathbf{X}' = [\mathbf{x}(1) - \bar{\mathbf{x}} \quad \mathbf{x}(2) - \bar{\mathbf{x}} \ldots \mathbf{x}(\kappa) - \bar{\mathbf{x}}], \qquad (27)$$

in which case the $n \times n$ matrix $\mathbf{R}' = \mathbf{X}'\mathbf{X}'^T/(\kappa-1)$ is the covariance matrix of state variables as captured by the snapshots. In our study, we worked directly with the $n \times n$ matrix $\mathbf{R}_n = \mathbf{X}\mathbf{X}^T$, which is sometimes also referred to as a covariance matrix. It can be shown that the eigenvectors of the eigenvalue problem

$$\mathbf{R}_n \mathbf{p} = \lambda \mathbf{p}, \qquad (28)$$

determine principal directions in the projected state space that best fit the collected state snapshots, measured in terms of the relative "energy" (i.e., mean square fluctuations, associated with particular directions in the state space). Because generally $\kappa << n$, the rank of $\mathbf{R}_n$ can be at most κ ($\kappa - 1$ if $\mathbf{X}'$ is used), and therefore it is sufficient to solve the much smaller eigenvalue problem

$$\mathbf{q}^T \mathbf{R}_\kappa = \lambda \mathbf{q}^T, \qquad (29)$$

where $\mathbf{R}_\kappa = \mathbf{X}^T\mathbf{X}$ is a κ x κ matrix. In analogy to Eq. 10, we can write

$$\mathbf{\Lambda}_n = \mathbf{P}^T \mathbf{X}\mathbf{X}^T \mathbf{P}. \qquad (30)$$

$$\mathbf{\Lambda}_\kappa = \mathbf{Q}\mathbf{X}^T\mathbf{X}\mathbf{Q}^T, \qquad (31)$$

where $\mathbf{\Lambda}_n$ and $\mathbf{\Lambda}_\kappa$ are $n \times n$ and $\kappa \times \kappa$ diagonal matrices, respectively, with ordered eigenvalues λ_i on the diagonal; $\mathbf{P}$ and $\mathbf{Q}$ are $n \times n$ and $\kappa \times \kappa$ orthogonal matrices, respectively, containing the eigenvectors $\mathbf{p}$ and $\mathbf{q}$ as columns and rows, respectively; and where we have used the transpose instead of the inverse because $\mathbf{R}_n$ and $\mathbf{R}_\kappa$ are symmetric and therefore the eigenvectors are orthogonal. The required (right) eigenvectors $\mathbf{p}$ can be obtained from the (left) eigenvectors $\mathbf{q}$ with the aid of the relationship

$$\mathbf{P} = \mathbf{X}\mathbf{Q}\mathbf{\Lambda}_\kappa^{-1/2}. \qquad (32)$$

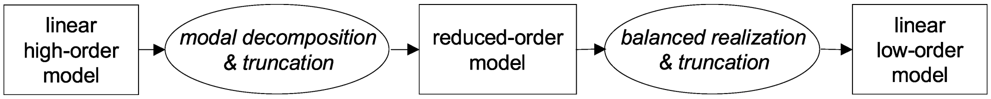
linear high-order model
modal decomposition & truncation
reduced-order model
balanced realization & truncation
linear low-order model

Fig. 4—Different steps in deriving a low-order model using modal decomposition and balanced realization.

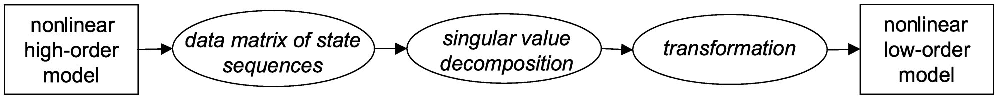
nonlinear high-order model
data matrix of state sequences
singular value decomposition
transformation
nonlinear low-order model

Fig. 5—Different steps in deriving a nonlinear low-order model from a nonlinear high-order model with POD.

This is equivalent to computing the eigenvectors $\mathbf{p}$ through the use of the singular value decomposition (SVD) of the data matrix[19]:

$$\mathbf{X} = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T, \qquad (33)$$

where the $n \times \kappa$ matrix $\mathbf{\Sigma}$ is given by

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_\kappa \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \qquad (34)$$

Here, $\sigma_1 \geq \ldots \geq \sigma_l >> \sigma_{l+1} \geq \ldots \geq \sigma_\kappa \geq 0$ are called the singular values of $\mathbf{X}$ and are the square roots of the eigenvalues λ_i, $i = 1, 2, \ldots, \kappa$. It is simply verified that Eqs. 30 and 31 can be obtained from Eq. 33 by working out the matrix products $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$. The required transformation $\mathbf{\Phi}$ is now defined as the first l columns of $\mathbf{P}$, where the cutoff point l depends on the magnitude of the singular values. An often followed approach is to choose an l for which

$$\sum_{i=1}^{l} \lambda_i \Big/ \sum_{i=1}^{\kappa} \lambda_i \geq \alpha, \qquad (35)$$

where α is close to one; typically $\alpha = 0.95$. Setting $\mathbf{x} \approx \mathbf{\Phi}\mathbf{z}$, the nonlinear low-order model is given by:

$$\mathbf{z}(k+1) = \mathbf{\Phi}^T \mathbf{f}[\mathbf{\Phi}\mathbf{z}(k), \mathbf{u}(k)].\ \mathbf{y}(k+1) = \mathbf{C}_d \mathbf{\Phi}\mathbf{z}(k). \qquad (36)$$

The reduced-order model Eq. 36 is "optimal" in the sense that this representation of the dynamic system, using basis functions obtained by POD, has a smaller mean square error than a representation by any other basis of the same dimension.[17] The numerical efficiency of POD will be discussed below.

Identification of a Linear Low-Order Model

We applied the deterministic subspace identification method (lower branch of Fig. 2) that we used earlier in a different context.[5] The idea of system identification is to extract a mathematical model directly from (measured) input/output data. We used the nonlinear high-order reservoir model to generate input/output data. Again, the inputs were the imposed flow rates in the well segments and the outputs were the pressures and saturations in the same well segments.

In subspace identification, the inputs and outputs are placed in so-called "past" and "future" block Hankel matrices, respectively $\mathbf{U}_p$, $\mathbf{U}_f$, $\mathbf{Y}_p$ and $\mathbf{Y}_f$. It is noted that the inputs need to be sufficiently "exciting" to create enough variety in the system response; for a strict definition of this property, see Ref. 20. The past inputs and outputs are combined in matrix $\mathbf{W}_p = (\mathbf{U}_p | \mathbf{Y}_p)^T$. From the SVD of the oblique (nonorthogonal) projection $\mathbf{O}$ of the future outputs $\mathbf{Y}_f$ on the future inputs $\mathbf{U}_f$ and the vectors of matrix $\mathbf{W}_p$ the extended observability matrix $\mathbf{\Gamma}_i = [(\tilde{\mathbf{C}}_d)^T\ (\tilde{\mathbf{C}}_d\tilde{\mathbf{A}}_d{}^T)\ldots(\tilde{\mathbf{C}}_d\tilde{\mathbf{A}}_d^{i-1})^T]^T$ and the future state sequences $\tilde{\mathbf{X}}_f$ can be found. The rank of the oblique projection equals the order of the identified system. The row space of $\mathbf{O}$ equals the row space of $\tilde{\mathbf{X}}_f$, while the column space of $\mathbf{O}$ equals the column space of $\mathbf{\Gamma}_i$. The state sequences can be used to find a least square solution for the system matrices $\tilde{\mathbf{A}}_d$, $\tilde{\mathbf{B}}_d$, $\tilde{\mathbf{C}}_d$, and $\tilde{\mathbf{D}}_d$, where we use a tilde to indicate that the matrices have been identified:

$$\begin{bmatrix} \tilde{\mathbf{X}}_{i+1} \\ \mathbf{Y}_{i|i} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_d & \tilde{\mathbf{B}}_d \\ \tilde{\mathbf{C}}_d & \tilde{\mathbf{D}}_d \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}_i \\ \mathbf{U}_{i|i} \end{bmatrix}. \qquad (37)$$

The process is schematically described in **Fig. 6.** The algorithm is explained in more detail in Ref. 20. This algorithm is only one of the possible subspace identification methods available.[21] The identified state space system is given by

$$\tilde{\mathbf{x}}(k+1) = \tilde{\mathbf{A}}_d\tilde{\mathbf{x}}(k) + \tilde{\mathbf{B}}_d\mathbf{u}(k),\ \mathbf{y}(k) = \tilde{\mathbf{C}}_d\tilde{\mathbf{x}}(k) + \tilde{\mathbf{D}}_d\mathbf{u}(k). \qquad (38)$$

The identified states $\tilde{\mathbf{x}}$ do not have a physical interpretation. The input and the output vectors $\mathbf{u}$ and $\mathbf{y}$, however, are variables with a physical meaning. The identified system can be used for optimization applications if a controller can be designed on these nonphysical states $\tilde{\mathbf{x}}$ and physical inputs $\mathbf{u}$ and outputs $\mathbf{y}$. To be able to compare responses from the identified low-order model with the high-order model, we took for the identified model $\tilde{\mathbf{x}}(0) = 0$ and added the initial output of the high-order model $\mathbf{y}(0) = \mathbf{y}_0$ to the output vector of the identified low-order model $\mathbf{y}(k)$.

Numerical Example 1

Simulation. We applied the five reduction techniques discussed above to the reservoir model (order 128) described by Eq. 1, Table 1, and **Figs.** 1 and **7.** Our first simulations were only meant for comparing the performance of the different methods. In contrast with "real" reservoir engineering, simulation time was short, and injected volumes, and thus saturation changes, were small. We choose for a short simulation time (10 days) so we could assume that the linearization was valid. We applied two different pseudo-randomly generated injection and production profiles (on/off well segment rates with a given amplitude but random duration) given in **Fig. 8** to our reservoir model. Total injected volumes were the same for both profiles and equaled the total produced volumes. After ten days, 0.04 pore volumes were injected at rates of 10^{-7} or 0 $\text{m}^3/(\text{m}^3/\text{s})$ with random duration for the first profile and at continuous rates of 0.32×10^{-7} $\text{m}^3/(\text{m}^3/\text{s})$ for the second profile. The second profile consisted of a constant injection rate to minimize the effect of changing inputs in neighboring injection cells on the output signals. The production rates were for both profiles 10^{-7} or 0 $\text{m}^3/(\text{m}^3/\text{s})$. Two types of low-order models (those based on subspace identification and POD) were derived directly from the nonlinear high-order model (i.e., without linearization). In these cases, the first profile was used for the derivation of the low-order models, while the second profile was used for verification.

Results. We assessed the quality of the reduced-order models by computing the average simulation error in the outputs for both sets of inputs according to[20]:

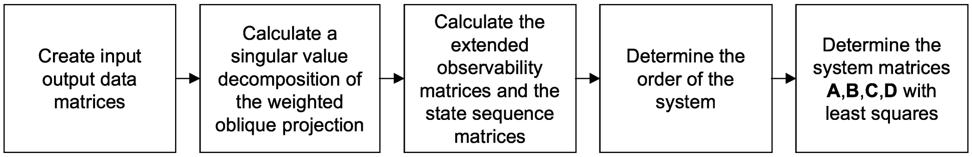
Create input output data matrices
Calculate a singular value decomposition of the weighted oblique projection
Calculate the extended observability matrices and the state sequence matrices
Determine the order of the system
Determine the system matrices A,B,C,D with least squares

Fig. 6—A short outline of the identification process developed in Ref. 5.

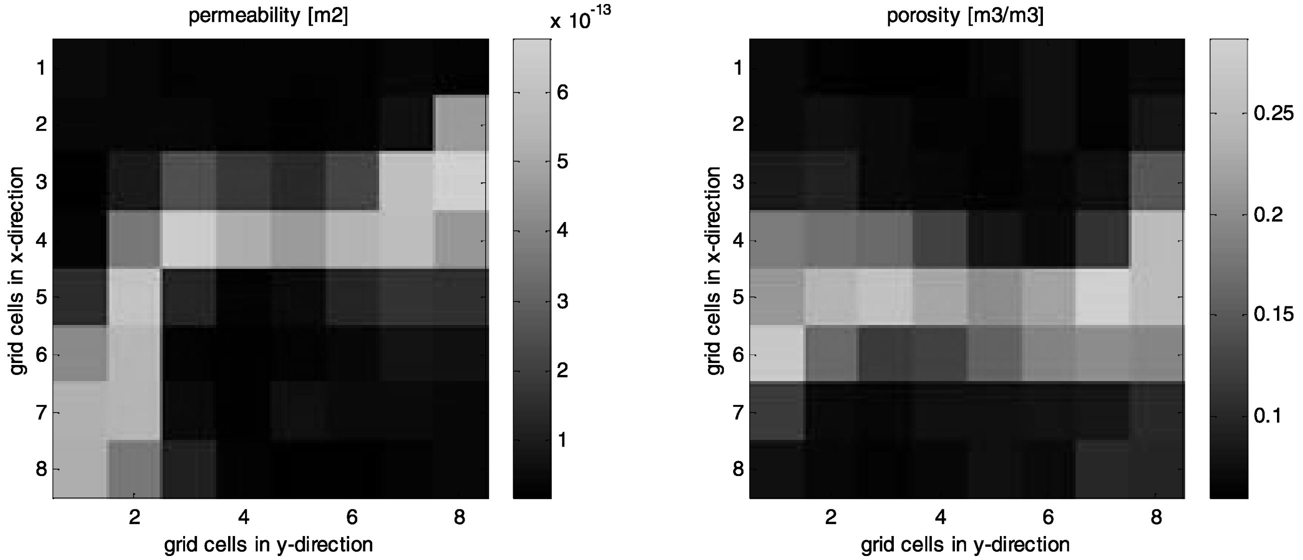

Fig. 7—Permeability (left) and porosity (right) heterogeneities used in our numerical examples.

$$\varepsilon = 100 \cdot \frac{1}{4\overline{m}} \sum_{i=1}^{4\overline{m}} \sqrt{\frac{\sum_{k=1}^{K} (y_{k,i} - \hat{y}_{k,i})^2}{\sum_{k=1}^{K} y_{k,i}^2}} [\%], \qquad (39)$$

where $y_{k,i}$ = the ith output (pressure or saturation) at timestep k of the nonlinear high-order model, $\hat{y}_{k,i}$ = the simulated output of the low-order model, K = the total number of timesteps, and $4\overline{m}$ is the total number of outputs. The simulation error (Eq. 39) is an averaged error over all timesteps and all gridblocks in which outputs are measured. It should be realized that in those grid cells where changes in the outputs are large, the error will be larger than the average error calculated.

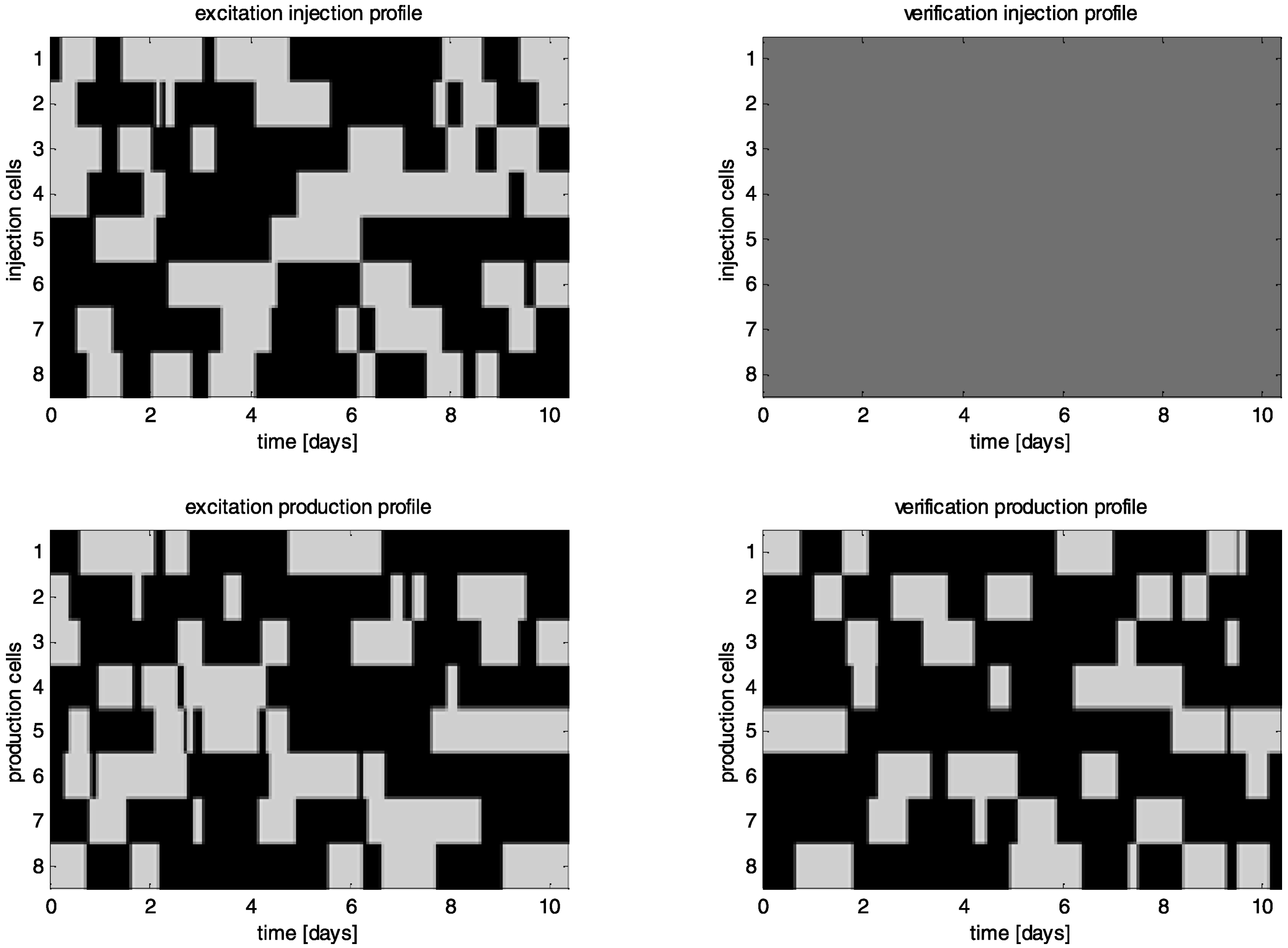

Fig. 8—Injection and production profiles for each smart well segment. Profile 1 (left) was used for excitation and derivation of the low-order models, while Profile 2 (right) was used for verification of the low-order models. Light colors indicate open valves; dark colors indicate closed valves.

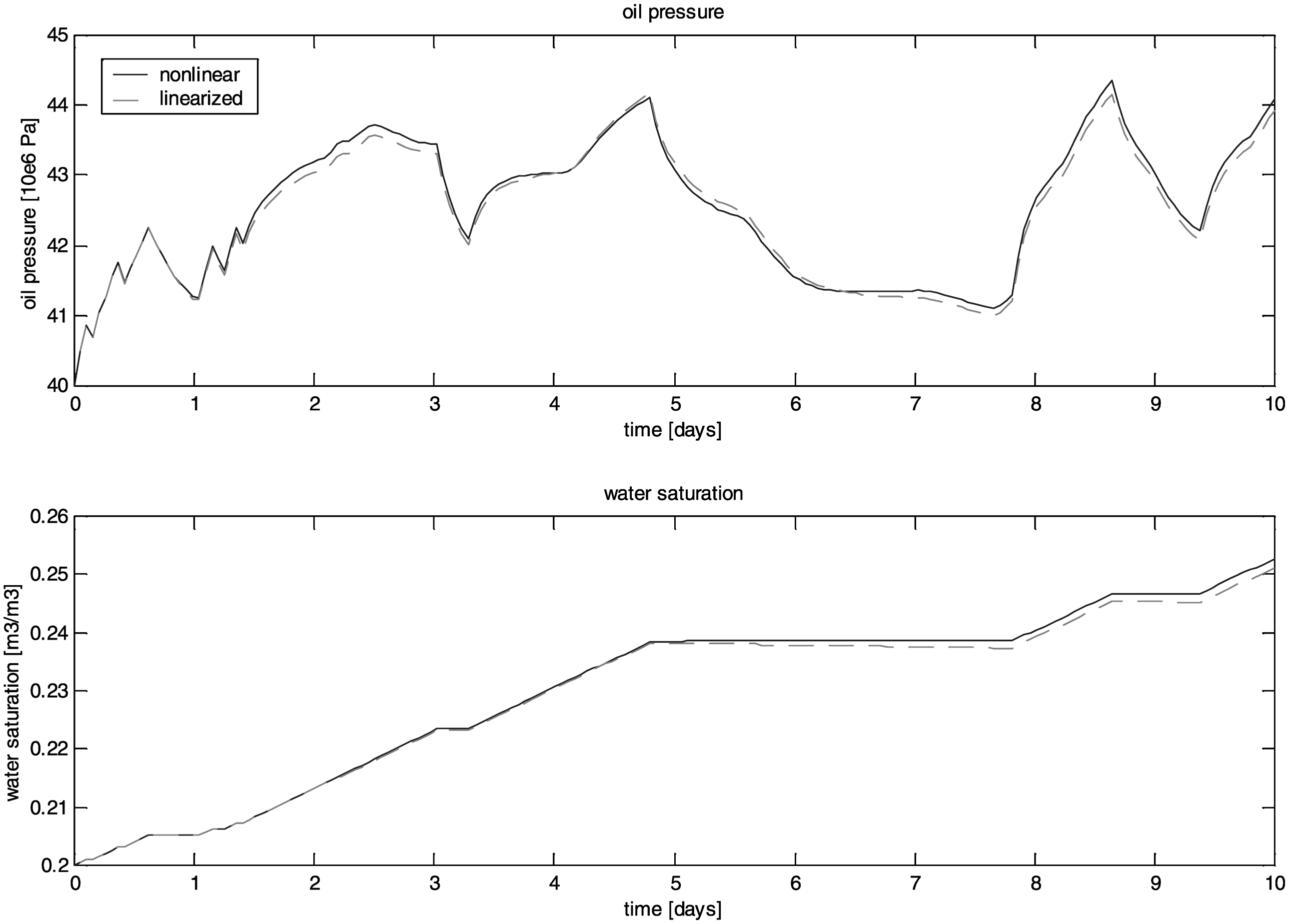

Fig. 9—Pressure and saturation responses in grid cell (1,1) for Profile 1 generated with the high-order nonlinear model and the linearized high-order model.

Figs. 9 and 10 show that the linearized high-order model is a good approximation of the nonlinear high-order model for ten days. The pressure and saturation responses in grid cell (1,1) reconstructed by a low-order model derived from the linearized high-order model through modal decomposition, balanced realization, and a combination of these techniques are shown in **Figs. 11 through 13** for the verification profile (Profile 2). The responses generated by the low-order models derived from the nonlinear high-order model (i.e., reduction based on POD and identification) are shown in **Figs. 14 and 15.** We choose for showing grid cell (1,1) because at the injection side, the changes in saturation are largest.

Tables 2 and 3 and Fig. 16 show that the low-order models based on the nonlinear high-order model result in significantly more accurate reconstruction of the pressure and the saturation responses than the low-order models based on the linearized high-order model. The tables also show that, as expected, the average simulation errors using the verification Profile 2 are somewhat higher than the average errors using Profile 1, which was used to derive the low-order models.

Discussion. Each eigenvector contains some information on pressure and saturation in each grid cell. However, it was found that by the initial reduction based on modal decomposition, only the pres-

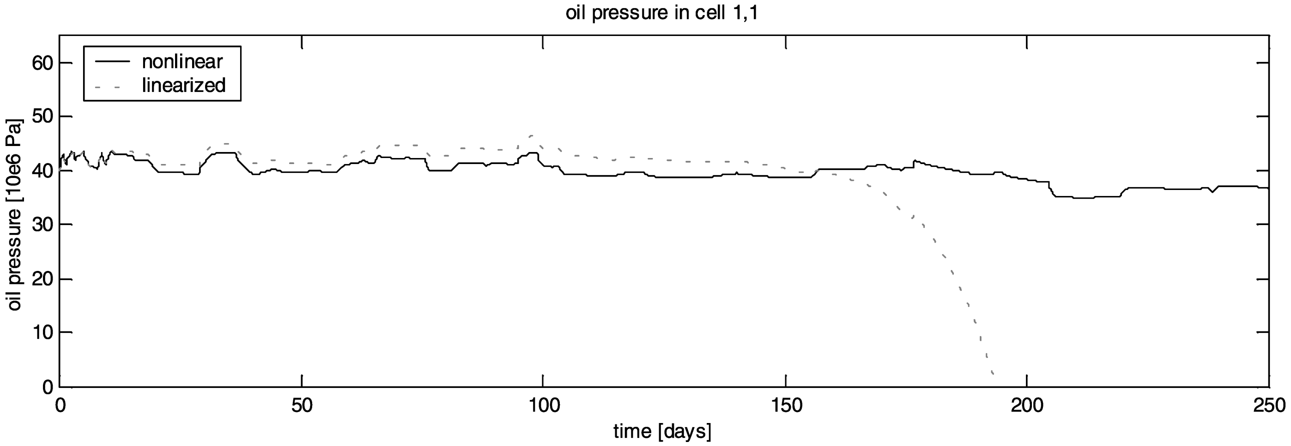

Fig. 10—Pressure response in grid cell (1,1) generated with the high-order nonlinear model and the linearized high-order model for a simulation time of 250 days using Profile 1 in the first 10 days. It can be seen that the linearization is only valid for a limited time interval.

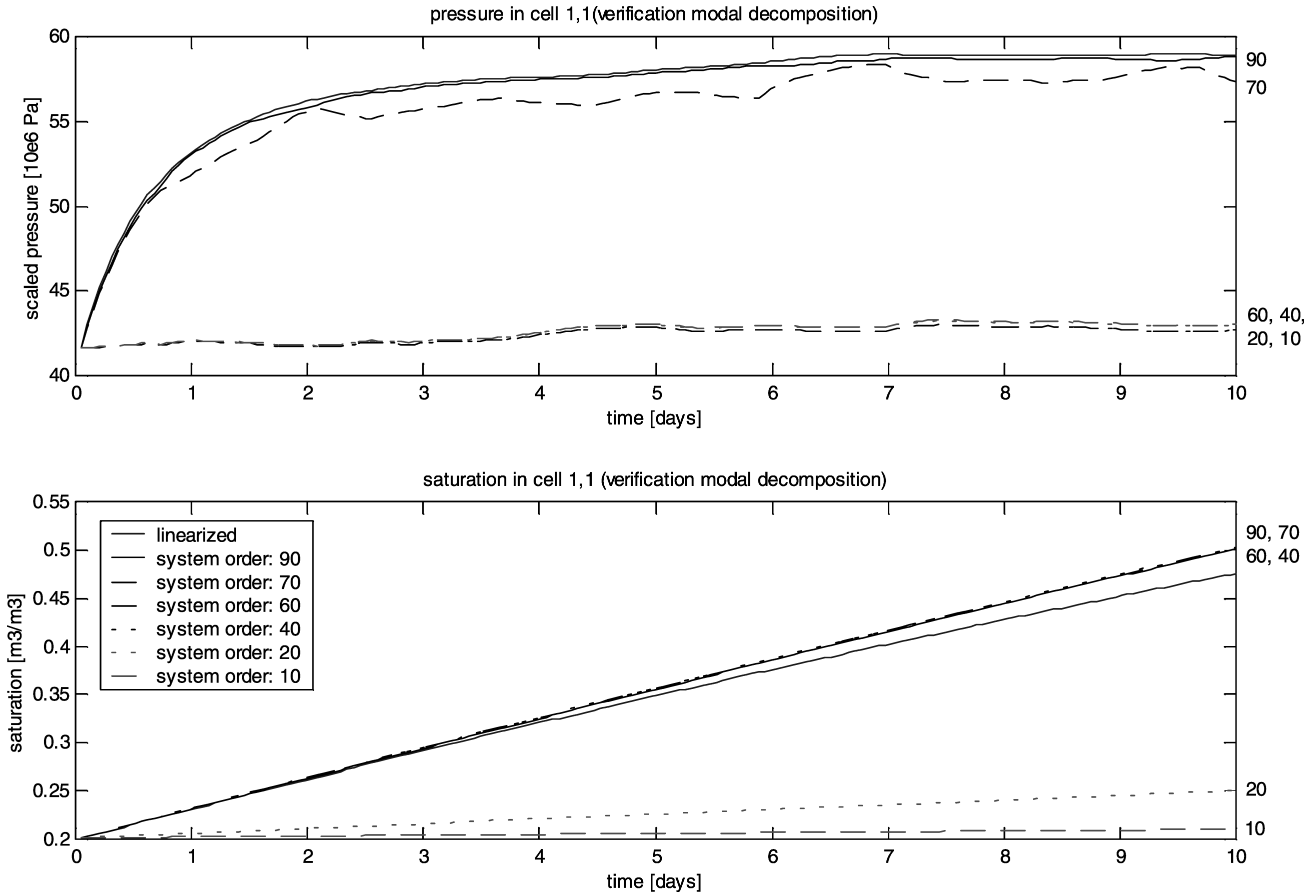

Fig. 11—Pressure and saturation responses in grid cell (1,1) for Profile 2 generated with the reduced-order model based on modal decomposition using 90, 70, 60, 40, 20, and 10 modes.

sure response is significantly affected (i.e., that the highest absolute eigenvalues are associated with the pressure modes of the model; see Fig 11). The near-zero eigenvalues, associated with the saturation modes, are very close to each other. The simulation error in the output of low-order models based on modal decomposition is smaller than the error in the output of models based on balanced realization as long as somewhat less than half of the modes are discarded (see Tables 2 and 3).

A feature of balanced realization is that the saturation and pressure modes are not separated into two groups as occurred for modal decomposition. This explains that for models of order 40, 20, and 10, the pressure is still reasonably reconstructed (see Fig. 11). For models of an order less than 40, the reconstructed saturation response becomes unacceptable. This might be explained by the fact that the saturation modes of the model are not sufficiently excited by the inputs applied. The inputs are only affecting the saturation in the grid cells close to the injectors.

In some cases, the results of a combination of modal decomposition and balanced realization can be more accurate. In our example, we first reduced the order of the model to 90 based on modal decomposition (so we truncated 38 pressure modes). Further reduction was achieved based on balanced realization. We see that the simulation errors in the output are smaller than for each of the methods separately (see Figs. 11, 12, and 13). In Fig. 13, we see that even for 20 modes, the saturation response can be reconstructed to some extent.

During subspace identification, the original order of the model (in this case 128) could not be retrieved. In the case of a sufficiently persistent input excitation, this would indicate that for the representation of the collected input/output data, a lower dimension state-space representation would suffice. In the different models tested, the retrievable order varied from 10 to 35 (in this numerical example, 24). In our example, the signal is probably not sufficiently exciting all modes of the reservoir model. The only modes identified are those associated with grid cells in which saturation changes. These grid cells are thus at the oil/water contact; the grid cells behind as well as in front of the oil/water contact will only show relatively small saturation changes or no changes at all.

A discussion of the POD results will be made after the next example. However, we already note that balanced realization and subspace identification are concerned with input/output behavior, whereas POD is only considering the relation between inputs **u** and states **x.** A potential way of extending the POD approach to also include the relation between states **x** and output **y** was presented in Ref. 9 through the use of so-called empirical Gramians.

Numerical Example 2

In this example, we will look at the performance of a nonlinear model derived with POD for a somewhat larger time intervals. We used the same reservoir model and heterogeneities as in Example 1. **Fig. 17** displays the relative permeability curves used. We applied Profile 3 to derive a nonlinear low-order model. The pattern of Profile 3 is similar to the pattern of Profile 1, but it is stretched over 1,500 days (see **Fig. 18**). We constructed a data matrix of snapshots taken over the entire time interval of 1,500 days, and we found a transformation **Φ** from the SVD of that data matrix. For verification we used Profile 4 with a simulation length of 3,000 days (see Fig. 18). Again, we assessed the quality of the models by computing the average simulation error given by Eq. 39.

After 3,000 days of simulation using the verification profile, the reservoir is almost water-saturated (see Fig. 17). **Table 4** shows that the calculated average simulation errors are surprisingly low. From **Figs. 19 through 22** we see that the pressure and saturation responses of the nonlinear low-order model both at the injection side [grid cell (1,1)] and at the production side [grid cell (1,8)] match very well with the responses generated by the original nonlinear high-order model. Even for a model of order 5, the

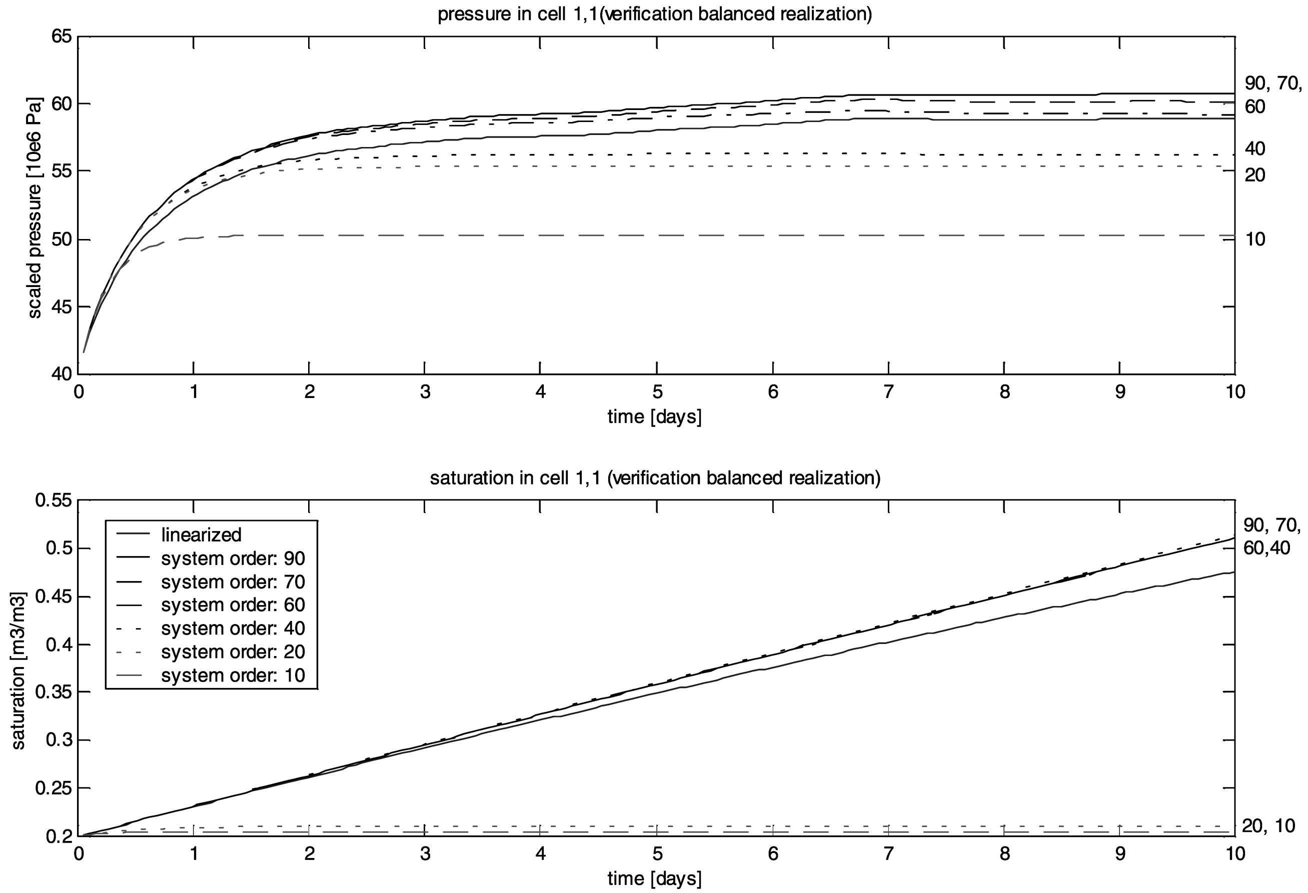

Fig. 12—Pressure and saturation responses in grid cell (1,1) for Profile 2 generated with the reduced-order model based on balanced realization using 90, 70, 60, 40, 20, and 10 modes.

reconstruction is reasonable. In Figs. 21 and 22, we see that after approximately 1,500 days the reconstructed responses start to increasingly diverge from the original responses. This can be explained by the fact that we only took snapshots into account up to that time. The response generated thereafter by the low-order model is a prediction and not a reconstruction. Although the accuracy of the prediction is lower than the accuracy of the reconstruction, the response of the low-order model is still close to the response of the high-order model.

Discussion

Low-Order Dynamics. The main purpose of this paper was to demonstrate that the dynamics of reservoir flow is governed by a system of differential equations of much lower order than the number of equations in conventional reservoir models. We demonstrated that it is indeed possible to describe the dynamics at any time in terms of a relatively small number of state variables. These new state variables have no direct physical meaning themselves but can sometimes, depending on the reduction method applied, be transformed to the "physical" state variables (pressures and saturations) of a high-order model. However, although we demonstrated that the essential dynamics can be represented in a low-order state space, we could not demonstrate that it is possible to directly and accurately compute future states in this low-order state space from a known state; in case of the linear methods, the future states will drift away from the true solution, whereas in the nonlinear method (POD) we first need to transform back the state variables to the high-order solution before we can compute the future state. The main potential benefit of our work, therefore, appears to be in the development of low-order control algorithms and history matching. Nevertheless, in the following, we will briefly assess the scope to use reduction methods for speeding up reservoir simulation.

Computational Efficiency—Modal and Balanced Reduction. At the request of one of the reviewers, we performed an operations order analysis of the modal reduction process. The efficiency of this linear method will strongly depend on choice of the solution routines for time integration and eigenvalue analysis. In particular, the eigenvalue routine should be able to compute only a limited number of eigenvalues and to recompute them with little effort for a slightly changed permeability. This points at the use of an iterative method, such as the subspace iteration method,[22] where the starting values could be formed by the converged solution from an earlier time. Furthermore, the previously discussed results indicate that modal reduction is much more effective for the pressure variables than for the saturation variables, which suggests use of the reduction process in an IMPES procedure to reduce the operations count of the computationally expensive implicit pressure computation. The "nonphysical" state variables obtained with modal reduction can be transformed back to the pressures of the original high-order model, which allows for the explicit computation of the saturations, as required in IMPES, in high-order state space. In the following, we assume that the sparseness of the system matrix can be expressed in terms of an effective half bandwidth *m,* and that the implicit pressure computation requires the solution of a system of *n* equations every timestep. If we use a Gauss elimination process that makes use of the banded matrix structure,[22] the number of operations for solving the system once is $\frac{1}{2}nm^2 + 2nm$. We do not take into account the operations required to reorder the equations to minimize the effective bandwidth. For large systems, it may be more efficient to use an iterative method, in particular to reduce storage requirements, but the operations count for the Gauss elimination provides a good estimate for our purpose. Computation of the input **Bu** is not taken into account because usually the number of inputs is much smaller than the number of states. Computation of the *l* lowest eigenvectors using subspace iteration

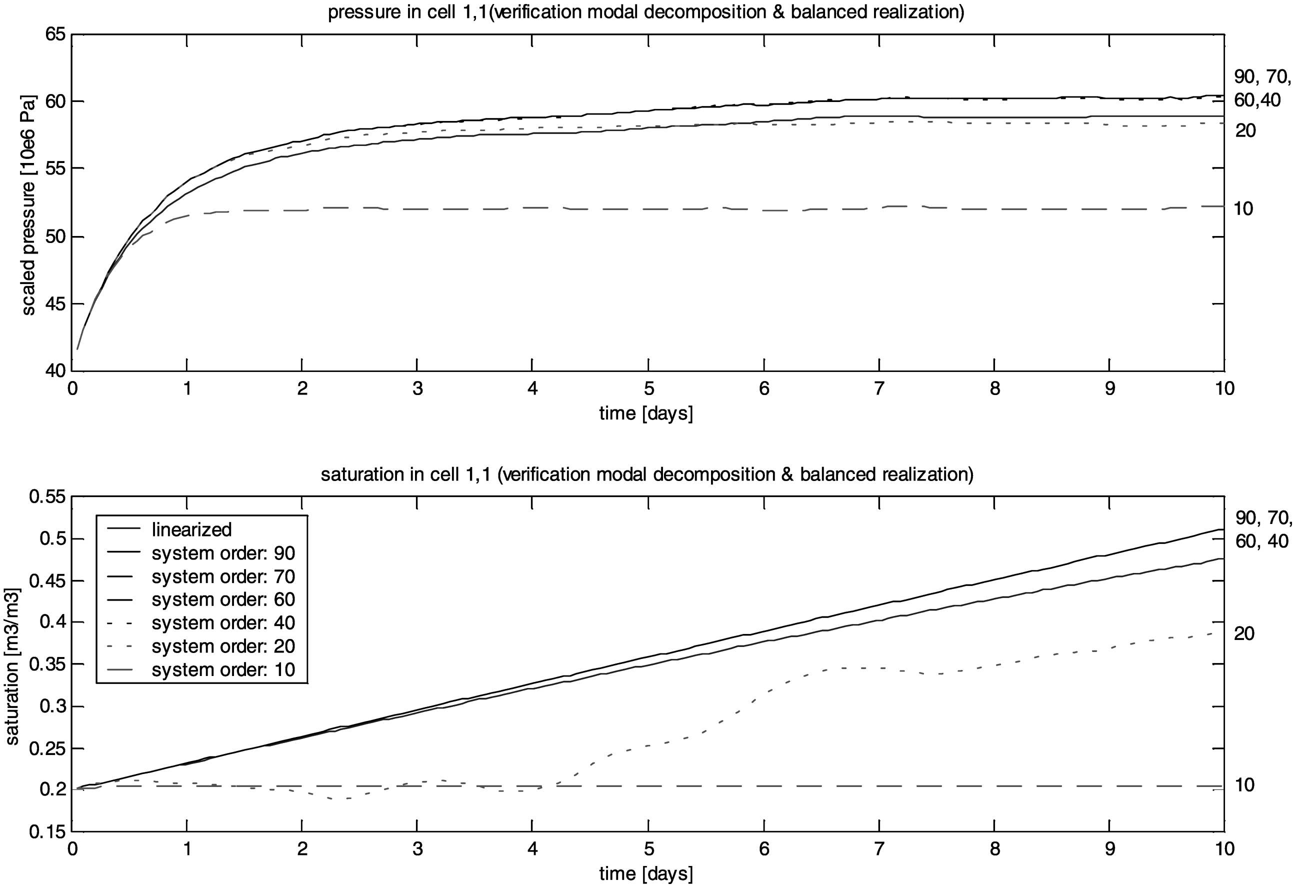

Fig. 13—Pressure and saturation responses in grid cell (1,1) for Profile 2 generated with the reduced-order model using 90, 70, 60, 40, 20, and 10 modes. The model was first reduced to 90 modes based on modal decomposition; further reduction was achieved based on balanced realization.

requires $nm^2 + nm(3+2l) + 5nl + 20n\,(l+8)\,(m+l+9\tfrac{1}{2})$ operations, assuming that 10 iterations are needed.[22] We assume that for recomputation of the eigenvectors (for instance, to account for a small change in the permeabilities caused by saturation changes), only two iterations are required, thus reducing the operations count with a factor of five. Propagation of $\mathbf{z}_l$ in time and back-transformation to $\mathbf{x}$ require $l+ln$ operations per timestep. **Table 5** illustrates the effect of the various parameters on the efficiency, expressed as the ratio between the number of operations N_{red} required for the reduced-order simulation and the number N_{conv} for conventional simulation of a theoretical example. The first column of numerical values represents a typical situation as encountered in dynamic analysis of large mechanical structures where modal reduction is often applied and clearly offers large computational benefits: $N_{\text{red}}/N_{\text{conv}} = 0.16$. The next columns give an indication of the effect of changing the various parameters. The total number of equations n has the least influence. Reducing the effective bandwidth m reduces the efficiency, where it is noted that $m = 4$ is a theoretical lower bound for 3D analysis, which can never be reached in reality. Reducing the number of timesteps K has a relatively small negative effect. However, a serious collapse of efficiency is experienced when either the number of eigenvalues l, or the number of eigenvalue updates K_{upd}, is increased. Further research is required to quantify the necessary values for these variables for realistic simulation models, but, based on the results displayed in Table 5, reduction of computing time with modal reduction does not look very promising. For balanced reduction, the situation is even worse because the similarity transformation $\mathbf{\Phi}^{-1}\mathbf{A}_c\mathbf{\Phi}$ ruins the sparse data structure of the system matrix $\mathbf{A}_c$ and therefore the reduced matrix is no longer banded.

Computational Efficiency—Identification and POD. A somewhat different situation occurs for model order reduction with subspace identification and POD. Subspace identification requires the simulation of a sufficiently "exciting" input signal in the high-order model in order to extract the low-order model. This procedure has to be repeated frequently because the low-order model is linear, and the resulting low-order models cannot be reused to simulate the same reservoir with, for example, different well operating conditions. Therefore, model reduction with subspace identification has no scope to increase computational efficiency.

For POD, it is required to first run a full-order simulation to obtain the necessary snapshots, and next perform an SVD before a reduced-order simulation can be performed. However, this may still be beneficial in situations where a model has to be run several times for the same well configuration, but with different flow rates or bottomhole pressures, as is often required for smart-well optimization studies.[1,2] In that case, the computational "overhead" of the POD, which is on the order of κ^3 operations for a full $\kappa \times \kappa$ matrix $\mathbf{R}_\kappa$, where $\kappa << n$, is shared by multiple runs of the low-order model. At first sight, it may appear as if the reduced-order model in Eq. 36 does not lead to a reduction in simulation time. If we compute $\mathbf{z}(k+1)$ explicitly, we even obtain a slight increase in simulation time, because for every timestep we have to perform two additional transformations (from $\mathbf{z}$ to $\mathbf{x}$ and back) because we need the original state vector $\mathbf{x}$ to compute the functions $\mathbf{f}$. However, if we consider implicit or semi-implicit computation of $\mathbf{z}(k+1)$, and if $\mathbf{f}(\mathbf{x})$ can be written in the form of matrices with coefficients that are functions of $\mathbf{x}$, a considerable efficiency gain may be achieved. Starting from the continuous-time form, the nonlinear equations can then be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c[\mathbf{x}(t)]\mathbf{x}(t) + \mathbf{B}_c[\mathbf{x}(t)]\mathbf{u}(t). \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (40)$$

Semi-implicit Euler discretization by treating the state and input vectors implicitly, but the matrix coefficients explicitly, can be written as

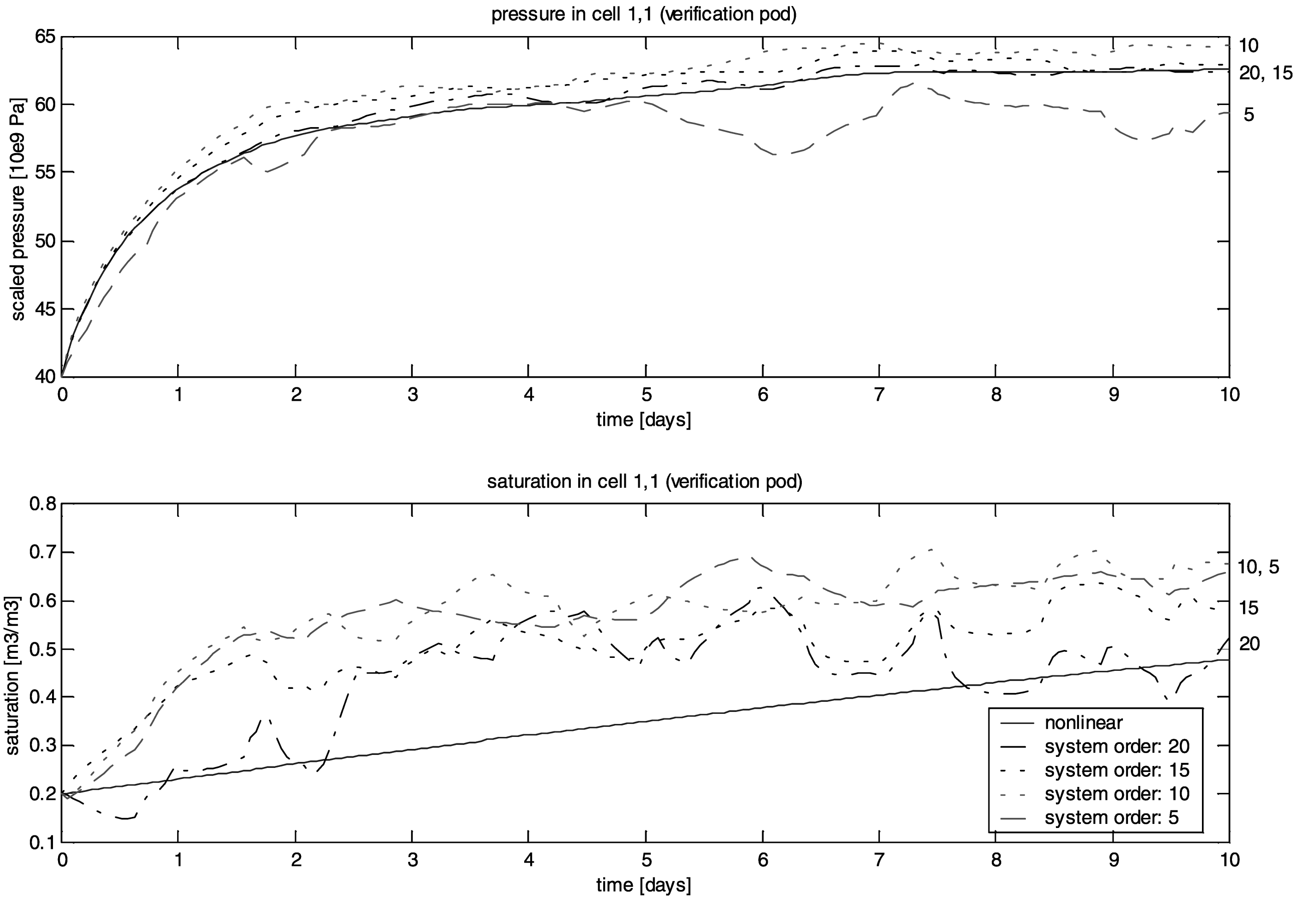

Fig. 14—Pressure and saturation responses in grid cell (1,1) for Profile 2 generated with the nonlinear reduced-order model based on POD using 20, 15, 10, and 5 modes.

$$\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{\Delta t} = \mathbf{A}_c[\mathbf{x}(k)]\mathbf{x}(k+1) + \mathbf{B}_c[\mathbf{x}(k)]\mathbf{u}(k+1), \; \ldots\ldots (41)$$

resulting in

$$\mathbf{x}(k+1) = \mathbf{A}_d[\mathbf{x}(k)]\mathbf{x}(k) + \mathbf{B}_d[\mathbf{x}(k)]\mathbf{u}(k+1), \; \ldots\ldots\ldots\ldots\ldots (42)$$

where the nonconstant matrices $\mathbf{A}_d$ and $\mathbf{B}_d$ have now been defined as

$$\mathbf{A}_d[\mathbf{x}(k)] = \{\mathbf{I} - \Delta t\mathbf{A}_c[\mathbf{x}(k)]\}^{-1},$$

$$\mathbf{B}_d[\mathbf{x}(k)] = \{\mathbf{I} - \Delta t\mathbf{A}_c[\mathbf{x}(k)]\}^{-1}\Delta t\mathbf{B}_c[\mathbf{x}(k)]. \; \ldots\ldots\ldots\ldots\ldots\ldots (43)$$

In an actual numerical implementation, the matrix inverses in the expressions in Eq. 43 are not computed because it is computationally more efficient to solve the equivalent system of linear equations

$$\{\mathbf{I} - \Delta t\mathbf{A}_c[\mathbf{x}(k)]\}\mathbf{x}(k+1) = \mathbf{x}(k) + \Delta t\mathbf{B}_c[\mathbf{x}(k)]\mathbf{u}(k+1), \; \ldots\ldots (44)$$

for the unknown state $\mathbf{x}(k{+}1)$. Applying the transformation $\mathbf{\Phi}$ to Eq. 44, we can now write

$$\{\mathbf{I} - \Delta t\mathbf{A}_c[\mathbf{\Phi z}(k)]\}\mathbf{\Phi z}(k+1) = \mathbf{\Phi z}(k) + \Delta t\mathbf{B}_c[\mathbf{\Phi z}(k)]\mathbf{u}(k+1), \quad (45)$$

from which follows

$$\underbrace{\mathbf{\Phi}^T\{\mathbf{I} - \Delta t\mathbf{A}_c[\mathbf{\Phi z}(k)]\}\mathbf{\Phi}}_{lxl}\mathbf{z}(k+1)$$

$$= \mathbf{z}(k) + \mathbf{\Phi}^T\Delta t\mathbf{B}_c[\mathbf{\Phi z}(k)]\mathbf{u}(k+1). \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (46)$$

The computational advantage of the model order reduction stems from the fact that the system of Eq. 46 involves only l unknowns, whereas the original system (Eq. 44) was expressed in terms of n unknowns. Although a transformation from $\mathbf{z}$ to $\mathbf{x}$ is still required every timestep to compute the coefficients of matrices $\mathbf{A}_c$ and $\mathbf{B}_c$, the simulation time reduction can become substantial, especially for large systems with $l{<<}n$. The benefit may be even stronger in fully implicit simulation, where multiple systems of equations need to be solved at every timestep. Further assessment of the efficiency of the nonlinear formulation (Eq. 36) is the subject of ongoing research.

Conclusions

1. High-order reservoir models can be reorganized and reduced to linear low-order models using modal decomposition, balanced realization, or system identification, or to nonlinear low-order models using POD.
2. The linear low-order models can sufficiently describe the reservoir behavior of a high-order model only for small time intervals during which the linearization is valid. For larger time intervals, the reduction process should be repeated.
3. The two-phase reservoir model has a "two-fold" character. The responses associated with the pressure modes are heavily damped, whereas the responses associated with the saturation modes are only lightly damped. This is reflected in the eigenvalues of the system matrix.
4. Modal decomposition only takes the system matrix into account. It enables a drastic reduction of the number of variables required to accurately describe the pressure response, but is ineffective in reducing the number of saturation variables.
5. Balanced realization takes both input-to-state and state-to-output into account. The behavior of the high-order model can be approximated by models of an order significantly less than half the original order (i.e., it also reduces the number of saturation modes).
6. A combination of modal decomposition and balanced realization leads in some cases to a further reduction where the dynamics of the high-order models are represented with the

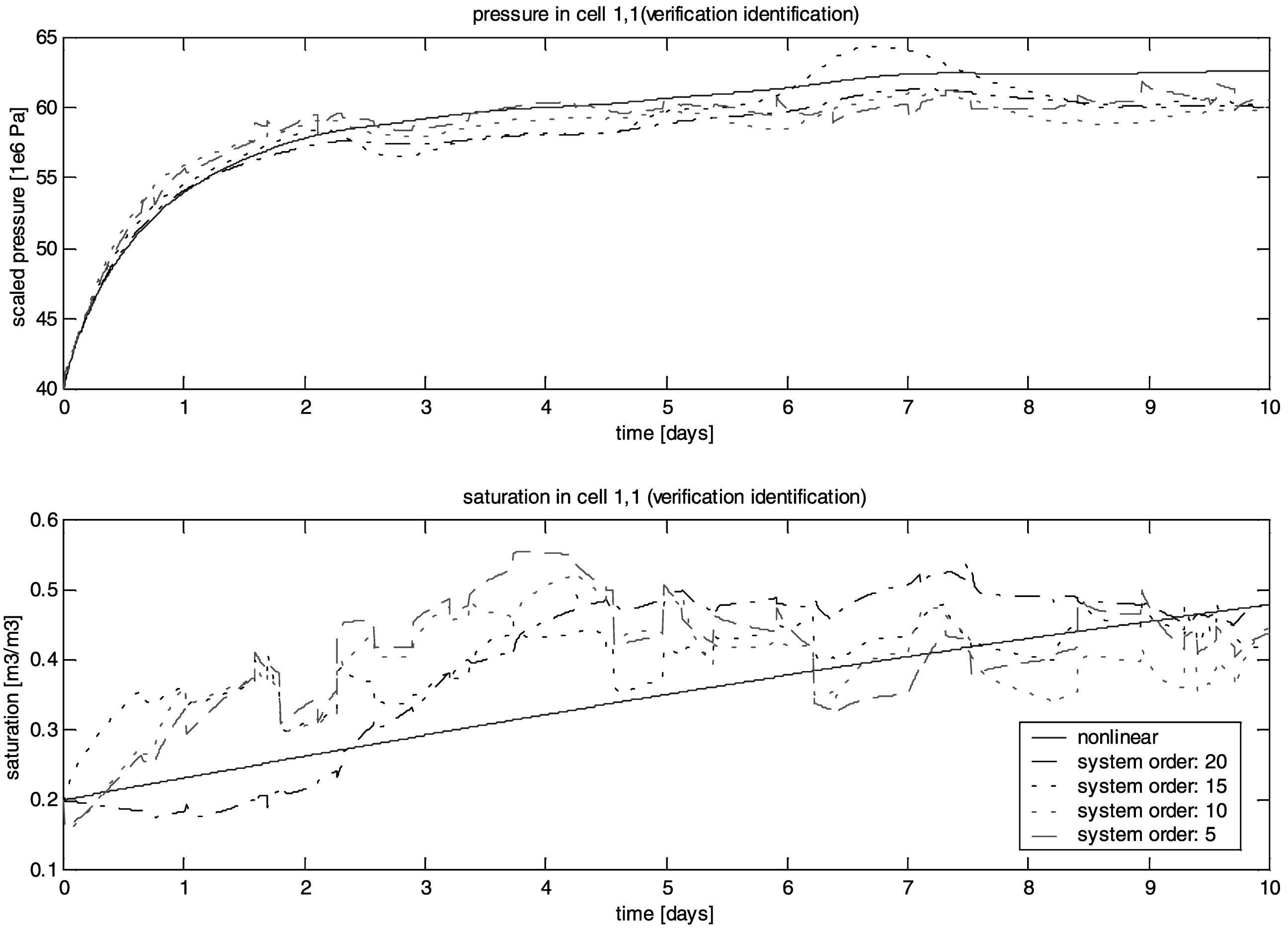

Fig. 15—Pressure and saturation responses in grid cell (1,1) for Profile 2 generated with the reduced-order model based on subspace identification using 20, 15, 10, and 5 modes.

same error, but with fewer modes than for the individual reduction methods.

7. Subspace identification of data generated by a nonlinear high-order model results in a linear low-order model that approximates the pressure and the saturation behavior of the high-order model with only a few modes but that remains valid only for short time periods.
8. POD of data generated by a high-order nonlinear model results in a nonlinear low-order model that approximates the behavior of the high-order model closely with a small number of modes. Furthermore, POD-based low-order models are generally valid for much longer time periods than the ones derived by linear reduction techniques.
9. Methods that result in linear low-order models are not very promising for speeding up reservoir simulation because the requirement to frequently repeat the reduction procedure creates a computational overhead that is not offset by the increase in simulation speed.
10. POD has the potential to improve computational efficiency in the case of multiple simulations of the same reservoir for dif-

TABLE 2—AVERAGE SIMULATION ERROR (%) FOR DIFFERENT LINEAR LOW-ORDER MODELS COMPARED TO THE HIGH-ORDER MODEL

Number of Modes	Modal Decomposition	Balanced Realization	Combined Modal/Bal.
	Profile 1		
90	0.14	0.26	0.29
70	0.38	0.35	0.29
60	2.98	0.45	0.29
40	3.61	1.48	0.32
20	3.63	2.86	2.59
10	3.63	3.41	2.93
	Profile 2		
90	1.26	2.52	2.65
70	3.38	2.61	2.65
60	18.9	2.93	2.65
40	20.50	8.59	2.60
20	23.34	16.90	14.39
10	24.47	21.57	16.98

TABLE 3—AVERAGE SIMULATION ERROR (%) FOR LINEAR (IDENTIFIED) AND NONLINEAR (POD) LOW-ORDER MODELS COMPARED TO THE HIGH-ORDER MODEL

No. of Modes	POD	Subspace Identification
	Profile 1	
15	0.34	0.24
10	0.46	0.36
5	0.79	0.71
	Profile 2	
15	8.05	5.87
10	11.25	5.51
5	11.78	6.31

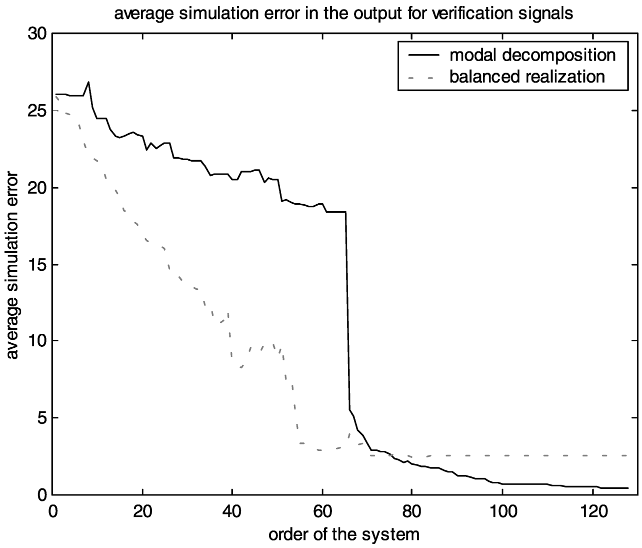

Fig. 16—Average simulation error (%) vs. the order of the reduced models both for reduction based on modal decomposition and for reduction based on balanced realization using Profile 2.

ferent well operating strategies, as needed, for example, in waterflooding optimization, but further research is required to quantify this scope.

11. The potential benefit of low-order models is therefore mainly in the development of low-order control algorithms, and in history matching, where the use of reduced models may form an alternative to classical regularization methods.

Nomenclature

$\mathbf{A}$ = system matrix
$\mathbf{B}$ = input matrix
c = compressibility, L t^2/m, 1/Pa
$\mathbf{C}$ = output matrix
$\mathbf{f}$ = nonlinear system function vector
h = gridblock height, L, m
i = counter
k = discrete timestep counter, or permeability, L^2, m^2
K = total number of timesteps
$\mathbf{K}$ = selection matrix
l = reduced system order
L = energy function
m = equivalent half bandwidth
$\overline{m}$ = number of gridblocks in x *direction*
$\mathbf{m}$ = eigenvector
$\mathbf{M}$ = matrix of eigenvectors
n = system order
$\overline{n}$ = number of gridblocks in y direction
N = number of operations
$\mathbf{O}$ = oblique projection
p = pressure, m/(L t^2), Pa
$\mathbf{p}$ = pressure vector
$\mathbf{P}$ = orthogonal matrix in SVD
q = flow rate, L^3/t, m^3/s
$\mathbf{q}$ = flow-rate vector
$\mathbf{Q}$ = orthogonal matrix in SVD
$\mathbf{R}$ = scaled covariance matrix
$\mathbf{s}$ = saturation vector
t = time
$\mathbf{T}$ = transmissibility matrix
$\mathbf{u}$ = input vector
$\mathbf{U}$ = Hankel matrix of input vectors
$\mathbf{V}$ = mass matrix
$\mathbf{W}$ = Gramian used in balanced realization, or matrix used in subspace identification
x = coordinate, L, m, or state
$\mathbf{x}$ = state vector
$\mathbf{X}$ = matrix of state vectors
y = coordinate, L, m, or output
$\mathbf{y}$ = output vector
$\mathbf{Y}$ = Hankel matrix of output vectors
z = transformed state variable
$\mathbf{z}$ = transformed state vector
α = cutoff factor for POD-based reduction
$\mathbf{\Gamma}$ = extended observability matrix
ε = simulation error
κ = number of snapshots in POD
λ = eigenvalue
$\mathbf{\Lambda}$ = diagonal matrix of eigenvalues
μ = viscosity, m/(L t), Pa s
ρ = density, m/L^3, kg/m^3

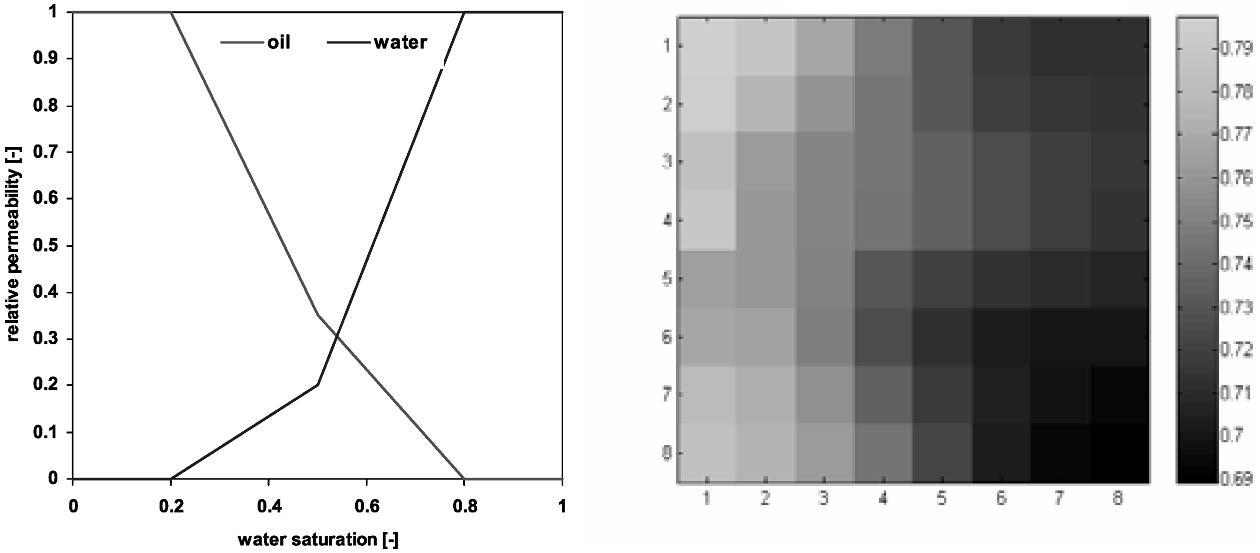

Fig. 17—Relative permeability curves for oil and water (left) and water saturation after 3,000 days of simulation using Profile 4 (right).

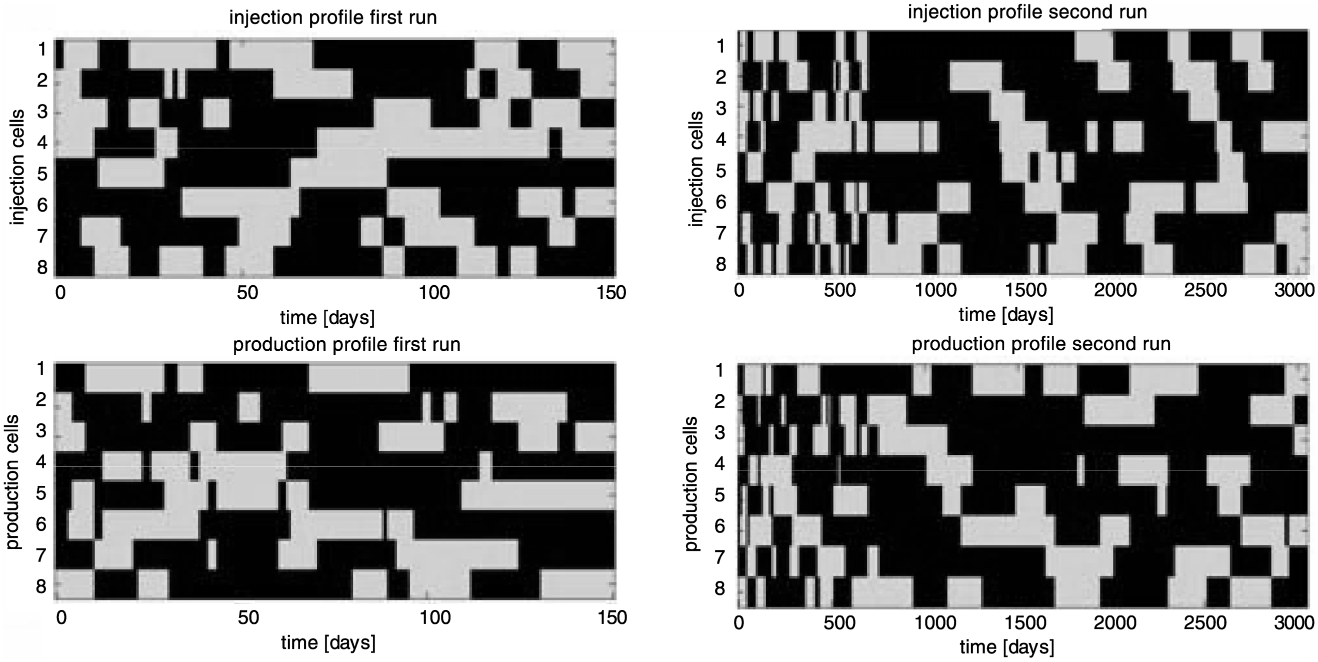

Fig. 18—Injection and production profiles for each smart well segment. Profile 3 (left) was used for excitation and derivation of the low-order models, while Profile 4 (right) was used for verification of the low-order models. Light colors indicate open valves; dark colors indicate closed valves.

TABLE 4—AVERAGE SIMULATION ERROR (%) FOR DIFFERENT NONLINEAR (POD) LOW-ORDER MODELS COMPARED TO THE HIGH-ORDER MODEL

No. of modes	POD, Profile 3	POD, Profile 4
15	0.95	5.36
10	1.56	5.88
5	2.18	7.27

σ = singular value
Σ = diagonal matrix of singular values
ϕ = porosity
$\mathbf{\Phi}$ = transformation matrix
$\mathbf{\Psi}$ = compressibility/porosity matrix

Subscripts

c = continuous
con = controllability

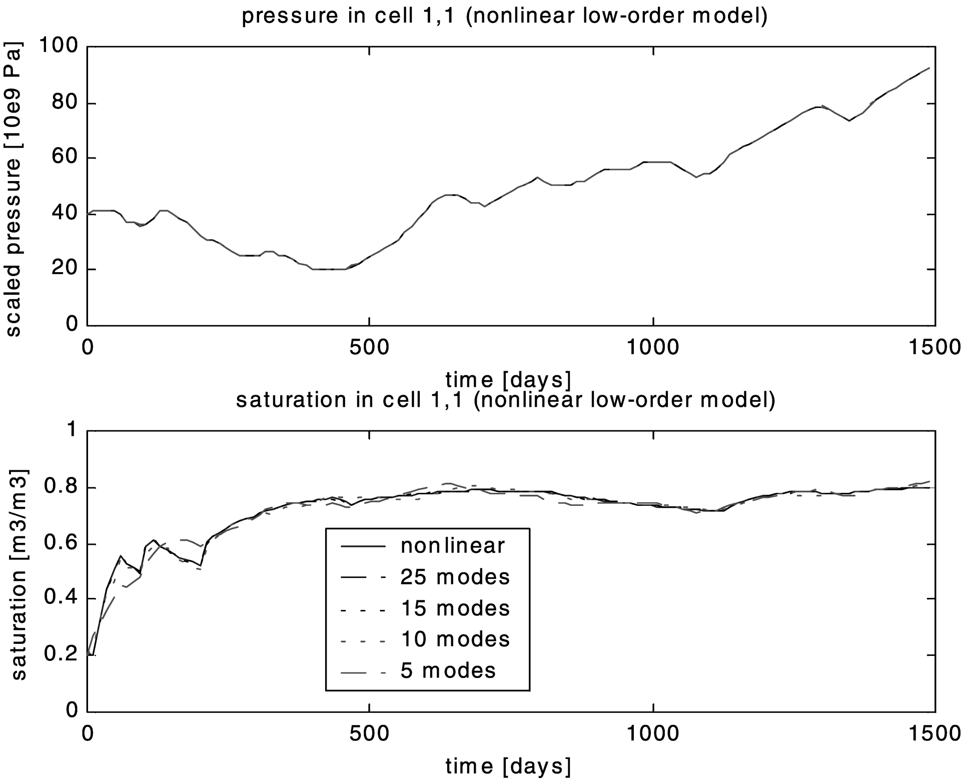

Fig. 19—Pressure and saturation responses in grid cell (1,1) for Profile 3 generated with the nonlinear reduced-order model based on POD using 25, 15, 10, and 5 modes.

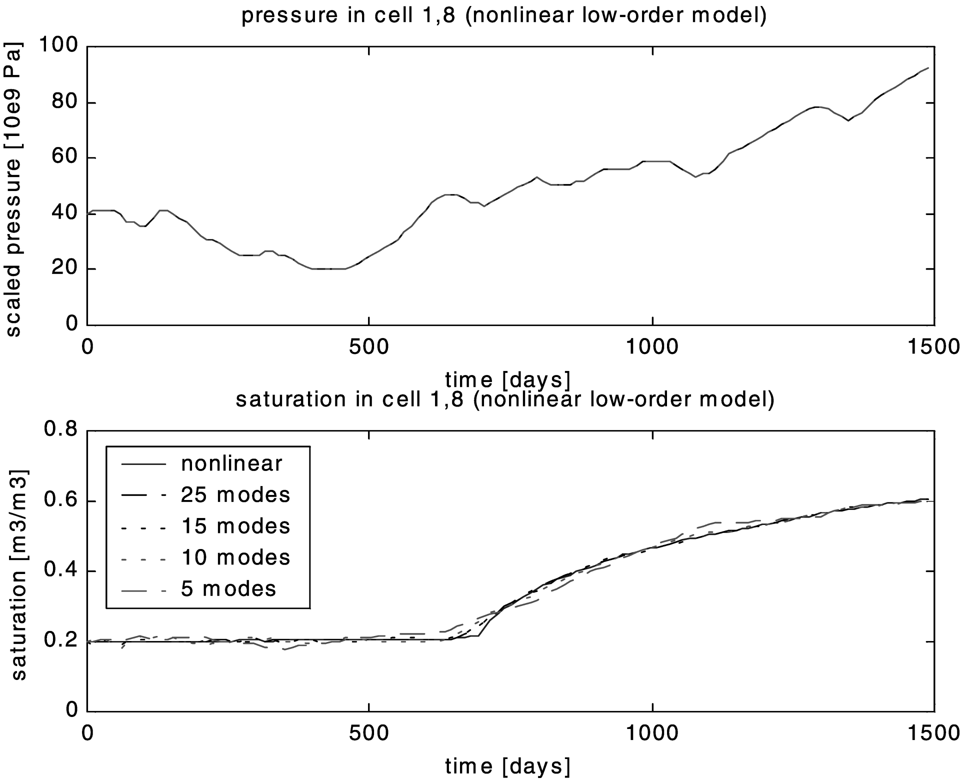

Fig. 20—Pressure and saturation responses in grid cell (1,8) for Profile 3 generated with the nonlinear reduced-order model based on POD using 25, 15, 10, and 5 modes.

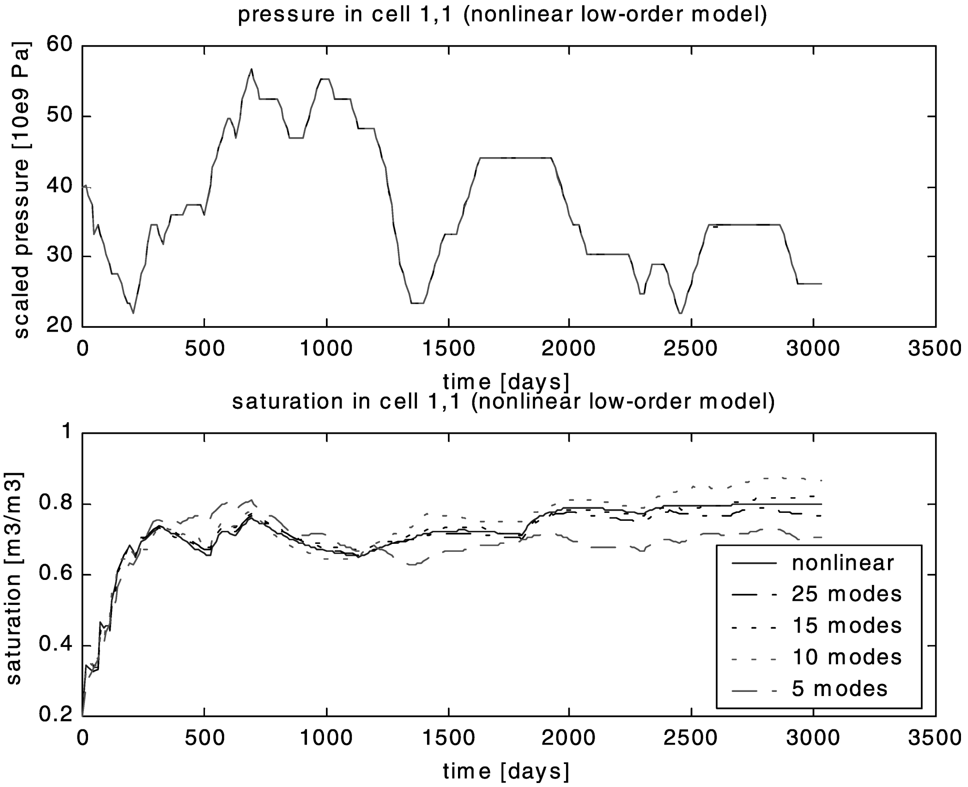

Fig. 21—Pressure and saturation responses in grid cell (1,1) for Profile 4 generated with the nonlinear reduced-order model based on POD using 25, 15, 10, and 5 modes.

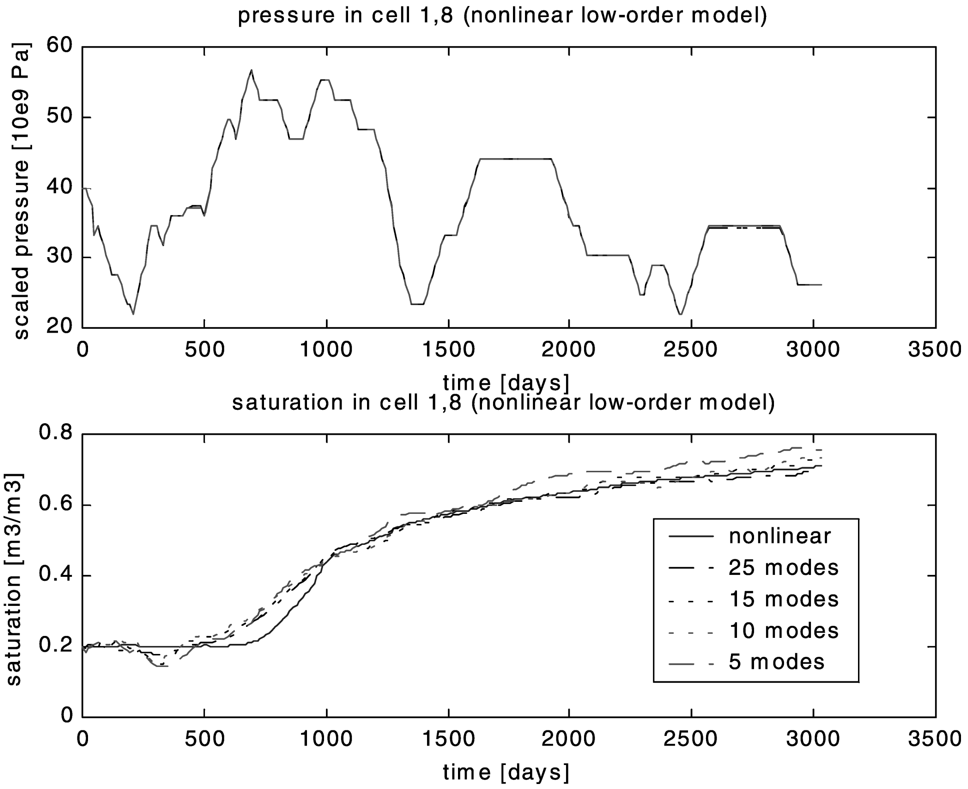

Fig. 22—Pressure and saturation responses in grid cell (1,8) for Profile 4 generated with the nonlinear reduced-order model based on POD using 25, 15, 10, and 5 modes.

conv =	conventional
d =	discrete
f =	future
ini =	initial
o =	oil
obs =	observability
p =	past
red =	reduced
t =	total
upd =	update
w =	water

References

1. Brouwer, D.R. and Jansen, J.D.: "Dynamic Optimization of Waterflooding With Smart Wells Using Optimal Control Theory," paper SPE 78278 presented at the 2002 SPE European Petroleum Conference, Aberdeen, 29–31 October.
2. Yeten, B., Durlofsky, L.J., and Aziz, K.: "Optimization of Smart Well Control," paper SPE 79031 presented at the SPE International Thermal Operations and Heavy Oil Symposium and International Horizontal Well Technology Conference, Calgary, 4–7 November.
3. Rowan, G. and Clegg, M.W.: "The Cybernetic Approach to Reservoir Engineering," paper SPE 727 presented at the 1963 SPE Annual Fall Meeting, Aberdeen, New Orleans, 6–9 October.
4. Chierici, G.L.: *Principles of Petroleum Reservoir Engineering,* Vol. 2, Springer, New York City (1995).
5. Markovinović, R., Geurtsen, E.L., and Jansen, J.D.: "Subspace Identification of Low-Order Reservoir Models," *Proc.,* 2002 International Conference on Comput. Meth. in Water Resources, Delft, The Netherlands (2002) **1,** 281–288.
6. Gharbi, R., Smaoui, N., and Peters, E.J.: "Prediction of unstable fluid displacements in porous media using the Karhunen-Loéve decomposition," *In Situ* (1997) **21,** No. 4, 331.
7. Gharbi, R.B., Quasem, F., and Smaoui, N.: "Characterizing Miscible Displacements in Heterogeneous Reservoirs Using the Karhunen-Loéve Decomposition," *Petroleum Science and Technology* (2003) **21,** Nos. 5 & 6, 747.
8. Vermeulen, P.T.M., Heemink, A.W., and Te Stroet, C.B.M.: "Reduction of large-scale numerical ground water flow models," *Proc.,* 2002 International Conference on Comput. Meth. in Water Resources, Delft, The Netherlands (2002) **1,** 397.
9. Markovinović, R. *et al.:* "Generation of Low-Order Reservoir Models Using POD, Empirical Gramians and Subspace Identification," *Proc.,* 2002 European Conference on the Mathematics of Oil Recovery (ECMOR VIII), Freiberg, Germany (2002) **E31,** 1–10.
10. Peaceman, D.W.: *Fundamentals of Reservoir Simulation*, Elsevier, New York City (1977).
11. Aziz, K. and Settari, A.: *Petroleum Reservoir Simulation*, Elsevier, New York City (1979).
12. http://www.mathworks.com.
13. Luenberger, D.G.: *Introduction to Dynamic Systems, Theory, Models & Applications,* Wiley, New York City (1979).
14. Moore, B.C.: "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Trans. Automat. Contr.* (1981) **AC-26,** No. 1, 17.
15. Skogestad, S. and Postlethwaite, I.: *Multivariable Feedback Control*, Wiley, New York City (1996).

TABLE 5—OPERATIONS COUNT TO ASSESS THE EFFICIENCY OF SIMULATION WITH MODAL REDUCTION

Variable	Magnitude					
n	10^4	10^6	10^4	10^4	10^4	10^4
m	10	10	5	10	10	10
l	10	10	10	100	10	10
K	10^4	10^4	10^4	10^4	10^3	10^4
K_{upd}	0	0	0	0	0	10^3
$N_{\text{red}}/N_{\text{conv}}$	0.16	0.16	0.48	1.80	0.30	3.30

16. Laub, A.J. *et al.:* “Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms,” *IEEE Trans. Automat. Contr*. (1987) **AC-32,** No. 2, 115.

17. Holmes, P., Lumley, J.L., and Berkooz, G.: *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge U. Press, Cambridge (1996).

18. Sirovich, L.: “Turbulence and the dynamics of coherent structures. Part 1: Coherent structures,” *Quart. of Appl. Math.* (1987) **55,** No. 3, 561.

19. Golub, G.H. and Van Loan, C.: *Matrix Computations,* third edition, John Hopkins U. Press, Baltimore, Maryland (1983).

20. Van Overschee, P. and De Moor, B.: *Subspace Identification For Linear Systems: Theory, Implementation, Applications,* Kluwer, Rotterdam, The Netherlands (1996).

21. Viberg, M.: “Subspace-Based Methods for the Identification of Linear Time-Invariant Systems,” *Automatica* (1995) **31,** No. 12, 1835.

22. Bathe, K.J.: *Finite Element Procedures,* Prentice Hall, New York City (1996).

Tim Heijn is a reservoir engineer for the Brent field at Shell E&P UK. He holds a degree in petroleum engineering from the Delft U. of Technology. **Renato Markovinović** is a PhD student in the Petroleum Engineering Dept. at the Delft U. of Technology. Previously, he was a researcher in the field of control engineering. He holds a degree in electrical engineering from the Delft U. of Technology. **Jan-Dirk Jansen** has a split assignment as Associate Professor of Petroleum Engineering at the Delft U. of Technology, and as Principal Research Engineer with the Exploratory Research group of Shell Intl. E&P. e-mail: j.d.jansen@citg.tudelft.nl. He has worked with Shell since 1986 in research and operations in The Netherlands, Norway, and Nigeria. Jansen holds MSc and PhD degrees from the Delft U. of Technology.