



**SPE 125557**

## **On State Constraints of Adjoint Optimization in Oil Reservoir Water-flooding**

E. Suwartadi, NTNU; S. Krogstad, SINTEF ICT; B. Foss, NTNU Norway

Copyright 2009, Society of Petroleum Engineers

This paper was prepared for presentation at the 2009 SPE Reservoir Characterization and Simulation Conference to be held in UAE, Abu Dhabi, 19-21 October 2009.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgement of SPE copyright.

### **Abstract**

Optimization on large scale reservoir models may be prohibitive in terms of run-time. The adjoint based method is an efficient way to compute gradients in such systems. The presence of state constraints however may be detrimental to computational efficiency. In this paper we explore the use of adjoints and the interior point method for reservoir problems, in particular the optimal production setting problem with state constraints. State constraints typically include limits on water saturation in grid blocks close to producer wells or limits on water production rate. Results are encouraging since computational efficiency is maintained in the two examples which are studied in this paper.

### **Introduction**

Many problems in reservoir engineering can be casted as optimization problems, among others optimal production settings, parameter/state estimation or history matching, and well placements and design. Optimal production settings have been addressed by Montleau et al. (2006), Sarma et al. (2008), parameter estimation or parameterizations in Wu (1999), Sarma et al. (2006), Rommelse (2009), and well placement in Zandvliet (2008), Vlemmix et al. (2009), Castineira et al. (2009). In order to perform optimization efficiently, gradient based methods are usually preferable. Since oilfields often are modeled as large scale systems consisting of millions of grid blocks, computing gradients can be a prohibitive task. Therefore, gradient computations using adjoints, which originates from optimal control theory, can be very useful. Adjoint based gradient computations require only two simulations regardless of the number of decision variables in the optimization problem. The conventional way to compute gradients, i.e. using finite differences, however, implies that the number of simulations increases proportionally to the number of decision variables.

The goal of the optimization problem may vary depending on the application. In the production optimization problem one may hope to increase some economical measure such as the net present value, recovery factor, cumulative oil production, and so on. In the history matching problem the objective is to minimize some misfit error between observations and model outputs. In this work we focus on the optimal production setting problem. The decision variables in these cases may include rates or bottom-hole pressures in producer and injector wells.

The closed-loop reservoir management paradigm introduced by Jansen et al. (2005) embeds the optimal production problem into a feedback structure as shown in **Fig. 1**. Data is used to update a model using some appropriate history matching technique and this model, possibly with an uncertainty description, is then used for decision support in an optimizer. Further, the feedback loop, i.e. the estimator and the optimizer, is traversed with regular intervals to exploit the information in newly received data as far as possible. Finally, it should be added that the estimator and optimizer blocks always include human judgement simply because it is impossible to capture all relevant issues in the mathematical formulation.

An optimization problem will usually include constraints. In production optimization there will be constraints on the control inputs and state variables. There will for instance always be capacity constraints on an injector well, i.e. on the control inputs of a model. More challenging, however, are state or output constraints. An example of this is an upper bound on water saturation in the wells of a two-phase (oil and water) oil reservoir.

In this paper we propose an adjoint based method to include state constraints in an adjoint formulation. The goal is to reduce the loss of computational efficiency in this case by applying an interior point optimization method. We continue the paper by first describing the optimization problem in the presence of state constraints. Then we explain existing methods which handle this problem and explain in detail our proposed method. Further, we present two rather simple case examples applying our method. In the last part we discuss each case separately before drawing our conclusions.

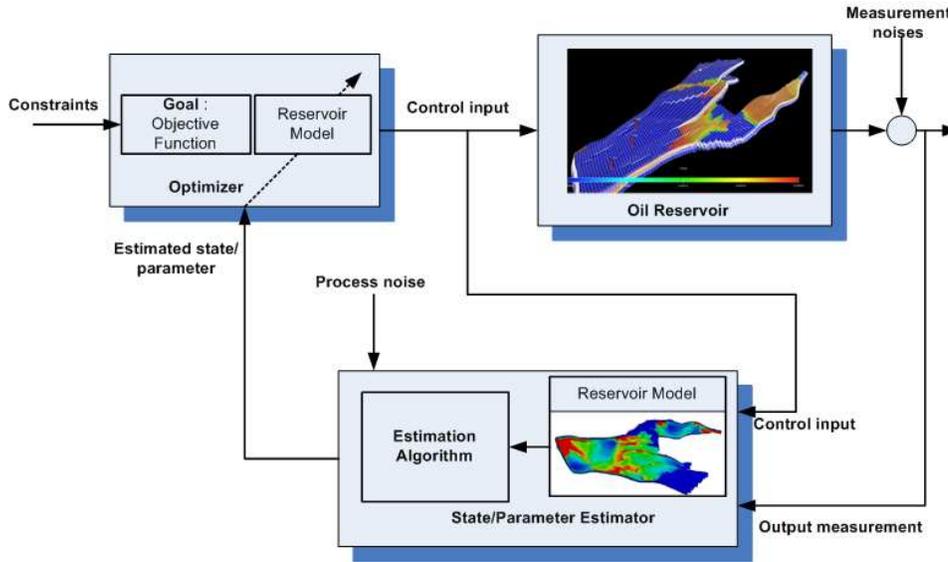


Fig. 1—Closed-loop reservoir management

### Problem Formulation

**The reservoir model** We focus on the optimal production case for two-phase (oil and water) reservoirs. Further, we assume immiscible and incompressible fluids and rocks, no gravity effects or capillary pressure, no-flow boundaries, and finally isothermal conditions. The corresponding state equations are referred to as the *pressure equation* and the *saturation equation*. The pressure equation is given by

$$\vec{v} = -\mathbf{K}\lambda(s)\nabla p, \quad \nabla \cdot \vec{v} = q, \quad (1)$$

where  $\vec{v}$  is the Darcy velocity,  $\mathbf{K}$  is the permeability tensor, and  $q$  is the volumetric source/sink term. Finally  $\lambda_t$  is the total mobility, which in this setting is the sum of the water and oil mobility functions,

$$\lambda_t(s) = \lambda_w(s) + \lambda_o(s) = k_{rw}(s)/\mu_w + k_{ro}(s)/\mu_o. \quad (2)$$

Here,  $k_{rw}$ ,  $k_{ro}$  and  $\mu_w$ ,  $\mu_o$  are the water and oil relative permeabilities and viscosities, respectively. Assuming no-flow boundaries means that the normal component of the Darcy velocity across boundaries is zero. The saturation equation is given by

$$\phi \frac{\partial s}{\partial t} + \nabla \cdot (f_w(s)\vec{v}) = q_w, \quad (3)$$

where  $\phi$  is the porosity and  $q_w$  the volumetric water source. Finally,  $f_w$  is the water fractional flow function  $f_w(s) = \lambda_w(s)/\lambda_t(s)$ . The nonlinear behavior of the above equations is mainly dictated by the shape of the relative permeability functions, which in this paper are taken to be quadratic.

The reservoir simulation model is typically given as a (non-matching) hexahedral grid, where each grid block is assigned reservoir properties such as permeability and porosity. For ease of exposition we here assume that the pressure equation Eq. 1 is discretized using a *two-point flux approximation* (TPFA) (see e.g., Aziz and Settari (1979)), although the MATLAB implementation employed in this work (Krogstad et al. 2009) is more general in the sense that a wider range of discretization schemes can be used. For a given saturation field, the TPFA assumes that the Darcy flux from one grid block  $i$  to its neighbor  $j$  is proportional to the pressure drop between the two blocks, i.e.,

$$v_{ij} = t_{ij}(s_i, s_j)(p_i - p_j). \quad (4)$$

In the above equation,  $t_{ij}$  is referred to as the transmissibility which in this formulation is taken to be dependent on the saturation in the two blocks (or rather  $\lambda(s_i)$  and  $\lambda(s_j)$ ). Wells are implemented using the Peaceman well model (Peaceman 1983)

$$q_i^w = -WI_i^w(s_i)(p^w - p_i). \quad (5)$$

Here  $q_i^w$  is the flow rate from well  $w$  into grid block  $i$  and  $p^w$  is the wellbore pressure (assumed to be constant since we neglect gravity and wellbore flow effects). Finally,  $WI_i^w(s_i)$  is the Peaceman well-index as a function of the grid block saturation  $s_i$ . The implementation uses a sequential splitting, that is, the pressure field at time-step  $n$  is calculated based on the saturation at time step  $n-1$ , and the saturation at time step  $n$  is calculated based on the pressure field at time step  $n$ . Let  $\mathbf{p}^n$  denote the vector

containing the grid block pressures and unknown wellbore pressures at time-step  $n$ . Similarly let  $\mathbf{s}^{n-1}$  denote the grid block saturations at time step  $n - 1$ . Enforcing volume balance, i.e., setting the sum of all out-fluxes (expressed as Eq. 3 and Eq. 4) from each block equal to the source, leads to a positive definite matrix  $\mathbf{A}(\mathbf{s}^{n-1})$  in a linear system equation

$$\mathbf{A}(\mathbf{s}^{n-1})\mathbf{p}^n = \mathbf{B}\mathbf{u}^n. \quad (6)$$

Here we have taken the right-hand-side as a function of a control input vector  $\mathbf{u}^n$  for time step  $n$ , which can either be well rates or well pressures. We note that the right-hand-side is linear in  $\mathbf{u}^n$  and refer to Eq. 6 as the discretized pressure equation.

We discretize the saturation equation Eq. 3 using a standard upstream weighted implicit finite volume method to form

$$\mathbf{s}^n = \mathbf{s}^{n-1} + \Delta t^n \mathbf{D}_{PV}^{-1} (\mathbf{M}(\mathbf{v}^n) f_w(\mathbf{s}^n) + \mathbf{q}(\mathbf{v}^n)_+). \quad (7)$$

Here  $\Delta t^n$  is the time step and  $\mathbf{D}_{PV}$  is the diagonal matrix containing the grid block pore volumes. The matrix  $\mathbf{M}(\mathbf{v}^n)$  is the sparse flux matrix based on the upstream weighted discretization scheme, and  $\mathbf{q}(\mathbf{v}^n)_+$  is the vector of positive sources (in this setting, water injection rates). We note that the matrix  $\mathbf{M}$  and vector  $\mathbf{q}$  are linear functions of  $\mathbf{v}^n$ , while  $\mathbf{v}^n = \mathbf{T}(\mathbf{s}^{n-1})\mathbf{p}^n$  where  $\mathbf{T}(\mathbf{s}^{n-1})$  is a matrix containing the transmissibilities and well indices based on  $\mathbf{s}^{n-1}$ . We refer to Eq. 7 as the discretized saturation equation.

**Optimization problem** As mentioned there are different goals which can be used as the key performance index in production optimization. Net present value (NPV), recovery factor and sweep efficiency are typical objective functions. In this work, we use the NPV with a fixed well configuration and apply this to a fixed optimization horizon. This horizon is divided into  $N$  equal time steps, i.e.  $n = 1, \dots, N$ . The discrete state equations Eqs. 6 and 7 can be rewritten in the following form

$$\begin{aligned} \mathbf{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) &= \begin{pmatrix} \mathbf{F}^1(\mathbf{p}^1, \mathbf{s}^0, \mathbf{s}^1, \mathbf{u}^1) \\ \vdots \\ \mathbf{F}^N(\mathbf{p}^N, \mathbf{s}^{N-1}, \mathbf{s}^N, \mathbf{u}^N) \end{pmatrix} \\ \mathbf{x}^{nT} &= (\mathbf{p}^{nT}, \mathbf{s}^{nT}), \quad n = 1, \dots, N, \\ \tilde{\mathbf{x}}^T &= (\mathbf{x}^{1T}, \dots, \mathbf{x}^{NT}), \\ \tilde{\mathbf{u}}^T &= (\mathbf{u}^{1T}, \dots, \mathbf{u}^{NT}). \end{aligned} \quad (8)$$

We assume  $\mathbf{x}^n \in \mathbb{R}^{n_x}$  and  $\mathbf{u}^n \in \mathbb{R}^{n_u}$ .

The control inputs are usually bounded and some of the states should be maintained around desired values. We use  $\mathbf{g}(\mathbf{u})$  and  $\mathbf{h}(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^{n_h}$  as functions to represent input- and state- constraints, respectively. The state constraint is a function of both the control inputs and states since the states are influenced by the control inputs. As an example any change in the bottom hole pressure (BHP) or well rate will change pressures and saturations accordingly. For the sake of brevity, we denote the state constraint by  $\mathbf{h}(\mathbf{x}, \mathbf{u})$  instead of  $\mathbf{h}(\mathbf{x}(\mathbf{u}), \mathbf{u})$ . Let  $J^n(\mathbf{x}^n, \mathbf{u}^n)$  be the objective function at time instance  $n$ . Hence, we formulate the optimal production setting with known initial states,  $\mathbf{x}^0$ , into the following optimization problem :

$$\begin{aligned} \max_{\mathbf{u} \in \mathcal{U}} \mathcal{J}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) &= \sum_{n=1}^N J^n(\mathbf{x}^n, \mathbf{u}^n) \\ \text{subject to :} \quad & \mathbf{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = 0, \\ & \mathbf{g}(\mathbf{u}^n) \leq 0, \quad n = 1, \dots, N, \\ & \mathbf{h}(\mathbf{x}^n, \mathbf{u}^n) \leq 0, \quad n = 1, \dots, N, \\ & \mathbf{x}(0) = \mathbf{x}^0. \end{aligned} \quad (9)$$

The reservoir model is posed as an implicit constraint while the input and state constraints are explicit. The state constraint may be imposed as a final time constraint meaning that we may constrain the state only at the final time  $N$ . Hence, we may have used  $\mathbf{h}^n$  to capture this case but leave this out in the interest of simple notation. Some references (Sarma et al. 2008), (Kraaijevanger et al. 2007) use an output constraint term instead of state constraints. This case is however captured with our formulation.

## Solution Methods

In terms of run time the optimization problem in Eq. 9, if it is solved using nonlinear programming (NLP) methods, is far more expensive than a simulation run of the reservoir model Eq. 8 itself. A bottleneck lies in the gradient calculations due to the large scale nature of the oil reservoir models. Explanation on nonlinear programming methods can be found in books like Nocedal and Wright (2006), Bonnans et al. (2006). There are two main frameworks, the line search and trust region methods. In this work we use both of the frameworks although the line search framework is dominantly found in the flow control community (Gunzburger 2003). This section will focus on how to obtain the gradient and how to handle the state constraints.

**Adjoint Gradient and Jacobian computation** There are three methods available for computing gradients; the finite difference method, sensitivities, and the adjoint method. Finite differences is the most commonly used method. For few decision variables or control inputs finite difference works fine with low computational cost while many control inputs add substantial computational cost. Since  $\mathbf{u}^n \in \mathbb{R}^{n_u}$  there will be  $N \cdot n_u$  decision variables on a  $N$  step horizon if a straightforward parametrization is applied. In this case finite differences require  $N \cdot n_u + 1$  simulation runs in the forward difference case and  $2 \cdot N \cdot n_u + 1$  simulation runs in the central difference case for only one gradient calculation.

Although finite difference methods are problematic they are trivial to code, easy to understand and is a standard tool to check gradient calculations from other methods. As a side remark it may be noted that in the meteorology community gradients are often checked by using Frechet derivative properties (Talagrand and Courtier 1987).

The sensitivity method can be used to compute gradients. The sensitivity equations are derived by using the chain rule, such that

$$\frac{D\mathcal{J}}{D\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} = \frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{x}}}\bigg|_{\tilde{\mathbf{u}}} \frac{d\tilde{\mathbf{x}}}{d\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} + \frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}}. \quad (10)$$

In the above equation,  $\frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{x}}}$  and  $\frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{u}}}$  can be determined directly by taking their derivatives in the objective function. Then  $\frac{d\tilde{\mathbf{x}}}{d\tilde{\mathbf{u}}}$  is again derived from using the chain rule in Eq. 8, namely :

$$\left(\frac{\partial\mathbf{F}}{\partial\tilde{\mathbf{x}}}\bigg|_{\tilde{\mathbf{u}}}\right) \frac{d\tilde{\mathbf{x}}}{d\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} = -\frac{\partial\mathbf{F}}{\partial\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}}. \quad (11)$$

Eq. 11 are the so-called sensitivity equations and the number of equations is equal to the dimension of the decision variables. In other words, we need to solve  $N$  equations and then plug the solutions into Eq. 10. This also applies to the gradient of the state constraints since this gradient is necessary in a NLP algorithm. One should note that the dimension of the sensitivities is  $n_u$  times the dimension of the states, hence Eq. 11 is usually a much larger system than Eq. 10. It may be added that by exploiting causality, computations can be halved since a control input change at time step  $n$  only influences the states on the interval  $n, \dots, N$ .

The idea of the adjoint method is based on the adjoint equation which originates from the Karush-Kuhn-Tucker (KKT) conditions:

$$\left(\frac{\partial\mathbf{F}}{\partial\tilde{\mathbf{x}}}\bigg|_{\tilde{\mathbf{u}}}\right)^T \boldsymbol{\lambda} = -\left(\frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{x}}}\bigg|_{\tilde{\mathbf{u}}}\right)^T, \quad (12)$$

In the above equation,  $\boldsymbol{\lambda}$  is the solution of Eq. 12 and usually called the Lagrange multipliers or adjoint variable. Using Eq. 10 we can eliminate  $\frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{x}}}$  to obtain

$$\frac{D\mathcal{J}}{D\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} = -\boldsymbol{\lambda}^T \left(\frac{\partial\mathbf{F}}{\partial\tilde{\mathbf{x}}}\bigg|_{\tilde{\mathbf{u}}}\right) \frac{d\tilde{\mathbf{x}}}{d\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} + \frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}}. \quad (13)$$

Substituting with Eq. 11, this yields

$$\frac{D\mathcal{J}}{D\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} = \boldsymbol{\lambda}^T \frac{d\mathbf{F}}{d\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}} + \frac{\partial\mathcal{J}}{\partial\tilde{\mathbf{u}}}\bigg|_{\tilde{\mathbf{u}}}. \quad (14)$$

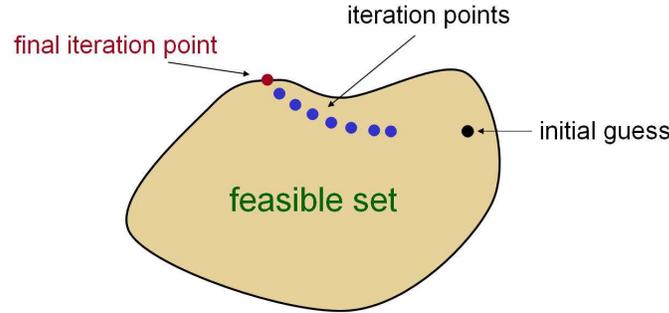
For clarity the  $\boldsymbol{\lambda}^T$  refers to  $(\boldsymbol{\lambda}^{1T}, \dots, \boldsymbol{\lambda}^{NT})$  with  $\boldsymbol{\lambda}^{nT} = (\boldsymbol{\lambda}_p^{nT}, \boldsymbol{\lambda}_s^{nT})$ . Here  $\boldsymbol{\lambda}_p^n$  and  $\boldsymbol{\lambda}_s^n$  correspond to  $\mathbf{p}^n$  and  $\mathbf{s}^n$ , respectively.

All derivative terms of Eq. 12 with respect to the states must vanish as required by necessary optimality condition. Furthermore, due the fact that all  $\boldsymbol{\lambda}^{N+1}$  are zero, we can obtain  $\boldsymbol{\lambda}^n$  by solving Eq. 12 in a backward manner, namely  $n = N, \dots, 1$ . We then use the obtained  $\boldsymbol{\lambda}$  in Eq. 14.

Similarly to the above, we have to derive adjoints for the state constraints  $\mathbf{h}$  since these constraints depend on the control input. This implies that it is necessary to compute adjoint variables for all state constraints. Remembering that  $\mathbf{h} \in \mathbb{R}^{n_h}$ , this gives rise to  $n_h$  adjoint equation sets which may reduce computational efficiency significantly. It is however possible to exploit the fact that the adjoints equal zero for all non-active constraints. Hence, some of the adjoint calculations are trivial.

In the optimal control society, there are attempts to find workarounds to the state constraint problem in adjoint calculations. Among others, Bloss et al. (1999) proposed the use of the Kreisselmeier-Steinhauser function to approximate the state constraint gradient. As mentioned above, we need both the gradient of the objective function and the state constraints. Bloss et.al suggested to use constraint aggregation to reduce the number of constraints as seen from the optimization algorithm, ultimately from  $n_h$  to 1. Further, he added the state constraints term as a penalty term in the objective function. Similarly, Sarma et al. (2008) used a smoothed penalty term in the objective function. These penalty terms may however result in infeasible solution, i.e., a solution where the state constraints are not satisfied. Mehra and Davis (1972) proposed the Generalized Reduced Gradient method and Kraaijevanger et al. (2007) approached the problem by using dependent variables and free/independent variables. However, it turns out that it is difficult to change the the set of dependent and free variables whenever the decision variables hit the boundary. Other approaches in the optimization community is to use numerical approximations of the state constraint gradient or Jacobian (see Griewank and Korzec (2005)).

**Optimization method** Our proposed method is to remove the state constraints and add them as a barrier term in the objective function. To this end, we are still able to utilize the efficiency of the adjoint method to compute the gradient. The difference between a barrier term and a penalty term is that the barrier function must use a strictly feasible initial guess (Boyd and Vandenberghe 2004). Hence, the control input must lie within the feasible set given by all the constraints in the optimization problem. For clarity the picture below displays the principle of the interior point method. Such methods always move within the interior of the feasible set. They may however converge to the boundary at the limit, i.e. at the point where the algorithm terminates.



**Fig. 2—Interior point method maintains feasibility of decision variables**

From now on the objective function is given by

$$\mathcal{J}_b(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \sum_{n=1}^N \left[ J^n(\mathbf{x}^n, \mathbf{u}^n) + \frac{1}{\mu} \sum_{i=1}^{n_h} \log(\mathbf{h}(\mathbf{x}^n, \mathbf{u}^n)) \right], \quad (15)$$

where  $\mu$  is a barrier parameter and the optimal solution is found when  $\mu \rightarrow \infty$ . By using Eq. 15 we compute one gradient since the objective incorporates the state constraints. The logarithmic function is used as the barrier function. The barrier function must be *self-concordant* and an alternative function is  $\frac{1}{(\mathbf{h}(\mathbf{x}^n, \mathbf{u}^n))^2}$  (Nesterov and Nemirovsky 1994).

To deal with the optimization problem Eq. 9, we choose the interior point method as our optimization algorithm. The main ingredients of an interior point method are: backtracking line search, a Newton method for equality constrained minimization, and a barrier function.

We need one more barrier function to tackle the input constraints. Let us again modify the objective function such that

$$\mathcal{J}_f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \mathcal{J}_b(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) + \frac{1}{t} \sum_{i=1}^N \sum_{i=1}^{n_g} \log(\mathbf{g}(\mathbf{u}^n)). \quad (16)$$

To find the minimum value of the modified objective function in Eq. 16, we perform the following procedure :

- Step 0 : Give strictly feasible  $\tilde{\mathbf{u}}$ , and choose  $\mu$
- Step 1 : Compute the gradient by using the adjoints
- Step 2 : Compute optimal solution  $\tilde{\mathbf{u}}$  of Eq. 16
- Step 3 : Check stopping criteria
- Step 4 : Repeat step 1 until the stopping criteria is met.

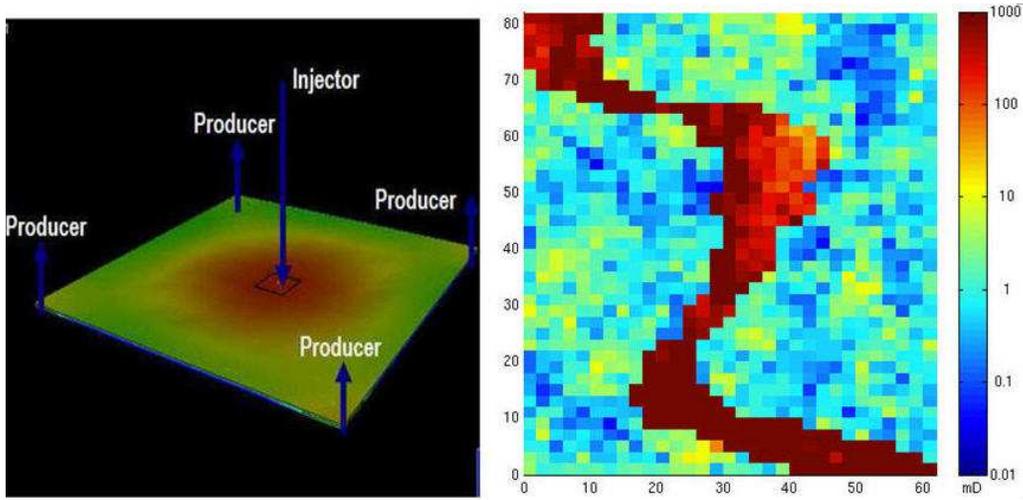
Steps 2, 3, 4 are done by an interior point method solver which in this paper is IPOPT (Wachter and Biegler 2006) or KNITRO (Byrd et al 2006). IPOPT is a line search based method while KNITRO is based on a trust region method. Both solvers have an adaptive way to parameterize the barrier parameter  $t$ . The barrier parameter,  $\mu$  is not parameterized in order to keep the number of iterations low. Hence, we fix  $\mu$  to some, possibly, large number. Ideally, one should add an outer iteration loop to adjust  $\mu$ .

## Numerical examples

We present two numerical case examples to clarify our proposed method. All examples use NPV as their objective function. Both cases have well rates as control inputs. The oil reservoirs in all cases are 2D incompressible systems with heterogenous permeability and homogenous porosity. Further, we assume perfect knowledge of all variables meaning that the estimator block in **Fig. 1** can be removed.

**Case example 1** This case originates from the SPE 10th Comparative Study (Christie and Blunt 2001) with five wells, one injector in the middle and four producers at the corners. We take layer 36 to make a 2D oil reservoir with fixed well locations. The grid consists of  $31 \times 41$  grid blocks, constant porosity 0.2, initial water saturation 0.2, connate water saturation and residual

oil saturation 0.2, and oil-to-water mobility ratio 5. The grid block dimension is  $2m \times 2m \times 2m$ . The permeability field is depicted in **Fig. 3**.



**Fig. 3**—The plots depict the well locations and permeability field in case no. 1

The state constraint is water saturation at the producer wells, denoted  $S_{prod,i}^N$  in the equation below. This should not exceed 0.65 at the end time  $N = 407$  days. In principle this constraint could have been enforced explicitly at all time steps. We do however know that by the construction of our problem, water saturation will always be a monotonic increasing function, hence enforcing the constraint at the end time will ensure compliance also at earlier time steps. The modified objective function corresponding to Eq. 15 is given by

$$\mathcal{J}_b(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \sum_{n=1}^N J^n(\mathbf{x}^n, \mathbf{u}^n) + \frac{1}{\mu} \sum_{i=1}^{n_h=4} \log(0.65 - S_{prod,i}^N),$$

where  $J^n$  is the NPV with a discount rate of zero. In order to really achieve maximum water saturation constraints, we use normalized prices with oil price is set equal to 1 per barrel, and water separation and water injection costs, are set to zero. With initial control input setting, the injection well will inject water up until 0.2 of the pore volume of the reservoir. This takes  $N$  days and this time interval is always divided into 5 control intervals. Hence, we have 25 control variables since there are 5 well rates for each control interval. Since we do not take feedback from measurements into account, this case is an open loop optimization problem. The injection rate must always equal the total producer rates, as it is an incompressible flow case. Moreover, we set an upper bound of  $100 \text{ m}^3/\text{day}$  on the rates.

**Case example 2** In this case, we constrain the total water production rate. This case originally comes from Brouwer (2004) where a reservoir has 45 horizontal injector well segments on one side and 45 horizontal producer well segments on the other side. The control inputs are well rates at the 90 well segments. We parameterize the control inputs into 5 time intervals which results in 450 control variables. We fix each control interval to 120 days, hence in total we have a 600 day simulation horizon. The permeability of the reservoir is shown in **Fig. 4** and it has constant porosity 0.2, initial water saturation 0.2, connate water saturation and residual oil saturation 0.2, unit oil-to-water mobility. The reservoir consists of  $45 \times 45$  grid blocks with grid block dimension is  $10m \times 10m \times 10m$ . The total water production rate constraint is set to  $500 \text{ m}^3/\text{day}$  with an oil price \$125 per barrel, water production cost is \$10 per barrel and zero water injection cost. Having the total water production constraint, then a modified objective function for this case is

$$\mathcal{J}_b(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \sum_{n=1}^N J^n(\mathbf{x}^n, \mathbf{u}^n) + \frac{1}{\mu} \sum_{n=1}^N \log\left(500 - \sum_{i=1}^{45} q_{prod,i}^n f_w(S_{prod,i}^n)\right).$$

The well rates are bounded by  $0 \text{ m}^3/\text{day}$  and  $200 \text{ m}^3/\text{day}$ .

## Results

**Case 1** We run the optimization with IPOPT with an initial injector rate of  $10 \text{ m}^3/\text{day}$  and  $2.5 \text{ m}^3/\text{day}$  for the producers on the whole interval. These initial values yield a water saturation below 0.65 at each producer well at the end time. For this particular setting the NPV is 3228.

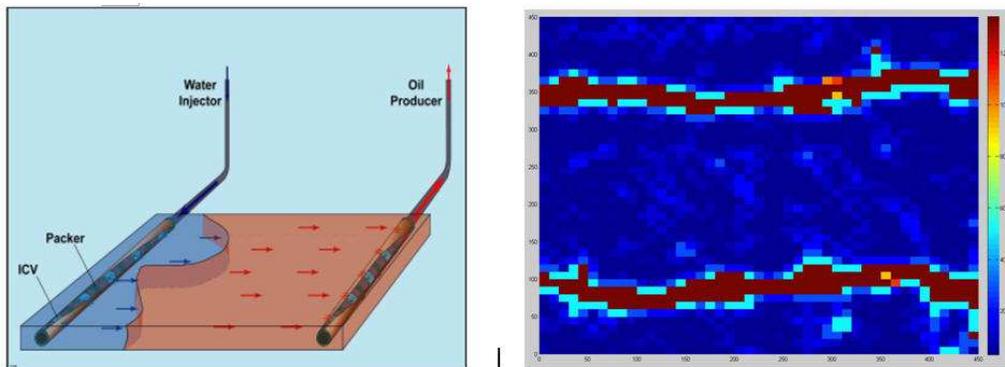


Fig. 4—The well settings and permeability field case no. 2

The injection rate must equal the total producer rates. This gives rise to an equality constraint with Jacobian

$$\begin{pmatrix} \mathbf{1}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{1 \times 5} & \mathbf{1}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{1}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{1}_{1 \times 5} & \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{0}_{1 \times 5} & \mathbf{1}_{1 \times 5} \end{pmatrix} \in \mathbb{R}^{5 \times 25},$$

which must be provided to IPOPT. Note that we supply the Jacobian with respect to input constraints only, since the state constraints are added to the objective function. The barrier parameter is set to  $\mu = 10^5$ . After running the optimization algorithm, with the adjoints and the interior point method, we end up with an NPV of 4698. **Fig. 5** and **Fig. 6** below are the plots of NPV as a function of the iteration number, water saturation at the producer wells, and the optimized control inputs. **Fig. 6** shows that the injector acts to decrease its rate as the saturation constraints at the producer wells are active. One of the producers, *prd1* at grid block (1, 1), has the highest rate change since it is located in low permeability area. This also applies to *prd4*, grid block (31, 41), but not as drastically as *prd1*. Note that in **Fig. 6** the convention sign for the injector is positive and producers are negative. Moreover, with these setting of prices, NPV is the same as maximizing the recovery factor. It takes 7.2 of pore volume of injected water to achieve this water saturation level. The interior point method successfully equalizes the water saturation at each producer well, which is consistent with our optimization goal. This can be seen in **Fig. 6** since the saturation values are approaching 0.65 but they do not exactly reach this value.

If we change the barrier parameter the optimization ends close to NPV of 4699 for a large range of values. If we increase the value we do observe a final water saturation closer to 0.65 and vice versa. It does however never reach 0.65 for positive  $\mu$  values. The optimization run appears to be robust towards changes in the initial values provided they are feasible. The optimization runs suggest a rapid improvement after only a few iterations. Hence, in some situations it may make sense to terminate the optimization after 5 – 10 iterations. We also run the optimization using KNITRO which gives almost identical results as IPOPT.

**Case 2** If the producer rates are chosen to be  $4 \text{ m}^3/\text{day}$  this yields an NPV of \$72.9 million and 0.27 pore volume of water injection. After running IPOPT in this case, we end up with an NPV of \$130.4 million and 0.85 of pore volume water injection. Almost identical results are obtained for other feasible input sequences. Similar to the first case we have to supply the Jacobian of the constraint to IPOPT, which has the same pattern as the first case, but with dimension  $\mathbb{R}^{5 \times 450}$ . We then run the optimization using the KNITRO which yields an NPV of \$137.2 and 0.87 of pore volume water injection.

As seen in **Fig. 7** the total water production rate honor the  $500 \text{ m}^3/\text{day}$  constraint. Moreover, the IPOPT result is not close to the boundary as opposed to the KNITRO. Consequently, KNITRO achieves a higher NPV than IPOPT. In general the line search method fails to handle a barrier function with a large barrier parameter,  $\mu$ , since it leads to an ill-conditioned Hessian matrix of modified objection function  $\mathcal{J}_b$ . Therefore a trust region method is more preferable. As a consequence, seen in **Fig. 8**, **Fig. 9**, KNITRO has a large spread in the optimized well rates. Furthermore, the well segments which are located in the low permeability regions have higher well rates. This is shown in **Fig. 10** and **Fig. 11** where it is important to note the difference in the color coding in the two figures.

## Discussion

In both cases, it is shown that the interior point method succeeds to achieve our optimization goal. In the first case the optimization terminates due to the number of iteration while in the second case because of the step length. The gradient from the adjoint is implemented analytically as explained in Eqs. 12 and 14.

One may notice that the barrier function term in the modified objective function, shown in Eqs. 15 and 16, may lead to an undefined number since we use the logarithmic function. This might happen if the optimized control inputs yield a value on the

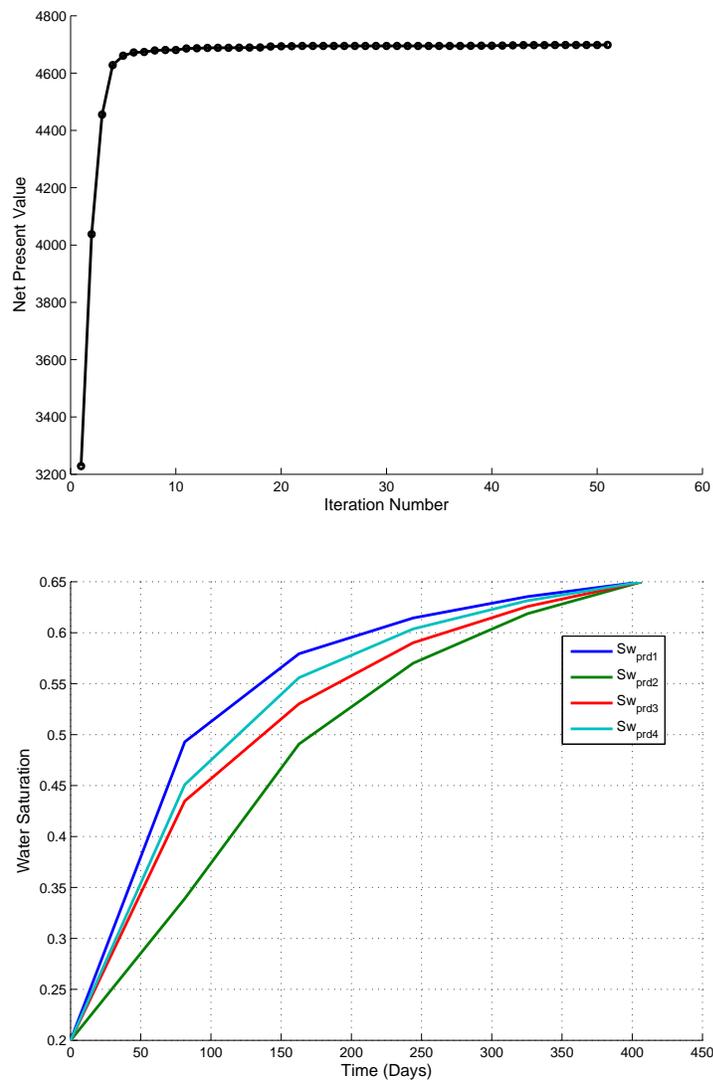


Fig. 5—The evolution of Net Present Value and state constraint satisfaction

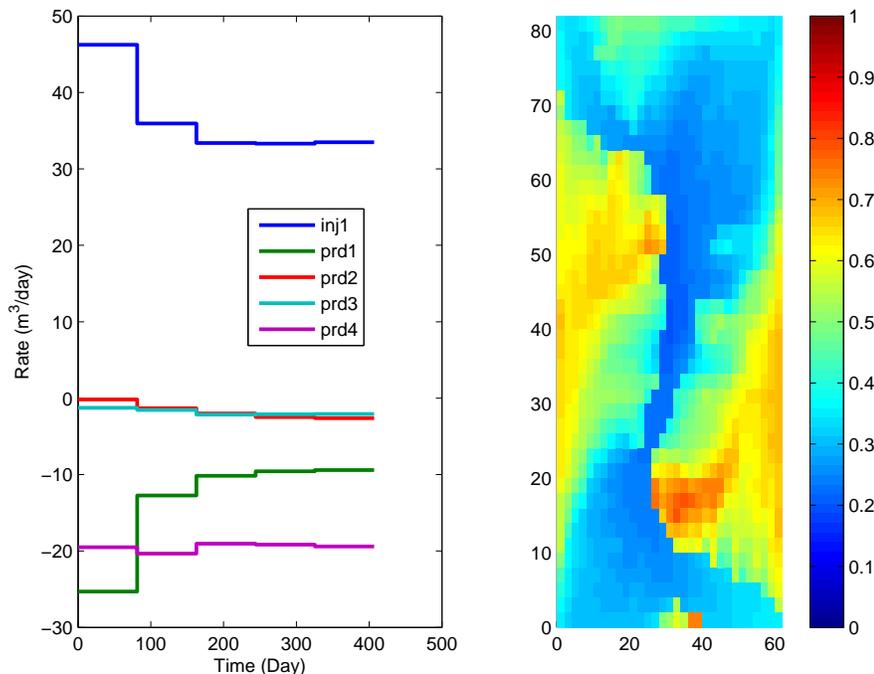


Fig. 6—The optimized well rates and oil saturation at final time

CPU time (in seconds)	with state constraint	without state constraint
Case 1	410-717	90-122
Case 2	232-250	282-294

TABLE 1—Comparison of CPU time. In case 1 IPOPT is used with  $\mu = 10^5$  and in case 2 KNITRO is used with  $\mu = 10^8$ . For each solver, ten runs have been completed and the minimum and maximum run time are shown

boundary of its feasible set. To prevent this, we put a condition in the objective function that if the value is undefined we set it to a real number whose value is negative infinite. The obtained gradients for both cases are verified by using the *derivative check* facility in IPOPT.

The results in this work are obtained using large barrier parameter values. Hence, the first term in Eq. 15 dominates the expression except for situations where the constraint is almost active. We will continue this work by studying the trade off between  $\mu$  and the distance output constraints.

We compare CPU time for the both cases with and without state constraints using IPOPT and KNITRO, respectively. In all cases the output constraints are violated in the "without state constraint" column as opposed to the runs in the "with state constraint" column. In case 1 we use IPOPT and experience a substantial increase in CPU time when we include state constraints. It is however far below the CPU time if we were to apply finite differences. In case 2 we use KNITRO and experience a slightly shorter run time when state constraints are included. The reason is the fact that in this particular case the optimization algorithm with state constraints converges faster, i.e. with fewer iterations than without state constraints. We made a similar test using IPOPT where the run time was similar in both cases, i.e. with and without state constraints.

## Conclusions

We have shown that the interior point method is able to tackle the state constraint problem by adding these constraints as a barrier function. The efficiency of computing gradients using the adjoint method is still to a large degree maintained. Furthermore, there is only one gradient needed for both the objective function and state constraints. If there are input constraints, then computing the Jacobian of these can be done easily by hand.

## Acknowledgements

This work is conducted as a research project within Program 2, Reservoir Management and Production Optimization, at The Center of Integrated Operation at the Norwegian Science and Technology University, Norway.

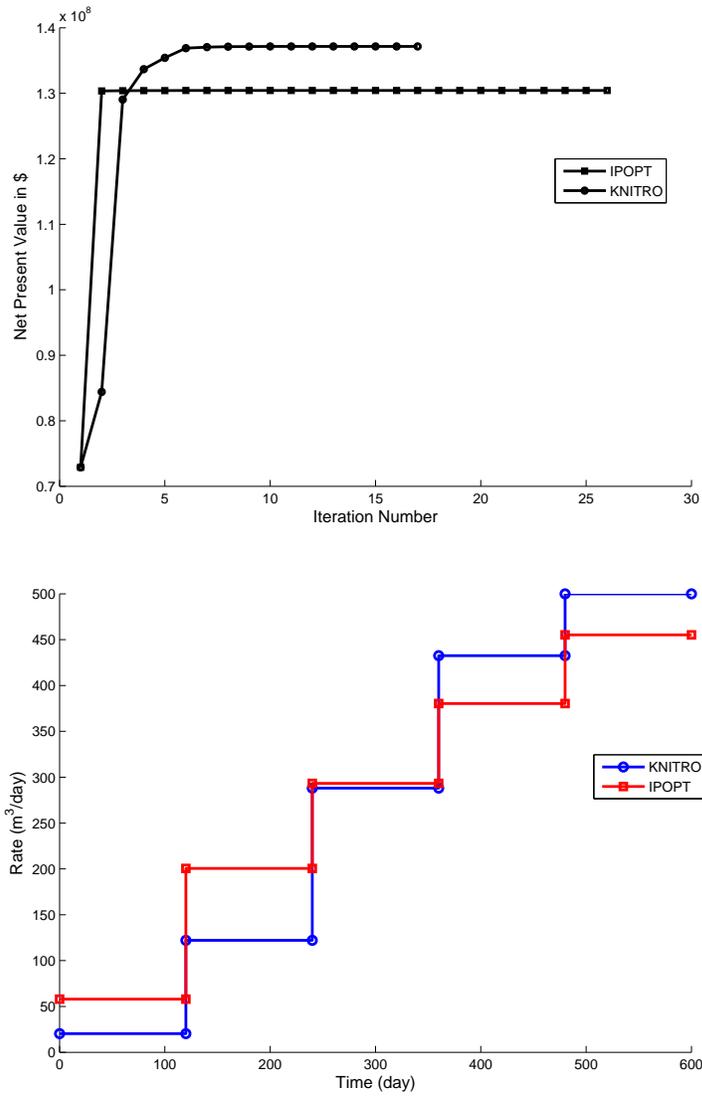


Fig. 7—The evolution of Net Present Value and Total water production constraint satisfaction

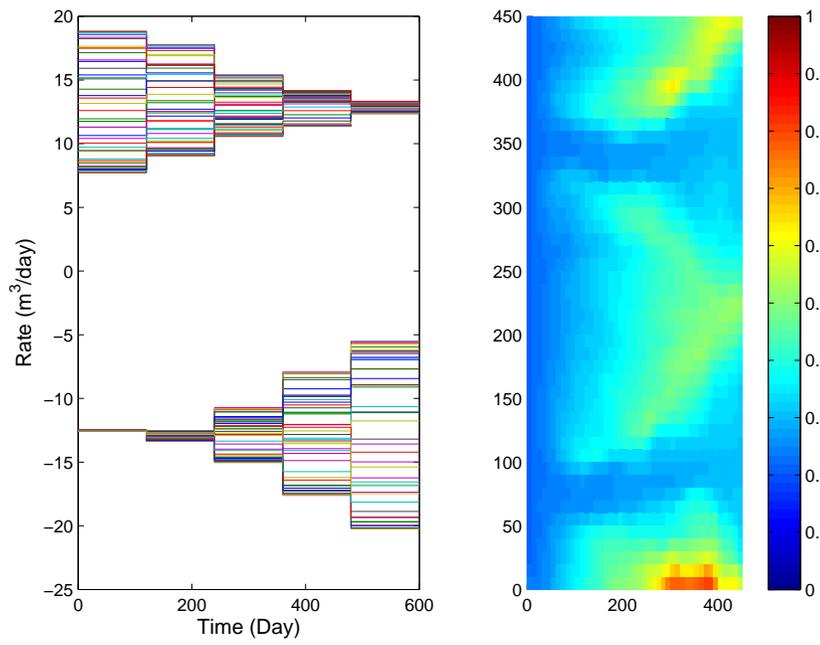


Fig. 8—Optimized well rates and oil saturation at final time of IPOPT result

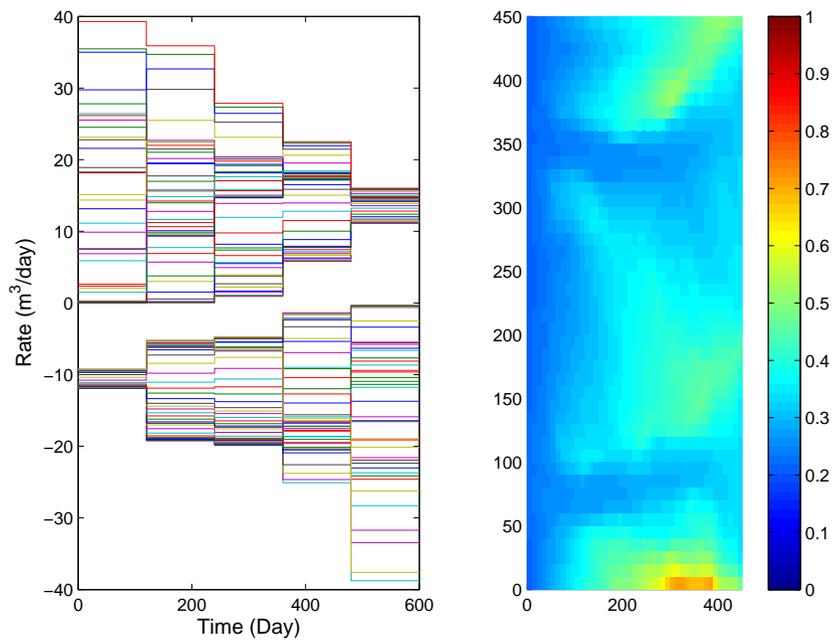


Fig. 9—Optimized well rates and Oil saturation at final time of KNITRO result

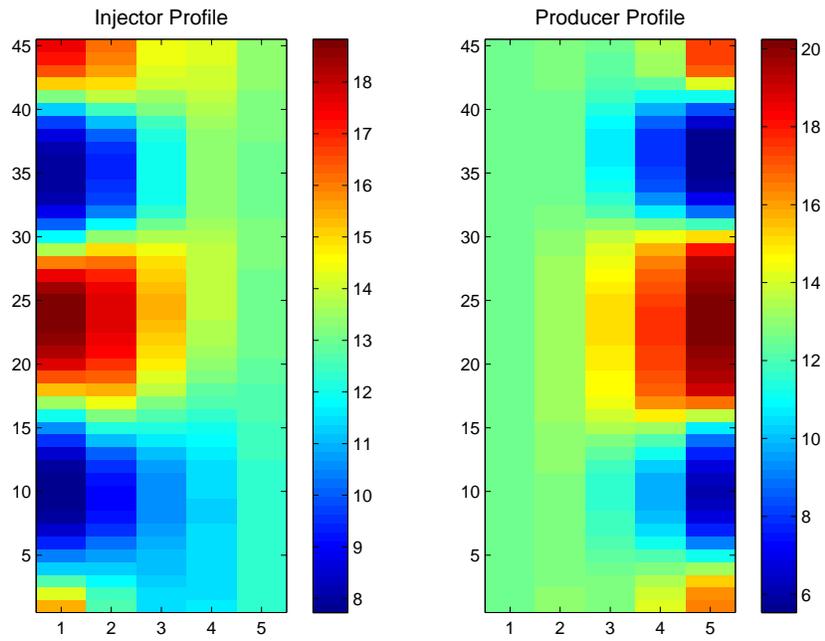


Fig. 10—IPOPT - injector and producer rates profile in 5 control intervals

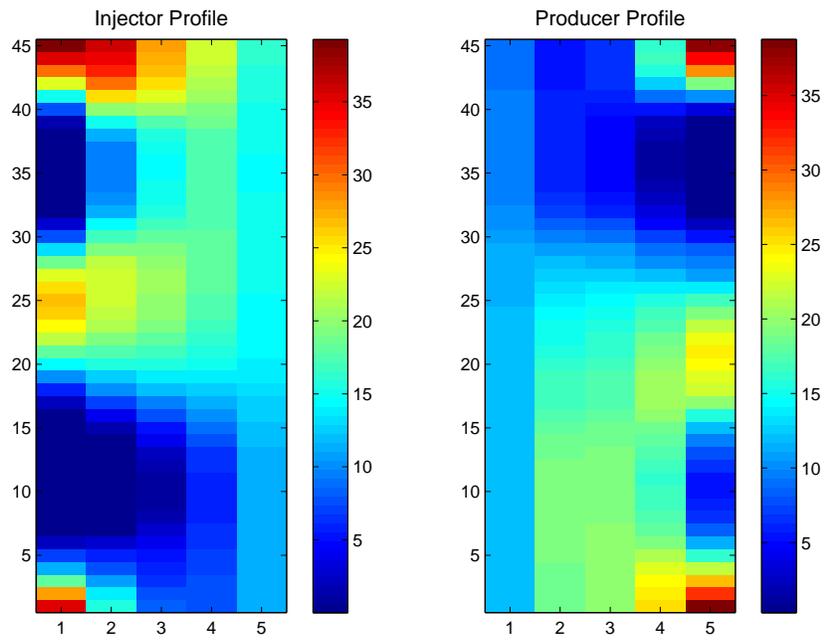


Fig. 11—KNITRO - injector and producer rates profile in 5 control intervals

## Nomenclature

### Physical quantities:

$f$	=	fractional flow function
$\mathbf{K}$	=	absolute permeability
$p$	=	pressure
$S, s$	=	saturation
$t$	=	time
$\vec{v}$	=	total Darcy velocity
$q$	=	volumetric rate
$WI$	=	well index
$\lambda$	=	total mobility
$\phi$	=	porosity
$\pi$	=	face pressure

### Functions, etc:

$F$	=	reservoir equations
$\mathcal{J}$	=	objective function
$t, \mu$	=	barrier parameter

### Vectors and matrices:

$s$	=	vector of cell/block saturations
$p$	=	vector of cell/block pressures
$\pi$	=	vector of cell/block face pressures
$v$	=	vector of outward fluxes on cell/block faces
$T$	=	transmissibility matrix
$g$	=	input constraint
$h$	=	state constraint
$x$	=	state vector
$u$	=	control input vector
$\lambda_\alpha$	=	Lagrange multiplier corresponding to state $\alpha$

### Numbers:

$N$	=	number of time steps
-----	---	----------------------

### Subscripts:

$i, j$	=	block/cell numbers
$f$	=	fine grid/model quantity
$w$	=	well number

### Superscripts:

$w$	=	well number
$n$	=	time step

## References

- Aarnes, J. E., Gimse, T., and Lie, K.-A. 2007. *An introduction to the numerics of flow in porous media using Matlab*. Springer Verlag, pp. 265-306. Url: <http://sintef.org/Projectweb/GeoScale/Simulators/MRST/>.
- Aziz, K., and Settari, A. 1979. *Petroleum Reservoir Simulation*. Applied Science Publishers.
- Bloss, K.F., Biegler, L.T., and Schiesser, W.E. 1999. Dynamic Process Optimization through Adjoint Formulations and Constraint Aggregation. *Ind. Eng. Chem. Res.*, 38(2):421–432.
- Bonnans, J.F., Gilbert, J.C., Lemarechal, C., and Sagastizabal, C.A. 2006. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Verlag.
- Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.
- Brouwer, D.R. 2004. Dynamic water flood optimization with smart wells using optimal control theory. PhD Thesis, TU Delft.
- Bryd, R.H., Nocedal, J., and Waltz, R.A. 2006. KNITRO: An Integrated Package for Nonlinear Optimization. In G. di Pillo and M. Roma, editor. *Large-Scale Nonlinear Optimization*. Springer Verlag, pp. 35-59.
- Castineira, D., Alpak, F.O., and Hohl, D. 2009. Automatic Well Placement Optimization in A Channelized Turbidite Reservoir Using Adjoint Based Sensitivities. Paper SPE 119156 presented at 2009 Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2–4 February 2009.
- Christie, M. A. and Blunt, M. J. 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *SPE Reservoir Eval. Eng.*, 4:308–317. Url: <http://www.spe.org/csp/>.
- Griewank, A., Korzec, and M. 2005. Approximating Jacobians by the TR2 formula. *Proc. Appl. Math. Mech.*, 5:791–792.
- Gunzburger, M.D. 2003. *Perspectives in Flow Control and Optimization*. SIAM Publisher.
- Jansen, J.D., Brouwer, D.R., Naevdal, G., and van Kruijsdijk, C.P.J.W. 2005. Closed-loop reservoir management. *First Break.*, 23:43–48.
- Kraaijevanger, J. F. B. M., Egberts, P. J. P., Valstar, J. R., and Buurman, H. W. 2007. Optimal Waterflood Design Using the Adjoint Method. Paper SPE 105764 presented at 2007 SPE RRS, Houston, TX, U.S.A., 26–29 February.
- Krogstad, S., Hauge, V.L., and Gulbransen, A.F. 2009. Adjoint Multiscale Mixed Finite Elements. Paper SPE 119112 presented at 2009 SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February.
- Mehra, R., and Davis, R., D. 1972. A Generalized Gradient Method for Optimal Control Problems with Inequality Constraints and Singular Arch. *IEEE Transactions on Automatic Control*, 17:69–79.

- Montleau, P. de., Cominelli, A., Neylon, K., Rowan, D., Pallister, I., Tesaker, O., and Nygard, I. 2006. Production Optimization under Constraints Using Adjoint Gradient. In *Proceedings of ECMOR X-10th European Conference on the Mathematics of Oil Recovery* number A041, Amsterdam, The Netherlands. EAGE.
- Nesterov, Y., and Nemirovsky, A. 1994. Interior-point polynomial methods in convex programming. *Studies in Applied Mathematics vol 13*, SIAM.
- Nocedal, J., and Wright, S.J. 2006. *Numerical Optimization*. Springer Verlag.
- Peaceman, D. 1983. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability. *SPE J.*, 23(3):531–543. Paper SPE 10528.
- Rommelse, J.R. 2009. Data Assimilation in Reservoir Management PhD Thesis, TU Delft.
- Sarma, P., Chen, W. H., Durlofsky, L. J., and Aziz, K. 2008. Production Optimization with Adjoint Models Under Nonlinear Control-State Path Inequality Constraints. *SPEREE*, 11(2):326–339. Paper SPE 99959.
- Sarma, P., Durlofsky, L., Aziz, K., and Chen, W. 2006. Efficient Real-Time Reservoir Management Using Adjoint-Based Optimal Control and Model Updating. *Comput. Geosci.*, 10(1):3–36.
- Talagrand, O., and Courtier, P. 1987. Variational Assimilation of Meteorological Observations with The Adjoint Vorticity Equation I : Theory. *Q.J.R. Meteorol. Soc.*, 113:1311–1328.
- Vlemmix, S., Joosten, G.J.P., Brouwer, D.R., and Jansen, J.D. 2009. Adjoint-Based Well Trajectory Optimization in a Thin Oil Rim. Paper SPE 121891 presented at 2009 SPE EUROPAC/EAGE Annual Conference, Amsterdam, The Netherlands, 8–11 June.
- Wu, Z. 1999. Conditioning Geostatistical Models to Two-Phase Flow Production Data PhD Thesis, University of Tulsa.
- Wachter, A., and Biegler, L. T. 2006. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57.
- Zandvliet, M.J. 2008. Model-based Lifecycle Optimization of Well Locations and Production Settings in Petroleum Reservoirs. PhD Thesis, TU Delft.