Abstract: The water flooding optimization problem is an important challenge in reservoir engineering, and is often approached using adjoint-based techniques. Optimal control theory, using an adjoint formulation, offers an efficient way to obtain gradients of an objective function with respect to the control inputs. However, a known limitation in this approach is the difficulty of systematically incorporating constraints. In this work, the system is fully discretized in space and time into a large-scale nonlinear program, with an objective function and a set of constraints representing the model equations. This simultaneous method is commonly termed direct transcription, and allows a simultaneous approach to the optimization problem. The approach relies on effective general purpose optimization solvers for an optimal control policy, and does not require repeated simulation of the system. Constraints are given either as bounds on variables or as more complex expressions, such as limits on power consumption, implemented as equalities or inequalities. The approach we suggest here is well suited to handle such path constraints. We also present a smooth approximation to upstream weighting of relative permeabilities to avoid the use of integer variables in the optimization problem. The method is demonstrated by solving a test problem.

1. INTRODUCTION

One of the main goals in reservoir management is to maximize the oil recovery of the reservoir, and optimization of the water-flooding process has been a subject of active research in recent years. Several studies, including Brouwer and Jansen [2004], Sarma et al. [2008], and Jansen et al. [2008], focus on classical control theory, using an adjoint formulation and a gradient-based approach with repeated simulation to obtain a locally optimal set of controls. Known disadvantages with this approach include that it can be difficult to find the correct switching structure and a suitable guess for initial state and adjoint variables if the problem requires handling of active inequality constraints [Kameswaran and Biegler, 2006], as well as the possibly high cost of multiple simulations of the system.

In this study, we apply a so-called simultaneous approach [Biegler, 1984]. This class of techniques fully discretizes the state and control variables, leading to large-scale nonlinear programming (NLP) problems which usually require special optimization strategies [Cervantes et al., 2000]. As a result, these methods directly couple the solution of the differential-algebraic equation (DAE) system with the optimization problem; the DAE system is solved only once (provided an infeasible path algorithm is used, such as sequential quadratic programming or interior-point methods), at the optimal point, and therefore can avoid intermediate solutions that may not exist or may require excessive computational effort [Kameswaran and Biegler, 2006]. This method has been successfully used on large-scale optimal control problems in process engineering (see, e.g., Cervantes and Biegler [1998]). The method has not been applied in reservoir engineering, but has been successfully applied to a number of downstream problems (see the work by Biegler and coworkers.)

This article is organized as follows: Section 2 gives a brief description of the reservoir model, introduces the smooth approximation to upstream weighting of relative permeabilities (9) and states the continuous-time optimization problem. The system is discretized in time in Section 3 and the finite-dimensional NLP is presented. A test problem that illustrates the method is presented in Section 4, and Section 5 contains some conclusions and descriptions of ongoing and future work.
2. RESERVOIR FLOW MODEL

Rather than using the conventional state-space formulation (a set of ordinary differential equations (ODEs) for the states), we use a network formulation for the reservoir model, as this lends itself better to the direct transcription approach. Here we consider a classical line-drive flooding problem for a horizontal, 2D, two-phase (oil/water), heterogeneous reservoir with two horizontal smart wells, one injector and one producer, located at opposite sides of the reservoir (see Figure 1). This problem is similar to the one studied by Brouwer and Jansen [2004]. The reservoir has no-flow boundary conditions at all sides. Each well is divided into segments with inflow control valves (ICVs).

Fig. 1. Schematic of water flooding with horizontal smart injection and production wells. Figure taken from Brouwer and Jansen [2004].

In the following model, all variables are time dependent, which is not explicitly noted. The reservoir is divided into a finite number of grid blocks whose geological and fluid properties are assumed to be homogeneous. Grid block number is denoted by subscripts i and j (j is assumed to be adjacent to i), and phase (oil: o, water: w) is denoted by subscript α. See Table 1 for parameter names and values and Section 6 for additional nomenclature.

The saturation of a fluid α is denoted \( S_\alpha \) and defined as the fraction of a volume filled by that phase. We assume that the petroleum reservoir contains only hydrocarbons and water. Hence, since the two phases \( o \) and \( w \) fill the available volume, it follows that

\[
S_o + S_w = 1. \tag{1}
\]

We also assume that neither saturation can be 0 or 1. The minimum oil saturation is usually called residual oil saturation and denoted \( S_{or} \); the minimum water saturation is usually called connate water saturation and denoted \( S_{wc} \). This yields

\[
\begin{align*}
S_{or} &\leq S_o \leq 1 - S_{wc}, \tag{2a} \\
S_{wc} &\leq S_w \leq 1 - S_{or}. \tag{2b}
\end{align*}
\]

With these bounds, we can introduce the scaled saturations \( \sigma_\alpha \in [0, 1] \) through a two-phase version of the model due to Stone [1970]:

\[
\begin{align*}
\sigma_o &= \frac{S_o - S_{or}}{1 - S_{wc} - S_{or}}, \tag{3a} \\
\sigma_w &= \frac{S_{wc} - S_{wc}}{1 - S_{or} - S_{wc}}. \tag{3b}
\end{align*}
\]

The curvature and surface tension in the interface between the two phases leads to a higher pressure in oil than in water. The difference between these two pressures is called the capillary pressure

\[
p_c = p_o - p_w. \tag{4}
\]

However, this capillary effect will be neglected, and it is hence assumed that

\[
p := p_o = p_w, \quad p_c \equiv 0. \tag{5}
\]

Grid block absolute permeabilities \( k_i \) are given as parameters, and assumed to be isotropic. The geological permeability at the interface between the grid blocks \( i \) and \( j \) is denoted \( k_{ij} \), and calculated using harmonic average,

\[
k_{ij} = \frac{2}{k_i + k_j}. \tag{6}
\]

The relative permeability in grid block \( i \) for a given phase \( \alpha \) is denoted \( k_{i,r,\alpha} \) and is saturation dependent. This is modeled through the Corey-type relation [Corey, 1954]

\[
k_{i,r,\alpha} = k_{i,r,\alpha}(\sigma_{\alpha})^{n_\alpha}, \tag{7}
\]

where \( k_{i,r,\alpha} \) is the endpoint relative permeability and \( n_\alpha \) is the Corey exponent. Here, we set \( n_o = 1 \) and use linear permeability curves.

Relative permeability at the interface between the grid blocks \( i \) and \( j \) is denoted \( k_{i,j,r,\alpha} \) and is normally calculated using upstream weighting:

\[
k_{i,j,r,\alpha} = \begin{cases} k_{i,r,\alpha}, & \text{if } p_i \geq p_j, \\
k_{j,r,\alpha}, & \text{if } p_i < p_j. \end{cases} \tag{8}
\]

However, implementation of (8) requires the use of integer variables resulting in a mixed-integer nonlinear program (MINLP), which is far more complex to solve than an NLP. To avoid the use of integer variables, we approximate (8) through

\[
(k_{i,j,r,\alpha} - k_{i,r,\alpha}) e^{c(p_i - p_j)} = (k_{i,j,r,\alpha} - k_{j,r,\alpha}) e^{-c(p_i - p_j)}, \tag{9}
\]

where \( c \) is a constant. A large \( c \) gives a more accurate approximation than a small \( c \), but will also introduces large derivatives which must be compensated for by scaling the problem appropriately. Even though (9) is a nonlinear equation, containing exponentials which can be challenging for NLP solvers, experience shows that using the approximation instead of (8) offers a significant increase in convergence rate.

Assuming uniform reservoir thickness, isothermal conditions, slightly compressible fluids and rock formation, and neglecting gravity forces, the mass balance (see, e.g., Aziz and Settari [1979]) for the reservoir is

\[
h \rho_o \phi \left( S_o(c_o + c_w) \frac{\partial}{\partial t} \right) + \nabla \cdot \left( h \frac{\rho_o}{\mu_o} \nabla \cdot (\rho_o k_{r,\alpha} k \nabla p) + h \rho_o q^\prime o \right) = 0, \tag{10}
\]

where \( \rho_o \) is density and \( q^\prime o \) is a possible source/sink term representing a well flow rate with unit \( s^{-1} \). (\( q^\prime o \equiv 0 \) in grid blocks without wells.) Discretizing this equation in space...
with the finite-difference method and a uniform block-centered grid results in
\[ V\phi ˙S_{i,\alpha} + V\phi S_{i,\alpha}(c_r + c_\alpha) ˙p_i = \ldots \] (17a)
where \( v \) and \( q \) are defined in (12) and (13), respectively.

The transmissibility between two adjacent grid blocks is proportional to the product of the relative permeability (9) and the absolute permeability (6), which again gives the flow between two block grids when multiplied by the pressure difference:
\[ v_{ij,\alpha} = \frac{h}{\mu_\alpha} \Delta y K_{ij} k_{ij,r,\alpha} \Delta p_{ij}. \] (12)
The source/sink term in (10) represents well flow rates, and since square grid block bases are assumed, we can use the model by Peaceman [1978],
\[ q_{i,\alpha} = \frac{2\pi h k_{i,r,\alpha} h}{\mu_\alpha} \frac{p_{\text{well},i} - p_i}{\ln(0.2 \Delta x r_{\text{well}})}, \] (13)
for production wells. For injection wells, we set \( q_{i,\alpha} = 0 \) and let \( q_{i,w} \) be a degree of freedom with appropriate bounds.

For simplicity, maximized oil recovery was chosen as the objective, formulated as
\[ \max \sum_i S_{i,w}(t_F), \] (14)
where \( t_F \) is the final time. That is, maximization of the total volume of water in the reservoir at final time, which is equivalent to minimizing the total volume of oil left in the reservoir.

This gives the optimal control problem
\[ \max \varphi(x(t_F)) \]
\[ \text{s.t. } \dot{x} = f(x(t), z(t), u(t))), \] (15a)
\[ g(x(t), z(t), u(t)) = 0 \] (15b)
\[ x_L \leq x(t) \leq x_U \] (15c)
\[ z_L \leq z(t) \leq z_U \] (15d)
\[ u_L \leq u(t) \leq u_U \] (15e)
which is an infinite dimension NLP, where \( x(t) \in \mathbb{R}^{n_x} \) is a vector of differential state variables, \( z(t) \in \mathbb{R}^{n_z} \) is a vector of algebraic state variables, and \( u(t) \in \mathbb{R}^{n_u} \) is a vector of control variables. The set of differential equations (15b) contains (11) for both phases and all blocks, while (15c) contains all algebraic relationships. Tight bounds for all variables can be calculated based on the physical well specifications. Note that the objective function can be extended to include an integral over time by introducing an extra differential state to the problem.

3. TIME DISCRETIZATION

The reservoir is discretized using collocation at Radau points (also known as Gauss-Radau or Legendre-Gauss-Radau quadrature). Direct transcription methods using collocation fully discretize the optimal control problem by approximating the control and state variables as piecewise polynomial functions over finite elements, often by implicit Runge-Kutta methods (Kameswaran and Biegler [2006]).

In this section, \( i \) refers to finite elements while \( j \) refers to collocation points. Given \( K \) collocation points (or Runge-Kutta stages), Gauss quadrature has the highest precision at \( 2K \), while Radau quadrature has precision \( 2K - 1 \) [Ascher and Petzold, 1998]. Despite its lower order, we choose Radau quadrature because the endpoint of the interval is included, allowing easy formulation of constraints at the end of each finite element. The Radau collocation methods considered here are fully implicit Runge-Kutta methods (corresponding to the Radau IIA class), and are both A-stable and stiffly accurate and hence have stiff decay [Ascher and Petzold, 1998]. Moreover, they are L-stable and B-stable – and hence AN-stable [Hairer and Wanner, 2010], which is very relevant for this problem. (See the two aforementioned references for definitions of these properties of Runge-Kutta methods.)

We use the following time discretization from Kameswaran and Biegler [2006]. Let \( \tau_j, j \in \{1, \ldots, K\} \), denote the \( K \) collocation points, with \( \tau_0 = 0 \). Within each finite element of length \( h_i \), \( \tau \) represents normalized time, so that
\[ t = t_i \tau_i, \] (16a)
\[ t \in \left[ t_i \tau_i, t_{i+1} \tau_{i+1} \right], \] (16b)
\[ \tau \in [0, 1]. \] (16c)
The differential state variables \( x(t) \) are approximated in discretized form over each finite element as
\[ x_{K+1}(t) = \sum_{j=0}^{K} \ell_{j}(\tau)x_{ij}, \] (17a)
where
\[ \ell_{j}(\tau) = \frac{\tau - \tau_{k-j}}{\tau_{j} - \tau_{k}} \] (17b)
is a Lagrange interpolation polynomial of degree \( K + 1 \), (hence the notation \( x_{K+1}(t) \)). Over the same element, the algebraic states \( z(t) \) and the control variables \( u(t) \) are approximated in an analogous manner by
\[ u_{K}(t) = \sum_{j=1}^{K} \tilde{\ell}_{j}(\tau)u_{ij}, \] (18a)
\[ z_{K}(t) = \sum_{j=1}^{K} \tilde{\ell}_{j}(\tau)z_{ij}, \] (18b)
where
\[ \tilde{\ell}_{j}(\tau) = \frac{\tau - \tau_{k-j}}{\tau_{j} - \tau_{k}} \] (18c)
is a Lagrange interpolation polynomial of degree \( K \). The difference in order is due to the initial condition on \( x(t) \). The polynomial coefficients \( x_{ij}, u_{ij} \) and \( z_{ij} \) will be variables in the NLP and have the same bounds as their continuous counterparts. This comes from the property \( \ell_{j}(\tau_{k}) = \delta_{jk} \) of the Lagrange polynomials (\( \delta_{jk} \) is the Kronecker delta). Moreover, this leads to \( x(t_{ij}) = x_{ij} \), which means that the approximation is exact at the collocation points.

For discretization of the ODE, we take the time derivative of (17a) and require that the ODE (15b) be satisfied at the collocation points. From (16a) we have \( dt = h_i d\tau \), so that
\[ \sum_{j=0}^{K} \dot{\ell}_j(\tau_k)x_{ij} = h_if(x_{ij}, z_{ij}, u_{ij}), \quad x_{1,0} = x_0, \quad (19) \]

where \( \dot{\ell}_j(\tau) = \frac{d\ell_j(\tau)}{d\tau} \). Continuity of the differential states on the element boundaries is enforced through

\[ x_{i+1,0} = \sum_{j=0}^{K} \ell_j(1)x_{ij}, \quad i \in \{1, \ldots, N-1\}. \quad (20) \]

Requiring that the algebraic constraints are satisfied at the collocation points is formulated as

\[ g(x_{ij}, z_{ij}, u_{ij}) = 0. \quad (21) \]

We can now represent the infinite dimensional optimal control problem (15) as the finite-dimensional NLP

\[ \max \varphi(x_F) \quad \text{(22a)} \]

s.t.

\[ \sum_{j=0}^{K} \ell_j(\tau_k)x_{ij} = h_if(x_{ij}, z_{ij}, u_{ij}), \quad x_{1,0} = x_0 \quad \text{(22b)} \]

\[ x_{i+1,0} = \sum_{j=0}^{K} \ell_j(1)x_{ij} \quad \text{(22c)} \]

\[ g(x_{ij}, z_{ij}, u_{ij}) = 0 \quad \text{(22d)} \]

\[ x_L \leq x_{ij} \leq x_U \quad \text{(22e)} \]

\[ z_L \leq z_{ij} \leq z_U \quad \text{(22f)} \]

\[ u_L \leq u_{ij} \leq u_U \quad \text{(22g)} \]

where \( x_F = x_{N,K} \).

4. TEST PROBLEM

The presented approach was applied to a test problem. The reservoir in Figure 1 was divided into 16 grid blocks (4 by 4), numbered one-dimensionally (grid block 2 is to the right of grid block 1, which is at the top-left corner, cf. Figure 2). The permeabilities are chosen to be in the range \([10^{-13}, 10^{-12}]\), the permeability field is shown in Figure 2. Grid blocks 1, 5, 9 and 13 have injection wells, while grid blocks 4, 8, 12 and 16 have production wells. The porosity field is homogeneous, and we assume incompressible fluids and porous rock. The finite element lengths \( h_i \) are short in the beginning of the time horizon, and longer towards the end. Furthermore, injection rates are required to be piecewise constant over each finite element. This clearly leads to a suboptimal solution, and is discussed further in Section 5. The parameters used are summarized in Table 1.

A nice feature with this problem structure is that we can assume both phases to always flow in the positive \( x \)-direction (from left to right in Figure 2). From this assumption it follows that flow \( v_{i,j,o} \) from a grid block \( i \) to the adjacent grid block \( j \) to the right of \( i \) can always use \( k_{j,i,r,o} \) as the relative permeability \( k_{i,j,r,o} \). This allows us to only implement the upstream weighting (8) or the approximation (9) for grid blocks that are adjacent in the \( y \)-direction (vertically in Figure 2). Hence, the nonlinearity of the problem is reduced (or the number of binary variables is reduced if (8) is implemented), which again reduces the problem complexity.

4.1 Implementation

The model was implemented in GAMS [GAM, 2009] with IPOPT [Cervantes et al., 2000] as the local NLP solver. IPOPT is a state-of-the-art interior-point NLP solver, which exploits the sparse structure of the NLP and is capable of solving very large problems. MATLAB was used to set up the collocation and reservoir parameters, and for generating the polynomial representations. Formulating the reservoir model as a network has a range of advantages for this implementation method. We started the problem at a feasible but trivial point, which offers a slight increase in performance compared to starting all variables at zero. Starting the problem from a nontrivial feasible trajectory (obtained from a simulator) could accelerate convergence to the local optimum; this, however, was not attempted.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h )</td>
<td>grid block height</td>
<td>20</td>
<td>m</td>
</tr>
<tr>
<td>( \Delta x )</td>
<td>grid block length</td>
<td>500</td>
<td>m</td>
</tr>
<tr>
<td>( \Delta y )</td>
<td>grid block width</td>
<td>500</td>
<td>m</td>
</tr>
<tr>
<td>( V )</td>
<td>grid block volume</td>
<td>( 5000 \times 10^3 )</td>
<td>m³</td>
</tr>
<tr>
<td>( \mu_o )</td>
<td>oil dynamic viscosity</td>
<td>( 5 \times 10^{-4} )</td>
<td>Pas</td>
</tr>
<tr>
<td>( \mu_w )</td>
<td>water dynamic viscosity</td>
<td>( 10^{-3} )</td>
<td>Pas</td>
</tr>
<tr>
<td>( \phi )</td>
<td>porosity</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>( k_i )</td>
<td>absolute permeability ( i )</td>
<td>([10^{-13}, 10^{-12}])</td>
<td>m²</td>
</tr>
<tr>
<td>( r_{well} )</td>
<td>well bore radius</td>
<td>0.114</td>
<td>m</td>
</tr>
<tr>
<td>( p_r )</td>
<td>initial reservoir pressure</td>
<td>( 3 \times 10^7 )</td>
<td>Pa</td>
</tr>
<tr>
<td>( c_r )</td>
<td>rock compressibility</td>
<td>0</td>
<td>Pa⁻¹</td>
</tr>
<tr>
<td>( c_o )</td>
<td>oil compressibility</td>
<td>0</td>
<td>Pa⁻¹</td>
</tr>
<tr>
<td>( c_w )</td>
<td>water compressibility</td>
<td>0</td>
<td>Pa⁻¹</td>
</tr>
<tr>
<td>( S_{or} )</td>
<td>residual oil saturation</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td>( S_{wc} )</td>
<td>connate water saturation</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td>( n_o )</td>
<td>Corey exponent, oil</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>( n_w )</td>
<td>Corey exponent, water</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>( k_{0,r,o} )</td>
<td>endpoint rel. perm.</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>( k_{0,r,w} )</td>
<td>endpoint rel. perm.</td>
<td>0.6</td>
<td>-</td>
</tr>
</tbody>
</table>

4.2 Results

The test problem was solved in less than 10 minutes on a standard desktop computer, starting from equilibrium conditions in the reservoir. Figure 3 shows snapshots of the saturation field for six different time instants. We clearly see that even though the reservoir contains two streaks with significantly lower permeability, the water front moves close to uniformly. Some oil is left in grid block 11, which has high permeability and is surrounded by grid blocks with lower permeability. Figure 4 shows water saturations in the grid blocks with wells plotted against time. The saturation first rises in the injection side of the
reservoir, and the water front does not hit the production side of the reservoir until later.

Fig. 3. Snapshots of saturation fields at different times (month) for the test problem. Black indicates $\sigma_{i,o} = 1$ while white indicates $\sigma_{i,o} = 0$.

Fig. 4. Water saturations in all grid blocks with wells for the test problem. Solid lines indicate injection wells, dashed lines indicate production wells.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The test problem shows that the approach presented here has merit and possible potential for solving larger problems. The results make physical sense, and show that this kind of problem can be solved with direct transcription methods. Although much work remains in the formulation and verification of results for larger problems, the current results motivate a deeper investigation of the capabilities of the method.

5.2 Future Work

Ongoing work includes investigating the performance of the method on much larger problems, without the assumptions of incompressible fluids and porous rock. Computational efficiency and the quality of the solutions will be compared to existing methods, including adjoint-based strategies. In order to reveal how the modeling assumptions impact the robustness and optimality of the results, the solutions will be tested on an advanced simulator.

We heuristically choose shorter finite elements for the most transient phase of the flooding process. This enables more accurate capture of the dynamics and a finer degree of control in the wells. A more advanced strategy is to let the optimization algorithm chose the element lengths. This increases the number of variables and the nonlinearity of the problem, but allows the algorithm to find optimal breakpoints for the controls when they are required to be constant within each element.

Intelligently choosing knot locations (the boundaries between finite elements) as part of the optimization problem can, in addition to the advantages mentioned above, provide control, minimization, estimation, and an equidistribution of error [Cuthrell and Biegler, 1987]. Vasantharajan and Biegler [1990] developed two strategies to control the approximation error within the framework of collocation on finite elements. These strategies were embedded within the nonlinear program and adjusted the length of the finite elements over the course of the optimization. We will develop all of those features in the current method imminently.

Zandvliet et al. [2007] investigated situations where the optimal injection strategy was either bang-bang or a combination of bang-bang and singular arcs. Extensions of that work could fit well with the framework we present here. Allowing the controls to be piecewise polynomials increases the problem size, but would allow for more interesting control profiles to be found.

When implementing the model in GAMS, a large range of high-quality optimization codes are available through the application’s many interfaces. These include state-of-the-art deterministic global optimization solvers, such as BARON [Sahinidis, 1996]. With careful formulation of the problem, we will investigate how well this problem lends itself to global optimization. Wartmann et al. [2010] showed how to aid the global solver in reaching the global optimum by adding redundant relaxation constraints in the form of topological network invariants. We have derived such network invariants for the water-flooding problem, and these can result in faster convergence to the global optimum with our network formulation. Other global optimization strategies, such as simulated annealing or genetic algorithms, could also give insight into the quality of the local solutions.

Additional future work includes introducing gravity effects in a 3D reservoir and including the gas phase in the model.

6. NOMENCLATURE

- $S_{i,\alpha}(t)$: Saturation of phase $\alpha$ in grid block $i$.
- $\sigma_{i,\alpha}(t)$: Scaled saturation of phase $\alpha$ in grid block $i$.
- $p_i(t)$: Pressure in grid block $i$.
- $k_{ij}$: Absolute permeability between grid blocks $i$ and $j$ (parameter).
- $k_{i,r,\alpha}(t)$: Relative permeability of phase $\alpha$ in grid block $i$.
- $k_{ij,r,\alpha}(t)$: Relative permeability of phase $\alpha$ at the interface between grid blocks $i$ and $j$.
- $q_{i,\alpha}(t)$: Volumetric well flow rate of phase $\alpha$ in grid block $i$.
- $v_{ij,\alpha}(t)$: Volumetric flow rate of phase $\alpha$ between grid blocks $i$ and $j$. 
ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding provided by the Center of Integrated Operations in the Petroleum Industry at NTNU, Trondheim, Norway.

REFERENCES