Use of Linearized Reduced-order Modeling and Pattern Search Methods for Optimization of Oil Production

Jincong He and Louis J. Durlofsky

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

Abstract

Computational optimization holds great promise for the management of oil field operations. These optimizations can be very expensive computationally, however, because each function evaluation requires a reservoir simulation, which is itself time consuming. In this paper, we present and apply a surrogate modeling procedure which greatly accelerates optimizations based on subsurface flow simulations. The surrogate model, a trajectory piecewise linearization (TPWL) technique, represents new states in terms of expansions around states simulated during training runs (the training runs require full-order simulations). The TPWL representation is incorporated into a generalized pattern search optimization procedure. Results for example problems demonstrate significant improvement in the objective function and a two order of magnitude reduction in the number of full-order simulations required for the optimization.

Keywords: Oil production optimization, surrogate modeling, trajectory piecewise linearization, TPWL, direct search methods, reservoir simulation

1. Introduction

Direct search methods are potentially of use for the optimization of oil production as they are straightforward to apply and do not require access to source code. These methods have recently been investigated by [1] and have been shown to perform well for oil production optimization, particularly when parallel computing resources are available. When the search space is high dimensional, however, as it typically is for practical reservoir management problems, these algorithms require a large number of objective function evaluations. The use of full-order, high-fidelity models for these function evaluations can become expensive, as practical simulations entail transient computations involving two or more components and may include tens or hundreds of thousands (or more) of grid blocks. There is therefore a significant need for accurate surrogate models that can be used in place of full-order simulations for these optimizations.

In this work we present a linearized reduced-order modeling procedure and apply it as a surrogate in optimizations involving water injection in oil reservoirs. The method, referred to as trajectory piecewise linearization (TPWL), has been applied in a number of application areas (e.g., [2, 3, 4]) but has only recently been studied within the context of reservoir management [5, 6]. Using states and Jacobian matrices generated during one or more training runs, TPWL constructs new solutions based on linear expansions around previously simulated states and well settings (controls). The method is very efficient because (1) the model is linearized and (2) the states and matrices are represented in a highly compressed form using a linear subspace mapping (proper orthogonal decomposition).

Retraining of the TPWL model is required when the simulated states lose accuracy. We perform this retraining within the context of a generalized pattern search algorithm. The overall procedure is applied to two oil field problems. The first example involves a small reservoir model, so comparison with an optimization using the full-order model is readily accomplished. In this case the use of TPWL as a surrogate is shown to lead to essentially the same objective function value as is achieved using the full-order model. The second example involves a larger reservoir model, 54 optimization variables, and bound and nonlinear constraints. In this case we do not compare with an optimization based on the full-order model but we show significant improvement in the objective function using TPWL as a surrogate. For both cases, the optimizations using TPWL reduce the number of full-order simulations required by a factor of 100 or more.

This paper proceeds as follows. We first discuss briefly the equations governing oil-water subsurface flow, the discrete system, and the TPWL representation. We then illustrate the performance of the TPWL model for an example case. Next, the use of TPWL as a surrogate in a generalized pattern search optimization procedure is described. Optimization results that demonstrate the effectiveness of this approach
2. Governing Equations and TPWL Representation

Our interest here is in the optimization of oil production. Following the early stage (primary) production of oil, water is frequently injected to maintain reservoir pressure and to drive oil to production wells. Optimization problems arising within this context include the determination of optimal locations for injection wells and the subsequent determination of optimal operating parameters for injection and production wells. The latter problem will be the focus of this work. The operating (optimization) parameters of interest include injection and production rates or well bottom hole pressures (BHPs) as a function of time.

We consider systems containing oil and water. The equations governing the subsurface flow of oil and water are derived by expressing mass conservation in the form of partial differential equations (PDEs). Constitutive relations, specifically Darcy’s law, are then used to relate the flow rates of the oil and water phases to pressure gradients. The resulting equations are nonlinear PDEs that can be solved to provide pressure and saturation (saturation is the volume fraction of a given phase) as a function of space and time. In practice, solutions are achieved numerically on discretized models. Finite-volume procedures are used most commonly in practice as they have a number of desirable features. The resulting system is, however, time consuming to solve as it typically contains $O(10^4 - 10^6)$ grid blocks, with each grid block requiring two unknowns (pressure and water saturation). The system must be solved in time. This motivates the need for fast surrogate models that are appropriate for use in optimization problems.

The detailed flow equations and discretization procedures are described in [7, 5]. Here we provide a high-level description of the discretized equations and the TPWL representation. The system states, pressure and water saturation, are designated $x$ and the controls, well BHPs, are designated $u$. The discretized nonlinear system can now be represented as:

$$g(x^{n+1}, x^n, u^{n+1}) = A(x^{n+1}, x^n) + F(x^{n+1}) + Q(x^{n+1}, u^{n+1}) = 0,$$  

(1)

where $g$ is the residual, which is zero upon convergence, $n$ and $n + 1$ indicate time levels, and $A$, $F$ and $Q$ are the discretized accumulation, flux and injection/production terms, respectively. Newton’s method is applied to solve Eq. (1). The Jacobian matrix $J$, given by $J = \partial g/\partial x$, is constructed at each iteration of each time step. The solution update $\delta$ is determined through solution of $J\delta = -g$.

The trajectory piecewise linearization (TPWL) procedure for these subsurface flow equations is developed in detail in [5] so our description here will be brief. Essentially, the method requires one or more full-order training simulations in which particular sequences of controls (BHPs) for each well are specified. From these training simulations, the states and Jacobian matrices (for the converged states) are saved at each time step. Then, in subsequent (test) simulations, such as those required for production optimization, rather than compute the full-order solution, the solution is represented as a linear expansion around a saved state. This allows us to represent new solutions ($x^{n+1}$) using:

$$J^{i+1}(x^{n+1} - x^{i+1}) = -\left[\frac{\partial A^{i+1}}{\partial x}(x^n - x^i) + \frac{\partial Q^{i+1}}{\partial u^{i+1}}(u^{n+1} - u^{i+1})\right].$$  

(2)

In this equation, superscripts $i$ and $i+1$ designate saved information (states or matrices). Because Eq. (2) is linear, its solution does not require iteration. This representation is not very practical, however, since all terms are in the original high-dimensional (full-order) space.

We proceed by introducing a linear mapping from the high-dimensional space to a subspace of much lower dimension; i.e., we apply $x \approx \Phi z$, where $\Phi$ is an orthonormal basis. In our implementation, $\Phi$ is computed from singular value decompositions of the matrices containing the saved pressure and saturation states. The $\Phi$ matrix is of dimensions $2N_c \times l$, where $N_c$ is the number of grid blocks in the high-fidelity model and $l = l_p + l_s$ is the total number of basis vectors, with $l_p$ and $l_s$ the number of basis vectors associated with pressure and water saturation states respectively. Significant reduction is achieved because $l << N_c$.

Representing $x$ in Eq. (2) as $x \approx \Phi z$, and then premultiplying both sides of the equation by $\Phi^T$, gives, after some manipulation,

$$z^{n+1} = z^{i+1} - (J^{i+1}_r)^{-1} \left[\frac{\partial A^{i+1}}{\partial x} \right]_r (z^n - z^i) + \left[\frac{\partial Q^{i+1}}{\partial u^{i+1}} \right]_r (u^{n+1} - u^{i+1}),$$  

(3)

are then presented. We conclude with a brief summary.
where the subscript \( r \) indicates reduced and \( J^+ = \Phi^T J^+ \Phi \). \( (\partial A^{i+1}/\partial x^i)_r = \Phi^T (\partial A^{i+1}/\partial x^i) \Phi \) and \( (\partial Q^{i+1}/\partial u^{i+1})_r = \Phi^T (\partial Q^{i+1}/\partial u^{i+1}) \). The vectors and matrices appearing in Eq. (3) are of dimensions \( l \) and \( l \times l \) respectively, so this equation can be solved very efficiently.

The full-order runs needed to generate the saved states, Jacobian matrices and \( \Phi \) matrix are in all cases performed using Stanford’s General Purpose Research Simulator, GPRS [8, 9]. The simulator has been modified to output the arrays required by TPWL.

The TPWL model given by Eq. (3) was shown to provide accurate results for a variety of examples in [5]. It was observed, however, that the model can encounter numerical stability problems for cases where the densities of the oil and water phases differ significantly. In recent work ([10]; J. Sætrom, private communication), enhanced procedures were introduced to stabilize the TPWL representation. The stabilization approach used in the examples presented here entails the selection of an optimized basis (i.e., the determination of optimal \( l_p \) and \( l_S \)) such that the stability of the resulting model is maximized. This is accomplished by minimizing the spectral radius of an appropriately defined amplification matrix. This method cannot guarantee numerical stability, but it has been found to perform reliably in extensive tests.

3. Model Definition and TPWL Simulation Results

We now demonstrate the application of TPWL to an example problem. The geological model used here, described in [6], represents a portion of a fluvial channel system. The model (referred to as Model 1), shown in Figure 1, contains a total of 20,400 grid blocks. Flow is driven by two injection wells and four production wells. Detailed model and fluid properties are given in [10]. We note that there is a significant difference (10 lb/ft\(^3\)) between the oil and water densities, so the stabilization procedure referred to above must be applied for this case.

![Figure 1: Model 1 and wells. Color indicates permeability in the x-direction.](image)

The training and test BHP schedules for each of the four production wells are shown in Figure 2. The BHPs of the two injectors are held constant at 6,000 psi. The training schedule shown on the left was generated randomly over a specified range and update frequency. As demonstrated in [5], TPWL generally performs accurately when the test schedule is ‘similar’ to the training schedule, though it tends to lose accuracy when the test schedule is very different from the training schedule. The test schedule on the right in Figure 2 was generated by applying moderate perturbations (neither very small nor very large) to the training schedule.

Results for oil (red) and water (blue) production rates for each well as a function of time are shown in Figure 3. Results are presented both for the full-order model (these results were generated using GPRS and are shown as solid curves) and for the TPWL model (symbols). For the TPWL model, \( l = 408 \). Note that, at early times, all wells produce only oil, though at later times water ‘breakthrough’ has occurred. It is clear that the TPWL model performs reasonably well for this case, and that it is able to capture water breakthrough accurately. Results for water injection rates are shown in Figure 4. TPWL is quite accurate for modeling this quantity. Although mismatch is noticeable in some of the TPWL results (e.g., for water rate in well P1), these discrepancies will generally decrease as the difference between the training and test BHP schedules decreases. Conversely, error will tend to increase as the difference between training and test schedules increases.
Figure 2: Training and test case BHP schedules for example problem.

Figure 3: Oil and water production rates for test case (Model 1).
Figure 4: Water injection rates for test case.

The computational requirements for the TPWL model are very small compared to those for the full-order (GPRS) simulation. For this case, the GPRS model required 667 seconds of computation, while the TPWL model required only 0.57 seconds, which gives a runtime speedup of more than a factor of 1,000. Furthermore, the runtime for TPWL depends only on the dimension of the reduced-order model, which means that for larger problems, the speedup could potentially be even greater. TPWL does, however, require preprocessing (overhead) computations including data loading, state reduction and stabilization, which are equivalent to about 0.5-2 full-order simulations for this case. Thus it would not make sense to use TPWL if only a few simulations were to be performed. For optimization problems requiring many model evaluations, however, the TPWL representation can be used very effectively as a surrogate, as we now describe.

4. Use of TPWL for Production Optimization

In this section we first discuss the generalized pattern search algorithm used for optimization, as well as our treatment of nonlinear constraints. We then describe the use of TPWL for production optimization, including the retraining of the model as the optimization proceeds. We next present optimization results for two example cases. We note that TPWL was used previously as a surrogate in gradient-based optimizations (with gradients computed numerically) in [6]. That work did not consider nonlinear constraints or systematic retraining, both of which are applied here.

4.1. Direct Search Optimization with TPWL

Surrogate modeling is widely used for simulation-based optimization when the full-order (high-fidelity) model is computationally expensive to evaluate. A surrogate model should be computationally inexpensive and at least locally accurate. TPWL appears to be well suited for use as a surrogate as it is able to provide a reasonable approximation of the true solution within a reasonably sized neighborhood around the training case.

The direct search method used here is generalized pattern search (GPS). GPS computes a sequence of points that approach an optimal point. The algorithm applies polling, which entails the evaluation of solutions defined by a stencil (aligned with the coordinates) in the search space. The central point of the stencil is the current (best) solution. If an improvement in the cost function is found, the stencil is shifted such that it is centered on the improved point. If an improved solution is not found, the stencil size is decreased. See [11] for more detail on GPS.

In Case 2 below we include nonlinear constraints in the optimization. To handle these constraints, GPS with an incremental penalty function is used. In this method, a modified objective function, which is a
weighted combination of the original objective function and a penalty term that quantifies the violation of the constraint, is defined. The weighting of the penalty term is increased incrementally (a sequence of subproblems is solved) until an optimized solution satisfying the constraint is achieved.

Our approach for incorporating TPWL into GPS is depicted in Figure 5. We start by performing a training simulation with well BHPs defined by the initial guess. The states and Jacobian matrices are saved and the stabilized TPWL model is constructed as described in Section 2. Then, the GPS optimization is started using the TPWL surrogate for function evaluations. After a specified number of function evaluations are performed, GPS is paused and a training simulation is run at the current best point (the specified number of function evaluations can vary during the course of the optimization). TPWL is then retrained at this point and GPS is resumed. It occasionally happens that, upon retraining, the objective function of the current point, evaluated using the full-order model, is suboptimal relative to that of the previous full-order solution. This inconsistency can occur when the TPWL solution loses accuracy because it is too far from the most recent training case. When this problem is detected, we restart the search from the previous retraining point and reduce the number of function evaluations until the next retraining. The size of the GPS mesh may also be reduced. We note that it should be possible to incorporate more sophisticated criteria, possibly based on mass balance errors in the TPWL model (which are straightforward to compute), for retraining. Such procedures will be considered in future work.

![Figure 5: Flowchart for generalized pattern search with TPWL.](image)

4.2. Optimization Results

4.2.1. Production Optimization: Case 1

We first optimize a small problem to enable comparison of results using the surrogate procedure to those using the full-order reference solution. The reservoir model for this case, referred to as Model 2, comprises the first four layers of Model 1 (Model 1 is shown in Figure 1). Model 2 contains 4,800 grid blocks. The rock and fluid properties and well locations are the same as for Model 1. In this case, we optimize the production well BHPs to maximize net present value (NPV) over five years (1,800 days) of production. The BHP of each well is changed every 200 days. Thus there are nine control variables for each producer, giving a total of 36 control variables. Injection well BHPs are set to 6,000 psi for the entire simulation. The oil price is specified to be $80/bbl while the cost of water produced and injected are $36/bbl and $18/bbl, respectively. Water prices are set to be artificially high to limit the use of water. The bounds for the production well BHPs are 1,000 psi and 3,000 psi. Initially, the BHPs for the four production wells are set to 1,500 psi for the entire production period.

The evolution of NPV with the number of simulations is shown in Figure 6 and summarized in Table 1. In the figure, the red curve represents the optimization results using the full-order simulation model while the blue curve presents results using the TPWL model. The circles indicate points where the TPWL surrogate model was retrained. It is evident that, using only 15 full-order training simulations, the TPWL guided optimization provides essentially the same result as was achieved using the full-order model.
simulations. The TPWL overhead in this case requires the equivalent of about 10 training simulations. Thus the overall speedup for this example is about a factor of 100. Because our TPWL implementation is currently in Matlab, we expect that these speedups could be further improved through a careful C++ implementation.

The optimization results for Case 1 are shown in Table 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Initial NPV ($10^6$)</th>
<th>Final NPV($10^6$)</th>
<th># of full simulations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-order GPS</td>
<td>49.9</td>
<td>170.1</td>
<td>2500</td>
</tr>
<tr>
<td>TPWL guided GPS</td>
<td>49.9</td>
<td>169.0</td>
<td>15</td>
</tr>
</tbody>
</table>

4.2.2. Production Optimization: Case 2

We now consider a larger reservoir model, and therefore do not present a comparison with full-order optimization results. The reservoir model in this case is the full Model 1, shown in Figure 1, which contains 20,400 grid blocks. In this example we optimize the BHPs for the four production wells and the two injection wells. The objective function is again NPV over five years of production. The well controls are again changed every 200 days, so there will be a total of 54 control variables. The bounds for injector BHPs are 5,500 psi and 7,500 psi, while those for producers are again 1,000 psi and 3,000 psi. A nonlinear constraint is also imposed on the optimization. This constraint requires that the water cut (fraction of water in the produced fluid) for all producers is less than 50% at all times.

The optimization results for this case are shown in Figure 7. The blue curve indicates NPV and the green curve shows the constraint violation. At early iterations, NPV is improved and the water cut constraint is satisfied, but at later iterations the constraint is violated. As the penalty weight increases, the constraint violation decreases. Finally, after nearly 4,000 function evaluations, a feasible solution is obtained with an NPV that is 34% greater than that of the initial guess. The circles and stars in Figure 7 indicate where retraining of the TPWL model is performed. A total of 12 full-order training simulations are required. The TPWL overhead for this case is equivalent to around 8-10 full-order simulations. Thus the speedup relative to running GPS with the full-order model is around a factor of 200. We note that, because several days of computation would be required, we have not performed optimizations using the full-order model for this case.
5. Concluding Remarks
In this paper we demonstrated the use of surrogate models based on trajectory piecewise linearization within a direct search optimization framework. The TPWL approach uses saved states and Jacobian matrices from training simulations performed using the full-order flow model to represent solutions efficiently in optimization runs. Retraining is applied to update the TPWL model as the optimization proceeds. Optimization results for two example cases illustrated the applicability of TPWL as a surrogate within the generalized pattern search algorithm. Future work should be directed toward further developing the TPWL procedure for more general subsurface flow systems and on formalizing the determination of when retraining should be performed.

Acknowledgements
We are grateful to the industry sponsors of the Stanford Reservoir Simulation (SUPRI-B) and Smart Fields Consortia for partial funding of this work. We also thank Marco Cardoso for providing the original TPWL code, Jon Sætrom for his contributions to the development of the enhanced TPWL procedure, and Obi Isebor for discussions regarding the use of generalized pattern search.

References


