Enhanced linearized reduced-order models for subsurface flow simulation

J. He¹, J. Sætrom², L.J. Durlofsky¹

¹Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, U.S.A.
²Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, NO-7034, Norway

Abstract

Trajectory piecewise linearization (TPWL) represents a promising approach for constructing reduced-order models. Using TPWL, new solutions are represented in terms of expansions around previously simulated (and saved) solutions. High degrees of efficiency are achieved when the representation is projected into a low-dimensional space using a basis constructed by proper orthogonal decomposition of snapshots generated in a training run. In recent work, a TPWL procedure applicable for two-phase subsurface flow problems was presented. The method was shown to perform well for cases with no density differences between phases, though accuracy and robustness were found to degrade when there were substantial differences in phase densities. In this work, these limitations are shown to be related to model accuracy at key locations and model stability. Enhancements addressing both of these issues are introduced. A new TPWL procedure, referred to as local resolution TPWL, enables key grid blocks (such as those containing injection or production wells) to be represented at full resolution; i.e., these blocks are not projected into the low-dimensional space. This leads to high accuracy at selected locations, and will be shown to improve the accuracy of important simulation quantities such as injection and production rates. Next, two techniques for enhancing the stability of the TPWL model are presented. The first approach involves a basis optimization procedure in which the number of columns in the basis matrix is determined to minimize the spectral radius of an appropriately defined amplification matrix. The second procedure incorporates a basis matrix constructed using snapshots from a simulation with equal phase densities. Both approaches are compatible with the local resolution procedure. Results for a series of test cases demonstrate the accuracy and stability provided by the new treatments. Finally, the TPWL model is used as a surrogate in a direct search optimization algorithm, and comparison with results using the full-order model demonstrate the efficacy of the enhanced TPWL procedures for this application.

Key words: trajectory piecewise linearization, TPWL, reservoir simulation, subsurface flow, two-phase flow, optimization, model order reduction, reduced-order model, surrogate model, proper orthogonal decomposition

*Principal corresponding author
Email addresses: jincong@stanford.edu (J. He¹), jsaetrom@stanford.edu (J. Sætrom²), lou@stanford.edu (L.J. Durlofsky¹)

Preprint submitted to Journal of Computational Physics October 12, 2010
1. Introduction

Optimization and uncertainty quantification are essential components in many model-based design procedures. The associated computations, which typically require large numbers of simulations, can be extremely time-consuming if highly resolved models are used. This may be the case even if parallel processing is applied, as multiobjective optimization and optimization under uncertainty, in which simulations are performed over a large number of models, can quickly occupy multiple cores.

The use of reduced-order models provides a means for accelerating these simulations. Our interest here is in subsurface flow modeling, which includes simulation of oil reservoirs, aquifers, and carbon sequestration operations. Several reduced-order modeling procedures based on proper orthogonal decomposition have been previously applied within this context; see, e.g., [1, 2, 3, 4]. For the nonlinear problems associated with oil reservoir simulation, the speedups achieved by these procedures were, however, relatively modest, at most about a factor of 10.

Trajectory piecewise linearization (TPWL) is a promising approach for model-order reduction that can potentially provide much larger speedups. TPWL represents new solutions of the governing equations in terms of linear expansions around previously simulated (saved) states. This requires that we perform one or more training simulations, from which the states and converged Jacobian matrices at each time step are saved. High degrees of efficiency are achieved by projecting the saved states and matrices into a low-dimensional subspace. This projection can be accomplished in different ways. In the implementation discussed in this paper, the projection matrix is constructed by proper orthogonal decomposition (POD) of the saved states. The TPWL approach was first introduced by Rewienski and White [5, 6] for the modeling of nonlinear circuits and micromachined devices. Since then it has been applied in a number of disciplines including computational fluid dynamics [7], nonlinear heat-transfer modeling [8] and electromechanical systems [9, 10].

In recent work Cardoso and Durlofsky [11] applied trajectory piecewise linearization (TPWL) procedures for oil reservoir simulation. Systems involving two fluid components and two phases – oil and water – were considered. For test runs involving equal density fluids, TPWL results were shown to be in close agreement with reference (full-order) simulations for control schedules that were within the general range of those used in the training runs. Runtime speedups of a factor of 200-1000 were observed for the examples considered. For cases in which the fluid densities differed considerably, however, instabilities were observed in some runs. This can lead to inaccuracy or, in some instances, to the blowup of the TPWL solution. This is of concern as oil and water phases often display density differences in practical cases.

Our goal in this work is to enhance the TPWL procedure presented in [11] to address the limitations noted above. We proceed in two important directions. First, we introduce a localization treatment in which key grid blocks, such as those containing injection or production wells, are represented at full resolution; i.e., the states in these blocks are not projected into the low-dimensional subspace. A missing point estimation procedure [12] is used to determine which grid blocks (in addition to well blocks) to represent explicitly. This localization will be shown to improve the accuracy of the overall TPWL representation and to have relatively little impact on run times (assuming the number of resolved blocks is not too large). The second enhancement is the use of stabilization procedures. Two such approaches are investigated. In one approach, a stabilized basis is determined in a preprocessing step in which various combinations of basis vectors are considered, with the goal of minimizing the spectral radii of the amplification matrices that appear in the TPWL representation. In the other stabilization approach, the POD basis
is determined using simulations involving equal density fluids. Accuracy is restored through use of the localization treatment. Both procedures will be shown to greatly improve TPWL stability and performance.

The stability of TPWL models has been studied previously and our development here builds on earlier work. Rather than use the traditional Galerkin projection, Bond and Daniel [13, 14, 15] computed a left projection matrix that stabilizes the system. This new left projection matrix is based on Lyapunov stability theory. However, if only stability is considered, the accuracy of the reduced-order model can degrade considerably. Thus, in [14], a stabilization procedure was proposed where the left projection matrix is constrained by Lyapunov theory to guarantee stability and the difference between the basis matrix and this new left projection matrix is optimized to recover accuracy. Bui-Thanh et al. [16, 17] formulated a goal-oriented, model-constrained optimization problem to determine the optimal basis under Galerkin projection. This approach was also shown to improve stability. This method differs from that of [14] in that it retains the Galerkin projection and optimizes both the basis matrix and the left projection matrix. However, when both accuracy and stability are considered, both of these approaches can lead to high computational demands.

This paper proceeds as follows. In Section 2, the equations for the subsurface flow of oil and water are presented, followed by a brief description of the finite volume approach used for their solution. The POD-based TPWL representation is then described. In Section 3 we present the local resolution approach in which key grid blocks are resolved explicitly. The enhanced accuracy provided by this treatment is illustrated with an example. The two approaches for stabilizing the TPWL representation are presented in Section 4. The impact of the local resolution and stabilization procedures is demonstrated through two examples, both of which contain $O(10^5)$ grid blocks, in Section 5. Next, in Section 6, the enhanced TPWL method is combined with a generalized pattern search (GPS) optimization technique and applied to a production optimization problem. Finally, in Section 7, we present conclusions and suggestions for future work.

2. Problem Formulation

The governing equations for oil-water flow and the basic TPWL formulation were presented in detail in [11]. For the sake of completeness, an abbreviated description is also included here.

2.1. Oil-Water Flow Equations

The equations governing the two-phase flow of oil and water in porous formations are derived by combining expressions for mass conservation with Darcy’s law. Using the subscript $j$ to designate component/phase ($j = o$ for oil and $w$ for water), these equations can be written as:

$$\frac{\partial}{\partial t} \left( \phi \rho_j S_j \right) - \nabla \cdot \left[ \rho_j \lambda_j k \left( \nabla p_j - \rho_j g \nabla D \right) \right] + q_j^w = 0,$$

where $k$ is the (diagonal) absolute permeability tensor, $\lambda_j = k_j / \mu_j$ is the phase mobility, with $k_j$ the relative permeability to phase $j$ and $\mu_j$ the phase viscosity, $\rho_j$ is phase pressure, $\rho_j$ is the phase density, $g$ is gravitational acceleration, $D$ is depth, $t$ is time, $\phi$ is porosity, $S_j$ is saturation and $q_j^w$ is the source/sink term. Eq. 1 is written slightly differently here than in [11]. Specifically, the source term $q_j^w$ here differs by a constant factor of $\rho_j^0$ (where $\rho_j^0$ is the reference density of phase $j$) from that in [11], and the definition of $\lambda_j$ differs by a factor of $\rho_j / \rho_j^0$. The general oil-water model is completed by enforcing the saturation constraint ($S_o + S_w = 1$) and by specifying a
capillary pressure relationship \( p_c(S_w) = p_o - p_w \). Eq. 1 is nonlinear since functions of unknowns (e.g., \( k_r(S_j) \)) multiply unknowns.

We take \( p_o \) and \( S_w \) to be primary variables (\( p_w \) and \( S_o \) can be immediately computed once these are known). Eq. 1 is solved numerically using a fully-implicit finite volume procedure. Discretized forms for all terms are discussed in [11, 18]. Basically, the flow from block to block is given by the interface transmissibility multiplied by the difference in block pressures, the accumulation term is handled using a first-order implicit (backward Euler) method, and the source term is treated using a well index representation, in which well rates are expressed in terms of an appropriately defined transmissibility times the pressure difference between the well block and the well. Using these representations, and defining \( x = [p_o, S_w] \) as the state vector and \( u \) as the well controls (in this case the wells are controlled by specifying bottom hole pressure or BHP), the discrete system for the fully-implicit formulation can be written as:

\[
g(x^{n+1}, x^n, u^{n+1}) = A(x^{n+1}, x^n) + F(x^{n+1}) + Q(x^{n+1}, u^{n+1}).
\]  

Here \( g \) is the residual we seek to drive to zero, \( n \) and \( n + 1 \) designate time level, and \( A, F \) and \( Q \) are the discretized accumulation, flux and source/sink terms, respectively.

Typically, a full-order reservoir simulator is used to solve Eq. 2. Newton’s method, with the Jacobian matrix given by \( \partial g/\partial x \), is applied. This is computationally expensive because Eq. 2 can be highly nonlinear and practical models may contain on the order of \( 10^5 \) or \( 10^6 \) grid blocks. We now describe the application of the TPWL approach for the efficient solution of Eq. 2.

2.2. Solution of Discretized System using TPWL

The idea of trajectory piecewise linearization is to linearize the residual equation around states saved from previous (training) simulations. Here, at any given time, we linearize around a single point on the training trajectory. Methods that involve weighted linearizations around multiple points also exist (e.g., [6]), though optimal weights can be difficult to determine and improper weighting can lead to stability problems.

Given the current state \( x^n \), we designate the closest saved state encountered during the training run as \( x^* \). To determine \( x^{n+1} \), we linearize Eq. 2 around the state \((x^{n+1}, x^*, u^{n+1})\). This gives

\[
g^{n+1} = g^* + \frac{\partial g}{\partial x} (x^{n+1} - x^*) + \frac{\partial g}{\partial u} (u^{n+1} - u^*),
\]

where \( g^n = g(x^n, x^*, u^n) \) and \( g^* = g(x^*, x^*, u^*) \). Here \( g^{n+1} = 0 \) because it is the residual of the training simulation. Defining the Jacobian matrix as the derivative of the residual with respect to the state,

\[
J^{n+1} = \frac{\partial g^{n+1}}{\partial x^{n+1}},
\]

and using the fact that, upon solution \( g^{n+1} = 0 \), Eq. 3 can be expressed as

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

Given \( x^n \) and saved information, Eq. 5 allows us to linearly compute \( x^{n+1} \); i.e., no iteration is required. This equation is, however, still in a high-dimensional space.
In order to reduce the dimension of Eq. 5, we employ linear order reduction. This entails representing the state $\mathbf{x}$ in terms of a reduced state $\mathbf{z}$ and a basis matrix $\Phi$ using

\[
\mathbf{x} \approx \Phi \mathbf{z}.
\] (6)

There are many ways to construct $\Phi$ including optimal Hankel model [19, 20], balanced truncation [21], Krylov subspace methods [22], and proper orthogonal decomposition (POD) [23]. Within the context of TPWL, previous researchers have used Krylov subspace methods (e.g., [5, 9, 8]) and POD (e.g., [7]). The POD method was applied successfully for our problem in [11] and will be used again here.

In POD, the high-dimensional space is represented by a set of orthogonal basis vectors, which are the singular vectors of a snapshot matrix $\mathbf{X}$. Each column of $\mathbf{X}$ is simply the pressure or saturation state at a particular time step saved from a training simulation. POD is optimal in the sense that it minimizes the mean squared reconstruction error for the snapshots [24]. Therefore, it is reasonable to assume that POD can represent the states of other (test) simulations if these states are somewhat similar to those encountered during training runs. In this work, consistent with [2, 11, 25], POD is used to reduce pressure and saturation separately. As discussed in [11], to generate the pressure basis matrix we use oil potential $\varphi_o$ ($\varphi_o = p_o/\rho_o - gD$) rather than pressure. This was found to improve TPWL stability in [11].

If we denote the number of simulation grid blocks as $N_c$ and the number of reduced pressure and saturation basis vectors as $l_p$ and $l_s$ respectively, the dimension of the problem can be reduced from $2N_c$ to $l_p + l_s$. This is accomplished by expressing $\mathbf{x}$ in Eq. 5 using Eq. 6 and by premultiplying both sides of Eq. 5 by $\Phi^T$ (also called Galerkin projection [26]), which gives

\[
\mathbf{J}_p^{i+1} (\mathbf{z}^{n+1} - \mathbf{z}^{i+1}) = - \left[ \left( \frac{\partial A^{i+1}}{\partial \mathbf{x}} \right)_r (\mathbf{z}^n - \mathbf{z}) + \left( \frac{\partial Q^{i+1}}{\partial \mathbf{u}} \right)_r (\mathbf{u}^{n+1} - \mathbf{u}^{i+1}) \right],
\] (7)

where

\[
\mathbf{J}_p^{i+1} = \Phi^T \mathbf{J}_p^{i+1} \Phi, \quad \left( \frac{\partial A^{i+1}}{\partial \mathbf{x}} \right)_r = \Phi^T \left( \frac{\partial A^{i+1}}{\partial \mathbf{x}} \right)_r \Phi, \quad \left( \frac{\partial Q^{i+1}}{\partial \mathbf{u}} \right)_r = \Phi^T \left( \frac{\partial Q^{i+1}}{\partial \mathbf{u}} \right)_r.\] (8)

Rearranging Eq. 7 we have

\[
\mathbf{z}^{n+1} = \mathbf{z}^{i+1} - \left( \mathbf{J}_p^{i+1} \right)^{-1} \left[ \left( \frac{\partial A^{i+1}}{\partial \mathbf{x}} \right)_r (\mathbf{z}^n - \mathbf{z}) + \left( \frac{\partial Q^{i+1}}{\partial \mathbf{u}} \right)_r (\mathbf{u}^{n+1} - \mathbf{u}^{i+1}) \right].
\] (9)

Eq. 9 can be solved efficiently for two reasons. First, as a result of linearization, the new reduced state $\mathbf{z}^{n+1}$ can be calculated directly from $\mathbf{z}^n$ without any iteration. Second, all of the terms in Eq. 9 are in low-dimensional space, which means that the matrix operations are fast to compute. Specifically, the evaluation of Eq. 9 basically involves only two matrix-vector products and four vector additions in the reduced space, which can theoretically be performed in a fraction of a second for typical reduced-space dimension (~100-500) and number of time steps (~300). Therefore, most of the total TPWL computation time is spent on training simulations and projections, which only need to be done once during preprocessing.

This completes our description of the basic TPWL formulation. See [11] for additional details and algorithms and a discussion of some implementation issues. In the following section we will present an application example of the basic TPWL method and discuss some of its limitations, which we will address in subsequent sections.
2.3. Application Example: Reservoir Model 1

We now apply the basic TPWL procedure described in Section 2.2 to a reservoir simulation model. The simulation model, shown in Figure 1, is part of the so-called Stanford VI geological model developed by [27]. The model represents a fluvial system with high-permeability channels embedded in a low-permeability background region. The portion of the model considered here contains 20,400 grid blocks (with \( n_x = 30, n_y = 40, n_z = 17 \), where \( n_x, n_y \) and \( n_z \) indicate the number of blocks in each coordinate direction). The dimension of the full-order problem is 40,800 (pressure and saturation unknowns in each grid block). In this model the fluid and rock compressibility and the capillary pressure between the two phases are neglected. We specify oil and water viscosities as \( \mu_o = 3 \, \text{cp}, \mu_w = 0.5 \, \text{cp} \). The fluids are here specified to have equal densities (\( \rho_o = \rho_w = 55 \, \text{lb/ft}^3 \)). The relative permeability relationships are given by

\[
k_{ro}(S_w) = k^0_{ro} \left( \frac{1 - S_w - S_{or}}{1 - S_{wr} - S_{or}} \right)^a, \quad k_{rw}(S_w) = k^0_{rw} \left( \frac{S_w - S_{wr}}{1 - S_{wr} - S_{or}} \right)^b. \tag{10}
\]

We set \( k^0_{ro} = k^0_{rw} = 1, S_{wr} = S_{or} = 0.2 \) and \( a = b = 2 \), as was used in [11].

To systematically investigate the performance of TPWL, we define test cases based on the level of perturbation from the training run. Shown on the left in Figure 2 is the bottom hole pressure (BHP) control schedule for the production wells in the training simulation, which we designate as \( u_{\text{training}} \). This schedule is generated randomly, with BHPs between 1000 psi and 3000 psi, and is updated every 200 days. On the right is a different (target) schedule also generated randomly with the same update frequency as the training schedule, though this schedule varies between 2000 psi and 4000 psi. Test cases are specified as a weighted combination of these two schedules; i.e.,

\[
u_{\text{test}} = (1 - \alpha) u_{\text{training}} + \alpha u_{\text{target}}, \tag{11}
\]

where \( \alpha \) represents the “distance” of the test controls from the training controls. When \( \alpha \) is near zero, test cases are close to the training run and TPWL would be expected to provide accurate results. As \( \alpha \) increases toward 1, test cases are further from the training run and larger errors are
For this example, the BHPs of the two injection wells are held constant at 6000 psi throughout the simulations.

In this work, errors are quantified in terms of the mismatch of the production rates (for both oil and water) and water injection rates between the full-order solution ($Q_{\text{full}}$) and TPWL simulations ($Q_{\text{tpwl}}$). For example, for oil production rate, the error for the $j$th well ($E_{io}^j$) is calculated as:

$$E_{io}^j = \frac{\int_0^T |Q_{io,full}^j - Q_{io,tpwl}^j| \, dt}{\int_0^T Q_{io,full}^j \, dt},$$

(12)

where subscript $o$ designates oil and $T$ is the total simulation time. The overall average error of the oil production rates, designated $E_{io}$, is computing by averaging $E_{io}^j$ over all wells:

$$E_{io} = \frac{1}{n_{pw}} \sum_{j=1}^{n_w} E_{io}^j,$$

(13)

where $n_{pw}$ is the total number of production wells. Similar expressions are used to compute average water production error and water injection error ($E_{iw}$ and $E_i$, respectively).

We now compare full-order simulation results, generated using Stanford’s General Purpose Research Simulator (GPRS) [28, 29], with TPWL results. For the TPWL model we use $l_p = 40$ and $l_s = 60$. Results for $\alpha = 1.0$ for oil and water production rates are shown in Figure 3, while results for water injection rates appear in Figure 4. The errors in the TPWL solution for other levels of perturbation are shown in Table 1. Discrepancies are evident between the full-order and TPWL results, especially for water production and water injection rates. In Section 3 these discrepancies are discussed in detail and a local resolution method is proposed to improve the accuracy. However, it should be noted here that this inaccuracy results from the use of $\alpha = 1.0$, which means that the test case is quite different from the training run. As shown in Table 1, the use of smaller values of $\alpha$ leads to TPWL results that are much more accurate.

3. Local Resolution TPWL

As shown in Section 2.3, the basic TPWL method can display inaccuracy for cases with large perturbations. In this chapter we will first analyze the problem of underfitting and overfitting.
Figure 3: Model 1 production rates for $\alpha = 1.0$ using basic TPWL ($l_p = 40, l_s = 60$).

Figure 4: Model 1 injection rates for $\alpha = 1.0$ using basic TPWL ($l_p = 40, l_s = 60$).
We then propose the local resolution method to enhance the accuracy of the basic TPWL.

3.1. Underfitting and Overfitting

Since $\Phi$ is orthonormal, we have $z = \Phi^T x$. POD will by construction minimize the mean squared reconstruction error of the training snapshots, \((1/S)\sum_{i=1}^{S} |\Phi(\Phi^T x^i) - x^i|^2\), where $S$ is the number of snapshots [24]. Thus, if we consider TPWL solutions based on the full-order model (Eq. 5) and the reduced-order model (Eq. 9) for $u^n = u^i$, and if we take the number of basis vectors stored in $\Phi$ equal to the number of snapshots (and if the snapshots are based on pressure rather than potential), both approaches will reproduce the training states exactly (for Eq. 9 we need to apply $x = \Phi z$). However, this will not be the case for states from a new set of well controls $u^n \neq u^i$.

The POD dimension reduction technique is based on the assumption that most of the variability in the snapshot matrices can be represented using a limited number of orthonormal basis vectors $\Phi = [\phi_1, \ldots, \phi_l]$ [30]. We therefore expect that selecting too few basis vectors will potentially lead to the problem of model underfitting, which can result in large errors in the TPWL solution, both for $u^n = u^i$ and $u^n \neq u^i$. On the other hand, including too many basis vectors will potentially cause model overfitting [31], leading to large errors in the TPWL solution for $u^n \neq u^i$. This can occur because in the basis matrix obtained from POD, the basis vectors that correspond to smaller eigenvalues are more subject to noise in the training snapshots. Thus, including them in $\Phi$ can lead to a deterioration of the representation of $x$ stored in $z$.

These effects can be observed through an assessment of the sensitivity of the TPWL results to different numbers of basis vectors in $\Phi$. Errors in injection rate ($E_i$) for various $\alpha$ are presented in Table 2. The smallest error at each value of $\alpha$ is shown in bold. The other errors ($E_o$ and $E_w$) display generally similar behavior and are not shown. It is apparent from the table that selecting a small number of basis vectors ($l_p = 5, l_s = 5$) leads to large errors in the TPWL solution, particularly as $\alpha$ increases. Interestingly, however, error does not necessarily decrease monotonically as we include more basis vectors in $\Phi$. Consistent with this, the use of the largest $l_p$ and $l_s$ does not provide the most accurate results for water injection for any value of $\alpha$.

This is in part due to error introduced through overfitting. It is additionally because the POD reduction scheme focuses on the global reconstruction error, not the reconstruction error at the well blocks. Note also that the reason we do not observe a monotonic decrease in the error for the training schedule ($\alpha = 0$) is because, as noted earlier, we use oil potential rather than pressure snapshots to compute the pressure basis matrix. This results in improved accuracy in the TPWL solution for test schedules that differ from the training schedule [11], although it leads to an increase in the reconstruction error for the training schedule.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$E_o$</th>
<th>$E_w$</th>
<th>$E_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0039</td>
<td>0.0024</td>
<td>0.0121</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0047</td>
<td>0.0110</td>
<td>0.0156</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0071</td>
<td>0.0246</td>
<td>0.0209</td>
</tr>
<tr>
<td>0.6</td>
<td>0.0108</td>
<td>0.0441</td>
<td>0.0272</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0144</td>
<td>0.0733</td>
<td>0.0354</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0193</td>
<td>0.1266</td>
<td>0.0456</td>
</tr>
</tbody>
</table>
Table 2: Relative error in TPWL solutions for various test schedules for Model 1 with equal density. The notation TPWL(\(l_p, l_s\)) in this and subsequent tables denotes the numbers of pressure and saturation basis vectors retained in \(\Phi\). The smallest error at each value of \(\alpha\) is shown in bold.

<table>
<thead>
<tr>
<th>(\alpha = 0)</th>
<th>(\alpha = 0.2)</th>
<th>(\alpha = 0.4)</th>
<th>(\alpha = 0.6)</th>
<th>(\alpha = 0.8)</th>
<th>(\alpha = 1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPWL(5,5), (E_i)</td>
<td>0.0513</td>
<td>0.1245</td>
<td>0.1521</td>
<td>0.2078</td>
<td>0.2602</td>
</tr>
<tr>
<td>TPWL(20,30), (E_i)</td>
<td>0.0298</td>
<td>0.0355</td>
<td>0.0423</td>
<td>0.0494</td>
<td>0.0564</td>
</tr>
<tr>
<td>TPWL(40,60), (E_i)</td>
<td><strong>0.0121</strong></td>
<td><strong>0.0156</strong></td>
<td>0.0209</td>
<td>0.0272</td>
<td>0.0354</td>
</tr>
<tr>
<td>TPWL(70,90), (E_i)</td>
<td>0.0183</td>
<td>0.0177</td>
<td><strong>0.0187</strong></td>
<td><strong>0.0215</strong></td>
<td><strong>0.0281</strong></td>
</tr>
<tr>
<td>TPWL(120,120), (E_i)</td>
<td>0.0326</td>
<td>0.0332</td>
<td>0.0351</td>
<td>0.0367</td>
<td>0.0383</td>
</tr>
</tbody>
</table>

3.2. Description of Local Resolution Scheme

For our applications, we are particularly interested in the pressure and saturation at well locations because they directly affect injection and production rates, which are the key quantities needed for production optimization. To compute these quantities, we construct \(x_{n+1} = \Phi_w z_{n+1}\), where \(\Phi_w\) includes only the rows of \(\Phi\) corresponding to the grid blocks containing wells (subscript \(w\) here denotes well). Although we are especially interested in maintaining accuracy in \(\Phi_w\), the POD dimension reduction technique, by construction, minimizes the global reconstruction error of the training snapshots – it does not preferentially weight information at well locations. We can therefore expect reconstruction of the saturation and pressure at well locations to be suboptimal.

This effect is illustrated in Figure 5, where we plot the maximum reconstruction error (relative to the full-order solution) of the saturation and pressure at the injection and production wells, as a function of \(l\), for \(\alpha = 1.0\). From this figure we see that reconstruction of saturation at the production wells is problematic, and that increasing the number of basis vectors included in \(\Phi\) can lead to a clear increase in the reconstruction error.

It is thus evident that reconstruction error at well locations can have a large impact on the accuracy of TPWL results for production and injection rates. To eliminate reconstruction error at the well locations, we therefore propose a TPWL procedure in which selected key grid blocks are represented using the full-order (linearized) model. We let \(x_{LR}\) designate the full-order states for \(n_{LR}\) selected grid blocks and \(x_G\) the full-order states for the remaining grid blocks. Then, in place of Eq. 6, we write:

\[
\begin{bmatrix}
x_{LR} \\
x_G
\end{bmatrix}
\approx
\begin{bmatrix}
\Phi_{LR} & 0 \\
0 & \Phi_G
\end{bmatrix}
\begin{bmatrix}
z_{LR} \\
z_G
\end{bmatrix},
\]

where \(\Phi_{LR}\) is taken to be the identity matrix (thus \(z_{LR} = x_{LR}\)). This means that the grid blocks associated with \(x_{LR}\) are not subject to reduction, so high local resolution is maintained. Note that, although the local and global grid blocks are decoupled in Eq. 14, they do couple in the TPWL scheme defined in Eq. 9 through \((\mathbf{J}_i)^{-1}\).

In our implementation, the \(n_{LR}\) locally-resolved blocks include the well blocks and possibly additional blocks that are important for the flow solution. To determine these additional blocks, we apply the missing point estimation (MPE) procedure suggested by [12]. In this approach we retain blocks that have the largest impact on the condition number of \(\Phi^T \Phi\), which are blocks that strongly affect the flow solution. A computationally efficient algorithm for MPE is described in [4]. In our implementation, locally resolved blocks are determined separately for the saturation.
Figure 5: Log plot of the maximum reconstruction error for saturation and pressure at production and injection wells for Model 1 with $\alpha = 1.0$. 
and pressure portions of $\Phi$. The final set of $n_{LR}$ locally-resolved blocks is the union of these two sets of blocks.

The local resolution method provides flexibility for improving the accuracy of the TPWL representation. As is evident in Figure 5 and Table 2, this cannot necessarily be accomplished by adding more basis vectors. Even if it could, the maximum number of basis vectors is limited by the number of snapshots. The local resolution method does not have these limitations, and the TPWL model thus defined approaches the full-order model as $n_{LR}$ is increased. Thus, the local resolution method enables us to achieve a balance between accuracy and efficiency.

3.3. Numerical Results using TPWL(LR)

We now apply the local resolution TPWL scheme, designated TPWL(LR), to the case considered earlier (Model 1 with $\alpha = 1.0$). For these runs we use $l_p = 40$, $l_s = 60$ and $n_{LR} = 26$ (local resolution only at well blocks), which corresponds to a total of 152 unknowns. Results are shown in Figures 6 and 7. Comparing these results with those from the basic TPWL scheme using $l_p = 40$, $l_s = 60$ (Figures 3 and 4), we see that by eliminating the reconstruction error at wells, there is a large improvement in the TPWL results, particularly for the water injection rates.

Results for flow rate errors for a range of $\alpha$ are shown in Table 3. Errors are shown for TPWL(LR) (with $l_p = 40$, $l_s = 60$, $n_{LR} = 26$) as well as for the basic method for two different sets of $(l_p, l_s)$. The smallest errors for each value of $\alpha$ are shown in bold. TPWL(LR) is the most accurate in all cases, and for some quantities a large reduction in error is observed. These results clearly demonstrate the efficacy of the use of local resolution within a TPWL model.

![Figure 6: Model 1 production rates for $\alpha = 1.0$ using local resolution TPWL ($l_p = 40$, $l_s = 60$, $n_{LR} = 26$).](image-url)
Figure 7: Model 1 injection rates for $\alpha = 1.0$ using local resolution TPWL ($l_p = 40, l_s = 60, n_{LR} = 26$).

Table 3: Relative error in TPWL solutions for various test schedules for Model 1. Basic TPWL and local resolution TPWL with $l_p = 40, l_s = 60, n_{LR} = 26$ are compared. The smallest error at each value of $\alpha$ is shown in bold.

<table>
<thead>
<tr>
<th></th>
<th>$\alpha = 0$</th>
<th>$\alpha = 0.2$</th>
<th>$\alpha = 0.4$</th>
<th>$\alpha = 0.6$</th>
<th>$\alpha = 0.8$</th>
<th>$\alpha = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPWL(40,60), $E_o$</td>
<td>0.0039</td>
<td>0.0047</td>
<td>0.0071</td>
<td>0.0108</td>
<td>0.0144</td>
<td>0.0193</td>
</tr>
<tr>
<td>TPWL(70,90), $E_o$</td>
<td>0.0099</td>
<td>0.0113</td>
<td>0.0154</td>
<td>0.0212</td>
<td>0.0281</td>
<td>0.0357</td>
</tr>
<tr>
<td>TPWL(LR), $E_o$</td>
<td><strong>0.0002</strong></td>
<td><strong>0.0020</strong></td>
<td><strong>0.0041</strong></td>
<td><strong>0.0065</strong></td>
<td><strong>0.0089</strong></td>
<td><strong>0.0121</strong></td>
</tr>
<tr>
<td>TPWL(40,60), $E_w$</td>
<td>0.0024</td>
<td>0.0110</td>
<td>0.0246</td>
<td>0.0441</td>
<td>0.0733</td>
<td>0.1266</td>
</tr>
<tr>
<td>TPWL(70,90), $E_w$</td>
<td>0.0128</td>
<td>0.0229</td>
<td>0.0529</td>
<td>0.1106</td>
<td>0.2135</td>
<td>0.4126</td>
</tr>
<tr>
<td>TPWL(LR), $E_w$</td>
<td><strong>0.0002</strong></td>
<td><strong>0.0090</strong></td>
<td><strong>0.0193</strong></td>
<td><strong>0.0350</strong></td>
<td><strong>0.0547</strong></td>
<td><strong>0.1032</strong></td>
</tr>
<tr>
<td>TPWL(40,60), $E_i$</td>
<td>0.0121</td>
<td>0.0156</td>
<td>0.0209</td>
<td>0.0272</td>
<td>0.0354</td>
<td>0.0456</td>
</tr>
<tr>
<td>TPWL(70,90), $E_i$</td>
<td>0.0183</td>
<td>0.0177</td>
<td>0.0187</td>
<td>0.0215</td>
<td>0.0281</td>
<td>0.0381</td>
</tr>
<tr>
<td>TPWL(LR), $E_i$</td>
<td><strong>0.0003</strong></td>
<td><strong>0.0013</strong></td>
<td><strong>0.0025</strong></td>
<td><strong>0.0037</strong></td>
<td><strong>0.0049</strong></td>
<td><strong>0.0067</strong></td>
</tr>
</tbody>
</table>
4. Stability of TPWL Models

As demonstrated through extensive examples in [11, 25], the TPWL procedure can provide reasonable accuracy and robustness for cases with equal phase densities. It was, however, also reported in [11] that the method can become unstable when large density differences between the two phases exist. In this section we will first show an example of this instability. We will then discuss stability criteria and present two methods for stabilizing TPWL.

4.1. Example Showing Instability of TPWL

We consider a model that is the same as that used in Section 3, except that now the densities for the two phases are different (here we set $\rho_o = 45$ lb/ft$^3$ and $\rho_w = 55$ lb/ft$^3$). Figure 8 (upper) displays the oil production rate for well P1 for $\alpha = 0.3$. Small spikes in the solution can be observed at around 500 days and 900 days. Shortly after 2000 days the solution becomes completely unstable and blows up. Results for all other wells display similar behaviors and are not shown here. Clearly, the method requires improvement if it is to be applied to problems of this type. We now consider the stability of the TPWL model.

4.2. Stability Analysis

We first consider the linearized full-order system, Eq. 5. This equation can be viewed as a piecewise linear discrete-time system,

$$x^{n+1} = M^{i+1}x^n + b^{i+1},$$

where

$$M^{i+1}x^n + b^{i+1},$$

is the solution at each step. The stability of the TPWL model can be analyzed by examining the spectral radius of the amplification matrix.
where $b$ is a vector that involves the source term and $M$ is the amplification matrix given by

$$M^{i+1} = -(J^{i+1})^{-1} \frac{\partial A^{i+1}}{\partial x^i}. \quad (16)$$

For a constant $M$, the system defined in Eq. 15 is stable if and only if the spectral radius of $M$ ($\gamma(M)$) is less than or equal to 1 \[32\]. Here, stability means that the error in $z^n$ will not be amplified in $z^{n+1}$.

As can be seen from Eq. 16, the amplification matrix $M^{i+1}$ of Eq. 15 is the same as that for the training simulation. Therefore, the full-order linearized system (Eq. 15) displays the same numerical stability properties as the original system. This is not the case, however, for the reduced-order model. In this case, the counterpart to Eq. 9 is

$$z^{n+1} = M^{i+1}_r z^n + b^{i+1}_r, \quad (17)$$

where the amplification matrix is given by

$$M^{i+1}_r = -(\Phi^T J^{i+1} \Phi)^{-1} \Phi^T \left( \frac{\partial A^{i+1}}{\partial x^i} \right) \Phi. \quad (18)$$

For general matrices $J^{i+1}$ and $\partial A^{i+1}/\partial x^i$, $\gamma(M^{i+1}_r)$ can be greater than 1 even when $\gamma(M^{i+1}) \leq 1$. In general, only special choices of $\Phi$ can maintain the stability of the system. We note that the spectral radius of $M^{i+1}_r$ affects the stability of the linearized system at the time step when it is used. An isolated unstable $M^{i+1}_r$ will amplify the error at a specific time step and may create a spike in the solution. The solution may still be able to recover if the perturbation is not too large and if subsequent $M^{i+1}_r$ are stable. However, if we have several consecutive time steps with unstable $M^{i+1}_r$, the error will amplify and the solution may blow up. Therefore, to ensure that the error does not accumulate over time, it is necessary to require the linearized system to be stable for all time steps.

Figure 8 (lower) presents the log of the spectral radius of the amplification matrix at each time step. Instability occurs for $\log \gamma > 0$. It is apparent that the spikes and eventual blowup of the solution correspond to values of $\log \gamma$ that are much larger than zero and thus unstable. The isolated unstable points at around 500 days and 900 days produce small spikes in the solution, while the consecutive unstable points starting at around 2200 days lead to solution blowup. Thus it is clear that the use of the TPWL method with a POD basis and Galerkin projection can result in instability for this problem.

The loss of stability for reduced-order models, especially those based on POD and Galerkin projection, has been studied previously and several methods to enhance stability have been proposed. Two basic types of stabilization procedures have been considered. The idea of the first set of methods is to compute a left projection matrix that stabilizes the system, rather than use $\Phi^T$ as in Galerkin projection. In [13, 14, 15], this new left projection matrix was based on Lyapunov stability theory. However, if only stability is considered, the accuracy of the reduced-order model can degrade considerably. If accuracy is also taken into account, a matrix optimization has to be solved to obtain an optimal left projection matrix, which is computationally expensive for large systems. The other group of methods focuses on finding a basis other than POD that can guarantee stability under Galerkin projection. In [17], a goal-oriented, model-constrained optimization problem was formulated to determine the optimal $\Phi$. However, the procedure involves calculation in the full-order space. Even when the basis matrix is parameterized by the snapshots, as in [17], the optimization can still be computationally demanding.
Neither of the approaches described above maintains both POD and Galerkin projection while stabilizing the result. We will present two relatively efficient ways to stabilize TPWL in the following sections. Both apply a POD basis and Galerkin projection.

4.3. Stabilization of TPWL using Optimized Basis

The first stabilization method is based on the observation that the stability of the reduced-order model is sensitive to \( l_p \) and \( l_s \), which define the number of columns in \( \Phi \). The dependence of \( M_r \) on \( \Phi \) is evident in Eq. 18.

Figure 9 depicts \( \log(\gamma(M_r)) \) for different values of \( l_p \) and \( l_s \) for two particular saved points, \( i = 70 \) and \( i = 100 \), for the problem with different phase densities defined in Section 4.1. In the figure, the lower bound of the color bar is the linear stability limit. Therefore the dark blue regions indicate combinations of \( l_p \) and \( l_s \) that give a stable linearized reduced system while other colors represent different levels of instability. It can be seen that the relation between stability and the number of basis vectors is somewhat random and that the use of more basis vectors does not necessarily lead to improved stability. This demonstrates that the widely used energy criterion (see, e.g., [11]) for selecting the number of basis vectors based on singular values may lead to stability problems. However, it can also be seen that for many of the \((l_p, l_s)\) combinations, the spectral radius is less than 1. This means that if these combinations are used to generate the basis, the resulting reduced system will be stable.

The idea of basis optimization is to define a range of \( l_p \) and \( l_s \) and to determine values (designated \( l_p^* \) and \( l_s^* \)) that minimize \( \gamma(M_r) \); i.e.,

\[
(l_p^*, l_s^*) = \arg \min_{l_p, l_s} \gamma(M_r).
\]  
(19)

This is accomplished using an exhaustive search over the allowable range of \( l_p \) and \( l_s \) (with prescribed increments in \( l_p \) and \( l_s \)). If we select different \( l_p \) and \( l_s \) for different time steps, which may be necessary in some cases, the reduced space changes in time. It is therefore necessary to map \( z \) in one reduced space to \( z \) in another reduced space. This is accomplished using

\[
z_{p} = \Phi_{p} \Phi_{s} z_{s},
\]  
(20)
where subscripts \( \alpha \) and \( \beta \) indicate the two reduced spaces. We will refer to this procedure as basis switching.

The optimized basis (OB) procedure is summarized in Algorithm 1. First, a range for \( l_p \) and \( l_s \) is specified. Then the spectral radius is calculated for selected combinations of \( l_p \) and \( l_s \). The combination that gives the smallest \( \gamma \) is selected. This method takes into account both accuracy and stability. Accuracy is controlled approximately by the search range, and stability is improved by choosing the optimal \((l_p, l_s)\) combinations.

### Algorithm 1: Selecting the Number of Basis Vectors to Achieve Stability.

**Input:** Training data obtained using GPRS. Search region for \( l_p \in [l_{p,\min}, l_{p,\max}] \) and \( l_s \in [l_{s,\min}, l_{s,\max}] \)

1. \([\Phi, \Lambda] = \text{SVD}(X)\)
2. Keep first \( l_{p,\max} + l_{s,\max} \) components
3. Compute \( J'_{r,\max} \) and \((\partial A^{t+1}/\partial x)_{r,\max}\) based on \( l_{p,\max} \) and \( l_{s,\max} \) for all time steps \( i \)
4. For each time step do
   5. For \( l_p = l_{p,\min} \) to \( l_{p,\max} \) do
      6. For \( l_s = l_{s,\min} \) to \( l_{s,\max} \) do
         7. Extract columns \( 1, \ldots, l_p, 1, \ldots, l_s \) of \( J'_{r,\max} \) and \((\partial A^{t+1}/\partial x)_{r,\max}\)
         8. Construct amplification matrix \( M'_r \)
         9. Calculate \( \gamma(M'_r) \)
   10. Select the optimal values \( l_p' \) and \( l_s' \) that minimize \( \gamma(M'_r) \)
   11. Construct the basis \( \Phi' \)

The optimization procedure is reasonably efficient because \( J'_{r} \) and \((\partial A^{t+1}/\partial x)_{r,\max} \) only need to be constructed once, for \( l_{p,\max} \) and \( l_{s,\max} \). Denoting these matrices as \( J'_{r,\max} \) and \((\partial A^{t+1}/\partial x)_{r,\max}\), \( J'_{r} \) and \((\partial A^{t+1}/\partial x)_{r,\max}\), for any other \((l_p', l_s')\) combination only need to be determined once during the preprocessing and will not add to the runtime of TPWL. Thus this optimization does not overly affect the efficiency of the TPWL model.

This method differs from the two types of methods developed previously in that it maintains the advantages of POD and the Galerkin projection. Specifically, POD provides optimal accuracy in representing the snapshots and the Galerkin projection is straightforward and efficient. It is, however, important to note that the algorithm does not guarantee stability. Nevertheless, it does lead to considerable improvements in stability and enables the solution of challenging problems with large differences in density between phases.

Figure 10 displays the results when applying this technique to the problem described in Section 4.1. In this case a single set of optimized \( l_p \) and \( l_s \) was used \((l_p = 45, l_s = 60)\); i.e., basis switching was not required. It is evident from Figure 10 that, after basis optimization, \( \gamma(M'_r) \) is close to 1, which means that TPWL is stabilized. The oil production for well P1 is seen to be in reasonable agreement with the reference full-order results, with the solution blowup after 2000 days eliminated (compare with Figure 8). We note that the use of local resolution will act to further improve the accuracy of the TPWL solution.

It is of interest to note that spikes in \( \gamma(M'_r) \) usually correspond to spikes in the condition
Figure 10: Upper: production rate for well P1 using TPWL with optimized basis for case with different phase densities (Model 1, $\alpha = 0.3, l_p = 45, l_s = 60$). Lower: log of spectral radius of amplification matrix.

number of $J_{c}^{-1}$ (or $J_{r}$). This is potentially of concern since $J_{c}^{-1}$ appears not only in $M_{r}$ but also in $b$, in Eq. 17. Thus, even when the system is stable, spikes in the condition number of $J_{c}^{-1}$ can still cause inaccuracy in the solution. In such cases, we may need to determine optimum ($l_p, l_s$) such that both $\gamma(M_r)$ and the condition number of $J_{c}^{-1}$ are minimized to assure solution stability and accuracy. This will be addressed in future work.

4.4. Stabilization of TPWL using Modified Basis

We now introduce a stabilization procedure in which $\Phi$ is derived from a problem with better stability characteristics than the target problem with $\Delta \rho \neq 0$. This approach can be motivated with reference to stability maps. Figure 11(a) shows the maximum spectral radius for all saved points as a function of $l_p$ and $l_s$ for Model 1 with $\rho_o = \rho_w$. Here we see that the TPWL scheme is unstable only for a limited number of combinations of $l_p$ and $l_s$. However, the stability map for the same reservoir model but with $\Delta \rho = 10$, shown in Figure 11(b), indicates only isolated regions where the TPWL model is stable. This motivates the construction of a model reduction scheme where the basis matrix is formed using snapshots generated from a reservoir model with $\Delta \rho = 0$. The states and Jacobian matrices used in the actual TPWL solution (Eq. 9) are still from the specific problem of interest (with $\Delta \rho \neq 0$). We will refer to this procedure as the equal density projection (EDP) scheme. A stability analysis of the EDP scheme for different values of $l_p$ and $l_s$ reveals that the regions where the TPWL solution is stable, seen in Figure 11(c), correspond to the stability regions in Figure 11(a). Thus the approach inherits the stability properties of the $\Delta \rho = 0$ TPWL model.
Because the EDP basis matrix is constructed using snapshots which are different from the states in the actual solution, the global reconstruction error for $z'$ will necessarily increase dramatically compared with the approaches considered above. However, combining the EDP scheme with local resolution TPWL effectively eliminates the reconstruction error at key locations such as well blocks.

Figure 12 shows the oil and water production rates for Model 1 with $\Delta \rho = 10$ and $\alpha = 0.3$. Here we apply both EDP and LR and set $l_p = 70$, $l_s = 100$ and $n_{LR} = 27$ in the TPWL(EDP+LR) solution. Note that the use of $l_p = 70$, $l_s = 100$ in the basic TPWL method leads to instability (see Figure 8). As is evident in Figure 12, the TPWL(EDP+LR) scheme, by contrast, is able to provide a stable and reasonably accurate solution relative to the full-order results. This is further illustrated in the results for water injection rates shown in Figure 13.

The reason why the basis matrix generated from snapshots of the corresponding equal density case ($\Phi_{ED}$) leads to better stability than the basis matrix generated from snapshots of the actual (different density) case appears to be related to the magnitudes of the components appearing in the two basis matrices. Referring to the basis from the simulation with different densities as $\Phi_{DD}$, we observe that $\Phi_{DD}$ tends to have more isolated extreme values than $\Phi_{ED}$. These appear to derive from the fact that changes between snapshots in simulations with different densities are more localized than in cases with equal densities. More specifically, in runs with $\Delta \rho \neq 0$, substantial changes in the solution from one time step to the next may occur in only a few grid blocks, with little change in the global solution. This behavior, which is less prominent in the equal density case, results in extreme values in $\Phi_{DD}$. This in turn can lead to high condition numbers in $J_r$ and solution instability. By contrast, $\Phi_{ED}$ contains more evenly distributed values and as a result provides better model stability.

This argument is also consistent with results from the random projection (RP) scheme [33]. In the RP scheme, the pressure and saturation snapshots are replaced with random vectors generated independently from a Gaussian probability distribution with zero mean and unit variance. The variance between snapshots is thus evenly distributed spatially. Orthonormalization of these matrices provides the basis matrix. Interestingly, this approach leads to a stable TPWL scheme for any combination of $l_p$ and $l_s$. In numerous tests, we achieved stable TPWL results for all of the reservoir models considered in this paper, including cases where the basic TPWL method exhibited instability. The accuracy of the RP scheme is, however, quite poor as it does not use any information from actual snapshots. The EDP scheme can be viewed as an enhanced or supervised version of the RP scheme in the sense that, by using a particular set of snapshots, it achieves stability at the cost of accuracy. As noted above, accuracy is recovered through use of local resolution. We note finally that the accuracy of the RP scheme can also be improved using local resolution, but $n_{LR}$ needs to be very large before adequate levels of accuracy are consistently achieved.

The two stabilization schemes proposed in this section have somewhat different features. The optimized basis method requires only one full-order simulation, though it does require basis optimization computations. In many cases it performs reasonably well without the use of local resolution. However, it has limited flexibility in the choice of $l_p$ and $l_s$ as they are determined based on stability properties. The EDP method, by contrast, requires two full-order simulations and it must be combined with local resolution to provide reasonable accuracy. It has more flexibility, however, in the choice of $l_p$ and $l_s$. Thus both methods have advantages and limitations, and further application and development of both approaches appears to be warranted.
Figure 11: Maximum value for the spectral radius of the amplification matrix, $M_{ir}$, as a function of number of basis vectors retained for Model 1 using (a) $\Delta \rho = 0$ with standard POD, (b) $\Delta \rho = 10$ with standard POD, and (c) $\Delta \rho = 10$ with EDP scheme.
Figure 12: Model 1 production rates for $\alpha = 0.3$ using TPWL(EDP+LR) ($l_p = 70, l_s = 100, n_{LR} = 27$).

Figure 13: Model 1 injection rates for $\alpha = 0.3$ using TPWL(EDP+LR) ($l_p = 70, l_s = 100, n_{LR} = 27$).
5. Application of TPWL to Realistic Problems

In this section we apply the new TPWL procedures to two realistic reservoir models. The models contain $O(10^5)$ grid blocks and the oil and water phases are of different densities. For these examples we combine the stabilization methods with the local resolution procedure, and present results for a range of perturbations $\alpha$. The basic TPWL procedure has difficulty providing stable solutions for these cases.

5.1. Model 2: Upper Six Layers of SPE 10

The geological model used here is shown in Figure 14. The model, referred to as Model 2, comprises the upper six layers of the so-called SPE 10 geological model, developed by Christie and Blunt [34]. The model contains 79,200 grid blocks (with $n_x = 60$, $n_y = 220$, $n_z = 6$). This model was also studied in [11], where it was applied for an example with equal phase densities, for which accurate TPWL results were reported. The problems that can arise using the basic TPWL procedure for cases with unequal densities were also illustrated using this model in [11]. Model 2 includes four producers, which are perforated in the upper two layers, and two injectors, perforated in the lower two layers. The fluid and rock-fluid properties are the same as were used in Model 1 (see Section 2.3) except that here we set the density for water to be $\rho_w = 60$ lb/ft$^3$, which gives a larger density difference.

The performance of the TPWL procedure for this problem is again studied using perturbation tests. The training schedule and target BHP schedules for this case are shown in Figure 5.1. For producers, the training BHP schedules vary randomly between 1000 psi and 3000 psi and are changed every 200 days. The target BHP schedules vary randomly between 1000 psi and 4000 psi; they also change every 200 days. Clear differences are evident between the two schedules. For the injectors, the training BHP schedules are constant at 8000 psi while the target BHP schedules vary randomly between 7000 psi and 9000 psi. Again, input BHP schedules for test cases are generated as weighted combinations of the training and target schedules, as defined in Eq. 11. The training simulation was run for 5000 days and produced 311 pressure and saturation snapshots.

We apply both the optimized basis method and the EDP procedure to generate stable TPWL models. For the optimized basis method, we considered $l_p$ in the range [40, 90] and $l_s$ in the
range [60, 90]. The search increment for both was 5. A basis with $l_p = 90$ and $l_s = 85$ was found to be stable for the entire simulation, so no basis switching was applied in this case. A total of 154 blocks were locally resolved. These include the 12 well blocks and 144 additional blocks determined by the MPE procedure.

Test cases were run for values of $\alpha$ from 0 to 1. Results for oil and water production rates, for $\alpha = 0.5$, are shown in Figure 16. Water injection rates are presented in Figure 17. It is clear that the TPWL model performs reasonably well for this case. Although slight mismatches are observed for some quantities, the general level of accuracy of the TPWL solution is quite acceptable and the method is clearly stable.

We now present results for this case using the EDP method for stabilization. This requires that the training simulations be run twice – once using the actual densities and once using equal densities. The equal density run is used only to construct $\Phi$; the saved states and matrices are from the run using the actual densities. To allow direct comparison with TPWL(OB), we use $l_p = 90$, $l_s = 85$ and $n_{LR} = 154$.

Figures 18 and 19 display results for oil and water production rates and water injection rates for $\alpha = 0.5$. We again observe stability and a reasonable degree of accuracy in the TPWL results. In fact, these results are quite comparable to those using the optimized basis procedure, shown in Figures 16 and 17.

Table 4 presents errors for oil and water production rates and water injection rates for the two
Figure 16: Model 2 production rates for $\alpha = 0.5$ using TPWL with optimized basis and local resolution ($l_p = 90, l_s = 85, n_{LR} = 154$).

Figure 17: Model 2 injection rates for $\alpha = 0.5$ using TPWL with optimized basis and local resolution ($l_p = 90, l_s = 85, n_{LR} = 154$).
Figure 18: Model 2 production rates for $\alpha = 0.5$ using TPWL with EDP basis and local resolution ($l_p = 90, l_s = 85, n_{LR} = 154$).

Figure 19: Model 2 injection rates for $\alpha = 0.5$ using TPWL with EDP basis and local resolution ($l_p = 90, l_s = 85, n_{LR} = 154$).
sets of TPWL solutions at five values of \( \alpha \). Errors for both methods increase consistently with \( \alpha \), as would be expected. The EDP TPWL method is slightly more accurate than TPWL with the optimized basis, though the magnitudes of the errors are quite comparable.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \alpha = 0.1 )</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPWL(OB+LR), ( E_o )</td>
<td>0.0029</td>
<td>0.0088</td>
<td>0.0142</td>
<td>0.0203</td>
<td>0.0263</td>
</tr>
<tr>
<td>TPWL(OB+LR), ( E_w )</td>
<td>0.0116</td>
<td>0.0351</td>
<td>0.0582</td>
<td>0.0819</td>
<td>0.1063</td>
</tr>
<tr>
<td>TPWL(OB+LR), ( E_i )</td>
<td>0.0065</td>
<td>0.0194</td>
<td>0.0321</td>
<td>0.0449</td>
<td>0.0585</td>
</tr>
<tr>
<td>TPWL(EDP+LR), ( E_o )</td>
<td>0.0022</td>
<td>0.0067</td>
<td>0.0116</td>
<td>0.0166</td>
<td>0.0220</td>
</tr>
<tr>
<td>TPWL(EDP+LR), ( E_w )</td>
<td>0.0097</td>
<td>0.0293</td>
<td>0.0495</td>
<td>0.0702</td>
<td>0.0912</td>
</tr>
<tr>
<td>TPWL(EDP+LR), ( E_i )</td>
<td>0.0059</td>
<td>0.0174</td>
<td>0.0292</td>
<td>0.0409</td>
<td>0.0531</td>
</tr>
</tbody>
</table>

**5.2. Reservoir Model 3: Part of SPE 10 Upper 30 Layers**

We now consider a more complex case. This model, also extracted from the geological description in [34], contains 108,000 grid blocks (\( n_x = 60, n_y = 60, n_z = 30 \)). The permeability distribution is shown in Figure 20. The model is referred to as Model 3 and includes four producers perforated in the upper 12 layers and two injectors perforated in the lower 12 layers. There are thus a total of 72 well blocks. To avoid large discrepancies in the well rates, well indices were set to a specified value. The fluid and rock-fluid properties are the same as were used in Model 2; the density difference between phases is again 15 lb/ft\(^3\). This case is challenging not only because of its size but also because it has more and thicker layers, which means that density-driven gravitational effects can be large.

The training and target BHP schedules are the same as were used for Model 2, as shown in Figure 15. The training simulation was run for 5000 days. A total of 314 pressure and saturation snapshots were saved.

We again present results using both stabilization methods. For the optimized basis method, \( l_p \) and \( l_s \) were both evaluated over the range [60, 120] with increments of 5. The optimal combi-
Figure 21: Model 3 production rates for $\alpha = 0.5$ using TPWL with optimized basis and local resolution ($l_p = 90$, $l_s = 90$, $n_{LR} = 72$).

nation, $l_p = 90$ and $l_s = 90$, was found to be stable for the entire run, so basis switching was not required. Local resolution is applied only to the 72 well blocks. Therefore, the total dimension of the reduced space is 324, compared with 204,000 for the full-order model.

Test cases were again run for $\alpha$ between 0 and 1. Figures 21 and 22 display results for production and injection rates for $\alpha = 0.5$. The results are stable and generally accurate and are of about the same quality as those presented for Model 2. This is encouraging, as Model 3 represents a more challenging test case.

We also generated a TPWL model for this case using the EDP approach, again with $l_p = 90$, $l_s = 90$ and $n_{LR} = 72$. As indicated above, constructing this TPWL model requires that two training runs be performed. Results for production and injection rates for $\alpha = 0.5$ are shown in Figures 23 and 24. These results are again comparable to those using the optimized basis method.

Errors for both sets of runs for a range of $\alpha$ are presented in Table 5. As was observed for Model 2, error increases with increasing $\alpha$ and the EDP method is slightly more accurate than the optimized basis method, though both provide comparable levels of accuracy.

The runtimes for the full-order models for the two examples considered in this section were 30 minutes and 50 minutes respectively on an Opteron dual-core CPU. The TPWL models, by contrast, required at most a few seconds. Thus runtime speedups of $O(10^3)$ were achieved. We reiterate, however, that TPWL requires a training run and additional overhead comparable to about another full-order simulation.

Taken in total, the results presented in this section demonstrate that our enhanced TPWL procedures are able to provide stable results of reasonable accuracy for challenging reservoir
Figure 22: Model 3 injection rates for $\alpha = 0.5$ using TPWL with optimized basis and local resolution ($l_p = 90, l_s = 90, n_{LR} = 72$).

simulation problems. This suggests that these approaches may indeed be applicable in computational optimization or uncertainty quantification procedures. In the next section, we apply the enhanced TPWL method to a production optimization problem.

### Table 5: Errors for TPWL(OB+LR) and TPWL(EDP+LR) for Model 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\alpha$ = 0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPWL(OB+LR), $E_o$</td>
<td>0.0044</td>
<td>0.0132</td>
<td>0.0222</td>
<td>0.0308</td>
<td>0.0398</td>
</tr>
<tr>
<td>TPWL(OB+LR), $E_w$</td>
<td>0.0063</td>
<td>0.0181</td>
<td>0.0298</td>
<td>0.0412</td>
<td>0.0526</td>
</tr>
<tr>
<td>TPWL(OB+LR), $E_i$</td>
<td>0.0064</td>
<td>0.0192</td>
<td>0.0326</td>
<td>0.0457</td>
<td>0.0595</td>
</tr>
<tr>
<td>TPWL(EDP+LR), $E_o$</td>
<td>0.0033</td>
<td>0.0098</td>
<td>0.0164</td>
<td>0.0229</td>
<td>0.0295</td>
</tr>
<tr>
<td>TPWL(EDP+LR), $E_w$</td>
<td>0.0043</td>
<td>0.0128</td>
<td>0.0217</td>
<td>0.0308</td>
<td>0.0403</td>
</tr>
<tr>
<td>TPWL(EDP+LR), $E_i$</td>
<td>0.0057</td>
<td>0.0173</td>
<td>0.0291</td>
<td>0.0401</td>
<td>0.0530</td>
</tr>
</tbody>
</table>

### 6. Use of TPWL for Production Optimization

In this section we use TPWL as a surrogate within a generalized pattern search optimization algorithm. The optimization targets the maximization of net present value for oil production under water injection. We note that TPWL was used previously as a surrogate in gradient-based
Figure 23: Model 3 production rates for $\alpha = 0.5$ using TPWL with EDP basis and local resolution ($l_p = 90$, $l_s = 90$, $n_{LR} = 72$).

Figure 24: Model 3 injection rates for $\alpha = 0.5$ using TPWL with EDP basis and local resolution ($l_p = 90$, $l_s = 90$, $n_{LR} = 72$).
optimizations (with gradients computed numerically) in [11, 25]. That work did not consider systematic retraining of the TPWL model, which we do incorporate here.

6.1. 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 should be well suited for use as a surrogate as it appears to provide a sufficiently accurate 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 coordinate directions) 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 [35] for more detail on GPS and [36] for application of GPS and related procedures to oil production optimization problems.

Our approach for incorporating TPWL into GPS is depicted in Figure 25. 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 using the basis optimization procedure described earlier. 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.

6.2. Optimization Results
In this example we optimize using a small model to enable comparison of results with the surrogate (TPWL) procedure to those using the full-order model. The reservoir model for this case comprises the first four layers of Model 1 (Model 1 is shown in Figure 1) and contains 4800 grid blocks. The fluid and rock-fluid properties and well locations are the same as for Model 1 in Section 4.1 ($\rho_o, \rho_w$ for this case). In this example, we optimize the production well BHPs to maximize net present value (NPV) over five years (1800 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 6000 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 1000 psi and 3000 psi. Initially, the BHPs for the four production wells are set to 1500 psi for the entire production period.

The TPWL model here used basis optimization with basis switching. Typical TPWL parameter values were $l_p = 65, l_s = 75$ and $n_{LR} = 50$. The evolution of NPV with the number of
simulations is shown in Figure 26 and summarized in Table 6. 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 simulations. We note, however, that the detailed well BHP controls do show some differences between the two optimizations. It is frequently observed in oil production optimizations that, even when the resulting objective functions are essentially identical, the well controls determined by different optimization procedures can be somewhat different (see, e.g., [36]). This presumably results from the structure of the objective function in the high-dimensional search space.

The TPWL overhead in this case required the equivalent of about another 10 training simulations. Thus the equivalent number of full-order runs associated with TPWL model construction was about 25, which is a factor of 100 less than that required for the optimization based on full-order simulations. Because our TPWL implementation is currently in Matlab, we expect that TPWL model construction could be accelerated considerably through a careful C++ implementation.

Another example, involving a larger reservoir model, the determination of optimal BHPs for both injection and production wells, and nonlinear constraints in the optimization, is presented in [37]. This example, along with those presented in [11, 25], further demonstrate the applicability of TPWL for use in oil production optimizations.
7. Concluding Remarks

In this work, we developed several new techniques for use in trajectory piecewise linearization (TPWL) procedures. Although our target application area is subsurface flow – specifically oil reservoir simulation – the methods presented here should also find application in other disciplines. The specific methods developed in this paper include a local resolution treatment, in which particular grid blocks are not projected into the low-dimensional space but are instead represented in the original full-order space, and two stabilization procedures. One of the stabilization techniques (referred to as OB) entails the determination of the optimum number of columns in each portion of the basis matrix while the other approach (EDP) applies a basis derived from a simulation with phases of equal density. The use of snapshots from equal-density simulations is shown to lead to basis matrices with better stability properties. The EDP approach, when used by itself, results in a degradation of accuracy in the TPWL model. Accuracy in key quantities such as well rates is recovered, however, by using the EDP method in combination with the local resolution treatment.

Numerical simulations were performed using two highly heterogeneous reservoir models, both containing $O(10^5)$ grid blocks. These results demonstrated the accuracy and robustness of the new TPWL procedures. The EDP approach was observed to be slightly more accurate than the OB method, though the two techniques exhibited levels of accuracy that were generally comparable. The EDP technique is more flexible in terms of the size of the basis matrix, though it requires an additional full-order training run and must be used with local resolution. Since neither stabilization approach is superior in all regards, we believe further development and testing of both methods are warranted.

The TPWL model was also used as a surrogate within a generalized pattern search (GPS) optimization procedure. Retraining was applied to update the TPWL model during the course of the GPS run. The objective function computed using the surrogate model was very close to that using the full-order model. A reduction of about a factor of 100 in the number of full-order simulations required for the optimization was achieved through use of the TPWL surrogate. Future work should be directed toward formalizing the determination of when retraining is required in optimization applications. It may also be useful to view the TPWL surrogate within the context
of space-mapping procedures, which are widely used in surrogate-based optimization.

Acknowledgments

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

References