REDUCED-ORDER MODELING FOR THERMAL SIMULATION

A THESIS SUBMITTED TO THE DEPARTMENT OF
ENERGY RESOURCES ENGINEERING
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

Matthieu Rousset

November 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Energy Resources Engineering.

Louis J. Durlofsky (Principal Adviser)

Abstract

The simulation of complex thermal recovery processes such as in-situ upgrading is computationally demanding. Reduced-order modeling techniques allow the representation of high-dimensional computational problems in a reduced mathematical space, where most of the physical behavior can be reproduced. Use of such models can lead to significant reduction in computational demands. This work focuses on the application of the trajectory piecewise linearization (TPWL) procedure to nonlinear thermal problems. The thermal problem considered is a highly idealized representation of the in-situ upgrading process. Thus this work represents a first step in the application of reduced-order modeling for challenging thermal simulation problems.

The trajectory piecewise linearization procedure entails running one or more training high-fidelity (full-order) simulations. During these runs, snapshots of the states of the system are recorded at every timestep, along with Jacobian matrices and other derivative information. Proper orthogonal decomposition is then applied to produce a basis matrix for projection into the reduced space. The governing equations of the thermal simulation problem are then linearized around the previously saved states, and projected into the reduced space using the basis matrix.

The governing equations solved in this thesis describe the flow of a single component in a single phase coupled to an energy equation. Downhole heaters are modeled by fixing the temperatures of selected grid blocks in the energy equation. The models

that are simulated contain up to 75,000 grid blocks and involve heterogeneous permeability fields. Viscosity is taken to be a strong function of temperature and varies over several orders of magnitude during a simulation. Density is also a strong function of pressure and temperature.

Initially, one training run is used to construct the TPWL model. Results are found to be in reasonable agreement with the reference high-fidelity simulations when the heater and bottom hole pressure controls are set close to those used in the training run. When the controls differ significantly from those of the training run, the TPWL accuracy is shown to degrade, sometimes considerably. Accuracy is restored however when two additional training runs are used in the construction of the TPWL model. The same multiple training approach is applied for a more challenging example, which involves more significant nonlinearities. For this case, TPWL results consistently display close agreement with the reference high-fidelity simulations. For the examples considered in this work, the TPWL procedure provides runtime speedups of a factor of 400-500. The overhead requirements for TPWL depend on the number of training runs used. For the examples considered here, TPWL overhead corresponds to the simulation time for 3-10 high-fidelity runs. Thus, it only makes sense to use TPWL if many simulations are to be performed, as would be the case in computational optimization procedures.

Acknowledgments

First, I would like to express my most sincere gratitude to my advisor, Professor Louis Durlofsky. Lou has provided me with insightful suggestions, important advice and constant encouragement. His invaluable comments and thorough reviews have not only immensely contributed to this work, but also to my academic education.

Special thanks go to Marco Cardoso who provided me with the initial code for TPWL and whose work has greatly inspired this thesis. I would also like to thank Jincong He and Yaqing Fan for their help on this work and interesting discussions on the topics of reduced-order modeling and thermal reservoir simulation. Special thanks also go to David Echeverría and Obiajulu Isebor for their advice, help and comments.

I would also like to give special thanks to my parents and brothers, Philippe, Dominique, Romain and Clément, for their love and constant support through every step of my life. Many thanks also to my friends and colleagues at Stanford, who make my stay in California a very pleasant one. A special mention for Paige and those great memories that we share.

Also, I am very grateful to Jean-Charles Ginestra and G. Gao for their advice and support during my internship at Shell, where I started this work. I would also like to thank the industrial affiliates of the SUPRI-B (Reservoir Simulation), SUPRI-HW (Advanced Wells) and Smart Fields Research Consortia at Stanford University for financial support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

World population is expected to grow from six to nine billion in the next four decades and energy demand could increase by a factor of two over this same period [1]. Today, this demand is mainly fulfilled by the use of fossil fuels such as coal, conventional oil and natural gas, but this might not be the case in the future as conventional oil and gas are expected to be more scarce. Thus it will become necessary to produce new kinds of resources in order to meet this increasing demand. This is likely to include significant quantities of unconventional oil resources such as oil sands, and possibly more challenging resources such as oil shales.

Oil shale refers to a fine-grained sedimentary rock that contains a large amount of kerogen, and this kerogen can be converted into oil by heating to high temperature. An immense amount of oil shale has been identified in the Green River formation that covers parts of Colorado, Wyoming and Utah. According to the US Geological Survey [2], there are over one trillion barrels of oil (in oil shale) in place. Oil sands are a type of bitumen deposit and consist of sands mixed with an extremely dense and viscous type of petroleum. They are found in large quantities in the province of

Alberta, Canada, where it is estimated that there are about 1.7 trillion barrels of oil in place along the Athabasca river.

Oil from both oil sands and oil shales can be recovered using surface mining and further processing. However, only the shallower parts of the deposits can be recovered using this technology, leaving the large majority unrecovered. Shell has developed new technologies that enable the production of deeper formations: the in-situ conversion process (ICP) for oil shale, and the in-situ upgrading process (IUP) for oil sand and heavy oil [3]. Both processes use underground heaters, as shown in Figure 1.1, in order to alter the properties of bitumen or kerogen directly within the subsurface. Then the fluids can be produced using conventional wells. Although the technical feasibility of such techniques has been demonstrated at pilot scale, many challenges remain for large-scale commercial application. Use of numerical modeling and optimization may contribute significantly to resolving these challenges.

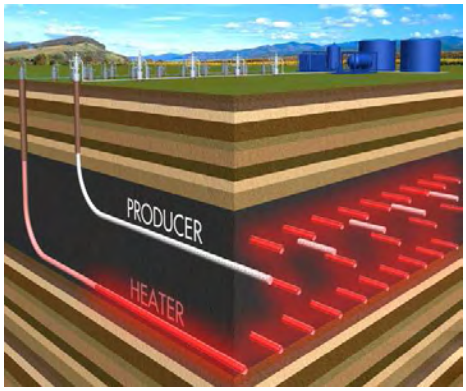


Figure 1.1: Schematic of Shell's in-situ upgrading process (IUP) (from Shell, 2007)

Modeling numerically the in-situ upgrading process is, however, a difficult task because many physical and chemical phenomena must be represented in the model. In addition to the transport of fluids through porous media, in-situ upgrading involves the transport of heat and multiple chemical reactions. The reactions and system properties are strongly dependent on temperature, which renders the problem highly

nonlinear. It is also necessary to account for a large number of components in the fluid. These and other complications lead to highly involved simulation models. For this reason, the simulation of practical problems of this type is CPU intensive, and can require days of calculation for a single production scenario.

Regardless of whether we are considering the production of conventional or unconventional resources, the use of computational optimization can improve recovery and lead to more efficient operations. Optimization here refers to mathematical procedures that provide a set of parameters or controls that maximize a selected response of the system. The system is a model of the production of oil or gas under some constraints. A classical optimization problem in reservoir engineering is to find the optimal locations of new wells in order to maximize future profits. Optimization algorithms can consider more than one objective function. For example, it is possible to find the best set of inputs in order to maximize profitability while simultaneously minimizing the environmental impact. The production of oil shales and oil sands is expensive and has a large environmental impact, so these operations could strongly benefit from the application of optimization procedures.

There are different types of optimization algorithms that can be applied and all have their advantages and limitations. Adjoint-based algorithms might require only tens of runs, but such approaches have yet to be applied for in-situ upgrading. Noninvasive approaches such as genetic algorithms or pattern search may require hundreds or thousands of runs. These algorithms are typically implemented on multiple cores, which significantly reduces the elapsed time, but the computational costs remain very high.

Although the potential benefits can be large, performing optimization for in-situ upgrading processes may not be practical because the simulation models require extremely long calculation times. Motivated by similar concerns in other application areas, so-called reduced-order modeling procedures, which attempt to reduce the

overall computational cost of the simulations, have been developed and applied. The intent of these approaches is to significantly decrease simulation time at the price of a relatively small loss of accuracy. Indeed, if the model retains sufficient accuracy, the optimal controls for the system can still be found through application of an optimization algorithm. In order to achieve this, the reduced-order model (ROM) should be able to model the response of the original (full-order) system, with a minimal loss of accuracy, when the system controls are modified.

1.2 Literature review

The general idea behind reduced-order modeling approaches is to represent a high-dimensional problem in a reduced mathematical space, where most of the physical behavior can be reproduced. A number of these procedures involve the use of proper orthogonal decomposition (POD). This approach requires the simulation of one or more high-fidelity training runs during which state vectors, referred to as snapshots, are saved at each timestep. A singular value decomposition (SVD) is then performed on the so-called snapshot matrix. The singular vectors associated with the largest singular values are used to define a projection from the full-order space to a reduced space. This procedure is explained in detail in Chapter 2. Discussions of previous applications using the POD procedure, and recent developments in the field of reduced-order modeling, have been presented by Cardoso and Durlofsky [4, 5]. The following overview follows their descriptions.

Lumley [6] introduced proper orthogonal decomposition and used it in order to identify coherent structures in dynamical systems. It has since been applied for a variety of applications. For example, Zheng et al. [7] applied POD to a system of partial differential equations describing a reactor system. Meyer and Matthies [8] applied it on a structural model for a horizontal-axis wind turbine rotor blade, while

Bui-Thanh et al. [9] used it to reconstruct flow fields from incomplete aerodynamic data sets. Cao et al. [10] created a POD-based reduced model for a large-scale upper tropical Pacific ocean model.

It appears that Vermeulen et al. [11] were the first to apply POD for subsurface flow modeling. They achieved significant runtime speedup for a heterogeneous aquifer model with 33,000 active nodes where the governing equation was linear or nearly linear. Jansen and coworkers [12, 13] applied POD to two-phase, oil-water reservoir models. For these nonlinear cases, they achieved limited speedups, of a factor of 5 or less. More specifically, Markovinović et al. [13] used POD to reduce the computational effort required in solving the linear systems resulting from the application of Newton's method to the nonlinear system. However, all other tasks such as the construction of the Jacobian matrix, which is required at each iteration, were not affected. In consequence, they only achieved a runtime speedup of about a factor of 3. Van Doren et al. [12] used POD to accelerate a gradient-based optimization procedure. They applied it to both the forward model and the adjoint system. The total computational speedup was less than a factor of 1.5 due to the need for retraining the reduced model during the optimization process.

POD-based procedures for oil-water reservoir simulation have been further enhanced by Cardoso et al. [4, 14]. These researchers specified that the controls in the training simulations, such as the well bottom hole pressures, vary randomly over a specified range of values. They also used a snapshot clustering procedure [15] and a missing point estimation technique [16] that reduced the computational requirements for the reduced-order model. They applied their procedure to geologically realistic models containing as many as 60,000 grid blocks and achieved speedups of up to a factor of about 10. However, POD was still applied only at the linear solver level so the speedup achievable was intrinsically limited. The trajectory piecewise linearization

(TPWL) method, which is a relatively recent development in the field of reduced-order modeling, can potentially overcome this problem by combining POD with a piecewise linearization. In practice, it requires recording, in addition to the solution snapshots, the Jacobian matrix and other matrices during the training simulations.

The TPWL procedure has shown real promise in reducing the computational time of simulations. Rewienski first applied TPWL to nonlinear transmission-line circuit models [17]. The system contained 1500 unknowns and used 21 linearization states to generate a reduced-order model of dimension 30. He achieved runtime speedups of 1150 compared to the full-order model. The error using the reduced model was between 0.4% and 13.5% compared to full-order simulations. TPWL has since been applied in other fields such as computational fluid dynamics (Gratton and Willcox [18]) and heat transfer modeling (Yang and Shen [19]). The first application of TPWL in oil reservoir simulation was performed by Cardoso and Durlofsky [20]. They applied the procedure to two-phase oil-water problems and used realistic geological models containing up to 60,000 grid blocks. When test runs had two fluids of equal density, TPWL results were in close agreement with their respective full-order solutions. The runtime speedups achieved were quite substantial (a factor of 200 to 2000). When the fluid densities differed considerably, however, they reported instabilities in some instances. He et al. [21, 22] proposed a stabilization scheme in order to address these limitations. In addition, they introduced a local high-resolution procedure that enhanced TPWL accuracy.

Cardoso and Durlofsky [20] employed TPWL in optimizations involving two-phase oil-water models. They applied it in conjunction with gradient-based techniques, where gradients were computed numerically using finite differences. They obtained optimized net present values in close agreement with those obtained from optimizations using high-fidelity models. He et al. [21] successfully incorporated two-phase oil-water TPWL models into a generalized pattern search optimization technique.

Occasional retraining was applied to maintain accuracy in the TPWL model.

TPWL has not yet been applied for more complex subsurface processes such as thermal reservoir simulation. Thermal production strategies introduce complicated physics and significant nonlinearities into the simulation problem. If TPWL, or related reduced-order modeling techniques, can be developed for these problems, then they could be applied for production optimization. As indicated above, optimizations for complex thermal processes are very challenging when full-order models are used. It is therefore of interest to evaluate the performance of TPWL for such applications.

1.3 Approach and objectives

In light of its potential for large runtime speedup in optimization, the objective of this work is to investigate the feasibility and performance of the trajectory piecewise linearization procedure for idealized nonlinear problems related to those that appear in thermal reservoir simulation.

This work represents a preliminary assessment of the applicability of TPWL for thermal problems, as the models considered here are much simpler than those used in realistic thermal simulations. As discussed previously, IUP and ICP are thermal processes that use downhole heaters in order to facilitate upgrading and alter fluid properties directly within the subsurface. Numerous chemical reactions, together with heat transfer, modify fluid properties and allow production with traditional wells. These in-situ upgrading and conversion processes are very complicated as they involve chemical kinetics, phase equilibrium between gas and liquid phases, mass transport, heat transfer, geomechanical effects, modification of porosity and permeability under the effect of pressure, temperature and composition, etc. In addition, the complex nature of the fluids and reactions requires the use of fully-compositional models with a large number of components.

In this work, our approach is to apply the TPWL procedure to equations that are much simpler and faster to solve, yet retain some of the nonlinear character of thermal problems. Heavy oil is modeled with a single liquid component, though its properties are functions of pressure and temperature. In particular, the fluid viscosity decreases by several orders of magnitude over the temperature range encountered during downhole heating. Density is also dependent on temperature in order to reproduce, to some extent, the increase in reservoir pressure with increasing temperature. Our model also includes an energy equation, in which the downhole heaters provide the energy input, to model temperature variation throughout the system. Chemical reactions, phase changes, geomechanical effects and other complicated physics are not considered. Although our representation is quite simplified relative to realistic IUP and ICP models, the large changes in density and viscosity with reservoir pressure and temperature should mimic some of the important nonlinearities appearing in the original problem.

This thesis proceeds as follows. In Chapter 2 we present briefly the thermal simulation equations solved under the assumptions stated above. We then describe the use of proper orthogonal decomposition. This is followed by a detailed description of the trajectory piecewise linearization procedure applied to our idealized thermal reservoir simulation. The necessary modifications to the full-order simulator are also presented in this chapter together with implementation details. In Chapter 3, numerical examples demonstrating the performance of TPWL for this problem are presented. The examples considered involve realistic geological models which contain up to 75,000 grid blocks. As described above, oil viscosity varies over several orders of magnitude and density is a function of pressure and temperature. Initially, a single training simulation is used, and test results show reasonable accuracy for controls that are close to those used for the training. A sensitivity analysis is presented in which well controls are varied significantly from those used in the training case. For the most

extreme test cases, the accuracy of the TPWL model is shown to degrade. However, reasonable accuracy is restored by using multiple training runs in the construction of the TPWL model. This approach is then tested further by increasing the nonlinear dependencies in viscosity and density. Taken in total, the results indicate that TPWL is well suited for the solution of the idealized thermal models considered in this study. Conclusions and directions for future work are presented in Chapter 4.

Chapter 2

Governing equations and TPWL formulation

2.1 Thermal reservoir simulation equations

Subsurface flow models for the general isothermal case are derived by combining mass conservation equations with the multiphase version of Darcy's law. In the case of thermal reservoir simulation, an energy conservation equation must also be solved. In this work, the fluid is modeled as a single component in a single phase, which greatly simplifies the formulation. Other simplifications and assumptions used in this work are explained in the following sections.

2.1.1 Mass conservation equation

Mass conservation for the oil component is expressed by the following partial differential equation:

$$\nabla \cdot [\lambda_o \mathbf{k} (\nabla p - \rho_o g \nabla D)] = \frac{\partial}{\partial t} \left(\frac{\phi}{B_o} \right) + \tilde{q}_o^w, \tag{2.1}$$

where **k** is the absolute permeability, assumed to be a diagonal tensor, p is pressure, ρ_o is oil density, g is gravitational acceleration, D is depth and $\lambda_o = k_{ro}/(\mu_o B_o)$ is the oil mobility. In this case, since we have only one phase, the relative permeability $k_{ro} = 1$ and therefore oil mobility reduces to $\lambda_o = 1/(\mu_o B_o)$. Here B_o is the formation volume factor of oil and μ_o is the oil viscosity. Other variables are t, which represents time, ϕ which designates porosity, and $\tilde{q}_o^w$, the source/sink term (the superscript w denotes well and tilde means per unit volume).

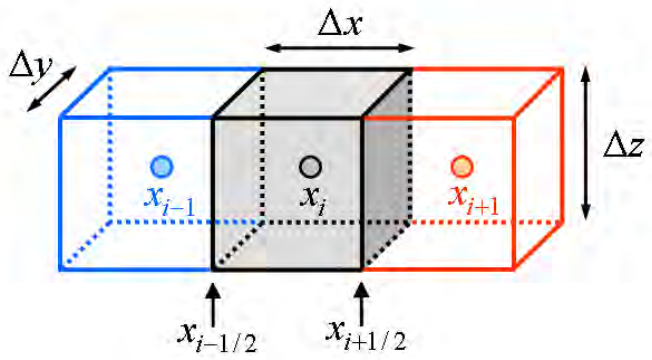


Figure 2.1: Representation of cell i and its neighboring cells

In this work, the governing equations are solved using Stanford's General Purpose Research Simulator (GPRS) [23, 24]. We briefly present here the finite volume discretization used to solve equation 2.1. This description closely follows that in [5] (see also [25] for more details). For simplicity, we consider a one-dimensional grid oriented horizontally (therefore $\nabla D = 0$) and we assume the grid block dimensions (Δx,Δy,Δz) to be constant. In the fully-implicit representation, the discretized form of the convective terms is

$$\frac{\partial}{\partial x}\left[k\lambda_o \frac{\partial p}{\partial x}\right] \approx \frac{1}{V}\left(\Upsilon^{n+1}_{i-1/2}\left[p^{n+1}_{i-1} - p^{n+1}_{i}\right] + \Upsilon^{n+1}_{i+1/2}\left[p^{n+1}_{i+1} - p^{n+1}_{i}\right]\right), \qquad (2.2)$$

where the superscript $n+1$ designates the next timestep and the subscript indicates the gridblock number as defined in Figure 2.1. Here, $V = \Delta x \Delta y \Delta z$ is the volume of

grid block i and $\Upsilon_{i-1/2}$ is the transmissibility at interface $i-1/2$. It relates the flow through this interface to the difference of pressure between grid blocks $i-1$ and i. This transmissibility is a function of pressure and temperature (through B_o and μ_o) and is given by

$$\Upsilon_{i-1/2}^{n+1} = \left(\frac{k\Delta y\Delta z}{\Delta x}\right)_{i-1/2} \left(\frac{1}{B_o\mu_o}\right)_{i-1/2}^{n+1}. \tag{2.3}$$

In this discretization scheme, the mass accumulation term in grid block i is represented as:

$$\frac{\partial}{\partial t}\left(\frac{\phi}{B_o}\right) \approx \frac{\phi}{\Delta t}\left[\left(\frac{1}{B_o}\right)_i^{n+1} - \left(\frac{1}{B_o}\right)_i^{n}\right], \tag{2.4}$$

where $\Delta t = t^{n+1} - t^n$ is the timestep. The second term on the right-hand side of equation 2.1 represents the producer wells. Its discrete representation is given by:

$$(q_o^w)_i^{n+1} = W_i\,(\lambda_o)_i^{n+1}\left[p_i^{n+1} - p_i^w\right], \tag{2.5}$$

where $(q_o^w)_i^{n+1}$ is the flow rate (volume per time) of oil from block i into the well at time $n+1$, W_i is the well index (defined below), p_i^{n+1} is the pressure in grid block i at timestep $n+1$, and p_i^w is the wellbore pressure of well w in grid block i. If the well is operated under bottom hole pressure (BHP) control, the value of p_i^w is specified (as a function of time) and the corresponding flow rate $(q_o^w)_i^{n+1}$ is computed using equation 2.5. The well index W_i is given by:

$$W_i = \left(\frac{2\pi k\Delta z}{\ln(r_0/r_w)}\right)_i, \tag{2.6}$$

where r_w is the wellbore radius and $r_0 \approx 0.2\Delta x$ for $\Delta x = \Delta y$ and isotropic permeability. One can consult [26] for more details and [24] for a discussion of other well representations. The discrete representation of mass conservation in three-dimensional systems can be obtained from a generalization of the one-dimensional case presented

above.

2.1.2 Energy conservation equation

Under the simplifications noted above, energy conservation is described by the following partial differential equation:

$$\nabla \cdot (\rho_o H_o \mathbf{u}_o) + \nabla \cdot (\kappa \nabla T) = \frac{\partial}{\partial t} \left[\phi \rho_o U_o + (1-\phi)\, \rho_R U_R\right] + \tilde{q}^H + \rho_o H_o \tilde{q}_o^w. \qquad (2.7)$$

The term $\nabla \cdot (\rho_o H_o \mathbf{u}_o)$ represents heat transfer through convection, where H_o is the oil specific enthalpy and $\mathbf{u}_o$ is the Darcy velocity of oil, given by $\mathbf{u}_o = (\mathbf{k}/\mu_o)(\nabla p - \rho_o g \nabla D)$. The term $\nabla \cdot (\kappa \nabla T)$ represents conductive heat transfer, where T represents oil temperature and κ is the thermal conductivity (here taken to be a scalar). Together they form the heat flux part of the equation. On the right hand side, U_o is the internal energy of oil and U_R is the internal energy of the rock, $\tilde{q}^H$ refers to the energy injected through the downhole heaters, and $\rho_o H_o \tilde{q}_o^w$ represents the energy transported by the oil into production wells (the superscript w denotes well and tilde means per unit volume).

We now briefly describe the finite volume discretization of equation 2.7 used in GPRS. We again consider a one-dimensional horizontal model. Using the notation presented in Figure 2.1, the conductive heat transfer term can be expressed in discrete form as

$$\nabla \cdot (\kappa \nabla T) \approx \frac{1}{V} \frac{\kappa \Delta y \Delta z}{\Delta x} \left(T_{i-1}^{n+1} - T_i^{n+1}\right). \qquad (2.8)$$

Although κ generally varies with temperature and location, in this work, for simplicity, we take it to be a constant. The convective part of the heat flux is given by:

$$\nabla \cdot (\rho_o H_o \mathbf{u}_o) \approx \frac{\rho_o^0}{V} \left((H_o \Upsilon)_{i-1/2}^{n+1} \left[p_{i-1}^{n+1} - p_i^{n+1}\right] + (H_o \Upsilon)_{i+1/2}^{n+1} \left[p_{i+1}^{n+1} - p_i^{n+1}\right] \right). \qquad (2.9)$$

Here ρ_o^0 is the reference density and the transmissibility $(\Upsilon)_{i-1/2}^{n+1}$ is computed according to equation 2.3 (and similarly for $(\Upsilon)_{i+1/2}^{n+1}$). Defining the total internal energy in grid block i as $E_i = (\phi\rho_o U_o + (1-\phi)\,\rho_R U_R)_i$, the first-order time discretization for the energy accumulation term is

$$\frac{\partial E}{\partial t} \approx \frac{1}{\Delta t}\left(E_i^{n+1} - E_i^n\right). \tag{2.10}$$

The internal energies of rock (U_R) and oil (U_o) are represented as

$$U_R = C_R\left(T_i - T_{ref}\right), \tag{2.11}$$

$$U_o = C_o\left(T_i - T_{ref}\right), \tag{2.12}$$

where C_o and C_R are the specific heat capacities of oil and rock respectively, at constant volume.

The well term $\rho_o H_o \tilde{q}_o^w$ describes the energy transported by the oil into the well. The flow rate q_o^w is computed from equation 2.5, and $\rho_o H_o$ is computed using the well block properties.

Heat injection q^H is the key driving force for both IUP and ICP. In this work, downhole heaters are modeled by specifying their temperature. Numerically, this is achieved by specifying the initial grid block temperature to be the heater temperature, and then fixing the heat capacity of the rock in this block to a very large number (in this work we use $C_R = 10^8$ Btu/lb/ft^3/°F in heater blocks). The heater temperature can however be changed at any time by simply specifying a new block temperature. This simple approach allows us to introduce heat without implementing a proper heater well model. This procedure is physically reasonable when the size of the heater well block is comparable to the size of the actual heater well. We note that this approach was used by Fan et al. [27]. More sophisticated heater well models

have been developed recently by Aouizerate et al. [28] and could be incorporated in our formulation. As a consequence of our simplified treatment, the term q^H vanishes from the discretized system, as it is represented, in effect, as a boundary condition.

2.1.3 Discrete system of equations

The overall discrete set of equations solved at every timestep in GPRS is constructed by combining the discrete representations for equations 2.1 and 2.7. This nonlinear system can be written as:

$$\mathbf{g}\left(\mathbf{x}^{n+1}, \mathbf{x}^{n}, \mathbf{u}^{n+1}\right) = \mathbf{F}\left(\mathbf{x}^{n+1}\right) + \mathbf{A}\left(\mathbf{x}^{n+1}, \mathbf{x}^{n}\right) + \mathbf{Q}\left(\mathbf{x}^{n+1}, \mathbf{u}^{n+1}\right) = 0, \tag{2.13}$$

where $\mathbf{x} = \left(p_1, T_1, \ldots, p_{N_g}, T_{N_g}\right)^T$ is the vector of states and $\mathbf{u} = \left(p_{w_1}, \ldots, p_{w_{N_w}}\right)^T$ is the vector of controls, with N_g the number of grid blocks and N_w the number of producer wells. The vector $\mathbf{g} = \left(g_{m_1}, g_{e_1}, \ldots, g_{m_{N_g}}, g_{e_{N_g}}\right)^T$ is the overall residual we seek to drive to zero, $\mathbf{F} = \left(F_{m_1}, F_{e_1}, \ldots, F_{m_{N_g}}, F_{e_{N_g}}\right)^T$ is the overall flux term, $\mathbf{A} = \left(A_{m_1}, A_{e_1}, \ldots, A_{m_{N_g}}, A_{e_{N_g}}\right)^T$ is the overall accumulation term, and $\mathbf{Q} = \left(Q_{m_1}, Q_{e_1}, \ldots, Q_{m_{N_g}}, Q_{e_{N_g}}\right)^T$ is the overall source/sink term. Here the subscript m indicates that the contribution derives from the mass conservation equation 2.1 and e that the contribution is from the energy equation 2.7. The number (1 to N_g) designates the grid block.

The system of equations 2.13 is nonlinear because it contains functions of unknowns (e.g., $\lambda_o(p, T)$) multiplying unknowns. Therefore it cannot be solved directly. We use the Newton-Raphson method, in which the solution is found iteratively by solving the following linear system at each iteration:

$$\mathbf{J}^{(\nu)} \Delta \mathbf{x}^{(\nu+1)} = -\mathbf{g}^{(\nu)}, \tag{2.14}$$

where $\mathbf{J}^{(\nu)}$ is the Jacobian matrix, $\Delta\mathbf{x}^{(\nu+1)} = \mathbf{x}^{(\nu+1)} - \mathbf{x}^{(\nu)}$ is the change of the unknown over the iteration, and $\mathbf{g}^{(\nu)}$ is the residual at the end of the ν^{th} iteration. The Jacobian matrix contains the derivatives of all elements of vector $\mathbf{g}$ with respect to all state variables (the elements of vector $\mathbf{x}$). This can be written mathematically as:

$$\mathbf{J} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}. \tag{2.15}$$

For three-dimensional systems, $\mathbf{J}$ is block hepta-diagonal and each block is of dimensions 2×2. For example, the block element containing the derivatives of the residual in grid block i with respect to the variables in grid block j is as follows:

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)_{ij} = \begin{pmatrix} \frac{\partial g_{m_i}}{\partial p_j} & \frac{\partial g_{m_i}}{\partial T_j} \\ \frac{\partial g_{e_i}}{\partial p_j} & \frac{\partial g_{e_i}}{\partial T_j} \end{pmatrix} \tag{2.16}$$

2.2 TPWL representation for simplified thermal model

The objective of this work is to apply the TPWL procedure to the model described above in order to significantly decrease the computational time of running a simulation, at the cost of a small loss of accuracy. As will be explained below, the main cost of using TPWL is the overhead computations. Indeed, some preprocessing is needed in order to construct the TPWL model before it can be used with new sets of controls. This phase is often referred to as training. As is the case for many other reduced-order modeling techniques, because of this overhead training cost, it only makes sense to use TPWL for applications that require many simulation runs. This is indeed the case for optimization or sensitivity analysis. The following descriptions

of the POD and TPWL procedures follow those provided by Cardoso and Durlofsky [5]. The details of the TPWL procedure differ from those in [5] as we are here addressing a thermal problem rather than an oil-water system.

2.2.1 Proper orthogonal decomposition

TPWL entails a piecewise linearization combined with a projection into a reduced space. The reduced space is intended to capture most of the physical behavior of the full-order space. In this work, the projection into reduced space $\mathbf{z}$ is achieved by using an orthonormal basis $\mathbf{\Phi}$, of dimensions $2N_g \times \ell$, found by proper orthogonal decomposition (POD). Specifically, we have $\mathbf{x} = \mathbf{\Phi z}$. The dimension of the initial high-fidelity model is $2N_g$ since there are two states (T and p) for each grid block, and ℓ is the dimension of the reduced space. In general ℓ is much smaller than N_g. We now briefly describe the construction of the basis matrix $\mathbf{\Phi}$ using POD.

The first step in constructing $\mathbf{\Phi}$ is to run one or more training simulations with the high-fidelity model. In order to do so, control variables such as well bottom hole pressures (BHP), well flow rates or heater temperatures must be set by the user. Cardoso [4] has shown that for oil-water systems, varying the BHP randomly over a prescribed range of values appears to provide a reasonable basis $\mathbf{\Phi}$. Indeed, the training simulation(s) should capture as much of the potential behavior of the system as possible. During these runs, state variables are saved at each timestep. These vectors of saved states are known as solution snapshots. Consistent with [4], POD is applied separately for pressure and temperature, so two snapshots are saved at a given timestep: $\mathbf{x}_p$ containing the pressures of all grid blocks and $\mathbf{x}_T$ containing the temperatures of all grid blocks. Two data matrices $\mathbf{X}_p$ and $\mathbf{X}_T$ can then be formed (the snapshots at each timestep represent the columns). For convenience, the subscript is dropped in the following description, though it should be kept in mind that the procedure is performed separately for $\mathbf{X}_p$ and $\mathbf{X}_T$.

The POD basis matrix $\mathbf{\Phi}$ contains as its columns the eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ that are associated with the largest eigenvalues. If there are k snapshots recorded during the training simulations, the data matrix $\mathbf{X}$ is of dimensions $N_g \times k$. Therefore the dimensions of $\mathbf{X}\mathbf{X}^T$ are $N_g \times N_g$. The eigenvectors of $\mathbf{X}\mathbf{X}^T$ are identical to the left singular vectors of $\mathbf{X}$. Using this property, it is sufficient to perform a singular value decomposition (SVD) of $\mathbf{X}$ to obtain the eigenvectors of $\mathbf{X}\mathbf{X}^T$. Eigenvalues of $\mathbf{X}\mathbf{X}^T$ (λ_i) are related to the singular values of $\mathbf{X}$ (σ_i) through the relation $\lambda_i = \sigma_i^2$.

Once the eigenvectors and eigenvalues of the covariance matrix are computed, the ℓ vectors associated with the largest eigenvalues are used in $\mathbf{\Phi}$. The choice of ℓ can be based on the distribution of energy among the eigenvalues or other criteria can be applied – see He at al. [21] for discussion. Here we apply an energy criterion. Specifically, the energy is computed as $E_\ell = \sum_{i=1}^{\ell} \lambda_i$ with the eigenvalues λ_i in decreasing order. When $\ell = k$, E_ℓ represents all of the energy in the system. The actual value of ℓ is found by specifying the percentage of the total system energy to be captured.

This procedure is applied successively to $\mathbf{X}_p$ and $\mathbf{X}_T$ to create two basis matrices $\mathbf{\Phi}_p$ and $\mathbf{\Phi}_T$ of respective dimensions $N_g \times \ell_p$ and $N_g \times \ell_T$. These matrices are then combined into one basis matrix $\mathbf{\Phi}$ of dimension $2N_g \times \ell$ where $\ell = \ell_p + \ell_T$. The matrix $\mathbf{\Phi}$ can now be used to project the high-fidelity state vector $\mathbf{x}$ into a reduced-order space of dimension ℓ ($\mathbf{z} = \mathbf{\Phi}^T\mathbf{x}$) or it can be used to represent the full-order states in terms of the reduced-order states ($\mathbf{x} = \mathbf{\Phi}\mathbf{z}$). We will see below how $\mathbf{\Phi}$ enters into the TPWL formulation.

2.2.2 Trajectory piecewise linearization

The basic idea behind the TPWL procedure is to run a small number of high-fidelity training simulations and save the solution snapshots, Jacobian matrices and other

derivatives at all timesteps. Then, in order to compute the solution (designated $\mathbf{x}^{n+1}$) for a new simulation with different controls (designated $\mathbf{u}^{n+1}$), we apply a first-order expansion around a previously saved state. Indeed, the residual equation 2.13 can be expanded around any of the saved states, denoted by $(\mathbf{x}^{i+1}, \mathbf{x}^i, \mathbf{u}^{i+1})$. Here, $\mathbf{x}^{i+1}$ and $\mathbf{u}^{i+1}$ are the vector of states and vector of controls respectively, saved at timestep $i+1$. The vector $\mathbf{x}^i$ refers to the saved states from the previous timestep.

Representation of the residual $\mathbf{g}\left(\mathbf{x}^{n+1}, \mathbf{x}^n, \mathbf{u}^{n+1}\right)$, which we seek to drive to zero, around saved state $(\mathbf{x}^{i+1}, \mathbf{x}^i, \mathbf{u}^{i+1})$, gives:

$$\begin{aligned} \mathbf{g}\left(\mathbf{x}^{n+1}, \mathbf{x}^n, \mathbf{u}^{n+1}\right) &= \mathbf{g}\left(\mathbf{x}^{i+1}, \mathbf{x}^i, \mathbf{u}^{i+1}\right) + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)_{i+1} \left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) \\ &+ \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)_{i} \left(\mathbf{x}^{n} - \mathbf{x}^{i}\right) + \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right)_{i+1} \left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right) + \cdots \end{aligned} \tag{2.17}$$

Note that $\mathbf{u}^{n+1}$, the new set of controls at timestep $n+1$, is specified by the user. Since $\mathbf{g}\left(\mathbf{x}^{n+1}, \mathbf{x}^n, \mathbf{u}^{n+1}\right) = \mathbf{g}\left(\mathbf{x}^{i+1}, \mathbf{x}^i, \mathbf{u}^{i+1}\right) = 0$, it is evident from this equation that, using the previously saved data, and given $\mathbf{x}^n$, the solution at the next timestep $\mathbf{x}^{n+1}$ can be approximated. In order to minimize the approximation error, the saved state $\mathbf{x}^{i+1}$ should be chosen to be as 'close' as possible to $\mathbf{x}^{n+1}$. However, only the solution at the current timestep n is known. But, if $\mathbf{x}^i$ is the 'closest' saved state to the current state $\mathbf{x}^n$, it is reasonable to use $\mathbf{x}^{i+1}$ as the 'closest' saved state to $\mathbf{x}^{n+1}$. There are different ways to define the 'distance' between the new and saved states. More details on this issue are presented below.

The exact terms that need to be saved from the training simulations can be determined by expanding all terms of equation 2.13. Starting with the flux terms, the first order expansion is given by

$$\mathbf{F}^{n+1} \approx \mathbf{F}^{i+1} + \frac{\partial \mathbf{F}^{i+1}}{\partial \mathbf{x}^{i+1}} \left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right), \tag{2.18}$$

where $\mathbf{F}^{n+1} = \mathbf{F}\left(\mathbf{x}^{n+1}\right)$ and $\mathbf{F}^{i+1} = \mathbf{F}\left(\mathbf{x}^{i+1}\right)$. The matrix $\partial\mathbf{F}^{i+1}/\partial\mathbf{x}^{i+1}$ represents the derivative of the flux terms (mass and energy components) with respect to the state variables (p and T).

Expanding the accumulation term, an additional term appears because $\mathbf{A}$ depends on both $\mathbf{x}^{n+1}$ and $\mathbf{x}^{n}$:

$$\mathbf{A}^{n+1} \approx \mathbf{A}^{i+1} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i+1}}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\left(\mathbf{x}^{n} - \mathbf{x}^{i}\right), \tag{2.19}$$

where $\mathbf{A}^{n+1} = \mathbf{A}\left(\mathbf{x}^{n+1}, \mathbf{x}^{n}\right)$ and $\mathbf{A}^{i+1} = \mathbf{A}\left(\mathbf{x}^{i+1}, \mathbf{x}^{i}\right)$. Finally, the first order approximation of the source term is

$$\mathbf{Q}^{n+1} \approx \mathbf{Q}^{i+1} + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{x}^{i+1}}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right), \tag{2.20}$$

where $\mathbf{Q}^{n+1} = \mathbf{Q}\left(\mathbf{x}^{n+1}, \mathbf{u}^{n+1}\right)$ and $\mathbf{Q}^{i+1} = \mathbf{Q}\left(\mathbf{x}^{i+1}, \mathbf{u}^{i+1}\right)$. The matrix $\partial\mathbf{Q}^{i+1}/\partial\mathbf{u}^{i+1}$ is technically of dimensions $2N_g \times N_w$ where N_w refers to the number of producer wells, and $\mathbf{u}$ is of dimension N_w. However, because only the small fraction of grid blocks that contain producer wells contribute to $\mathbf{Q}$, the entries for blocks without wells are zero. Thus the dimension of $\mathbf{Q}$ can be taken to be $2N_w$ and the dimensions of $\partial\mathbf{Q}/\partial\mathbf{u}$ can be taken to be $2N_w \times N_w$.

Inserting equations 2.18, 2.19 and 2.20 into equation 2.13 gives

$$\begin{aligned}
\mathbf{g}^{n+1} \approx\ & \mathbf{F}^{i+1} + \frac{\partial \mathbf{F}^{i+1}}{\partial \mathbf{x}^{i+1}}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \\
& \mathbf{A}^{i+1} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i+1}}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\left(\mathbf{x}^{n} - \mathbf{x}^{i}\right) + \\
& \mathbf{Q}^{i+1} + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{x}^{i+1}}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right).
\end{aligned} \tag{2.21}$$

Recognizing that $\mathbf{g}^{i+1} = \mathbf{F}^{i+1} + \mathbf{A}^{i+1} + \mathbf{Q}^{i+1} = 0$ and that the Jacobian matrix

associated with state $\mathbf{x}^{i+1}$ is defined by

$$\mathbf{J}^{i+1} = \frac{\partial \mathbf{g}^{i+1}}{\partial \mathbf{x}^{i+1}} = \frac{\partial \mathbf{F}^{i+1}}{\partial \mathbf{x}^{i+1}} + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i+1}} + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{x}^{i+1}}, \tag{2.22}$$

it is possible to rewrite equation 2.21 as

$$\mathbf{g}^{n+1} \approx \mathbf{J}^{i+1}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) + \frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\left(\mathbf{x}^{n} - \mathbf{x}^{i}\right) + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right). \tag{2.23}$$

Using the fact that $\mathbf{g}^{n+1} = 0$, a linearized representation of the governing equations can be written as

$$\mathbf{J}^{i+1}\left(\mathbf{x}^{n+1} - \mathbf{x}^{i+1}\right) = -\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\left(\mathbf{x}^{n} - \mathbf{x}^{i}\right) - \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right). \tag{2.24}$$

All quantities in this equation are known (or specified by the user) except for $\mathbf{x}^{n+1}$. It is evident from equation 2.24 that, in addition to the solution snapshots $\mathbf{x}^{i+1}$ and $\mathbf{x}^{i}$, the Jacobian matrix $\mathbf{J}^{i+1}$ and the two matrices $\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}$ and $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i}$ must be saved in order to compute $\mathbf{x}^{n+1}$ using this linearized representation.

Although it is possible to compute $\mathbf{x}^{n+1}$ from equation 2.24, this would require operations with very large matrices and vectors (e.g., $\mathbf{J}^{i+1}$ is of dimensions $2N_g \times 2N_g$) and it would not lead to significant computational savings. Instead, we now introduce the projection matrix $\mathbf{\Phi}$ discussed in Section 2.2.1. Specifically, $\mathbf{x}$ is now represented as $\mathbf{x} = \mathbf{\Phi}\mathbf{z}$ and equation 2.24 is projected into the reduced space by premultiplying by $\mathbf{\Phi}^T$. This gives

$$\mathbf{\Phi}^T\mathbf{J}^{i+1}\mathbf{\Phi}\left(\mathbf{z}^{n+1} - \mathbf{z}^{i+1}\right) = -\mathbf{\Phi}^T\left[\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\mathbf{\Phi}\left(\mathbf{z}^{n} - \mathbf{z}^{i}\right) + \frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right)\right]. \tag{2.25}$$

Defining

$$\mathbf{J}_r^{i+1} = \mathbf{\Phi}^T\mathbf{J}^{i+1}\mathbf{\Phi}, \tag{2.26}$$

$$\left(\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\right)_r = \mathbf{\Phi}^T \left(\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\right) \mathbf{\Phi}, \tag{2.27}$$

$$\left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)_r = \mathbf{\Phi}^T \left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right), \tag{2.28}$$

and rearranging, we can rewrite equation 2.25 more concisely as

$$\mathbf{z}^{n+1} = \mathbf{z}^{i+1} - \left(\mathbf{J}_r^{i+1}\right)^{-1} \left[\left(\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\right)_r \left(\mathbf{z}^n - \mathbf{z}^i\right) + \left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)_r \left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right)\right], \tag{2.29}$$

where subscript r denotes reduced. Note that while $\mathbf{J}^{i+1}$ is a sparse matrix of dimensions $2N_g \times 2N_g$, $\mathbf{J}_r^{i+1}$ is a full matrix of dimensions $\ell \times \ell$, where $\ell << N_g$. It is therefore feasible (and generally fast) to invert $\mathbf{J}_r^{i+1}$. Similarly, $\left(\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^i\right)_r$ is of dimensions $\ell \times \ell$ and $\left(\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}\right)_r$ is a matrix of dimensions $\ell \times N_w$.

2.3 Detailed treatments of various terms

2.3.1 Constructing the required matrices

As indicated above, high fidelity simulations are performed using Stanford's General Purpose Research Simulator GPRS [23, 24]. The simulator was modified to output the solution snapshots $\mathbf{x}^{i+1}$, the Jacobian matrices $\mathbf{J}^{i+1}$, the derivative matrices $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^i$, and selected fluid properties (ρ_o, H_o, μ_o and B_o) at every timestep.

Unlike the Jacobian matrix, the matrix $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^i$ is not needed in the solution of the governing system of equations in GPRS. Therefore, it would require some modification of the simulator to construct this matrix. However, $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i+1}$, which is a part of the Jacobian matrix (see equation 2.22), is computed at every timestep in GPRS. It is in fact possible to use this matrix to reconstruct $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^i$. Indeed, vector $\mathbf{A}^{i+1}$ is constructed by combining the accumulation terms from the mass conservation equation (2.4) and the energy conservation equation (2.10). For grid block

j, this vector is written as

$$\mathbf{A}_j^{i+1} = \frac{1}{\Delta t^{i+1}} \begin{bmatrix} \left(\frac{\phi}{B_o}\right)_j^{i+1} - \left(\frac{\phi}{B_o}\right)_j^{i} \\ E_j^{i+1} - E_j^i \end{bmatrix}, \tag{2.30}$$

where $\Delta t^{i+1} = t^{i+1} - t^i$ and $E = (\phi \rho_o U_o + (1-\phi)\, \rho_R U_R)$ is the total internal energy in block j. Now, the block-diagonal matrix $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i+1}$ (here, each block is of dimensions 2×2) contains the derivatives of the accumulation terms with respect to the state variables at timestep $i+1$. For grid block j, the corresponding block-element in matrix $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i+1}$ is

$$\left(\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i+1}}\right)_j = \frac{\mathbf{1}}{\Delta t^{i+1}} \frac{\partial}{\partial \mathbf{x}_j^{i+1}} \begin{bmatrix} \left(\frac{\phi}{B_o}\right)_j^{i+1} \\ E_j^{i+1} \end{bmatrix}. \tag{2.31}$$

Similarly, for grid block j, the corresponding diagonal block-element in $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i}$ is

$$\left(\frac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^{i}}\right)_j = -\frac{\mathbf{1}}{\Delta t^{i+1}} \frac{\partial}{\partial \mathbf{x}_j^{i}} \begin{bmatrix} \left(\frac{\phi}{B_o}\right)_j^{i} \\ E_j^{i} \end{bmatrix}. \tag{2.32}$$

Thus we see that matrix $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i}$ can be computed by multiplying the matrix $\partial \mathbf{A}^{i}/\partial \mathbf{x}^{i}$ (which is just $\partial \mathbf{A}^{i+1}/\partial \mathbf{x}^{i+1}$ at the previous timestep) by $-\Delta t^i/\Delta t^{i+1}$.

2.3.2 Representation of well derivatives

Well derivative matrix $\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}$ is not required in GPRS and thus it is not recorded directly during the training runs. Instead we compute it using the solution snapshots and the saved fluid properties. For the derivatives of the well term in grid block j with respect to the controls in grid block j (derivatives with respect to

other grid blocks are zero), the specific expressions used in our TPWL implementation are

$$\left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)_j = \begin{bmatrix} \left(\frac{\partial Q_m}{\partial p^w}\right)_j^{i+1} \\ \left(\frac{\partial Q_e}{\partial p^w}\right)_j^{i+1} \end{bmatrix} = \begin{bmatrix} (W_i \lambda_o)_j^{i+1} \\ (\rho_o H_o W_i \lambda_o)_j^{i+1} \end{bmatrix}. \tag{2.33}$$

Once all of the derivative matrices are available (saved Jacobians, saved accumulation derivatives and reconstructed well derivatives), these matrices are projected into the reduced-order space, using the projection matrix $\mathbf{\Phi}$, as described in equations 2.26, 2.27 and 2.28. However, as noted in Section 2.2.2, the dimension of vector $\mathbf{Q}$ can be taken as $2N_w$. It follows that matrix $\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}$ is of dimensions $2N_w \times N_w$. In consequence, when projecting $\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}$ into the reduced space, only the rows of matrix $\mathbf{\Phi}$ that correspond to grid blocks with producer wells are included. This new projection matrix is called $\mathbf{\Phi}_w$ and appears as follows:

$$\left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)_r = \mathbf{\Phi}_w^T \left(\frac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right), \tag{2.34}$$

where $\mathbf{\Phi}_w$ is of dimensions of $2N_w \times \ell$. The matrix $\left(\partial \mathbf{Q}^{i+1}/\partial \mathbf{u}^{i+1}\right)_r$, of dimensions $\ell \times N_w$, is then used in equation 2.29. Note that $\mathbf{\Phi}_w^T$ replaces $\mathbf{\Phi}^T$ appearing in equation 2.28.

2.3.3 Determination of closest saved state

As discussed previously, application of the TPWL procedure requires running one or more high-fidelity training simulations. Then, as described in Section 2.2.2, the 'closest' saved state to the current solution must be determined at every step of the procedure. There are many ways to determine which saved state to use. If only one training run is performed, during which data are saved at every timestep, a simple approach is to use the saved state at the timestep corresponding to the

timestep being simulated. Although this choice may not provide the closest saved state mathematically, we have found it to be a practical and robust choice. This is the approach applied in Chapter 3 when a single training run is used.

When more than one training run is performed, it is necessary to apply a different approach. We achieved accurate and robust results by computing distances based on (high-fidelity) temperatures of grid blocks containing producer wells. This approach is applied in the examples in Chapter 3 that involve multiple training runs. This treatment is of course heuristic and other approaches are possible.

2.3.4 Treatment of heater wells

The special treatment used to model heater wells in this work requires some modifications prior to applying the TPWL procedure. Since heater wells are modeled with grid blocks at (essentially) constant temperatures, they appear as boundary conditions in the governing system of equations. However, we would like, in effect, to use these temperatures as control variables. Therefore, we need to maintain the ability to change their values at any given time.

In order to achieve this, the small fraction of grid blocks that contain heater wells is not projected into the reduced space when applying TPWL. This is achieved with a method analogous to the local high-resolution procedure described by He et al. [21]. In this case, however, only the heater well block temperatures need be kept at the high-resolution level. If we let $\mathbf{x}_H = \left[T_{H_1}, \ldots, T_{H_{N_H}}\right]^T$ designate the full-order temperatures associated with the heater well blocks, where T_H designates the heater temperature and N_H is the number of heater well blocks, and $\mathbf{x}_G$ the full-order states for all other variables, then we write

$$\begin{bmatrix} \mathbf{x}_H \\ \mathbf{x}_G \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_H & 0 \\ 0 & \mathbf{\Phi}_G \end{bmatrix} \begin{bmatrix} \mathbf{z}_H \\ \mathbf{z}_G \end{bmatrix}, \tag{2.35}$$

where $\mathbf{\Phi}_H$ is the identity matrix (thus $\mathbf{z}_H = \mathbf{x}_H$) and $\mathbf{\Phi}_G$ is found using POD, as described in Section 2.2.1.

We will use equation 2.35 to project our linearized representation of the governing equations (2.24) into reduced space. First we isolate the equations corresponding to heater well blocks. The system can now be written as:

$$\begin{cases} \mathbf{x}_H^{n+1} = \mathbf{u}_H^{n+1} \\ \mathbf{J}_G^{i+1}\left(\mathbf{x}_G^{n+1} - \mathbf{x}_G^{i+1}\right) = -\left(\dfrac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\right)_G \left(\mathbf{x}_G^n - \mathbf{x}_G^i\right) + \left(\dfrac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)\left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right), \end{cases} \tag{2.36}$$

where the first line represents the equations solved in the heater well blocks and $\mathbf{u}_H^{n+1}$ contains the target temperatures at timestep $n+1$ (specified by the user) for these grid blocks. The second line in equation 2.36 shows the linearized governing equations for all other grid blocks (denoted by the subscript G). We then proceed by representing the high-fidelity state variables using equation 2.35 and by premultiplying by

$$\tilde{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{\Phi}_G \end{bmatrix}. \tag{2.37}$$

This gives:

$$\begin{cases} \mathbf{x}_H^{n+1} = \mathbf{u}_H^{n+1} \\ \mathbf{z}_G^{n+1} = \mathbf{z}_G^{i+1} - \left(\mathbf{J}_{G,r}^{i+1}\right)^{-1}\left[\left(\dfrac{\partial \mathbf{A}^{i+1}}{\partial \mathbf{x}^i}\right)_{G,r}\left(\mathbf{z}_G^n - \mathbf{z}_G^i\right) + \left(\dfrac{\partial \mathbf{Q}^{i+1}}{\partial \mathbf{u}^{i+1}}\right)_r \left(\mathbf{u}^{n+1} - \mathbf{u}^{i+1}\right)\right], \end{cases} \tag{2.38}$$

where the subscript r denotes reduced and G indicates that rows and columns associated with heater wells have been removed. It is evident from equation 2.38 that the heater well blocks are not projected into the reduced space. They can therefore be

specified by the user at any timestep.

Note that we can also apply the local high-resolution procedure described by He et al. [21] to resolve production well blocks. This approach, which is used in the examples in Chapter 3, is compatible with the treatment of heater wells described in this section. For this case, equation 2.35 becomes:

$$\begin{bmatrix} \mathbf{x}_H \\ \mathbf{x}_P \\ \mathbf{x}_G \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_H & 0 & 0 \\ 0 & \mathbf{\Phi}_P & 0 \\ 0 & 0 & \mathbf{\Phi}_G \end{bmatrix} \begin{bmatrix} \mathbf{z}_H \\ \mathbf{z}_P \\ \mathbf{z}_G \end{bmatrix}, \tag{2.39}$$

where $\mathbf{x}_P$ contains the producer well state variables, and $\mathbf{\Phi}_P$ is taken to be the identity matrix (thus $\mathbf{x}_P = \mathbf{z}_P$). Appropriate modifications also appear in equations 2.36 and 2.38. In the examples of Chapter 3, the local high-resolution procedure is applied to both pressure and temperature of the heater well blocks and producer well blocks.

Chapter 3

Numerical examples

In this chapter we apply the TPWL procedure to two reservoir models – one containing 35,200 grid blocks and the other containing 75,000 grid blocks. The simulations entail solution of the pressure and energy equations discussed in Chapter 2. Both models use realistic permeability distributions, generated using the software SGeMS [29].

3.1 Reservoir model 1

3.1.1 Model description

This reservoir model, shown in Figure 3.1, is three-dimensional and contains 35,200 grid blocks. The number of grid blocks in each direction is $n_x = 40$, $n_y = 40$ and $n_z = 22$. Each grid block is of dimensions 5 ft $\times$ 5 ft in the horizontal plane and 3 ft in the vertical direction. Thus the overall model is rather small, of dimensions 200 ft $\times$ 200 ft $\times$ 66 ft. There are four producer wells (designated P1-P4) that are completed in the five deepest layers of the reservoir (layers 18-22). There are also 21 heater wells that are completed in layers 8-22. Note that the heater blocks are

considerably larger than actual heater wells, so our constant-temperature treatment of heater blocks overpredicts the input of energy into the model.

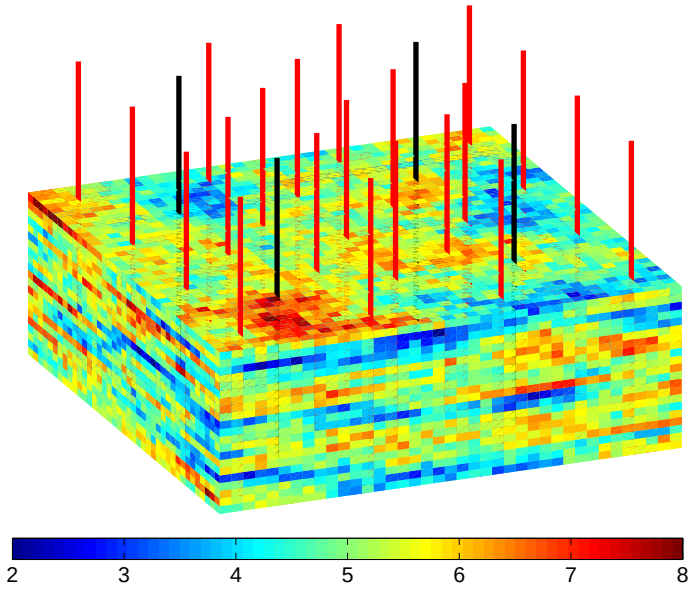


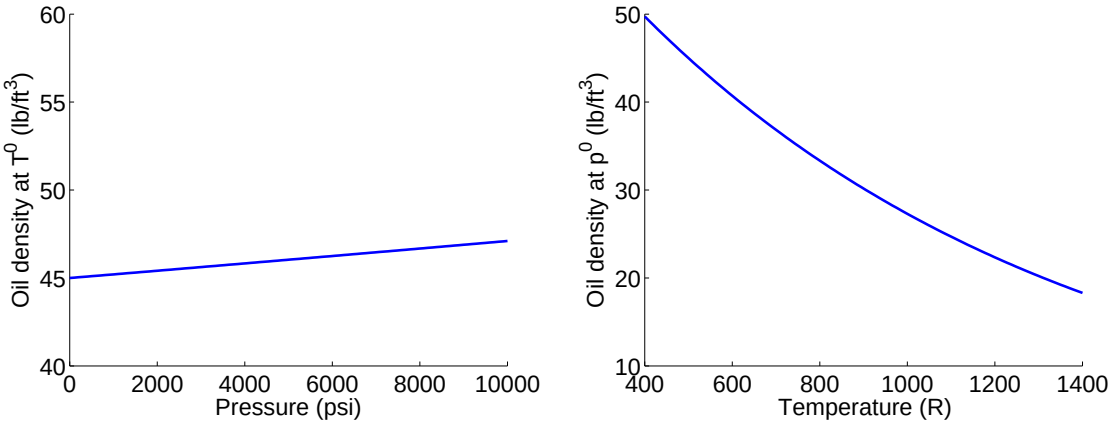
Figure 3.1: Reservoir model 1 showing log permeability (mD), production wells (black) and heater wells (red)

Permeability is taken to be a diagonal tensor ($k_x = k_y = k_z$) and follows a log-normal distribution with a mean value of 140 mD and a variance $\sigma_{\text{logk}} = 2$. Porosity is set constant in all grid blocks and equal to 0.3. The fluid is modeled with a single oil component in one phase. The system is compressible, with density varying with pressure and temperature as follows:

$$\rho_o\left(p,T\right) = \rho^0 \exp\left(c_p\left(p-p^0\right) - c_T\left(T-T^0\right)\right), \tag{3.1}$$

where $\rho^0 = 45$ lb/ft^3 is the reference density of oil at reference pressure $p^0 = 14.7$ psi and reference temperature $T^0 = 500°$R. The coefficient c_p is the oil compressibility

with pressure, taken to be constant and equal to 4.57×10^{-6} psi^{-1}, and the coefficient c_T is the oil compressibility with temperature, taken to be constant and equal to 10^{-3} °R^{-1}. The variation of density with pressure and temperature is depicted in Figure 3.2.



(a) Density versus pressure

(b) Density versus temperature

Figure 3.2: Oil density versus pressure (at 500 °R) and temperature (at 14.7 psi) for reservoir model 1

Oil viscosity varies with temperature, according to the following expression [30]:

$$\mu_o(T) = a \exp\left(\frac{b}{T - T^0}\right), \tag{3.2}$$

where μ_o is in units of centipoise, $a = 0.7$ and $b = 600$ are two empirical coefficients, and T and T^0 are as defined above. The variation of viscosity with temperature is shown in Figure 3.3. Note that, over the range $T = 540$°R to $T = 1400$°R, μ_o varies by a factor of about 1000.

3.1.2 Training simulation

As discussed in Section 2.2.2, the first step in the TPWL procedure is to perform training simulations using the high-fidelity model to generate the states, Jacobian

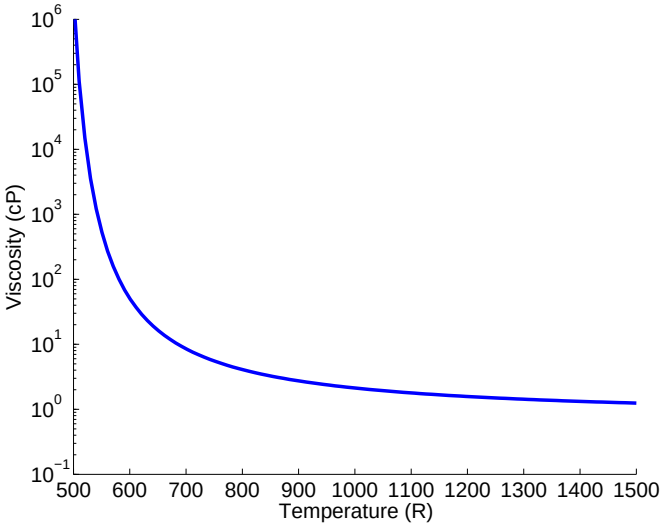
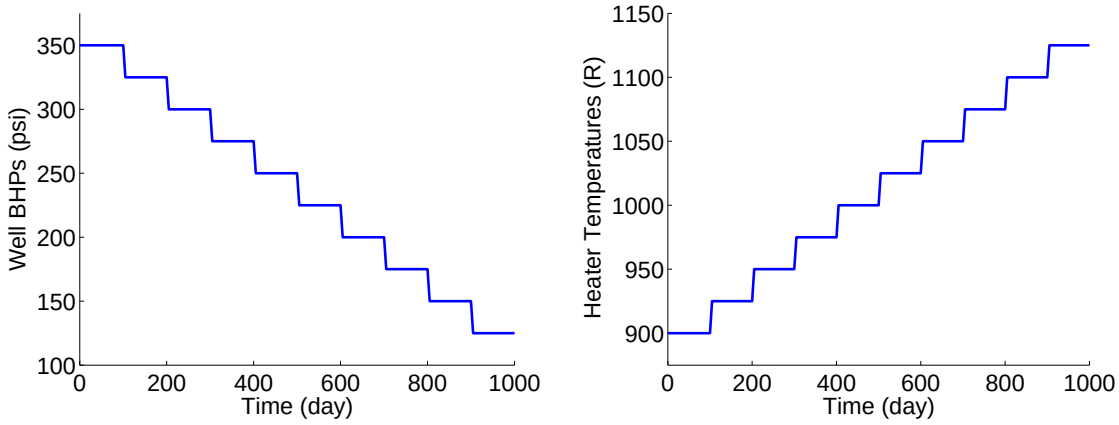


Figure 3.3: Variation of oil viscosity with temperature for reservoir model 1

matrices and other derivative matrices. For the first set of results, only one full-order training run is performed with GPRS. The training controls (BHP and heater temperature) are set for ten periods of 100 days each, for a total of 1000 days of production. All four producers have the same BHP training schedules and all 21 heaters have the same temperature training schedules. As displayed in Figure 3.4, producer well training BHPs are prescribed to decrease with time, while heater training temperatures are prescribed to increase with time. If the well settings to be used in the testing stage differ considerably from those shown in Figure 3.4, then the TPWL model may lose accuracy. In this case, a new training run should be performed. In the training run, the maximum timestep is 5 days. In total, 208 pressure and temperature snapshots are recorded, and the reduced basis $\mathbf{\Phi}$ is computed as described in Section 2.2.1. It contains 24 columns, of which 12 correspond to pressure states and 12 correspond to temperature states. This basis matrix captures 99.99% of the system energy for the pressure snapshots and 99.997% of the system energy for the



(a) Producer well BHP training schedule (b) Heater well temperature training schedule

Figure 3.4: Training schedules for production wells and heater wells for reservoir model 1

temperature snapshots (recall that POD is applied separately for pressure and temperature). For both variables, a local high-resolution method is applied for heater well blocks and producer well blocks, following the procedure described in [21], as discussed in Section 2.3.4. In consequence, 630 (2 states $\times$ 21 heaters $\times$ 15 completions) columns corresponding to high-resolution states in the heater well blocks and 40 (2 states $\times$ 4 wells $\times$ 5 completions) columns corresponding to high-resolution states in the producer well blocks are added to the matrix. In total, the matrix $\mathbf{\Phi}$ thus contains 694 columns.

We now present the simulation results obtained with the high-fidelity model (GPRS) using the control schedules described above. Figure 3.5 shows the oil production rates for the four producer wells. Figure 3.6(a) and (b) show the producer well block temperatures and pressures respectively, for the upper completion block (layer 18) of each well. In each graph, results from the high-fidelity training runs are depicted by continuous blue lines. In addition, we assess the ability of TPWL to reproduce results from the training runs by applying the same BHP and temperature schedules as those used for the training simulation. We see from these figures that

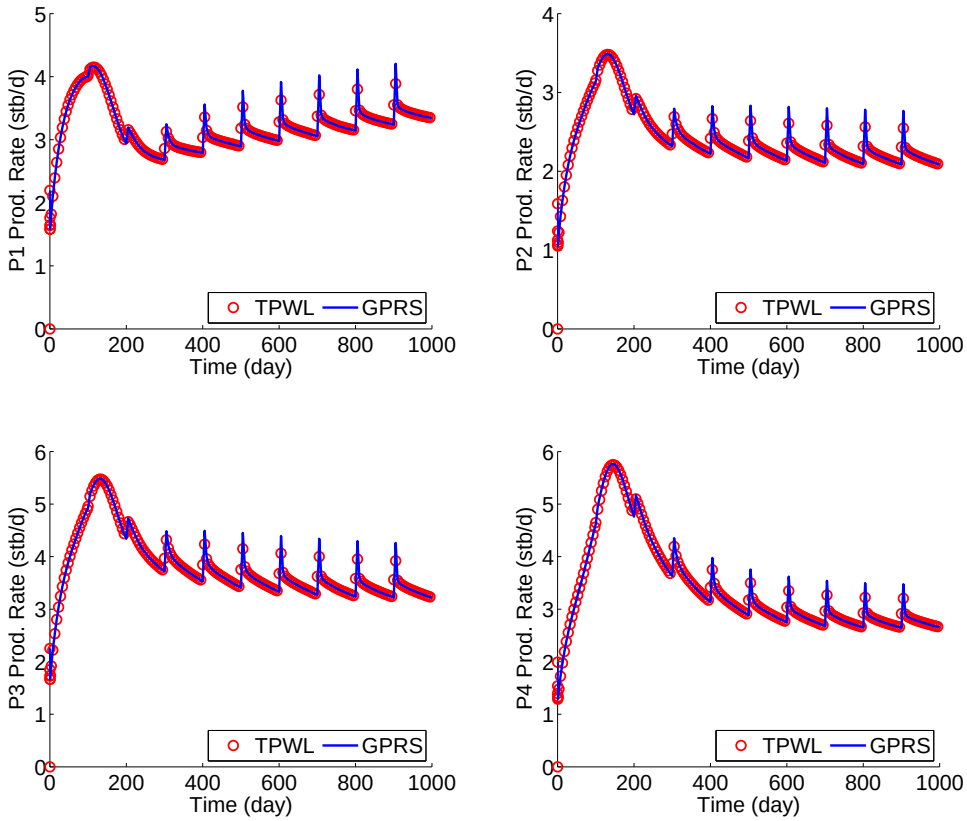
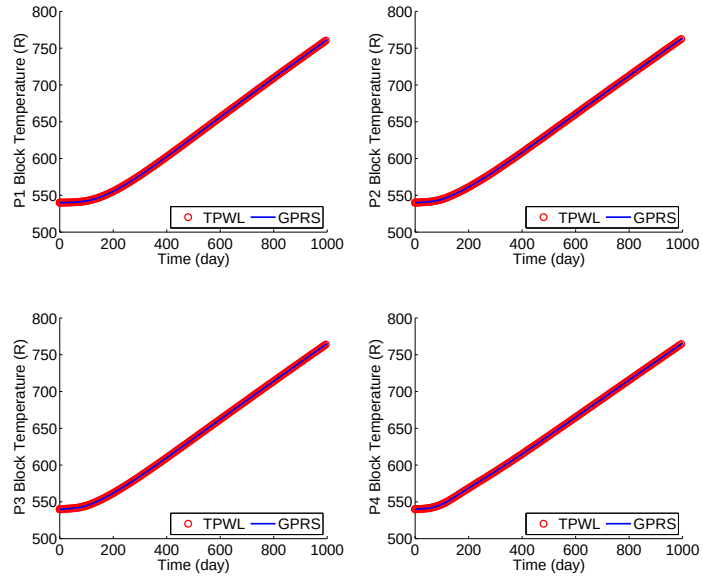


Figure 3.5: Oil production rates for the training run for reservoir model 1

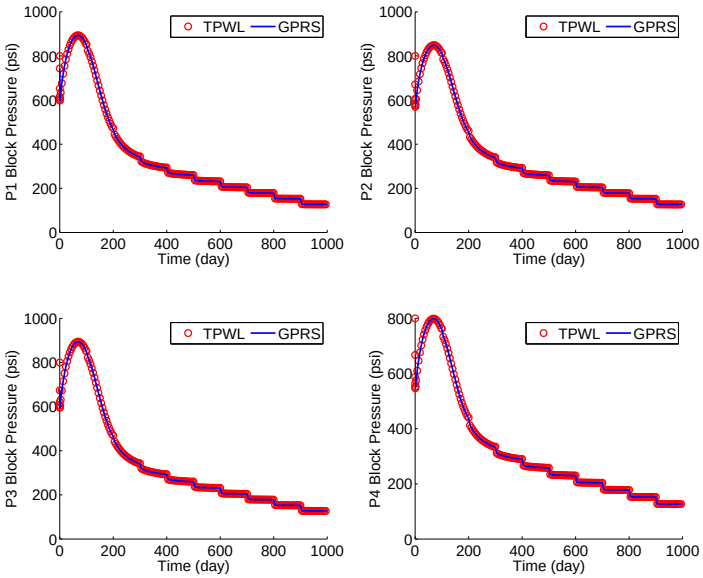
the TPWL results, depicted by red circles, are in essentially perfect agreement with the reference high-fidelity (GPRS) solutions, as would be expected for the training case.

In Figure 3.5, oil flow rates of all producer wells show a rapid increase, followed by a decrease at early production time. During this period there is an increase in reservoir pressure, evident in Figure 3.6(b). This is due to the increase in oil volume with increasing temperatures. The high reservoir pressure, together with the propagation of heat into the reservoir (which lowers oil viscosity), boosts the production of oil. Oil production peaks at about 3 to 6 barrels per day, depending on the producer well.

Following the initial production peak, viscosity continues to decrease and pressure decreases rapidly as the reservoir is depleted. In this model, since we use a constant value of heat capacity over the entire domain, differences in well production are due to the heterogeneous permeability field and the relative distances to heater well blocks. Subsequent changes in the BHP controls result in local peaks in production. These peaks are evident about every 100 days after the first 200 days of production (Figure 3.5). The decreasing pressure of the producers along with the increasing reservoir temperature acts to sustain the oil production over the course of the simulation.



(a) Producer well block temperatures for the training run



(b) Producer well block pressures for the training run

Figure 3.6: Producer well block (layer 18) temperatures and pressures for the training run for reservoir model 1

3.1.3 TPWL results and sensitivity analysis

In this section we assess the performance of the TPWL procedure for the reservoir model described above. We define six new heater temperature schedules. As depicted in Figure 3.7, these schedules differ by varying degrees from the training temperature schedule. For example, temperature schedule 4 represents a relatively small deviation from the training schedule, while temperature schedules 2 and 7 deviate significantly. Similarly, we define seven new BHP schedules that also differ from the training BHPs (see Figure 3.8). Since the TPWL procedure is based on a piecewise linearization, we expect that the accuracy of the results will decrease as the testing schedules increasingly deviate from the training schedules.

We now perform a sensitivity analysis by applying the TPWL procedure for all combinations of heater temperature schedules and producer well BHP schedules. This represents 56 distinct TPWL simulations. The accuracy of each simulation is then assessed quantitatively by computing the average error, relative to the reference high-fidelity simulation, for the oil production rate. The error computation follows the procedure presented in [4]. Rates from the TPWL simulation are interpolated to fit the simulation times of the corresponding high-fidelity simulation. Then, for each producer well k, we compute the high-fidelity oil production rate Q^k_{hf}, and the TPWL oil production rate Q^k_{tpwl}, at every timestep. The time-average error for each well, designated E_k, is calculated as the average of the absolute differences, normalized by the time-average oil flow rate $\bar{Q}^k_{\text{hf}}$ for the well. Error is then given by

$$E_k = \frac{1}{N_t \bar{Q}^k_{\text{hf}}} \sum_{i=1}^{N_t} \left| Q^{k,i}_{\text{hf}} - Q^{k,i}_{\text{tpwl}} \right|, \tag{3.3}$$

where the superscript i represents the timestep at which each flow rate is computed and N_t is the total number of timesteps. The total error E is then calculated as the

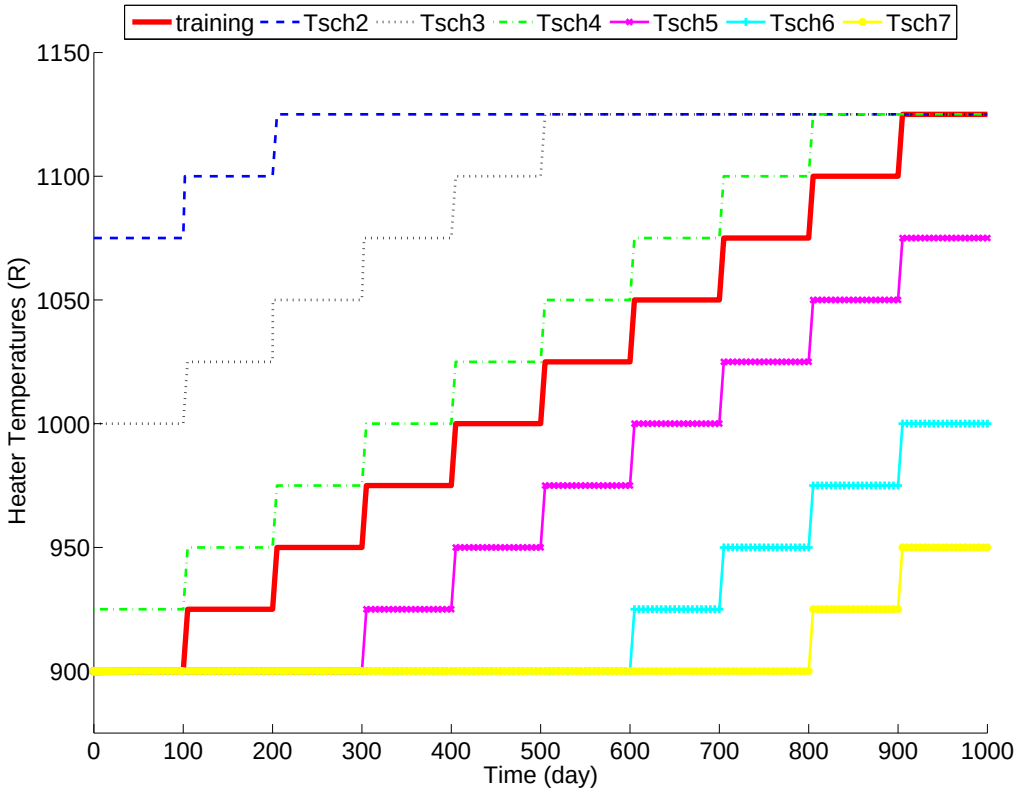


Figure 3.7: Heater temperature schedules for testing runs

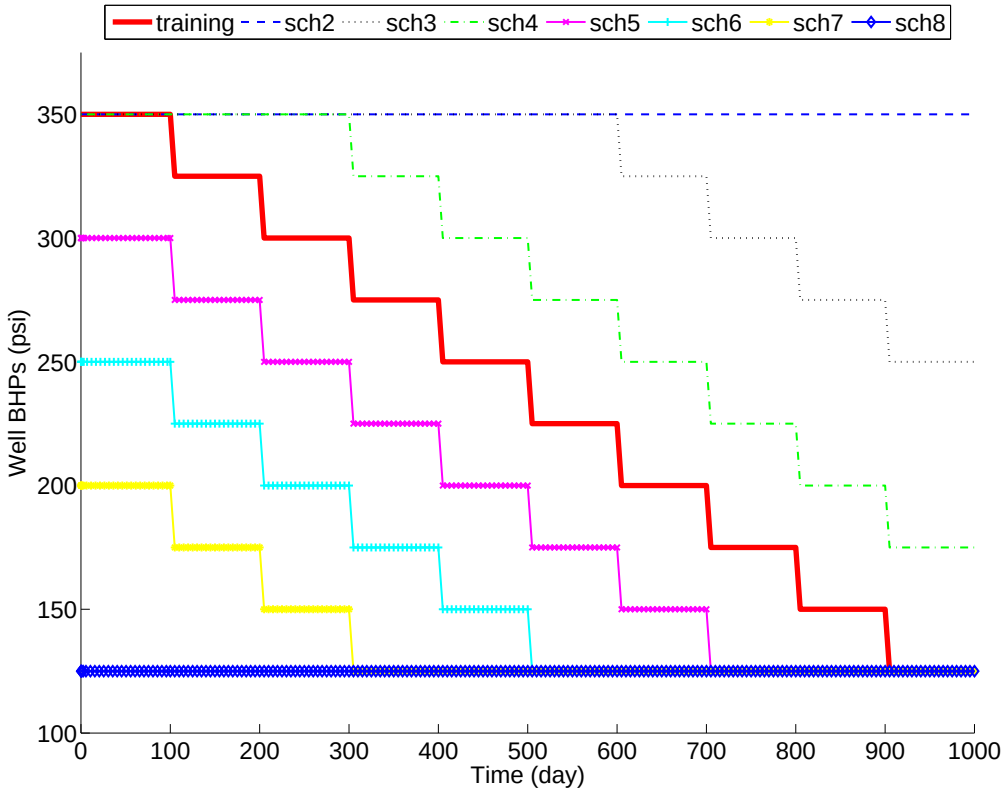


Figure 3.8: Producer well BHP schedules for testing runs

arithmetic average of the errors of all wells. Specifically:

$$E = \frac{1}{N_w} \sum_{k=1}^{N_w} E_k, \tag{3.4}$$

where N_w is the total number of producer wells (four in this case).

Table 3.1: Error for oil production rates for reservoir model 1

T sch	BHP 1	BHP 2	BHP 3	BHP 4	BHP 5	BHP 6	BHP 7	BHP 8
1	**0.0009**	0.0018	0.0014	0.0010	0.0013	0.0016	0.0020	0.0025
2	0.2249	0.2269	0.2255	0.2244	0.2245	0.2250	0.2328	0.2347
3	0.0689	0.0695	0.0690	0.0687	0.0726	0.0736	0.0752	0.0739
4	0.0065	0.0060	0.0062	0.0063	0.0069	0.0072	0.0077	0.0080
5	0.0068	0.0075	0.0073	0.0070	0.0070	0.0073	0.0076	0.0086
6	0.0256	0.0272	0.0266	0.0259	0.0261	0.0267	0.0273	0.0285
7	0.0420	0.0441	0.0431	0.0424	0.0426	0.0433	0.0441	0.0454

Errors computed using equation 3.4 are compiled in Table 3.1. The training simulation, which has the lowest error, is indicated in bold. Each row in this table displays the TPWL error for a fixed heater temperature schedule, while each column gives the error for a particular BHP schedule. It is evident that the errors are relatively similar on a given row, though they vary along a given column. This indicates that varying BHP schedules over the range considered has a relatively limited impact on the accuracy of the TPWL procedure for this reservoir model. TPWL production rate results for temperature schedule 1, BHP schedule 8 are shown in Figure 3.9. The TPWL production curves match closely those from the reference high-fidelity simulation. This high level of agreement is consistent with the small error (0.25%) for this case.

The next set of production results, shown in Figure 3.10, are for temperature schedule 6 and BHP schedule 5. For this case, the relative error is about 2.6%. It

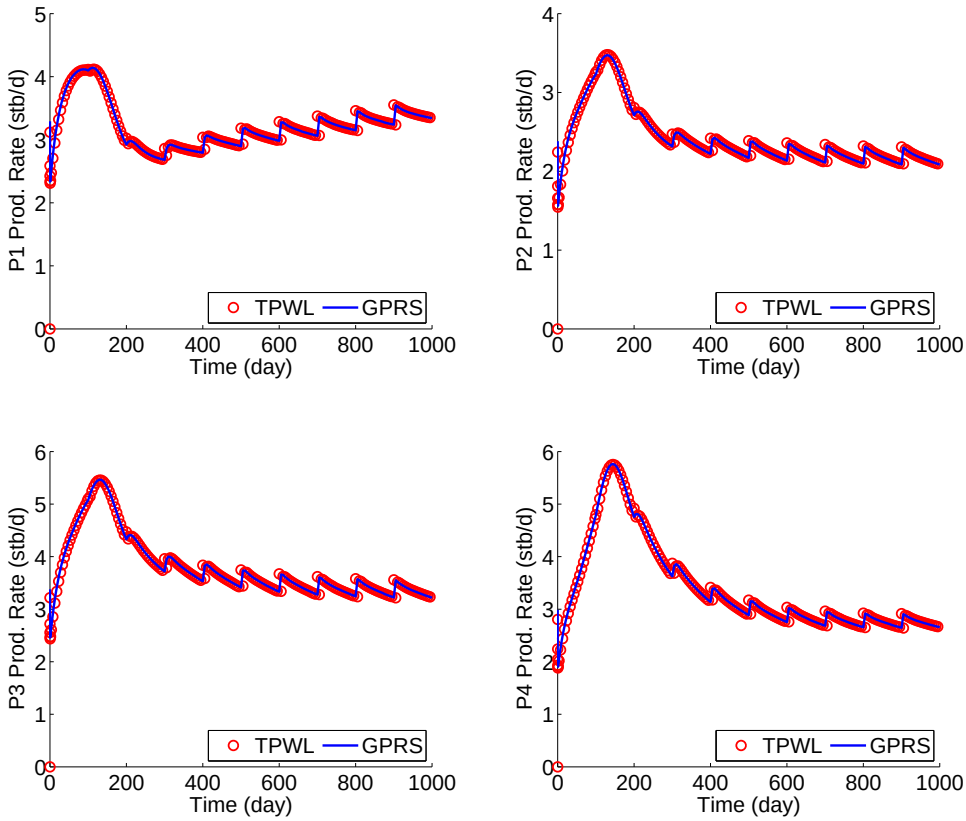


Figure 3.9: Oil production rates using temperature schedule 1 and BHP schedule 8

is noteworthy that, despite the fact that both the temperature and BHP schedules differ from the training schedules, TPWL provides results of high accuracy. We next consider a case with moderate error – temperature schedule 3 and BHP schedule 3 (relative error of 6.9%). Results for this case are shown in Figure 3.11. Here we see that, although TPWL provides an accurate solution at early time, by around 300 days, clear discrepancy in production rates is evident. We note that, as is evident from Table 3.1, error is even larger for TPWL runs with temperature schedule 2. These results illustrate the potential loss of accuracy for testing runs with controls that differ significantly from those used in the training runs.

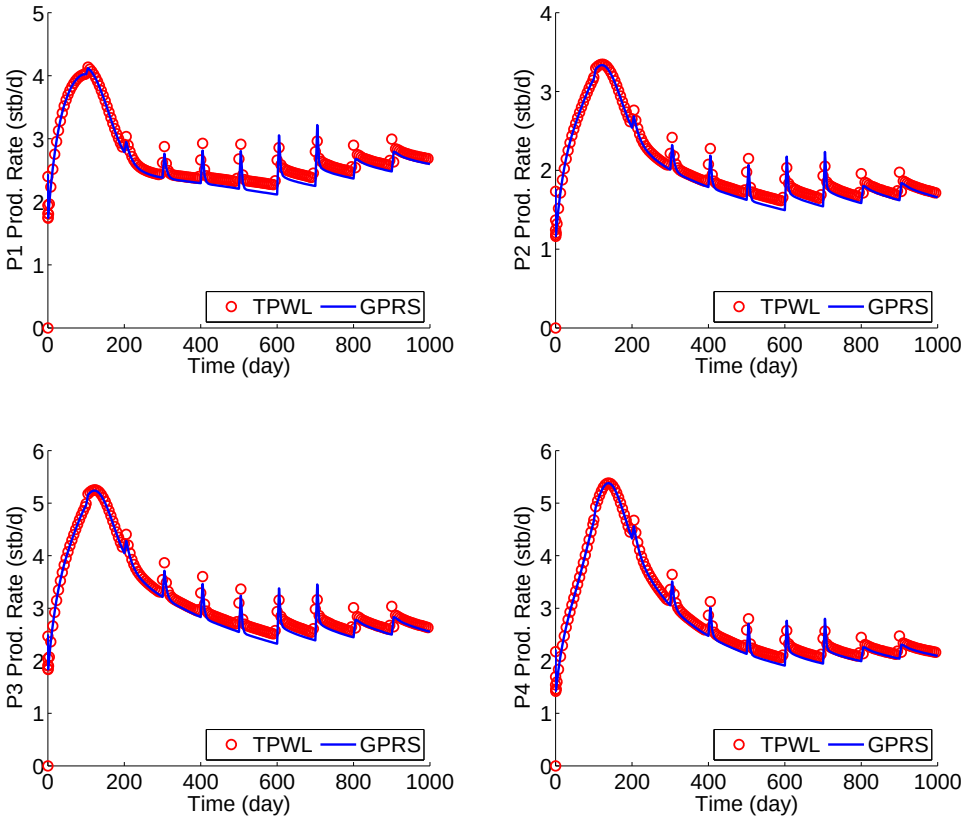


Figure 3.10: Oil production rates using temperature schedule 6 and BHP schedule 5

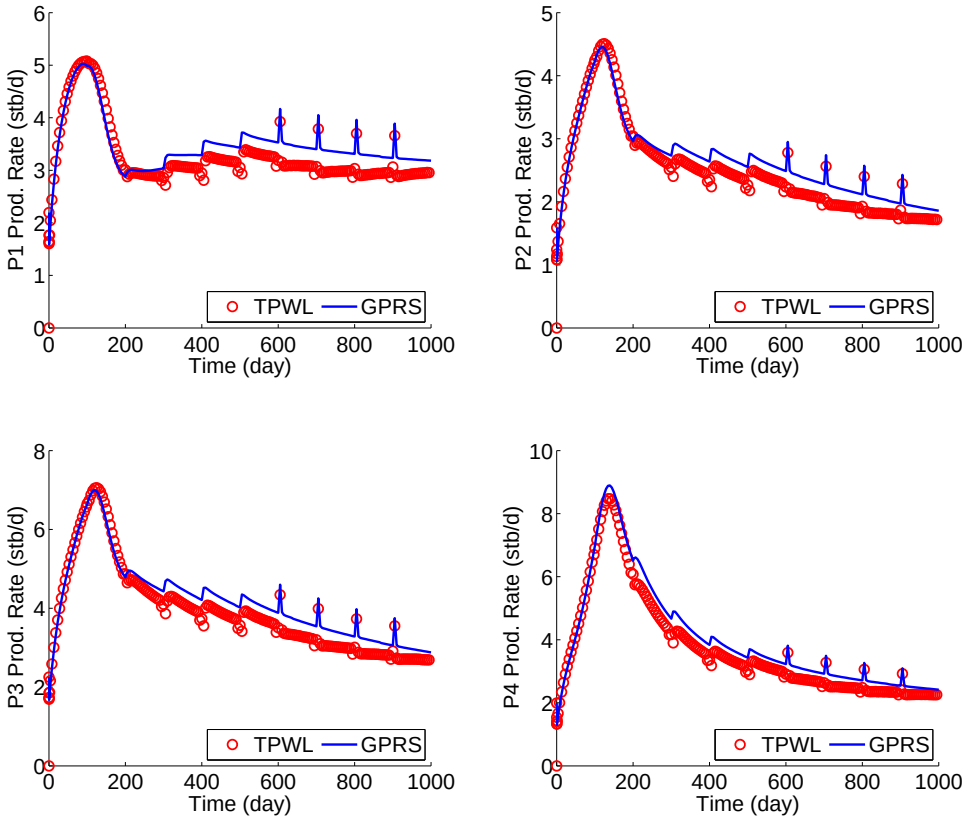
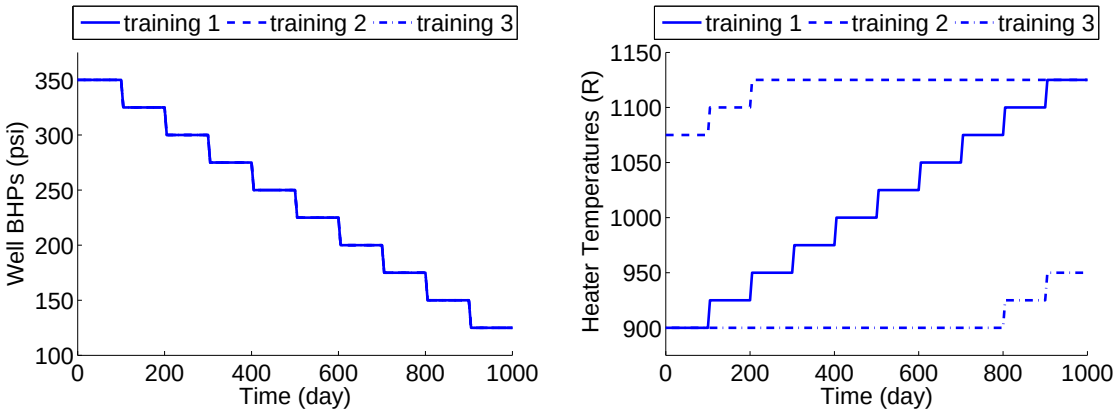


Figure 3.11: Oil production rates using temperature schedule 3 and BHP schedule 3

3.1.4 TPWL results with multiple training runs

We now seek to improve the accuracy of TPWL by using a multiple training approach. In addition to the training simulation presented in Section 3.1.2, we run two new training runs using BHP schedule 1 (as defined in Figure 3.8), and heater temperature schedules 2 and 7 (as defined in Figure 3.7). These new training schedules are shown in Figure 3.12.



(a) Producer well BHP training schedule (b) Heater well temperature training schedules

Figure 3.12: Training schedules for multiple training approach for reservoir model 1

Each of the three training simulations generates 208 pressure and temperature snapshots. The reduced basis $\mathbf{\Phi}$ is again computed with POD, as described in Section 2.2.1. The matrix $\mathbf{\Phi}$ now contains 47 columns, of which 21 correspond to pressure states and 26 to temperature states. This basis matrix captures the same percentage of the system energy as for the single training case considered above. Again, we use local high-resolution for heater well blocks and producer well blocks, resulting in a total of 670 extra columns. Thus in total, $\mathbf{\Phi}$ contains 717 columns.

By selecting the training schedules to span a large range of temperatures, we expect to capture model behavior over a wider range of testing schedules. We show in Table 3.2 the relative errors between results found with TPWL, using multiple

Table 3.2: Error for oil production rates for the multiple training approach for reservoir model 1

T sch	BHP 1	BHP 2	BHP 3	BHP 4	BHP 5	BHP 6	BHP 7	BHP 8
1	**0.0009**	0.0018	0.0014	0.0010	0.0053	0.0085	0.0108	0.0049
2	**0.0013**	0.0030	0.0025	0.0018	0.0019	0.0056	0.0089	0.0107
3	0.0152	0.0153	0.0150	0.0149	0.0148	0.0157	0.0155	0.0203
4	0.0101	0.0088	0.0084	0.0097	0.0106	0.0104	0.0115	0.0105
5	0.0074	0.0076	0.0075	0.0074	0.0122	0.0156	0.0164	0.0117
6	0.0026	0.0027	0.0027	0.0028	0.0081	0.0117	0.0142	0.0080
7	**0.0003**	0.0018	0.0013	0.0009	0.0060	0.0108	0.0160	0.0068

training runs, and the corresponding reference high-fidelity simulations. These errors are indeed much smaller than those in Table 3.1. Again, TPWL reproduces the training cases with high accuracy (errors for those cases are indicated in bold in Table 3.2). Errors for all cases are now less than 2%. Note in particular that the errors for temperature schedule 2 are now less than 1%, in contrast to the 22% errors for these cases when a single training was used. These significant reductions in error result because temperature schedule 2 is now used for training.

Production rates using temperature schedule 3 and BHP schedule 3 are shown in Figure 3.13. For this case, the relative error was reduced from 6.9% to 1.5% by switching to a multiple training approach. The results in Figure 3.13 are clearly more accurate than those in Figure 3.11, where a single training run was used.

For reservoir model 1, the high-fidelity (GPRS) reference simulations require about 2500 seconds to run. For both the single and multiple training approaches, TPWL simulations run in about 6 seconds. This results in runtime speedups of a factor of about 400. Note that our TPWL implementation was performed using Matlab, which may not render the best runtime speedups achievable. The overhead computational requirement is equivalent to less than three high-fidelity reference simulations for the single training case. For the multiple training approach, using three high-fidelity

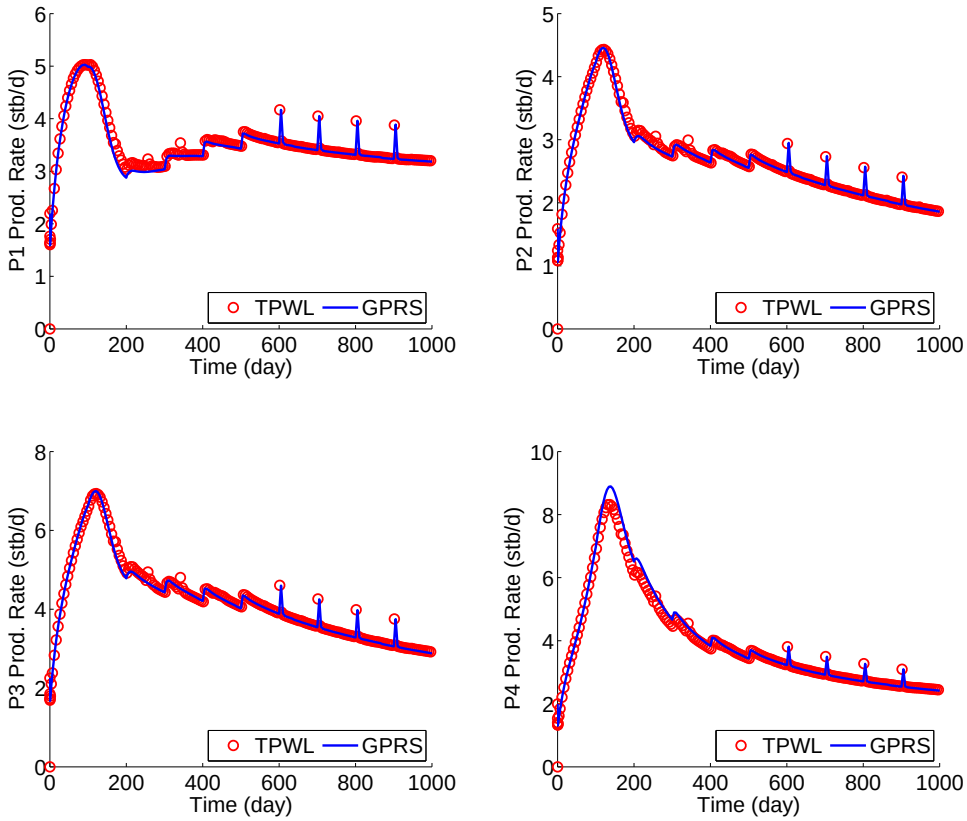


Figure 3.13: Oil production rates using temperature schedule 3 and BHP schedule 3, using multiple training simulations

training runs, the overhead work requirement is equivalent to about six high-fidelity reference simulations. Each training run entails about twice the computational work of running a reference simulation. The extra time is due to the need to save the required (high-fidelity) data and to perform some TPWL pre-processing. Some of these overhead computations are performed using Matlab, and we expect that they could be accelerated.

Increasing the number of training runs provides more saved states in the TPWL procedure. Thus, this approach requires an efficient and robust point selection

method. Here we use a distance-based selection method based on the high-fidelity producer well grid block temperatures, as described in Section 2.3.3. Although our implementation was found to be reasonably robust, this is a heuristic approach and improved treatments may be possible.

3.2 Reservoir model 2

In this section, we apply the TPWL procedure to a more challenging example. This case involves a larger reservoir model and increased nonlinearities in pressure and temperature. We assess the performance of TPWL for this example using a multiple training approach.

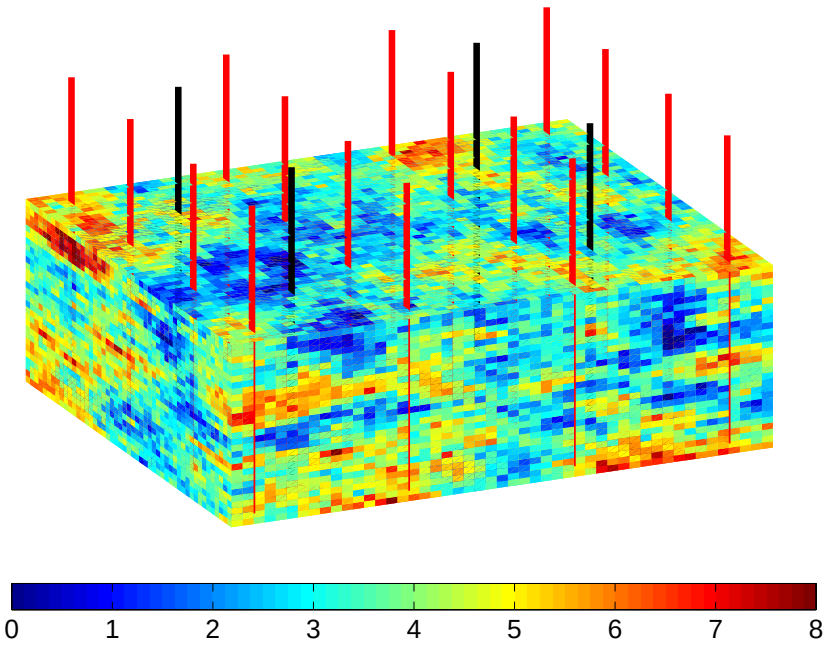
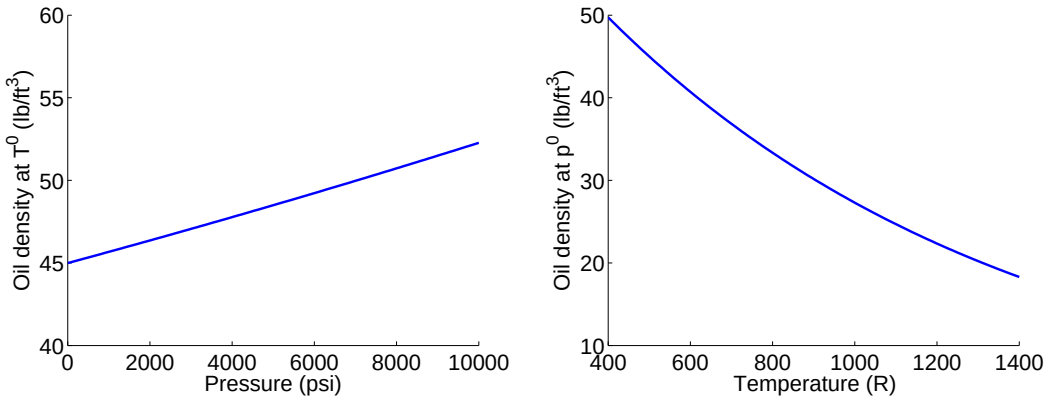


Figure 3.14: Reservoir model 2 showing permeability (mD), producer wells (black) and heater wells (red)

3.2.1 Model description

Reservoir model 2 (Figure 3.14) contains 75,000 grid blocks, with $n_x = 50$, $n_y = 50$ and $n_z = 30$. For this model, grid block dimensions are $\Delta x = 5$ft, $\Delta y = 5$ft and $\Delta z = 3$ft. There are four producer wells (designated P1-P4) that are completed in layers 20-26, and 16 heater wells in layers 11-30. Permeability follows a log-normal distribution with a mean value of 94 mD and a variance $\sigma_{\text{logk}} = 2$.



(a) Density versus pressure

(b) Density versus temperature

Figure 3.15: Oil density versus pressure (at 500 °R) and temperature (at 14.7 psi) for reservoir model 2

Porosity is set constant and equal to 0.3 in all grid blocks. The dependence of density on pressure is larger than in reservoir model 1; here we use $c_p = 10^{-3}$ psi^{-1} rather than $c_p = 4.57 \times 10^{-6}$ psi^{-1}, which was used in reservoir model 1. The variation of oil density with temperature and pressure is plotted in Figure 3.15. Viscosity is again modeled using the correlation in equation 3.2, now with $a = 0.5$ and $b = 800$. The resulting curve is shown in Figure 3.16. For the example considered below, this correlation leads to a variation in viscosity over about four orders of magnitude during the first 1000 days of production. This is one order of magnitude more variation than what was observed in reservoir model 1.

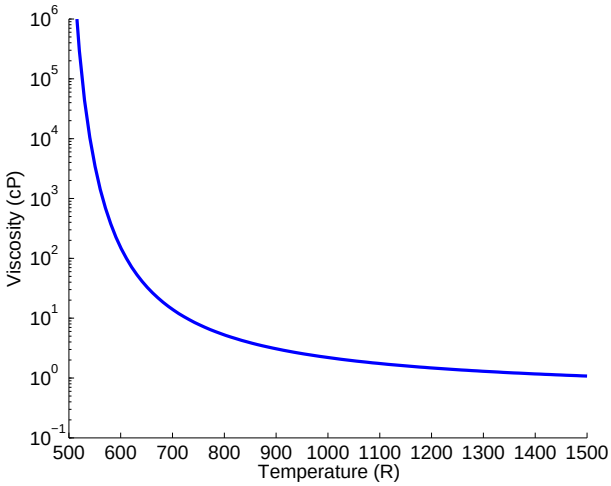
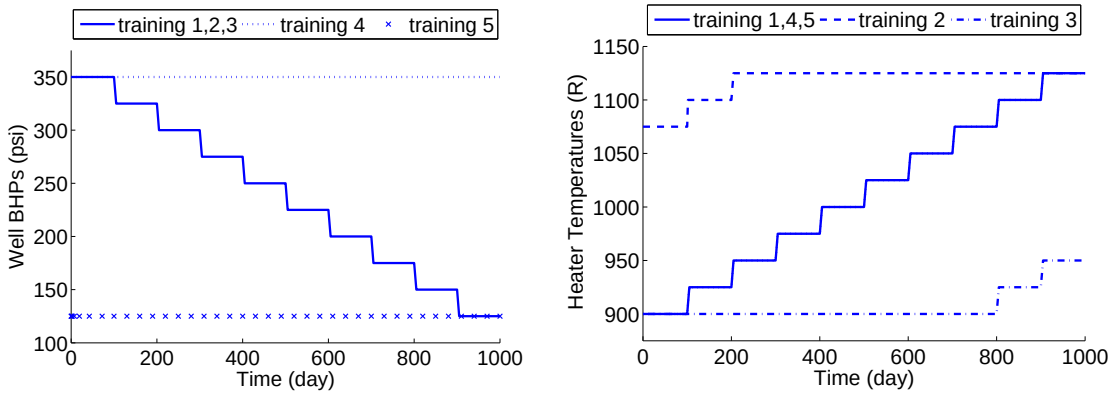


Figure 3.16: Variation of oil viscosity with temperature for reservoir model 2

3.2.2 Training simulations

We perform five training runs for this case. For the first training run, BHPs of the four producer wells are set to follow BHP schedule 1, as defined in Figure 3.8. Heater temperatures for the heater well blocks are set to follow temperature schedule 1, shown in Figure 3.7. Two additional training simulations are performed using BHP schedule 1, along with temperature schedules 2 and 7. The last two training runs are performed using temperature schedule 1 along with BHP schedules 2 and 8. The controls used for these five training simulations are plotted in Figure 3.17. By setting these training simulations to cover a wide range of pressures and temperatures, our intent is to construct a TPWL model that can capture a wide range of behavior.

The training simulations are performed using a maximum timestep of 5 days. This results in 1132 saved states for the 5 training runs (about 226 timesteps/run on average). The basis matrix $\mathbf{\Phi}$, found by applying POD, contains a total of 146 columns, of which 90 correspond to pressure states and 56 to temperature states. This basis matrix captures 99.99% of the system energy for pressure and 99.997% of



(a) Producer well BHP training schedules (b) Heater well temperature training schedules

Figure 3.17: Training schedules for multiple training approach for reservoir model 2

the system energy for temperature. In addition, a local high-resolution procedure is applied for both states, in all heater well blocks and all producer well blocks. This results in a total of 696 (2 states $\times$ 16 heaters $\times$ 20 completions + 2 states $\times$ 4 producers $\times$ 7 completions) additional columns. The final matrix $\mathbf{\Phi}$ used in the TPWL procedure thus has a total of 842 columns.

3.2.3 TPWL results and sensitivity analysis

Following the same procedure used with reservoir model 1, we apply TPWL using all combinations of the control schedules defined in Figures 3.7 and 3.8. Relative errors with respect to the corresponding high-fidelity runs are compiled in Table 3.3, with the training cases indicated in bold.

These errors are found to be relatively low over the range of BHP and temperature schedules considered. Indeed, a majority of the cases leads to errors smaller than 2%, indicating that TPWL results are in close agreement with the corresponding reference high-fidelity simulation. We show, for example, oil production rates obtained with

Table 3.3: Error for oil production rates for the multiple training approach for reservoir model 2

T sch	BHP 1	BHP 2	BHP 3	BHP 4	BHP 5	BHP 6	BHP 7	BHP 8
1	**0.0000**	**0.0000**	0.0218	0.0218	0.0087	0.0025	0.0002	**0.0000**
2	**0.0010**	0.0673	0.0599	0.0262	0.0197	0.0253	0.0217	0.0190
3	0.0128	0.0210	0.0491	0.0230	0.0227	0.0240	0.0240	0.0242
4	0.0039	0.0037	0.0277	0.0208	0.0078	0.0064	0.0075	0.0073
5	0.0230	0.0566	0.0456	0.0296	0.0144	0.0138	0.0134	0.0134
6	0.0024	0.0489	0.0331	0.0126	0.0092	0.0109	0.0121	0.0122
7	**0.0004**	0.0520	0.0359	0.0143	0.0075	0.0099	0.0105	0.0106

TPWL using temperature schedule 7 and BHP schedule 6, along with the corresponding high-fidelity results in Figure 3.18. For this case, the relative error is about 1%. Figure 3.19 displays results obtained by setting the well BHPs to follow schedule 5 and the heater temperatures to follow schedule 3. For this case, although both BHP and temperature schedules differ from the training schedules, TPWL results are again in close agreement with those obtained from high-fidelity simulations. The relative error is about 2.3%. Note also that these oil production rates differ considerably from those shown in Figure 3.18, which demonstrates the ability of the reduced model to capture a range of physical behaviors.

A few TPWL simulation results, however, show some mismatch with the reference high-fidelity runs (errors between 5% and 7%). We show, for example, in Figure 3.20, a comparison of oil production rates found with TPWL using temperature schedule 2 and BHP schedule 3. For this case, TPWL error is about 6%. Though the main production trends are reproduced, additional training is probably needed for this case, depending on the level of accuracy required.

For reservoir model 2, the high-fidelity (GPRS) reference simulations run in about 6,000 seconds. TPWL simulations require less than 12 seconds, which means they provide runtime speedups of a factor of about 500. The total overhead computational

requirements represent the equivalent of about ten high-fidelity runs (two for each training run used). We reiterate that it would only make sense to use TPWL if many sensitivity runs (or simulations in an optimization procedure) are to be performed.

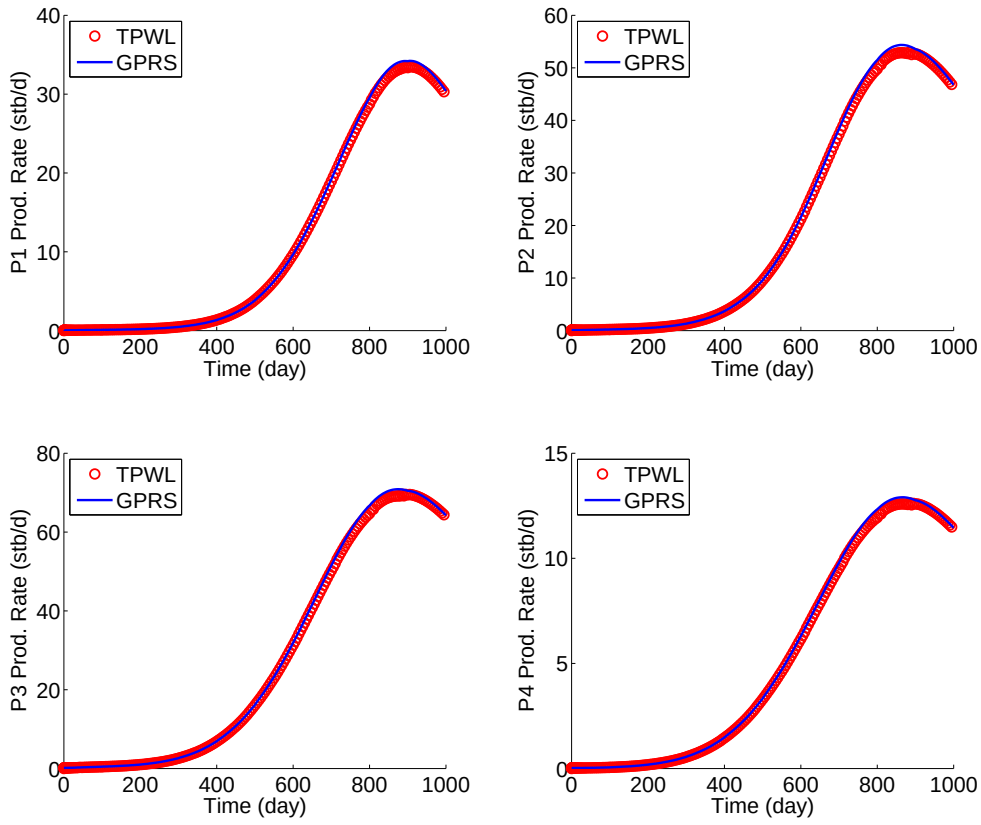


Figure 3.18: Oil production rate using temperature schedule 7 and BHP schedule 6 for reservoir model 2

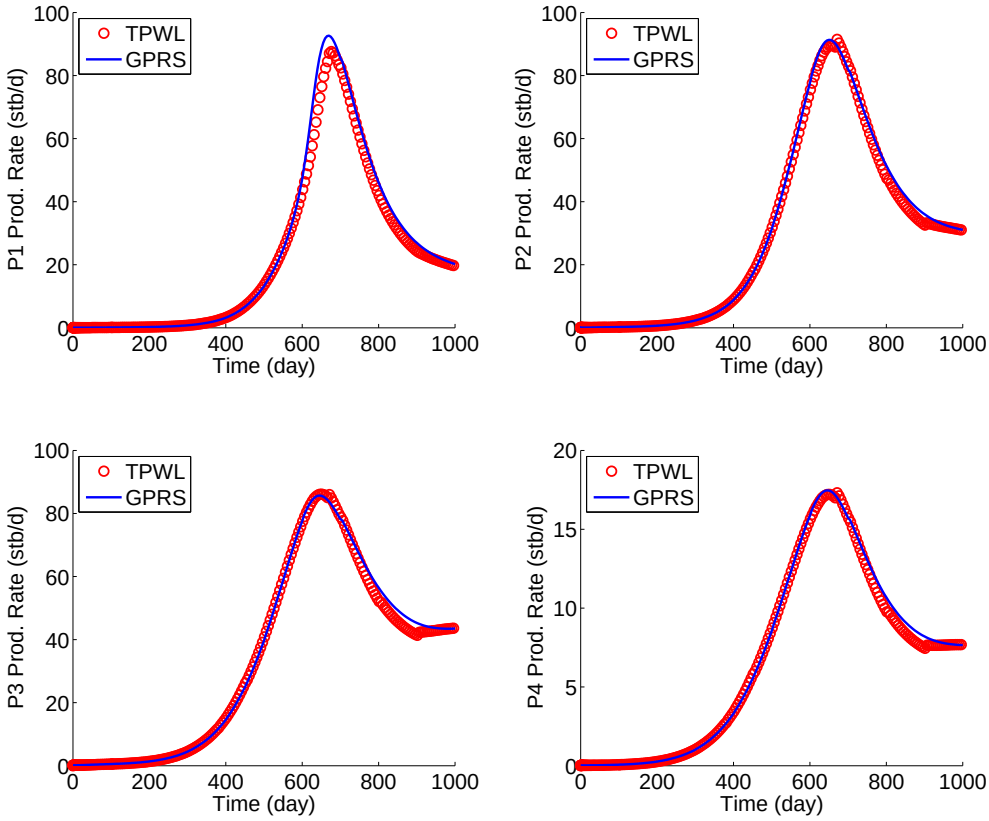


Figure 3.19: Oil production rate using temperature schedule 3 and BHP schedule 5 for reservoir model 2

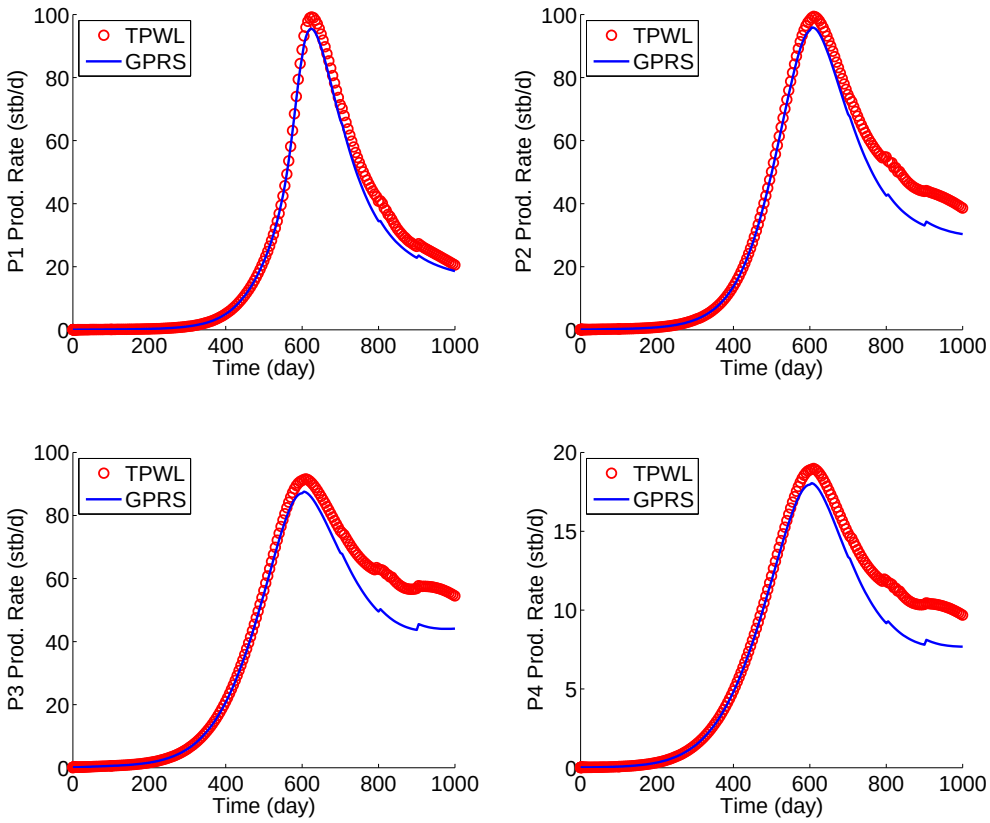


Figure 3.20: Oil production rate using temperature schedule 2 and BHP schedule 3 for reservoir model 2

Chapter 4

Conclusions and future work

In this thesis, we have applied the reduced-order modeling procedure trajectory piecewise linearization (TPWL) to nonlinear thermal reservoir simulation problems. The two examples considered involved realistic permeability distributions, and contained 35,200 and 75,000 grid blocks respectively. The main outcomes of this work are as follows.

- The TPWL method was extended to thermal problems involving single-phase flow coupled to an energy equation. Downhole heater wells were modeled by fixing the temperature of selected grid blocks.

- For the first reservoir model, density and viscosity were set to vary significantly with pressure and temperature. A TPWL representation of this model was generated from a single high-fidelity training simulation and tested with a wide range of well BHP and heater temperature controls. TPWL results were found to be in close agreement with the respective high-fidelity runs when the controls were close to those used in the training simulation. For the most extreme test cases, the accuracy of the TPWL model was shown to degrade, in some cases quite considerably.

- TPWL accuracy was restored, however, by using multiple training simulations. With this approach, relative errors between results obtained from the TPWL model and those obtained from the high-fidelity runs were reduced to less than 2% for all test cases.

- The multiple training approach was also used for the second reservoir model. For this case, the nonlinear dependencies in viscosity and density were greater than in the first model. Nonetheless, for this more challenging case, TPWL continued to display a reasonable level of accuracy.

- Significant runtime speedups were achieved for the examples considered in this thesis. For reservoir model 1, the runtime speedup was a factor of about 400, and for reservoir model 2, the runtime speedup was a factor of about 500. The overhead computational requirements, in our implementation, were equivalent to approximately two runs of the high-fidelity model for each training run used.

The models considered in this work are much simpler than those used in realistic situations, and this work represents a preliminary assessment of the applicability of TPWL for thermal problems. Our conclusion is that TPWL is well suited for the solution of the idealized models considered here. In consequence, its application to more complex thermal reservoir models should be considered. Following are some suggestions for improvement and future work.

- Heat injection was achieved, in the current formulation, by setting an entire grid block at (essentially) constant temperature. For this approach to be accurate, grid blocks must be the same size as heater wells. New heater well models [28] have been developed recently, and they should be integrated into our TPWL formulation. This will enable the accurate representation of downhole heaters in models with large grid blocks.

- For the multiple training approach, we have used a heuristic procedure for the selection of the closest saved state to the current TPWL state. This approach was based on the difference in well block temperatures between the current TPWL solution and the saved state. Our technique was shown to be reasonably robust, but it may be possible to develop more accurate approaches.

- In the formulation developed here, oil was modeled with a single component in a single phase. A multiphase model, involving the presence of water and gas in the formation, should be formulated and tested. Also, complex phenomena such as chemical reactions and geomechanical effects were neglected. Such physics, along with multiple components, could be gradually introduced into the TPWL model in order to reproduce the actual in-situ upgrading process more accurately.

- In light of its potential for the idealized thermal problems used in this work, other thermal applications could be considered. The so-called steam assisted gravity drainage (SAGD) process could be an interesting candidate for the application of TPWL. This would require the implementation of several of the capabilities noted in the previous point.

- The application of TPWL to more challenging and realistic thermal problems would naturally lead to its use in optimization. Coupling TPWL to gradient-free optimization methods is relatively straightforward, although an appropriate retraining strategy must be developed. A successful implementation could enable optimizations of complex thermal processes.

Nomenclature

Abbreviations

BHP	bottom hole pressure
GPRS	General Purpose Research Simulator
ICP	in-situ conversion process
IUP	in-situ upgrading process
POD	proper orthogonal decomposition
SVD	singular value decomposition
TPWL	trajectory piecewise linearization

Variables

$\mathbf{A}$	accumulation term
a, b	coefficients in viscosity correlation
B	formation volume factor
C	specific heat capacity
c_p	compressibility with respect to pressure
c_T	compressibility with respect to temperature
D	depth
E	total internal energy in a grid block, or TPWL error
$\mathbf{F}$	flux term
$\mathbf{g}$	vector of residual equations

g	gravitational acceleration
H	specific enthalpy
$\mathbf{J}$	Jacobian matrix
$\mathbf{k}$	permeability tensor
ℓ	dimension of the reduced space
N_g	number of grid blocks
N_H	number of heater well blocks
N_w	number of wells
p	pressure
$\mathbf{Q}$	source/sink term
Q_{hf}	high-fidelity oil production rate
Q_{tpwl}	TPWL oil production rate
q	volumetric flow rate
$\tilde{q}$	volumetric flow rate per unit volume
$\tilde{q}^H$	heat injection rate per unit volume
r_0	equivalent well radius
r_w	wellbore radius
T	temperature
t	time
U	internal energy
$\mathbf{u}$	vector of controls
$\mathbf{u}_o$	Darcy velocity of oil
V	grid block volume
W	well index

$\mathbf{X}$	matrix of snapshots
$\mathbf{x}$	vector of states
$\mathbf{z}$	reduced vector of states

Greek symbols

$\mathbf{\Phi}$	POD basis matrix
κ	thermal conductivity
λ	mobility
ϕ	porosity
ρ	density
σ	singular value
Υ	transmissibility

Superscripts

0	reference value
ν	Newton iteration
i	timestep level of saved data
n	timestep level
T	transpose
w	well

Subscripts

e	contribution from energy conservation equation
G	global
H	heater well block
i	grid block number
m	contribution from mass conservation equation

o	oil
p	pressure
R	rock
r	reduced
T	temperature

Bibliography

[1] *Shell Energy Scenarios to 2050*. www.shell.com/scenarios, 2008.

[2] R.C. Johnson, T.J. Mercier, M.E. Brownfield, and J.G. Self. Assessment of in-place oil shale resources of the Green River Formation, Uinta Basin, Utah and Colorado. *U.S. Geological Survey Fact Sheet 2010-3010*, 2010. http://pubs.usgs.gov/fs/2010/3010/.

[3] R.C. Ryan, T.D. Fowler, G.L. Beer, and V. Nair. *Shell's In Situ Conversion Process-From Laboratory to Field Pilots*, volume 1032, chapter 9, pages 161–183. American Chemical Society Symposium Series, 2010.

[4] M.A. Cardoso. *Development and application of reduced-order modeling procedures for reservoir simulation*. PhD thesis, Stanford University, 2009.

[5] M.A. Cardoso and L.J. Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229:681–700, 2010.

[6] J.L. Lumley. Atmospheric turbulence and radio wave propagation. *Journal of Computational Chemistry*, 23(13):1236–1243, 1967.

[7] D. Zheng, K.A. Hoo, and M.J. Piovoso. Low-order model identification of distributed parameter systems by combination of singular value decomposition and

the Karhunen-Loève expansion. *Industrial & Chemistry Research*, 41 (6):1545–1556, 2002.

[8] M. Meyer and H.G. Matthies. Efficient model reduction in non-linear dynamics using Karhunen-Loève expansion and dual-weighted-residual methods. *Computational Mechanics*, 31:179–191, 2003.

[9] T. Bui-Thanh, M. Damodoran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AAIA Journal*, 42 (8):1505–1516, 2004.

[10] Y. Cao, J. Zhu, Z. Luo, and I.M. Navon. Reduced order modeling of the upper tropical Pacific ocean model using proper orthogonal decomposition. *Computers & Mathematics with Applications Research*, 52 (8-9):1373–1386, 2006.

[11] P.T.M. Vermeulen, A.W. Heemink, and C.B.M.T. Stroet. Reduced models for linear groundwater flow models using empirical orthogonal functions. *Advances in Water Resources*, 27:57–69, 2004.

[12] J.F.M. van Doren, R. Markovinović, and J.D. Jansen. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences*, 10:137–158, 2006.

[13] R. Markovinović and J.D. Jansen. Accelerating iterative solution methods using reduced-order models as solution predictors. *International Journal for Numerical Methods in Engineering*, 68(5):525–541, 2006.

[14] M.A. Cardoso, L.J. Durlofsky, and P. Sarma. Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering*, 77 (9):1322–1350, 2009.

[15] J. Burkardt, M. Gunzburger, and H.C. Lee. Centroidal Voronoi tessellation-based reduced-order modeling of complex systems. *SIAM Journal of Scientific Computing*, 28 (2):459–484, 2006.

[16] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Eindhoven University of Technology, 2004.

[17] M.J. Rewienski. *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems*. PhD thesis, Massachusetts Institute of Technology, 2003.

[18] D. Gratton and K. Willcox. Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics. 34^{th} AIAA Fluid Dynamics Conference and Exhibit, Portland, Oregon, February 2004.

[19] Y.J. Yang and K.Y. Shen. Nonlinear heat-transfer macromodeling for MEMS thermal devices. *Journal of Micromechanics and Microengineering*, 15 (2):408–418, 2005.

[20] M.A. Cardoso and L.J. Durlofsky. Use of reduced-order modeling procedures for production optimization. *SPE Journal*, 15(2):426–435, 2010. SPE-119057-PA.

[21] J. He, J. Sætrom, and L.J. Durlofsky. Enhanced linearized reduced-order models for subsurface flow simulation. In review, 2010.

[22] J. He. *Enhanced linearized reduced-order models for subsurface flow simulation*. Master's thesis, Stanford University, 2010.

[23] Hui Cao. *Development of techniques for general purpose simulators*. PhD thesis, Stanford University, June 2002.

[24] Y. Jiang. *A flexible computational framework for efficient integrated simulation of advanced wells and unstructured reservoir models.* PhD thesis, Stanford University, 2007.

[25] K. Aziz and A. Settari. *Petroleum Reservoir Simulation.* Applied Science Publishers, 1979.

[26] D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *SPE Journal*, 23 (3):531–543, 1983.

[27] Y. Fan, L.J. Durlofsky, and H.A. Tchelepi. Numerical simulation of the in-situ upgrading of oil shale. *SPE Journal*, 15 (2):368–381, 2010. SPE-118958-PA.

[28] G. Aouizerate, L.J. Durlofsky, and P. Samier. New models for heater wells in reservoir simulation. Proceedings of the 12th European Conference on the Mathematics of Oil Recovery, Oxford, UK, September 2010.

[29] N. Remy, A. Boucher, and J. Wu. *Applied Geostatistics with SGeMS: A User's Guide.* Cambridge University Press, 2009.

[30] CMG STARS user guide. *Computer Modelling Group Ltd.*, 2008.