Reduced-Order Modelling for Production Optimisation

26-08-15  C.W. Hewson
Title : Reduced-Order Modelling for Production Optimisation

Author(s) : C.W. Hewson

Date : August 2015
Professor(s) : J.D. Jansen
Supervisor(s) : E.G.D. de Barros

Postal Address : Section for Petroleum Engineering
                Department of Geoscience & Engineering
                Delft University of Technology
                P.O. Box 5028
                The Netherlands

Telephone : (31) 15 2781328 (secretary)
Telefax : (31) 15 2781189

Copyright ©2015 Section for Petroleum Engineering

All rights reserved.
No parts of this publication may be reproduced,
Stored in a retrieval system, or transmitted,
In any form or by any means, electronic,
Mechanical, photocopying, recording, or otherwise,
Without the prior written permission of the
Section for Petroleum Engineering
Abstract

Production optimisation of a reservoir simulation problem can be very computationally demanding as the reservoir model may contain many variables and nonlinearities, thus many iterations may be needed to obtain an optimal production schedule. Using Proper Orthogonal Decomposition and Trajectory Piecewise Linearization (POD-TPWL) developed in [1] and [2], simulations were performed on the Delft Egg model 100-200 times faster than the high-fidelity simulation with reasonable accuracy, depending on the distance from the trained solution.

Production optimisation was performed using the gradient-based adjoint method. A reduced version of the adjoint equation was used by incorporating POD, as presented in [3] and by performing a first-order Taylor series expansion around a training point (similar to TPWL). This method allowed for time gains of 50-100 times when compared to the high-fidelity adjoint method. The results from the high-fidelity production optimisation compared with POD-TPWL showed a similar Net Present Value (NPV) for both optimisation methods, with an error of 0.1% between the two-values. However, the optimal injection schedules were not the same. When the high-fidelity model was run using the input schedules from POD-TPWL optimisation, the error in NPV was 4%. The speed-up observed for the optimisation loop using POD-TPWL was 6 times faster than the high-fidelity model. This is due to the number of snapshots that needed to be generated and the processing of the data from these snapshots.

A robust optimisation was performed on the Egg model ensemble using a POD-TPWL model incorporating geological model parameters, states and well controls. Results showed a 0.7% deviation from the mean NPV value calculated in MoReS and a 4 million dollar increase in the standard deviation of the NPV. POD-TPWL was able to complete the robust optimisation 25 times faster than a high-fidelity simulation.

POD-TPWL shows promise as a reduced-order modelling application for reservoir simulation and production optimisation. The accuracy needs to be improved in order to move to an operational application.
Acknowledgements

This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of TNO, Delft University of Technology, ENI, Statoil and Petrobras.

I would like to thank Dr. Jan Dirk Jansen for his invaluable advice and direction throughout the thesis. Somehow you were able to comprehend the blithering nonsense I was talking about and give me a very clear and concise research direction.

Eduardo Goncalves Dias de Barros, your daily support and constant availability was greatly appreciated throughout the thesis. Having somebody who was so thorough helped to solidify the research.

I would like to thank Dr. Hadi Hajibeygi and Dr. Joerg Benndorf for taking the time to read, evaluate and assess this work as a part of the thesis committee.

To my parents, brother and wonderful wife, Jenna, you have all supported me and been there for me during every meltdown, identity crisis and sleepless night. In the past two years I have completed 25 master level courses, 1 Norwegian Internship and a gruelling thesis, without your love and support all that would have never been possible, YNWA.
Contents

ABSTRACT .................................................................................................................. II
ACKNOWLEDGEMENTS ............................................................................................. III
1 INTRODUCTION ...................................................................................................... 1
  1.1 BACKGROUND ..................................................................................................... 1
    1.1.1 Closed-loop Reservoir Management ............................................................ 1
    1.1.2 Reduced Order Modelling .......................................................................... 2
  1.2 EGG MODEL DESCRIPTION ............................................................................ 3
  1.3 SCOPE OF WORK ............................................................................................. 6
  1.4 THESIS OUTLINE ........................................................................................... 6
2 POD-TPWL FOR RESERVOIR SIMULATION .................................................... 7
  2.1 RESERVOIR GOVERNING EQUATIONS ......................................................... 7
  2.2 TPWL ............................................................................................................... 9
    2.2.1 TPWL Results ............................................................................................ 10
  2.3 POD ................................................................................................................. 13
    2.3.1 POD-TPWL Results ................................................................................ 14
    2.3.2 Local Resolution Scheme ....................................................................... 16
    2.3.3 Projection Methods for POD-TPWL ....................................................... 18
  2.4 FLUID PROPERTY EFFECT ON POD-TPWL ACCURACY ............................. 19
3 REDUCED-ORDER MODELLING FOR ADJOINT-BASED OPTIMISATION .......... 23
  3.1.1 POD Adjoint ............................................................................................... 24
  3.1.2 POD-TPWL Adjoint .................................................................................. 24
4 PRODUCTION OPTIMISATION ........................................................................... 28
  4.1 OBJECTIVE FUNCTION .................................................................................. 28
  4.2 IMPLEMENTATION ......................................................................................... 28
    4.2.1 Retraining Criteria ................................................................................... 29
    4.2.2 Multiple Snapshot Pre-Training and Point Selection Strategy ................. 31
  4.3 POD-TPWL WATER FLOODING OPTIMISATION RESULTS ....................... 31
5 ROBUST OPTIMISATION ...................................................................................... 37
  5.1 POD-TPWL ROBUST OPTIMISATION ............................................................ 37
  5.2 IMPLEMENTATION ........................................................................................ 38
    5.2.1 Retraining Criteria ................................................................................... 38
  5.3 ROBUST OPTIMISATION RESULTS .............................................................. 39
6 CONCLUSION ....................................................................................................... 41
  6.1 RECOMMENDATIONS ..................................................................................... 41
REFERENCES .......................................................................................................... 43
APPENDIX A  HISTORY MATCHING.............................................................................. 46
A.1  POD-TPWL FORMULATION.............................................................................. 46
A.2  POD-TPWL HISTORY MATCHING RESULTS...................................................... 47
List of Figures

Figure 1 Closed-loop Reservoir Management as a combination of production optimisation and history matching. ................................................................. 2
Figure 2 One permeability realization of the Egg Model showing locations of the producers (red) and injectors (blue) [15] ................................................................. 4
Figure 3 Training and experimental schedule for TPWL simulation .................. 11
Figure 4 Well water injection rates compared for TPWL and HF simulation .......... 11
Figure 5 Oil production rates compared for TPWL and HF simulation .................. 12
Figure 6 Water production rates compared for TPWL and HF simulation .......... 12
Figure 7 BHP of the training simulation and POD-TPWL schedule compared .......... 15
Figure 8 Water production rates for a high-fidelity simulation and the POD-TPWL simulations retaining different amounts of energy for POD ......................................... 16
Figure 9 Water saturation profiles for layer 1 of realization 1 of the Egg Model using POD and LR-POD as compared to the high-fidelity model................................. 17
Figure 10 Relative permeabilities for the Egg model and simple fluid model .......... 19
Figure 11 Training run and simulation BHP schedule for both fluid property cases ...... 20
Figure 12 Oil production rate comparison for the regular Egg model fluid property case ........................................................................................................ 21
Figure 13 Oil production rate comparison for the simple fluid property case .......... 21
Figure 14 Adjoint gradient comparison between the high-fidelity, POD and POD-TPWL equations. Note 40 control steps were used, but for plotting purposes a step plot was used where control step 1 spans 1-2, step 2 spans 3-4, etc.......................... 26
Figure 15 Flow chart for POD-TPWL optimisation ........................................... 29
Figure 16 Training run and the input schedule for the retraining criteria determination ........................................................................................................ 30
Figure 17 Training run and optimised well schedules for the 8 injection wells, single training run ....................................................................................................... 32
Figure 18 Training run and optimised well schedules for the 8 injection wells, multiple training run ....................................................................................................... 33
Figure 19 Time comparison between the HF and POD-TPWL for the water flooding optimisation of the Egg model ........................................................................ 34
Figure 20 NPV comparison between the HF and POD-TPWL using water flooding optimisation ..............................................................34

Figure 21 Optimal well injection BHP schedules for the HF, POD-TPWL multiple training, and POD-TPWL single training optimisation ..........................................................35

Figure 22 Cumulative distribution function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL ........................................39

Figure 23 Probability density function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL ........................................39

Figure 24 Evolution of the objective function value for both the HF procedure (blue) and POD-TPWL procedure (red) ..........................................................48

Figure 25 Time for both the HF and ROM procedure broken down into individual components of the modelling process .........................................................48

Figure 26 Comparison of the oil rates after history matching for the POD-TPWL, HF and truth model ..........................................................49

Figure 27 Comparison of the water production rates after history matching for the ROM, HF and truth model ..........................................................50

Figure 28 Comparison of the log-permeability fields for the HF, ROM, prior and truth model ..........................................................50
List of Tables

Table 1 Reservoir fluid properties of the Egg model [15] ..............................................................5
Table 2 Error comparison between the TPWL simulation and the HF and training run. 13
Table 3 Cumulative Error in Production Rates using Dissimilar Number of Values in the
   POD Basis Functions ..................................................................................................................15
Table 4 Error in cumulative production values for LR-POD-TPWL ........................................17
Table 5 Comparison of Different POD-TPWL Projection Methods in terms of Time and
   Error from the High-fidelity Solution .....................................................................................18
Table 6 Cumulative production errors for both the standard Egg model and simple fluid
   properties ..................................................................................................................................20
Table 7 Time comparison from the three adjoint calculation comparisons .............................26
Table 8 Error between POD-TPWL and HF model when run with schedules different
   distances from the training image .............................................................................................30
Table 9 Economic parameters used for the NPV objective function .....................................32
Table 10 Cumulative production and injection errors from the HF optimal solution for
   single and multiple POD-TPWL training methods ..........................................................36
Table 11 Error of the water and oil production through the 4 production wells using
   varying values for the model parameters ............................................................................38
## 1 INTRODUCTION

In the past 20 years, reservoir simulation has enhanced the ability of the oil and gas industry to predict reservoir performance more precisely and with more accurate data obtained from the reservoir. Reservoir simulators traditionally solve a set of nonlinear equations to achieve a converged answer iteratively. The iterative nature of these solvers can lead to an exceedingly time consuming calculation, particularly when presented with millions of variables, as is the norm for field-scale problems.

Computational costs become even larger when used within the closed-loop reservoir management (CLRM) framework [4] [5] [6], which requires multiple simulations of the same geological model. CLRM involves both optimising the production and injection pressures (or injection rates) and history matching, where the mismatch between actual production data and simulated data is minimized.

A multitude of reservoir simulations are needed in order to determine the optimal well control strategies as well as to minimize the data mismatch. Depending on the optimisation algorithm, this process may take hundreds of extra reservoir simulations. The problem with a large number of optimisation variables is that a large number of iterations and function evaluations are required. This comprehensive study uses reduced order modelling (ROM) in order to decrease the reservoir simulation time and applies this method to production optimisation, history matching, robust optimisation and CLRM.

ROM is used in place of traditional nonlinear simulators in order to reduce the computational burden for applications such as production optimisation and history matching. There are a variety of options available for ROM and this work will focus on computationally efficient methods that still take into account the underlying physics and dynamics of the full-order reservoir simulation. ROM method used in this body of work will follow the approach outlined in [1] and [2].

### 1.1 Background

#### 1.1.1 Closed-loop Reservoir Management

Closed-Loop reservoir management (CLRM) has been studied in depth and has shown promise in its capability in order to improve reservoir performance [7] [6] [8] [9]. Using a combination of production optimisation and history matching, reservoir objectives are met using systems and control theory. The workflow diagram of the CLRM framework is shown in Figure 1.
Chapter 1 INTRODUCTION

CLRM optimises the reservoir based on economic, environmental and/or operational objectives. These objective functions are then used with the optimisation algorithms to calculate a search direction usually using the first derivative of the objective function with respect to the control inputs for gradient-based optimisation. The optimisation algorithms may include control variables such as well rates, well bottom hole pressures (BHPs), well location and inflow control device (ICD) settings. The number of control variables is directly proportional to the number of degrees of freedom available in the surface facilities. For example, in a well you can either control the BHP or rate, but not both. For history matching, the reservoir model parameters, such as porosities, permeabilities (or transmissibilities) and fault multipliers are changed using a similar method to the production optimisation in order to achieve the closest match to the true model (or production data from the true model), labelled as system in Figure 1. History matching is performed to reduce the model uncertainty that is inherent in geological models.

1.1.2 Reduced Order Modelling

Reduced order modelling (ROM) can be performed using grid-based methods, system based methods or snapshot methods. For grid-based methods, the number of grid blocks in the problem is reduced by constructing a reduced grid size and the problem is solved on a coarser grid. The reduced grid size is calculated using either an upscaling method or multiscale method. Upscaling methods have the disadvantage of only being able to produce results of the coarser grid and the finer grid resolution is thus lost [1]. Multiscale methods retain the finer scale by representing the coarser grid with a selection of basis functions (a set of functions that retain information of a higher order system but project the information into a lower dimensional space). In most multiscale methods, computation on the finer scale grid is required during the simulation, therefore limiting the potential speed up for a CLRM application. Work using multiscale methods in a CLRM framework has been done, presenting modest speedup results [10].

Figure 1 Closed-loop Reservoir Management as a combination of production optimisation and history matching
System based methods, such as Krylov subspace methods (calculating the inverse of a matrix by using a linear combination of its powers) and the balanced truncation methods (choosing the projection subspaces based on the controllability and observability of the underlying control system), use a time-invariant linear system and basis functions to a nonlinear problem to reduce the calculation required and the order of the system. Krylov subspace methods have been applied to reservoir simulation problems [11], these methods tend to create larger projection subspaces than are required. The resulting reduced order model has no guaranteed error bounds [11].

Balanced truncation techniques have probable error bounds and the stability of the original system is preserved in the reduced model [12]. Although successful results have been obtained in the reservoir setting using the balanced truncation technique, the computational cost is high, requiring the calculation of large square matrices and the inverse of these matrices [12].

Snapshot methods derive basis functions from a collection of time variant problems to create basis functions in order to reduce the model order. Unlike system models, snapshot methods use “snapshots” of the time variant states to produce basis functions for order reduction. Most literature uses proper orthogonal decomposition (POD) in order to calculate the basis functions. This method will be discussed ad nausea in this thesis. An alternative snapshot based method, called Response Surface Modelling, was also investigated. These methods use statistical methods (such as Bayes linear methodology) to estimate the response of the wells using statistically selected input values [13]. This method was deemed unsuitable for this application because the potential number of changing variables when doing both history matching and production optimisation is very large. This implies the need for many experiments to construct an accurate response surface model to capture the dynamics of the system, defeating the purpose of ROM.

Trajectory piecewise linearization (TPWL) for calculation simplicity in conjunction with POD for model order reduction has been used with success in many different applications and field sizes for reservoir simulation [1] [2] [14]. Previous work showed the ability to capture model nonlinearities and great improvements in calculation efficiency (100-300 times faster). Although these works have investigated both optimisation and history matching, a model combining the two for a purpose such as CLRM has yet to be tested. Additionally, these works have looked at either using gradient-free optimisation algorithms [2] or ensemble methods [1] to perform optimisation. Gradient-based methods with the TPWL-POD procedure will be looked at in this work. The TPWL-POD method was chosen as it was shown to be a relatively nonintrusive method (in terms of the reservoir simulator) and can effectively use multiple snapshots to track the dynamic trajectories of the outputs of the reservoir equation [1].

1.2 Egg Model Description

The Egg model has been used for the work presented in this thesis for a few different reasons. The model has no discernible aquifer or gas cap, therefore the primary recovery from the model is almost negligible and thus is a good candidate for research work on production optimisation [15]. The Egg model was developed as part of the
work with optimization in [16] and [7] and CLRM in [4]. The Egg model is a two-phase (oil and water) model, free of associated or dissolved gas, making it an excellent candidate for use with the POD-TPWL method. Although studies have been performed using the POD-TPWL in a compositional [1] and thermal setting [17], this is not the main objective of this thesis. The model contains 100 different permeability realizations with the same porosity and fluid properties. The well location also remains constant throughout the optimisation as well location was not optimised in this body of work. The dimension of the model in grid blocks is $60 \times 60 \times 7$ for a total of 25 200 grid cells and 18 553 active grid cells as the edges of the model are inactive, giving it an egg-like shape. The size of the grid is shown in Table 1 and the average spacing between the wells is 160 m. The porosity field is homogenous, with a value of 20% throughout. The permeability field of one of the realizations can be seen in Figure 2.

![Figure 2 One permeability realization of the Egg Model showing locations of the producers (red) and injectors (blue) [15]](image)

In Figure 2, the permeability clearly shows a channelized heterogeneous geological environment. The rest of the ensemble helps to capture the uncertainty of the geological setting by changing the location and shape of the channels.

Although the model is small by field standards, it includes many of the properties that a field geological model would contain. One notable missing component is faulting in the reservoir model. The fluids also exhibit important field considerations, such as compressibility and can be seen in Table 1.
Table 1 Reservoir fluid properties of the Egg model [15]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Variable</th>
<th>Value</th>
<th>SI units</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>Grid-block height</td>
<td>4</td>
<td>m</td>
</tr>
<tr>
<td>Δx, Δy</td>
<td>Grid-block length/width</td>
<td>8</td>
<td>m</td>
</tr>
<tr>
<td>φ</td>
<td>Porosity</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>c_o</td>
<td>Oil compressibility</td>
<td>$1.0 \times 10^{-10}$</td>
<td>Pa$^{-1}$</td>
</tr>
<tr>
<td>c_r</td>
<td>Rock compressibility</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>c_w</td>
<td>Water compressibility</td>
<td>$1.0 \times 10^{-10}$</td>
<td>Pa$^{-1}$</td>
</tr>
<tr>
<td>μ_o</td>
<td>Oil dynamic viscosity</td>
<td>$5.0 \times 10^{-3}$</td>
<td>Pa s</td>
</tr>
<tr>
<td>μ_w</td>
<td>Water dynamic viscosity</td>
<td>$1.0 \times 10^{-3}$</td>
<td>Pa s</td>
</tr>
<tr>
<td>k_o</td>
<td>End-point relative permeability, oil</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>k_w</td>
<td>End-point relative permeability, water</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>n_o</td>
<td>Corey exponent, oil</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>n_w</td>
<td>Corey exponent, water</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>S_or</td>
<td>Residual-oil saturation</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>S_sc</td>
<td>Connate-water saturation</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>p_c</td>
<td>Capillary pressure</td>
<td>0.0</td>
<td>Pa</td>
</tr>
<tr>
<td>p_r</td>
<td>Initial reservoir pressure (top layer)</td>
<td>$40 \times 10^6$</td>
<td>Pa</td>
</tr>
<tr>
<td>S_w,0</td>
<td>Initial water saturation</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>q_wi</td>
<td>Water injection rates, per well</td>
<td>79.5</td>
<td>m$^3$/d</td>
</tr>
<tr>
<td>p_b</td>
<td>Production well bottom hole pressures</td>
<td>$39.5 \times 10^6$</td>
<td>Pa</td>
</tr>
<tr>
<td>r_well</td>
<td>Well-bore radius</td>
<td>0.1</td>
<td>M</td>
</tr>
<tr>
<td>T</td>
<td>Simulation time</td>
<td>3600</td>
<td>d</td>
</tr>
</tbody>
</table>

For ROM, the Egg model is an excellent candidate as it is a relatively simple synthetic example. Although the relative permeabilities are not linear, they do follow the Corey relationship making it easier to capture the nonlinearities in the model. The Egg model also includes oil and water compressibility, contributing to the time-dependency of the model. The compressibility values are relatively small, but still have an effect on the time-dependency of the model, important for snapshot based ROM, such as with POD-TPWL. Previous works [1] [17] [2] had used similar densities, viscosities, linear relative permeability relations and no compressibility to test the POD-TPWL method. This work attempts to determine whether or not these model nonlinearities can be captured with the POD-TPWL method.
1.3 Scope of Work

This work will focus on using POD-TPWL models previously presented in [1] [2] and [14] in order to utilize a CLRM framework. The work will also look at using POD-TPWL within a gradient-based optimisation setting. The specific goals of the dissertation are:

- To use the POD-TPWL reduced order modelling approach for CLRM in oil-water problems, specifically applied to the Delft Egg Model [7]. This will include producing one POD-TPWL snapshot for both the reservoir states (pressures and saturations) and geological parameters (transmissibilities).
- To develop a POD-TPWL method for gradient-based optimisation and history matching. This will require modifying the automatic differentiation fully implicit (AD-FI) adjoint module in SINTEF’s Matlab reservoir simulation toolbox (MRST) [18].
- To perform a robust optimisation using the POD-TPWL model developed for changing both model parameters and control inputs. This will be done to hopefully extend the POD-TPWL application to very computationally demanding reservoir management techniques, such as robust optimisation.

1.4 Thesis Outline

- Chapter 2 consists of a brief overview of flow modelling through porous media. The derivation of the TPWL method is introduced. Model order reduction using POD is discussed and the POD-TPWL method is derived. The chapter concludes by testing different methods of POD-TPWL and there is an application of the POD-TPWL method to reservoir flow scenarios.
- Chapter 3 derives the adjoint equation for use with gradient-based optimisation. The adjoint equation is then expressed in reduced form using POD and POD-TPWL. The gradients from the three methods are then compared for water flooding optimisation.
- Chapter 4 applies POD-TPWL to the Egg model, a synthetic reservoir model, in order to perform water flooding optimisation. POD-TPWL is used to solve both the adjoint equation and reservoir equations.
- Chapter 5 introduces robust optimisation for production optimisation to determine the optimal production strategy for an ensemble of geological realizations. The POD-TPWL reservoir equation is derived in terms of both well controls and geological model parameters. The results of the POD-TPWL robust optimisation are compared with results using MoReS, Shell’s in-house reservoir simulation software.
- In Chapter 6, conclusions are made and recommendations for future work are given.
- Appendix A discusses the use of POD-TPWL for gradient-based history matching. A history match is performed on the Egg model and the results are presented.
2 POD-TPWL for Reservoir Simulation

In CLRM, one of the key steps is to perform production optimisation using defined algorithms. The gradient-based method used to optimise the control inputs for the wells in this work will be the adjoint method. The theory and results of the POD-TPWL method for the reservoir simulation will be discussed in this section. The gradient-based adjoint method will then be incorporated into this POD-TPWL framework and robust optimisation will be used in order to reduce the impact of the model uncertainties on the economic life cycle of the field.

2.1 Reservoir Governing Equations

The governing equations used for reservoir simulation are derived from the combination of the mass conservation equations and the multiphase version of Darcy’s law. The mass balance equations for a grid block volume are as follows:

\[ \nabla (\alpha \rho_w \tilde{v}_w) + \alpha \frac{\partial (\rho_w \phi S_w)}{\partial t} - \alpha \rho_w q_w^m = 0, \tag{2.1} \]

\[ \nabla (\alpha \rho_o \tilde{v}_o) + \alpha \frac{\partial (\rho_o \phi S_o)}{\partial t} - \alpha \rho_o q_o^m = 0, \tag{2.2} \]

where the \( o \) and \( w \) subscripts denote either the oil or water phase respectively, \( S \) denotes the saturation of each phase, \( \rho \) is the density of the respective phases, \( \phi \) is the porosity of the rock, \( q \) is the flow rate from the source (wells in our case) and \( \tilde{v} \) is the flow rate as calculated by Darcy’s law. The source term of the equation, \( q \), has, for our purposes, been defined as the Peaceman well equation:

\[ q_o^w = \left[ \frac{2\pi k \Delta d}{\ln \left( \frac{r_o}{r_w} \right)} \right] \lambda_o (p_o - p_o^w), \tag{2.3} \]

\[ q_w^o = \left[ \frac{2\pi k \Delta d}{\ln \left( \frac{r_o}{r_w} \right)} \right] \lambda_w (p_w - p_w^o), \tag{2.4} \]

where \( \lambda \) is the phase mobility, \( k \) is the absolute permeability of the rock properties associated with the grid block, \( r_w \) is the radius of the wellbore and \( r_o \approx 0.2 \Delta x \), where \( \Delta x \) is defined as the width of the grid block. The Darcy flow from equation from equation (2.1) and (2.2) is defined as:

\[ \tilde{v}_w = \frac{k_w}{\mu_w} \mathbf{K} (\nabla p_w - \rho_w g \nabla d), \tag{2.5} \]

\[ \tilde{v}_o = \frac{k_o}{\mu_o} \mathbf{K} (\nabla p_o - \rho_o g \nabla d), \tag{2.6} \]
where $k_r$ is the relative permeability, $\mu$ is the viscosity, $\mathbf{K}$ is the permeability tensor, $p$ is the phase pressure, $g$ is the gravitational constant and $d$ is the depth of the grid block. Combining both the mass balance and Darcy equations gives:

\[
-\nabla \left( \alpha \rho_w \frac{k_{rw}}{\mu_w} \mathbf{K} (\nabla p_w - \rho_w g \nabla d) \right) + \alpha \frac{\partial (\rho \phi S_w)}{\partial t} - \alpha \rho_w q_w^m = 0, \tag{2.7}
\]

\[
-\nabla \left( \alpha \rho_o \frac{k_{ro}}{\mu_o} \mathbf{K} (\nabla p_o - \rho_o g \nabla d) \right) + \alpha \frac{\partial (\rho \phi S_o)}{\partial t} - \alpha \rho_o q_o^m = 0. \tag{2.8}
\]

As is evident from the above equations, there are four unknowns ($p_o$, $p_w$, $S_w$ and $S_o$) and two equations to solve. For this case, an oil-water system with no capillary pressure was considered. This gives two additional equations to help solve the problem:

\[
S_o + S_w = 1,
\]

\[
p_o - p_w = p_c(S_w) = 0 \rightarrow p_w = p_o. \tag{2.9}
\]

Compressibility of the oil and water can be represented as:

\[
c_o \triangleq \frac{1}{\rho_o} \frac{\partial \rho_o}{\partial p_o} \bigg|_{T}, \tag{2.10}
\]

\[
c_w \triangleq \frac{1}{\rho_w} \frac{\partial \rho_w}{\partial p_w} \bigg|_{T} = \frac{1}{\rho_w} \frac{\partial \rho_w}{\partial p_o} \bigg|_{T}, \tag{2.11}
\]

where $c$ is the compressibility and $T$ is the reservoir temperature. Note that $p_w$ is replaced by $p_o$ due to the absence of capillary pressure. Equations (2.5) and (2.6) can be combined with equations (2.7), (2.8) and (2.9) to give the governing equation:

\[
-\nabla \left( \alpha \rho_w \frac{k_{rw}}{\mu_w} \mathbf{K} (\nabla p_w - \rho_w g \nabla d) \right) + \rho_o \phi \left[ S_w (c_w) \frac{\partial P}{\partial t} + \frac{\partial S_w}{\partial t} \right] - \rho_w q_w^m = 0, \tag{2.12}
\]

\[
-\nabla \left( \alpha \rho_o \frac{k_{ro}}{\mu_o} \mathbf{K} (\nabla p_o - \rho_o g \nabla d) \right) + \rho_o \phi \left[ S_o (c_o) \frac{\partial P}{\partial t} + \frac{\partial S_o}{\partial t} \right] - \rho_o q_o^m = 0. \tag{2.13}
\]

Equations (2.12) and (2.13), which are defined as the governing equations for the discretized system, produces a nonlinear set of partial differential equations (PDEs). These equations can be re-written as:

\[
g_k(x_{k-1}, x_k, u_k) = 0 \tag{2.14}
\]

Where $k$ is the discretized time step, $g$ is the residual vector, $x$ is the state vector (oil phase pressure and water saturation in every grid block) and $u$ is the well control.
parameters. The well control parameters can be either well pressures or well flow rates that are specified by the operator.

Equation (2.14) is nonlinear and, using the automatic differentiation fully implicit (AD-FI) module in MRST, is solved using Newton’s method. This solving method requires the construction of the Jacobian matrix, $\mathbf{E}$:

$$\mathbf{E} = \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k}$$  \hspace{1cm} (2.15)

Solving the above equation requires using a perturbation of $\mathbf{x}$, for this instance we will use $\delta \mathbf{x}$ as the notation for the perturbation. The equation then becomes:

$$\mathbf{E} \delta \mathbf{x}_k = -\mathbf{g}_k,$$  \hspace{1cm} (2.16)

where the residual, $\mathbf{g}$, is reduced to zero or the user-defined maximum allowable tolerance. The updated state vector, $\mathbf{x}^{i+1}_k$, is calculated using:

$$\mathbf{x}^{i+1}_k = \mathbf{x}^i_k - \frac{\mathbf{g}_k}{\mathbf{E}},$$  \hspace{1cm} (2.17)

where the superscript $i$ refers to the iteration number. If further information is required for reservoir simulation equations refer to [19]. The TPWL approach attempts to approximate the solution of equations (2.12) and (2.13) by eliminating the need to perform nonlinear iterations to update the state vector.

### 2.2 TPWL

Similar to [1] and [2], equation (2.14) is linearized around a training point, $i$, by performing a first order Taylor series expansion around $i$. The governing equation for TPWL then becomes:

$$\mathbf{g}^n_k(x^n_k, x^n_{k-1}, u^n_k) - \mathbf{g}^l_k(x^l_k, x^l_{k-1}, u^l_k) = \begin{bmatrix} \left( \frac{\partial \mathbf{g}_k^l}{\partial x^n_{k-1}} \right) (x^n_{k-1} - x^l_{k-1}) + \ldots \\ \left( \frac{\partial \mathbf{g}_k^l}{\partial x^l_k} \right) (x^n_k - x^l_k) + \ldots \\ \left( \frac{\partial \mathbf{g}_k^l}{\partial u^l_k} \right) (u^n_k - u^l_k) \end{bmatrix},$$  \hspace{1cm} (2.18)

where superscript $l$ indicates the training values from the high-fidelity (HF) snapshots and superscript $n$ indicates the new calculated values from TPWL. The snapshots from the HF simulation are created at each report step of the simulation. The approximation $\mathbf{g}^n_k(x^n_k, x^n_{k-1}, u^n_k) - \mathbf{g}^l_k(x^l_k, x^l_{k-1}, u^l_k) = 0$ will be used, eliminating the need to calculate the residual of the new simulation as it would require nonlinear iterations to determine the
$g_k^*$ value. The following notation will be used to represent the partial derivatives written above:

$$
\bar{E}_k = \left( \frac{\partial g_k}{\partial x_k} \right),
$$

$$
\bar{A}_k = \left( \frac{\partial g_k}{\partial x_{k-1}} \right),
$$

$$
\bar{B}_k = \left( \frac{\partial g_k}{\partial u_k} \right),
$$

Equation (2.18) can then be re-arranged to solve for $x_k^*$ and written as:

$$
x_k^* = x_k^* + \left( \bar{E}_k \right)^{-1} \left[ \bar{A}_k (x_{k-1} - x_{k-1}^*) + \bar{B}_k (u_k^* - u_k^*) \right].
$$

### 2.2.1 TPWL Results

The partial derivatives listed in equation (2.19) were obtained using MRST's AD-FI module and were saved in order to calculate the updated state vector using the TPWL method. The cumulative oil and water production error was calculated using:

$$
E_o = \sum_{k=1}^{K} \left( \frac{q_{o,k}^{bf} - q_{o,k}^{mul}}{q_{o,k}^{bf}} \right),
$$

$$
E_{w, inj} = \sum_{k=1}^{K} \left( \frac{q_{w,k}^{bf} - q_{w,k}^{mul}}{q_{w,k}^{bf}} \right),
$$

$$
E_{w, prod} = \sum_{k=1}^{K} \left( \frac{q_{w,k}^{bf} - q_{w,k}^{prod}}{q_{w,k}^{bf}} \right),
$$

Chapter 2 POD-TPWL for Reservoir Simulation 10
where $E_o$ and $E_w$ are the error in oil and water production respectively, $q$ is the flow rate and superscript $hf$ and $tpwl$ correspond to the high-fidelity and POD-TPWL simulations. The results of the TPWL simulation can be seen in Figure 3 to Figure 6 and Table 2.

The training run and experimental BHP schedule in Figure 3 were randomly selected values between 400 and 405 bars. This range was selected to prevent very high or negative injection flow rates. A random selection of these values was done to reduce bias and thus better test the ability of the TPWL model to predict the reservoir response.

Figure 3 Training and experimental schedule for TPWL simulation

Figure 4 Well water injection rates compared for TPWL and HF simulation
In Figure 4, there is a mismatch between the HF and POD-TPWL simulation. Note that the training simulation rates are displayed in the figure. BHP was controlled in the injectors, but this did not translate to being able to precisely determine the injection rates of the wells. Equation (2.4) shows that the water injection rate is dependent on the well BHP, reservoir grid block pressure and the water saturation (from the water mobility ratio). The latter two of the aforementioned properties were updated during TPWL and small deviations from the HF simulation can lead to a large error in the water injection rate. Similar issues were observed in the production wells.

![Figure 5 Oil production rates compared for TPWL and HF simulation](image)

![Figure 6 Water production rates compared for TPWL and HF simulation](image)

Figure 5 and Figure 6 show the production rates of the HF simulation, TPWL and the training simulation. Many of the production rates of the training simulation are similar to the experiment with the updated BHP schedule. TPWL follows the production rates of
both the HF and training simulation. There is a clear departure from the training simulation (for example the water rate in producer 3 at 2250 days) and a closer match to the HF simulation for TPWL. This would indicate that some points of the TPWL simulation are better equipped to capture the model nonlinearities. If the training simulations could capture all of the nonlinearities of the model, a very precise match could be obtained. The cumulative errors of oil and water for the TPWL experiment are presented in Table 2.

<table>
<thead>
<tr>
<th>Error Comparison</th>
<th>HF Simulation</th>
<th>Training Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_o$ [%]</td>
<td>0.90</td>
<td>2.42</td>
</tr>
<tr>
<td>$E_{w, inj}$ [%]</td>
<td>4.05</td>
<td>7.99</td>
</tr>
<tr>
<td>$E_{w, prod}$ [%]</td>
<td>3.84</td>
<td>9.19</td>
</tr>
</tbody>
</table>

The error between the HF simulation and TPWL is much smaller than the error between the training run and TPWL. TPWL thus does not follow the training run completely and is able to more closely match the HF simulation with the updated BHP controls.

Modest calculation gains of 10-12 times for the Egg model (100 s for TPWL, 1100 s for MRST) were obtained using the TPWL method. [2], [1] and [2] have shown that gains of 100-500 times were possible by using further model order reduction in the form of POD. With the need for multiple training simulations using the high-fidelity model and the application to CLRM and Robust Optimisation, a further speed up using POD is necessary.

### 2.3 POD

Similar to [1] and [2], POD was used in order to reduce the model order. The full order state vector can be represented by:

$$x = \Phi z,$$

(2.22)

where $x$ is the full order state vector, $\Phi$ represents the constructed basis matrix and $z$ is the reduced order state vector. The concept of using POD and applying it to TPWL has been well developed in [1] [2] and [20]. Substituting equation (2.22) into equation (2.20) gives the combined POD-TPWL equation:

$$z^*_k = \tilde{z}^*_k + (\tilde{E}^T \Phi)^{-1} \left[ \tilde{A}^*_k \Phi (z^*_k - \tilde{z}^*_k) + \tilde{B}^*_k (u^*_k - u^*_k) \right].$$

(2.23)

The calculated states can then be projected back into the full order space using equation (2.22). The projection method used in equation (2.23) is known as the Galerkin method. There are different projection methods that will be discussed later in this chapter. In practice, POD was achieved by performing singular value decomposition (SVD) on the snapshot matrix to produce the orthonormal basis matrix $\Phi$. POD is a good fit for this purpose due to the fact that it minimizes the mean-square reconstruction error for the snapshots [1].
The approach involves separating the state vector into both pressure and saturation values as the scaling of the values between pressure (generally in the magnitude range of $10^7$ - $10^8$ for reservoir problems) and water saturation (between 0 and 1) are dissimilar. POD depends on the amount a particular grid block values contribute to the change in the model. Each snapshot had a SVD performed across all of the time steps in order to focus on which values changed the most throughout the snapshot. The separated matrices for pressure and saturation thus became:

$$X_p = [p_1 - \bar{p} \quad p_2 - \bar{p} \quad \ldots \quad p_K - \bar{p}], \quad X_s = [s_1 - \bar{s} \quad s_2 - \bar{s} \quad \ldots \quad s_K - \bar{s}], \quad (2.24)$$

where \( s \) is the water saturation, \( \bar{s} \) and \( \bar{p} \) refer to the mean values and the time is the subscript for \( p \) and \( s \) (\( t = 1, 2, \ldots, K \)). The SVD of the pressure and saturation values was performed in Matlab using the SVD function, where the representation of \( X_p \) and \( X_s \) in equation (2.24) is:

$$X = U \Sigma V^T, \quad (2.25)$$

where \( U \) is the basis matrix \( \Phi \) and \( \Sigma V^T \) is the reduced state vector \( z \). The basis matrix retains some number of the columns in \( X_p \) and \( X_s \), with the maximum number being \( K \). Creating separate basis functions for the saturation and pressure and combining the two basis matrices gives the matrix solution of:

$$x = \begin{bmatrix} X_p \\ X_s \end{bmatrix} \approx \Phi z = \begin{bmatrix} \Phi_p \\ \Phi_s \end{bmatrix} \begin{bmatrix} z_p \\ z_s \end{bmatrix}, \quad (2.26)$$

### 2.3.1 POD-TPWL Results

It is generally unnecessary to retain the entire number of columns of \( X_p \) and \( X_s \), but instead using a fraction of these columns based on the energy wished to be retained during the SVD. The pressure and saturation require a different number of basis functions to be retained for the same amount of energy during the SVD, thus two basis functions are combined as per equation (2.26). The results from testing different values of energy retention using the Galerkin projection within TPWL can be seen in Figure 7.

The simulation was performed using a single training run (as shown in Figure 7) by changing BHP values. The producers were kept at a constant BHP of 395 bars. Similar to section 2.2.1, a random training and experimental BHP schedule (between 400 and 405 bars, updated every 90 days) was chosen to decrease the bias towards using a similar schedule to the training run. The results are displayed in Table 3 and Figure 7 to Figure 8.
Table 3 shows that for the same projection energies, more of the saturation values were included than pressure values. This is as you would expect as the saturation of the system exhibits larger (if both values are scaled) and more frequent changes (due to transport effects) than for the pressure of the field, which tends to be a more gradual change. POD-TPWL somewhat accurately predicted the behaviour of the reservoir. The difference in speed to calculate the projection matrix of the various number of basis functions was significant (2-6 times difference). The 99.99% energy retention value will be used for the rest of the calculations in order to obtain accuracy (maximum of 1.5% error difference with 99.9999%) and to maintain the speed of calculations.
The results in Figure 8 show a poor correlation between the number of basis functions and accuracy, possibly due to the accuracy of saturation and pressure around the wellbore. The well equation for the flow rate is dependent on the reservoir connection grid block pressure and the saturation in that grid block. Small errors in those values can lead to large discrepancies in the results of Figure 8. Other methods have to be investigated in order to obtain better accuracy.

### 2.3.2 Local Resolution Scheme

Local resolution (LR) scheme was developed in [20] with the purpose to increase the accuracy of the POD-TPWL method around grid cells or other important cells within the reservoir. A separate basis function, $\Phi_{LR}$, was constructed for the well grid blocks. By retaining all of the information at the well grid block locations, the global reconstruction error is minimized and preferentially weighting the information at the well locations. Equation (2.26) then becomes:

$$
\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{x}_{LR} \\ \mathbf{x}_G \end{bmatrix} \approx \mathbf{P} \begin{bmatrix} \Phi_{LR} & 0 \\ 0 & \Phi_G \end{bmatrix} \begin{bmatrix} z_{LR} \\ z_G \end{bmatrix},
$$

(2.27)

where $\mathbf{P}$ is a permutation matrix constructed to isolate the well grid blocks, $\mathbf{x}_{LR}$ is the state vector for the well grid blocks, $\mathbf{x}_G$ is the state vector for the rest of the grid blocks, $\Phi_{LR}$ and $\Phi_G$ are the respective basis functions and $z_{LR}$ and $z_G$ are the respective reduced state vectors. $\Phi_G$ is constructed using SVD of $\mathbf{x}_G$ and $\Phi_{LR}$ is taken to be the identity matrix. The modified basis matrix, $\Phi$ is defined as:
\[
\tilde{\Phi} = P \begin{bmatrix}
\Phi_{LR} & 0 \\
0 & \Phi_G
\end{bmatrix}.
\] (2.28)

The results of using LR basis functions can be seen in Figure 9 and Table 4.

![Figure 9 Water saturation profiles for layer 1 of realization 1 of the Egg Model using POD and LR-POD as compared to the high-fidelity model](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>LR-POD</th>
<th>POD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E_o) [%]</td>
<td>1.10</td>
<td>8.44</td>
</tr>
<tr>
<td>(E_{w,\text{inj}}) [%]</td>
<td>5.15</td>
<td>5.83</td>
</tr>
<tr>
<td>(E_{w,\text{prod}}) [%]</td>
<td>4.21</td>
<td>5.80</td>
</tr>
</tbody>
</table>

By using LR, many of the saturation artefacts were eliminated throughout the layer and the well location produced much better results. In the Egg model, the 12 wells are all connected to each of the 7 layers, therefore 84 values were added to \(\tilde{\Phi}\). The speed of the projection was 21 seconds slower due to these added values, but the accuracy gain was significant, especially for the oil production. Table 4 shows a large decrease in the oil production error and modest reductions in the water production and injection error. At 900 days in Figure 9, the difference between the saturation in LR-POD and POD is much greater than at 2500 days. Water breakthrough occurs at about 1000 days in the producers and before this, the oil production is at the peak. The error reduction using LR-POD is therefore much greater for oil production than water production. This method will be used throughout the rest of the thesis when referring to POD.
2.3.3 Projection Methods for POD-TPWL

Another option for POD introduced in [20] was the use of different projection methods for the reduction of the TPWL model. Shown in equation (2.23) is the Galerkin projection. Two other methods were discussed in [1] as well. These were the Petrov-Galerkin method and inverse projection method. The reduced Jacobians for the Petrov-Galerkin and inverse projections can be seen in equations (2.29) and (2.30) respectively:

\[
\begin{align*}
\bar{E}_r^i &= \Phi^T (\bar{E}_i^i)^T \bar{E}_i^i \Phi, \\
\bar{A}_r^i &= \Phi^T (\bar{A}_i^i)^T \bar{A}_i^i \Phi, \\
\bar{B}_r^i &= \Phi^T (\bar{B}_i^i)^T \bar{B}_i^i, \\
\bar{E}_r^i &= \Phi^T (\bar{E}_i^i)^{-1} \bar{E}_i^i \Phi, \\
\bar{A}_r^i &= \Phi^T (\bar{A}_i^i)^{-1} \bar{A}_i^i \Phi, \\
\bar{B}_r^i &= \Phi^T (\bar{B}_i^i)^{-1} \bar{B}_i^i, \\
\end{align*}
\] (2.29)

To simplify the notation, equation (2.23) will be re-written with reduced Jacobian notation:

\[
z_k^e = z_k^i + (\bar{E}_r^i)^{-1} \left[ \bar{A}_r^i (z_{k-1}^i - z_k^{i-1}) + \bar{B}_r^i (u_k^i - u_k^{i-1}) \right].
\] (2.31)

Note that when the inverse projection is chosen, a large sparse inverse \((\bar{E}_i^i)^{-1}\) must be calculated for each time step instead of the transpose of \(\bar{E}\). The inverse of the matrix requires much more time to be calculated, but the inverse projection is a more accurate mathematical representation. The results of the different projection methods are displayed in Table 5. The results differ from those in section 2.3.1 as local resolution was used in combination with the projection methods and the training and experimental schedules were changed.

<table>
<thead>
<tr>
<th>Projection Type</th>
<th>Time for Projection [s]</th>
<th>(E_e %)</th>
<th>(E_w %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galerkin</td>
<td>71.98</td>
<td>9.278</td>
<td>8.01</td>
</tr>
<tr>
<td>Petrov Galerkin</td>
<td>142.38</td>
<td>0.107</td>
<td>10.96</td>
</tr>
<tr>
<td>Inverse</td>
<td>1860.02</td>
<td>0.040</td>
<td>2.20</td>
</tr>
</tbody>
</table>

The results of the projection comparison show that, although there is a larger processing time, the inverse projection produces more accurate results than the other two projection methods. The choice of projection should be chosen based on the reduction in the training simulations needed. Although the inverse projection requires much more time to calculate, if POD-TPWL simulation can be performed in lieu of many time consuming high fidelity simulations, then it would be worth investing the time for

Chapter 2 POD-TPWL for Reservoir Simulation 18
the projection. As the remainder of this work will require many accurate simulations, the inverse projection will be used for the subsequent simulations.

### 2.4 Fluid Property Effect on POD-TPWL Accuracy

The error observed using POD-TPWL can be attributed to the nonlinearities of the system. The source of the nonlinearities in the Egg model includes the nonlinear nature of the relative permeabilities. Simplifying these values in the model by using linear relative permeabilities, the results can be compared to determine the effect that these properties have on the POD-TPWL accuracy. The relative permeabilities of the simple fluid model and the Egg model are displayed in Figure 10.

![Figure 10 Relative permeabilities for the Egg model and simple fluid model](image)

The Egg model relative permeabilities follow the quadratic Corey relation, the simple case uses a linear relation. The error comparison between the simple and Egg model fluid case can be seen in Table 6 and Figure 11 to Figure 13.
Figure 11 Training run and simulation BHP schedule for both fluid property cases

Figure 11 was trained using a constant schedule and then for the experimental cases with POD-TPWL, a control schedule was used that had random changes in the BHP every 90 days. This was done to have a fairly constant training schedule and a fluctuating experiment to see how well the fluctuations were captured by POD-TPWL.

Table 6 Cumulative production errors for both the standard Egg model and simple fluid properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Regular Egg model</th>
<th>Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_p$ [%]</td>
<td>0.42</td>
<td>0.0010</td>
</tr>
<tr>
<td>$E_{w, inj}$ [%]</td>
<td>2.27</td>
<td>0.0010</td>
</tr>
<tr>
<td>$E_{w, prod}$ [%]</td>
<td>7.37</td>
<td>0.010</td>
</tr>
</tbody>
</table>
Figure 12 Oil production rate comparison for the regular Egg model fluid property case

Figure 13 Oil production rate comparison for the simple fluid property case

Figure 12, Figure 13 and Table 6 show that the error in the water and oil production are greatly reduced when using POD-TPWL and a simple fluid model. This can be attributed to the linear relative permeability values. In Figure 13, the fluctuations in the oil production are captured with ease using POD-TPWL. In comparison, Figure 12 exhibits that the Egg model fluid properties exhibit a large mismatch when the oil production rate exhibits drastic changes. The states of the well grid blocks have a large impact on the well production values. A small error in water saturation (due to the nonlinearities in the relative permeability) would continually have a large effect on the production values and the error is extrapolated throughout the rest of the reservoir simulation.
The system was also made to be incompressible. By making the system incompressible, the problem became elliptic and time independent. Equations (2.12) and (2.13) become:

\[-\nabla (\rho_w \frac{k_w}{\mu_w} \bar{K} (\nabla p_o - \rho_o g \nabla d)) - \rho_w q_w = 0, \tag{2.32}\]

\[-\nabla (\rho_o \frac{k_o}{\mu_o} \bar{K} (\nabla p_o - \rho_w g \nabla d)) - \rho_o q_o = 0, \tag{2.33}\]

The need for iterative solving is thus eliminated and the lack of time dependency makes the construction of basis functions for the time independent problem more easily constructible as the variation becomes spatial, instead of spatial and temporal.
3 Reduced-Order Modelling for Adjoint-Based Optimisation

This thesis investigates the use of gradient-based methods to optimise reservoir engineering problems. The gradient-based method that was used during this work was the adjoint method. By using this method, an objective function $J$ is either maximised (NPV) or minimized (history matching). The objective function can be expressed generally as:

$$J(u_{1,K}, y_{1,K}) = \sum_{k=1}^{K} J_k(u_k, y_k),$$  \hspace{1cm} (3.1)

where $y$ is a vector of measured outputs and $u$ is a vector of control inputs. Using the derivation outlined in [21], a modified objective function can be defined to take into account the temporal dependency of $u$ and $y$. The modified objective function, in terms of Lagrange multipliers is:

$$\tilde{J}(u_{1,K}, x_{0,K}, y_{1,K}, \lambda_{0:K}) \triangleq \sum_{k=1}^{K} \begin{bmatrix} J_k(u_k, y_k) \\ + \lambda_0^T (x - \bar{x}_0) \delta_{k-1} \\ + \lambda_k^T g_k(u_k, x_{k-1}, x_k) \end{bmatrix},$$  \hspace{1cm} (3.2)

where the initial condition constraint $x - \bar{x}_0 = 0$, and the system constraint, $g_{1:K} = 0$, have been adjoined to the objective function using the Lagrange multipliers $\lambda_0$ and $\lambda_{1:K}$ respectively. The Kroenecker delta, $\delta_{k-1}$, in equation (3.2) ensures that the initial constraint is included in the summation ($\delta_0 = 1$, else $\delta_k = 0$). In order to achieve the optimal solution for the modified objective function, all first order derivatives must equal zero. These first order conditions can be seen in [21] and are referred to as the Euler-Lagrange equations. Following the derivation in [21], the adjoint equation can be expressed as:

$$\lambda_k = - \left[ \left( \frac{\partial g_{k+1}}{\partial x_k} \right)^T \right]^{-1} \left[ \left( \frac{\partial g_{k+1}}{\partial x_k} \right)^T \lambda_{k+1} + \left( \frac{\partial J_k}{\partial x_k} \right)^T \right],$$  \hspace{1cm} (3.3)

$\lambda_k$ is calculated backwards in time using $k = K-1, K-2, ..., 1$. The final condition at $K$ is calculated using:

$$\lambda_K = - \left[ \left( \frac{\partial g_K}{\partial x_K} \right)^T \right]^{-1} \left( \frac{\partial J_K}{\partial x_K} \right)^T,$$  \hspace{1cm} (3.4)

the Lagrange multiplier can then be used to calculate the modified objective function:

$$\frac{\partial \tilde{J}}{\partial u_k} = \frac{\partial J_k}{\partial u_k} + \lambda_k^T \left( \frac{\partial g_k}{\partial u_k} \right).$$  \hspace{1cm} (3.5)
The gradient in equation (3.5) can then be used to update the control vector. The control vector is updated using the steepest ascent method \( u_{k,\text{new}} = u_{k,\text{old}} + \frac{\partial J_k}{\partial u_k} \) (NPV) or steepest descent method \( u_{k,\text{new}} = u_{k,\text{old}} - \frac{\partial J_k}{\partial u_k} \) (history matching). The usual practice for calculating the derivatives for the adjoint equation is done using a backwards run. In [22] it was noted that all of the gradients needed to calculate the adjoint solution are calculated during the forward run of the simulator. In MRST, due to the nature of the AD calculations, it was deemed beneficial to save the derivatives calculated during the forward run to calculate the adjoint calculation. This was done using the algorithm outlined in [22] and the code for the method can be found in [23].

3.1.1 POD Adjoint

The POD concepts outlined earlier in the chapter were applied to the adjoint calculation. This work had already been developed in [3] and showed moderate speed-ups and accuracy. After POD was applied, the adjoint equations in (3.3) and (3.5) become:

\[
\lambda_k = - \left( \Phi^T \frac{\partial g_k}{\partial \mathbf{x}_k} \Phi \right)^{-1} \left( \Phi^T \frac{\partial g_{k+1}}{\partial \mathbf{x}_k} \Phi \right) \lambda_{k+1} + \left( \frac{\partial J_k}{\partial \mathbf{x}_k} \Phi \right)^T, \tag{3.6}
\]

\[
\frac{\partial J}{\partial u_k} + \lambda_k \Phi^T \frac{\partial g_k}{\partial u_k}, \tag{3.7}
\]

This produces a computational advantage when calculating the inverse in equation (3.6). The POD method has the disadvantage, in terms of reduced order modelling, of having to calculate the derivatives of the forward model for each optimisation iteration. Meaning that for every iteration of the optimisation, the derivatives listed in equations in (3.6) and (3.7) are calculated and the inverse of:

\[
\left( \Phi^T \frac{\partial g_k}{\partial \mathbf{x}_k} \Phi \right)^T, \tag{3.8}
\]

must be calculated, which is very time consuming even for the reduced projection from POD.

3.1.2 POD-TPWL Adjoint

Similar to the forward run, a first order Taylor series expansion can be applied to the adjoint equation by rearranging the equation. As the adjoint equation is already in a linear form, the expected error using this method will be coupled with the results from the forward equation (i.e. states and outputs). By taking equation (3.2) and modifying it further in order to get an appropriate form for a Taylor series expansion, the equation becomes:
where \( f \) can be defined as the modified objective function. A first order Taylor series expansion of \( f \) is then done around point \( i \). The general formula for this expansion is:

\[
\sum_{k=1}^{K} \left[ J_k(u_k, y_k) + \lambda^x_0 (x - x_0) \delta_k + \lambda^y_0 g_k (u_k, x_{k-1}, x_k) \right] \approx 0, \quad (3.9)
\]

if we consider \( f^n \) and \( f^i \) as equal to zero and rearrange to solve for \( \lambda^n \) we obtain:

\[
\lambda^n = \lambda^l + \left( \frac{\partial f^i}{\partial \lambda^l} \right)^{-1} \left[ \frac{\partial f^i}{\partial u_k} (u^n_k - u^i_k) \right] + \frac{\partial f^i}{\partial x_0} (x^n_0 - x^i_0) + \frac{\partial f^i}{\partial x_k} (x^n_k - x^i_k) + \frac{\partial f^i}{\partial y_k} (y^n_k - y^i_k) + \frac{\partial f^i}{\partial \lambda_0} (\lambda^n_0 - \lambda^i_0), \quad (3.11)
\]

using this approximation, lambda can be updated and the modified objective function can be calculated using equation (3.7). Similar to the forward run, the first order derivatives needed are stored during the high-fidelity model forward run. Using POD-TPWL, equations (3.3) and (3.5) then become:

\[
\lambda^n_k = \lambda^l + \Phi^T \left[ \left( \frac{\partial g_k}{\partial x_k} \right) \right]^{-1} \left[ \left( \frac{\partial g_{k+1}}{\partial x_k} \right) \Phi^T \lambda_{k+1} + \left( \frac{\partial f^i}{\partial \lambda^l} \right) \Phi^T (x^n_k - x^i_k) \right], \quad (3.12)
\]

\[
\frac{\partial \tilde{J}}{\partial u_k} = \frac{\partial f^i}{\partial u_k} (u^n_k - u^i_k) + (\lambda^l)^T \left( \frac{\partial g_k}{\partial u_k} \right) (u^n_k - u^i_k), \quad (3.13)
\]
The accuracy of the POD and POD-TPWL adjoint gradient was checked against the high-fidelity adjoint gradient from MRST. These results are shown in Figure 14 and Table 7. Note that for the optimisation in Table 7 and Figure 14, the producer BHPs were kept constant at 395 bars.

![Figure 14 Adjoint gradient comparison between the high-fidelity, POD and POD-TPWL equations. Note 40 control steps were used, but for plotting purposes a step plot was used where control step 1 spans 1-2, step 2 spans 3-4, etc.](image)

<table>
<thead>
<tr>
<th>Adjoint Calculation</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-fidelity</td>
<td>207.99</td>
</tr>
<tr>
<td>POD</td>
<td>109.15</td>
</tr>
<tr>
<td>POD-TPWL</td>
<td>9.06</td>
</tr>
</tbody>
</table>

The POD and POD-TPWL methods showed a deviation from the gradient calculated by the high-fidelity calculation in MRST. The inability of the POD-TPWL method to capture the nonlinearities produced in the outputs used for the objective function is a probable cause of this deviation. Injectors 1, 2, 3, 6, 7 and the first half of 4 exhibited good matches with the HF gradient. The accuracy of the POD-TPWL adjoint gradient is shown to be relatively independent of the distance of the controls from the training run. The error can be attributed to accounting for some of the states during POD instead of every state. If some important states were overlooked during the SVD, this would create a mismatch in the calculated gradient.

The POD method showed a further mismatch as the nonlinearities of the output were ‘smoothed’ out by the basis function projection. The gains in speed are quite significant (≈2x for POD and ≈23x for POD-TPWL) and can be attributed to model order reduction (POD) and the elimination of the need to calculate the inverse in equation (3.12). These speed-ups can have a significant impact on time for the optimisation of a model requiring large numbers of iterations.
The retrain criteria mentioned previously will be applied to the forward POD-TPWL model only. It is assumed that if the forward run requires retraining, the backward run will require retraining at the same point. This is due to implementation, as a forward simulation would be needed in order to retrain the adjoint equation.
4 Production Optimisation

4.1 Objective Function

For this body of work, water flooding optimisation was performed on the Egg model. BHP of the injection wells was controlled and the producer BHP was kept constant at 395 bars. This was performed on well BHPs as the AD framework in MRST did not produce Jacobians in terms of well rate controls. Rate control should be investigated further in future work performed with POD-TPWL optimisation. The objective function, $J$, used is calculated using:

$$J(u_{i,k}, y_{i,k}(u_{i,k})) = \sum_{k=1}^{K} \left[ \sum_{i=1}^{N_{inj}} r_{wi,i} \cdot (y_{wi,i})_k + \sum_{j=1}^{N_{prod}} \left( r_{wp,j} \cdot |y_{wp,j,k}| + r_o \cdot |y_{o,j,k}| \right) \right] \times \Delta t_k$$

(4.1)

where $y_{wi,i}$ is an input variable representing the water injection rate of well $i$ (positive in our sign convention and different from the control input vector $u$), $y_{wp,j}$ is an output representing the water production rate of well $j$ (negative), $y_{o,j}$ is also an output, representing the oil production rate of well $j$ (also negative), $r_{wi}$ and $r_{wp}$ are the (constant) water injection and production costs (negative, with units $$/m^3$), $r_o$ is the (constant) oil price (positive, $$/m^3$), $\Delta t_k$ is the time interval of time step $k$ in days, $b$ is the discount rate for a reference time interval $\tau$ (which is usually taken as a year), and $N_{inj}$ and $N_{prod}$ are the number of injection wells and production wells respectively [21].

4.2 Implementation

The workflow for the optimisation procedure using POD-TPWL is detailed in Figure 15. The proxy for POD-TPWL is built during the HF forward simulation and the HF adjoint simulation, afterwards the reservoir simulator (MRST AD-FI for this case) is no longer needed. The POD-TPWL forward and backward simulation both speedup the workflow for optimisation problems.
During the optimisation loop, the bounds for the error of the POD-TPWL must be specified prior to running the loop. The retraining criteria of the model must be defined.

4.2.1 Retraining Criteria

The distance of the states and controls from the surrogate model should be considered to determine whether or not the simulation of the POD-TPWL remains valid. The error calculation used for this work has been derived from the work in [14]. The error in the states \( E_x \) and the controls \( E_u \) is as follows:

\[
E_x = \frac{1}{T \times \sqrt{N_x}} \sum_{k=1}^{K} \left( \|x_k^N - x_k^e\| \times \Delta t_k \right), \tag{4.2}
\]

\[
E_u = \frac{1}{T \times \sqrt{N_u}} \sum_{k=1}^{K} \left( \|u_k^N - u_k^e\| \times \Delta t_k \right), \tag{4.3}
\]

where \( T \) is the total time of the simulation, \( N_x = 37106 \) for this case and \( N_u = 12 \) (one for each well) are the total number of states and controls and \( \Delta t_k \) is the change in time for the particular time step \( k \). Simulations were run using different input schedules, but the same training run, to determine the optimal distance from the training states and controls to perform a retrain of the training image model, results can be seen in Table 8.
Figure 16 shows the schedules used for the retraining criteria determination. A gradient for the BHP controls was calculated during the first time step where a constant schedule was originally used. The one training run was then done using the first update of the original schedule where the maximum change of the BHP for each control step was 0.5 bar. Runs 1 through 4 were then updated with the same gradient and the maximum allowable change in the schedule was increased by 0.5 bar. This was done as the changes to the states would be similar in nature, but differ by the magnitude.

Table 8 shows that as the error in the controls increase, so does the error in the states from the training simulation. The error in the oil production, when compared to the HF simulation with the same control schedule as the run number, follows a similar trend in that it increases with the increase in control distance from the training controls. The error in the POD-TPWL water injection and production values from the HF simulation increases between run 1 and 2, but then begins decreasing thereafter. This is because the cumulative water values begin drastically increasing after run number 2, thus when the error is calculated, a larger mismatch in rate would correspond to a similar or lower value in the error with a larger value in the denominator for the calculation.

From previous works [1] [2] [14], an acceptable error limit occurs between run number 3 and 4. For our purposes with this model, $E_{u, \text{max}} = 0.006$ and $E_{x, \text{max}} = 6.5$ will be used.

Table 8 Error between POD-TPWL and HF model when run with schedules different distances from the training image

<table>
<thead>
<tr>
<th>Run #</th>
<th>$E_u$</th>
<th>$E_x$</th>
<th>$E_o$ [%]</th>
<th>$E_{w,\text{inj}}$ [%]</th>
<th>$E_{w,\text{prod}}$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0021</td>
<td>3.739</td>
<td>1.469</td>
<td>7.171</td>
<td>16.46</td>
</tr>
<tr>
<td>2</td>
<td>0.0040</td>
<td>5.745</td>
<td>2.305</td>
<td>11.99</td>
<td>27.47</td>
</tr>
<tr>
<td>3</td>
<td>0.0056</td>
<td>6.506</td>
<td>2.915</td>
<td>11.47</td>
<td>25.88</td>
</tr>
<tr>
<td>4</td>
<td>0.0070</td>
<td>6.838</td>
<td>4.276</td>
<td>11.02</td>
<td>23.10</td>
</tr>
</tbody>
</table>

Chapter 4 Production Optimisation
4.2.2 Multiple Snapshot Pre-Training and Point Selection Strategy

In [1], a gradient-free method (Hooke Jeeves Direct Search) was used to perform production optimisation. This method uses a pattern search to evaluate the objective function. Systematic changes to the input parameters are made around a base point. Trust regions are used to search for the optimal solution and thus the POD-TPWL model can be trained to approximate the outputs of these trust regions prior to the optimisation procedure.

The gradient-based adjoint optimisation uses the results from the backward simulation to determine the search direction. Therefore you know the maximum and minimum values for the input values, but the general search pattern is unknown before the optimisation is performed.

The water flooding optimisation was performed using both multiple training runs and a single training run performed before the optimisation. The retraining criteria was implemented for both methods and the results are presented in section 4.3.

With the multiple number of snapshots involved, a point selection strategy was needed in order to select the training point that is most closely related to the model. Similar to the retrain criteria of production optimisation, the point selection method involves calculating the shortest distance between the model parameters, controls and the state of the previous time step. This concept was developed in [1] and was applied in a similar manner for this body of work. The distance between the current solution and the saved point \( k \) in snapshot \( i \) is defined as:

\[
d_{i}^{k} = \alpha d_{x_{i}}^{i,n} + \beta d_{u_{i}}^{i,n} + \delta d_{\theta_{i}}^{i,n},
\]

where \( \alpha \) (equal to 1 for this work), \( \beta \) (equal to 50) and \( \delta \) (equal to 2) are weight multipliers set by the user to scale the distances accordingly. \( d_{x_{i}}^{i,n}, d_{u_{i}}^{i,n} \) and \( d_{\theta_{i}}^{i,n} \) are the distance between the new \( (n) \) and training states \( (i) \) and are defined as:

\[
d_{x_{i-1}}^{i,n} = \sum |x_{i-1}^{i} - x_{i-1}^{k}|, \\
d_{u_{i}}^{i,n} = \sum |u_{i}^{i} - u_{i}^{k}|, \\
d_{\theta_{i}}^{i,n} = \sum |\theta_{i}^{i} - \theta_{i}^{k}|,
\]

the absolute difference between the vectors is summed for each available training run and the minimum distance is chosen for each time step.

4.3 POD-TPWL Water Flooding Optimisation Results

An optimisation loop was performed using the Egg model realization number 69. This was done using the criteria previously discussed. The optimisation was run using the POD-TPWL adjoint method and POD-TPWL forward simulation method. The economic parameters for the cost function are in Table 9.
Table 9 Economic parameters used for the NPV objective function

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_i$</td>
<td>6 USD/m</td>
</tr>
<tr>
<td>$w_p$</td>
<td>19 USD/m</td>
</tr>
<tr>
<td>$o_r$</td>
<td>126 USD/m</td>
</tr>
<tr>
<td>$b$</td>
<td>0%</td>
</tr>
</tbody>
</table>

Large uncertainties exist in the economic values presented in Table 9 throughout the life of an oil and gas project, these were chosen as fixed values as is common in oil field evaluation. No discount factor was used for the optimisation making production for a longer time more attractive as there is no time value assigned to money.

40 control steps were used in the 3600 day simulation (control update every 90 days) and the BHPs in the 8 injectors were updated, meaning that 320 variables were controlled during the optimisation. The retraining of the POD-TPWL single model was done at iterations 1, 6 and 21. Three training runs of the POD-TPWL multiple model were done before the optimisation began, the model was also retrained at iterations 7, 10, 18 and 24. The training run schedules are shown in Figure 17 and Figure 18. The results of the water flood optimisation for 3600 days are shown in Figure 19 and Figure 20.

Figure 17 Training run and optimised well schedules for the 8 injection wells, single training run
Figure 18 Training run and optimised well schedules for the 8 injection wells, multiple training run

Figure 17 and Figure 18 show that the multiple training run method covers a larger range of the potential control space. The disadvantage of the multiple training run method is that there may be training runs that are never utilized during the gradient-based optimisation. The single retrain method guarantees that the training runs will be used during POD-TPWL optimisation, although the lack of training runs prior to the optimisation may lead to inaccurate results. Figure 17 shows that the training simulations for the single retrain method are very similar after the initial schedule, this may cause some of the nonlinearities to be overlooked by POD-TPWL.
Chapter 4 Production Optimisation

Figure 19 Time comparison between the HF and POD-TPWL for the water flooding optimisation of the Egg model

Figure 20 NPV comparison between the HF and POD-TPWL using water flooding optimisation
The time for the entire POD-TPWL single optimisation (52 minutes) when compared to the HF optimisation (324 minutes) and POD-TPWL multiple optimisation (119 minutes) was significant. The POD-TPWL single optimisation was able to calculate a converged optimal solution in the HF time equivalent of 4 simulations. These times include all of the time needed to process the derivatives for the POD-TPWL model. Figure 19 shows that the largest percentage of time for the POD-TPWL is committed to running the HF model and performing the inverse projection. The retraining criterion, used to ensure accuracy and to limit the number of HF runs and projection computation, becomes a key aspect of the POD-TPWL optimisation procedure, where the accuracy of the optimisation and the time needed to perform the optimisation are dependent on the retraining criteria.

In Figure 20, a very small discrepancy (0.1%) in the NPV was observed between the HF and POD-TPWL models. The HF model achieved a converged optimal solution in fewer iterations, meaning that the HF simulation was able to find a solution more efficiently simulation number wise than the POD-TPWL models.

The POD-TPWL optimised schedule was run with the HF model at the end of the optimisation loop (iteration 40) to determine the ability of the POD-TPWL to match the HF results. The 3 optimal schedules are shown in Figure 21.

Figure 21 shows that the optimal schedule for the injectors are dissimilar for the POD-TPWL and HF optimisation. The inaccuracies of POD-TPWL may have allowed for the optimal solution to be overlooked and, thus the adjoint equation would not have converged on that correct solution. The HF simulation using the optimal schedules in Figure 21 of each of the POD-TPWL models produced a HF NPV of 44.5 MMUSD (POD-TPWL single pre-training) and 45.6 MMUSD (POD-TPWL multiple pre-training). These
NPVs lead to the conclusion that the errors in POD-TPWL allowed for the adjoint method to overlook the optimal solution.

The error in the optimal cumulative POD-TPWL multiple training proxy and single training proxy are compared to the HF optimal cumulative production in Table 10.

**Table 10** Cumulative production and injection errors from the HF optimal solution for single and multiple POD-TPWL training methods

<table>
<thead>
<tr>
<th></th>
<th>Training Method</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiple</td>
<td>Single</td>
<td></td>
</tr>
<tr>
<td>$E_o [%]$</td>
<td>4.48</td>
<td>3.82</td>
<td></td>
</tr>
<tr>
<td>$E_{w, inj} [%]$</td>
<td>13.01</td>
<td>22.3</td>
<td></td>
</tr>
<tr>
<td>$E_{w, prod} [%]$</td>
<td>14.22</td>
<td>41.62</td>
<td></td>
</tr>
</tbody>
</table>

Table 10 suggests that although the predicted NPV is somewhat accurate when used with the POD-TPWL method, the optimal well schedule and production rates are not. This has a far reaching affect in terms of reservoir management, as the produced optimal schedule may not be optimal even though a matching NPV is calculated. The water and oil production rate errors described in section 2 become a major issue, an issue that could be rectified with more accurate and efficient proxy training. The inherent problem with the POD-TPWL method and other ROM methods is demonstrated in these results, which is that 100% accuracy can never be retained. The NPV calculation includes many different outputs from the reservoir simulator and has the capability of being predicted with a small amount of error using POD-TPWL. If the objective of the investigation is to determine the precise production values or to find very precise control settings, then POD-TPWL is not a valid option.

Another explanation for the match in NPV between the POD-TPWL and HF model and the mismatch in optimal schedules is the prices used for the NPV calculation. The oil price is weighted much stronger than the water costs, meaning that errors in oil production are more strongly accounted for in the NPV. Table 10 shows a much higher (5-10 times higher) error for water production when compared to oil production, this would account for a smaller error in the NPV.
5 Robust Optimisation

One of the barriers of using systems based optimisation in an operational setting is the fact that geological models have some level of uncertainty associated with them. Using an ensemble of geological models, such as the Egg model ensemble, can help to reduce the uncertainty of the optimisation. [21], [24] and [7] explored the option of using robust optimisation to reduce the uncertainty by using an ensemble of geological models to perform production optimisation. The objective values of the ensemble are then used to maximize the expected value of the ensemble, $E_0$, according to:

$$\max_{u_{i\alpha}} E_0 \left[ J \left( u_{i\alpha}, y^{iR}_{i\alpha} \left( u_{i\alpha} \right), \theta^{iR} \right) \right] \approx \max_{u_{i\alpha}} \frac{1}{R} \sum_{i=1}^{R} J \left( u_{i\alpha}, y^{i}_{i\alpha} \left( u_{i\alpha} \right), \theta^i \right),$$

where $R$ is the number of geological realizations. This method was able to more accurately predict the NPV by reducing the standard deviation when compared to nominal optimisation and reactive water flooding control strategy. The drawback of this method is the amount of simulation time required to perform multiple optimisation strategies. ROM using the POD-TPWL optimisation method is an excellent candidate to reduce the amount of simulation time required to perform the robust optimisation.

5.1 POD-TPWL Robust Optimisation

As multiple geological realizations and the controls are changed during the robust optimisation process, a POD-TPWL model containing both controls and model parameters is needed. Using similar methods to both section 4.3 and section A.1 the governing equation becomes:

$$g_k (x_k, x_{k-1}, u_k, \theta_k) = 0,$$

Where $\theta_k$ is the log of the transmissibility. The linearized form of the governing equation is thus:

$$g^r_k (x^r_k, x^r_{k-1}, u^r_k, \theta^r_k) - g^l_k (x^l_k, x^l_{k-1}, u^l_k, \theta^l_k) = \left( \frac{\partial g^l_k}{\partial x^l_{k-1}} \right) (x^r_{k-1} - x^l_{k-1}) ...$$

$$+ \left( \frac{\partial g^l_k}{\partial x^l_k} \right) (x^r_k - x^l_k) ...$$

$$+ \left( \frac{\partial g^l_k}{\partial \theta^l_k} \right) (\theta^r_k - \theta^l_k) ...$$

$$+ \left( \frac{\partial g^l_k}{\partial u^l_k} \right) (u^r_k - u^l_k),$$

where the equation includes both model parameters and controls, satisfying the requirement for robust optimisation. The rearranged version of equation (5.3) then becomes:

---

Chapter 5 Robust Optimisation

37
\[ x_i^* = x_i^t + \left( E_i^t \right)^{-1} \left[ A_i^t (x_i^t - x_{t-1}^t) + B_i^t (u_i^t - u_i^t) + \bar{M}_i^t (\theta_i^t - \theta_i^t) \right], \quad (5.4) \]

where \( \bar{M}_i^t = \left( \frac{\partial g_i^t}{\partial \theta_i^t} \right) \). Equations (3.12) and (3.13) were used to optimise the schedule using the expected value of the NPV as the objective function.

### 5.2 Implementation

In section 4, training simulations were generated as the optimisation loop progressed using the retrain criteria. In robust optimisation, many of the models contain similar geological features and permeability values. The linearized models for the POD-TPWL procedure are thus similar when compared to other models with similar features and permeability values. The snapshots were shared for each of the 100 ensemble members to reduce the number of retraining snapshots that were needed. The point selection strategy in section 4.2.2 was used for robust optimisation.

#### 5.2.1 Retraining Criteria

The accuracy of the POD-TPWL model for robust optimisation has to be quantified. The distance from the training run in terms of model parameters (\( E_m^t \)) can be expressed as:

\[ E_m = \frac{1}{T \sqrt{N_0}} \sum_{i=1}^{K} \left( \| \theta_i^t - \theta_i^t \| \times \Delta_k \right), \quad (5.5) \]

POD-TPWL simulations were then run using different inputs of \( \theta \), which in our case was defined as the log of transmissibility. The oil and water production results were then compared to the HF simulation using MRST, the results are found in Table 11. In Table 11, the transmissibility field column refers to the transmissibility field used for the POD-TPWL simulation.

<table>
<thead>
<tr>
<th>Transmissibility field</th>
<th>( E_m )</th>
<th>( E_x )</th>
<th>( E_o ) [%]</th>
<th>( E_{w,\text{inj}} ) [%]</th>
<th>( E_{w,\text{prod}} ) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3341</td>
<td>0.1199</td>
<td>0.0057</td>
<td>0.0212</td>
<td>0.0749</td>
</tr>
<tr>
<td>2</td>
<td>1.0455</td>
<td>0.2487</td>
<td>0.0333</td>
<td>0.1008</td>
<td>0.2609</td>
</tr>
<tr>
<td>3</td>
<td>1.8108</td>
<td>0.3763</td>
<td>0.0709</td>
<td>0.1618</td>
<td>0.3778</td>
</tr>
<tr>
<td>4</td>
<td>2.5831</td>
<td>0.4847</td>
<td>0.1151</td>
<td>0.2103</td>
<td>0.4578</td>
</tr>
</tbody>
</table>

The error in the cumulative oil production (\( E_o \)) is again much smaller than that of the cumulative water injection and production (\( E_{w,\text{inj}} \) and \( E_{w,\text{prod}} \)). Using error limits, it can be concluded that the model error should be limited to 1.0. This is due to the large increase of the cumulative production error after this point.
5.3 Robust Optimisation Results

A robust optimisation was done for the ensemble of geological realizations of the Egg model. The POD-TPWL method generated 49 training images during the robust optimisation for the 100 geological realizations and the 25 optimisation iterations. These results were compared with results from Shell’s reservoir simulator, MoRes, where a robust optimisation of the Egg model was performed [25]. These results were used as a HF simulation with the MRST AD-FI module would have taken approximately 52 days. The POD-TPWL robust optimisation was run in 50 hours on one core of a 2.5 GHz Intel Core i5 with 16 GB of 1600 MHz DDR3 RAM. The results of the two robust optimisations are presented in Figure 22 and Figure 23. The probability density function was fit to a normal distribution.

![Cumulative distribution function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL](image1)

**Figure 22** Cumulative distribution function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL

![Probability density function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL](image2)

**Figure 23** Probability density function of the optimised NPV for 100 realizations of the Egg model ensemble using MoRes and POD-TPWL
Figure 22 and Figure 23 have different end-points as the probability density function was fit to a normal distribution and thus some values were binned with others. With the POD-TPWL robust optimisation, there a few values that deviated quite significantly from the mean value of the ensemble. These outliers have a significant impact on the optimisation, as the adjoint optimisation is based on the value obtained for the NPV. These outliers being averaged out would improve the robust optimisation deviation and mean when compared to a nominal optimisation (obtaining an optimal schedule for each member of the ensemble).

The mean of the NPV from the two different simulators differed by 0.7%. The variance in the POD-TPWL results was larger than that of the HF simulation done in MoRes. This was slightly lower than the single model optimisation performed in section 4.3, as was the expected result [7]. The impact of the inaccuracy experienced with POD-TPWL is decreased by averaging multiple values, which on their own would not be an accurate representation of the NPV. This is a similar conclusion as to what was observed for the single optimisation run in section 4.3. The question of accuracy will always be inherent with any sort of ROM. Depending on the application, the user must determine if the inaccuracy of the model is acceptable. For a field application of production optimisation, controls may be subject to uncertainties of +/- 10% [5], meaning that even if you have an optimal control schedule you may only be able to set the controls to within 10% of the determined optimum. Therefore, if the error in the schedule obtained from POD-TPWL is within the measurement tolerance when compared to a HF simulation, you may reasonably expect that the POD-TPWL schedule would remain useful.
6 Conclusion

- A ROM gradient-based production optimisation was performed using the POD-TPWL method in conjunction with MRST’s AD-FI module. The reduction techniques were able to linearize the model (TPWL) eliminating the need for nonlinear iterations during the simulation and reduce the number of parameters needed for investigation (POD). The linear ROM was subsequently solved in a low-dimensional space repeatedly during the production optimisation.
- The production optimisation was also performed with a ROM adjoint method where the model was both reduced (POD) and expanded around a point using a first order Taylor series expansion. All simulations were performed using the Egg model from [15].
- The results from the POD-TPWL numerical experiments using BHP control of the injectors showed a comparable result to the HF experiments for NPV. Both the adjoint gradient and production values from the wells showed to be dissimilar to the HF model. A speed increase of 6 times was observed using POD-TPWL. POD-TPWL was much faster than the HF model (100 times for the forward, 40 times for the adjoint), the actual speedup was much less as many training runs and projections of the HF simulation was necessary. Greater speeds can thus be achieved if the retraining and projection is further optimised.
- The robust optimisation was performed an estimated 30 times faster than using the MRST AD-FI module. The results were compared to results from [25], which used MoRes as the HF simulator. The error in the mean was very low (0.7%), but the inherent inaccuracies in the POD-TPWL method led to a greater variance of the NPV results.
- The accuracy in both the forward and adjoint was not exact, but the objective function values proved to be accurate. This makes the case that precision is not completely necessary for every part of the reservoir modelling process involved in production optimisation.

6.1 Recommendations

- The accuracy of the jacobians obtained from MRST’s AD-FI module are unknown. [1] and [2] were able to conduct POD-TPWL using Stanford’s automatic differentiation general purpose reservoir simulator (AD-GPRS). This was available to the author and was experimented with some of the optimisation runs not presented here. AD-GPRS, similar to MRST, was able to save the derivatives needed for the POD-TPWL method [26]. This work was not done using AD-GPRS as the derivatives with respect to model parameters was not available to the author at the time of doing this work. The simulation speed using AD-GPRS was approximately 10 times faster than using MRST. The POD-TPWL method should be implemented in a faster programming language, such as C++, or incorporated into AD-GPRS or another reservoir simulator (such as Shell’s MoRes) to calculate true gains possible.
- All of the literature [1] [2] [14], as well as this body of work, performed POD-TPWL using BHP control. This leads to issues, such as high injection rate fluctuations with a small change in the BHP for low permeability grid blocks. Derivatives (ie. \( \partial g_i / \partial u_i \)) where \( u_i \) is the well rate) should, theoretically, be able
to be saved during the automatic differentiation procedure. However, during this investigation, the author was unable to do so in an accurate or efficient manner for POD-TPWL. More work should be done in this area, as it would greatly increase the robustness of the POD-TPWL method.

- The retraining criteria, as previously mentioned, is currently being investigated in [14]. One of the major limitations of the POD-TPWL is the need for many surrogate, HF models. A streamlined process where the minimum number of surrogate models would be optimal and should be continued to be investigated. Methods such as genetic algorithms or artificial neural networks could have potential for this area.

- Promising work in the field of basis function construction for nonlinear model order reduction has been investigated by multiple other authors. The use of spatial-temporal tensor decomposition [27] or discrete empirical interpolation (DEIM) [28] can be used instead of using POD.

- The length of calculating a stable model using the Inverse Projection was quite time consuming (equal to approximately one forward simulation). Stability of the model and calculation time can be improved by either using one of the methods discussed in the previous bullet or by programming the projection in a more efficient way.

- If a high level of accuracy is needed and a fast simulation, GPU computing or parallel computing with an accurate reservoir simulator may be a better alternative to ROM [29]. This would allow for both computational speed-up and the retention of an accurate result.
References


“Quantification of the impact of ensemble size on the quality of an ensemble
gradient using principles of hypothesis testing,” Under review @ SPE Journal, 2015.


[27] E. Insuasty, P. van den Hof, S. Weiland and J. Jansen, “Tensor-based reduced order
modeling in reservoir engineering: An application to production optimization,” in 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Florianopolis, Brazil, 2015.


[29] K. Esler, K. Mukundakrishnan, V. Natoli, J. Shumway, Y. Zhang and J. Gilman,
“Realizing the potential of GPUs for Reservoir Simulation,” in ECMOR XIV - 14th


gradient-based optimization in closed-loop reservoir management,” Computational
Appendix A  History Matching

During reservoir characterization, uncertain parameters are in general grid block permeabilities and porosities, relative permeabilities, or fault multipliers. In some reservoir evaluation cases, one geological model is constructed using localized borehole information, seismic data and geological knowledge and is applied throughout the reservoir to create an inherently uncertain model. This uncertainty can be accounted for by using an ensemble of geological models, which can incorporate larger range of uncertain parameters, for the reservoir evaluation. History matching attempts to reduce the uncertainty in the model parameters by using data from the production in the field. The reservoir models can then be dynamically updated, providing useful insight to determine good production strategies.

In our case, the transmissibilities were updated in the prior model using water and oil production rate measurements from the real field production measurements. As this work is not being applied to an actual field, a twin experiment was done where one of the members of the Egg model ensemble was selected as the truth to obtain field production measurements. The mismatch between the truth production measurements and the outputs from the reservoir simulator was minimized by updating the transmissibilities using the adjoint method. This method was used as it is well developed and was used in a number of studies [4] [30] [5] [31]. The objective function honours both the prior model parameters and the data mismatch:

\[ J(y_{ik},\theta_{ik}) = \sum_{k=1}^{K} \left[ (d_k - y_k)^T P_y^{-1} (d_k - y_k) + (\theta_k - \bar{\theta}_k)^T P_\theta^{-1} (\theta_k - \bar{\theta}_k) \right], \quad (6.1) \]

where \( d_k \) is the production data from the truth, \( y_k \) is the production data from the model, \( P_y \) is the error covariance matrix for the data measurements, \( \theta_k \) is the transmissibility of the model, \( \theta_i \) is the transmissibility of the prior model and \( P_\theta \) is the covariance matrix of the prior permeability field. To avoid scaling affects, the log of the transmissibilities was used for history matching. In order to minimize the objective function in equation (6.1), many iterations are needed to improve the match, thus large numbers of simulations were needed in order to accurately move towards the true reservoir geological setting, a need for surrogate models exists.

A.1 POD-TPWL Formulation

The work in [1] included the development of a POD-TPWL model to be used with varying model parameters. The formulation was similar to the construction of the equations introduced for production optimisation. By changing the governing equation to be a function of the model parameters it then becomes:

\[ g_\theta(x_k, x_{k-1}, \theta_k) = 0. \quad (6.2) \]

This can then be linearized around a training point \( i \), equation (2.18) then becomes:
where the approximation made in section 2.2, \( g_k^n(x^n_k, x^n_{k-1}, \theta^n_k) - g_k^l(x^l_k, x^l_{k-1}, \theta^l_k) = 0 \), is assumed to hold true. The derivatives with respect to model parameters can be written in a more concise form using the same method from the POD-TPWL section:

\[
\bar{M}_k = \left( \frac{\partial g_k^l}{\partial \theta_k} \right),
\]

Equation (6.5) was again reduced using POD and the model was projected using the inverse projection. The reduced derivatives and corresponding POD-TPWL equation are:

\[
\begin{align*}
\bar{E}_{r,k} &= \Phi^T \left( \bar{E}_k^l \right)^{-1} \bar{E}_k^l \Phi, \\
\bar{A}_{r,k} &= \Phi^T \left( \bar{E}_k^l \right)^{-1} \bar{A}_k^l \Phi, \\
\bar{M}_{r,k} &= \Phi^T \left( \bar{E}_k^l \right)^{-1} \bar{M}_k^l, \\
\bar{z}_{r,k} &= \bar{z}_k^l + \left( \bar{E}_{r,k}^T \right)^{-1} \left[ \bar{A}_{r,k} (\bar{z}_{k-1}^n - \bar{z}_{k-1}^l) + \bar{M}_{r,k} (\theta_k^n - \theta_k^l) \right].
\end{align*}
\]

POD-TPWL adjoint equations (3.12) and (3.13) were used for the history matching problem as well. The objective function used was (6.1), thus replacing \( \partial J / \partial u_k \) with \( \partial J / \partial \theta_k \) and the other derivatives in equations (3.12) and (3.13) also being in terms of \( \theta \) instead of \( u \). The steepest descent method was used to find the minimum value of the objective function (instead of the maximum outlined for production optimisation).

### A.2 POD-TPWL History Matching Results

The Egg model, as described in section 1.2, was used to perform the history matching. The truth was chosen from one of the realizations constructed in [15]. The prior model parameters used was the mean of the Egg model parameters. The results of the POD-TPWL history matching and the HF simulation using MRST were compared to the truth (another realization of the Egg model). The objective function value over the history matching procedure (22 iterations for both ROM and HF) can be seen in Figure 24 and the computing time between the HF and ROM procedure can be seen in Figure 25.
Figure 24 Evolution of the objective function value for both the HF procedure (blue) and POD-TPWL procedure (red)

Figure 25 Time for both the HF and ROM procedure broken down into individual components of the modelling process

The ROM had to be retrained 4 times during the history matching procedure (iteration 1, 2, 7 and 14). The training at iteration 1 did not require a projection matrix as the HF model was also run at iteration 2. These two close retraining were done in order to obtain the $\partial J/\partial \theta$ variable from MRST. Using MRST and the objective function, $\partial J/\partial \theta$ was equal to...
to zero for the first iteration because $\theta_0 = \theta_{\text{prior}}$. The ROM history matching procedure ran approximately 3 times faster than the HF model (5000 s versus 15000 s). Again, although the ROM ran approximately 100 times faster, the need for retraining and projecting the matrix required large amounts of time.

The objective function in Figure 24 greatly improves whenever a retrain is done. The adjoint model is reasonable at calculating the gradient needed to improve the transmissibility, but the forward POD-TPWL has poorer performance than the HF simulation in determining the rates of the updated model parameters and thus the objective function.

The history matching was done after running an optimisation of the production schedule for 1800 days. The predictive capability was also compared, all of the results can be seen in Figure 26, Figure 27 and Figure 28. Note the discontinuity in the production graphs is due to the fact that the schedule was optimised for the first 1800 days, the schedule thereafter was the initial schedule.

Figure 26 Comparison of the oil rates after history matching for the POD-TPWL, HF and truth model
Figure 27 Comparison of the water production rates after history matching for the ROM, HF and truth model

Figure 28 Comparison of the log-permeability fields for the HF, ROM, prior and truth model
Figure 26 and Figure 27 show a good match for the well production data both for history matching and the predictive capabilities. The deviation of the production data from the prior was a fair amount and was able to be reproduced fairly accurately.

Figure 28 shows the channels represented in the truth were not able to be reproduced by the history matching procedure for neither the HF nor POD-TPWL fields. In both the HF and POD-TPWL fields, an increase of permeability was observed around some of the wells that lie in or around high permeability channels of the truth. This is to be expected as the history matching objective function relies on both production data and the prior model geological parameters to match the geological parameters of the history matched model.

These results show promise for POD-TPWL to be used for history matching. More investigation should be done regarding the update of model parameters before making more definitive conclusions.