

PRESSURE PRECONDITIONING USING PROPER  
ORTHOGONAL DECOMPOSITION

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

Rui Jiang

December 2013

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

---

(Hamdi A. Tchelepi) Principal Co-Adviser

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

---

(Louis J. Durlofsky) Principal Co-Adviser

# Abstract

A linear-solver preconditioner for the pressure systems in reservoir simulation based on Proper Orthogonal Decomposition (POD) is developed and investigated. The POD approach entails the use of a lower-dimensional space to approximate the original high-dimensional space. Essentially, POD can be considered as a Singular Value Decomposition (SVD) approach. The POD approximation error can be estimated by examining the singular values. The POD-based Reduced-Order Modeling (ROM) strategy projects the full linear system ( $\mathbf{Ax} = \mathbf{b}$ ) into a reduced form ( $\mathbf{A}_r\mathbf{z} = \mathbf{b}_r$ ). The resulting ROM can be used to explore the parameter space of interest at a much lower computational cost compared with the full model.

The utility of the ROM model depends on its accuracy in representing the solutions of the full-order model. It is shown that among the different schemes, Galerkin projection gives the best accuracy. In order to quantify the quality of a POD basis for a specific linear system, algebraic indicators are proposed. Various issues are discussed related to designing an adaptive pressure preconditioning strategy, in which the simulator can adjust the POD space adaptively when the approximation becomes less accurate. A spectral-analysis approach for assessing POD as a pressure preconditioner is presented. A two-dimensional example is used to demonstrate the high and low frequencies that comprise the solution space, of which the low-frequency errors are more difficult to resolve. Numerical experiments for oil/water reservoir

models using the IMPES scheme show that the POD preconditioner can resolve certain eigenvectors, though the errors associated with significant parts of the spectrum are left untouched. The POD method is effective when the right-hand-side (RHS) encountered in the simulation contains only a few components with frequencies that the POD basis has no effect on. Unlike other preconditioners that solve the problem equally well for any RHS, the POD preconditioner is RHS dependent. Hence, the POD preconditioner does not appear to be an ideal choice as a generic pressure preconditioner.

# Acknowledgements

I would like to express my deepest gratitude to my advisers, Professors Hamdi Tchelepi and Louis Durlofsky, for their constant support and encouragement. They have played an essential role in my academic life ever since I came to Stanford. They offered me many valuable insights. Their professional guidance greatly improved my ability. They always point out promising ways to handle problems and bring me out of the mist. I have also benefited from their working style, time management and spirit of dedication. All of these merits are beyond words. I am really in debt to my two great advisers.

I would like to thank the Energy Resources Engineering department, and every faculty member for their selfless work in teaching, researching and improving our department. I also want to express my appreciation to all ERE students and especially my friends, such as Yixuan Wang, Boxiao Li, Jincong He, Hamza Aljamaan, Sumeet Trehan, Mehrdad Gharib Shirangi, and Khalid Alnoaimi, and my office mates such as John and Joseph Lagasca and Hangyu Li. I am so honored to spend my most valuable time in such a dynamic, diversified and positively energized place. Moreover, I would like to thank my friend Shang Deng in the GES (Geological and Environmental Sciences) department for lots of generous help and many inspiring discussions.

Finally, I would like to thank my parents and my parents-in-law. Their unconditional support lays the foundation for my improvement and success. I would like to express my gratitude and love to my wife Wen Zhao. It is her love that has made my

previously monochrome life colorful.

# Contents

<b>Abstract</b>	iii
<b>Acknowledgements</b>	v
<b>1 Introduction</b>	1
<b>2 POD–ROM for Linear Systems</b>	7
2.1 POD-Based ROM for Linear Systems . . . . .	7
2.2 Different Choices of $\Psi$ . . . . .	9
2.3 Choosing $\Psi$ for Coarse-Scale Mass Balance . . . . .	11
2.4 Adaptive POD Solution Strategy . . . . .	12
2.4.1 Quality of POD basis . . . . .	12
2.4.2 Contribution of Different Columns . . . . .	14
2.4.3 Dynamic POD Subspace . . . . .	14
2.4.4 Computational Cost . . . . .	14
<b>3 Spectral Analysis of the POD Preconditioner</b>	17
3.1 Spectral Analysis . . . . .	17
3.2 High and Low Frequency Components . . . . .	19
3.3 Effect of the POD Space . . . . .	21
3.4 Iteration Matrix Spectrum . . . . .	24

<b>4 Numerical Experiments</b>	<b>27</b>
4.1 Governing Equations . . . . .	27
4.2 Problem Specification and Solution . . . . .	29
4.3 Convergence Behavior . . . . .	31
4.4 Iteration Matrix Spectrum . . . . .	34
4.5 Effect of POD on Error Spectrum . . . . .	38
4.6 RHS Dependence of the POD Preconditioner . . . . .	44
4.7 Additional Examples . . . . .	48
4.7.1 Heterogeneous Case ( $10 \times 10$ ) . . . . .	48
4.7.2 Heterogeneous Case ( $30 \times 30$ ) . . . . .	50
4.8 Summary . . . . .	50
<b>5 Conclusions and Future Work</b>	<b>55</b>
<b>Nomenclature</b>	<b>57</b>
<b>A Basics of the POD Method</b>	<b>61</b>
A.1 Definition of POD . . . . .	61
A.2 Principal Components . . . . .	64
A.3 POD Approximation of High-Dimensional States . . . . .	67
A.4 Computational Cost of SVD . . . . .	70
<b>Bibliography</b>	<b>73</b>

# List of Tables

2.1	Computational cost of the POD preconditioner for each linear problem.	15
2.2	Computational cost of the POD preconditioner for each preconditioner call.	15

# List of Figures

3.1	Simple 2D analysis on the POD method, for $\alpha = 37^\circ$ and $\theta = 58^\circ$ . Eigenvalues of $\mathbf{A}$ : $\lambda_1 = 2, \lambda_2 = 0.5$ . . . . .	21
3.2	Illustration of $  \mathbf{b}'   >   \mathbf{b}  $ , for $\alpha = 37^\circ$ and $\theta = 87^\circ$ . Eigenvalues of $\mathbf{A}$ : $\lambda_1 = 2, \lambda_2 = 0.5$ . . . . .	22
3.3	Error/residual reduction ratio versus $\theta$ for the POD method for $\alpha = 90^\circ$ and $\alpha = 85^\circ$ . The blue circle shows the direction of remaining $\mathbf{x}'$ for the case where $\hat{\mathbf{x}}$ is in the direction $\theta = 135^\circ$ . Eigenvalues of $\mathbf{A}$ : $\lambda_1 = 10, \lambda_2 = 0.1$ . . . . .	23
4.1	Reservoir model used for preconditioner investigation. . . . .	29
4.2	Solutions for the base case: water saturation $S_w$ (left) and pressure change $\delta\mathbf{p}$ (right). . . . .	32
4.3	Energy plot for the POD basis showing the importance and contribution of different singular vectors. The error indicator gives us an idea of the reconstruction error, see (A.9). . . . .	33
4.4	Eigenvalues of the Jacobian matrix for Step 15. . . . .	33
4.5	Jacobian matrix condition numbers. . . . .	34
4.6	Convergence history: Step 2 (top) and 15 (bottom). The red curves in each plot correspond to $\ell = 5, 10, 15, 20, 39$ . The blue curve is for the ILU preconditioner. . . . .	35

4.7	Convergence history: Step 35 (top) and 45 (bottom). The red curves in each plot correspond to $\ell = 5, 10, 15, 20, 39$ . The blue curve is for ILU-only preconditioner. . . . .	36
4.8	Convergence history: Step 100 (top) and 221 (bottom). The red curves in each plot correspond to $\ell = 5, 10, 15, 20, 39$ . The blue curve is for ILU-only preconditioner. Note that POD+ILU with $\ell = 5$ increases the residual norm for the first iteration. . . . .	37
4.9	Plots of eigenvalues of iteration matrix for Step 15. ILU-only (top left), POD+ILU with $\ell = 5$ (top right), $\ell = 15$ (middle left), $\ell = 32$ (middle right), $\ell = 33$ (bottom left), and $\ell = 39$ (bottom right). The largest five (magnitude) eigenvalues are listed in each plot. . . . .	39
4.10	Convergence history with the trend given by spectral radius for Step 15, $\ell = 5, 10, 15, 20$ . The dashed line shows the convergence trend given by the spectral radius. . . . .	40
4.11	Eigen-analysis by taking eigenvectors of $\mathbf{A}$ as RHS for Step 15, using ILU-only. . . . .	41
4.12	Eigen-analysis by taking one eigenvector of $\mathbf{A}$ as the RHS: Eigenvector number 50 (upper) and eigenvector number 78 (lower), Step 15, using ILU-only. . . . .	42
4.13	Eigen-analysis by taking eigenvectors of $\mathbf{A}$ as RHS: Step 15, POD-only, $\ell = 5, 10, 15, 20$ . Full range (top) and local detail (bottom). Note that the POD has no effect on many frequency modes. . . . .	43
4.14	Eigen-analysis by taking one eigenvector of $\mathbf{A}$ as the RHS: Eigenvector number 3 with $\ell = 20$ (upper) and eigenvector number 64 with $\ell = 10$ (lower), Step 15, POD preconditioner. . . . .	45

4.15 Convergence history with additional random RHS: Step 15, $\ell = 10, 15, 20, 39$ . Randomly generated RHS shown in black. RHS encountered in the problem shown in red. The red dashed lines are convergence trends given by the spectral radii of iteration matrices.	46
4.16 Convergence history with random RHS, where $\Phi$ additionally includes the 33rd column.	46
4.17 Spectral comparison between POD effectiveness and RHS components. Residual norm reduction after one iteration (thin curves). Normalized RHS spectrum (blue thick curve). Results for Step 15.	47
4.18 Porosity (left) and $\log_{10}$ -permeability (right) distribution for the heterogeneous case.	48
4.19 Saturation maps for the heterogeneous case, before (left) and after (right) water breakthrough.	49
4.20 Plots of eigenvalues of iteration matrix for the heterogeneous case, Step 70. POD+ILU with $\ell = 10$ (top left), $\ell = 20$ (top right), $\ell = 25$ (bottom left), and $\ell = 30$ (bottom right). The largest (magnitude) eigenvalues are listed in each plot.	49
4.21 Convergence history for the heterogeneous case ( $10 \times 10$ ) for Step 70. The red curves correspond to $\ell = 10, 20, 25, 30$ . The dashed lines show convergence trends given by the spectral radius. The lower plot shows comparison with the case with a random RHS vector, $\ell = 30$ .	51
4.22 Log <sub>10</sub> -permeability distribution for the larger case.	52
4.23 Plots of eigenvalues of iteration matrix for the larger case for Step 90. $T = 77.4$ days. ILU-only (top left), POD+ILU with $\ell = 25$ (top right), $\ell = 55$ (bottom left), and $\ell = 75$ (bottom right). The largest eigenvalues are listed in each plot.	52

4.24 Convergence history for the larger case for Step 90. The red curves correspond to $\ell = 25, 55, 75$ . The dashed lines show convergence trends given by the spectral radius. The black curves correspond to a random RHS. . . . .	53
A.1 POD illustration. Top: three pressure fields (snapshots), $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ . Bottom: first principal component $\mathbf{u}_1$ , with singular value 1678. . . . .	65
A.2 POD illustration. Top: snapshots after subtracting the first PC component, $\mathbf{x}_i - \mathbf{u}_1 \mathbf{u}_1^T \mathbf{x}_i, i = 1, 2, 3$ . Bottom: second principal component, $\mathbf{u}_2$ , with singular value 34.49. . . . .	66
A.3 POD illustration. Top: snapshots after subtracting the first two PC components, $\mathbf{x}_i - \mathbf{u}_1 \mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_2 \mathbf{u}_2^T \mathbf{x}_i, i = 1, 2, 3$ . Bottom: third principal component, $\mathbf{u}_3$ , with singular value 0.8132. . . . .	66
A.4 POD illustration. Top: three pressure fields (snapshots), $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ . Bottom: three principal components ( $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ ) with singular values 1.152, 0.7644, 0.5484, respectively. . . . .	67
A.5 Reconstruction error for different snapshots. For the red curves from top to bottom, $\ell=2, 6, 10, 14, 18$ , and 22. The thick black curve is the norm of the snapshots, shown as a reference. . . . .	70



# Chapter 1

## Introduction

The aim of Reduced-Order Modeling (ROM) methods is to transform a high-dimensional problem into a model with a reduced number of dimensions. The idea is to then use the reduced-order model to explore the parameter space of interest at a lower cost compared with the full model. ROM is a rather broad concept, and it is used in a wide range of fields. Detailed description of ROM can be found in [1–3].

Projection-based model reduction is an important component of ROM. In terms of linear systems, which are the primary focus of this thesis, projection-based ROM can be described as follows: project the original problem  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A}$  is  $n \times n$  and  $\mathbf{x}$  and  $\mathbf{b}$  are vectors of size  $n$ , into a low-dimensional space spanned by an orthonormal basis  $\Phi$  ( $\in \mathbb{R}^{n \times m}$ ,  $m \ll n$ ). Through the approximate representation  $\mathbf{x} \approx \Phi\mathbf{z}$ , where  $\mathbf{z}$  is the reduced-order variable of size  $m$ , the reduced-order system can be written as  $\Phi^T \mathbf{A} \Phi \mathbf{z} = \Phi^T \mathbf{b}$ . Thus, the reduced system  $\mathbf{A}_r \mathbf{z} = \mathbf{b}_r$  is of order  $m$ , so we should be able to solve this smaller system at a lower cost. In order to approximate the solution of the original problem accurately, the true solution should lie in (or be as close as possible to) the reduced space,  $\mathcal{R}(\Phi)$ .

Proper Orthogonal Decomposition (POD), or Principal Component Analysis (PCA), is an efficient way to calculate the projection space,  $\mathcal{R}(\Phi)$ , from sample solutions,

which are referred to in this context as snapshots. The POD method ensures the maximum possible 2-norm accuracy for the approximate representation,  $\mathbf{x} \approx \Phi\mathbf{z}$  [4]. Given a collection of data points in high-dimensional space, POD captures their proper orthogonal modes, or principal components, using a series of vectors that are orthogonal to each other. The importance of these vectors can be quantified using the singular values obtained from the POD algorithm.

POD-based ROM has been applied in many different areas, such as computational fluid dynamics [5] and structural mechanics [6]. In recent years, POD-ROM has also been applied for reservoir simulation [7–11]. POD-ROM may be particularly useful when large numbers of simulations, with only minor changes in the problem description, need to be run. This is often the case for optimization or history matching applications (within the context of reservoir simulation).

Cardoso [3] and Cardoso et al. [8] developed a POD-ROM framework that was integrated into a general purpose reservoir simulator [12–15], where the linear solver was replaced by a POD reduced solution at each Newton iteration. They assumed that convergence was achieved when the solution update was smaller than a pre-specified threshold. They also implemented the “missing point estimation” technique to reduce the cost of linear solutions, and the “snapshot clustering” technique to improve the POD effectiveness. With numerical tests on an oil/water reservoir model with 60,000 grid cells, they reported speedup factors of  $O(10)$  with little loss in accuracy. Furthermore, they explored a method called Trajectory Piecewise Linearization (TPWL) for reservoir simulation, in which the entire simulation was run in the reduced space. In this method, new solutions are represented as linear approximations around previously converged solutions. This method was further extended by He et al. [11, 16], who applied the “local resolution” technique in well regions to improve the accuracy. In order to address the occasional instability of the TPWL method, He et al. proposed “basis optimization” and “equal density projection” methods to

reduce the spectral radius of an appropriately defined amplification matrix, which made TPWL more stable. In that work, significant speedups were achieved; namely speedup factors of 500–2000 for oil/water reservoir models with up to 80,000 grid cells. Applications of TPWL for history matching [17], optimization [18], and compositional modeling [19] have also been reported. Other research on POD-based ROM methods, involving the incorporation of nonlinear effects, includes the work of Trehan [20] and Chaturantabut and Sorensen [21]. The latter developed a method called Discrete Empirical Interpolation Method (DEIM), and this method may be applicable for reservoir simulation.

However, two aspects of the methods discussed above may limit their effectiveness. First, all of these methods require some redesign of the simulator. For example, the POD-ROM method [8] requires one to replace the original linear solver and use different convergence criteria. The TPWL method [11] has a completely different solution strategy at the nonlinear level because all the computations are performed in the reduced space. Second, one has to give up some accuracy in the process of speeding up the simulations, and the effectiveness of these approaches is usually demonstrated through numerical trial and error with little guidance by theoretical bounds, or a-priori error estimates.

An alternate approach is to use POD-ROM to accelerate the linear solution of the full-order system. The linear solver deals with the system  $\mathbf{Ax} = \mathbf{b}$ , or more conventionally  $\mathbf{J}\delta\mathbf{x} = -\mathbf{R}$ , where  $\mathbf{J} = \frac{d\mathbf{R}}{d\mathbf{x}}$  is the Jacobian matrix. It is a relatively independent part in the simulator workflow. Hence, modifications at the linear-solver level may be less intrusive to the simulator.

There are only two papers that investigated the use of POD-ROM for the linear systems associated with reservoir simulation problems. Markovinović and Jansen [9] used the POD-ROM method to improve the initial guess of the iterative linear solver. They reported a modest speedup of less than two, compared with the case

where the solution from the previous time step was used as the initial guess. Astrid et al. [10] used POD–ROM to replace the Algebraic MultiGrid (AMG) [22] method as the pressure preconditioner in a Constrained Pressure Residual (CPR)-type solver [23]. A speedup factor of three to five was reported. Although these studies are not exhaustive, they indicate that using POD–ROM at the linear-solver level may be beneficial.

There has been a great deal of effort on developing multi-level linear solvers for the pressure equation [22, 24–26]. The multi-level structure used in these studies is similar to that of the POD–ROM method, with the  $\Phi$  matrix defining restriction operators  $\mathcal{R}$  and prolongation operators  $\mathcal{P}$ , and the coarse-grid (or reduced) problem expressed as  $\mathcal{R}\mathbf{A}\mathcal{P} = \mathcal{R}\mathbf{b}$ . Therefore, there is a natural parallel between two-level methods, such as the Operator-Based Multiscale Method (OBMM) [24] and the Two-stage Algebraic MultiScale (TAMS) method [25], and the POD–ROM approach. In particular, Zhou and Tchelepi [25, 26] developed a general framework to assess the performance of a preconditioner for the near-elliptic pressure system in heterogeneous reservoirs. In their analysis of the preconditioner, the eigenvalue spectrum of the iteration matrix is analyzed. The low-frequency and high-frequency components of the error are treated separately by the preconditioner. Hence, this is a natural way to assess the quality of a pressure preconditioner.

Here, we follow a strategy similar to that of Zhou and Tchelepi [25, 26] to investigate the properties of POD–ROM as a pressure preconditioning approach. Chapter 2 focuses on the general ROM method for linear systems. Chapter 3 describes our spectral analysis of the POD pressure preconditioner. Finally, numerical results from two-dimensional oil-water reservoir models are presented and discussed in Chapter 4. We find that the POD preconditioner resolves some low-frequency eigenvectors but leaves others intact. We also show that the POD–ROM method is right-hand-side dependent. Unless one can find a way to either overcome the RHS dependency, or

somehow take advantage of it, the POD approach cannot serve as a robust preconditioner. Appendix A presents basic POD concepts and gives an intuitive picture about POD.



# Chapter 2

## POD–ROM for Linear Systems

In this chapter, we investigate the POD-based ROM method for linear systems in the context of reservoir simulation.

### 2.1 POD-Based ROM for Linear Systems

A linear system can be represented by the equation

$$\mathbf{Ax} = \mathbf{b}, \quad (2.1)$$

where  $\mathbf{x}$  and  $\mathbf{b}$  are  $n$ -dimensional vectors and  $\mathbf{A}$  is an  $n \times n$  square matrix. In the context of solving the pressure equation,  $n$  is the number of grid cells. Usually,  $n$  is of  $O(10^4 - 10^6)$ , so the solution of (2.1) in its original form can be time consuming (since this equation must be solved at each Newton iteration).

To reduce the order, or the size of the system, we substitute the approximate representation  $\mathbf{x} = \Phi \mathbf{z}$  into the original equation, where  $\Phi$  is the POD basis with  $\ell$  columns with  $\ell \ll n$ , and  $\mathbf{z}$  is an  $\ell$ -dimensional vector. Then, we have the reduced

equation

$$\mathbf{A}\Phi\mathbf{z} = \mathbf{b}. \quad (2.2)$$

However, this is an overdetermined problem. In order to obtain an approximate solution, we need to project the full set of equations into a lower-dimensional space. So, by applying the projection operator, also called the “constraint reduction matrix” [27],  $\Psi$ , on the left, we get:

$$\Psi^T \mathbf{A}\Phi\mathbf{z} = \Psi^T \mathbf{b}. \quad (2.3)$$

Hence, the reduced solution  $\mathbf{z}$  can be written as

$$\mathbf{z} = (\Psi^T \mathbf{A}\Phi)^{-1} \Psi^T \mathbf{b}, \quad (2.4)$$

and the approximate solution  $\tilde{\mathbf{x}}$  can be written as

$$\tilde{\mathbf{x}} = \Phi(\Psi^T \mathbf{A}\Phi)^{-1} \Psi^T \mathbf{b} \approx \mathbf{A}^{-1} \mathbf{b}. \quad (2.5)$$

This framework for generating the reduced problem (2.3) is called the Petrov–Galerkin procedure [28]. It provides the basis for many iterative methods for solving linear problems. For example, one can consider the column space,  $\mathcal{R}(\Phi)$ , as the search subspace  $\mathcal{K}$ , and  $\mathcal{R}(\Psi)$  as the left subspace of constraints  $\mathcal{L}$ . Finding the reduced solution  $\tilde{\mathbf{x}}$  is equivalent to the problem of

$$\text{Find } \tilde{\mathbf{x}} \in \mathcal{R}(\Phi), \text{ such that } (\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}) \perp \mathcal{R}(\Psi). \quad (2.6)$$

When  $\Psi = \Phi$ , this projection scheme is called Galerkin projection. For  $\Psi = \mathbf{A}\Phi$ , it is referred to as least-squares projection [28].

## 2.2 Different Choices of $\Psi$

Different choices of the projection operator,  $\Psi$ , have different implications. The different choices can be viewed as minimization of the error subject to different norms, or measures. This approach is similar to the analysis in He [27] in the context of trajectory piecewise linearization (TPWL).

We start by transforming the linear problem  $\mathbf{Ax} = \mathbf{b}$  into a minimization problem, namely,

$$\text{minimize } \|\Phi\mathbf{z} - \hat{\mathbf{x}}\|_{\mathbf{S}}^2 \quad (\text{over } \mathbf{z}), \quad (2.7)$$

where  $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$  is the true solution. Here, the norm is given by the inner product  $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{S}} = \mathbf{x}^T \mathbf{S} \mathbf{y}$ , and  $\mathbf{S}$  is a symmetric positive definite (SPD) matrix. By expanding the least-squares objective function and taking the derivative, we can write the solution as

$$\mathbf{z} = (\Phi^T \mathbf{S} \Phi)^{-1} \Phi^T \mathbf{S} \mathbf{A}^{-1} \mathbf{b}. \quad (2.8)$$

Comparing (2.8) with (2.4), we can see that the two solutions are equivalent if we take  $\Psi = (\mathbf{A}^T)^{-1} \mathbf{S} \Phi$ . Now, we examine different  $\Psi$  operators by choosing different  $\mathbf{S}$ .

Perhaps the most natural choice is to take  $\mathbf{S} = \mathbf{I}$ ; hence,  $\Psi = (\mathbf{A}^T)^{-1} \Phi$ . This leads to an “ideal” scheme that minimizes the Euclidean distance between the approximate solution  $\Phi\mathbf{z}$  and the true solution  $\mathbf{A}^{-1}\mathbf{b}$ , which corresponds to minimizing the error in the 2-norm space. In this case, the reduced solution is  $\mathbf{z} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{A}^{-1} \mathbf{b}$ . This formulation is not helpful, since one must calculate the true result  $\mathbf{A}^{-1}\mathbf{b}$  before obtaining the approximate result  $\mathbf{z}$ . However, this can be used in the context of TPWL, in which the projection operators  $\Psi = (\mathbf{A}^T)^{-1} \Phi$  are computed and stored during a preparatory offline computation period (also called the “overhead”), and one

reuses the saved data during the actual ROM simulation. He [27] showed that the use of  $\mathbf{S} = \mathbf{I}$  leads to TPWL models that are more expensive to construct but display higher degrees of accuracy than other choices for  $\mathbf{S}$ .

The second choice is to take  $\mathbf{S} = \mathbf{A}^T \mathbf{A}$ , so  $\Psi = \mathbf{A}\Phi$ . This is called least-square projection (LSP), or Petrov–Galerkin projection [27, 30]. The objective function in the minimization becomes

$$\|\Phi\mathbf{z} - \hat{\mathbf{x}}\|_{\mathbf{A}^T \mathbf{A}}^2 = \|\mathbf{A}\Phi\mathbf{z} - \mathbf{b}\|_2^2, \quad (2.9)$$

which is identical to the minimization of the residual in the 2-norm space. The performance of this method depends strongly on the condition number of  $\mathbf{A}$  [28]. In LSP, the expression for the approximate solution is:  $\tilde{\mathbf{x}} = \Phi(\Phi^T \mathbf{A}^T \mathbf{A} \Phi)^{-1} \Phi^T \mathbf{A}^T \mathbf{b}$ .

In the third choice, we assume  $\mathbf{S} = \mathbf{A}$ , and we obtain the Galerkin projection method with  $\Psi = \Phi$ . Here, we need to assume that  $\mathbf{A}$  is SPD, since numerical stability is not guaranteed if it is not (see [31]). For our purposes, Jacobian matrices based on the pressure equation in reservoir simulation are usually nearly-SPD matrices [32].

Galerkin projection lies somewhere between  $\mathbf{S} = \mathbf{I}$  and  $\mathbf{S} = \mathbf{A}^T \mathbf{A}$  (LSP scheme), and it gives better accuracy than LSP [19]. In Galerkin projection, we are minimizing  $\|\Phi\mathbf{z} - \hat{\mathbf{x}}\|_{\mathbf{A}}^2$ , or equivalently, we are minimizing  $\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$ . Therefore, we can draw a parallel between these projection schemes and iterative methods for linear systems. LSP is similar to the steepest descent method, as it is designed to drive the residual norm  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$  to zero; Galerkin projection is similar to the conjugate gradient method, as it is designed to minimize  $\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$ . It is well known that the conjugate gradient method performs better than the steepest descent method [28]. From now on, we take the Galerkin projection procedure as our default strategy.

Our goal is to investigate the POD method as a preconditioner for the linear

system arising from the pressure equation. A preconditioner is a matrix  $\mathbf{M}^{-1}$  (usually written in its inverse form) that makes the linear system  $\mathbf{Ax} = \mathbf{b}$  easier to solve; hence, we solve  $\mathbf{AM}^{-1}(\mathbf{Mx}) = \mathbf{b}$ , or  $\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$  instead of the original system. The matrix  $\mathbf{M}$  is designed to approximate  $\mathbf{A}$ , so the product  $\mathbf{AM}^{-1}$ , or  $\mathbf{M}^{-1}\mathbf{A}$ , has a lower condition number (closer to unity). Computing  $\mathbf{M}^{-1}\mathbf{v}$  should have much lower cost for any vector  $\mathbf{v}$  than computing  $\mathbf{A}^{-1}\mathbf{v}$  to ensure the efficiency of the preconditioner. Refer to [13] for more details.

We write the POD-based approximate solution as:

$$\tilde{\mathbf{x}} = \mathbf{M}_{POD}^{-1}\mathbf{b}, \quad \text{where}$$

$$\mathbf{M}_{POD}^{-1} = \Phi(\Phi^T \mathbf{A} \Phi)^{-1} \Phi^T \quad (\text{Galerkin as default}) \quad (2.10)$$

$$\mathbf{M}_{POD,LS}^{-1} = \Phi(\Phi^T \mathbf{A}^T \mathbf{A} \Phi)^{-1} \Phi^T \mathbf{A}^T \quad (\text{LSP}), \quad (2.11)$$

where the subscripts  $POD$  and  $POD, LS$  denote POD with Galerkin projection and POD with least-square projection, respectively. Note that, for the POD-ROM method,  $(\Phi^T \mathbf{A} \Phi)^{-1}$  and  $(\Phi^T \mathbf{A}^T \mathbf{A} \Phi)^{-1}$  are of much smaller dimension than  $\mathbf{A}$ . Since the matrix-matrix multiplication only gives a matrix of equal or lower rank, the rank of the preconditioning operator  $\mathbf{M}^{-1}$  is less than, or equal to,  $\ell$ .

## 2.3 Choosing $\Psi$ for Coarse-Scale Mass Balance

The framework of the Multiscale Finite-Volume Method (MsFVM), as described by Zhou and Tchelepi [25], involves a restriction operator  $\mathcal{R}$  and a prolongation operator  $\mathcal{P}$ , which serve as bridges between the coarse-scale and fine-scale systems. With the help of the two operators, the fine-scale linear system  $\mathbf{Ax} = \mathbf{b}$  is transformed into a coarse-scale problem  $\mathcal{R}\mathbf{A}\mathcal{P}\mathbf{x} = \mathcal{R}\mathbf{b}$ . The so-called “finite-volume” type restriction

operator  $\mathcal{R}$  transforms the overdetermined system  $\mathbf{A}\mathcal{P}\mathbf{x} = \mathbf{b}$  into the reduced coarse-scale system by grouping and adding the fine-scale equations contained in every coarse cell. This ensures that the coarse-scale system honors mass conservation, which makes it possible to reconstruct a locally conservative fine-scale velocity field [25].

Along this line, by taking  $\Psi$  equal to the finite-volume restriction operator,  $\mathcal{R}$ , in the POD–ROM method, it is possible to reconstruct a mass conservative velocity field and solve the transport equations, even when the tolerance for the pressure solution is relatively loose. This opens up more flexibility in designing more efficient reservoir simulators. However, this is not investigated further in this thesis.

## 2.4 Adaptive POD Solution Strategy

In this section, we discuss some numerical features of POD that may be helpful in designing an adaptive POD solution strategy, in which the reservoir simulator can adjust the POD basis to improve the computational efficiency.

### 2.4.1 Quality of POD basis

Generally, the true solution,  $\hat{\mathbf{x}}$ , will not lie exactly in the POD subspace  $\mathcal{R}(\Phi)$ . Therefore, it is useful to evaluate the quality of the basis,  $\Phi$ , in terms of its ability to yield a solution to the linear problem  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . For this purpose, one can examine the distance between the true solution,  $\hat{\mathbf{x}}$ , and the POD space  $\mathcal{R}(\Phi)$ . This can be done through an orthogonal operator,  $\mathbf{P}$ , which projects the vector  $\hat{\mathbf{x}}$  onto  $\mathcal{R}(\Phi)$ . That is,

$$\begin{aligned}\hat{\mathbf{x}} &= \tilde{\mathbf{x}} + \mathbf{x}' = \mathbf{P}\hat{\mathbf{x}} + (\mathbf{I} - \mathbf{P})\hat{\mathbf{x}}, \text{ where} \\ \mathbf{P}\hat{\mathbf{x}} &\in \mathcal{R}(\Phi) \text{ and} \\ (\mathbf{I} - \mathbf{P})\hat{\mathbf{x}} &\in \mathcal{R}(\Phi)^\perp.\end{aligned}\tag{2.12}$$

The superscript  $\perp$  indicates the orthogonal complement. Therefore, the quality of the basis,  $\Phi$ , with respect to  $\mathbf{Ax} = \mathbf{b}$ , can be evaluated through the normalized indicator  $\frac{\|(\mathbf{I}-\mathbf{P})\hat{\mathbf{x}}\|}{\|\hat{\mathbf{x}}\|}$ .

Using the generalized inner product with SPD matrix,  $\mathbf{S}$ , namely,

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{S}} = \mathbf{x}^T \mathbf{S} \mathbf{y}, \quad (2.13)$$

the projection operator can be expressed as  $\mathbf{P} = \Phi(\Phi^T \mathbf{S} \Phi)^{-1} \Phi^T \mathbf{S}$ , with the properties:  $\mathbf{P}^2 = \mathbf{P}$ , and  $\langle \mathbf{Px}, (\mathbf{I} - \mathbf{P})\mathbf{x} \rangle_{\mathbf{S}} = 0$  for any  $\mathbf{x}$ . We have tested several indicators, namely,

1.  $\mathbf{S} = \mathbf{A}^T \mathbf{A}$  (LSP); thus, the projector becomes  $\mathbf{P}_{POD,LS} = \Phi(\Phi^T \mathbf{A}^T \mathbf{A} \Phi)^{-1} \Phi^T \mathbf{A}^T \mathbf{A}$ . We can calculate the indicator by evaluating  $\frac{\|(\mathbf{I}-\mathbf{P}_{POD,LS})\hat{\mathbf{x}}\|_{\mathbf{A}^T \mathbf{A}}}{\|\hat{\mathbf{x}}\|_{\mathbf{A}^T \mathbf{A}}} = \frac{\|(\mathbf{I}-\mathbf{AM}_{POD,LS}^{-1})\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$ .
2.  $\mathbf{S} = \mathbf{A}$  (Galerkin). Assuming  $\mathbf{A}$  SPD,  $\mathbf{P} = \Phi(\Phi^T \mathbf{A} \Phi)^{-1} \Phi^T \mathbf{A}$ . It follows that we can evaluate the indicator as  $\frac{\|(\mathbf{I}-\mathbf{P}_{POD})\hat{\mathbf{x}}\|_{\mathbf{A}^T \mathbf{A}}}{\|\hat{\mathbf{x}}\|_{\mathbf{A}^T \mathbf{A}}} = \frac{\|(\mathbf{I}-\mathbf{AM}_{POD}^{-1})\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$ . Note, we also need to use the  $\mathbf{A}^T \mathbf{A}$ -norm instead of the  $\mathbf{A}$ -norm to avoid explicit use of  $\hat{\mathbf{x}}$ .

For either choice, one can evaluate the quality of  $\Phi$  with respect to the quality of the approximate solution of the linear problem  $\mathbf{Ax} = \mathbf{b}$ . This expression can be calculated with little extra computational cost, since we can utilize the intermediate results  $\mathbf{M}^{-1}\mathbf{b}$  and  $\mathbf{AM}^{-1}\mathbf{b}$  calculated by the iterative linear solver.

This quality indicator may be useful in designing an “adaptive POD” preconditioning strategy. If the value of the quality indicator is too large (say, greater than  $10^{-1}$ ), then we know that the true solution lies outside the POD subspace  $\mathcal{R}(\Phi)$ , and that the POD preconditioner will not be effective in reducing the residual norm. So, in this case, we can resort to some other method such as AMG to solve the system,

or we can recompute the POD basis  $\Phi$ . In this way, we can maximize the effectiveness of the POD-ROM preconditioner. However, the quality indicator may suffer from the fact that the  $\mathbf{A}$ -norm, and the  $\mathbf{A}^T\mathbf{A}$ -norm, magnify the effects of high-frequency components and downplay the importance of low-frequency components (see Section 3.2).

### 2.4.2 Contribution of Different Columns

In approximating the solution through  $\tilde{\mathbf{x}} = \Phi \mathbf{z}$ , different columns of  $\Phi$  (different proper orthogonal modes) contribute to the approximate solution  $\tilde{\mathbf{x}}$  by different amounts. One can examine the relative contributions of different columns of  $\Phi$  by looking at the components in the reduced vector  $\mathbf{z}$ . This is helpful when one has to choose which columns to retain in  $\Phi$ . For example, we might discard column  $i$ , if the component  $z_i$  is consistently close to zero. This technique can also be used to design an adaptive POD preconditioner.

### 2.4.3 Dynamic POD Subspace

Computational issues also arise in designing an adaptive POD preconditioning framework. When a minor modification to the snapshot collection,  $\mathbf{X}$ , is made, one has to recompute the POD basis and the inversion of the reduced matrix  $(\Phi^T \mathbf{A} \Phi)^{-1}$ . It is natural to search for a faster algorithm to perform these calculations. Hall et al. [33] offer a procedure to adjust the POD subspace incrementally when, for example, a new snapshot is added. Another investigation on updating the matrix factorizations can be found in [34]. This method had been used in POD-based reservoir simulation methods by [9]. Moreover, Orr [35] discussed the inversion of a matrix that has undergone an incremental change.

#### 2.4.4 Computational Cost

For each linear problem  $\mathbf{Ax} = \mathbf{b}$ , we must perform two matrix-matrix multiplications to construct  $\mathbf{A}_r (= \Phi^T \mathbf{A} \Phi)$ , and an LU decomposition for the reduced matrix  $\mathbf{A}_r (= \mathbf{LU})$ . The computational cost is summarized in Table. 2.1, where  $\xi$  is the number of nonzero elements in each row of matrix  $\mathbf{A}$ . For three-dimensional system with the usual 7-point stencil,  $\xi = 7$ . Considering the fact that  $n \gg \ell$ , the total cost for each linear problem is  $\frac{2}{3}\ell^3 + 2\ell n(\xi + \ell)$ .

Table 2.1: Computational cost of the POD preconditioner for each linear problem.

Operation	Cost
$\mathbf{C} \leftarrow \mathbf{A} \Phi$	$\ell n(2\xi - 1)$
$\mathbf{A}_r \leftarrow \Phi^T \mathbf{C}$	$\ell^2(2n - 1)$
LU Decomposition of $\mathbf{A}_r$	$2\ell^3/3$

For each linear iteration, the linear solver calls the preconditioner routine once. In each preconditioner call, the solver needs to project the right-hand-side to the lower dimension, then a back substitution is performed to give the reduced solution, and finally one needs to reconstruct the solution in the original high-dimensional space. The computational cost is summarized in Table. 2.2. The total cost for each preconditioner call is  $2\ell^2 + 4\ell n$ .

Table 2.2: Computational cost of the POD preconditioner for each preconditioner call.

Operation	Cost
$\mathbf{b}_r \leftarrow \Phi^T \mathbf{b}$ (Restriction)	$\ell(2n - 1)$
$\mathbf{z} \leftarrow \mathbf{A}_r^{-1} \mathbf{b}_r$ (Back substitution using $\mathbf{L}$ and $\mathbf{U}$ )	$2\ell^2$
$\tilde{\mathbf{x}} \leftarrow \Phi \mathbf{z}$ (Prolongation)	$n(2\ell - 1)$



# Chapter 3

## Spectral Analysis of the POD Preconditioner

In this chapter, we analyze the error reduction in terms of the spectrum of frequencies that make up the solution space. A framework for this spectral analysis is developed. Some results are shown regarding the approximation properties of high and low frequencies. Then, through a simple example, we discuss some interesting properties of the POD preconditioner.

### 3.1 Spectral Analysis

Analysis of the spectral properties of the target problem  $\mathbf{Ax} = \mathbf{b}$  and of the preconditioning operator  $\mathbf{M}^{-1}$  is critical to the ultimate determination of the best strategies. The key idea in spectral analysis is to view the problem under the eigenbasis of the matrix  $\mathbf{A}$ .

Let us assume that  $\mathbf{A}$  is derived from the discretization of the pressure equation in a reservoir simulation problem (near-elliptic PDE system), and that it is symmetric

positive definite (SPD). The orthogonal diagonalization can be written as

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \quad (3.1)$$

where  $\mathbf{D}$  is a diagonal matrix containing the eigenvalues  $(\lambda_1, \dots, \lambda_n)$  of  $\mathbf{A}$ . The columns of the matrix  $\mathbf{V}$  form a normalized orthogonal basis  $(\mathbf{v}_1, \dots, \mathbf{v}_n)$  that is called the eigenbasis. For vector  $\mathbf{b}$ , the components  $\tilde{b}_i$  ( $i = 1, \dots, n$ ) under the eigenbasis can be calculated by  $(\tilde{b}_1, \dots, \tilde{b}_n)^T = \mathbf{V}^T \mathbf{b}$ . Therefore, under this eigenbasis, the linear problem  $\mathbf{Ax} = \mathbf{b}$  can be solved by multiplication of coordinate  $i$  by  $\lambda_i^{-1}$ ,  $i = 1, \dots, n$ .

The idea of iterative solution strategies can also be understood in this way. For preconditioned Richardson iteration [28], where the entire solution process relies on the preconditioner  $\mathbf{M}^{-1}$ , the algorithm is:

$$\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}_k \quad (3.2)$$

$$\delta\mathbf{x}_{k+1} = \mathbf{M}^{-1}\mathbf{r}_k \approx \mathbf{A}^{-1}\mathbf{r}_k \quad (\text{approximately solve}) \quad (3.3)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}_{k+1}, \quad (3.4)$$

where  $\mathbf{r}_k$  is the residual, and  $\delta\mathbf{x}_{k+1}$  is the update. For each iteration, we are reducing the residual components  $\mathbf{V}^T \mathbf{r}_k$  until the norm  $\|\mathbf{r}_k\|$  is smaller than some threshold.

We can also view the effect of a preconditioner  $\mathbf{M}^{-1}$  in the eigenbasis. If  $\mathbf{M}^{-1}$  can approximate certain components of the true inverse, i.e.,  $\mathbf{M}^{-1}\mathbf{b}$  approximates  $\mathbf{A}^{-1}\mathbf{b}$  very well along some eigen-directions of  $\mathbf{A}$ , or  $\mathbf{v}_i^T \mathbf{M}^{-1} \mathbf{b} \approx \mathbf{v}_i^T \mathbf{A}^{-1} \mathbf{b}$  for some  $i$ , we can say that the preconditioner  $\mathbf{M}^{-1}$  approximates these eigenvectors of  $\mathbf{A}$  very well. Correspondingly, components in these directions  $(\mathbf{r} - \mathbf{AM}^{-1}\mathbf{r})$  will be reduced significantly after one Richardson iteration. In this case, we can say that those eigenvectors are “resolved” by the preconditioner  $\mathbf{M}^{-1}$ .

An ideal preconditioner should have a clear preference in terms of which eigenvectors it can resolve well, but it should also keep the other eigenvectors intact. Therefore, we can choose, or combine, different preconditioners according to our specific problem. However, in practice, preconditioners tend to “disturb” the components of different eigenvectors, i.e., the vectors in the subspace corresponding to some eigenvectors will jump far from this subspace after applying the preconditioner. If these disturbances between different groups of eigenvectors are significant, the preconditioner will not be effective.

An example of this spectral framework is given by Zhou and Tchelepi [25, 26], where they analyze the behavior of their Two-Stage Algebraic Multiscale (TAMS) method. They show that the first stage resolves low-frequency eigenvectors very well by using the AMS (Algebraic Multiscale) method, and the second stage resolves high-frequency errors using a local preconditioner, such as Block Incomplete LU decomposition (BILU) with zero fill-in.

Here, the term frequency means the matrix eigenvalue [28]. High-frequency modes correspond to modes of large eigenvalues of  $\mathbf{A}$ , while low-frequency modes correspond to eigenvectors of smaller eigenvalues of  $\mathbf{A}$ . This is the case for many linear systems, such as those derived from elliptic problems or wave problems.

## 3.2 High and Low Frequency Components

A fundamental difference exists between high-frequency components and low-frequency components. We illustrate this difference for the POD–ROM preconditioning strategy using a simple two-dimensional model. We will see that it is more difficult to resolve low-frequency eigenvectors compared with high-frequency eigenvectors. The argument is also valid for general Galerkin-type ROM schemes.

Consider Fig. 3.1, which describes  $\mathbf{Ax} = \mathbf{b}$  under an eigenbasis. Here,

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

The  $x$ -axis corresponds to the high-frequency direction (eigenvalue  $\lambda_1 = 2$ ), and the  $y$ -axis corresponds to the low-frequency direction (eigenvalue  $\lambda_2 = 0.5$ ). In the plot we show the true solution  $\hat{\mathbf{x}}$  and the RHS vector  $\mathbf{b} = \mathbf{A}\hat{\mathbf{x}}$ . The direction of  $\hat{\mathbf{x}}$  is associated with angle  $\theta$ .

To set up the projection, we assume that the POD subspace is one-dimensional with a basis direction of angle  $\alpha$ . In Fig. 3.1,  $\alpha = 37^\circ$ , and  $\theta = 58^\circ$ . From Section 2.4.1, we know that the projection operator is  $\mathbf{P} = \mathbf{M}^{-1}\mathbf{A}$ , projecting  $\hat{\mathbf{x}}$  onto the POD subspace  $\mathcal{R}(\Phi)$ . The projection is orthogonal in the  $\mathbf{A}$ -norm, i.e., the projection is along the  $\mathbf{A}$ -conjugate direction  $\mathcal{R}(\Phi)_{\mathbf{A}}^\perp$  (marked in the plot). The usual orthogonal complementary direction  $\mathcal{R}(\Phi)^\perp$  is also marked by a dashed line. One can see that the direction of  $\mathcal{R}(\Phi)_{\mathbf{A}}^\perp$  is oblique, leaning toward the low-frequency axis.

The right-hand-side is shown as  $\mathbf{b}$  in Fig. 3.1, where  $\mathbf{x}_{POD}$  is the approximate solution given by the POD method,  $\mathbf{x}'$  is the remaining error, and  $\mathbf{b}'$  is the residual after one step of POD. We can see that, for both  $\mathbf{b}'$  and  $\mathbf{x}'$ , the high-frequency part decreases faster than the low-frequency part.

There are significant differences between high-frequency and low-frequency components, which are associated with the different eigenvalues of  $\mathbf{A}$ . In evaluating the  $\mathbf{A}$ -norm of a vector, a high-frequency component contributes more than it would to a 2-norm, and a low-frequency component contributes less than it would to a 2-norm. Therefore, the POD solution strategy given by the projection operator  $\mathbf{P}$  eliminates high-frequency errors efficiently and leaves the low-frequency errors. The extent of this difference depends on the condition number of  $\mathbf{A}$ . While the condition number is 4 in this example, in real problems it can easily go up to  $10^4$ . So we would expect

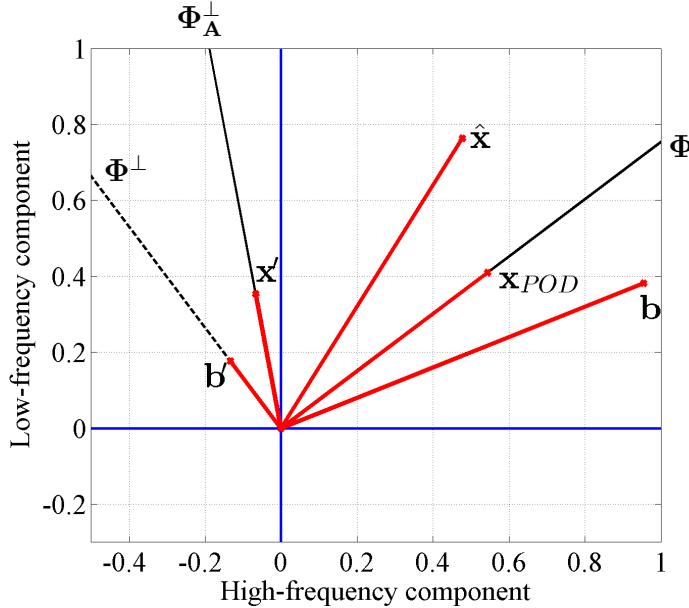


Figure 3.1: Simple 2D analysis on the POD method, for  $\alpha = 37^\circ$  and  $\theta = 58^\circ$ . Eigenvalues of  $\mathbf{A}$ :  $\lambda_1 = 2, \lambda_2 = 0.5$ .

that the low-frequency errors are crucial in designing a POD method, as is the case for many iterative solution strategies [28].

It is interesting to note that the POD method sometimes increases the residual norm for the first iteration rather than decreasing it. An example is shown in Fig. 3.2, where  $\|b'\| > \|b\|$ . For our two-dimensional case, this happens when the true solution  $\hat{\mathbf{x}}$  contains predominantly low-frequency components.

### 3.3 Effect of the POD Space

Using the example above, we can examine the low-frequency approximation ability of the POD method. In Fig. 3.3, we plot the error norm reduction ( $\frac{\|\mathbf{x}'\|}{\|\hat{\mathbf{x}}\|}$ , red curve) and the residual norm reduction ( $\frac{\|b'\|}{\|b\|}$ , blue curve), versus  $\theta$ , the direction of  $\hat{\mathbf{x}}$ . Here, the matrix

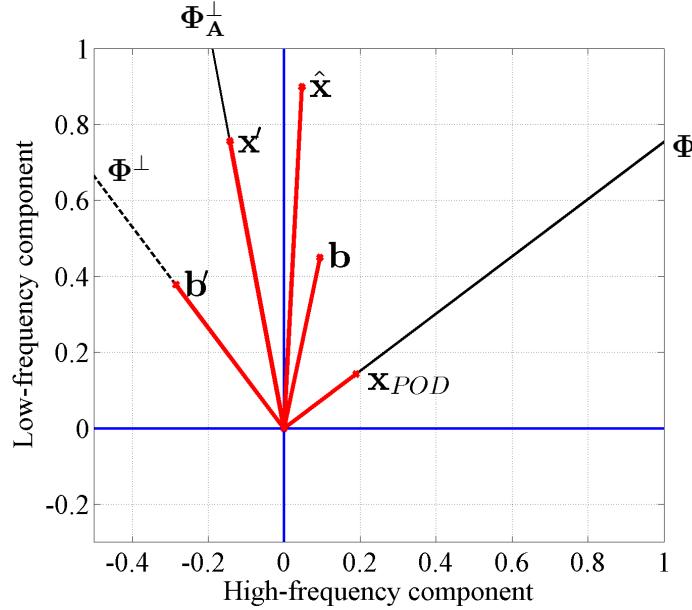


Figure 3.2: Illustration of  $\|\mathbf{b}'\| > \|\mathbf{b}\|$ , for  $\alpha = 37^\circ$  and  $\theta = 87^\circ$ . Eigenvalues of  $\mathbf{A}$ :  $\lambda_1 = 2, \lambda_2 = 0.5$ .

$$\mathbf{A} = \begin{bmatrix} 10 & 0 \\ 0 & 0.1 \end{bmatrix},$$

which has a condition number of 100 is used. One can see that both curves go to zero when  $\theta = \alpha$ . This corresponds to the situation when the true solution lies exactly in the POD subspace, and thus the POD method gives the true solution. In practice, we would like to use the POD method for the case where  $\hat{\mathbf{x}}$  does not lie in  $\mathcal{R}(\Phi)$  exactly. This corresponds to the nonzero value of the curves.

The upper plot in Fig. 3.3 shows the case where  $\alpha = 90^\circ$ , whereby the  $\mathcal{R}(\Phi)$  subspace captures the low-frequency component exactly. We see that the norm reduction of both the error and residual is less than one, and the error is reduced considerably for a large range of possible  $\theta$ . However, the lower plot of Fig. 3.3 ( $\alpha = 85^\circ$ ) shows a case where the POD subspace  $\mathcal{R}(\Phi)$  is somewhat different from the exact low-frequency direction. In this case, both curves are not bounded by unity and the residual curve

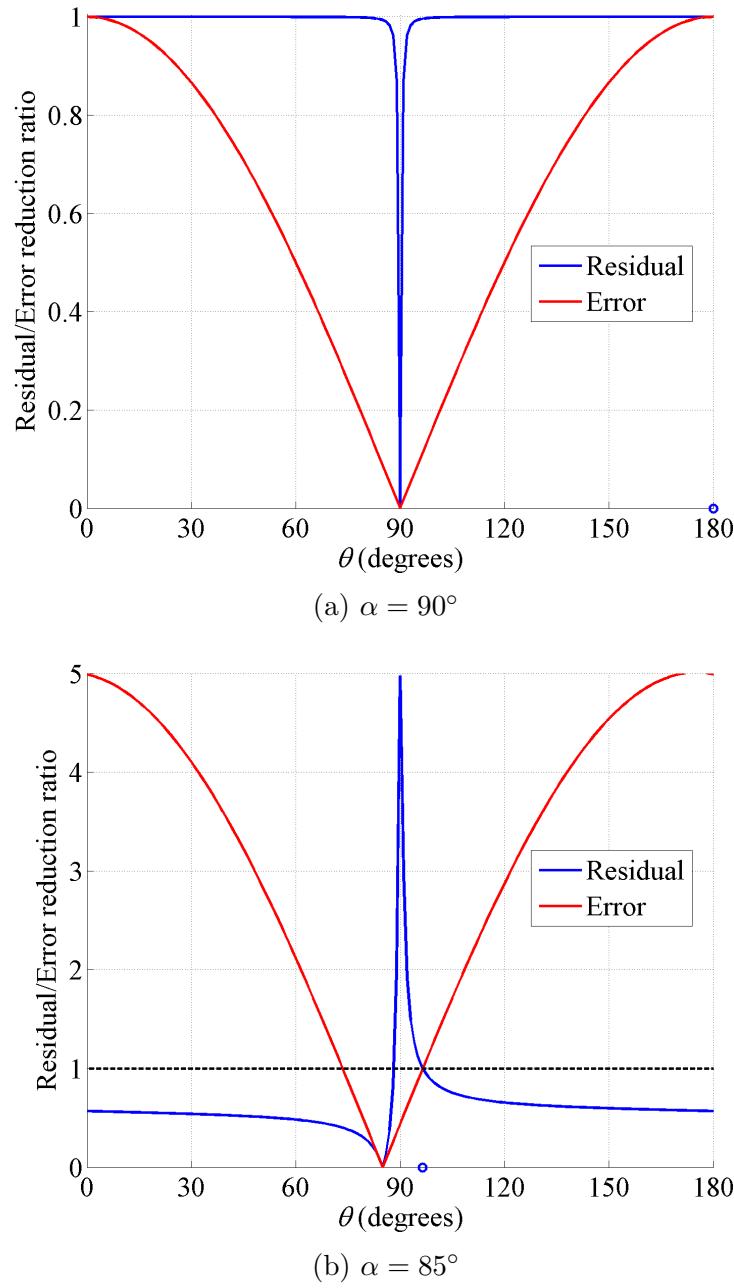


Figure 3.3: Error/residual reduction ratio versus  $\theta$  for the POD method for  $\alpha = 90^\circ$  and  $\alpha = 85^\circ$ . The blue circle shows the direction of remaining  $\mathbf{x}'$  for the case where  $\hat{\mathbf{x}}$  is in the direction  $\theta = 135^\circ$ . Eigenvalues of  $\mathbf{A}$ :  $\lambda_1 = 10$ ,  $\lambda_2 = 0.1$ .

shows a sharp peak. The resulting residual-norm reduction may not be significant in this case, since the POD subspace  $\mathcal{R}(\Phi)$  does not lie exactly along the low-frequency eigenvector.

For POD preconditioning ( $\mathbf{M}^{-1} = \Phi(\Phi^T \mathbf{A} \Phi)^{-1} \Phi^T$ ) to work effectively, the POD subspace  $\mathcal{R}(\Phi)$  should lie exactly along the low-frequency directions of matrix  $\mathbf{A}$ . In other words, we want the basis  $\Phi$  to capture the low-frequency eigenvectors accurately. However, this is not the case for the POD method. The snapshots of POD come from one or more training simulations, and they are not generated with any particular focus on the low-frequency part of the spectrum. Hence, one may expect that the POD basis is deficient in this sense. To improve things, we may want to design some methods to enable the POD basis to capture the low-frequency part of the spectrum.

### 3.4 Iteration Matrix Spectrum

From the Richardson iteration (3.2)-(3.4), we have the following relation:

$$\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})\mathbf{r}_k. \quad (3.5)$$

The iteration matrix  $\mathbf{G}$  is defined as  $\mathbf{I} - \mathbf{A}\mathbf{M}^{-1}$ . The spectrum of the iteration matrix is another important tool to assess the performance of a preconditioner. That the spectral radius  $\rho(\mathbf{G})$  is strictly less than one is a necessary and sufficient criterion for the iterations to converge:

$$\rho(\mathbf{G}) < 1 \iff \forall \mathbf{r}_0, \lim_{k \rightarrow \infty} \mathbf{r}_k = 0. \quad (3.6)$$

The convergence rate is defined as  $\mu = \lim_{k \rightarrow \infty} \frac{\|\mathbf{r}_{k+1}\|}{\|\mathbf{r}_k\|}$ , which is bounded by  $\rho(\mathbf{G})$ .

This can also be understood in terms of the eigenbasis of  $\mathbf{A}$ . The preconditioner

$\mathbf{M}^{-1}$  should be a close approximation of the true inverse of  $\mathbf{A}$  for certain frequencies. Hence, ideally  $\mathbf{M}^{-1}$  is diagonal in the eigenbasis. Thus, the eigenvalues of the iteration matrix  $\mathbf{G}$  are the shrinking ratios of the residual vector components under the eigenbasis. The smaller the values are, the faster the residual components decrease. Moreover, the preconditioned generalized minimal residual method (GMRES) yields fast convergence for the situation where the eigenvalues of  $\mathbf{AM}^{-1}$  are clustered together [36].

If we only use the POD method as the preconditioner, the iteration matrix becomes:

$$\mathbf{G} = \mathbf{I} - \mathbf{A}\Phi(\Phi^T\mathbf{A}\Phi)^{-1}\Phi^T. \quad (3.7)$$

We compare this with the projection operator  $\mathbf{P}$  discussed in Section 2.4.1. We find that  $\mathbf{P} = \mathbf{M}_{POD}^{-1}\mathbf{A}$  and  $\mathbf{I} - \mathbf{P}$  are the POD projectors acting on  $\mathbf{x}$  in the solution space, and  $\mathbf{AM}_{POD}^{-1}$  and  $\mathbf{G} = \mathbf{I} - \mathbf{AM}_{POD}^{-1}$  are the corresponding projection operators acting on  $\mathbf{b}$  in the residual space. We know that projection matrices are idempotent, i.e.,  $\mathbf{P}^2 = \mathbf{P}$ , and the eigenvalues of an idempotent matrix can only be 0 or 1 [37]. Moreover, since  $(\Phi^T\mathbf{A}\Phi)$  is a nonsingular matrix of order  $\ell$ , we know that the rank of the idempotent matrix  $\mathbf{I} - \mathbf{G} = \mathbf{AM}_{POD}^{-1}$  is  $\ell$ . Hence,  $\mathbf{I} - \mathbf{G}$  has eigenvalue 0 of multiplicity  $n - \ell$ , and eigenvalue 1 of multiplicity  $\ell$ . It follows that the iteration matrix  $\mathbf{G}$  has eigenvalue 0 of multiplicity  $\ell$ , and eigenvalue 1 of multiplicity  $n - \ell$ , so its spectral radius is 1.

For the POD-preconditioned Richardson iteration, since the iteration matrix  $\mathbf{G}$  is idempotent, the residuals will have the relation:

$$\mathbf{r}_k = \mathbf{A}^k \mathbf{r}_0 = \mathbf{A} \mathbf{r}_0 = \mathbf{r}_1, \quad (3.8)$$

which means that the residual will stop changing after the first Richardson iteration. Hence, we cannot use the POD method as the only preconditioner. This property

also characterizes the AMS method, as discussed by Zhou and Tchelepi [25]. It is however possible to combine POD with other methods to have an effective multistage preconditioner, as will be discussed through numerical examples in the next chapter.

# Chapter 4

## Numerical Experiments

In this chapter, we demonstrate the POD pressure preconditioning using numerical results of reservoir simulation problems. Several observations are made based on our results.

### 4.1 Governing Equations

We now present the governing equations for an oil/water two-phase reservoir simulation problem. Combining mass conservation equations and Darcy's law, we have:

$$\frac{\partial(\phi S_o)}{\partial t} = \nabla \cdot \left( \frac{kk_{ro}}{\mu_o} \nabla p_o \right) - \tilde{q}_o \quad (4.1)$$

$$\frac{\partial(\phi S_w)}{\partial t} = \nabla \cdot \left( \frac{kk_{rw}}{\mu_w} \nabla p_w \right) - \tilde{q}_w, \quad (4.2)$$

where subscript  $o$  designates oil and  $w$  water,  $S$  is phase saturation,  $\phi$  is porosity,  $k$  is permeability,  $k_{ro}$  and  $k_{rw}$  are oil and water relative permeability,  $p_o$  and  $p_w$  are phase pressures,  $\mu_o$  and  $\mu_w$  are phase viscosities, and  $t$  is time. The variables  $\tilde{q}_o$  and  $\tilde{q}_w$  are source terms, with units of surface volume rate per reservoir volume. The fluids are incompressible. Gravitational effects are neglected.

We solve this problem using an Implicit Pressure Explicit Saturation (IMPES) scheme [38, 39] in order to focus on the performance of our POD preconditioner for the pressure equation. In an IMPES scheme, (4.1) and (4.2) can be transformed into a pressure equation and a saturation equation, namely,

$$c_r \phi \frac{\partial p}{\partial t} = \nabla \cdot (k \lambda_t \nabla p) - \tilde{q}_o - \tilde{q}_w \quad (4.3)$$

$$\phi \frac{\partial S_w}{\partial t} + S_w \phi c_r \frac{\partial p}{\partial t} = \nabla \cdot (k \lambda_w \nabla p) - \tilde{q}_w. \quad (4.4)$$

Here,  $\lambda_o = k_{ro}/\mu_o$ ,  $\lambda_w = k_{rw}/\mu_w$ , and  $\lambda_t = \lambda_o + \lambda_w$  are mobilities calculated according to the upstream rule [12]. Note that  $c_r = \frac{1}{\phi} \frac{\partial \phi}{\partial p}$  is the rock compressibility. Using oil field units, the pressure equation for grid block  $i$  in discrete form is given by:

$$0 = R_i = \sum_j T_{ij} (\lambda_t)_{ij} (p_j^{n+1} - p_i^{n+1}) - c_r V \phi^n \frac{p_i^{n+1} - p_i^n}{5.615 \Delta t} - q_{o,i} - q_{w,i}, \quad (4.5)$$

where  $R_i$  is the residual we want to drive to zero,  $T_{ij} = 0.001127 \frac{Ak}{\Delta x}$  is the geometric part of the transmissibility linking blocks  $i$  and  $j$ , including the conversion factor, and the sum is over all blocks neighboring block  $i$ . Here  $(\lambda_t)_{ij}$  is calculated from the upstream block between blocks  $i$  and  $j$ , and  $V$  is the bulk volume for each block. We use the porosity from the previous time step, which is equivalent to the approximation  $(\phi_i^{n+1} - \phi_i^n) \approx \phi_i^n c_r (p_i^{n+1} - p_i^n)$ . In this way, the pressure system is linear. So, we do not need to use Newton's method for this solution. After we have computed the pressure, the saturation is determined by an explicit update:

$$\begin{aligned} & \frac{V}{5.615} \left( \phi_i^{n+1} \frac{S_{w,i}^{n+1} - S_{w,i}^n}{\Delta t} + S_{w,i}^n \phi^n c_r \frac{p_i^{n+1} - p_i^n}{\Delta t} \right) \\ &= \sum_j T_{ij} (\lambda_w)_{ij} (p_j^{n+1} - p_i^{n+1}) - q_{w,i}. \end{aligned} \quad (4.6)$$

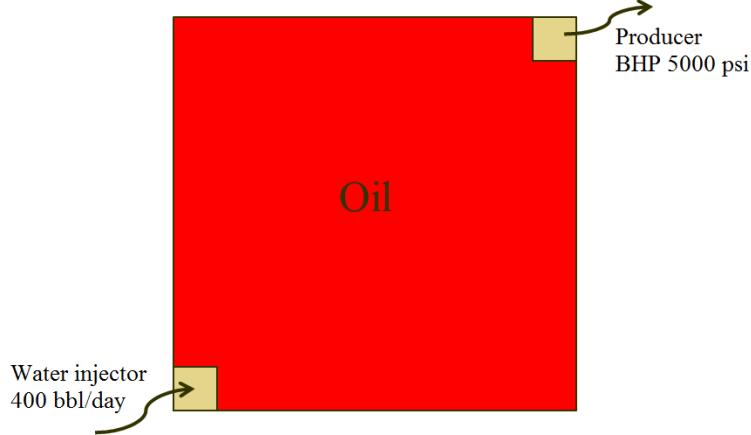


Figure 4.1: Reservoir model used for preconditioner investigation.

Moreover, the source terms  $q_o$  and  $q_w$  are expressed as

$$q_{o,i} = \text{WI}(\lambda_o)_i(p_i^{n+1} - p_{well}) \quad (4.7)$$

$$q_{w,i} = \text{WI}(\lambda_o)_i(p_i^{n+1} - p_{well}), \quad (4.8)$$

where WI is the usual Peaceman type well index [40], and  $p_{well}$  is the wellbore pressure in block  $i$ .

## 4.2 Problem Specification and Solution

We consider a horizontal two-dimensional reservoir initially saturated with oil. The reservoir region has a square shape and is discretized using  $10 \times 10$  blocks. Each block has dimensions  $50 \text{ ft} \times 50 \text{ ft} \times 25 \text{ ft}$ . One water injector and one producer are placed diagonally at two opposite corners. Water is injected at a fixed rate of 400 bbl/day, and the producer operates with a fixed bottom-hole pressure (BHP) of 5000 psi. The reservoir configuration is shown in Fig. 4.1.

We assume zero fluid compressibility, and no capillarity. We also assume that the oil and water viscosities are equal (1 cp). We set the rock compressibility to

$10^{-5}$  psi $^{-1}$ . The initial reservoir pressure is 5800 psi. For our base case, we use quadratic relative permeability functions, i.e.,  $k_{rw} = S_w^2$ ,  $k_{ro} = S_o^2$ . The permeability and porosity fields are homogeneous with values of 1000 md and 0.2.

The simulation period is 1000 days. We use the same time step sizes used by Stanford’s General Purpose Research Simulator (GPRS) [12–15]. The first 90-day period is used as the “training simulation.” The whole procedure is implemented using MATLAB.

For the POD basis, we form the snapshot collection,  $\mathbf{X}$ , by collecting all pressure changes,  $\delta\mathbf{p}$ , obtained during the first 90 days (i.e., training period), one snapshot per time step, and use  $\mathbf{X}$  to construct the POD basis. Each snapshot is normalized before performing SVD, so the POD algorithm minimizes the relative error (see Appendix A.3). This treatment is motivated from the fact that the aim of the linear solver is to reduce the residual by a certain factor, instead of by an absolute value. The POD basis  $\Phi$  contains the first  $\ell$  left singular vectors,  $\mathbf{u}_1, \dots, \mathbf{u}_\ell$ , that result from the SVD of  $\mathbf{X}$ .

We employ the Richardson iteration scheme preconditioned by the POD-ROM method (see (2.10)). We therefore have a two-stage composite preconditioner  $\mathbf{M}_{comp}^{-1}$ , which consists of the POD preconditioner  $\mathbf{M}_{POD}^{-1} = \Phi(\Phi^T \mathbf{A} \Phi)^{-1} \Phi^T$ , and an ILU(0) preconditioner  $\mathbf{M}_{ILU}^{-1} = \mathbf{U}^{-1} \mathbf{L}^{-1}$ , where  $\mathbf{LU} \approx \mathbf{A}$  is the incomplete LU decomposition. The two-stage operator can be written as [13]:  $\mathbf{M}_{comp}^{-1} = \mathbf{M}_{ILU}^{-1} (\mathbf{I} - \mathbf{A} \mathbf{M}_{POD}^{-1}) + \mathbf{M}_{POD}^{-1}$ . Although  $\mathbf{M}_{POD}^{-1}$  is singular,  $\mathbf{M}_{comp}^{-1}$  is not. The single-stage ILU(0) preconditioner is used as a reference. The tolerance for the linear solver is  $10^{-5}$ , i.e.,  $\frac{\|\mathbf{Ax}-\mathbf{b}\|}{\|\mathbf{b}\|} < 10^{-5}$  upon convergence.

The left four plots in Fig. 4.2 show the water-saturation maps for several time steps. The injector is located at the left corner, and the producer is located at the right corner. The right four plots in Fig. 4.2 are maps of the pressure change,  $\delta\mathbf{p}$ , at the same time steps. The pressure change is obtained by solving a linear system

of pressure equations for each time step. There are 221 time steps for the entire 1000-day simulation, and 39 steps for the 90-day training period. The water front sweeps about one third of the reservoir by the end of the training simulation. The water breakthrough occurs at about 400 days.

The POD basis uses information from the 90-day training period, and the objective is to see how well the POD basis can approximate the pressure field evolution for the entire 1000-day simulation. This means we are trying to “extrapolate” using the POD method. We do not have a priori knowledge of how far the solutions lie outside the POD subspace  $\mathcal{R}(\Phi)$  and how well the POD method will perform.

A total of 39 snapshots are used to construct the POD basis, so we can have a maximum of 39 columns, or proper orthogonal modes, in  $\Phi$ . The energy plot (Fig. 4.3) of POD shows the importance of each column in the basis  $\Phi$ . Here, the term “energy” refers to the squared singular values shown in the figure [4]. We see that 20 columns can represent the snapshots up to a tolerance of  $10^{-4}$ – $10^{-5}$  (refer to Appendix A.3 for error estimation). We explore the performance by taking the number of columns,  $\ell$ , as 5, 10, 15, 20, and we also use all the snapshots, i.e.,  $\ell = 39$ , for comparison.

A plot of the eigenvalues (from low to high) of the Jacobian matrix for Step 15 is shown in Fig. 4.4. Other time steps show very similar patterns. The condition numbers of Jacobian matrices for different steps are plotted in Fig. 4.5.

### 4.3 Convergence Behavior

Several convergence history plots are shown in Figs. 4.6-4.8. From the plots, we can see the improved performance of POD+ILU versus ILU-only. Moreover, including more columns in the POD basis (larger  $\ell$ ) leads to faster convergence.

In particular, for the first 39 steps, which correspond to the training simulation, the problems converge in very few iterations when  $\ell = 20$ , which is consistent with

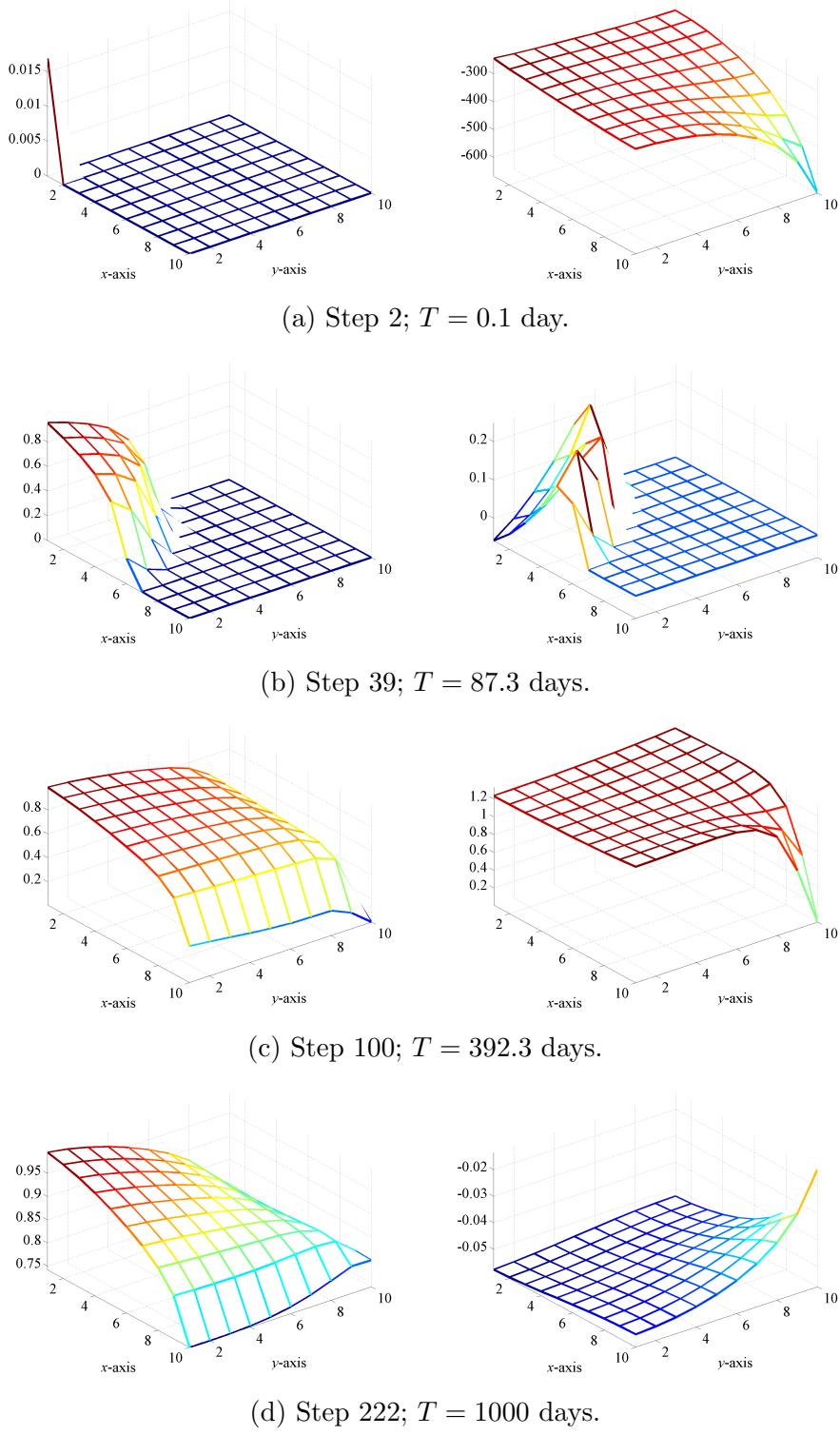


Figure 4.2: Solutions for the base case: water saturation  $S_w$  (left) and pressure change  $\delta \mathbf{p}$  (right).

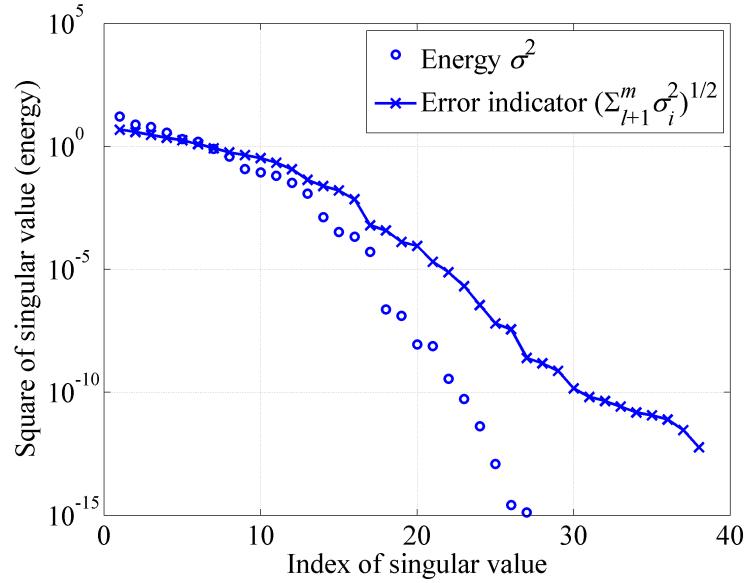


Figure 4.3: Energy plot for the POD basis showing the importance and contribution of different singular vectors. The error indicator gives us an idea of the reconstruction error, see (A.9).

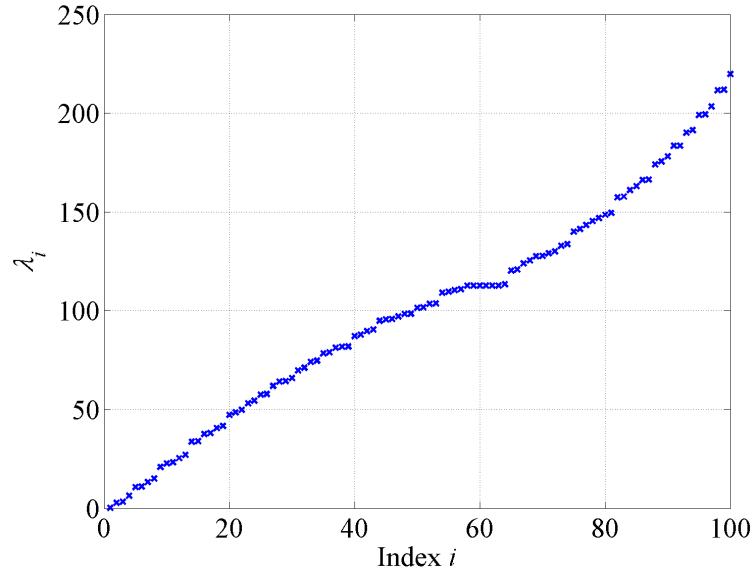


Figure 4.4: Eigenvalues of the Jacobian matrix for Step 15.

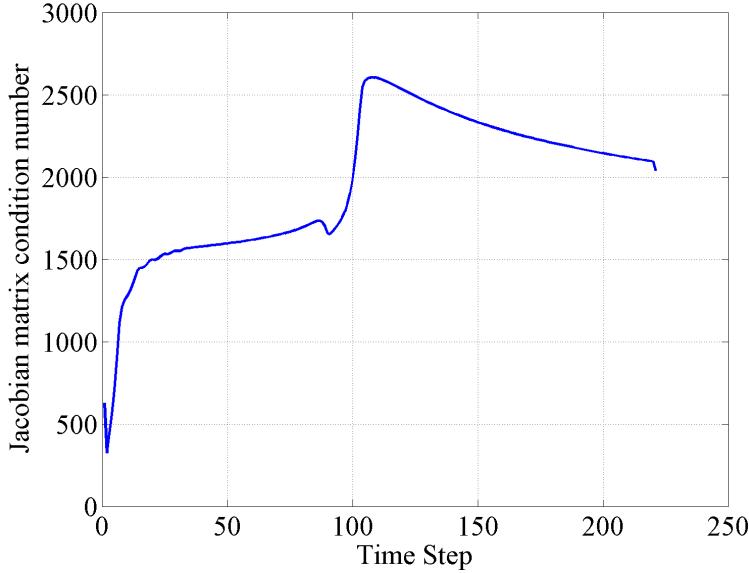


Figure 4.5: Jacobian matrix condition numbers.

the energy plot (Fig. 4.3). However, for problems that are outside the training period, more iterations are needed, even for  $\ell = 39$ . This is because the Jacobian matrices and right-hand-sides for the later period have not been sampled by the training simulation. Hence, they are likely to include components or features that are not captured by the POD basis  $\Phi$ .

## 4.4 Iteration Matrix Spectrum

As discussed in Section 3.4, it is important to examine the eigenvalues of the iteration matrix  $\mathbf{I} - \mathbf{AM}_{comp}^{-1}$  to assess the quality of the two-stage preconditioner  $\mathbf{M}_{comp}^{-1}$ . If the largest eigenvalue (spectral radius) is less than unity, convergence is guaranteed. In addition, the spectral radius usually suggests the convergence rate of a preconditioner.

The iteration matrix spectrum for Step 15 is plotted in Fig. 4.9. Note that, the iteration matrix  $\mathbf{G} = \mathbf{I} - \mathbf{AM}^{-1}$  is not SPD and can have negative eigenvalues. The figure indicates that the eigenvalues are scattered for the ILU-only case, and

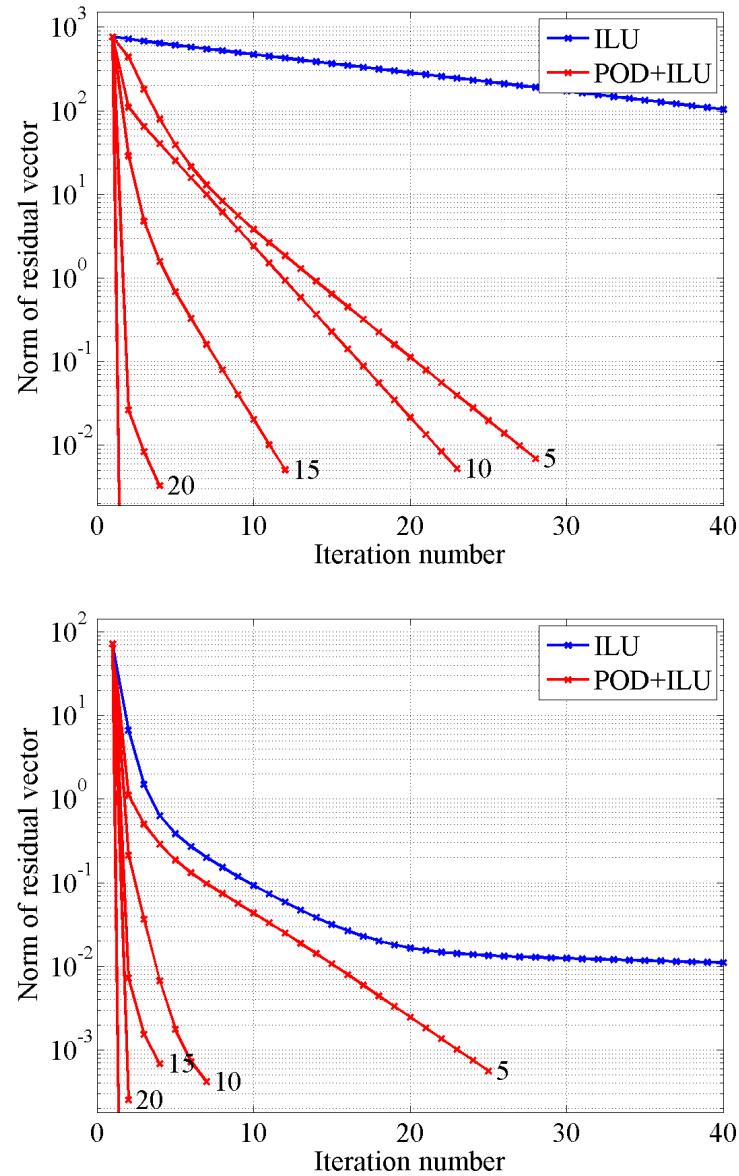


Figure 4.6: Convergence history: Step 2 (top) and 15 (bottom). The red curves in each plot correspond to  $\ell = 5, 10, 15, 20, 39$ . The blue curve is for the ILU preconditioner.

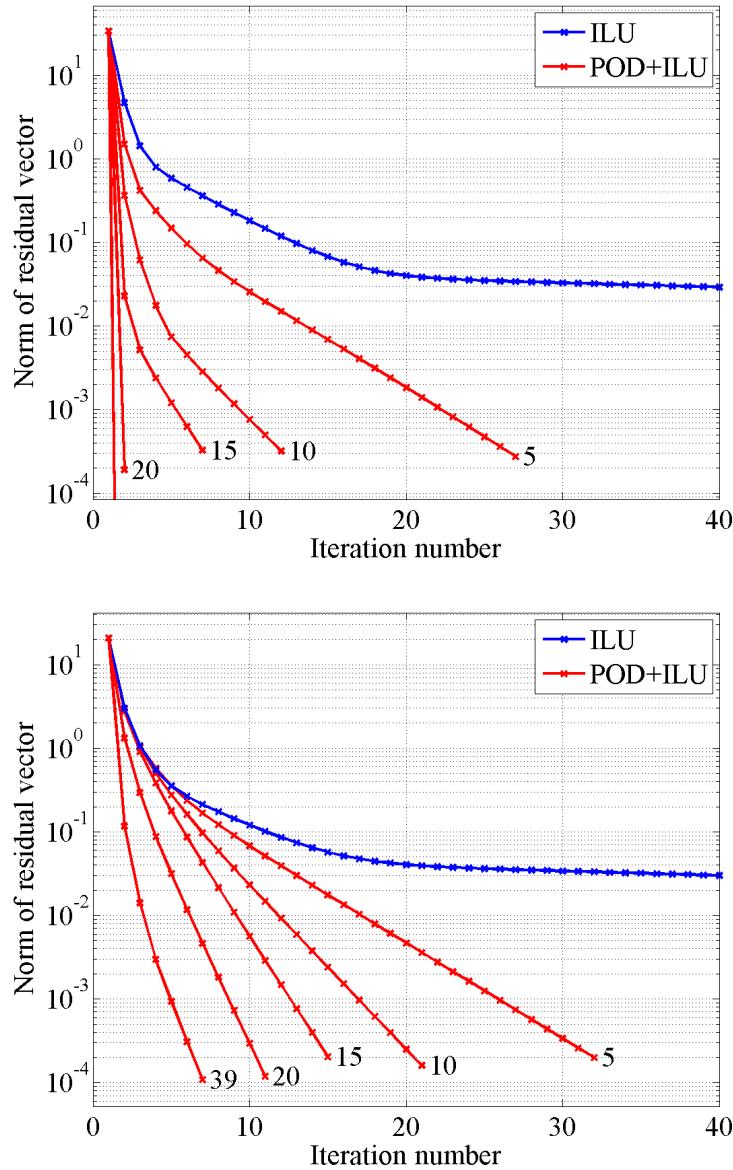


Figure 4.7: Convergence history: Step 35 (top) and 45 (bottom). The red curves in each plot correspond to  $\ell = 5, 10, 15, 20, 39$ . The blue curve is for ILU-only preconditioner.

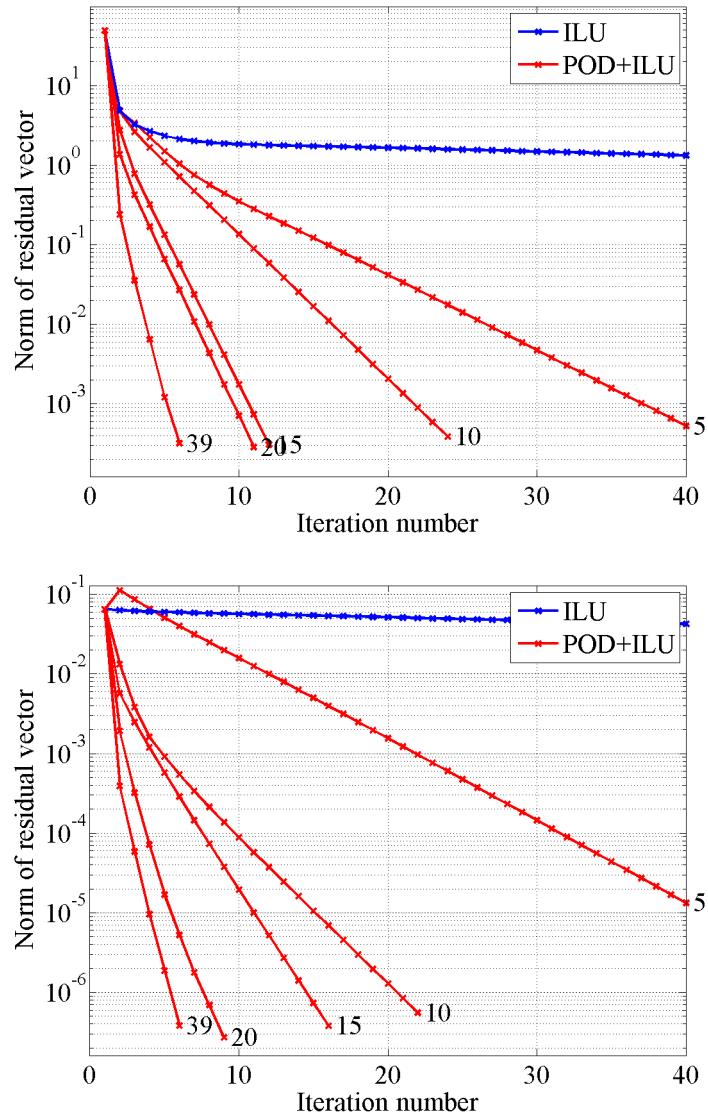


Figure 4.8: Convergence history: Step 100 (top) and 221 (bottom). The red curves in each plot correspond to  $\ell = 5, 10, 15, 20, 39$ . The blue curve is for ILU-only preconditioner. Note that POD+ILU with  $\ell = 5$  increases the residual norm for the first iteration.

the maximum eigenvalue is quite close to one. For POD+ILU, as we increase  $\ell$ , the spectrum changes in a specific way. Namely, although the overall patterns are similar, a few eigenvalues shift toward the origin. In particular, the spectral radius remains the same from  $\ell = 5$  up to  $\ell = 32$ , although it becomes smaller for  $\ell \geq 33$ . Thus, increasing the number of columns appears to have a relatively small (and inconsistent) impact on the spectral radius.

Usually, the convergence rate is equal to the iteration matrix spectral radius. Fig. 4.10 shows the convergence history, together with the convergence trend calculated by the spectral radius (dashed line). It is worth noticing that while the actual convergence rate increases as  $\ell$  increases, the spectral radius stays about the same. Therefore, for the POD+ILU preconditioner, the convergence rate seems not to follow the trend given by the spectral radius of the iteration matrix.

## 4.5 Effect of POD on Error Spectrum

It is important to know how the POD and ILU stages resolve the error spectrum. To study this issue, we now use the eigenvectors of the Jacobian matrix as the right-hand-sides to see how the POD and ILU preconditioners resolve the error in terms of frequency. The eigenvectors are ordered based on their frequency from low to high.

**Analysis of the ILU Preconditioner** We first examine the performance of the ILU-only preconditioner. By taking a normalized eigenvector of the Jacobian matrix as the RHS, and performing one Richardson iteration with the ILU preconditioner, we can assess the behavior of the ILU preconditioner by examining the residual norm after one iteration. The result for Step 15 is shown in Fig. 4.11. One can see that the ILU preconditioner has a clear preference for resolving high-frequency modes rather than low-frequency modes. This is in accordance with our knowledge that ILU is

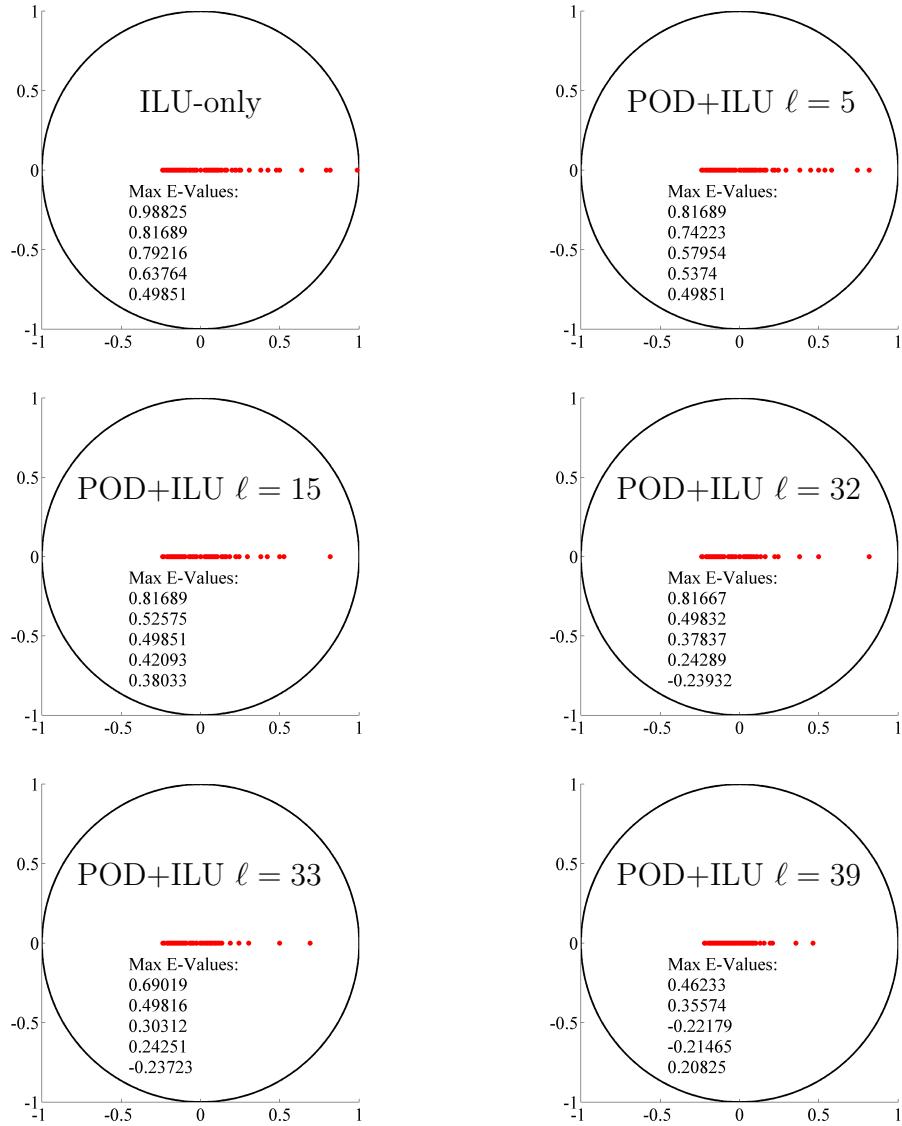


Figure 4.9: Plots of eigenvalues of iteration matrix for Step 15. ILU-only (top left), POD+ILU with  $\ell = 5$  (top right),  $\ell = 15$  (middle left),  $\ell = 32$  (middle right),  $\ell = 33$  (bottom left), and  $\ell = 39$  (bottom right). The largest five (magnitude) eigenvalues are listed in each plot.

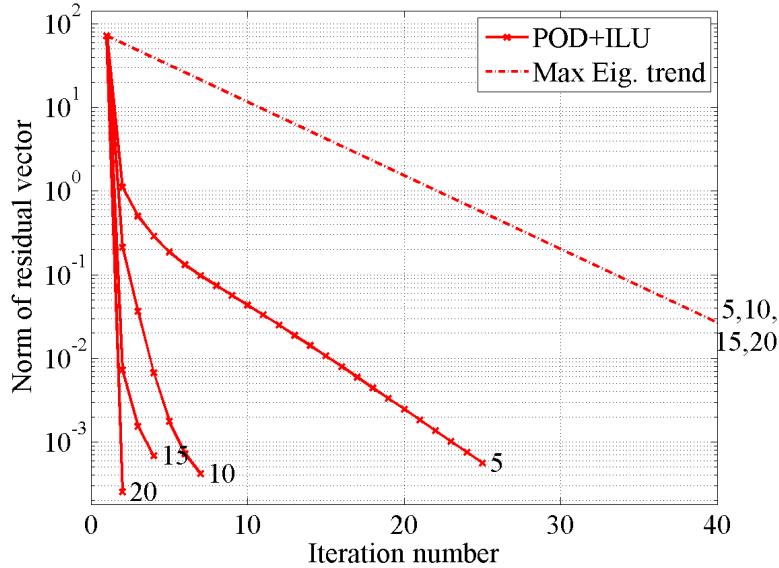


Figure 4.10: Convergence history with the trend given by spectral radius for Step 15,  $\ell = 5, 10, 15, 20$ . The dashed line shows the convergence trend given by the spectral radius.

effective at eliminating high-frequency errors [26, 41].

However, the ILU preconditioner preserves the frequency spectrum of the original vector only weakly. For example, we take the 50th eigenvector of  $\mathbf{A}$  as the RHS, which gives the residual vector  $\mathbf{r}$  after one ILU iteration. The frequency spectrum of the residual ( $= \mathbf{V}^T \mathbf{r}$  where  $\mathbf{V}$  are eigenvectors) is shown in the upper plot of Fig. 4.12. One can see that the residual vector still has large portions of the 50th frequency mode. By contrast, if we take the 78th eigenvector as the RHS, the residual components are distributed away from the 78th frequency, though they are still mainly in the high-frequency half of the spectrum. In a test that takes all 100 eigenvectors of  $\mathbf{A}$  as the right-hand-sides, 63 of the 100 cases show that the residual still has the same largest component as the original RHS. Hence, we can say that the ILU preconditioner preserves the error frequency weakly.

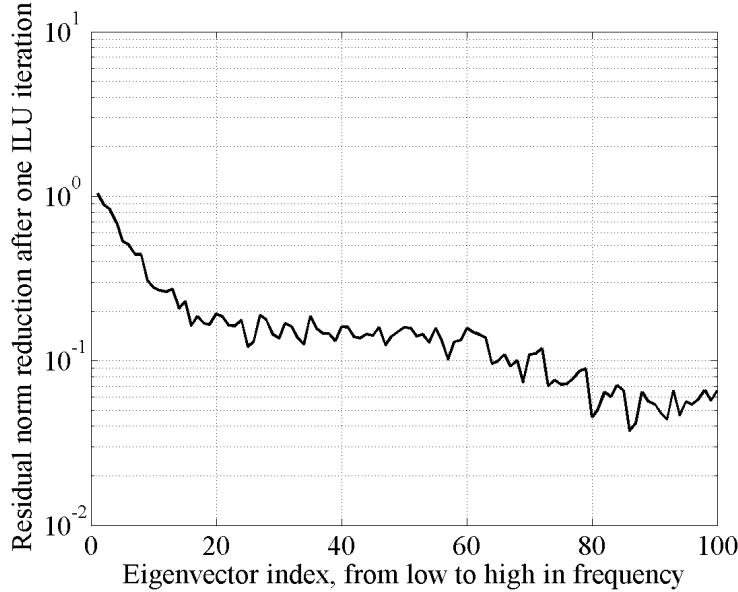


Figure 4.11: Eigen-analysis by taking eigenvectors of  $\mathbf{A}$  as RHS for Step 15, using ILU-only.

**Analysis of the POD Preconditioner** Following the same procedure, we examine the POD preconditioner. The residual norm reduction behavior for different eigenvectors of  $\mathbf{A}$  is shown in Fig. 4.13.

In Fig. 4.13, the dotted line is the reference line of value one. The colored curves are the norm reduction curves for different  $\ell$  values. From the figure, several observations can be made. First, the curve for a larger value of  $\ell$  is lower than that for a smaller  $\ell$ , which means that increasing  $\ell$  leads to better norm reduction. This is most apparent in the lower plot. Second, we notice from the upper plot that the residual norms can even be greater than one for some low frequencies. This phenomenon was noted in Section 3.2.

An important feature of the results in Fig. 4.13 is that the curves coincide at a value of one for many frequencies in the spectrum, and these frequencies are evenly distributed among the different parts of the spectrum. This indicates that POD method has little or no effect on the eigenvectors of many frequencies. This sheds

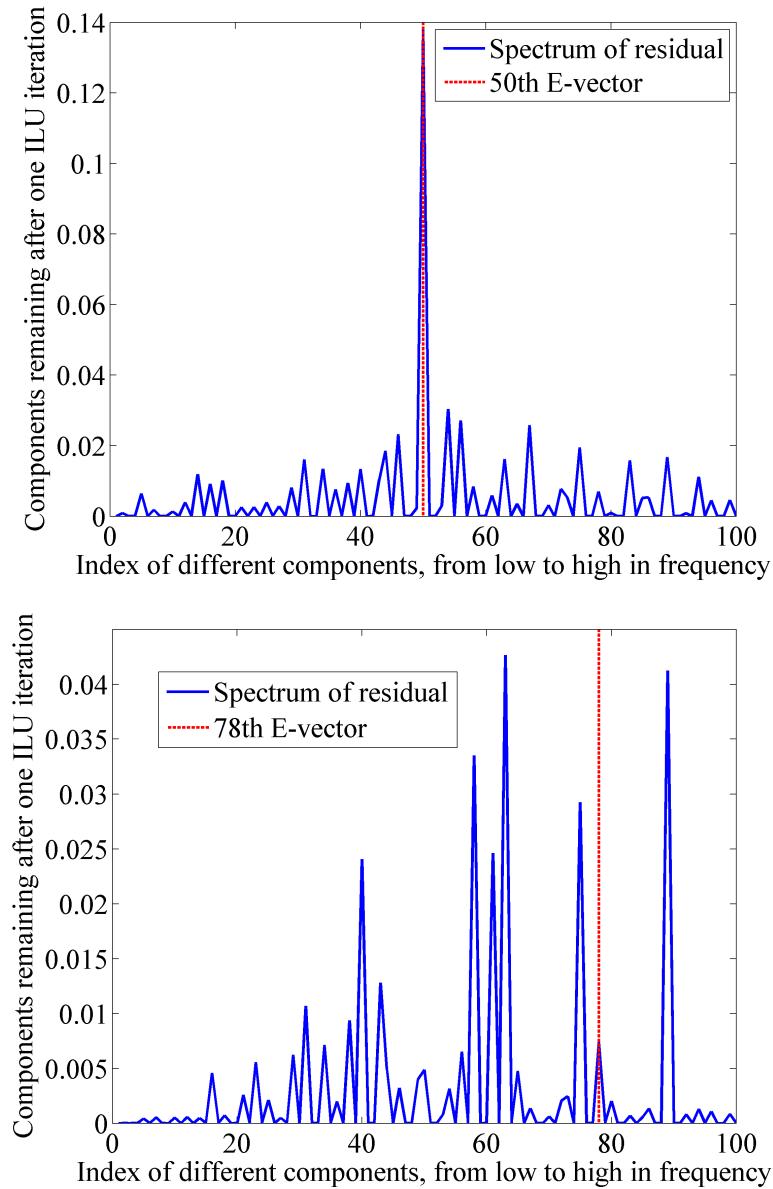


Figure 4.12: Eigen-analysis by taking one eigenvector of  $\mathbf{A}$  as the RHS: Eigenvector number 50 (upper) and eigenvector number 78 (lower), Step 15, using ILU-only.

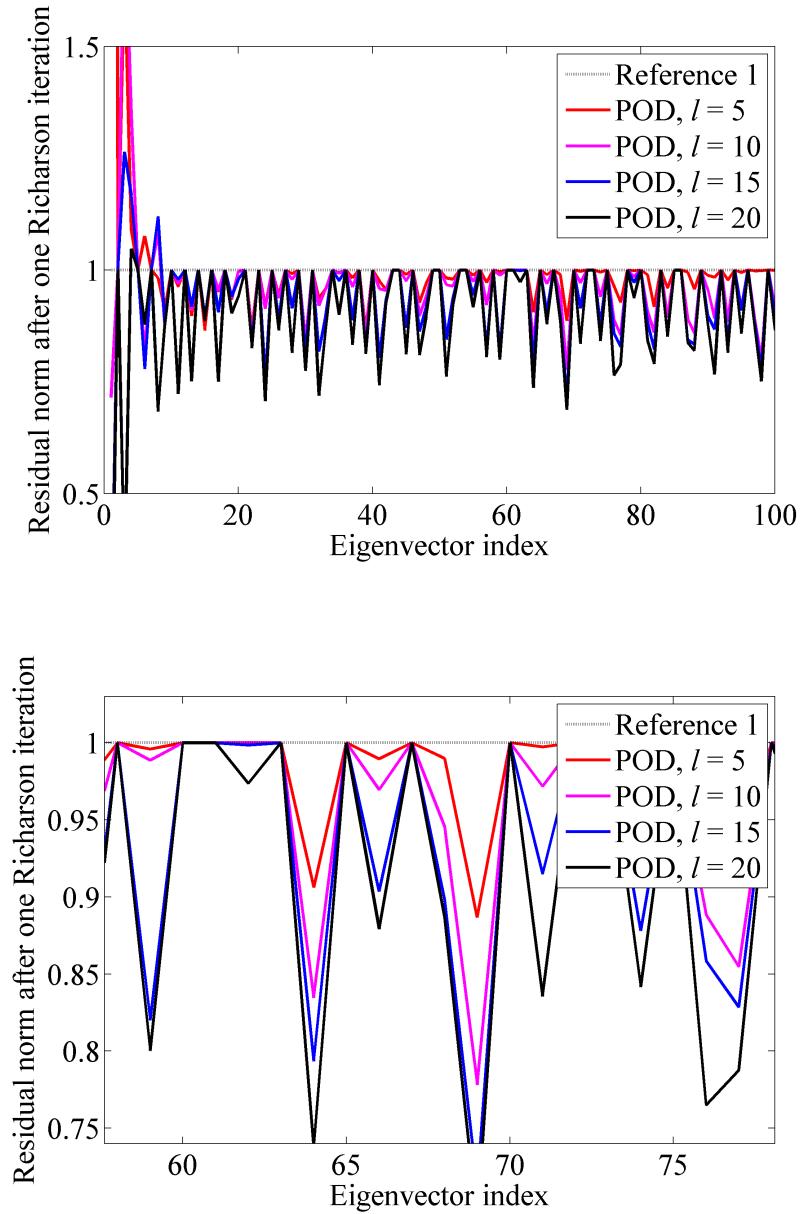


Figure 4.13: Eigen-analysis by taking eigenvectors of  $\mathbf{A}$  as RHS: Step 15, POD-only,  $\ell = 5, 10, 15, 20$ . Full range (top) and local detail (bottom). Note that the POD has no effect on many frequency modes.

additional light on the observed behavior of the POD preconditioner. The POD strategy is RHS dependent.

In addition, we find that the frequency preserving ability of POD is directly correlated with its norm reduction ability. For example, if we take the 3rd eigenvector, which the POD basis ( $\ell = 20$ ) approximates very well according to Fig. 4.13, as the RHS, the residual has significant components in other frequencies, including some high-frequencies (Fig. 4.14, upper plot). In contrast, if we take the 64th eigenvector, which the POD basis ( $\ell = 10$ ) approximates rather poorly, as the RHS, the residual components of other frequencies are relatively small (Fig. 4.14, lower plot). Generally, greater norm reduction ability implies poorer frequency preserving ability, and vice versa. Hence, larger  $\ell$  leads to poorer frequency preserving ability because of the greater norm reduction of the POD preconditioner.

## 4.6 RHS Dependence of the POD Preconditioner

To clearly show the RHS-dependent nature of the POD preconditioner, we randomly generate a RHS vector, and then assess how the POD+ILU preconditioner behaves. The convergence history is plotted in Fig. 4.15. It seems that the convergence rate for a random RHS (black) is slower than that of the problem RHS (red). In addition, the convergence rate of the random RHS coincides with the rate given by the spectral radius of the iteration matrix (red dashed lines). The convergence rate of the random RHS does not improve as  $\ell$  increases from 10 to 20. This indicates that the POD preconditioner is not a preconditioner that works equally well for any RHS.

To further understand the nature of the POD preconditioner, recall that the spectral radius of the iteration matrix does not change for  $\ell = 5$  up to 32. Analysis shows that if we include the 33rd column in  $\Phi$ , by for example taking the first 10 columns and the 33rd column as the POD basis, we obtain an iteration matrix that has a

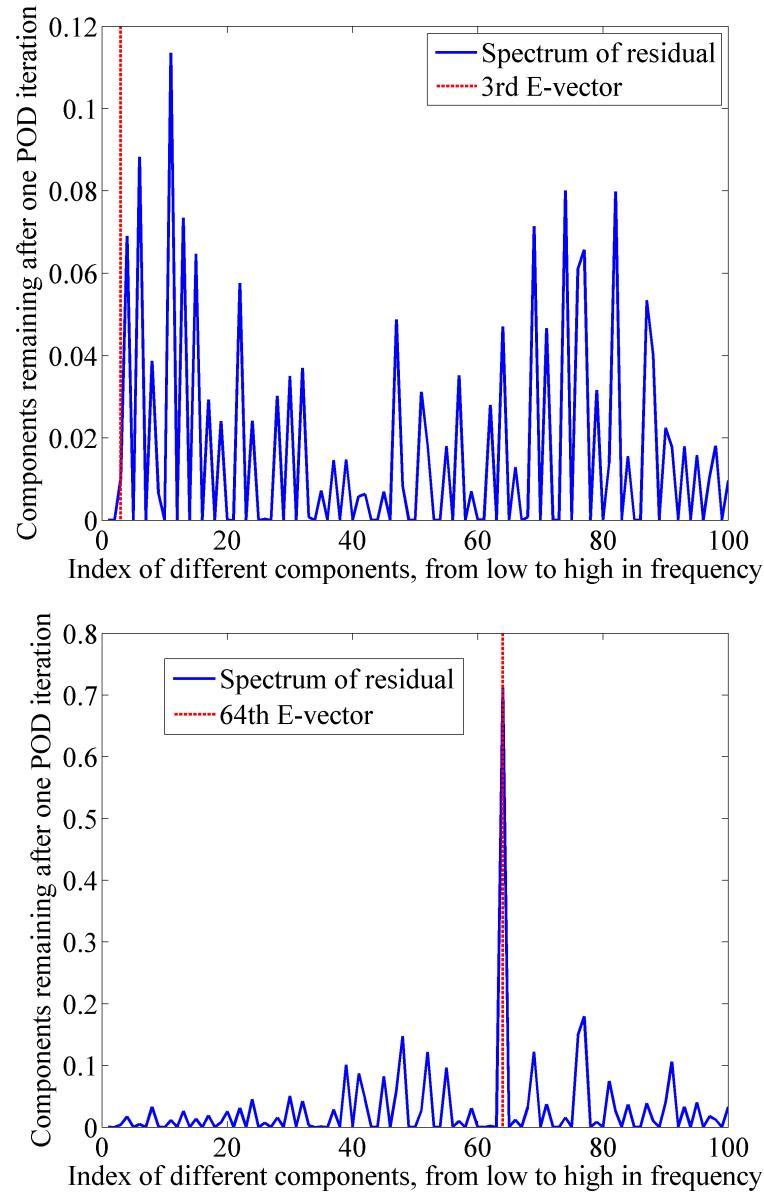


Figure 4.14: Eigen-analysis by taking one eigenvector of  $\mathbf{A}$  as the RHS: Eigenvector number 3 with  $\ell = 20$  (upper) and eigenvector number 64 with  $\ell = 10$  (lower), Step 15, POD preconditioner.

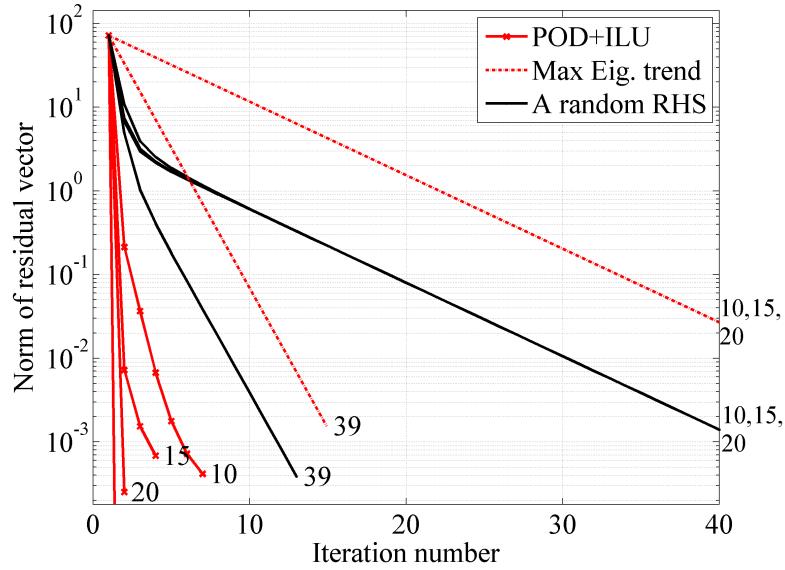


Figure 4.15: Convergence history with additional random RHS: Step 15,  $\ell = 10, 15, 20, 39$ . Randomly generated RHS shown in black. RHS encountered in the problem shown in red. The red dashed lines are convergence trends given by the spectral radii of iteration matrices.

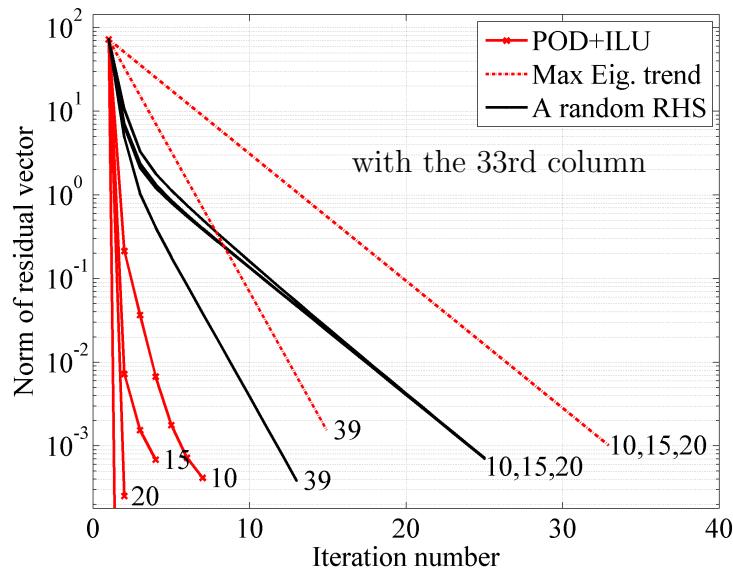


Figure 4.16: Convergence history with random RHS, where  $\Phi$  additionally includes the 33rd column.

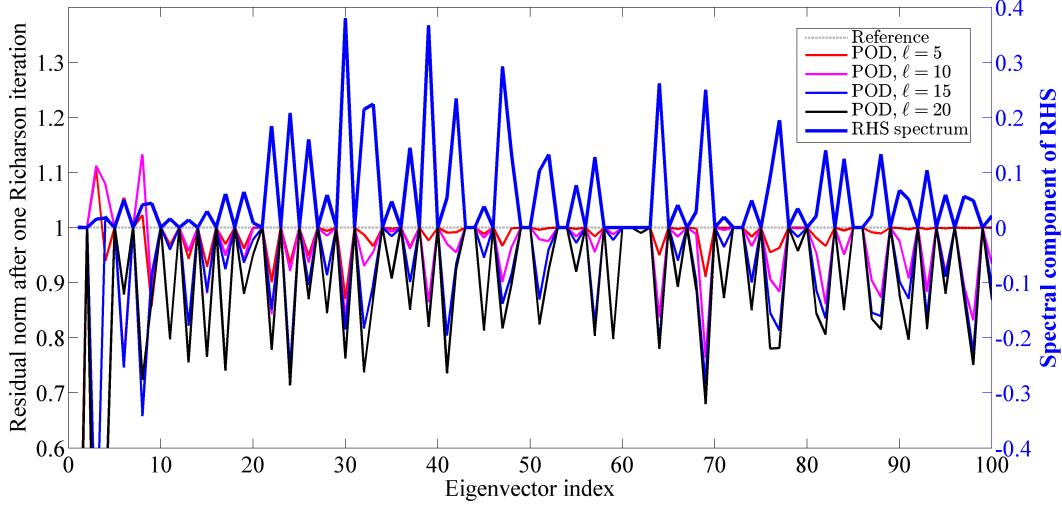


Figure 4.17: Spectral comparison between POD effectiveness and RHS components. Residual norm reduction after one iteration (thin curves). Normalized RHS spectrum (blue thick curve). Results for Step 15.

smaller spectral radius than the one without the 33rd column. We now use this new  $\Phi$  (with the 33rd column) and reassess the convergence behavior, which is shown in Fig. 4.16. A comparison between Fig. 4.15 and Fig. 4.16 shows that by including the 33rd column, the spectral radius of the iteration matrix becomes smaller, and the convergence for the random RHS (black curve) becomes faster. However, the convergence history for the RHS actually encountered in the problem displays no change at all (red curves).

Therefore, we can conclude that the POD preconditioner is RHS-dependent. It is designed to work for a specific RHS that may be encountered in the problem. Actually, if we check the frequency spectrum of the RHS vectors encountered in the training simulation, we find that they contain none, or only a small portion, of the frequencies that POD has little or no effect on, as shown in Fig. 4.17. This explains why POD is effective when the RHS is similar to that encountered in the training simulation.

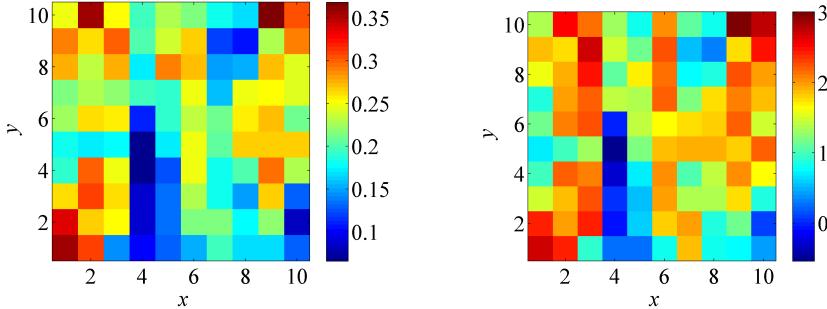


Figure 4.18: Porosity (left) and  $\log_{10}$ -permeability (right) distribution for the heterogeneous case.

## 4.7 Additional Examples

Two additional examples, which reiterate the detailed observations above, are now presented.

### 4.7.1 Heterogeneous Case ( $10 \times 10$ )

To further demonstrate the performance of the POD preconditioner, we now examine a more complex case. We change the permeability and porosity field from homogeneous to heterogeneous. The distributions are shown in Fig. 4.18. Moreover, we change the oil viscosity to 10 cp, while keeping the water viscosity as 1 cp. Thus, we have an unfavorable water/oil displacement scenario. Other model parameters are the same as in the base case. Saturation maps from two time steps are shown in Fig. 4.19.

Several iteration matrix eigenvalue distribution plots for Step 70 are shown in Fig. 4.20. We can see that, as  $\ell$  increases from 10 to 25, the spectral radius only decreases by a small amount; however for  $\ell = 30$ , the spectral radius shrinks significantly.

The convergence history for this time step is shown in Fig. 4.21. The figure shows

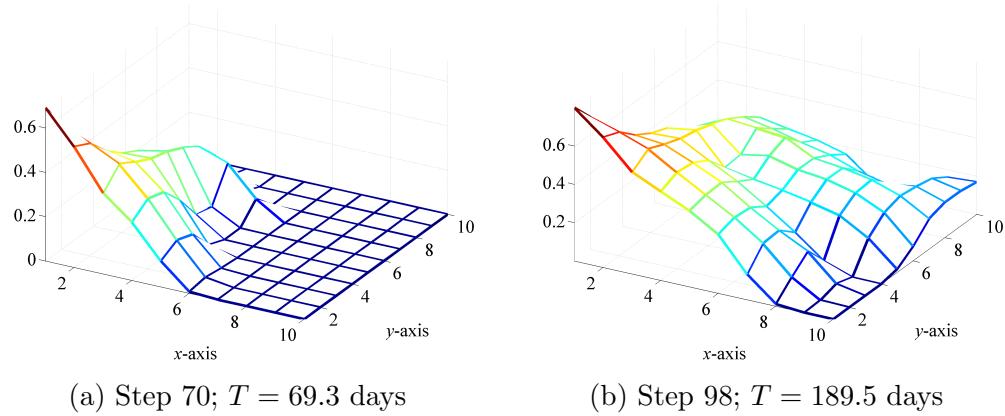


Figure 4.19: Saturation maps for the heterogeneous case, before (left) and after (right) water breakthrough.

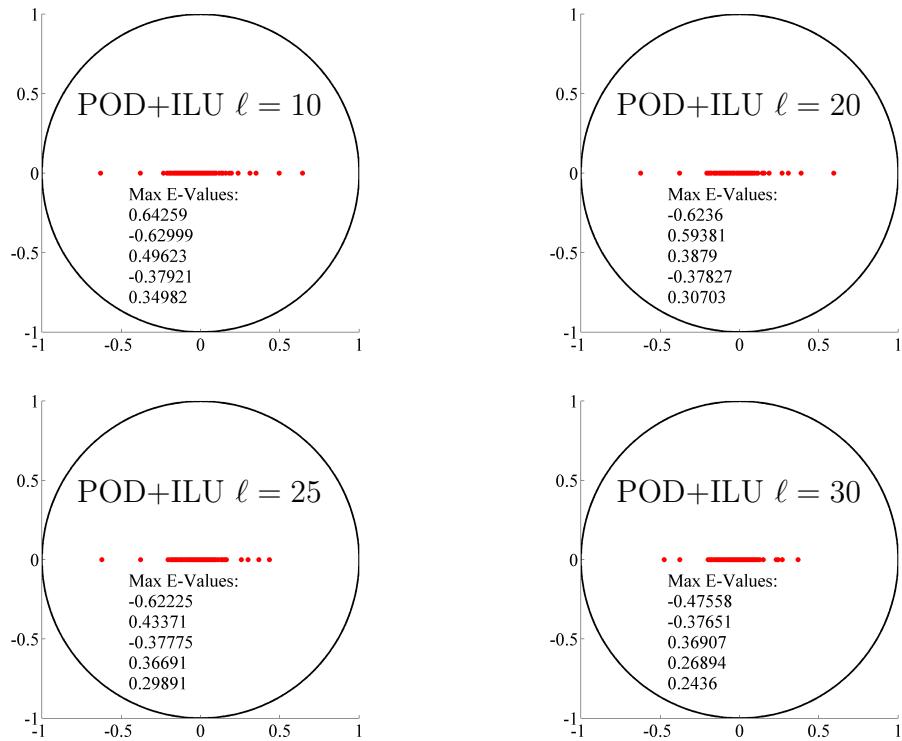


Figure 4.20: Plots of eigenvalues of iteration matrix for the heterogeneous case, Step 70. POD+ILU with  $\ell = 10$  (top left),  $\ell = 20$  (top right),  $\ell = 25$  (bottom left), and  $\ell = 30$  (bottom right). The largest (magnitude) eigenvalues are listed in each plot.

the effectiveness of the POD preconditioner compared to the ILU-only preconditioner. It also shows faster convergence as  $\ell$  increases from 10 to 25. However, the convergence trend given by the spectral radius changes only slightly. Moreover, in the lower plot of Fig. 4.21, the convergence history for a random RHS shows a different pattern, whereby it follows the spectral radius trend.

### 4.7.2 Heterogeneous Case ( $30 \times 30$ )

The reservoir model in this case contains  $30 \times 30$  grid blocks. The heterogeneous permeability field is shown in Fig. 4.22. Each grid block is of dimensions  $16.67 \text{ ft} \times 16.67 \text{ ft} \times 25 \text{ ft}$ . Other model specifications are as in the homogeneous case.

The eigenvalue plot for Step 90 ( $T = 77.4$  days) is shown in Fig. 4.23. One can see the eigenvalues are distributed around one. This implies the iterative solver is ineffective on some eigenvector components of  $\mathbf{A}$ . The spectral radius only shrinks a small amount as we increase  $\ell$ . The corresponding convergence history is shown in Fig. 4.24. The trends given by the spectral radius (dashed line) are very slow, while for the POD+ILU preconditioner, the convergence is faster as  $\ell$  goes from 25 to 75. The convergence for a random RHS (black curves) displays greater dependence on the spectral radius than it does on  $\ell$ .

## 4.8 Summary

Based on the results presented in this chapter, the main findings can be summarized as follows:

1. Compared with the ILU preconditioner, the POD+ILU pressure preconditioning strategy does accelerate the convergence rate uniformly.
2. The POD+ILU preconditioner has better performance for problems that had

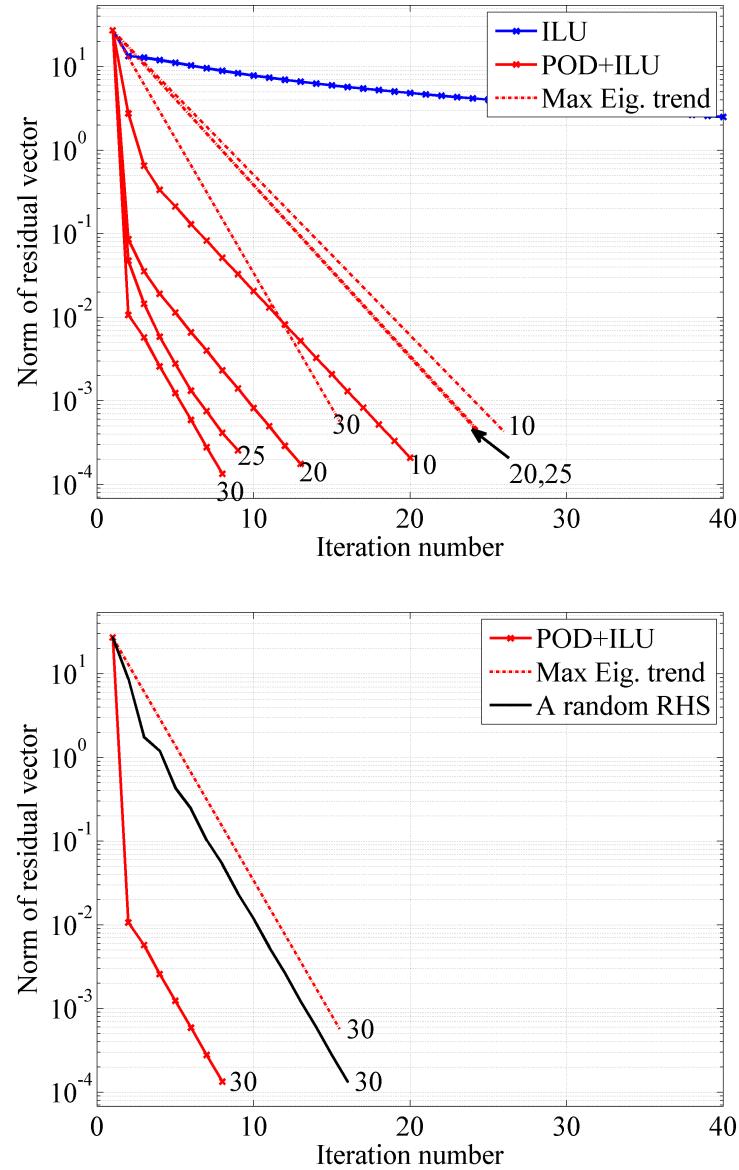


Figure 4.21: Convergence history for the heterogeneous case ( $10 \times 10$ ) for Step 70. The red curves correspond to  $\ell = 10, 20, 25, 30$ . The dashed lines show convergence trends given by the spectral radius. The lower plot shows comparison with the case with a random RHS vector,  $\ell = 30$ .

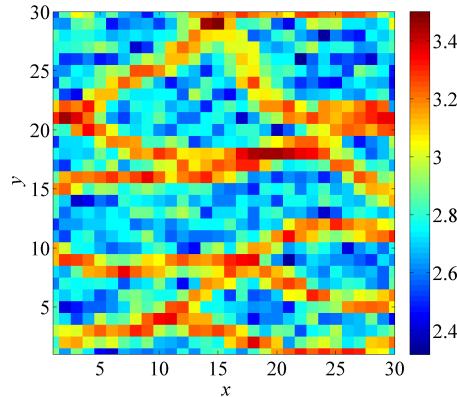


Figure 4.22: Log<sub>10</sub>-permeability distribution for the larger case.

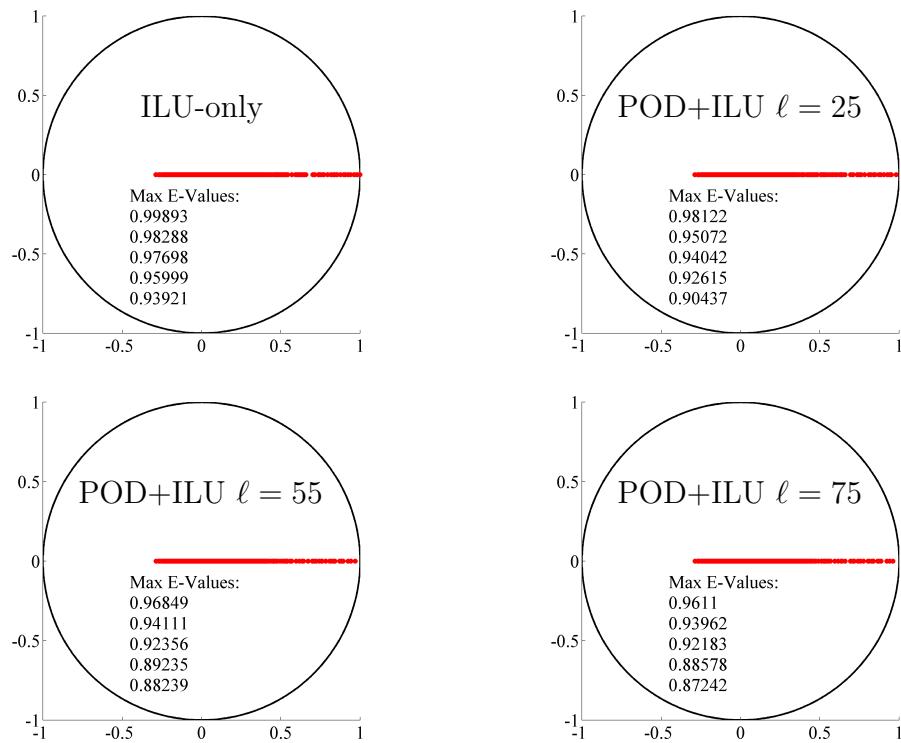


Figure 4.23: Plots of eigenvalues of iteration matrix for the larger case for Step 90.  $T = 77.4$  days. ILU-only (top left), POD+ILU with  $\ell = 25$  (top right),  $\ell = 55$  (bottom left), and  $\ell = 75$  (bottom right). The largest eigenvalues are listed in each plot.

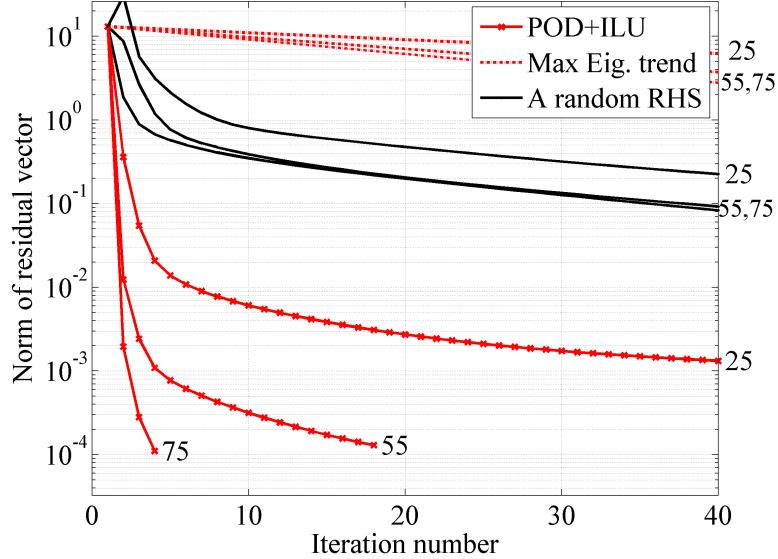


Figure 4.24: Convergence history for the larger case for Step 90. The red curves correspond to  $\ell = 25, 55, 75$ . The dashed lines show convergence trends given by the spectral radius. The black curves correspond to a random RHS.

been “seen” before (i.e., during the training simulation) compared with new or “unseen” problems.

3. The convergence rate is faster when we include more columns in the POD basis.
4. As we increase  $\ell$  over a certain range, the spectral radius of the POD+ILU iteration matrix does not change appreciably or consistently. Hence, the convergence of the POD+ILU preconditioner does not always follow the trend of the spectral radius.
5. For a random RHS, the convergence behavior does follow the trend given by the spectral radius.
6. The POD preconditioner can reduce the residual for certain frequency components (eigenvectors), though it has little or no effect on other frequencies.

We thus conclude that the POD method may not be suitable as a general pressure preconditioner, since it leaves certain low-frequency errors unresolved. However, POD does work well for problems that have been solved previously in the training simulation. Therefore, having solved the linear problem  $\mathbf{A}'\mathbf{x} = \mathbf{b}'$ , the POD method will be effective in solving similar problems  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A} \approx \mathbf{A}'$  and  $\mathbf{b} \approx \mathbf{b}'$ .

# Chapter 5

## Conclusions and Future Work

In this thesis, a linear-solver preconditioning strategy for the pressure system using the Proper Orthogonal Decomposition (POD) method was presented. The motivation of POD is to approximate points in a high-dimensional space using vectors of much lower dimensions. Chapter 2 discussed POD-based Reduced-Order Modeling (ROM) methods. Several projection schemes were presented. In terms of accuracy, the Galerkin projection scheme appears to give the best approximation. Some computational issues, such as POD quality indication and the recalculation of the dynamic POD subspace were discussed. These procedures could contribute to the design of an adaptive POD preconditioning strategy. In Chapter 3, we presented a spectral analysis to assess the POD preconditioner. Different preconditioners were shown to operate differently on the frequency spectrum of the matrix  $\mathbf{A}$ . A qualitative example was used to illustrate the different behaviors of the high-frequency and low-frequency errors, and the latter were shown to be more difficult to resolve. Next, we discussed the iteration matrix spectrum, which is crucial for the preconditioner assessment.

In Chapter 4, a  $10 \times 10$  homogeneous oil/water reservoir model using the IMPES solution scheme was presented. The performance of the POD pressure preconditioner

was studied in detail for this model. Various plots of convergence histories and iteration matrix spectra were shown. The results showed that the convergence of the POD+ILU preconditioner was faster for larger values of  $\ell$  (number of columns), compared with the ILU-only preconditioner. However, the spectral radii were about the same for a wide range of  $\ell$  values. This is contrary to the convergence histories, which were quite sensitive to the value of  $\ell$ . We demonstrated the effects of the ILU and POD preconditioners on the frequency spectrum of  $\mathbf{A}$ . Then, the eigenvectors of the Jacobian matrix were taken as the RHS. It was found that the POD preconditioner had no effect on many frequencies in the spectrum. Components of these frequencies were absent from the RHS that appeared in the simulation. This behavior suggested that the POD preconditioner is fundamentally RHS-dependent, and is not a traditional preconditioner that solves the problem equally well for any RHS. Findings for  $10 \times 10$  and  $30 \times 30$  heterogeneous models were consistent with those noted above.

Although the POD+ILU scheme seems not to be very effective for pressure systems, there are still many other ways to exploit the benefit of the POD-based ROM method. For example, one can combine POD, AMG and ILU to form a three-stage preconditioner to achieve higher accuracy and efficiency. Moreover, the idea of adaptive preconditioning may be worth pursuing.

# Nomenclature

## Abbreviations

AMG	algebraic multigrid
BHP	bottom-hole pressure
BILU	block incomplete LU decomposition
GMRES	generalized minimal residual method
GPRS	general purpose research simulator
ILU	incomplete LU decomposition
LSP	least-square projection
MsFVM	multiscale finite-volume method
PCA	principal component analysis
POD	proper orthogonal decomposition
RHS	right-hand-side
ROM	reduced-order modeling or reduced-order model
SPD	symmetric positive definite
SVD	singular value decomposition
TAMS	two-stage algebraic multiscale
TPWL	trajectory piecewise linearization

<b>A</b>	high-dimensional matrix ( $n \times n$ ) for a linear system
<b>A</b> <sub>r</sub>	reduced matrix ( $\ell \times \ell$ ) for a linear system
<b>b</b>	full right-hand-side, $n$ -dimensional
<b>b</b> <sub>r</sub>	reduced right-hand-side, $\ell$ -dimensional
<i>c</i> <sub>r</sub>	rock compressibility
<b>D</b>	diagonal matrix containing all the eigenvalues
<b>E</b>	matrix containing all error vectors
<b>e</b>	error vector
<b>G</b>	iteration matrix
<i>k</i>	absolute permeability
<i>k</i> <sub>r</sub>	relative permeability
$\ell$	order of the reduced problem or number of columns retained in $\Phi$
<b>M</b> <sup>-1</sup>	preconditioner matrix
<b>M</b> <sub>comp</sub> <sup>-1</sup>	POD+ILU two-stage preconditioner matrix
<b>M</b> <sub>POD</sub> <sup>-1</sup>	POD preconditioner matrix
<i>m</i>	number of snapshots used in POD algorithm
<i>n</i>	order of the full problem or number of grid blocks
<b>P</b>	projection operator
<b>P</b>	prolongation operator in multiscale context
<i>p</i>	pressure
<i>q</i>	source term in the units of surface volume rate
$\tilde{q}$	source term in the units of surface volume rate per reservoir volume
<b>R</b>	restriction operator in multiscale context
$\mathcal{R}(\Phi)$	column space of matrix $\Phi$
$\mathcal{R}(\Phi)^\perp$	orthogonal complement of $\mathcal{R}(\Phi)$
$\mathcal{R}(\Phi)_\mathbf{A}^\perp$	orthogonal complement of $\mathcal{R}(\Phi)$ in $\mathbf{A}$ -norm
<b>r</b> <sub><i>k</i></sub>	residual vector in iterative linear solver

<b>S</b>	SPD matrix defining a norm
<i>S</i>	saturation
<i>T</i>	rock and geometric part of transmissibility
<i>t</i>	time
<b>U</b>	matrix containing left singular vectors
<b>u</b>	left singular vector
<b>V</b>	matrix containing eigenvectors or right singular vectors
<i>V</i>	grid block volume
<b>v</b>	eigenvector or right singular vector
WI	well index
<b>X</b>	data matrix containing $m$ snapshots
<b>x</b>	full vector of unknowns, $n$ -dimensional
<b>x</b> <sub><i>i</i></sub>	snapshot vector
<b>z</b>	reduced vector of unknowns, $\ell$ -dimensional

**Greek**

$\alpha$	angle of POD subspace direction in 2D example
$\Delta t$	time step size
$\delta \mathbf{p}$	update in pressure solution
$\delta \mathbf{x}$	solution update
$\theta$	direction angle of $\hat{\mathbf{x}}$ (the true solution) in 2D example
$\lambda$	mobility or eigenvalue
$\mu$	viscosity
$\xi$	number of nonzero elements in a row of matrix <b>A</b>
$\rho(\mathbf{G})$	spectral radius of matrix <b>G</b>
<b><math>\Sigma</math></b>	diagonal matrix containing all singular values
$\sigma$	singular value

<b><math>\Phi</math></b>	POD basis matrix of dimensions $n \times \ell$
$\phi$	porosity
<b><math>\Psi</math></b>	left projection operator or constraint reduction matrix

### Subscripts

$F$	Frobenius norm indicator
$i$	grid block index or enumeration index
$k$	iteration index
$o$	oil phase
$r$	reduced representation
<b>S</b>	<b>S</b> -norm label
$t$	total
$w$	water phase

### Superscripts

$n$	time step index
$T$	matrix transpose
$\perp$	orthogonal complement label

# Appendix A

## Basics of the POD Method

Proper Orthogonal Decomposition serves as a fundamental element in many ROM procedures. In this Appendix, we review the basics of the POD method. In particular, the POD approximation error is discussed and examined in detail.

### A.1 Definition of POD

Proper Orthogonal Decomposition (POD) is also called Principal Component Analysis (PCA), discrete Karhunen–Loëve Transform (KLT), or the Hotelling Transform, depending on the field of application. We consider POD with the aim of reducing the order, or representing trajectories in high-dimensional space, using a minimum number of basis functions [4].

Suppose we want to approximate the pressure distribution over time  $p(x, t)$  using a finite sum in the variables-separated form

$$p(x, t) \approx \sum_{k=1}^M \phi_k(x) z_k(t), \quad (\text{A.1})$$

where  $\phi_k(x)$  are called the basis functions, which are normalized and orthogonal to

each other. Due to the orthonormality, the following relation holds for the coefficients:

$$z_k(t) = \int p(x, t)\phi_k(x)dx. \quad (\text{A.2})$$

We seek such an ordered orthonormal set of basis functions  $\{\phi_k(x)\}$ , such that for every value of  $M$  the least-square approximation error in (A.1) is as small as possible. That means that  $\phi_1(x)$  is chosen such that the one-term approximation expression  $p(x, t) \approx \phi_1(x)z_1(t)$  has the least possible error;  $\phi_2(x)$  is chosen, such that the two-term approximation  $p(x, t) \approx \sum_{k=1}^2 \phi_k(x)z_k(t)$  has the least possible error, and so on. Such a sequence of orthonormal basis functions  $\{\phi_k(x)\}$  is called the “proper orthogonal modes” of function  $p(x, t)$ . The expression (A.1) together with these functions is called the POD of  $p(x, t)$  [4].

In reservoir simulation, the spatially discretized grid cells are numbered from 1 to  $n$ . The simulation time period is divided into  $m$  time steps. The solution at each time step is referred to as a snapshot. Each snapshot  $\mathbf{x}_i$ ,  $i = 1, \dots, m$ , is represented as a column vector of length  $n$ . We collect all the solution snapshots and write them as an  $n \times m$  matrix  $\mathbf{X}$ :

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m). \quad (\text{A.3})$$

Usually  $n \gg m$ ; hence,  $\mathbf{X}$  is tall and thin. In some applications, such as statistics,  $n \ll m$  [42]. If we use  $\ell < m$  POD basis functions (columns), the POD approximation for this snapshot collection becomes

$$\tilde{\mathbf{X}} = \Phi\Phi^T\mathbf{X}, \quad (\text{A.4})$$

where the  $n \times \ell$  matrix  $\Phi$  contains the  $\ell$  basis columns.

POD can be calculated by a performing singular value decomposition (SVD) of

$\mathbf{X}$  [4, 36]:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \quad (\text{A.5})$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are, respectively, two orthogonal matrices of dimension  $n \times n$  and  $m \times m$ . The columns of  $\mathbf{U}$  and  $\mathbf{V}$  are called left and right singular vectors. The  $n \times m$  matrix  $\Sigma$  is a diagonal matrix in which the entries  $\{\sigma_i\}_{i=1}^m$  are sorted in descending order, and they are called the singular values of matrix  $\mathbf{X}$ . Here, the number of nonzero singular values  $r \leq m$  is equal to the rank of the matrix  $\mathbf{X}$ .

An important property of SVD is the low-rank approximation [34]. Suppose, we want to find a matrix  $\tilde{\mathbf{X}}$  to approximate  $\mathbf{X}$  under the Frobenius norm. We write the error as:

$$E = \|\mathbf{X} - \tilde{\mathbf{X}}\|_F = \|\mathbf{U}\Sigma\mathbf{V}^T - \tilde{\mathbf{X}}\|_F = \|\Sigma - \mathbf{U}^T\tilde{\mathbf{X}}\mathbf{V}\|_F, \quad (\text{A.6})$$

where the last equality comes from the norm invariant property under multiplication by orthogonal matrices. We have a better approximation the closer  $\mathbf{U}^T\tilde{\mathbf{X}}\mathbf{V}$  is to  $\Sigma$ . If we keep the first  $\ell$  diagonal elements in  $\Sigma$ , set  $\sigma_{\ell+1} = \dots = \sigma_m = 0$ , and denote the resulting matrix as  $\Sigma_r$ , then we see that the matrix

$$\tilde{\mathbf{X}} = \mathbf{U}\Sigma_r\mathbf{V}^T \quad (\text{A.7})$$

is the best  $r$ -rank approximation of  $\mathbf{X}$  [34].

Comparing the two equations (A.4) and (A.7), we find that, if we take the first  $\ell$  columns of  $\mathbf{U}$  as  $\Phi$ , then the approximation (A.4) satisfies the POD property due to the best low-rank approximation property of SVD by observing:

$$\begin{aligned} \Phi\Phi^T\mathbf{X} &= \Phi\Phi^T\mathbf{U}\Sigma\mathbf{V}^T \\ &= \mathbf{U}\Sigma_\ell\mathbf{V}^T, \end{aligned} \quad (\text{A.8})$$

where the second equality comes from fact that  $\Phi$  consists of the first  $\ell$  columns of  $\mathbf{U}$ .

So, POD is a SVD, and the proper orthogonal modes  $\{\phi_k\}$  are left singular vectors of  $\mathbf{X}$ .

In practice, it is more efficient to perform a reduced-size SVD, in which only the first  $r$  columns of  $\mathbf{U}$  and  $\mathbf{V}$  are calculated. Here,  $r$  can be less than or equal to the rank of  $\mathbf{X}$ . Usually, we take  $r = m$ . Therefore, in the SVD or POD expression (A.5), the size of  $\mathbf{U}$  is  $n \times m$  (tall and thin), which is the same as  $\mathbf{X}$ . Note that  $\Sigma$  and  $\mathbf{V}$  are  $m \times m$  square matrices.

From now on, we consider the case where  $n \gg m$ , and we denote the matrix  $\mathbf{X}$  as a collection of  $m$  snapshots, where each snapshot lives in  $\mathbb{R}^n$  space.

## A.2 Principal Components

The term “principal component analysis” (PCA) is used in the context of statistics [42], where the objective is to analyze data sets to summarize their main characteristics. Compared with POD, PCA usually requires the data set to have a zero mean [4, 43]. This issue will be discussed in the next section. The idea of “principal components,” which parallels the idea of proper orthogonal modes, can give us an intuitive view of the POD method.

The first column of  $\mathbf{U}$ , or the first Principal Component (PC), captures the most prominent feature in the snapshot collection by selecting the direction along which the data have the largest variance. Similarly, the second PC corresponds to the most prominent feature of the snapshots after subtracting the first PC; the third PC corresponds to the direction of the most prominent feature after subtracting the first two PCs, and so on. Note that PCA is sensitive to scaling, i.e., the principal components may change, if some vectors in  $\mathbf{X}$  are multiplied by a constant.

To illustrate this more clearly, we consider three pressure change solutions  $\delta\mathbf{p}$  that correspond to three consecutive time steps from a flow simulation, and we compute

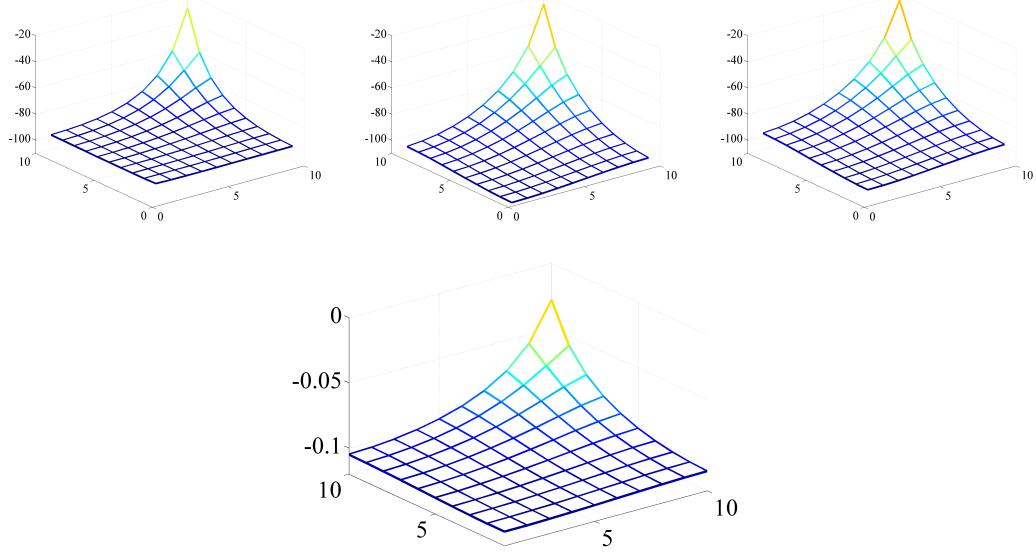


Figure A.1: POD illustration. Top: three pressure fields (snapshots),  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ . Bottom: first principal component  $\mathbf{u}_1$ , with singular value 1678.

the corresponding POD (see Fig. A.1-A.3). In Fig. A.1, we see that the first principal component (bottom) captures the overall pattern of the three snapshots (top) reasonably well. In Fig. A.2, the first three plots are the three snapshots after subtracting the first PC (top). The resulting pattern is captured well by the second principal component (bottom). Finally, in Fig. A.3, we further subtract the second PC (top), and the remaining pattern is the third PC (bottom). The magnitude of the singular values (1678, 34.49, 0.8132) decreases rapidly. This magnitude is a measure of the relative importance, or contribution, of each PC to the entire snapshot collection. That is, the first PC is much more important than the remaining two, and so on.

It is more difficult to interpret the principal components if the snapshots are very different, or equivalently, if the directions of the  $\mathbf{x}_i$  vectors are very different. For example, consider the three pressure fields from successive time steps (Fig. A.4). The principal components are shown in the bottom three plots. The singular values are 1.152, 0.7644, 0.5484. They do not decrease rapidly, which means that the first PC is

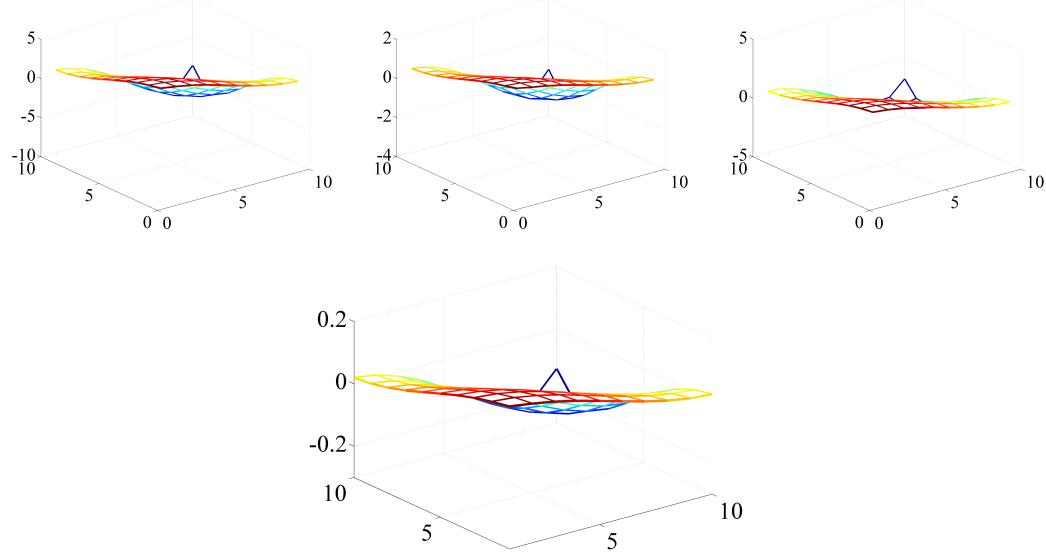


Figure A.2: POD illustration. Top: snapshots after subtracting the first PC component,  $\mathbf{x}_i - \mathbf{u}_1 \mathbf{u}_1^T \mathbf{x}_i, i = 1, 2, 3$ . Bottom: second principal component,  $\mathbf{u}_2$ , with singular value 34.49.

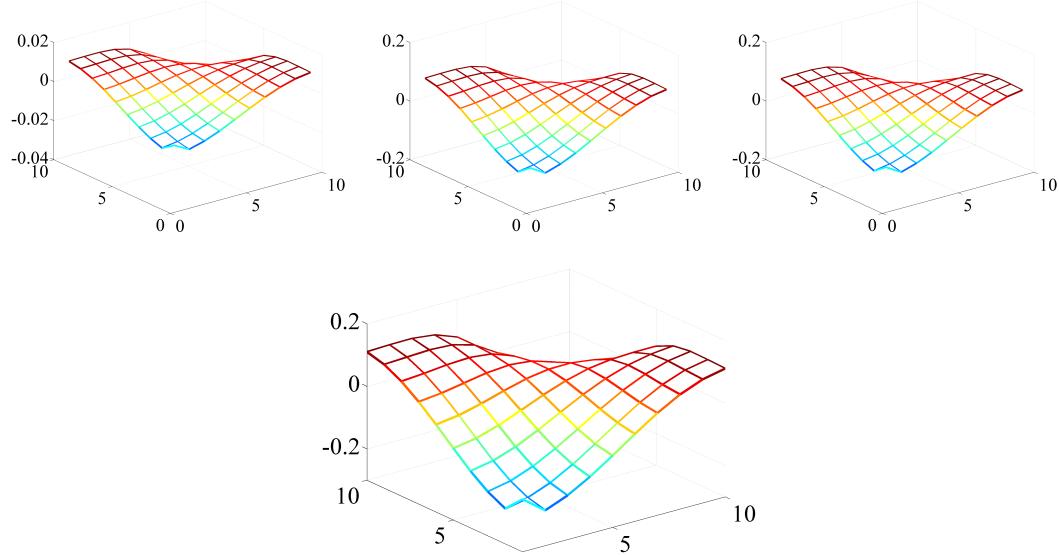


Figure A.3: POD illustration. Top: snapshots after subtracting the first two PC components,  $\mathbf{x}_i - \mathbf{u}_1 \mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_2 \mathbf{u}_2^T \mathbf{x}_i, i = 1, 2, 3$ . Bottom: third principal component,  $\mathbf{u}_3$ , with singular value 0.8132.

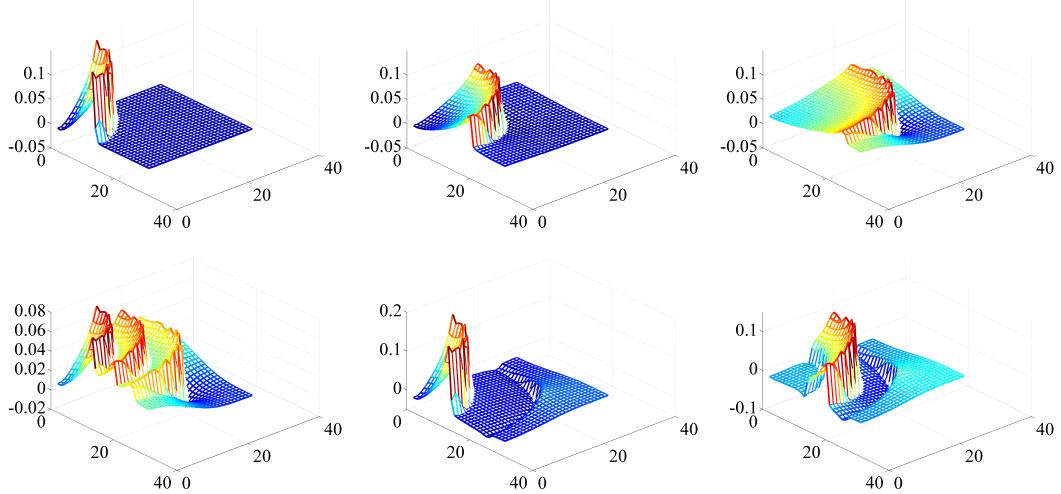


Figure A.4: POD illustration. Top: three pressure fields (snapshots),  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ . Bottom: three principal components  $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$  with singular values 1.152, 0.7644, 0.5484, respectively.

not significantly more important than the other PCs. In this case, the first few PCs do not capture the overall character of the snapshots; hence, one needs more PCs to describe the snapshots accurately. The relationship between the singular values and the POD approximation error is discussed in the next section.

### A.3 POD Approximation of High-Dimensional States

The POD approximation relation  $\mathbf{X} \approx \Phi \mathbf{z}$ , where  $\mathbf{z} = \Phi^T \mathbf{X}$ , is a low-rank representation of the snapshot collection  $\mathbf{X}$ , with the least possible error. The relation  $\ell \leq m \ll n$  ensures order reduction. A vector  $\mathbf{x}$  is approximated by discarding components of  $\mathbf{x}$  associated with the least important proper orthogonal modes (mode  $\ell + 1$  to mode  $m$ ).

Discarding columns will bring some error into the representation  $\mathbf{x} \approx \Phi \mathbf{z}$ . The error can be quantified using the singular values [36]. We denote the differences between the original vectors and the reconstructed vectors as  $\mathbf{e}_i = \mathbf{x}_i - \Phi \Phi^T \mathbf{x}_i$ ,

$i = 1, \dots, m$ . According to (A.6), the squared sum of the approximation error  $\Delta^2$  can be written as:

$$\Delta^2 = \sum_{i=1}^m \|\mathbf{x}_i - \Phi \Phi^T \mathbf{x}_i\|^2 = \text{tr}(\mathbf{E}^T \mathbf{E}) = \sum_{i=\ell+1}^m \sigma_i^2, \quad (\text{A.9})$$

where matrix  $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m)$ . This error relation can be generalized to a POD basis  $\Phi$  containing any combination of the columns in  $\mathbf{U}$ . The squared norm of the reconstruction error  $\text{tr}(\mathbf{E}^T \mathbf{E})$  is equal to the sum of the squared singular values that correspond to the discarded columns of  $\mathbf{U}$  (see, e.g., [36]).

**Effect of Scaling** Different scaling of the snapshots leads to a different POD due to the different weight factors for the snapshots. To illustrate this, we examine the effect of normalizing the snapshots.

Suppose that the snapshots in  $\mathbf{X}$  have different vector lengths, that POD is calculated after normalizing the snapshots, and that  $\Phi$  consists of the first  $\ell$  proper orthogonal modes. Then, the error equation (A.9) becomes

$$\Delta^2 = \sum_{i=1}^m \left\| \frac{1}{\|\mathbf{x}_i\|} \mathbf{x}_i - \frac{1}{\|\mathbf{x}_i\|} \Phi \Phi^T \mathbf{x}_i \right\|^2 = \sum_{i=\ell+1}^m \sigma_i^2. \quad (\text{A.10})$$

Instead of the case without normalization, where POD deals with absolute errors, this POD deals with relative errors. We minimize the relative errors by discarding the last few POD modes.

The error equation can also be written as

$$\Delta^2 = \sum_{i=1}^m \frac{1}{\|\mathbf{x}_i\|^2} \|\mathbf{x}_i - \Phi \Phi^T \mathbf{x}_i\|^2 = \sum_{i=\ell+1}^m \sigma_i^2, \quad (\text{A.11})$$

which means that the squared error of different snapshots is weighted by the reciprocal

of the squared norm of the snapshot. Therefore, a snapshot with a smaller norm does not necessarily have a smaller impact compared with one that has a larger norm. All snapshots equally contribute to the “most prominent feature” or the POD basis. In contrast, without normalization, a snapshot with a larger norm will be more important; hence, it contributes more to the POD modes.

**Mean Subtraction** In some areas, especially statistics, it is common to subtract the mean of the data points before performing POD. Some discussion of this issue can be found in [4, 43]. Essentially, subtracting the mean will reduce the approximation error by a small amount, which can be compensated for by including one more proper orthogonal mode in  $\Phi$ . Moreover, mean subtraction becomes unnecessary when  $\Phi$  is used to approximate snapshots that lie outside of the POD space, since we do not know the population mean in this case. There are many theoretical studies on the relationship between the sample PCA, which is relevant for the case of a finite number of snapshots, and the population PCA, which corresponds to the entire trajectory [44, 45]. We do not apply mean subtraction in this thesis.

**Errors Reduced by Specific Columns** It is interesting to examine how much each POD mode reduces the reconstruction error for different snapshots. Fig. A.5 shows  $\|\mathbf{x}_i - \Phi\Phi^T\mathbf{x}_i\|$ , which is the reconstruction error for snapshots from a reservoir simulation problem. This problem involves 39 time steps and thus 39 snapshots. POD here deals with absolute errors, since no normalization is applied. The distance between two adjacent red curves shows the incremental error reduction given by specific columns. For example, from the black curve and the uppermost red curve, we can see that the first two modes mainly approximate the first 12 snapshots. However, it is still not clear how we can utilize this analysis in the POD-ROM solution process.

We note finally that POD can be used to approximate points that lie outside, but

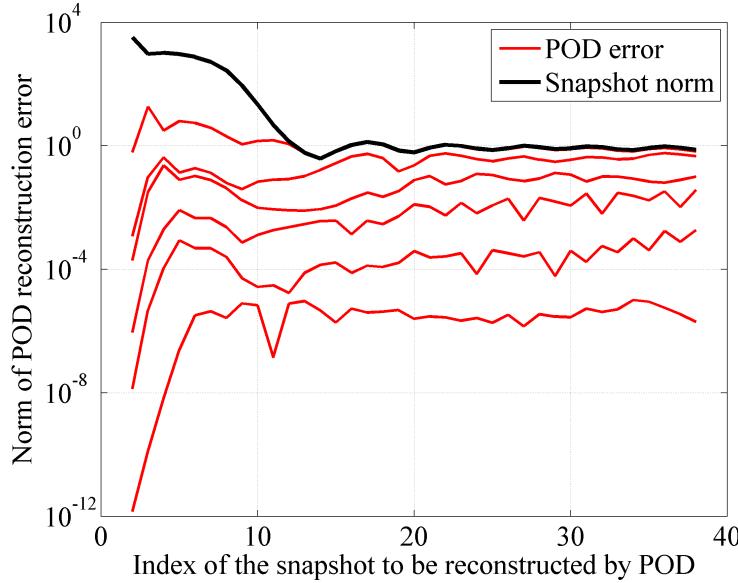


Figure A.5: Reconstruction error for different snapshots. For the red curves from top to bottom,  $\ell=2, 6, 10, 14, 18$ , and  $22$ . The thick black curve is the norm of the snapshots, shown as a reference.

are close to, the column space of  $\mathbf{X}$ . Analysis of this type of approximation error is discussed in Section 2.4.1 and Section 3.2.

## A.4 Computational Cost of SVD

The computational cost is obtained by counting the number of floating-point operations (flops), which include addition, subtraction, multiplication, and division of two floating-point numbers. The SVD calculation for an  $n \times m$  matrix  $\mathbf{X}$  involves two steps. We assume  $n \gg m$ . The first step involves a bi-diagonalization of matrix  $\mathbf{X}$ . This is done in an algebraic way, and it requires approximately  $2nm^2 + 2m^3$  flops. The second step is to iteratively eliminate the superdiagonal elements. This step entails  $O(m)$  flops. Hence, the first step is the most expensive. Furthermore, if one calculates the  $m$  left singular vectors of  $\mathbf{U}$  (reduced SVD), the computational cost is

$6nm^2 + 11m^3$ . Details can be found in [34, 36].



# Bibliography

- [1] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design And Control. Society for Industrial and Applied Mathematics, 2005.
- [2] W. H. A. Schilders, H. van der Horst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, 2008.
- [3] M. A. Cardoso. *Development and Application of Reduced-Order Modeling Procedures for Reservoir Simulation*. PhD thesis, Stanford University, 2009.
- [4] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.
- [5] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002.
- [6] P. Kerfriden, P. Gosselet, S. Adhikari, and S. P. A. Bordas. Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(58):850–866, 2011.
- [7] M.A. Cardoso and L.J. Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3):681–700, 2010.

- [8] M. A. Cardoso, L. J. Durlofsky, and P. Sarma. Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering*, 77(9):1322–1350, 2009.
- [9] R. Marković and J. D. Jansen. Accelerating iterative solution methods using reduced-order models as solution predictors. *International Journal for Numerical Methods in Engineering*, 68(5):525–541, 2006.
- [10] P. Astrid, G. Papaioannou, J. C. Vink, and J. D. Jansen. Pressure preconditioning using proper orthogonal decomposition (SPE paper 141922). In *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA, February 2011.
- [11] J. He, J. Sætrom, and L. J. Durlofsky. Enhanced linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 230(23):8313–8341, 2011.
- [12] H. Cao. *Deveopment of Techniques for General Purpose Simulators*. PhD thesis, Stanford University, 2002.
- [13] Y. Jiang. *Techniques for Modeling Complex Reservoirs and Advanced Wells*. PhD thesis, Stanford University, 2007.
- [14] Y. Fan. *Chemical Reaction Modeling in a Subsurface Flow Simulator with Application to In-Situ Upgrading and CO<sub>2</sub> Mineralization*. PhD thesis, Stanford University, 2010.
- [15] Y. Zhou. *Parallel General-Purpose Reservoir Simulation with Coupled Reservoir Models and Multisegment Wells*. PhD thesis, Stanford University, 2012.
- [16] J. He. Enhanced linearized reduced-order models for subsurface flow simulation. Master’s thesis, Stanford University, 2010.

- [17] J. He, P. Sarma, and L. J. Durlofsky. Use of reduced-order models for improved data assimilation within an EnKF context (SPE paper 141967). In *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA, February 2011.
- [18] M. A. Cardoso and L. J. Durlofsky. Use of reduced-order modeling procedures for production optimization (SPE paper 119057). *SPE Journal*, 15(2):426–435, 2010.
- [19] J. He and L. J. Durlofsky. Reduced-order modeling for compositional simulation using trajectory piecewise linearization (SPE paper 163634). In *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA, February 2013.
- [20] S. Trehan. Private communication, 2013.
- [21] S. Chaturantabut and D. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [22] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1):281–309, 2001.
- [23] J. R. Wallis, R. P. Kendall, T. E. Little, and J. S. Nolen. Constrained residual acceleration of conjugate residual methods (SPE paper 141922). In *SPE Reservoir Simulation Symposium*, Dallas, Texas, USA, February 1985.
- [24] H. Zhou and H. A. Tchelepi. Operator-based multiscale method for compressible flow (SPE paper 106254). *SPE Journal*, 13(2):267–273, 2008.
- [25] H. Zhou and H. A. Tchelepi. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models (SPE paper 141473). *SPE Journal*, 17(2):523–539, 2012.

- [26] H. Zhou. *Algebraic Multiscale Finite-Volume Methods for Reservoir Simulation*. PhD thesis, Stanford University, 2010.
- [27] J. He. *Reduced-Order Modeling for Oil-Water and Compositional Systems, with Application to Data Assimilation and Production Optimization*. PhD thesis, Stanford University, 2013.
- [28] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.
- [29] J. He. Private communication, 2013.
- [30] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.
- [31] B. N. Bond and L. Daniel. Guaranteed stable projection-based model reduction for indefinite and unstable linear systems. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 728–735. IEEE Press, 2008.
- [32] H. Cao, H. A. Tchelepi, J. Wallis, and H. Yardumian. Parallel scalable unstructured CPR-type linear solver for reservoir simulation (SPE paper 96809). In *SPE Annual Technical Conference and Exhibition*, Dallas, Texas, USA, October 2005.
- [33] P. M. Hall, D. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, pages 286–295, 1998.
- [34] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

- [35] M. J. L. Orr. Introduction to radial basis function networks. <http://www.anc.ed.ac.uk/rbf/intro/intro.html>, 1996. Online article.
- [36] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
- [37] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.
- [38] H. L. Stone and A. O. Garder Jr. Analysis of gas-cap or dissolved-gas drive reservoirs (SPE paper 001518). *SPE Journal*, 1(2):92–104, 1961.
- [39] K. H. Coats. A note on IMPES and some IMPES-based simulation models (SPE paper 65092). *SPE Journal*, 5(3):245–251, 2000.
- [40] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation (SPE paper 6893). *SPE Journal*, 18(3):183–194, 1978.
- [41] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer-Verlag, 2006.
- [42] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.
- [43] A. A. Miranda, Y. Le Borgne, and G. Bontempi. New routes from minimal approximation error to principal components. *Neural Processing Letters*, 27(3):197–207, 2008.
- [44] D. C. Hoyle and M. Rattray. PCA learning for sparse high-dimensional data. *EPL (Europhysics Letters)*, 62(1):117, 2003.
- [45] B. Nadler. Finite sample approximation results for principal component analysis: a matrix perturbation approach. *The Annals of Statistics*, 36(6):2791–2817, 2008.