Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

A Study of Reduced Order 4D-VAR with a Finite Element Shallow Water Model

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfilment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Martin Ambrozic

Delft, The Netherlands

November 19th, 2013

TU Delft Delft University of Technology

MSc THESIS APPLIED MATHEMATICS

A Study of Reduced Order 4D-VAR with a Finite Element Shallow Water Model

Martin Ambrozic

Delft University of Technology

Daily supervisor	**Responsible professor**
Dr. Umer Altaf	Prof. dr. ir. Arnold Heemink

Committee members

Prof. dr. ir. Arnold Heemink	Dr. ir. Martin van Gijzen
Dr. Umer Altaf	Dr. Peter Wilders

November 19th, 2013 Delft, The Netherlands

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgements

I would like to thank dr. Muhammad Umer Altaf and prof. dr. ir. Arnold Heemink for supervision and guidance throughout the creation of this thesis. Especially, I want to thank both for agreeing to let me spend the majority of the time in my home country.

For providing the shallow water equations model used herein, I extend thanks to prof. I. M. Navon and dr. X. Chen, both of Florida State University, USA.

This thesis represents the conclusion of my participation in the Erasmus Mundus master programme Computer Simulations in Science and Engineering. It has been an immensely valuable experience both educationally and otherwise and I wish to thank everyone who contributed to this, both in my first year at FAU Erlangen, Germany, and my second year at TU Delft.

Chapter 1

Introduction

1.1 Data Assimilation

An important use of numerical models of physical phenomena is forecasting the future from the known present state and history. Forecast models, however, depend on unknown parameters, such as model initial and boundary conditions, or other tunable parameters (not necessarily having any physical meaning), which affect the accuracy of the results. Calibration of these parameters is necessary to minimize errors between forecasted and observed states, otherwise the predictions made by the numerical model have no value in real life. Finding optimal values of these parameters by combining numerical model results with observations of the state of the physical system is called data assimilation (DA). A review of DA techniques can be found in [7] and [20].

Two basic approaches are known in DA:

- sequential assimilation, which uses only observations made before the time of analysis and
- non-sequential or retrospective assimilation, which uses also observations from the future of the time of analysis.

In sequential assimilation, a statistical technique called the Kalman filter is best known. Using the assumption of a linear underlying dynamical model and Gaussian distribution of error terms, it identifies the state of the system which has the highest probability given the previous state and measurements. It was introduced by R. E. Kalman in 1960 [21] and has since then seen numerous extensions, among which we mention G. Evensen's modification for systems with many variables, known as the Ensemble Kalman filter [13].

The focus of this thesis, however, is the non-sequential method of four-dimensional variational data assimilation, or 4D-VAR. This method estimates the unknown parameter set

by defining a cost function that takes as arguments the current parameter choices and penalizes misfits between observations and the corresponding numerical model results, obtained by running the model with the chosen configuration. This is done both in the spatial and temporal dimensions (thus the 4D in the name of the method). Optimization with regard to this cost function leads to an improved set of parameters with smaller misfits to the observed data.

Among the first and most prominent fields where 4D-VAR has been used, are weather systems [30, 19, 26], but applications in other fields also exist, such as oceanography [6], groundwater flows [32], tidal flows [1] and atmospheric chemistry [34]. In [4], 4D-VAR has also been proved effective with nested models.

One of the main problems when performing optimization in the context of 4D-VAR is obtaining the gradient of the cost function. Recognized as an efficient way of doing this is the adjoint method, which requires one forward model run and additionally only a single integration of the adjoint dynamical model back in time to obtain the required gradient. The computational complexity of the adjoint integration, however, can be equivalent to several forward model simulations [17, 18]. Depending on the complexity of the original dynamical model, programming its adjoint can also be difficult to do, owing to the need of in-depth understanding of the model code required to implement the partial derivatives of the model. Further programming load is added when making changes to the forward model, since the adjoint then also needs to be modified accordingly.

1.2 Model Order Reduction

Despite the enormous growth in computational power in recent decades, demands on complexity and volume of computations always seem to be one step ahead. Dynamical models with millions, if not billions of variables, that have become common today can be extremely computationally intensive even when using the latest hardware.

The goal of model order reduction is to approximate a large dynamical system with one that has (as far as possible) the same input-output behaviour, but lower dimension, thereby reducing computational times and storage requirements. There exist a variety of techniques to achieve this, all of which attempt to identify redundant information in the system's state and eliminating it from the computation, effectively reducing the system to fewer variables that encompass the majority of its dynamics. A common tool used to achieve this is the singular value decomposition (SVD), a matrix factorization that exposes dominant vector pairs involved in the corresponding transformation. These are called singular vectors and are associated with scalar singular values, which measure the contribution of each vector pair. Most well-known methods in model order reduction thus

include:

- SVD-based methods such as
  - Balanced truncation, which uses balancing to bring the system into a form where states that are hard to reach or hard to observe can easily be identified. Truncation then eliminates those states.
  - Hankel-norm approximation, a refinement of balanced truncation which is optimal in the 2-induced norm of finite-dimensional matrices and operators.
  - Proper orthogonal decomposition, which is an entirely data-driven method that uses system state snapshots in combination with the singular value decomposition to identify dominant states.
- Krylov-based methods using moment matching to approximate the transfer function of the system by a rational function of a lower degree. This is facilitated by expanding the transfer function into its Laurent series at various points of the complex plane.

An overview of model order reduction techniques as well as details on each of them can be found in [3].

1.2.1 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) was first introduced by K. Pearson in 1901 [27] and first named as POD by J. L. Lumley in 1967 when applied to turbulent flows in [24]. POD has since then proved to be a powerful and elegant method of data analysis, designed for the purpose of obtaining low-dimensional approximations to high-dimensional processes and data sets [9]. Also known as the Karhunen-Loeve decomposition, principal component Analysis and the Hotelling transform, it has been successfully applied to problems in a variety of fields, including thermal dynamics [25], fluid mechanics [16], structural dynamics [22] but also speech classification [23], reconstruction of images (particularly images of faces in [14]) and even inverse air foil design [8]. See also [28] for more examples of POD applications.

Essentially, POD analysis attempts to extract dominant components, called ”modes”, from sets of high-dimensional data obtained from experiments or simulations by a singular value decomposition (SVD) of the so-called correlation matrix. Using a Galerkin projector constructed from the dominant POD modes, low-dimensional approximate dynamical models can be obtained. A powerful property of POD is the guarantee of optimality, i.e. the basis of modes calculated by POD is guaranteed to provide minimal projection error in a least squares sense. Furthermore, since POD is entirely data-driven, it can be applied

to problems where the underlying mathematical model is unknown, but numerical data is available (such as in [25]).

1.3 Motivation and Research Goals

Many successful applications of the adjoint method in data assimilation have been performed in recent years. See for example [5], [11] and [6]. However, in order to reduce the programming effort involved in developing the adjoint model, order reduction techniques have been considered in this context. In [12] and [29], the model physics are first simplified and an approximate adjoint is built afterwards. This means that the optimization is done entirely in reduced space, which accelerates convergence, but there is no guarantee that the final solution in reduced space is the correct one in the full state space as well. In [33], the above was done in an efficient way using POD as the order reduction tool, but still operating entirely in POD-reduced space. A modification to this idea was proposed in [2] in 2013, using the POD-reduced adjoint only in gradient computation, keeping the full order forward model elsewhere and thus optimizing in the full order state space. This means that the programming and computational effort connected to the adjoint model is alleviated, but accuracy is improved because the full order space is used for cost function evaluations.

In this thesis, the method proposed in [2], named the Reduced Adjoint method, is studied and comparisons in terms of accuracy and computational intensity under various conditions are made with two other alternatives, namely the full order adjoint method and the method of Reduced Model Variation, where the low order model is used both in forward and adjoint directions.

To this end, the named methods are applied to two numerical models. Firstly, a linear model of subsurface contaminant transport from [1] is adopted and some of the experiments in the paper are examined to illustrate the effect of the methods on a linear model. Secondly, the same techniques are applied on a finite element model of the Shallow Water Equations developed by Chen and Navon in [11], where the methods are tested on a more complex non-linear model.

1.4 Content outline

The thesis is structured in the following manner.

Chapter 2 includes more detailed descriptions of techniques used later in the thesis. Proper orthogonal decomposition and 4D-variational data assimilation is described.

Chapter 3 applies inverse reduced order modelling techniques to a linear subsurface con-

taminant model [2]. POD is applied to the model in the forward direction and results of some of the experiments are examined to illustrate the methods of model order reduction on a relatively simple model.

Chapter 4 introduces the Shallow Water Equations and their Galerkin finite element model developed by Chen and Navon in [11]. The adjoint method is used in inverse modelling in the paper, which means that the adjoint model is available for comparison with its reduced order counterpart. Experiments are done with all three adjoint method versions (Adjoint method, Reduced Adjoint method, Reduced Model Variation). Dependence on the number of observation points and sensitivity to perturbations in observations is also tested.

Chapter 5 collects and summarizes results from previous chapters and gives some outlook on possible future research.

Chapter 2

Preliminaries

2.1 Proper Orthogonal Decomposition

Starting with a set of data from a vector space V, proper orthogonal decomposition (POD) aims at finding a vector subspace V_r of fixed dimension r, such that the error of projection (in a least square sense) is minimal. POD is entirely data-dependent, which means that, in principle, nothing of the underlying dynamics of the system that produced the data needs to be known. This makes it a very versatile tool, used in this thesis in the context of model order reduction.

2.1.1 The Singular Value Decomposition

The singular value decomposition (SVD) is defined in [15] in the following manner.

Theorem 2.1.1. *If A is an m-by-n matrix, then there exist orthogonal matrices*

$$U = [u_1, ..., u_m] \in \mathbb{R}^{m\times m} \quad \textit{and} \quad V = [v_1, ..., v_n] \in \mathbb{R}^{n\times n}$$

such that

$$U^T A V = diag(\sigma_1, ..., \sigma_p) \in \mathbb{R}^{m\times n} \quad p = min\{m, n\}$$

where

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0.$$

Proof of existence of the SVD for any real matrix A is omitted here, but also given in [15]. The σ_i are the singular values of A and the vectors u_i and v_i are the ith left singular vector and the ith right singular vector respectively. Normally, the diagonal matrix of singular values is denoted with Σ, by which we get the decomposition

$$A = U\Sigma V^T.$$

The SVD can be viewed as an eigenvalue problem by noting that

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^2 V^T$$

and

$$AA^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma^2 U^T$$

which means that left and right singular vectors can be calculated as eigenvectors of AA^T and A^TA, respectively.

Why the SVD is present in so many approximation schemes is illustrated in the following theorem from [31].

Theorem 2.1.2 (Low rank approximation)**.** *For any ν with $0 \leq \nu \leq r$ define*

$$A_\nu = \sum_{j=1}^{\nu} \sigma_j u_j v_j^*;$$

if $\nu = p = min\{m, n\}$, define $\sigma_{\nu+1} = 0$. Then

$$\|A - A_\nu\|_2 = \inf_{B \in \mathbb{R}^{m\times n}, rank(B) \leq \nu} \|A - B\|_2 = \sigma_{\nu+1}$$

Singular vector pairs thus capture as much of the energy of A as possible, where by energy the 2-norm or the Frobenius norm is meant.

2.1.2 The POD Method

Let $\mathbf{\Sigma}$ be a dynamical system of the form

$$\mathbf{\Sigma} : \dot{x}(t) = f(x(t), u(t)), \quad y(t) = g(x(t), u(t)). \tag{2.1}$$

We begin by collecting a set of state snapshots at times $t_1, ..., t_N$, assembling the so-called snapshot matrix $\mathbf{X}$

$$\mathbf{X} = [x(t_1), x(t_2), ..., x(t_N)] \in \mathbb{R}^{n\times N}. \tag{2.2}$$

We now define X^b as the background state and the subtract it from each snapshot, obtaining the corrected snapshot matrix E having columns E_i as

$$E_i = X_i - X^b, \quad i = 1, ..., N, \tag{2.3}$$

where X_i is the i-th column of $\mathbf{X}$. The objective is to find an orthonormal basis of vectors $u_j \in \mathbb{R}^n, j = 1, 2, ..., N$ such that

$$E_i = \sum_{j=1}^{N} \gamma_{ji} u_j, i = 1, 2, ..., N. \tag{2.4}$$

Additionally, we require that the truncated set of vectors $u_j, j = 1, 2, ..., k$ for any $k < N$ approximates the family $\{E_i\}$ optimally, i.e. denoting

$$\hat{E}_i = \sum_{i=1}^{k} \gamma_{ji} u_j$$

and

$$\underbrace{[\hat{E}_1 \cdots \hat{E}_N]}_{\hat{E}} = \underbrace{[u_1 \cdots u_k]}_{U_k} \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \vdots & \ddots & \vdots \\ \gamma_{k1} & \cdots & \gamma_{kN} \end{bmatrix} \tag{2.5}$$

we want $\|\hat{E} - E\|_2$ to be minimal.

It can be shown that the singular value decomposition (SVD) solves exactly the problem described. Given the matrix E, let

$$E = U\Sigma V^* \tag{2.6}$$

be its SVD. The columns of U are the orthonormal basis of vectors u_j that we were looking for and the coefficients γ_{ij} are contained in the matrix $\Gamma = \Sigma V^*$.

Computation of the SVD can be done by solving either the eigenvalue problem,

$$EE^* = U\Sigma^2 U^*,$$

thereby getting the matrix U directly, or the eigenvalue problem

$$E^*E = V\Sigma^2 V^*,$$

computing the required eigenvectors as $U = EV\Sigma^{-1/2}$. Which option we choose depends on which of the matrices EE^* and E^*E has smaller dimension.

Having computed the appropriate orthonormal basis, we can now define the Galerkin projection consisting of the leading k left singular vectors of the snapshot matrix, denoted by U_k in (2.5). The original dynamical system

$$\mathbf{\Sigma} : \dot{x}(t) = f(x(t), u(t)), \quad y(t) = g(x(t), u(t))$$

can then be approximated by the projected system

$$\hat{\mathbf{\Sigma}} : \dot{\hat{x}}(t) = U_k^* f(U_k\hat{x}(t), u(t)), \quad y(t) = g(U_k\hat{x}(t), u(t)). \tag{2.7}$$

The quality of approximation depends on the decay of the singular values of the snapshot matrix E. A useful concept in this context is the energy captured by the j-th singular vector, defined as

$$\psi_j = \frac{\sigma_j}{\sum_{i=1}^{N} \sigma_i}, \tag{2.8}$$

where σ_j is the j-th singular value. The total energy captured by the first r singular vectors is then simply the sum of their individual energies, i.e.

$$\psi^r = \sum_{i=1}^{r} \psi_i.$$

We summarize the POD procedure in the following algorithm

Algorithm 2.1 The POD algorithm

1: **INPUT:** dynamical system $\mathbf{\Sigma}$, state snapshot matrix E
2: **OUTPUT:** reduced order dynamical system $\hat{\mathbf{\Sigma}}$
3: compute SVD: $E = U\Sigma V^*$
4: U_k = first k columns of U
5: if necessary compute initial reduced state $\hat{x}_0 = U_k^* x_0$
6: define $\hat{\mathbf{\Sigma}} : \dot{\hat{x}}(t) = U_k^* f(U_k\hat{x}(t), u(t)), \quad y(t) = g(U_k\hat{x}(t), u(t))$

2.1.3 Example

Consider the tri-diagonal linear operator M, defined as

$$M = \frac{1}{4}\begin{bmatrix} 4 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & 1 & 4 & 1 \\ 0 & \cdots & 0 & 0 & 1 & 4 \end{bmatrix} \in \mathbb{R}^{20\times 20}.$$

We generate a matrix of snapshots $\mathbf{X}$ by starting with the vector

$$X_1 = (0,0,0,0,1,0,0,0,0,2,0,0,0,0,3,0,0,0,0,4)^T \in \mathbb{R}^{20}$$

and defining the columns of $\mathbf{X}$ as

$$X_i = M^{i-1}X_1, \quad i = 1,2,...,10.$$

For this example, we assume the background state to be zero and thus no corrections are made to the snapshots. Figure 2.1 shows the singular value plot of the matrix $\mathbf{X}$. Fast decay of the singular values is visible, therefore we assume that POD can be applied with success to this dataset. Following the POD procedure, we calculate the SVD of $\mathbf{X}$ and

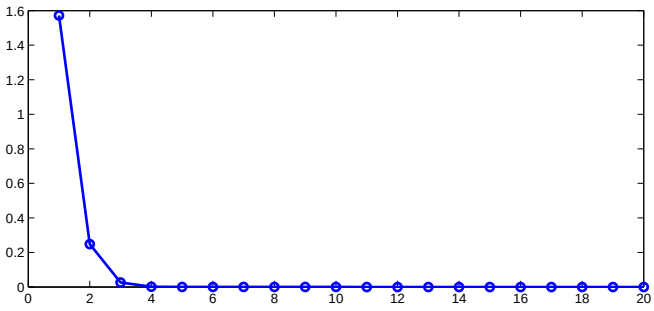


Figure 2.1: Singular value plot for the snapshot matrix $\mathbf{X}$.

extract 3 dominant modes from the singular vector matrix U that represent 99.96% total energy. They are shown in Figure 2.2.

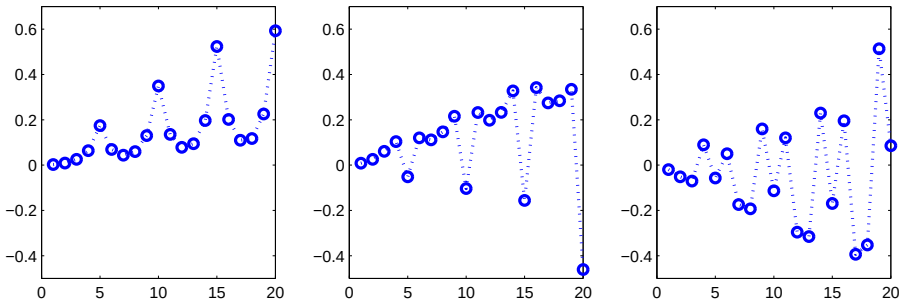


Figure 2.2: First 3 POD modes (left to right) for snapshot matrix $\mathbf{X}$.

2.2 Four Dimensional Variational Data Assimilation

We consider a discrete non-linear dynamical system, where the transition of the state X from time t_i to time t_{i+1} is governed by the non-linear dynamics operator $M_i : \mathbb{R}^n \to \mathbb{R}^n$ according to the equation

$$X(t_{i+1}) = M_i X(t_i), \quad i = 1, 2, ..., m-1. \tag{2.9}$$

Also given are observations of the system state, which we denote by $Y(t_i) \in \mathbb{R}^q$. These are related to the system state by the observation operator $H : \mathbb{R}^n \to \mathbb{R}^q$ by

$$Y(t_i) = H(X(t_i)) + \eta(t_i) \tag{2.10}$$

The term $\eta(t_i)$ models imperfections in the observations (such as measuring errors) and is assumed to be a white Gaussian observation noise process with mean zero and covariance matrix R_i.

The objective of four-dimensional variational data assimilation is to find the set of initial conditions to the system (2.9) that give the best fit to the available observed data. Put into more rigorous terms, we are looking for the initial state X_0 for which the function value

$$\begin{aligned} J(X_0) =& \frac{1}{2}(X_0 - X^b)^T B_0^{-1}(X_0 - X^b) + \\ & \frac{1}{2}\sum_{i=0}^{m}(Y(t_i) - H(X(t_i)))^T R_i^{-1}(Y(t_i) - H(X(t_i))) \end{aligned} \tag{2.11}$$

is minimal. The first term in the cost function $J(X_0)$ is the so-called background term and X^b is a prior estimate of X_0, assumed uncorrelated with covariance matrix B_0. The second term penalizes misfits between observations and obtained model data over the simulated time window. It is therefore clear that the J-minimal X_0 (constrained by the dynamics of the model) is the initial state which we are looking for.

Minimizing J is normally done with methods that require computing the gradient of J with respect to X_0 (which we denote with ∇J). This is most efficiently done by the adjoint

method, replacing the constrained optimization problem by unconstrained optimization of the functional

$$\begin{aligned}\bar{J}(X_0) =& \frac{1}{2}(X_0 - X^b)B_0^{-1}(X_0 - X^b) + \\ & \frac{1}{2}\sum_{i=1}^{m}(Y(t_i) - H(X(t_i)))^T R_i^{-1}(Y(t_i) - H(X(t_i))) + \\ & \sum_{i=1}^{m}\nu(t_i)^T(X(t_i) - M_i X(t_{i-1})).\end{aligned} \tag{2.12}$$

The vector $\nu(t_i)$ of Lagrange multipliers is then defined as the state vector of the adjoint system (see Appendix A for a more complete derivation). An expression for the gradient ∇J is given by

$$\nabla J(X_0) = -B_0^{-1}(X^b - X_0) - \nu(t_0). \tag{2.13}$$

Since we know that $\nu(t_m) = 0$, this requires solving the adjoint system backwards in time from t_m to t_0. Skipping the details of the derivation (see Appendix A), the backward time step scheme is given by

$$\nu(t_i) = \mathbf{M}_i^T \nu(t_{i+1}) + H^T R_i^{-1}(Y(t_i) - H(X(t_i))), \tag{2.14}$$

where $\mathbf{M}_i$ denotes the Jacobian of M_i with respect to X_i.

Summarizing the above steps is the following algorithm:

Algorithm 2.2 4D-VAR

1: **INPUT:** initial parameter set X_0 of dynamical system $\mathbf{\Sigma}$
2: **OUTPUT:** improved parameter set X_0^1
3: determine background state X^b from previous knowledge
4: execute optimization with X_0 as initial guess, use (2.11) as cost function and (2.13) as gradient to obtain X_0^1 as optimal initial state

One step of minimizing the cost function J then consists of one forward model run to determine the states $X(t_i)$ for $i = 1, ..., m$ and a backward run of the adjoint model, starting with $\nu(t_m) = 0$ and progressing towards $\nu(t_0)$, which is required to compute the gradient. The adjoint method is an efficient way of gradient computation in 4D-VAR, but its drawbacks include high computational complexity for large scale problems and even more unfortunate is the difficulty of actually programming the adjoint model. This is why approximate methods that avoid the full order adjoint model have become an important approach to the problem.

2.3 Inverse modelling using POD

Two approaches to reduced order inverse modelling will be considered, both formulated in [2]. The difference between the two methods is how the forward model is treated, as

we use the reduced order version in POD-reduced model variation and keep the full order forward model in the POD-reduced adjoint method.

2.3.1 POD-reduced Model Variation

The first idea of using model order reduction to approximate 4D-VAR by the adjoint method is to simply replace the entire model by its POD-derived approximate. The approximate state $\hat{X}(t_i)$ is used, satisfying

$$\hat{X}(t_i) = P\xi(t_i) + X^b \tag{2.15}$$

where the matrix P contains as columns the dominant POD modes of the system dynamics operator M_i. The approximate state $\hat{X}(t_i)$ is thus a linear combination of the dominant POD modes and $\xi(t_i)$, the reduced state vector, contains the corresponding coefficients. Its evolution in time is governed by

$$\xi(t_{i+1}) = \tilde{M}_i\xi(t_i), \tag{2.16}$$

where $\tilde{M}_i$ is the simplified dynamics operator, obtained from the Jacobian $\mathbf{M}_i$ as

$$\tilde{M}_i = P^T\mathbf{M}_iP.$$

Since we are now optimizing with respect to the reduced state, we modify also the cost function itself. We introduce the approximate cost function $\hat{J}$ and define it as

$$\begin{aligned}\hat{J}(\xi_0) =& \frac{1}{2}\xi_0^T P^T B_0^{-1} P\xi_0 + \\ & \frac{1}{2}\sum_{i=0}^{m}(Y(t_i) - H(\hat{X}(t_i)))^T R_i^{-1}(Y(t_i) - H(\hat{X}(t_i))),\end{aligned} \tag{2.17}$$

Since the reduced model is linear, its adjoint in reduced space is easy to obtain. The backward evolution equation for the adjoint states $\hat{\nu}(t_i)$ then reads as

$$\hat{\nu}(t_i) = \tilde{M}_i^T\hat{\nu}(t_{i+1}) + P^T H^T R_i^{-1}(Y(t_i) - H\hat{X}(t_i)), \tag{2.18}$$

and the gradient of the approximate cost function $\hat{J}$ can be computed as

$$\nabla\hat{J}(\xi_0) = -P^T B_0^{-1} P\xi_0 - \hat{\nu}(t_0). \tag{2.19}$$

Summarizing the above steps is the following algorithm:

Algorithm 2.3 POD-reduced Model Variation

1: **INPUT:** initial parameter set X_0 of dynamical system $\mathbf{\Sigma}$, state snapshot matrix E
2: **OUTPUT:** improved parameter set X_0^1
3: use POD to obtain the projection matrix P
4: compute the simplified dynamics operator $\tilde{M}_i$
5: calculate initial reduced state as $\xi_0 = P^T(X^0 - X^b)$
6: execute optimization with ξ_0 as initial guess, use (2.17) as cost function and (2.19) as gradient to obtain ξ_0^1 as optimal reduced initial state
7: output $X_0^1 = P\xi_0^1 + X^b$

This approach therefore performs the entire optimization in reduced space. This gives control over the speed of computation, since the number of used POD modes directly affects performance. The sacrifice, however, is lower accuracy. Furthermore, by optimizing in reduced space and then projecting back to the full space, no guarantee exists that the minimum found in reduced space is also the global minimum in the full space. To tackle these downsides, a different method was proposed by Altaf, Gharamti, Heemink and Hoteit in [2].

2.3.2 POD-reduced Adjoint Method

As mentioned in the previous section, optimizing with reference to the reduced cost function $\hat{J}$ has the drawback of decreased accuracy and possibly even convergence to the wrong solution. The POD-reduced adjoint method thus aims to rectify that by performing the cost function calculations in full space. In order to achieve that, this approach keeps the original forward model (2.9) and approximates the adjoint model only. The POD method is used to generate an approximate adjoint model by projecting the forward linear tangent operator onto the dominant POD modes and transposing it to get the approximate adjoint operator. This gives the following evolution equation for the adjoint states:

$$\hat{\nu}(t_i) = \tilde{M}_i^T \hat{\nu}(t_{i+1}) + P^T H^T R_i^{-1}(Y(t_i) - HX(t_i)). \tag{2.20}$$

The corresponding gradient calculation is now in full space and is given by

$$\nabla J(X_0) = -B_0^{-1}(X_0 - X^b)) - P\hat{\nu}(t_0). \tag{2.21}$$

In algorithmic form, the method reads as follows:

Algorithm 2.4 POD-reduced Adjoint Method

1: **INPUT:** initial parameter set X_0 of dynamical system $\mathbf{\Sigma}$, state snapshot matrix E
2: **OUTPUT:** improved parameter set X_0^1
3: use POD to obtain the projection matrix P
4: compute the simplified dynamics operator $\tilde{M}_i$
5: execute optimization with X_0 as initial guess, use (2.11) as cost function and (2.21) as gradient to obtain X_0^1 as optimal initial state

Since the forward model is now of full order, complexity is increased. We need to integrate the original model in every optimization iteration in order to compute the adjoint states needed in the gradient calculation (this can be seen from (2.20)). However, it was shown in [2] that accuracy is improved significantly by this approach.

2.4 The L-BFGS method

Optimization of the cost function requires a numerical minimization algorithm. A common choice in this area in 4D-VAR are quasi-Newton methods. In particular, the limited

memory Broyden-Fletcher-Goldfarb-Shanno update (L-BFGS) is widely used [2, 11].

Considering the minimization of an objective function $F(x)$ with respect to a vector x, the iterative scheme

$$x_{k+1} = x_k - \alpha_k H^{-1} \nabla F(x_k), \tag{2.22}$$

where H is the Hessian matrix, ∇F is the gradient vector and k the iteration number, is called Newton's method. In a large scale problem such as data assimilation, the calculation of the Hessian matrix H is too expensive. Quasi-Newton methods avoid its explicit computation by approximating H with first derivatives.

An example of such a method is the L-BFGS update, in which the approximation $\hat{H}$ of H in the k-th iteration is given by

$$\hat{H}_{k+1} = V_k^T \hat{H}_k V_k + \rho_k d_k d_k^T, \tag{2.23}$$

where

$$\rho_k = \frac{1}{y_k^T d_k}, \tag{2.24}$$

$$V_k = I - \rho_k y_k d_k^T \tag{2.25}$$

and

$$d_k = x_{k+1} - x_k, \tag{2.26}$$

$$y_k = \nabla F(x_{k+1}) - \nabla F(x_k). \tag{2.27}$$

To avoid storing the full matrices $\hat{H}_k$, a fixed number of pairs d_k, y_k is only stored. With these pairs, the k-th approximation of the Hessian $\hat{H}_k$ is defined implicitly and the product $\hat{H}_k \nabla F(x_k)$ can be computed as a series of inner products involving $\nabla(x_k)$ and the most recent pairs d_k, y_k, thus avoiding ever explicitly forming the Hessian and therefore saving memory.

Chapter 3

Ground Water Subsurface Contaminant Model

To demonstrate the effectiveness of model order reduction without having to deal with effects of non-linearities, we first consider a linear model, i.e. the ground water subsurface contaminant model. By applying proper orthogonal decomposition, a forward model of smaller dimension can be constructed and only small errors observed in the simulation of the reduced model. Furthermore, the same technique can be used in four-dimensional variational data assimilation to reduce the order of the adjoint model.

3.1 The Model

Modelling subsurface contaminant transport, the following two-dimensional advection-diffusion equation is considered:

$$\frac{\partial(\phi C)}{\partial t} + \nabla \cdot (UC - D(U)\nabla C) = r(C) + S, \tag{3.1}$$

where $U = (U_x, U_y)$ is the two-dimensional velocity field, ϕ is the porosity of the medium, C is the concentration of the contaminant, D the diffusion term, r the adsorption term and S the source term. For our purposes, we ignore the diffusion and adsorption terms as well as the source term, thereby reducing equation (3.1) to the form

$$\frac{\partial(\phi C)}{\partial t} + \frac{\partial}{\partial x}(U_x C) + \frac{\partial}{\partial y}(U_y C) = 0. \tag{3.2}$$

This is now a linear partial differential equation, which will allow easy assessment of the quality of our order reduction method.

3.1.1 Experiment set-up and POD suitability

A cell-centered finite difference scheme is used for discretization and the test domain is a 50×50 square grid, where a small rock with permeability 1 millidarcy is embedded into a surrounding layer with permeability 100 millidarcy. The cells in the subdomain

$[12, 12] \times [38, 38]$ belong to the small rock and the rest to the surrounding layer. The initial velocity field is obtained by solving the Darcy flow equation for incompressible flows and is assumed constant in time (i.e. the flow is steady). For the initial state, a contaminant plume is inserted in the subdomain $[4, 5] \times [6, 45]$ with concentration 100 ppm and 0 ppm is assumed elsewhere. The simulated time is 50 years with a time step of 2 months.

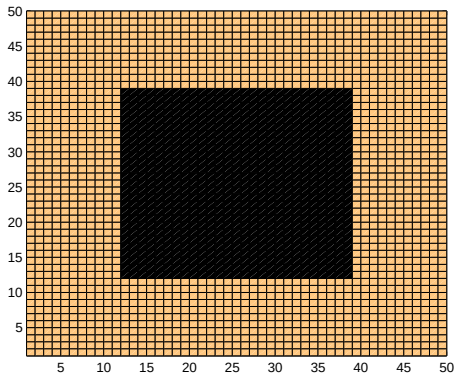


Figure 3.1: Test domain. Permeability is 1 millidarcy for black cells and 100 millidarcy for orange cells.

An indicator of the model being suitable for order reduction is the distribution of the singular values of the snapshot matrix E. Since the grid is of size 50×50, when taking all 305 time steps into account, the snapshot matrix has dimension 2500×305. Computing the eigenvalues of EE^T gives the energy plot that can be seen in Figure 3.2. The first 15 singular values already amount to 99.99% of the total energy, which indicates suitability for model order reduction.

Denoting with F the fraction of snapshots (out of the 305 available) used in the snapshot matrix and with N the number of singular vectors used, the projection matrix was constructed as described in the previous section for $F = 1\%, 3\%, 5\%, 8\%, 10\%, 15\%, 20\%, 25\%, 33\%, 50\%, 100\%$ and $N = 1, 5, 10, 15, 20, 25$. Each time the reduced order model was run with the same initial data as the full order model and the root mean square error (RMSE) was computed as a measure of the error introduced by the order reduction:

$$RMSE = \sqrt{\sum_{t=1}^{305} \sum_{i=1}^{2500} \frac{(y_t^i - \tilde{y}_t^i)^2}{2500 \cdot 305}}, \tag{3.3}$$

where y_t^i is the ith grid point value of time step t generated by the full order model and $\tilde{y}_t^i$ is the corresponding grid point value of the reduced order model.

Table 3.1 gives the computed RMSE values. Increasing both the number of snapshots and leading singular vectors increases accuracy of the reduced order model. However, as shown in the table, values as low as 10^{-5} can be achieved with as few as 20% of all

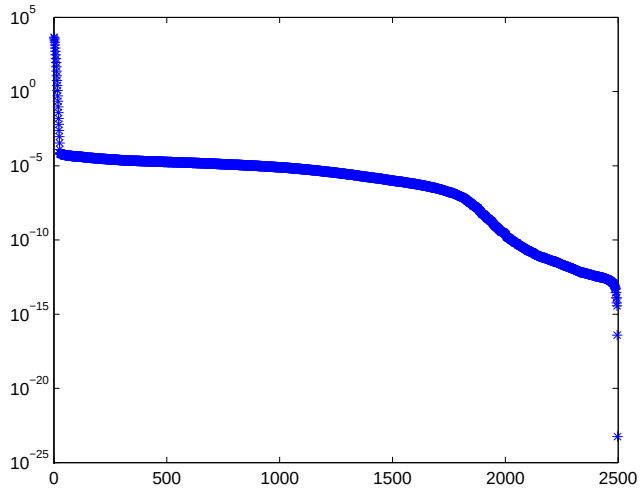


Figure 3.2: Plot of the singular values of the full snapshot matrix S.

RMSE	$N = 1$	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$
$F = 1\%$	7,57	6,52	6,46	6,42	6,41	6,41
$F = 3\%$	7,17	1,53	0,61	0,60	0,59	0,57
$F = 5\%$	6,63	1,39	0,14	0,073	0,067	0,065
$F = 8\%$	6,54	1,33	0,072	0,0073	0,0023	0,0016
$F = 10\%$	6,54	1,31	0,069	0,0037	0,00078	0,00037
$F = 15\%$	6,55	1,28	0,067	0,0018	0,00012	2,61e-05
$F = 20\%$	6,57	1,27	0,066	0,0016	3,53e-05	2,51e-06
$F = 25\%$	6,57	1,26	0,065	0,0016	2,39e-05	7,67e-07
$F = 33\%$	6,58	1,26	0,065	0,0015	2,03e-05	2,61e-07
$F = 50\%$	6,60	1,26	0,064	0,0015	1,98e-05	1,63e-07
$F = 100\%$	6,61	1,26	0,064	0,0015	1,96e-05	1,55e-07

Table 3.1: RMSE values for different reduced-order models.

snapshots and 20 leading singular vectors, Therefore it is safe to say that forward model order reduction can be applied successfully to this model.

Figures 3.3 and 3.4 show a visual comparison of the time evolution of the concentration field C over time. Figure 3.3 corresponds to the full order model, while Figure 3.4 shows the reduced order model with 15% of all snapshots and 15 leading singular vectors utilized. No visible differences can be observed.

3.2 Numerical experiments

As before, the ground water subsurface contaminant model with the same simplifications was considered. To test the application of a POD-reduced model in four dimensional variational data assimilation, a set of measurements (the truth) had to be chosen. This was done by taking the initial conditions described above and running the full model for 305 time steps, corresponding to a time of 50 years. Values from the solution of the final

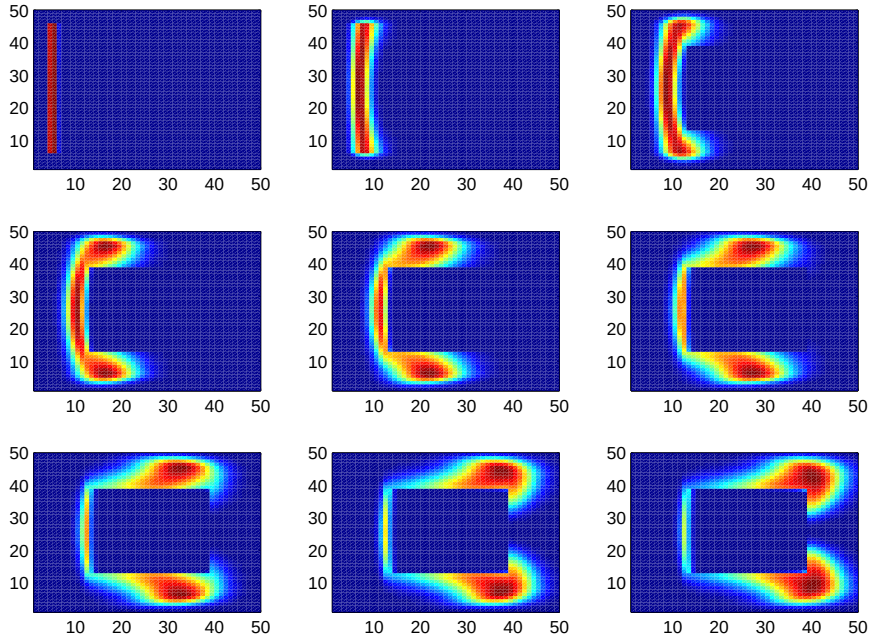


Figure 3.3: Evolution of the concentration field solution of the full order system over time. Evading the low permeability cells is clearly visible.

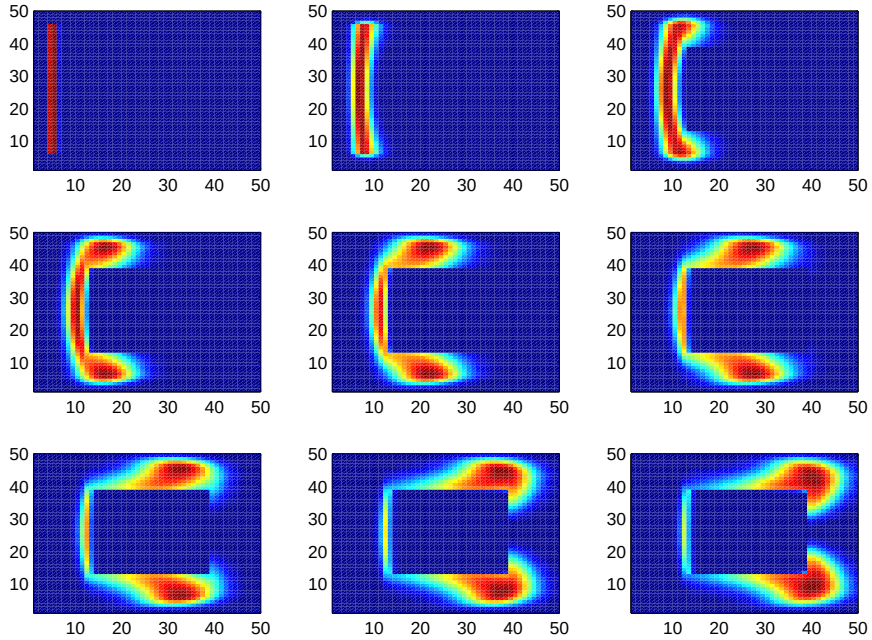


Figure 3.4: Evolution of the concentration field solution of the reduced order system with $F = 15\%$ and $N = 15$.

time step were taken in vertical columns spanning from the 6th to the 45th row at column numbers 10, 20, 30 and 40 in the 50-by-50 grid as shown in Figure 3.5.

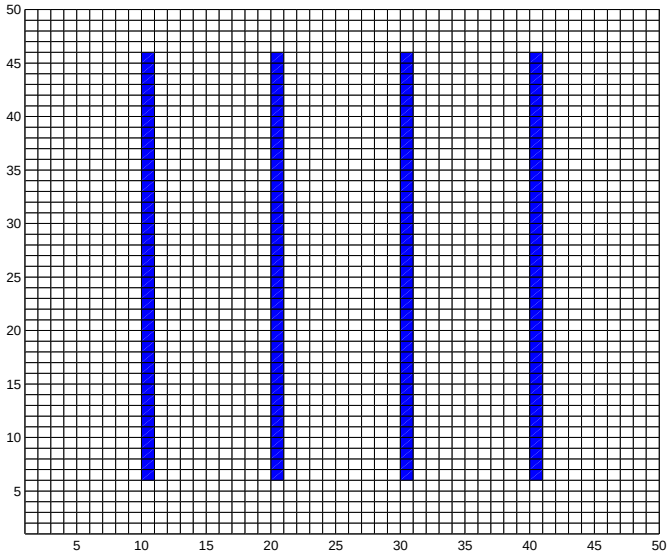


Figure 3.5: Observation points in the 50×50 domain.

From this known data in the final time step, the hope is to reconstruct the correct initial conditions (i.e. the vertical plume described earlier) by using the three different approaches: the full adjoint method, POD-reduced model variation and the POD-reduced adjoint method. The initial guess X_0 was the same for all three approaches and was taken as the average solution of the forward model over all time steps.

For the purposes of POD, 60 snapshots were taken with 5 time steps between them. In separate experiments, three sets of dominant singular vectors were included in the POD projection matrix P, namely 18 vectors capturing 99.999% of the total energy, 10 vectors capturing 99.99% of the total energy and 6 vectors capturing 99% of the total energy.

The reduced dynamical operator is not problematic in this case, since the model itself is linear and we can use

$$\tilde{M} = P^T M P$$

directly to compute the reduced dynamical operator $\tilde{M}$.

The optimization itself was carried out with a MATLAB implementation of a quasi Newton method with an L-BFGS update. The optimization procedure was stopped when the difference of two consecutive function values was less than 0.1. Considering that the initial cost function value is 2.95 million, this is certainly a small enough increment to stop

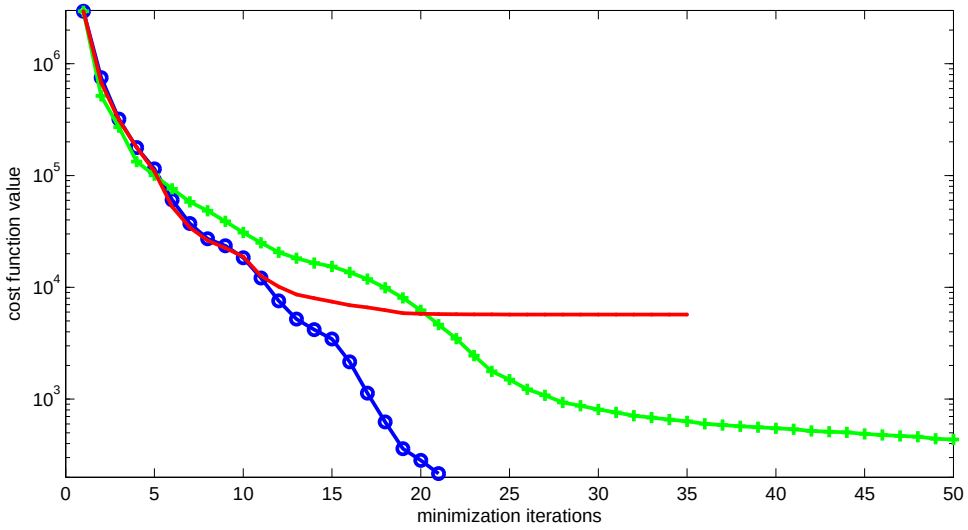
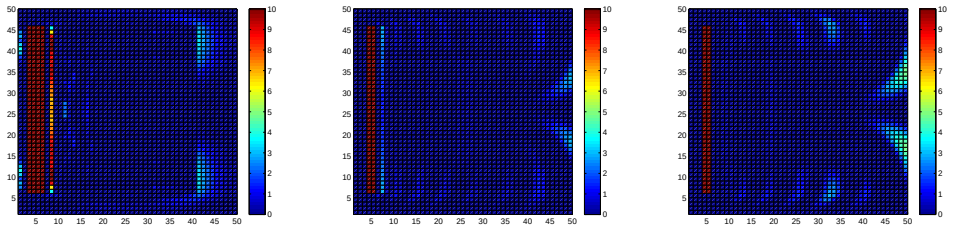


Figure 3.6: Cost function value over number of optimization iterations for full adjoint(green), POD-reduced adjoint(blue) and POD-reduced variation(red).

optimizing.

For the 18-vector POD basis, Figure 3.6 shows the cost function values over the number of iterations of the optimization process. Both POD-reduced methods show good convergence at first, but POD-reduced model variation starts to show a lack in accuracy after 10 iterations, eventually converging to a state with cost function value of about 5700. Keeping the original forward model space in the POD-reduced adjoint method, however, seems to have a great positive impact on accuracy, since the cost function decreases quickly (in 20 iterations) to a value of about 200. After that, the process terminates because no suitable step length can be found in the calculated gradient direction. Since the gradient calculation is approximate, this kind of termination is not unexpected.

The full adjoint method, somewhat surprisingly, shows slower convergence in addition to its higher computational complexity (which results from additional costly adjoint model runs backwards in time that are required to compute the gradient in each iteration). The cost function value is driven down to values as low as 10, but to achieve this, about 300 minimization iterations are required. In addition, the final solution, shown in figure 3.7, is not identical to the initial conditions of the forward simulation even though the cost function value is very low. A possible explanation to this is that the problem of minimizing the cost function J is ill-conditioned and/or solutions other than the one we expect exist [2]. Both POD-driven methods, however, result in the vertical plume with a width of two cells that we expected. The evolution of the approximation of X_0 can be seen in Figures 3.8, 3.9 and 3.10.



(a) Full adjoint (b) POD-reduced adjoint (c) POD-reduced variation

Figure 3.7: Final solutions of X_0 as produced by the three methods (number of iterations varies with each method).

No. of POD modes	POD-reduced variation	POD-reduced adjoint
18	5711.8	215.65
10	46232.3	13983.4
6	175891	327642

Table 3.2: Final cost function values for POD-reduced variation and POD-reduced adjoint methods depending on the number of POD modes in the POD basis.

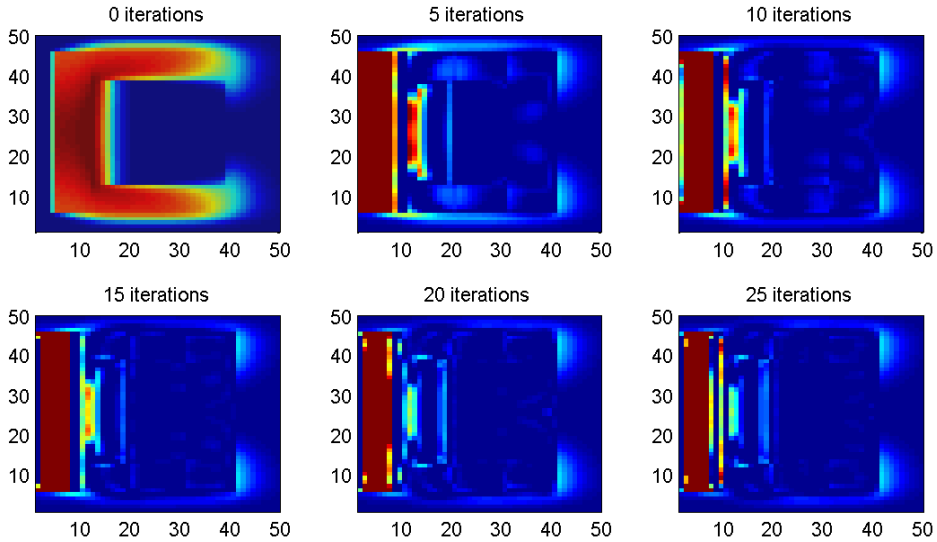


Figure 3.8: Initial state approximations by full adjoint method.

Using a lower-dimensional POD basis has a detrimental impact on the accuracy of the two POD-derived methods. The POD-reduced adjoint method exhibits the same behaviour as before, reducing the cost function value quickly, but at some point stopping because the line search in the computed gradient direction failed to find a suitable step size to continue. This point occurs sooner if the POD basis has lower dimension (after 13 iterations for 10 POD modes and after only 4 iterations for 6 POD modes). Table 3.2 shows the final cost function values for 10 and 6 POD modes.

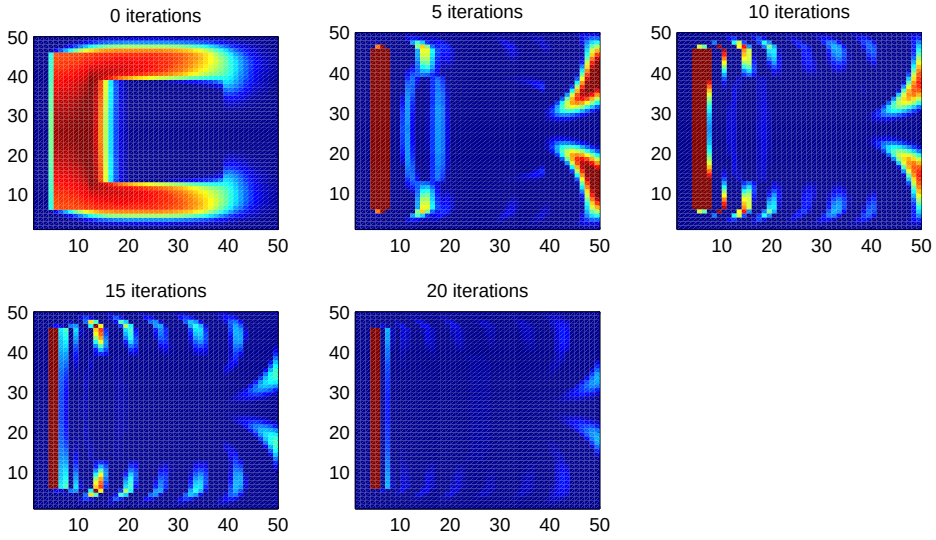


Figure 3.9: Initial state approximations by POD-reduced adjoint method.

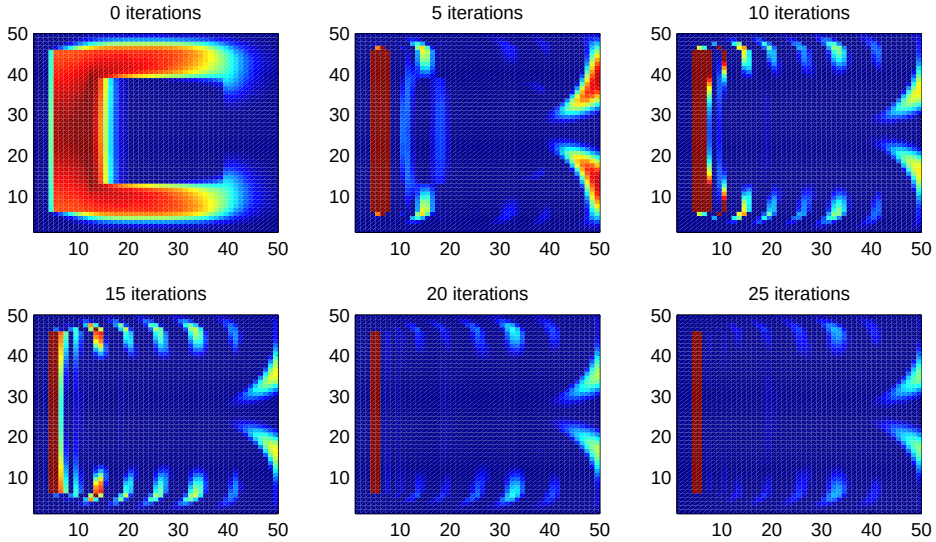


Figure 3.10: Initial state approximations by POD-reduced model variation.

Chapter 4

Shallow Water Equations

4.1 The Shallow Water Equations

The shallow water equations are a model of motion of incompressible fluids for which the depth is relatively small in comparison to the horizontal dimensions. The equations describe the evolution of an incompressible fluid on which gravitational and rotational accelerations are applied. Typically these conditions are associated with modelling an atmosphere.

By defining a vector function of velocity components u in x direction and v in y direction:

$$\vec{v} = (u(x, y, t), v(x, y, t)).$$

the shallow water equations can be written in vector form as

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} + \nabla \phi + f\vec{k} \times \vec{v} = 0 \tag{4.1}$$

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \vec{v}) = 0 \tag{4.2}$$

for

$$(x, y) \in [0, L] \times [0, D], t \geq 0,$$

where

- $[0, L] \times [0, D]$ is a rectangular domain of dimensions L and D,
- $\phi = gh$ is the geopotential height
- h is the depth of the fluid
- g is the gravitational acceleration
- $\vec{k}$ is the vertical unit vector pointing away from the center of the Earth
- f is the Coriolis parameter, i.e. a scalar function defined by the β-plane approximation as outlined in [11].

Initial and boundary conditions need to be specified for the shallow water equations model. Denoting the state variables as

$$w(x, y, t) = (u(x, y, t), v(x, y, t), \phi(x, y, t)) \tag{4.3}$$

we prescribe the initial condition $w(x, y, 0) = \psi(x, y)$.

Boundary conditions in x-direction are assumed periodic:

$$w(x + kL, y, t) = w(x, y, t) \quad , k \in \mathbb{Z} \tag{4.4}$$

and a solid wall boundary condition is employed in the y-direction:

$$\vec{v}(x, 0, t) = \vec{v}(x, D, t) = 0. \tag{4.5}$$

The initial geopotential $\phi(x, y) = gh(x, y)$ is derived from the same height field as in [11]:

$$h(x, y) = H_0 + H_1 \tanh(\frac{9(D/2 - y)}{2D}) + H_2 \frac{\sin(\frac{2\pi x}{L})}{\cosh^2(\frac{9(D/2)-y}{D})}. \tag{4.6}$$

From the geopotential, the initial wind field is calculated by using the geostropic relationship

$$u = -\left(\frac{g}{f}\right)\frac{\partial h}{\partial y}, \quad v = \left(\frac{g}{f}\right)\frac{\partial h}{\partial x} \tag{4.7}$$

Keeping to [11], we use the following values for the parameters needed to calculate the height field:

$$\begin{aligned} L &= 4400\,\text{km}, \quad D = 6000\,\text{km} \\ \beta &= 1.5 \times 10^{-11}\,\text{s}^{-1}\text{m}^{-1} \\ f &= 10^{-4}\,\text{s}^{-1}, \quad g = 10\,\text{ms}^{-1} \\ H_0 &= 2000\,\text{m}, \quad H_1 = 220\,\text{m}, \quad H_2 = 133\,\text{m}. \end{aligned}$$

The corresponding initial geopotential field on a grid of size 15×15 can be seen on figure 4.1. The wind vector field, calculated from the height field for this configuration, is shown in figure 4.2. Recovering these initial conditions will be the goal of the optimization procedure later on.

4.2 Discretization

In the implementation by Navon and Chen [11], a Galerkin finite element approach is used to discretize the shallow water equations (4.1)-(4.2) using a regular triangular mesh.

Denote with V_j the basis functions defined by node interpolation. Then the wind and

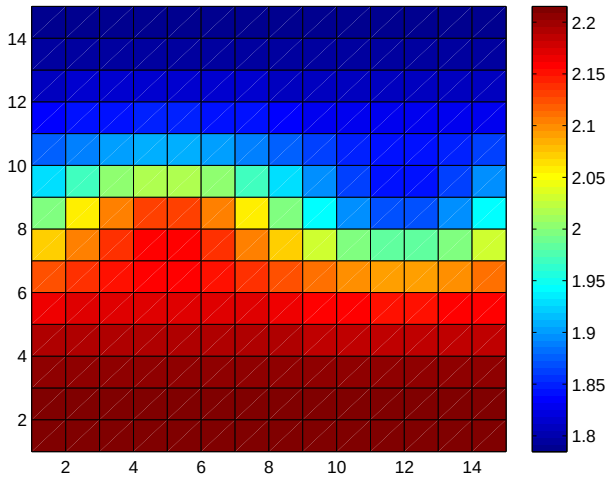


Figure 4.1: Initial geopotential field on a 15-by-15 grid.

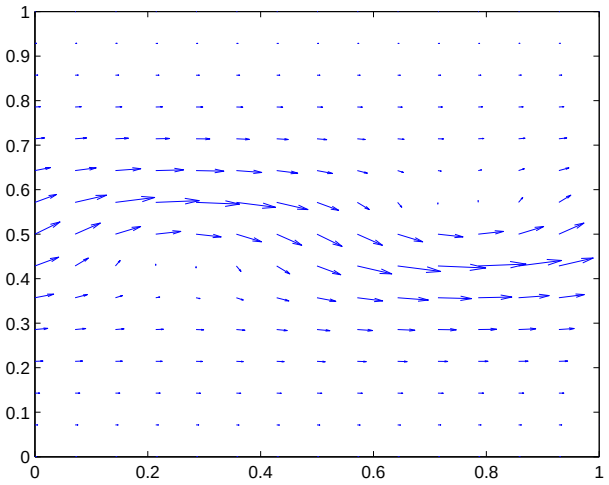


Figure 4.2: Initial wind field on a 15-by-15 grid.

geopotential fields are given by:

$$\vec{v} = \sum_{j=1}^{3} \vec{v}_j(t)\, V_j(x,y)$$
$$\phi = \sum_{j=1}^{3} \phi_j(t)\, V_j(x,y) \tag{4.8}$$

After applying Green's theorem to shift the derivative from the variable to the basis function in (4.1) and (4.2), substituting in (4.8) and finally employing a time-extrapolated Crank-Nicholson scheme for integration in time, we arrive at the following linear systems to be solved (for details see [11]):

First, to obtain ϕ at time $n+1$, solve:

$$\mathbf{M}(\phi_j^{n+1} - \phi_j^n) - \frac{\Delta t}{2}\mathbf{K}_1(\phi_j^{n+1} + \phi_j^n) = 0 \tag{4.9}$$

where

$$\mathbf{M} = \iint_{ele} V_i V_j dA \tag{4.10}$$

and

$$\mathbf{K}_1 = \iint_{ele} V_j V_k u_k^* \frac{\partial V_i}{\partial x} dA + \iint_{ele} V_j V_k v_k^* \frac{\partial V_i}{\partial y} dA. \tag{4.11}$$

Here, u^* and v^* denote approximations of the state variables at time $n+\frac{1}{2}$, for which we use the second order formula

$$w^* = \frac{3}{2}w^n - \frac{1}{2}w^{n-1} + o(\Delta t^2). \tag{4.12}$$

Second, from the u-momentum equation, we get

$$\begin{aligned}\mathbf{M}(u_j^{n+1} - u_j^n) + \frac{\Delta t}{2}\mathbf{K}_2(u_j^{n+1} + u_j^n) \\ + \frac{\Delta t}{2}(\mathbf{K}_{21}^{n+1} + \mathbf{K}_{21}^n) + \Delta t \mathbf{P}_2 = 0\end{aligned} \tag{4.13}$$

where

$$\mathbf{K}_2 = \iint_{ele} V_i V_k u_k^n \frac{\partial V_j}{\partial x} dA + \iint_{ele} V_i V_k v_k^* \frac{\partial V_j}{\partial y} dA, \tag{4.14}$$

$$\mathbf{K}_{21}^n = \iint_{ele} V_i \phi_k^n \frac{\partial V_k}{\partial x} dA \tag{4.15}$$

and

$$\mathbf{P}_2 = -\iint_{ele} f V_i V_k v_k^* dA. \tag{4.16}$$

Note that, since ϕ^{n+1} has already been calculated, we use the available value instead of the extrapolated one.

Finally, from the v-momentum equation, we get:

$$\mathbf{M}(v_j^{n+1} - v_j^n) + \frac{\Delta t}{2}\mathbf{K}_3(u_j^{v+1} + v_j^n) \tag{4.17}$$

$$+ \frac{\Delta t}{2}(\mathbf{K}_{31}^{n+1} + \mathbf{K}_{31}^n) + \Delta t \mathbf{P}_3 = 0 \tag{4.18}$$

where

$$\mathbf{K}_3 = \iint_{ele} V_i V_k u_k^{n+1} \frac{\partial V_j}{\partial x} dA + \iint_{ele} V_i V_k v_k^n \frac{\partial V_j}{\partial y} dA, \tag{4.19}$$

$$\mathbf{K}_{31}^n = \iint_{ele} V_i \phi_k^n \frac{\partial V_k}{\partial x} dA \tag{4.20}$$

and

$$\mathbf{P}_3 = \iint_{ele} f V_i V_k u_k^{n+1} dA. \tag{4.21}$$

Solutions of the systems are obtained by a Gauss-Seidel iterative solver, taking into account that due to the local support nature of the finite element structure, only at most 7 non-zero entries are possible in each row of the matrices, allowing for compacting the storage and optimizing the iteration process.

4.3 The adjoint models

As in the linear case, we consider three different approaches to inverse modelling:

- POD-reduced model variation, using reduced models for both forward and adjoint simulation,
- POD-reduced adjoint method, using the full forward model but the reduced adjoint one,
- full adjoint method, using the full adjoint model provided in [11].

An extra difficulty arises from the non-linearity of this model, however, since the reduced dynamical operator $\tilde{M}_i$ cannot be computed directly, as it could be in the subsurface contaminant transport case. Instead, linearization is required.

4.3.1 Computing the reduced dynamical operator

A model can be reduced (see [1]) if its approximate linearised state $\bar{X}$ can be written as

$$\bar{X}(t_{i+1}) = X^b(t_{i+1}) + P\xi(t_{i+1}) \tag{4.22}$$

where ξ is the reduced state vector, whose time-varying dynamics are described by the reduced dynamical operator $\tilde{M}_i$ as

$$\xi(t_{i+1}) = \tilde{M}_i\xi(t_i). \tag{4.23}$$

and X^b is the background state, which in this case is taken as the average state of all time steps, i.e.

$$X^b = \frac{\sum_{i=1}^{t_{max}} X(t_i)}{t_{max}} \tag{4.24}$$

The reduced dynamical operator is an approximation of the full Jacobian $\frac{\partial M_i}{\partial X^b}$:

$$\tilde{M}_i = P^T \frac{\partial M_i}{\partial X^b} P. \tag{4.25}$$

We obtain the Jacobian by linearising the full non-linear operator M_i around the background state X^b. However, in order to avoid computing the perturbations $M_i(X^b(t_i)+\varepsilon E_j)$ for every j-th unit vector E_j, in turn having to do this as many times as there are nodes in the mesh, only the perturbations in the direction of POD modes p_j are computed by

$$\frac{\partial M_i}{\partial X^b} p_j = \frac{M_i(X^b(t_i) + \varepsilon p_j) - M_i(X^b(t_i))}{\varepsilon}. \tag{4.26}$$

This way we get for reduced state space of dimension r

$$\frac{\partial M_i}{\partial X^b} P = (\frac{\partial M_i}{\partial X^b} p_1, ..., \frac{\partial M_i}{\partial X^b} p_r) \tag{4.27}$$

and finally, to obtain the reduced dynamical operator $\tilde{M}_i$, we pre-multiply by P^T

$$\tilde{M}_i = P^T(\frac{\partial M_i}{\partial X^b} p_1, ..., \frac{\partial M_i}{\partial X^b} p_r). \tag{4.28}$$

4.4 Numerical experiments

4.4.1 POD-reduced Model Variation

Set-up

In order to maximize speed of computations, we first attempt to apply POD-based order reduction to both the forward and the adjoint directions of the simulation.

To obtain observations (the "truth"), the full forward model is run for a timespan of 30.000 seconds with the initial geopotential and wind fields described in Section 4.1. In two different set-ups, the time step is 300 seconds for one and 600 seconds for the other, meaning 100 and 50 time steps, respectively. To get the POD basis vectors, a snapshot matrix is constructed with state vectors from all time steps included and a singular value decomposition is performed. The projection matrix P is composed of 8 dominant singular vectors, which correspond to more than $99,9999\%$ total energy (see Figure 4.3).

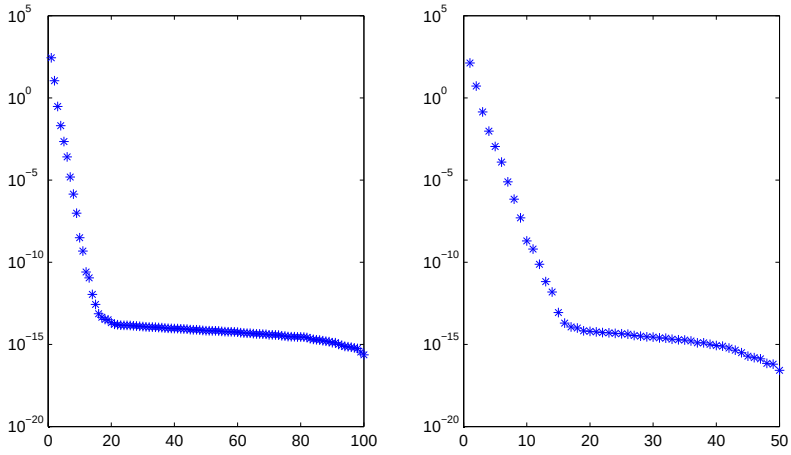


Figure 4.3: Distribution of singular values for time steps of 300 seconds (left) and 600 seconds (right).

From there, the reduced dynamical operator $\tilde{M}_i$ is constructed by perturbing the non-linear full model operator M_i (which in this case is the procedure that advances the current state by one time step) in the directions of the dominant POD modes. Once this is available, the adjoint model can easily be constructed, since we only need to transpose the operator $\tilde{M}_i$. Also, we assume the reduced dynamical operator to be constant for all time steps, so the index i can be removed: $\tilde{M}_i \equiv \tilde{M}$.

Because the optimization using L-BFGS updates runs in reduced space for this method and we are not using any background terms, the appropriate version of the cost function

is (see equation (2.17))

$$\hat{J}(\xi_0) = \frac{1}{2}\sum_{i=0}^{m}(Y(t_i) - H(\hat{X}(t_i)))^T R_i^{-1}(Y(t_i) - H(\hat{X}(t_i))),$$

with

$$\hat{X}(t_i) = P\xi(t_i) + X^b$$

and the gradient of the approximate cost function $\hat{J}$ is computed as

$$\nabla\hat{J}(\xi_0) = -\hat{\nu}(t_0), \tag{4.29}$$

where the adjoint state evolves in time by

$$\hat{\nu}(t_i) = \tilde{M}^T\hat{\nu}(t_{i+1}) + P^T H^T R_i^{-1}(Y(t_i) - H\hat{X}(t_i)). \tag{4.30}$$

Results

If the POD-reduced model variation method worked reasonably well with the subsurface contaminant transport model, the same cannot be said for the shallow water model.

In the smaller time step case, the initial reduced cost function value is 1259 and is lowered in five iterations down to 1184, which is a reduction of only 6%. The full space cost function was also calculated by projecting the reduced solution back into the original state space. Its value is 1755 before and 1751 after optimization, which is a 0.2% decrease, that is hardly any decrease at all.

For the larger time step, the initial cost function values are 598.9 in the reduced space and 863.6 in the full space before optimization. Afterwards, they are 566.8 and 835.6, which means a 7% and 3.2% decrease, respectively.

Optimization terminates in both experiments after five iterations due to line search not finding a suitable point in the computed direction. Most likely this is due to the additional error in gradient approximation, introduced by the need to compute the reduced dynamical operator by linearization. Figure 4.4 shows the cost function value progressions for both time step settings. A confirmation of the optimization being bound to terminate early is also the infinity norm of the gradient, which goes from 71 to 0.003 for the smaller time step and from 34.2 to 0.001 for the larger time step. The gradient norm values are shown in Figure 4.5. Visual feedback of the final approximate of the initial conditions being rather far from the truth can be observed in Figure 4.6, which shows the geopotential state after optimization for the 300 second and 600 second time step.

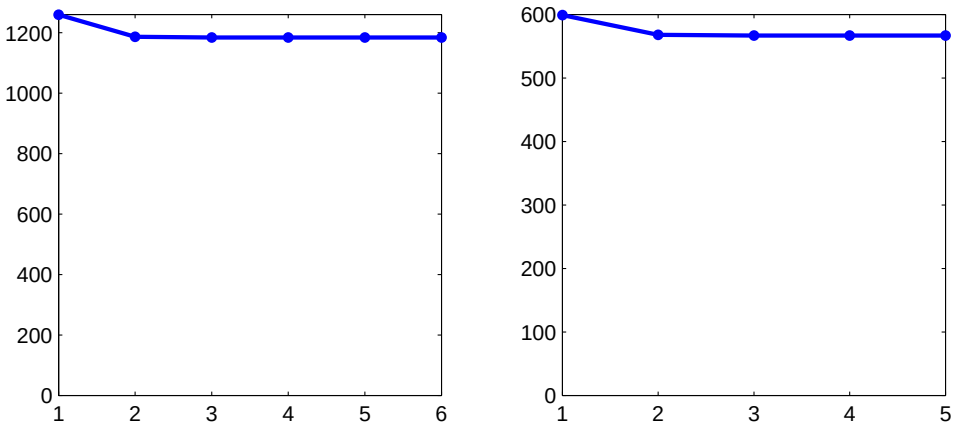


Figure 4.4: Cost function values by optimization iteration for time steps of 300 seconds (left) and 600 seconds (right).

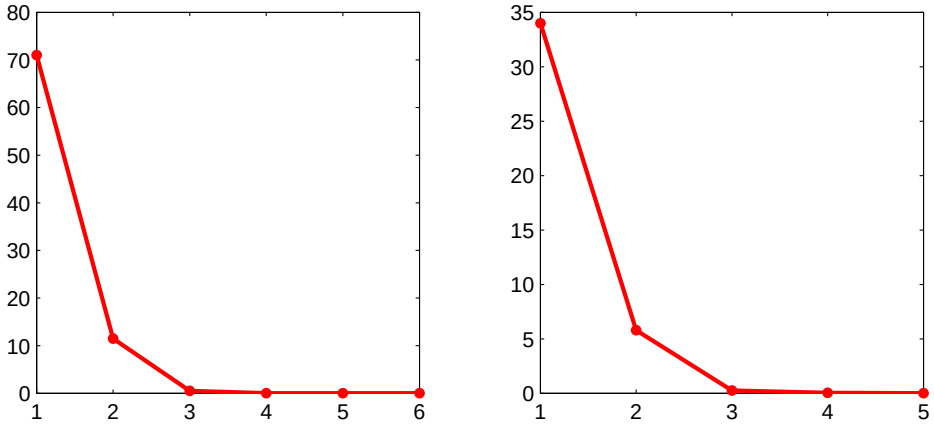


Figure 4.5: Gradient norm by optimization iteration for time steps of 300 seconds (left) and 600 seconds (right).

4.4.2 POD-reduced Adjoint Method

Set-up

Since the order reduction in both forward and adjoint models did not deliver satisfactory results, we attempt to utilize the POD-reduced adjoint method, suggested in [2] and employed already in chapter 3 of this thesis. In this method, the forward model is used to evaluate the cost function

$$J(X_0) = \frac{1}{2}\sum_{i=0}^{m}(Y(t_i) - H(X(t_i)))^T R_i^{-1}(Y(t_i) - H(X(t_i))) \tag{4.31}$$

which means that the optimization is performed in the full state space, as opposed to the POD-reduced model variation, where we optimized in reduced space.

The adjoint model, however is still constructed as before by linearization around the

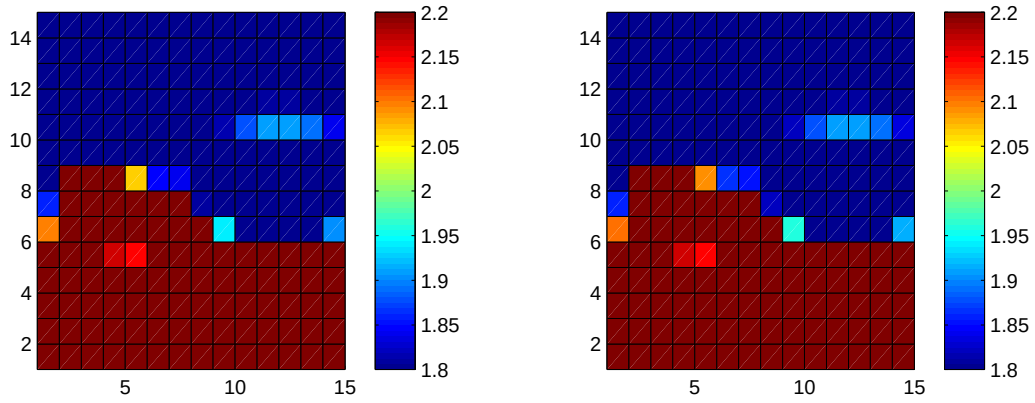


Figure 4.6: Final approximate of initial geopotential field for 300 second time step (left) and 600 second time step (right).

background state, obtaining the reduced dynamical operator $\tilde{M}$ and using

$$\hat{\nu}(t_i) = \tilde{M}^T \hat{\nu}(t_{i+1}) + P^T H^T R_i^{-1}(Y(t_i) - HX(t_i)) \tag{4.32}$$

to calculate the gradient of the cost function via

$$\nabla J(X_0) = -P\hat{\nu}(t_0). \tag{4.33}$$

Results

Performing the optimization in the full state space improves the performance drastically over the POD-reduced model variation method for both versions of the experiment.

For the 300 second time step case, the initial cost function value is 1755 and drops down to 17.6 in four iterations for a decrease of two orders of magnitude or more precisely, 99.0%. After that, no more decrease can be found along the computed direction. Table 4.1 shows the cost function values and the gradient norms for this case.

Iteration	cost value	gradient norm
0	1755	139.0
1	133.7	33.7
2	19.40	4.37
3	17.62	0.31
4	17.57	0.39

Table 4.1: Cost function values and gradient norms for Reduced Adjoint method with 300 second time step.

For the 600 second time step case, the initial cost function value is 863.7 and decreases similarly rapidly to 8.43 in four iterations. The relative decrease is again 99.0%. In the fifth iteration, no further decrease can once again be found. Table 4.2 shows the corresponding values for the cost function and the gradient norm for this experiment.

Figures 4.7 and 4.8 show the values from tables 4.1 and 4.2 graphically while the evolution of the initial geopotential field can be seen in figures 4.9 and 4.10. Compared to Figure 4.6, the final approximations are visibly closer to the correct initial conditions as depicted in Figure 4.1. It is thus shown that using the full forward model to evaluate the cost function is very beneficial as opposed to using the reduced model in both directions.

Iteration	cost value	gradient norm
0	863.7	68.5
1	61.85	16.0
2	9.18	1.92
3	8.45	0.13
4	8.43	0.16

Table 4.2: Cost function values and gradient norms for Reduced Adjoint method with 600 second time step.

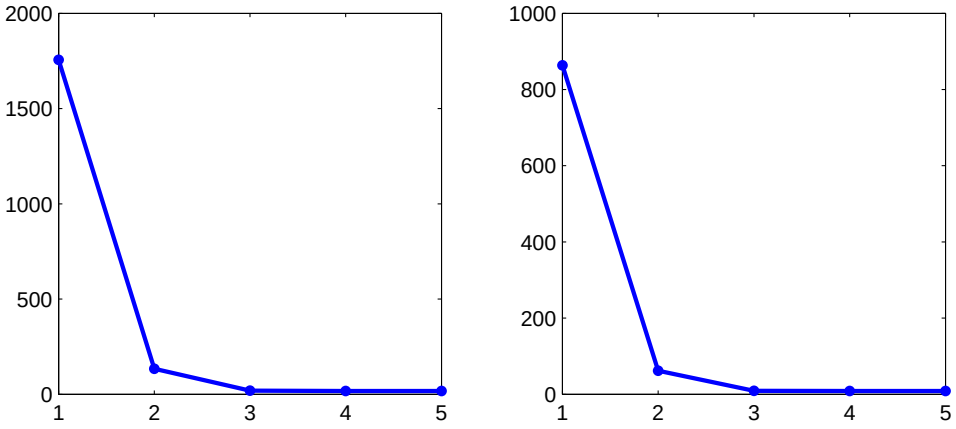


Figure 4.7: Cost function values by optimization iteration for time steps of 300 seconds (left) and 600 seconds (right).

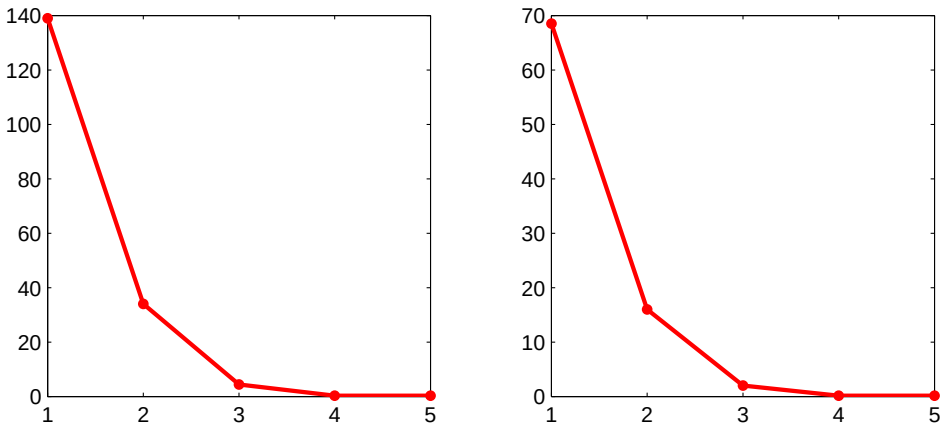


Figure 4.8: Gradient norm by optimization iteration for time steps of 300 seconds (left) and 600 seconds (right).

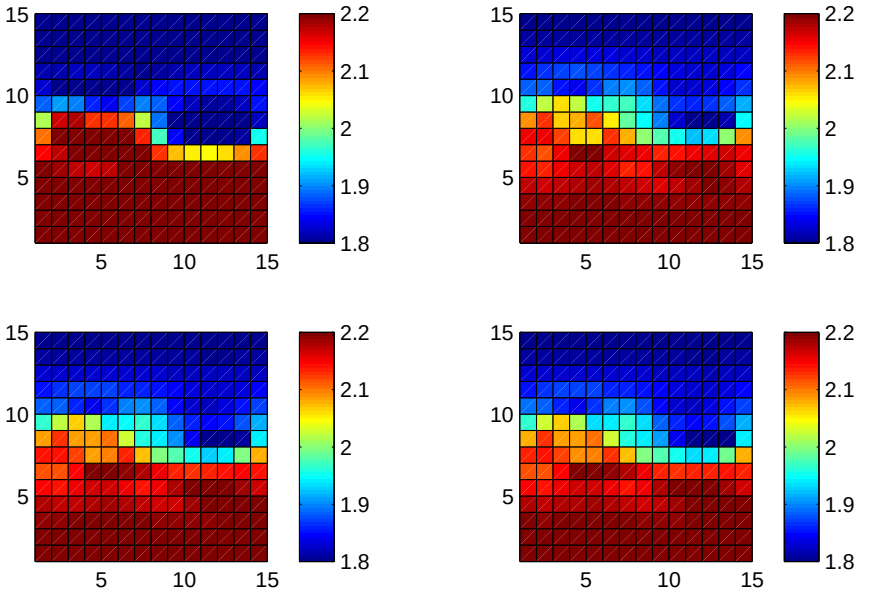


Figure 4.9: Initial geopotential field evolution for 300 second time step.

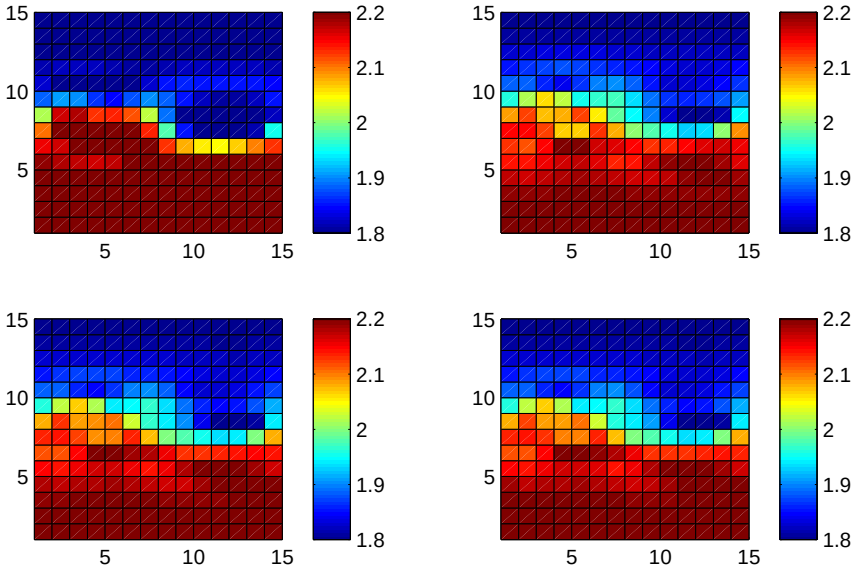


Figure 4.10: Initial geopotential field evolution for 600 second time step.

4.4.3 Comparison with Full Adjoint Method

In [11], a full order adjoint model was programmed by X. Chen and I. M. Navon with the purpose of calculating optimal control for the shallow water equations. We compare the performance of this full adjoint to the reduced order one obtained by POD in this thesis.

Contrary to the reduced adjoint method, the full order version never terminates because no further decrease can be found. It reaches the set minimal cost function value of 10^{-6} in 8 iterations. This is of course possible due to the more accurate gradient computation, giving better search directions in each step of the optimization process. The price to pay for increased accuracy, though, is higher computational cost.

Figure 4.11 shows the comparison between the two methods per iteration for the first four iterations (until a decrease of two orders of magnitude is reached). It seems that the two adjoint methods exhibit similar behaviour until at some point the reduced adjoint terminates. However, this iteration-based view does not give the full story, as we can see if we set the number of function evaluations as the horizontal axis in place of the number of iterations. Every cost function evaluation is a forward model run and this corresponds roughly to the time spent in optimization (not completely so, because some function calls include gradient computations and some do not). This comparison is shown in Figure 4.12 and it is clear that, until it terminates, the reduced adjoint method is considerably faster, using in this case only 10 cost function evaluations to reduce it by two orders of magnitude, while the full adjoint takes 22 evaluations, more than twice as many, to achieve about the same result. In addition, gradient computations in the reduced adjoint method are performed in reduced order space, which makes them faster than their full order counterparts as well.

The conclusion from this experiment is that the reduced adjoint method performs very well and computationally more efficiently than the adjoint method. Furthermore, it avoids the difficulties of actually programming the full order adjoint model.

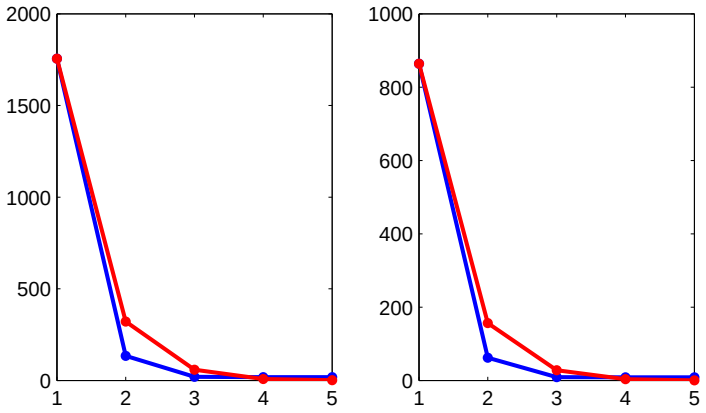


Figure 4.11: Cost value by iteration for full (red) and reduced (blue) adjoint. 600 second time step left, 300 second time step right.

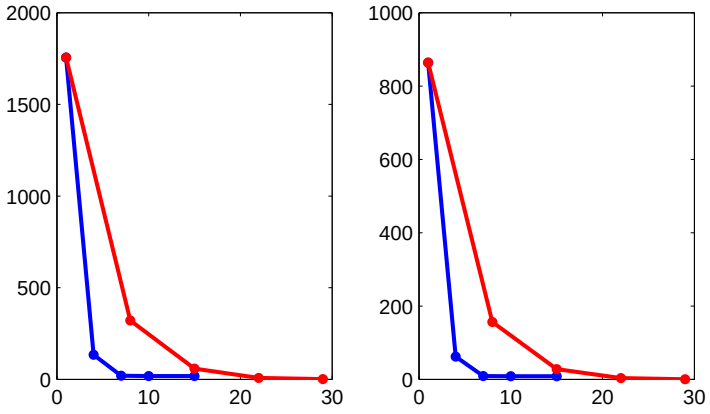


Figure 4.12: Cost value by forward model runs for full (red) and reduced (blue) adjoint. 600 second time step left, 300 second time step right.

4.4.4 Perturbing Observations

Throughout the experiments so far we have assumed the observations (our measured "truth") to be perfect, meaning that they were exactly the values calculated by the model in a separate run. In practical applications, however, observations are collected by measurements and those are prone to measurement errors. Therefore, it is sensible to investigate sensitivity of the data variation method, specifically the reduced adjoint variant, to these errors. The full adjoint method need not be tested because its gradient computation is independent of the observed truth.

Set-up

To simulate the situation of measurements with errors, perturbations were induced in the process of obtaining the truth. Each entry in the observations set was multiplied with a random number between $1+\varepsilon$ and $1-\varepsilon$, corresponding to measuring with a relative error of ε. The reduced adjoint method was used and the rest of the optimization procedure remained unchanged. The experiment was done with ε values of 1%, 3%, 5%, 10%, 20%, 30% and 50%. The grid size remained as before at 15-by-15 in this and the following section.

One obvious effect of perturbing observation we expect is a change in the initial cost function value, since it evaluates the differences between the model results and the measured truth. If we were using the full adjoint method, this would also be the only effect of the perturbations. The reduced adjoint method, however, also includes the truth values in the calculation of the approximate gradient. This means that the results are possibly further affected by perturbing observations.

Results

As expected, the amount of perturbation to the observations has a significant impact on the cost function values due to the larger deviations between the initial guess and the now slightly skewed truth, which are accumulated in the cost function. For the smaller time step case the initial cost goes from 1755 without perturbation up to as high as 9205 for the 50% perturbing factor. For the larger time step, the initial cost without perturbation is 863.7 and is increased up to as far as 4619 in the extreme case.

The optimization procedure lowers these values as shown in Table 4.3 for the 300 second time step and Table 4.4 for the 600 second time step. The final cost values are clearly very much impacted by the inaccurate observations. Figures 4.13 and 4.14 show this graphically for all the different perturbation factors.

Iteration	0	1	2	3
1% perturbation	1760	137.3	22.32	20.55
3% perturbation	1779	159.4	45.9	44.1
5% perturbation	1839	208.3	92.56	90.73
10% perturbation	2051	432.8	313.9	312.7
20% perturbation	2919	1319	1206	1204
30% perturbation	4383	2773	2675	2673
50% perturbation	9205	7547	7458	7456

Table 4.3: Cost function values per iteration for 300 second time step with different observation perturbation magnitudes.

Iteration	0	1	2	3
1% perturbation	864.6	63.22	10.62	9.88
3% perturbation	875.8	74.53	22.86	22.12
5% perturbation	898.8	97.52	45.25	44.53
10% perturbation	1018	208.4	154.4	153.7
20% perturbation	1459	657.4	601.2	600.6
30% perturbation	2178	1384	1338	1337
50% perturbation	4619	3790	3721	3720

Table 4.4: Cost function values per iteration for 600 second time step with different observation perturbation magnitudes.

However, this is not necessarily bad news. Inaccurate observations add to the penalty terms of the cost function, thus we cannot expect to converge to a zero cost. Finding its minimum, however, perhaps still leads to the correct solution. To check this, the final approximations for each perturbation factor were evaluated again, but this time against the unperturbed truth. Results are shown in Figure 4.15 and actual cost function values are shown to be, in fact, quite low. Significant deviation from the minimal value is not visible with perturbations up to 10%.

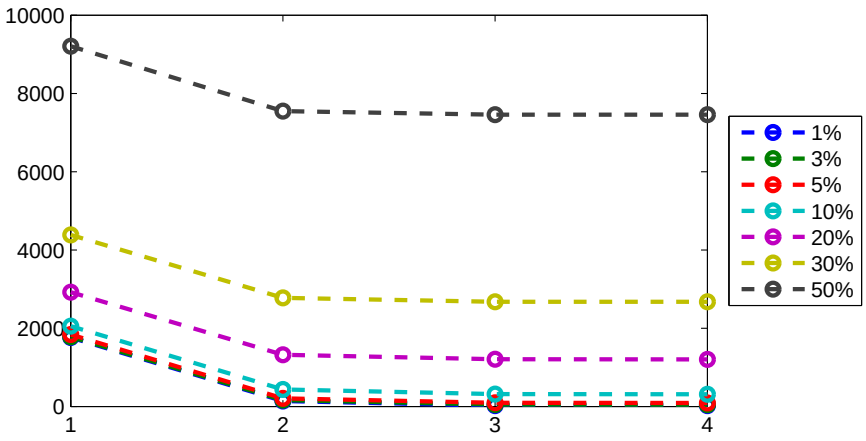


Figure 4.13: Cost function values per iteration for 300 second time step with different observation perturbation magnitudes.

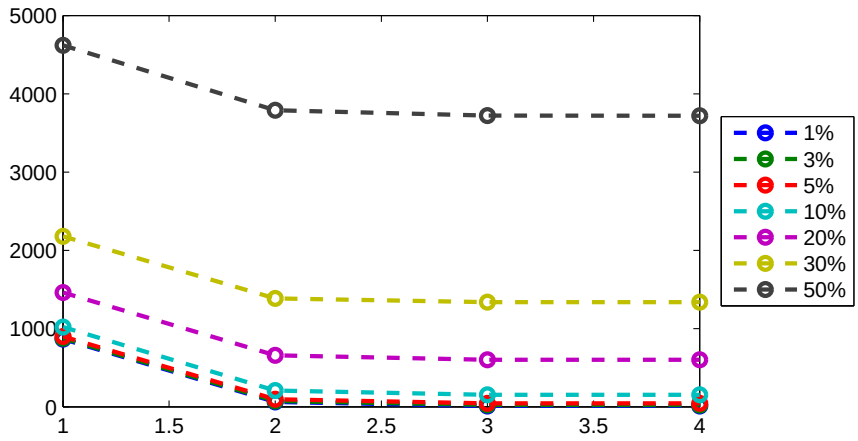


Figure 4.14: Cost function values per iteration for 600 second time step with different observation perturbation magnitudes.

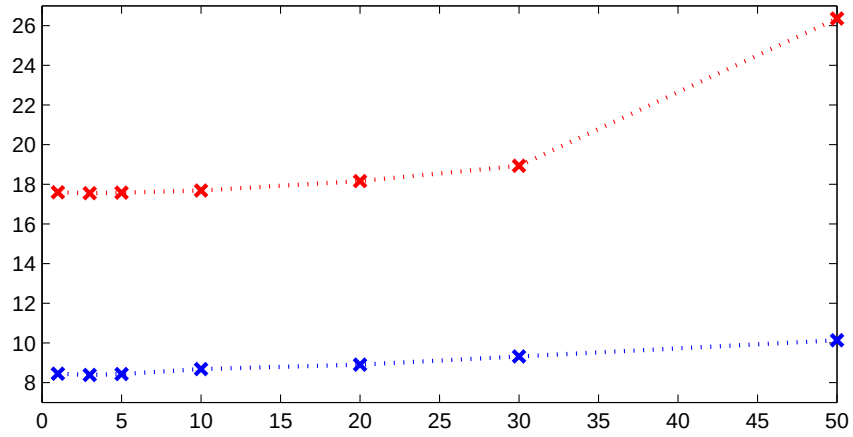


Figure 4.15: Final corrected cost function values for increasing perturbation magnitude from left to right. 300 second time step red, 600 second time step blue.

4.4.5 Using Fewer Observation Points

Another assumption that has been made in the first experiments was that every point in the discretization grid was included in the set of observations. In practice this would most likely not be the case, which is why in the following experiment the optimization procedure is done with smaller sets of observed grid points.

Because the cost function J reads as

$$J(X_0) = \frac{1}{2}\sum_{i=0}^{m}(Y(t_i) - H(X(t_i)))^T R_i^{-1}(Y(t_i) - H(X(t_i))), \tag{4.34}$$

which includes the observation operator H, the cost of the same approximate X_0 varies with the choice of the points included in observations. The fewer the points being observed, the lower the cost function value is likely to be. Whether and how this affects convergence to the correct minimum is what we want to see.

Set-Up

As before, two time step scenarios were kept, i.e. the 300 second and 600 second time step version of the model. Sets of observed points were generated such that the ratio of observed points to all points was 25%, 17%, 14%, 12%, 10%, 5% and 2%. The points observed were spaced equally through the grid to avoid areas starved of observation. An example is shown in Figure 4.16 for the 25% observation ratio. The data variation procedure was done with all these sets in the same manner as in previous experiments.

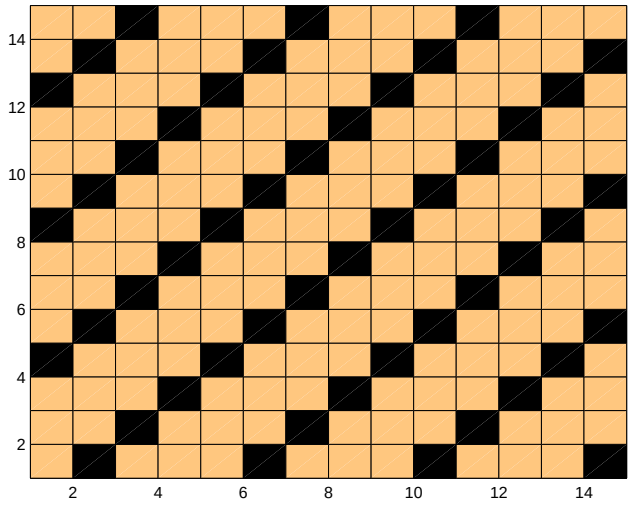


Figure 4.16: Geopotential field points included in observation set for the 25% observed ratio. Black points are observation points.

Results

Reductions in the amount of observed points do not have a drastic effect on the data variation procedure in terms of hindering convergence. In all experiments, the cost function was decreased by at least two orders of magnitude, but the individual values differed between observation point sets because of the differences in the cost function calculation described in the beginning of this experiment. Generally, the initial cost function was lower for fewer points being observed, but sometimes, presumably because of some critical high magnitude points being observed, this was also not the case. Figure 4.17 shows the cost function values by iteration for the 300 second time step and Figure 4.18 the same for the 600 second time step.

As in the perturbed observation experiments, because of the different cost function calculations, comparison between results is not possible directly. Instead, the end approximations are re-evaluated using the original cost function which includes all grid points in the observation set. These values then serve to assess the quality of the final approximates.

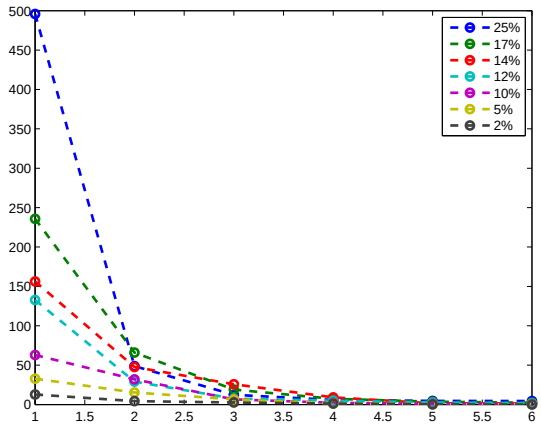


Figure 4.17: Cost function values per iteration for 300 second time step with different observation point ratios.

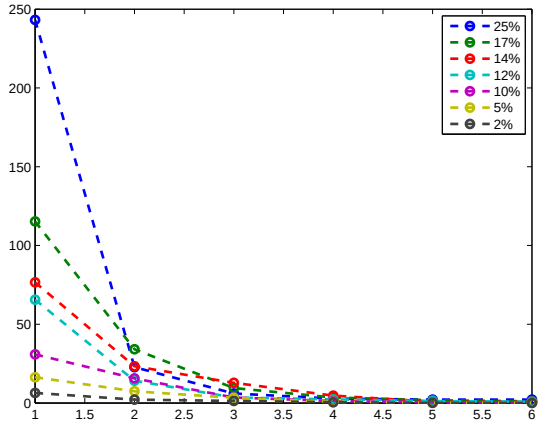


Figure 4.18: Cost function values per iteration for 600 second time step with different observation point ratios.

Figure 4.19 depicts the corrected cost values for approximates obtained with different observation sets. Significant deteriorations in the final cost do not appear for observation ratios above 15% for both time steps, which indicates a high level of robustness of the methods used.

Finally, Figures 4.20 and 4.21 show approximates of the geopotential field obtained with the observation ratios of 25%, 17%, 12%, 10%, 5% and 2%. A visual confirmation of what the cost function values indicated can be seen, since artefacts in the image start to become visible once the observation ratio falls below 15%.

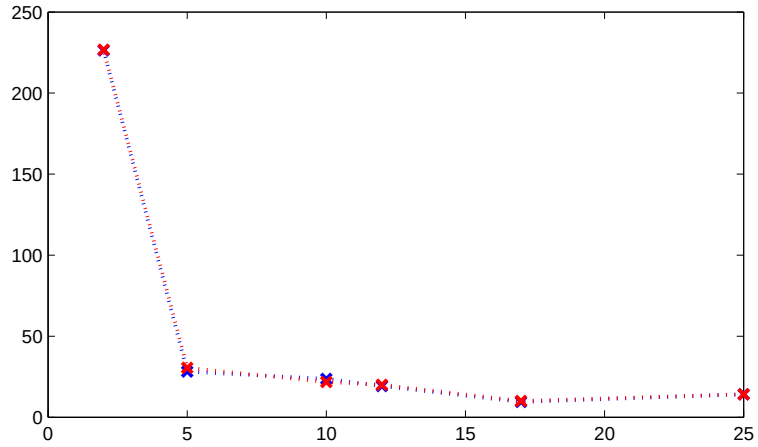


Figure 4.19: End cost function values for 300 second time step in red and 600 second time step in blue. Horizontal axis marks per cent ratio of observed grid points.

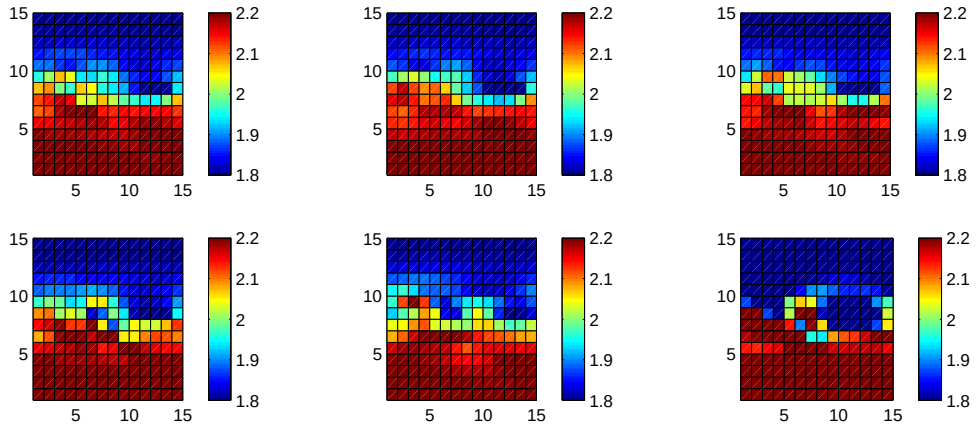


Figure 4.20: Geopotential field approximations for 300 second time step with 25%, 17% and 12% points observed in top row and 10%, 5% and 2% points observed in bottom row.

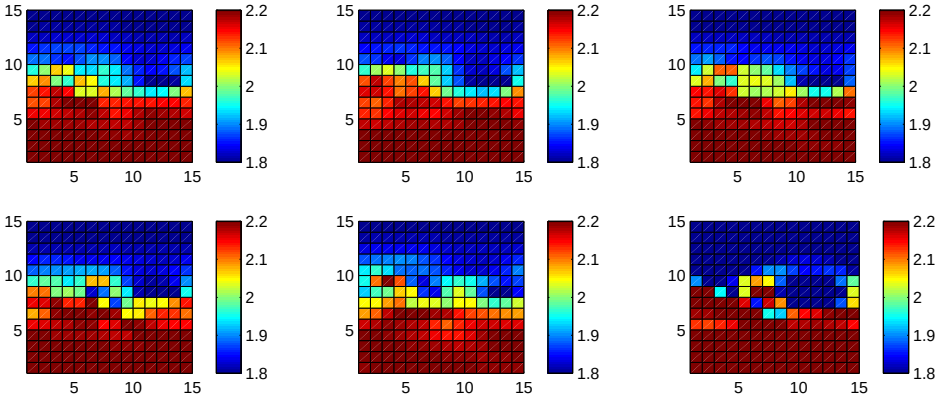


Figure 4.21: Geopotential field approximations for 600 second time step with 25%, 17% and 12% points observed in top row and 10%, 5% and 2% points observed in bottom row.

Chapter 5

Summary and Future Research

5.1 Summary

Inverse modelling and parameter estimation can be done using variational methods, of which specifically four dimensional data variation is considered in this thesis. Associated with this type of methods, however, are difficulties in connection with programming the adjoint model, which is needed to compute the exact gradient of the cost function. Additionally, having to integrate the adjoint model backwards in time adds significantly to the computational cost of the data assimilation process. To avoid manual implementation of adjoint code and to reduce computational complexity, approximation of the gradient calculation is considered.

In this thesis, proper orthogonal decomposition is studied as a model order reduction method. Being a fully data-driven method, it is exceptionally flexible in the sense that nothing of the physics of the underlying model needs to be known for its implementation. This can also be viewed as a weakness of the method, however, since the quality of approximation depends greatly on the set of snapshots, which is required to compute the POD projection matrix. The snapshots thus need to capture as much of the dynamics of the model as possible.

Two ways of using POD in 4D-VAR are presented in the thesis, namely reduced model variation and the reduced adjoint method. Both of these methods employ proper orthogonal decomposition to obtain a reduced order approximation of the forward linear tangent operator, which is itself an approximation of the Jacobian matrix. The difference between the two methods lies in the treatment of the forward model. Reduced Model Variation performs optimization of the parameter set entirely in reduced space, thereby achieving very low computational costs, but sacrificing accuracy of the end result. On the other hand, the reduced adjoint method uses POD to approximate only the adjoint model and

keeps the full order forward model in cost function evaluations. This slows down the optimization procedure, but significantly improves the quality of the final parameter set.

Firstly, both of the described methods as well as the full order adjoint method were tested with a linear model, i.e. the ground water subsurface contaminant model. Twin experiments were done, comparing the three methods in terms of accuracy and computational speed. Reduced model variation exhibited high speed of computation, but after some iterations eventually converged to a considerably higher cost function value that the other two methods. In terms of convergence, the reduced adjoint method performed best of all three, outperforming even the full adjoint method. The latter also converged, but required a significantly larger number of iterations.

The same methods were then tested on a more complex non-linear model, namely the Shallow Water Equations implementation by I. M. Navon and X. Chen. In the non-linear case, the additional approximation error that is introduced by using the tangent linear operator in place of the Jacobian, contributes significantly to the error of both approximate methods. In the case of reduced model variation, the effect is especially detrimental, since experiments showed that hardly any improvement of the cost function was achieved, rendering the method unsuitable for further experimentation. The reduced adjoint version, however, still achieved cost function reductions of at least two orders of magnitude and also still outperformed the full adjoint method in computational speed up to that point, using considerably fewer cost function evaluations, less than half as many, in fact. This put the reduced adjoint method at an advantage among the three approaches. In the situation, however, when exceptionally low cost values are required, they can only be reached by using the full adjoint method.

Additionally, sensitivity of the methods to perturbations in observations (i.e. the values that are available either as measurements or numerical experiments and that we take as the truth) and the number of observation points was examined. In the first set of experiments, very little deviation from the unperturbed results was visible when using perturbation factors up to 10%. Furthermore, the same behaviour was observed when the ratio of observation points was above 15%. Both of these results suggest that the reduced adjoint method is an easy to implement, robust way of achieving reasonably high cost function reductions in the context of variational data assimilation.

5.2 Future Research

In this section, some further research directions are outlined, which might improve on the work of this thesis.

Non-linear Approximation

When using POD to reduce the order of non-linear models, the tangent operator is introduced in order to obtain a linear reduced model. Otherwise the original non-linear operator would have had to be kept in (2.7). This was identified as a considerable source of approximation error, as is visible by the deterioration of experimental results when comparing the non-linear to the linear case. Additional gain in accuracy might be achieved by considering methods specifically designed to approximate non-linearities, such as the discrete empirical interpolation method (DEIM), introduced in 2010 by Chaturantabut and Sorensen in [10].

The Optimization Procedure

By example of other data assimilation work (see [2, 11]), the L-BFGS quasi Newton method was adopted throughout the thesis for all optimization. Since no other methods were considered in the extent of this work, it is a possibility that a different choice of optimization method might have beneficial effects on the results.

Choosing Snapshots for the POD basis

In this thesis, the set of model state snapshots required to compute the POD basis was collected by integrating the forward model once and selecting either all state vectors or part of them to be included in the snapshot matrix. If the snapshots could be chosen in a more intelligent way, this would likely benefit the reduced order model, decreasing its approximation error and thus improving the accuracy of the final estimate of the parameter set.

Bibliography

[1] M. U. Altaf. *Model reduced variational data assimilation for shallow water flow models.* PhD thesis, Delft University of Technology, 2010.

[2] M. U. Altaf, M. E. Gharamti, A. W. Heemink, and I. Hoteit. A reduced adjoint approach to variational data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 254:1–13, 2013.

[3] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Advances in Design And Control. Society for Industrial & Applied, 2005.

[4] G. M. Baxter, S. L. Dance, A. S. Lawless, and N. K. Nichols. Four-dimensional variational data assimilation for high resolution nested models. *Computers & Fluids*, 46:137 – 141, 2011.

[5] E. Belanger, A. Vincent, and A. Fortin. Data assimilation (4d-var) for shallow-water flow: the case of the chicoutimi river. *Visual Geosciences*, 8(1):1–17, 2003.

[6] Andrew F Bennett. *Inverse modeling of the ocean and atmosphere.* Cambridge University Press, 2002.

[7] F. Bouttier and P. Courtier. Data assimilation concepts and methods march 1999. *Meteorological training course lecture series. ECMWF*, 2002.

[8] T. Bui-Thanh, M. Damodaran, and K. E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal*, 42(8):1505–1516, 2004.

[9] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817, 2000.

[10] S. Chaturantabut and D. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[11] X. Chen and I. M. Navon. Optimal control of a finite-element limited-area shallow-water equations model. *Studies in Informatics and Control*, 18(1):41–62, 2009.

[12] P. Courtier, J. N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

[13] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans (1978–2012)*, 99(C5):10143–10162, 1994.

[14] R. Everson and L. Sirovich. Karhunen-loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[15] G. H. Golub and C. F. van Van Loan. *Matrix computations, 3rd edition*. The Johns Hopkins University Press, 1996.

[16] R. Gorder. Use of proper orthogonal decomposition in the analysis of turbulent flows. *University of Washington, Tech. Rep*, 2010.

[17] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and software*, 1(1):35–54, 1992.

[18] I. Hoteit, B. Cornuelle, V. Thierry, and D. Stammer. Impact of resolution and optimized ecco forcing on simulations of the tropical pacific. *Journal of Atmospheric and Oceanic Technology*, 25(1):131–147, 2008.

[19] C. Johnson, N. K. Nichols, B. J. Hoskins, S. P. Ballard, and A.S. Lawless. Four dimensional variational data assimilation in the presence of idealised rapidly growing weather systems. *Numerical analysis report 1/02, Department of Mathematics, The University of Reading*, 2002.

[20] M. P. Kaleta. *Model-Reduced Gradient-Based History Matching*. PhD thesis, Delft University of Technology, 2011.

[21] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[22] G. Kerschen, J. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dynamics*, 41(1-3):147–169, 2005.

[23] T. Leen, M. Rudnick, and D. Hammerstrom. Hebbian feature discovery improves classifier efficiency. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 51–56. IEEE, 1990.

[24] J. L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.

[25] H. V. Ly and H. T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33(1–3):223 – 236, 2001. Computation and control, proceedings of the sixth Bozeman conference.

[26] I. M. Navon, X. Zou, J. Derber, and J. Sela. Variational data assimilation with an adiabatic version of the nmc spectral model. *Monthly weather review*, 120(7):1433–1446, 1992.

[27] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[28] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[29] A. Schiller and J. Willebrand. A technique for the determination of surface heat and freshwater fluxes from hydrographic observations, using an approximate adjoint ocean circulation model. *Journal of marine research*, 53(3):433–451, 1995.

[30] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.

[31] L. N. Trefethen and D. Bau III. *Numerical linear algebra*. Number 50. Siam, 1997.

[32] J. R. Valstar, D. B. McLaughlin, C. te Stroet, and F. C. van Geer. A representer-based inverse method for groundwater flow and transport applications. *Water Resources Research*, 40(5), 2004.

[33] P. T. M. Vermeulen and A. W. Heemink. Model-reduced variational data assimilation. *Monthly Weather Review*, 134(10):2888–2899, 2006.

[34] K.-Y. Wang, D. J. Lary, D. E. Shallcross, S. M. Hall, and J. A. Pyle. A review on the use of the adjoint method in four-dimensional atmospheric-chemistry data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 127(576):2181–2204, 2001.

Appendices

Appendix A

Adjoint Model Derivation

Recalling the objective function J, we expand it with Lagrange multipliers in (2.12) to the form

$$\bar{J} = J(x_0, ..., x_m) + \sum_{i=1}^{m} \nu_i^T g_i(x_i, x_{i-1}). \tag{A.1}$$

Taking total differentials, we obtain

$$d\bar{J} = dJ + \sum_{i=1}^{m} \nu_i^T \frac{\partial g_i}{\partial x_i} dx_i + \sum_{i=1}^{m} \nu_i^T \frac{\partial g_i}{\partial x_{i-1}} dx_{i-1} \tag{A.2}$$

and by shifting the index of the second sum to join terms

$$\begin{aligned} d\bar{J} = dJ + \sum_{i=1}^{m} (\nu_i^T \frac{\partial g_i}{\partial x_i} dx_i + \ \nu_{i+1}^T \frac{\partial g_{i+1}}{\partial x_i} dx_i) \\ + \nu_1^T \frac{\partial g_1}{\partial x_0} dx_0 + \nu_{m+1}^T \frac{\partial g_{m+1}}{\partial x_m} dx_m. \end{aligned} \tag{A.3}$$

We require the boundary term $\nu_{m+1}^T \frac{\partial g_{m+1}}{\partial x_m} dx_m$ to be zero for any differential dx_m, which implies $\nu_{m+1} = 0$. Plugging in the differential of J, i.e.

$$dJ = \sum_{i=0}^{m} \frac{\partial J}{\partial x_i} dx_i \tag{A.4}$$

we arrive to

$$d\bar{J} = \sum_{i=1}^{m} (\nu_i^T \frac{\partial g_i}{\partial x_i} + \nu_{i+1}^T \frac{\partial g_{i+1}}{\partial x_i} + \frac{\partial J}{\partial x_i}) dx_i + \frac{\partial J}{\partial x_0} dx_0 + \nu_1^T \frac{\partial g_1}{\partial x_0} dx_0. \tag{A.5}$$

The adjoint model is obtained by setting the coefficients for terms $dx_1, ...dx_m$ to zero:

$$\nu_i^T \frac{\partial g_i}{\partial x_i} + \nu_{i+1}^T \frac{\partial g_{i+1}}{\partial x_i} + \frac{\partial J}{\partial x_i} = 0. \tag{A.6}$$

Taking the transpose gives us the system of adjoint equations for $i = 1, ..., m$

$$\left(\frac{\partial g_i}{\partial x_i}\right)^T \nu_i + \left(\frac{\partial g_{i+1}}{\partial x_i}\right)^T \nu_{i+1} + \left(\frac{\partial J}{\partial x_i}\right)^T = 0, \tag{A.7}$$

which is solved backwards in time with the initial condition $\nu_{m+1} = 0$. The total differential of $\bar{J}$ is then given by

$$d\bar{J} = (\frac{\partial J}{\partial x_0})^T dx_0 + (\frac{\partial g_1}{\partial x_0})^T \nu_1 dx_0 \tag{A.8}$$

and from there we get the gradient

$$\frac{d\bar{J}}{dx_0} = (\frac{\partial J}{\partial x_0})^T + (\frac{\partial g_1}{\partial x_0})^T \nu_1. \tag{A.9}$$

Inserting the actual terms for g_i and J into this general form gives us the formulae presented in sections 2.2 and 2.3.