.Januarv, 2001

LIDS- TH 2490

Research Supported By:

NSF Graduate Research Fellowship

Krylov Subspace Estimation

Michael K. Schneider

Krylov Subspace Estimation

by

Michael K. Schneider

B.S.E., Electrical Engineering
Princeton University, 1994

S.M., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1996

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

February, 2001

Krylov Subspace Estimation

by Michael K. Schneider

Submitted to the Department of Electrical Engineering
and Computer Science on January 11, 2001
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis proposes a new iterative algorithm for the simultaneous computation of linear least-squares estimates and error variances. There exist many iterative methods for computing only estimates. However, most of these will not also compute error variances. A popular method for computing only estimates is the conjugate gradient algorithm. The algorithm proposed in this thesis for the simultaneous computation of estimates and error variances is a variant of the conjugate gradient algorithm for computing estimates. The convergence of the proposed algorithm is extensively characterized both analytically and experimentally.

Variants of the proposed estimation algorithm are applied to two other statistical problems. The first is that of realization. Specifically, an iterative algorithm is developed for the simultaneous generation of a sample path of a given Gaussian random process and a low-rank approximation to the covariance matrix of a given process. The algorithm is compared to existing algorithms for realization in terms of an analytical estimate of computational cost and an experimental characterization of overall performance. The second statistical problem is that of space-time estimation. This thesis proposes an implementation of the Kalman filter and smoother in which each step of these recursive algorithms is solved iteratively. The resulting space-time estimation algorithm is especially suited for remote sensing problems. In particular, the algorithm is applied to the assimilation of measurements of sea surface height into a model of the ocean, the dynamics of which are given by a Rossby wave equation.

Lastly, this thesis examines the stability of infinite-dimensional discrete-time Kalman filters of a type arising in remote sensing problems. This is accomplished by developing a Lyapunov theory for infinite-dimensional linear systems whose states are elements in a Hilbert space. Two theorems, proved in this thesis, provide sufficient conditions for the state trajectories to converge either strongly or weakly to 0. This general theory is then used to establish sufficient conditions for strong and weak stability of infinite-dimensional Kalman filters.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering

Acknowledgments

Throughout my graduate studies, Alan Willsky has given me considerable support to pursue my research interests. His broad knowledge base and keen intuition have wisely guided my work. Moreover, his confidence in my abilities has allowed me to overcome many obstacles encountered during the past several years and to complete this thesis.

The two other members of my thesis committee, Jacob White and George Verghese, have also supported my research. I first learned about Krylov subspaces by taking a course taught by Jacob White. Since completing the course and starting my doctoral research, I have received important feedback from him about my research. In particular, a suggestion of his led to the development of the block algorithms in Chapter 3. George Verghese also aided my research by providing many references to the model reduction and smoothing literatures.

Hamid Krim and Paul Fieguth are two friends and former group members who have helped me at key points in my research. Hamid Krim provided references to the extreme value literature, and Paul Fieguth provided the ATSR data and a model for sea surface temperature.

Sanjoy Mitter has helped shape my perspective on system theory. My discussions with him on infinite-dimensional linear systems aided in developing the results in Chapter 6.

Two members of the Department of Earth, Atmospheric, and Planetary Sciences have proven invaluable in completing the space-time estimation research. In particular, Carl Wunsch initiated my work on the sea surface anomaly data by suggesting the data set and helping to formulate the ocean model. The preprocessed anomaly data were provided by Charmaine King.

Gilbert Strang generously gave some of his time early on in my work to discuss my research ideas with me. He provided much insight and many suggestions, including using the Cauchy interlace theorem to study convergence rates.

The Stochastic Systems Group is supportive, and I would like to thank all of them for assisting me and for just being themselves. Taylore Kelly helped put together the IGARSS paper and acknowledged that Erin Brockovitch is the best movie of all-time. John Richards edited various writings of mine and has always been willing to discuss random movies. Andrew Kim skillfully administers our computers and has admitted that he is more liberal now than before entering MIT. Jason Johnson has discussed information theory and presidential politics with me. Andy Tsai, who was admirably

dedicated to researching important health problems, has been willing to chat about his curve evolution implementations. I have had many discussions on all sorts of research topics in estimation and optimization with Dewey Tucker, who is nobly devoted to his music. The details of computational biology have been instilled in me by Ron Dror, who recognizes the importance of understanding the science in his applications. Martin Wainwright has discussed graphical models and optimization algorithms with me and deserves special thanks just for being Canadian. Alex Ihler installed the precious zip drive and forged so many signatures (none on this thesis, though). Erik Sudderth and I have had several discussions about multiscale modeling. Junmo Kim introduced me to the Kolmogorov-Smirnov distance. John Fisher taught me some learning theory and deserves special thanks just for being from Florida.

I would also like to acknowledge the support of several former members of the Stochastic Systems Group who have had a profound impact on me. Austin Frakt has been a good friend and has chatted with me on topics ranging from statistical modeling to public policy. Terrence Ho shared his thoughts with me on space-time estimation, the details of the Stochastic damped heat equation, and US economic policy. Cedric Logan has always been willing to talk politics. Ilya Pollak listened to my ideas on Radon transforms of random processes and tolerated my companionship in Phoenix. Mike Daniel talked to me about various statistical modeling issues and the world of finance. Paul Fieguth set an example of how to live an energy-efficient lifestyle. Bill Irving introduced me to the importance of good technical writing.

I have benefited from interacting with Laboratory for Information and Decision Systems students outside the Stochastic Systems Group. In particular, I would like to acknowledge Sekhar Tatikonda for providing feedback on the results in Chapter 6, Sean Warnick for discussions about control theory and functional analysis, and Peter Marbach for co-organizing the LIDS Colloquium with me.

While studying at MIT, I have occasionally gotten away. A considerable amount of this time has been spent in the companionship of Greg and Jill Warrington. I have greatly appreciated the many meals and good times spent together.

Over the years, I have gradually become assimilated into the close-knit Buske family. Their friendliness has been greatly appreciated. I have been particularly inspired by political discussions with Anne and Rachel and by Neil's devotion to crossword puzzles.

My parents, Gerd and Georgia Schneider, have instilled in me a devotion to education that led me to pursue a doctorate and, ultimately, finish this thesis. I admire them for their scholarly ways and their interest in living life to the fullest, and I would like to thank them for all of the love and support they have given me.

Finally, I would like to thank my wife, Dana Buske, for tolerating my many quirks and for listening to me ramble on about statistics. Her intelligence and strength have supported me through my graduate studies, and her adventurous taste in food, travels, and literature have added spice to our life together.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis focuses on the computation of linear least-squares estimates and the associated error variances. The error variances provide an important measure of the quality of the estimates. In particular, the error variances are useful for subsequent analysis of the estimates and for fusing the estimates with new data.

The interest here is in methods for simultaneously computing both the linear least-squares estimates and the associated error variances. Most existing methods are recursive in nature. That is, the algorithms break the computation down into a finite sequence of steps, after the completion of which the algorithm terminates with an exact solution. This class of algorithms includes Kalman filtering and Cholesky factorization.

For computing just estimates, on the other hand, there exist many iterative methods. These algorithms will compute a sequence of approximations to the exact solution. This approach can often lead to the computation of a good approximation with less work that an exact solution. One such method frequently used for computing estimates is the conjugate gradient algorithm [33]. This algorithm is a type of Krylov subspace method. Krylov subspace methods perform computations using quantities that lie in a type of subspace known as a Krylov subspace. Krylov subspace methods have been developed not just for computing linear least-squares estimates but also for performing many other linear algebra computations, particularly eigenanalysis.

In this thesis, a new Krylov subspace iterative method is proposed for the simultaneous computation of linear least-squares estimates and error variances. This Krylov subspace estimation algorithm is a variant of the standard conjugate gradient algorithm for computing estimates. The new method is widely applicable and is especially suited for estimation problems in which the quantity to be estimated is smoothly varying over space, and the data are mostly taken pointwise. Such problems arise in geophysical remote sensing, among other fields.

The Krylov subspace estimation algorithm can also be applied to the problems of synthesizing sample paths of a stochastic process and to the estimation of processes varying in both space and time. This thesis discusses the connection that both of these problems have to the static estimation problem for which the Krylov subspace estimation algorithm is initially developed. The resulting Krylov subspace algorithms for realization and space-time estimation estimation are characterized with various examples in this thesis. In particular, the algorithm for space-time estimation is used to analyze some oceanographic data.

The algorithms for static and space-time estimation are also characterized analytically in this thesis. In particular, a convergence theory is developed for the Krylov subspace estimation algorithm. The theory bounds convergence rates in terms of fundamental quantities in the estimation problem being solved. In addition, an analysis is carried out of how approximations made at various steps of the space-time estimation algorithm affect the final solution.

■ 1.1 Problems Addressed

There are principally three problems addressed in this thesis. Specifically, methods are developed for the computation of estimates and error variances for static estimation problems, the realization of random processes, and the computation of estimates and error variances for space-time estimation problems. Each of these problems is elaborated on next.

■ 1.1.1 Computation of Estimates and Error Variances for Large Static Linear Estimation Problems

One of the primary problems addressed in this thesis is that of computing estimates and error variances for large static linear estimation problems of a type arising in geophysics. The estimation problems considered in this thesis generally involve estimating a physical quantity smoothly varying over space from many irregularly spaced point measurements. An example of such a problem would be the estimation of sea surface temperature from satellite data.

The general static estimation problem has been considered by many others for certain types of prior models of the quantity to be estimated. In particular, a variety of techniques have been developed for cases were the prior models have some Markovian structure. The Markovianity of the models implies that inverses of covariance matrices in the estimation problem are sparse. The sparsity can be exploited by various recursive methods such as Kalman filtering and Cholesky factorization. More details on such existing methods are presented in Chapter 2.

In contrast, the estimation problems considered in this thesis are generally not Markov and the algorithmic focus is not on recursive methods, but iterative ones. Many existing iterative methods in numerical linear algebra assume the existence of efficient routines for multiplying vectors by matrices in the linear algebra problem. Such routines exist for the estimation problem if the covariance matrices can be made sparse in special bases. For example, covariance matrices of stationary processes can be factorized into a product of a diagonal matrix and discrete Fourier transform matrices. Thus, this thesis examines estimation problems for which the inverses of covariance matrices are not necessarily sparse, but the covariance matrices can be made sparse in bases associated with fast transforms.

This thesis contains not only a derivation of an iterative method for computing estimates and error variances but also an extensive analysis of its convergence. The analysis is statistical and involves characterizing the behavior of extreme values of certain random processes. The analysis is important for two reasons. First, it indicates how the convergence rate will change as parameters in the estimation problem vary. Second, the analysis suggests methods for accelerating convergence.

The issue of how to accelerate convergence is addressed in this thesis, as well. In particular, some simple preconditioners are developed and characterized experimentally for some test estimation problems. The goal of this work is not necessarily to find the best preconditioner but is to establish that convergence of the method for computing error variances can be accelerated.

Finally, the utility of the method is investigated by applying it to a static oceanographic estimation problem. Specifically, the algorithm is used to interpolate sea surface temperature from ATSR satellite data [29]. The data coverage tends to be sparse and irregular because measurements cannot be made through cloud cover. The error variances for this problem are important because they provide a quantitative characterization of the quality of estimates over the regions with no data.

■ 1.1.2 Realization Of Gaussian Processes

Another problem addressed in this thesis is that of approximate realization of Gaussian random processes. For this problem, one is given the covariance of a Gaussian random process. Then, one is interested in generating sample paths of an approximating random process and a representation of its covariance so that the approximating covariance closely matches the given one.

Realizations are important for many reasons. Although not explored in this thesis, the sample paths can be used for Monte Carlo studies. The other quantity generated by the realization algorithm, the approximate covariance, can be used for simplifying estimation problems. This application is considered in this thesis in the context of space-time estimation, which is discussed subsequently in Section 1.1.3.

There exist a variety of algorithms that can be used for realization. In particular, an eigendecomposition of the given covariance matrix can be used to perform approximate realization. Thus, many algorithms for eigenanalysis can be modified for approximate realization. Moreover, if the process to be realized is stationary, one can often use FFT's to generate approximate realizations [23] .

In this thesis, the previously mentioned Krylov subspace estimation algorithm is considered for addressing the realization problem. This is possible because of the close connections between realization of Gaussian random processes and linear least-squares estimation. The Krylov subspace estimation algorithm has the potential to offer key advantages over existing algorithms. Specifically, the error variances calculated by the Krylov subspace estimation algorithm have the interpretation of approximation error in the context of realization. Thus, the Krylov subspace realization algorithm provides the approximation error and does so at every iteration. The approximation error would be more difficult to obtain, at every iteration, from iterative eigenanalysis algorithms such as the Lanczos algorithm. Moreover, the Krylov subspace realization algorithm is not restricted to realizing stationary processes, as are the FFT methods.

A variety of random processes are considered as test cases for the Krylov subspace realization algorithm. One of these is a fractional Brownian motion. Fractional Brownian motions have also been used to test other realization algorithms and so serve as a good test case [30]. Yet another test case is an isotropic two-dimensional random known as the spherical covariance in the geostatistical community [23, 44].

1.1.3 Computation of Estimates and Error Variances for Space-Time Estimation

Many estimation problems arising in remote sensing involve estimating quantities varying over both space and time. This is especially true for oceanographic estimation problems. An example of such a problem is the estimation of sea surface anomaly, which is the deviation of sea surface height from a mean. Measurements of sea surface anomaly are taken pointwise by satellite altimeters, but the coverage of the ocean may be sparse. One is interested in estimating sea surface anomaly on a fine grid using current and past data.

The focus in this thesis is on iterative methods that can be used for computing both estimates and a representation of the error covariances. As for the static estimation problem, the error covariances are important for assessing the quality of the estimates and for fusing the estimates with new data. This may be especially important for a space-time estimation problem where new data are constantly being acquired.

A variety of methods have been developed for computing just estimates. These include the iterative methods of Bennett [4, 6]. These methods use the conjugate gradient algorithm to solve linear estimation problems. The problems are solved in batch. That is, data at all points in time are used to compute the estimates at every time point. This approach is often a very efficient one for computing estimates given a block of data. However, it is not a good approach if one is acquiring new data. This is due to the fact that one has to reprocess all of the data to get new estimates since the error covariances have not been computed to allow for efficient data fusion.

The method for space-time estimation proposed in this thesis is not a batch method but an implementation of a Kalman filter. The Kalman filter processes the data sequentially, one time step at a time, computing both estimates and error covariances. The issue addressed in this thesis is how to make use of the Krylov subspace methods for estimation and realization, previously discussed, for accelerating the processing of the Kalman filter steps. Although using the Kalman filter generates an estimate of the state at a given time using data from previous times, one can also consider using a smoothing algorithm to generate an estimate at each time step that uses all of the given data. This thesis also examines the issue of how to accelerate smoothing algorithms using the Krylov subspace estimation and realization algorithms.

The algorithms for space-time estimation developed in this thesis are tested on an oceanographic problem with real data. Specifically, the algorithm is used to estimate sea surface anomaly from TOPEX/ POSEIDON altimeter data [28]. The prior model for the estimation problem assumes that the sea surface anomalies propagate in time as a particular type of ocean waves, Rossby waves. Although the problem considered in this thesis is small, the model has a relatively high degree of statistical sophistication that makes it a good test problem.

Since the method proposed in this thesis for solving space-time estimation problems involves approximating steps of a Kalman filter, an important issue is the stability of the filter for space-time estimation problems. The degree of stability determines how the approximation errors propagate through the filter dynamics. Most existing stability studies have focused on proving a strict form of stability such as exponential stability [43] or on empirical results [74]. This thesis addresses the problem of establishing a form of stability, weaker than exponential stability, for infinite-dimensional Kalman filters. The theoretical framework captures the behavior of the high-dimensional Kalman filters used for certain remote sensing problems such as those considered in this thesis.

■ 1.2 Thesis Organization and Main Contributions

The remainder of this thesis is organized as follows.

Chapter 2, Linear Least-squares Estimation, Markovianity, and Sparse Linear Systems Solvers

This chapter provides some background and context for the results in the subsequent chapters. The linear least-squares estimation equations are derived in a general setting. Then, some classical techniques for computing the estimates are discussed. These include the conjugate gradient method and a recursive estimation algorithm. The context for the algorithmic discussion is Markovian estimation problems. Such problems have motivated the development of many existing algorithms. In contrast, the new algorithms proposed in this thesis are suitable for estimation problems that typically are not Markov.

Chapter 3, Krylov Subspace Estimation

The core Krylov subspace estimation algorithm and two different stopping criteria are derived in Chapter 3. The algorithm's convergence is extensively analyzed. The analysis necessitates the development of new results on the extreme values of random processes, which are contained in the chapter. The convergence analysis naturally leads to a consideration of two methods for accelerating convergence. Specifically, block and preconditioned forms of the algorithm are presented. The algorithm is characterized with various numerical examples. These include the results of applying the algorithm to the estimation of sea surface temperature.

Chapter 4, Krylov Subspace Realization

Chapter 4 contains a derivation of the Krylov subspace realization algorithm. The algorithm is compared to existing techniques for realization, particularly Lanczos and FFT-based methods. The performance of the Krylov subspace realization algorithm is characterized with a set of numerical examples. These include a fractional Brownian motion and a two-dimensional isotropic random field that has been used for geophysical modeling.

Chapter 5, The Krylov Subspace Method for Space-Time Estimation

Chapter 5 describes how the Krylov subspace methods for estimation and realization can be used to implement the Kalman filter for space-time estimation problems. A perturbation analysis of the Kalman filter is carried out. The analysis examines how the approximations made by the Krylov subspace methods affect the filter results. The chapter also contains a description of how these Krylov subspace methods can be used for smoothing. The computational complexity of the Krylov subspace methods for filtering and smoothing are analyzed and compared to straightforward implementations. The proposed methods are characterized with two sets of numerical examples. For the first set, the state obeys a stochastic heat equation, and the measurements are synthetic. For the second set, the state obeys a stochastic Rossby wave equation, and the measurements are real data of sea surface anomaly.

Chapter 6, Stability of Kalman Filters for Space-Time Estimation

Chapter 6 contains a study of the stability of the Kalman filters for space-time estimation considered in Chapter 5. This theoretical study is performed in an infinite-dimensional Hilbert space framework. Sufficient conditions for both strong and weak stability of the filters are provided. Proofs that the conditions are sufficient rely on a Lyapunov theory for infinite-dimensional time-varying linear systems that is developed in the chapter.

Chapter 7, Conclusions

The main contributions are summarized in Chapter 7. Directions for further research in the area of iterative methods for estimation are also suggested.

Chapter 2

Linear Least-Squares Estimation, Markovianity, and Sparse Linear Systems Solvers

This chapter presents background material from estimation theory and numerical linear algebra. Specifically, the linear least-squares estimation equations are derived, and three methods for solving the equations are briefly discussed. These include a recursive estimation algorithm, Cholesky factorization, and conjugate gradient. The conjugate gradient algorithm is presented in more detail than the other two methods since it plays an important role in subsequent chapters.

The presentation in this chapter focuses on Markovian estimation problems. Many existing estimation algorithms have been developed primarily for Markovian problems. The discussion of Markovianity provides some context for subsequent chapters in the thesis. Those chapters focus on estimation problems that are not necessarily Markov.

■ 2.1 Linear Least-Squares Estimation

Consider two zero-mean random vectors x and y with known covariances Λ_x and Λ_y and cross-covariance Λ_{xy}. Let the n components of x be quantities one would like to estimate and the m components of y be data. Let L denote the set of estimates linear in the data:

$$L = \left\{ Ay : A \in \mathbb{R}^{n\times m} \right\}. \tag{2.1}$$

The linear least-squares estimate $\hat{x}(y)$, is the linear estimator that minimizes the expected squared error:

$$\hat{x}(y) = \operatorname*{argmin}_{l\in L} \mathrm{E}\left[\|x - l\|^2 \right]. \tag{2.2}$$

This problem is best viewed in an inner-product space framework. The underlying real vector space is the set of all random variables that can be expressed as a linear combination of the random variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, the components of x

and y. The inner product between any two random variables u and v is given by their correlation: $\langle u, v\rangle = \mathrm{E}[uv]$. Now, consider the problem of estimating only x_1 and not all of x. Then, the estimator can be written $\hat{x}_1(y) = l_1^T y$ where

$$\begin{aligned} l_1 &= \operatorname*{argmin}_l \mathrm{E}\left[\left(x_1 - l^T y\right)^2\right] \\ &= \operatorname*{argmin}_l \left\|x_1 - l^T y\right\|^2 \end{aligned} \tag{2.3}$$

where $\|\cdot\|$ is the norm induced by the correlation inner product. This is a standard projection problem, which can be solved with the aid of the following theorem [52, Theorem 1 in Section 3.3].

Theorem 2.1.1. *Let S be a subspace and x an element in an inner-product space. Then, for a fixed $s_0 \in S$,*

$$\|x - s_0\| \leq \|x - s\| \quad \forall s \in S \Longleftrightarrow x - s_0 \perp s \quad \forall s \in S. \tag{2.4}$$

In the case of estimating x_1, from y, the subspace is the span of the components of y. Thus, one would like to find an l_1 such that

$$\langle x_1 - l_1^T y, y_i\rangle = 0 \qquad i = 1, \ldots, m. \tag{2.5}$$

This can be rewritten as

$$\Lambda_{x_1 y} = l_1^T \Lambda_y. \tag{2.6}$$

Thus, $\hat{x}_1(y) = \Lambda_{x_1 y}\Lambda_y^{-1} y$. Since the problem of estimating all of x can be solved componentwise,

$$\hat{x}(y) = \Lambda_{xy}\Lambda_y^{-1} y. \tag{2.7}$$

Note that the linear least-squares estimate only depends on the covariance matrices and not on higher-order statistics.

The quality of the estimate can be measured with the covariance of the error, $x - \hat{x}(y)$. Now,

$$(\mathrm{Cov}(x - \hat{x}(y)))_{ij} = \langle (x - \hat{x}(y))_i, (x - \hat{x}(y))_j\rangle = \langle (x - \hat{x}(y))_i, x_j\rangle \tag{2.8}$$

because the error, $x - \hat{x}(y)$, is uncorrelated with linear functions of y. Thus,

$$\begin{aligned} (\mathrm{Cov}(x - \hat{x}(y)))_{ij} &= \langle x_i, x_j\rangle - \langle (\Lambda_{xy}\Lambda_y^{-1} y)_i, x_j\rangle \\ &= (\Lambda_x)_{ij} - (\Lambda_{xy}\Lambda_y^{-1}\Lambda_{xy}^T)_{ij} \end{aligned} \tag{2.9}$$

So, the error covariance is given by

$$\Lambda_e = \Lambda_x - \Lambda_{xy}\Lambda_y^{-1}\Lambda_{xy}^T. \tag{2.10}$$

The expressions for the estimate (2.7) and error covariances (2.10) just derived are convenient, but they are often written in a different form. Write y as a linear measurement of x embedded in uncorrelated additive noise:

$$y = Cx + n. \tag{2.11}$$

Then, $\Lambda_{xy} = \Lambda_x C^T$, and $\Lambda_y = C\Lambda_x C^T + \Lambda_n$. If Λ_x and Λ_n are invertible, one can write

$$\begin{aligned}\hat{x}(y) &= \Lambda_x C^T \left(C\Lambda_x C^T + \Lambda_n\right)^{-1} y \\ &= \left(\Lambda_x^{-1} + C^T\Lambda_n^{-1}C\right)^{-1} C^T\Lambda_n^{-1} y\end{aligned} \tag{2.12}$$

and

$$\begin{aligned}\Lambda_e &= \Lambda_x - \Lambda_x C^T \left(C\Lambda_x C^T + \Lambda_n\right)^{-1} C\Lambda_x \\ &= \left(\Lambda_x^{-1} + C^T\Lambda_n^{-1}C\right)^{-1}\end{aligned} \tag{2.13}$$

by using the identity

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1} DA^{-1}. \tag{2.14}$$

The latter forms are frequently used because Λ_x^{-1} and Λ_n^{-1} are often sparse as discussed next.

■ 2.2 Graphical Representations of Markovianity

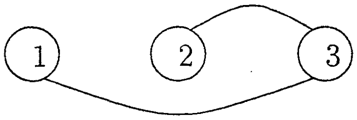


Figure 2.1. This sparsity graph is of the matrix A in (2.15).

Both Markovianity of a process and sparseness of a covariance matrix are best described with graphs [32, 66]. The sparsity graph of a symmetric matrix A has as many nodes as rows in the matrix, and the nodes are numbered accordingly. An edge exists between two nodes $i \neq j$ if and only if $A_{ij} \neq 0$. An example is given by the graph in Figure 2.1 for the matrix

$$A = \begin{pmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1/2 & 1/2 & 1 \end{pmatrix}. \tag{2.15}$$

A similar graph can be used to describe the Markovian structure of a process $x_1, \ldots, x_n$. The graph has n nodes labeled 1 through n. An edge exists between nodes i and j if and only if x_i and x_j are not conditionally independent given $\{x_1, \ldots, x_n\} \setminus \{x_i, x_j\}$. As it turns out, these graphs are closely connected, as described by the Hammersley-Clifford Theorem specialized to Gaussian processes. We specialize to Gaussian processes because they have the characteristic that conditional independence is equivalent to conditional decorrelation, the structure of which is wholly captured by the covariance matrix.

Theorem 2.2.1 (Hammersley-Clifford [8]). *For a finite-length Gaussian process the graph of the Markovian structure is the same as the sparsity graph of the inverse of the covariance matrix.*

The Hammersley-Clifford Theorem has important computational implications. In particular, it shows that for the important class of Gauss-Markov processes, one can compute the linear least-squares estimate by solving the sparse system of equations in (2.12). The degree of sparsity will depend on the structure of the Markovianity. Conversely, any method for computing the linear least-squares estimate of a Gauss-Markov process can be used to solve a sparse symmetric positive-definite system of equations. As it turns out, many methods for computing the linear least-squares estimate have exactly the same structure, as modeled by operations on the associated graph. In particular, recursive estimation and Cholesky factorization (Gaussian elimination) are the same structurally for graphs with no cycles; although the methods make very different intermediate calculations when computing the linear least-squares estimate.

2.3 Graph-Based Algorithms for Estimation

There are many methods for recursively computing estimates on graphs. One such is as follows [2, 15]. The linear least-squares estimation problem considered here is that of estimating a zero-mean Gaussian process $x_1, \ldots, x_n$ which is Markov on a cycle-free graph G, from measurements $y_i = C_i x_i + n_i$ such that the n_i are independent and independent of the x_i. The algorithm then consists of the following operations. The first step is to find a node with only one neighbor. Such a node is termed a leaf node. Then, for any leaf node i with neighbor j, one computes the linear least-squares estimate, $\hat{x}(i|i)$, of x_i given y_i and the estimate of x_j given y_i, denoted by $\hat{x}(j|i)$. Next, one modifies the graph G by eliminating node i and adding the measurement $\hat{x}(j|i) = x_j + e_j$ to node j. The algorithm iterates until there are no nodes left (see Figure 2.2). A second sweep of the algorithm computes the linear least-squares estimate of each x_i given all of the y_i. The structure of the primary sweep consists of visiting consecutive leaf nodes in the graph and deleting them. This is the same structure one finds in Cholesky factorization.

Of the many implementations of Cholesky factorization, the following is a useful one for the estimation problem described above [22]. Specifically, if we group the x_i, y_i, and C_i to a create a system of equations of the form (2.12), the matrix $\Lambda_x^{-1} + C^T \Lambda_n^{-1} C$ will have the same sparsity graph as that describing the Markovian structure of $x_1, \ldots, x_n$. Thus, the graph is cycle-free. Cholesky factorization consists of finding a leaf node i and its neighbor j and then performing one elementary operation on $\Lambda_x^{-1} + C^T \Lambda_n^{-1} C$

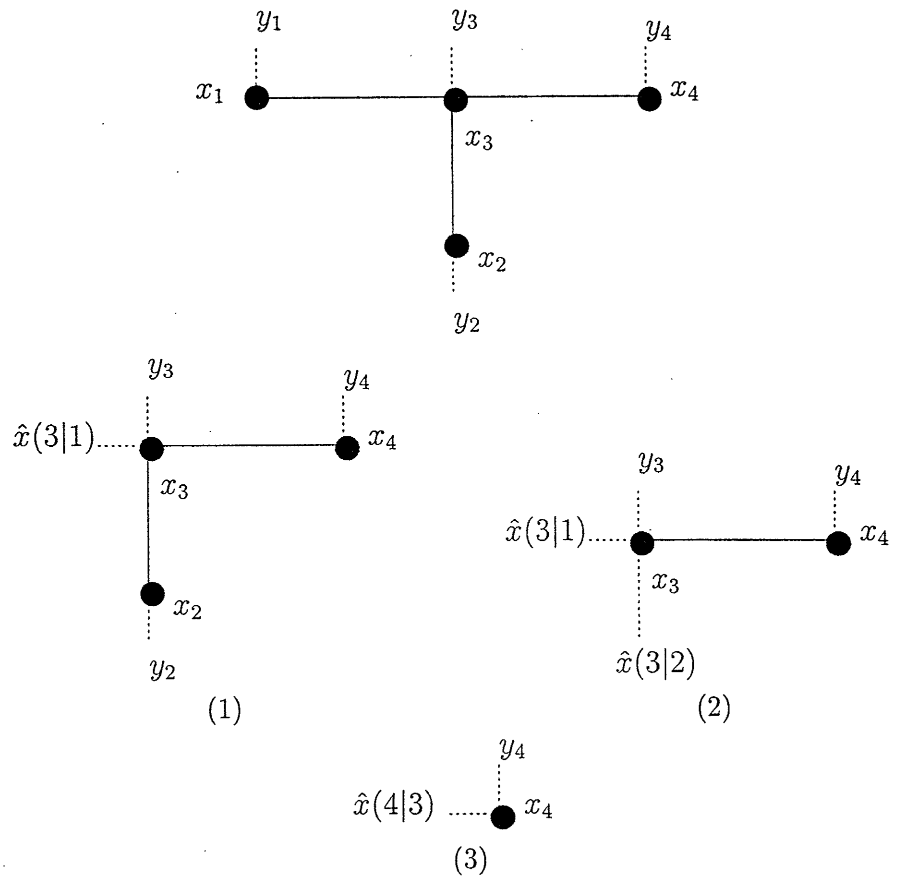
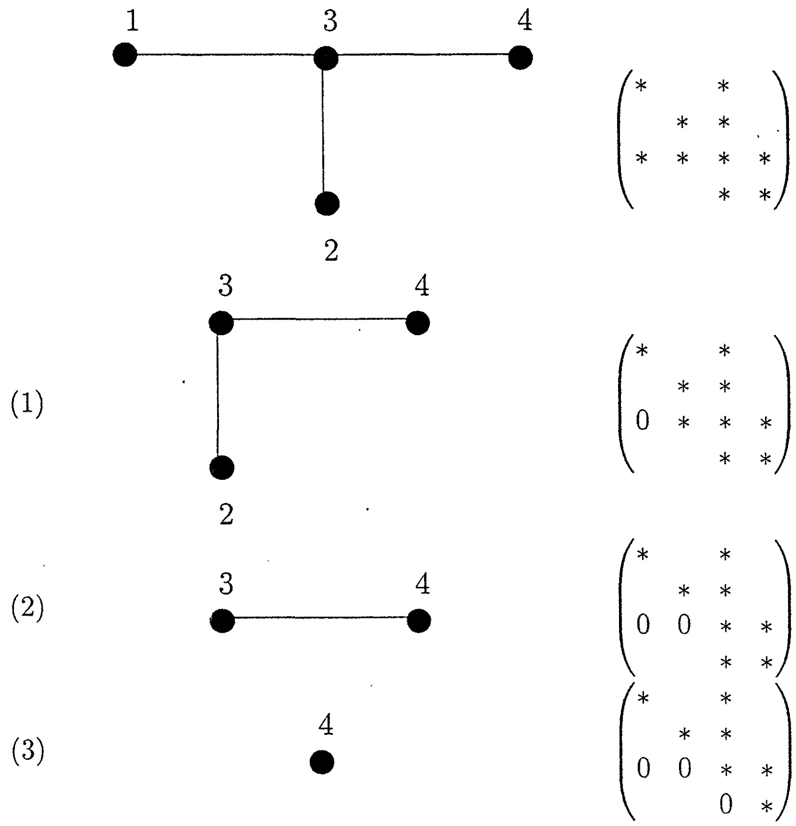


Figure 2.2. The structure of the recursive estimation algorithm is illustrated here for the the estimation of x_1, x_2, x_3, x_4 from associated measurements y_1, y_2, y_3, y_4. At top is the conditional decorrelation graph. Measurements are connected to their associated nodes by a dotted line. A single recursive estimation sweep consists of three steps. One possible set of associated modifications to the graph is shown here.

to eliminate the ij-th entry. Then, one deletes node i from the graph. The procedure iterates until there are no nodes left, as illustrated in Figure 2.3. At this point, one has completed the Cholesky factorization. The estimate of x given y can be computed by back-substitution. Note that this Cholesky factorization procedure has exactly the same structure as recursive estimation in that leaf nodes are visited and deleted from the same graph.

One consequence of both recursive estimation and Cholesky factorization having the same structure is that the two algorithms have the same limitations. In particular, if the graph has cycles, applying the algorithms may become computationally intensive. One method for applying both algorithms in this setting is to group nodes together to form a cycle-free graph and then apply block forms of recursive estimation and Cholesky



$$\begin{pmatrix} * & & * & \\ & * & * & \\ * & * & * & * \\ & & * & * \end{pmatrix}$$

$$\begin{pmatrix} * & & * & \\ & * & * & \\ 0 & * & * & * \\ & & * & * \end{pmatrix}$$

$$\begin{pmatrix} * & & * & \\ & * & * & \\ 0 & 0 & * & * \\ & & * & * \end{pmatrix}$$

$$\begin{pmatrix} * & & * & \\ & * & * & \\ 0 & 0 & * & * \\ & & 0 & * \end{pmatrix}$$

Figure 2.3. The structure of Cholesky factorization is illustrated here for the matrix structure at top. This matrix has the structure of $\Lambda_x^{-1} + C^T \Lambda_n^{-1} C$ for the example in Figure 2.2. The three elimination steps and associated graph manipulations are illustrated here. For each step, the matrix is illustrated by indicating the non-zero entries with $*$ and eliminated (zeroed) entries with 0's. Notice that the graphs in each step are the same as those in Figure 2.2.

factorization. One example of how nodes in a square $n \times n$ grid could be grouped together is illustrated for a 5×5 grid in Figure 2.4. Notice that the largest node is formed by grouping together $O(n)$ nodes, creating a rather large vector. However, one can't group nodes so as to have a maximal grouping involving fewer than $O(n)$ nodes [32]. Thus, these methods are of limited value when the graph contains many cycles, as often arises in the modeling of spatially-varying processes. In these cases, iterative techniques may be of more value.

■ 2.4 The Conjugate Gradient Algorithm

One approach to solving a general symmetric positive definite system of equations, $Ax = b$, is to use a Krylov subspace iterative method. Such methods compute, at

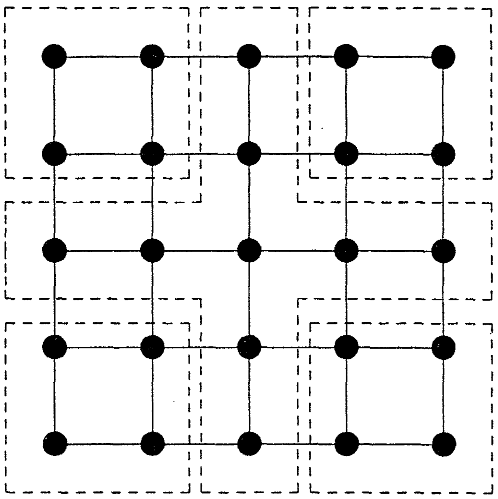


Figure 2.4. The nodes and edges in an $n \times n$ grid can be grouped together to form a tree. In the 5×5 case illustrated here, the state at the root of the tree consists of the middle random variables, and the states at the four children consist of the groupings of the four corner random variables.

iteration k, a vector x_k^* that minimizes

$$\|A(x_k + x_0) - b\|_{A^{-1}} \tag{2.16}$$

over all x_k in a Krylov subspace of dimension k generated by A, where x_0 is an initial guess. The Krylov subspace of dimension k generated by the matrix A and a vector s is the span of products of the vector s and powers of the matrix A:

$$\mathcal{K}(A, s, k) \triangleq \operatorname{span}\left(s, As, A^2 s, \ldots, A^{k-1} s\right). \tag{2.17}$$

The implicit assumption is that s does not lie in a proper eigenspace of the matrix A. These Krylov subspaces have a structure that is useful for solving certain linear systems. In particular, one can update x_{k-1}^* to obtain x_k^* with relatively few computations. The workload is dominated by the multiplication of a vector by the matrix A. When this is efficient, *e.g.* if A is sparse, then each iteration is efficient.

The conjugate gradient iteration is one method for computing the x_k^* [33, Section 10.2]. The Krylov subspace used is the one generated by A and the initial residual $r_0 = b - Ax_0$. The method computes bases $p_1, \ldots, p_k$ for $\mathcal{K}(A, s, k)$ such that the p_i are A-conjugate, *i.e.* $p_i^T A p_j = \delta_{ij}$. Then, computing x_k^* boils down to finding a z_k^* that minimizes

$$\|A(Pz_k + x_0) - b\|_{A^{-1}}, \tag{2.18}$$

where

$$P_k = \begin{bmatrix} p_1 & \cdots & p_k \end{bmatrix}. \tag{2.19}$$

The solution is $z_k^* = P_k^T(b - Ax_0)$. Thus, one has the following simple recursion for x_k^* in terms of the p_i: $x_k^* = x_{k-1}^* + p_k p_k^T b$. The vectors $p_1, p_2, \ldots$, known as conjugate search directions, are computed, in turn, with the following iteration:

$$p_1 = \frac{r_0}{\sqrt{r_0^T A r_0}} \tag{2.20}$$

$$r_k = b - Ax_k^* \tag{2.21}$$

$$\nu_{k+1} = r_k - (r_k^T A p_k) p_k \tag{2.22}$$

$$p_{k+1} = \frac{\nu_{k+1}}{\sqrt{\nu_{k+1} A \nu_{k+1}}}. \tag{2.23}$$

That the p_i computed by this recursion span Krylov subspaces and are A-conjugate is not immediately obvious. The following proofs establish both facts(*c.f.* [33, Section 10.2.4]).

Proposition 2.4.1. *The k-th estimation residual is in a $(k+1)$-dimensional Krylov subspace: $r_k \in \mathcal{K}(A, r_0, k+1)$. The k-th direction is in a k-dimensional Krylov subspace: $p_k \in \mathcal{K}(A, r_0, k)$.*

Proof. The proof is by induction.

First, note that $p_1 \propto r_0 \in \mathcal{K}(A, r_0, 1)$, and $r_1 = b - A(p_1 p_1^T b + x_0) \in \mathcal{K}(A, r_0, 2)$ since $p_1 \propto b$.

Now, suppose that $p_{k-1} \in \mathcal{K}(A, r_0, k-1)$ and $r_{k-1} \in \mathcal{K}(A, r_0, k)$. Then, $p_k \propto r_{k-1} - (r_{k-1}^T A p_{k-1}) p_{k-1} \in \mathcal{K}(A, r_0, k)$. Furthermore, $r_k = b - A(p_1 p_1^T b + \cdots + p_k p_k^T b + x_0) \in \mathcal{K}(A, r_0, k+1)$. □

Proposition 2.4.2. *The $p_i^T y$ are A-conjugate. In other words, $p_k^T A p_i = 0$ for $i < k$.*

Proof. Fix k, and consider $p_{k+1}^T p_i$ for $i \leq k$.

Note that

$$\|A(P_k z_k + x_0) - b\|_{A^{-1}} = \|P_k z_k + x_0 - x^*\|_A \tag{2.24}$$

where $x^* = A^{-1}b$. By the orthogonality principle, $A^{-1} r_k = P_k z_k^* + x_0 - x^*$ is A-conugate to every vector in $\mathcal{K}(A, r_0, k)$. Thus, r_k is A-conjugate to $p_1, \ldots, p_{k-1}$ since $Ap_1, \ldots, Ap_{k-1} \in \mathcal{K}(A, r_0, k)$. Since p_{k+1} is r_k made A-conjugate to p_k via (2.22) and (2.23), p_{k+1} is A-conjugate to p_i for $i \leq k$. □

Not only is each iteration of conjugate gradient often efficient, but one also requires typically few iterations to achieve convergence. The standard convergence theory for conjugate gradient makes use of an analysis involving matrix polynomials [22, Section 6.6.4]. The polynomials arise because the structure of the Krylov subspaces allows one to write $r_k = q_k(A) r_0$, where q_k is a k-th order polynomial such that $q_k(0) = 1$. It turns out that conjugate gradient picks the best polynomial for minimizing squared error, as stated in the following theorem [22, p. 313].

Theorem 2.4.1. *The k-th residual, r_k, of the conjugate gradient method for solving $Ax = b$ satisfies*

$$\begin{aligned} \frac{\|r_k\|_{A^{-1}}}{\|r_0\|_{A^{-1}}} &\leq \min_{q \in \mathcal{Q}_k} \max_{\lambda_i \in \sigma(A)} |q(\lambda_i)| \\ &\leq \frac{1}{T_k\left(\frac{\kappa+1}{\kappa-1}\right)} \end{aligned} \tag{2.25}$$

where $\sigma(A)$ is the spectrum of A, $\mathcal{Q}_k$ is the set of all k-th order polynomials taking on the value 1 at 0, T_k is the k-th Chebyshev polynomial, and κ is the condition number of A.

The implication of this convergence theory is that convergence speed increases as the eigenvalues become clustered. As a result, much research has been focused on preconditioning the system of equations to cluster the eigenvalues. Preconditioning involves transforming a system $Ax = b$ by a matrix B to yield $(BAB^T)(B^{-T}x) = Bb$. A good preconditioner B is one for which matrix-vector multiplies are efficient and for which the condition number of BAB^T is close to 1. For many estimation problems with Markovian (hence, sparse) structure, one can find good preconditioners. Thus, conjugate gradient is a popular choice for computing estimates.

One can also precondition the iterative methods for computing error variances proposed in subsequent chapters. The convergence theory established in Section 3.3 suggests a different kind of preconditioning, however. The goal of preconditioning for error variance computations is to separate the eigenvalues of the preconditioned matrix, BAB^T. Thus, the classical result on preconditioning stated here for solving linear systems of equations with conjugate gradient differs from those in this thesis.

The next chapter focuses on how conjugate gradient can be used to compute estimation error variances for static estimation problems, albeit for ones that are not necessarily Markov.

Chapter 3

A Krylov Subspace Method for Static Estimation Problems

The subject of this chapter is finite-dimensional linear least-squares estimation of an l-dimensional random vector x from an m-dimensional random vector y. The relationship between x and y is written as $y = z + n$, where n is noise uncorrelated with x and

$$z = Cx \tag{3.1}$$

for a matrix C reflecting the type of measurements of x. In the Bayesian framework considered here, x, z, and n have known means and covariances. The covariance matrices are denoted by Λ_x, Λ_z, and Λ_n, respectively, and, without loss of generality, the means are assumed to be zero. Recall from Chapter 2 that the linear least-squares estimate (LLSE) of x given y is

$$\hat{x}(y) = \Lambda_x C^T \Lambda_y^{-1} y \tag{3.2}$$

where $\Lambda_y = \Lambda_z + \Lambda_n = C\Lambda_x C^T + \Lambda_n$ is the covariance of y.

Direct computation of $\hat{x}(y)$ is difficult if x and y are of high dimension. In particular, the work in this chapter was motivated by problems in which x represents a spatially-distributed phenomenon and y measurements encountered in applications ranging from image processing to remote sensing. For example, when x and y represent natural images, they typically consist of $256 \times 256 = 65536$ pixels. In problems from physical oceanography, the dimensions of x and y are typically upwards of 10^5 and 10^4, respectively (*e.g.* see [29]). Furthermore, in applications such as remote sensing in which the measurement sampling pattern is highly irregular, Λ_z is typically a full matrix that is far from Toeplitz. This prevents one from solving the linear system (3.2) by spectral or sparse matrix methods. However, Λ_y often has a considerable amount of structure. For example, the covariance, Λ_x, of the full spatial field, is often either Toeplitz or well-approximated by a very sparse matrix in an appropriate basis, such as a local cosine basis [53]. The measurement matrix C is often sparse, and the noise covariance Λ_n is often a multiple of the identity. Thus, multiplying vectors by Λ_y is often efficient, and an iterative method for solving linear systems that makes use of Λ_y-multiplies, such as a Krylov subspace method, could be used to compute $\hat{x}(y)$.

For linear least-squares estimation problems, one is often interested not only in computing the estimates but also some portion of the estimation error covariance matrix. Recall from Chapter 2 that the covariance of the estimation error,

$$\Lambda_{e_x}(y) = \Lambda_x - \Lambda_x C^T \Lambda_y^{-1} C \Lambda_x, \tag{3.3}$$

is the difference between the prior covariance and the error reduction. The terms on the diagonal of this matrix are the estimation error variances, the quantities most sought after for characterizing the errors in the linear estimate. A natural question to ask is whether a Krylov subspace method for computing the linear estimate $\hat{x}(y)$, such as the method of conjugate gradients (CG) outlined in Section 2.4, could be adapted for computing portions of the error covariance matrix. This chapter presents an interpretation of CG in the context of linear least-squares estimation that leads to a new algorithm for computing estimation error variances.

Many researchers in the geosciences have used CG for computing LLSEs. In particular, Bennett, Chua, and Leslie [4–6] and da Silva and Guo [19] use CG for computing LLSEs of atmospheric variables. The structures of these estimation problems are very similar to the ones considered here. In particular, the quantities to be estimated, x, are spatially-varying processes, and the measurement matrices, C, are sparse. However, they do not consider using a Krylov subspace method for the computation of error variances. We not only propose such a method in this chapter but also provide a detailed convergence analysis.

Paige and Saunders [63] and Xu, Kailath, *et al.* [79–82] have developed Krylov subspace methods for solving statistical problems that are closely related to linear least-squares estimation. The LSQR algorithm of Paige and Saunders solves a regression problem and can compute approximations to the standard errors. The regression problem is a more general version of linear least-squares estimation in which a prior model is not necessarily specified. In the special case of linear least-squares estimation, the standard errors of the regression problem are the estimation error variances. Thus, LSQR can compute approximations to the error variances. The novelty of our work is that it focuses specifically on linear least-squares estimation and takes advantage of the structure inherent in many prior models for image processing problems. In particular, many such prior models imply a covariance of the data, $\Lambda_y = \Lambda_z + \Lambda_n$, in which the signal covariance matrix, Λ_z, has eigenvalues that decay rapidly to zero and the noise covariance matrix, Λ_n, is a multiple of the identity. Such properties are exploited by our algorithm. These assumptions were also made in the work of Xu, Kailath, *et al.* for signal subspace tracking. For that problem, one is interested in computing the dominant eigenvectors and eigenvalues of Λ_z. Although computing the dominant eigenvectors and eigenvalues of Λ_z is sufficient to compute an approximation to the estimation error variances, it is not necessary. We do not explicitly compute eigenvectors or eigenvalues. This provides us with the opportunity to exploit preconditioning techniques in a very efficient manner.

Section 3.1 discusses our interpretation of CG as used to compute LLSEs. This naturally leads to the presentation of a new iterative algorithm for computing estimation error variances. Section 3.2 proposes two alternative stopping criteria. The main convergence result is presented in Section 3.3. Techniques for accelerating convergence, including preconditioned and block algorithmic forms, are discussed in Section 3.4. The main convergence result is proved in Section 3.5. Finally, Section 3.6 illustrates the proposed techniques with various numerical examples.

■ 3.1 The Estimation Algorithm

The primary difficulty in computing the LLSE $\hat{x}(y)$ in (3.2) is the large dimension of the data y. The signal in the data, however, typically lies primarily in a much lower dimensional subspace. One can take advantage of this fact to compute an approximation to $\hat{x}(y)$ by computing, instead of $\hat{x}(y)$, the LLSE of x given a small number of linear functionals of the data, $p_1^T y, p_2^T y, \ldots, p_k^T y$. For a particular sequence of linearly independent linear functionals, $p_1^T, p_2^T, \ldots, p_k^T$, let $\hat{x}_k(y)$ denote the LLSE of x given $p_1^T y, p_2^T y, \ldots, p_k^T y$. If most of the signal components in y lie in the span of $p_1, p_2, \ldots, p_k$, then the estimate $\hat{x}_k(y)$ approximates $\hat{x}(y)$. In this case, the covariance of the error in the estimate $\hat{x}_k(y)$, $\Lambda_{e_x,k}(y) \triangleq \text{Cov}(x - \hat{x}_k(y))$, approximates the optimal error covariance, $\Lambda_{e_x}(y) \triangleq \text{Cov}(x - \hat{x}(y))$.

The principal novelty of the algorithm we propose in this chapter is the use of linear functionals that form bases for Krylov subspaces. The use of Krylov subspaces for solving linear algebra problems is not new, but the application of Krylov subspaces to the computation of error covariances is new. Recall from Section 2.4 that a Krylov subspace of dimension k, generated by a vector s and the matrix Λ_y, is the span of $s, \Lambda_y s, \ldots, \Lambda_y^{k-1} s$ and is denoted by $\mathcal{K}(\Lambda_y, s, k)$. The advantage of using linear functionals that form bases for Krylov subspaces is twofold. One reason is theoretical. Specifically, one can consider the behavior of the angles between $\mathcal{K}(\Lambda_y, s, k)$ and the dominant eigenvectors, u_i, of Λ_y: $\arcsin \|(I - \pi_k)u_i\| / \|u_i\|$, where π_k is the orthogonal projection onto $\mathcal{K}(\Lambda_y, s, k)$. As noted in [70], these angles are rapidly decreasing as k increases. Thus, linear functionals from Krylov subspaces will capture most of the dominant components of the data. Another reason for using functionals from Krylov subspaces is computational. As discussed in the introduction, the structure of Λ_y in many problems is such that multiplying a vector by Λ_y is efficient. A consequence of this fact is that one can generate bases for the Krylov subspaces efficiently.

The specific linear functionals used in this chapter are the search directions generated by standard CG for solving a linear system of equations involving the matrix Λ_y. The conjugate search directions, $p_1, \ldots, p_k$, form a basis for $\mathcal{K}(\Lambda_y, s, k)$ and are Λ_y-conjugate (see Section 2.4). The Λ_y-conjugacy of the search directions implies that $\text{Cov}(p_i^T y, p_j^T y) = \delta_{ij}$; so, these linear functionals of the data are white. Thus, we can draw the novel conclusion that CG whitens the data. The whiteness of the linear functionals of the data allows one to write

$$\hat{x}_k(y) = \sum_{j=1}^{k} \left(\Lambda_x C^T p_i\right) p_i^T y \tag{3.4}$$

$$\Lambda_{e_x,k}(y) = \Lambda_x - \sum_{j=1}^{k} \left(\Lambda_x C^T p_i\right) \left(\Lambda_x C^T p_i\right)^T \tag{3.5}$$

which follows from $\mathrm{Cov}(p_1^T y, \ldots, p_k^T y) = I$. [1] One can now write recursions for the estimates and error variances in terms of the quantities $b_{y,k} = \Lambda_x C^T p_k$. We call these the *filtered backprojected* search directions because the prior covariance matrix Λ_x typically acts as a low-pass filter and C^T is a backprojection (as the term is used in tomography) since C is a measurement matrix. In terms of the $b_{y,k}$, the recursions have the following form:

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + b_{y,k} p_k^T y \tag{3.6}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - ((b_{y,k})_i)^2 \tag{3.7}$$

with initial conditions

$$\hat{x}_0(y) = 0 \tag{3.8}$$

$$(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii} \tag{3.9}$$

where $i = 1, \ldots, l$. Unfortunately, the vectors $p_1, p_2, \ldots$ generated by standard CG are not Λ_y-conjugate to a reasonable degree of precision because of the numerical properties of the method.

The numerical difficulties associated with standard CG can be circumvented using a Lanczos iteration, combined with some form of reorthogonalization, to generate the conjugate search directions [33, §9.1 and §9.2]. The Lanczos iteration generates a sequence of vectors according to the following recursion:

$$\alpha_k = q_k^T \Lambda_y q_k \tag{3.10}$$

$$h_k = \Lambda_y q_k - \alpha_k q_k - \beta_k q_{k-1} \tag{3.11}$$

$$\beta_{k+1} = \|h_k\| \tag{3.12}$$

$$q_{k+1} = \frac{h_k}{\beta_{k+1}} \tag{3.13}$$

which is initialized by setting q_1 equal to the starting vector s, $q_0 = 0$, and $\beta_1 = 0$. The Lanczos vectors, $q_1, q_2, \ldots$, are orthonormal and such that

$$\begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix}^T \Lambda_y \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} \tag{3.14}$$

[1] Specifically (3.4) and (3.5) follow from (3.2) and (3.3) with the substitution of I for Λ_y and $[p_1^T C \ \cdots \ p_k^T C]$ for C.

is tri-diagonal for all k. Let $T_{y,k}$ denote this tri-diagonal matrix and $L_{y,k}$ the lower bi-diagonal Cholesky factor. Then, the vectors defined by

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_k \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} L_{y,k}^{-T} \tag{3.15}$$

are equal, up to a sign, to the conjugate search directions generated by CG in exact arithmetic. That $L_{y,k}$ is lower bi-diagonal allows one to use a simple one-step recursion to compute the p_i from the q_i. Note also that the $b_{y,k} = \Lambda_x C^T p_i$ can be computed easily in terms of a recursion in $\Lambda_x C^T q_i$. These latter quantities are available since the computation of q_{k+1} requires the product $\Lambda_y q_k = C(\Lambda_x C^T)q_k + \Lambda_n q_k$.

One of the main advantages to using the Lanczos iteration followed by Cholesky factorization is that one can use a variety of reorthogonalization schemes to ensure that the Lanczos vectors remain orthogonal and, in turn, that the associated conjugate search directions are Λ_y-conjugate. The simplest scheme is full orthogonalization [22, §7.4]. This just recomputes h_k as

$$h_k := h_k - \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T h_k \tag{3.16}$$

between the steps in (3.11) and (3.12). This is typically sufficient to ensure orthogonality among the q_i. However, one can also use more complicated schemes that are more efficient such as selective orthogonalization [65]. A discussion of the details can be found in Appendix B. We have found that the type of orthogonalization used does not significantly affect the quality of the results.

Although one must use an orthogonalization scheme in conjunction with the Lanczos iteration, the added complexity is not prohibitive. Specifically, consider counting the number of floating point operations (flops) required to perform k iterations. We will assume that full orthogonalization is used and that the number of flops required for Λ_y-vector multiplies is linear in either the dimension m of the data or the dimension l of the estimate. Then, the only contribution to the flop count that is second order or higher in k, l, and m is from orthogonalization, $2mk^2$. For comparison, consider a direct method for computing the error variances that uses Gaussian elimination to invert the symmetric positive definite Λ_y. The flop count is dominated by the elimination, which requires $m^3/3$ flops [33, p. 146]. Thus, our algorithm typically provides a gain if $k < m/6$. For many estimation problems, a reasonable degree of accuracy is attained for $k \ll m$. Some examples are given in Section 3.6.

A summary of the steps outlined above to compute an approximation to the optimal linear least-squares estimate and associated estimation error variances is as follows.

Algorithm 3.1.1. *A Krylov Subspace Method for Static Estimation.*

1. *Initialize* $\hat{x}_0(y) = 0$, $(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}$ *for* $i = 1, \ldots, l$.

2. *Generate a random vector* s *to initialize the Lanczos iteration.*

3. *At each step* k,

(a) *compute the conjugate search direction p_k and filtered backprojection $b_{y,k}$ using a reorthogonalized Lanczos iteration, and*

(b) *update*

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + b_{y,k} p_k^T y \tag{3.17}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - ((b_{y,k})_i)^2 \quad \text{for } i = 1, \ldots, l. \tag{3.18}$$

3.2 Stopping Criteria

A stopping criterion is needed to determine when a sufficient number of iterations has been run to obtain an adequate approximation to the error variances. Two alternative stopping criteria are proposed in this section. The first is a simple scheme that we have found works well. However, there is no systematic method for setting the parameters of the criterion to guarantee that a specified level of accuracy is achieved. The second stopping criterion is a more complicated scheme for which one can establish bounds on the approximation error. However, the criterion tends to be overly conservative in establishing the number of iterations needed to achieve a specified level of accuracy.

3.2.1 Windowed-Maximal-Error Criterion

Under this first criterion, the algorithm stops iterating after k steps if

$$\tau_{k,\varepsilon_{\min}} \triangleq \max_{k-K_{\text{win}} \leq j \leq k} \max_i \frac{((b_{y,j})_i)^2}{\max((\Lambda_{e_x,k}(y))_{ii}, \varepsilon_{\min})} < \varepsilon_{\text{tol}} \tag{3.19}$$

where K_{win}, $\varepsilon_{\min}$, and ε_{tol} are parameters. This criterion guarantees that no components of the error variances have been altered over the last $K_{\text{win}} + 1$ iterations by more than ε_{tol} relative to the current approximation to the error variances. The motivation for this criterion is the analysis in Section 3.3 which suggests that the vectors $b_{y,k}$, representing the contribution to error reduction from $p_k^T y$, get smaller as k increases. However, this behavior is not always monotone; so, the criterion takes into account gains over a window of the last few iterations.

3.2.2 Noiseless-Estimation-Error Criterion

The second stopping criterion examines how well the Krylov subspace at the kth step, $\mathcal{K}(\Lambda_y, s, k-1)$, captures the significant components of the signal z, as defined in (3.1). As for the first stopping criterion, the motivation for the second criterion is Theorem 3.3.1 in Section 3.3. The theorem relates the optimal error covariance for estimating z from y, $\Lambda_{e_z}(y)$, to the optimal error covariance for estimating z from $p_1^T y, \ldots, p_k^T y$, $\Lambda_{e_z,k}(y)$. The implication is that as $\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y)$ gets smaller, the difference between $\Lambda_{e_x,k}(y)$ and $\Lambda_{e_x}(y)$ also decreases, albeit possibly at a slower rate. So, a relatively small difference between $\Lambda_{e_z,k}(y)$ and $\Lambda_{e_z}(y)$ implies a relatively small difference between $\Lambda_{e_x,k}(y)$ and

$\Lambda_{e_x}(y)$. This fact motivates the interest in efficiently computable bounds for $\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y)$. One such bound can be written, as follows, in terms of the error covariance for the noiseless estimation problem of estimating z from $p_1^T z, \dots, p_k^T z$.

Proposition 3.2.1. *Suppose $\Lambda_n = \sigma^2 I$ for $\sigma^2 > 0$. Let $\Lambda_{e_z,k}(z)$ be the optimal estimation error covariance for estimating z from $p_1^T z, \dots, p_k^T z$. Then, the difference between the error covariance for estimating z from y and z from $p_1^T y, \dots, p_k^T y$ is bounded by:*

$$\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y) \leq \Lambda_{e_z,k}(z) + f_k f_k^T \tag{3.20}$$

where

$$\|f_k\|^2 \leq \|\Lambda_z p_{k-1}\|^2 + \|\Lambda_z p_k\|^2 + \|\Lambda_z p_{k+1}\|^2 + \|\Lambda_z p_{k+2}\|^2. \tag{3.21}$$

Proof. The proof makes use of the Lanczos vectors q_i discussed at the end of Section 3.1. The Lanczos vectors are useful because they form bases for the Krylov subspaces, and they tri-diagonalize both Λ_y and Λ_z since $\Lambda_n = \sigma^2 I$, by assumption. The Lanczos vectors tri-diagonalizing Λ_y implies that $q_i^T y$ is correlated with $q_j^T y$ if and only if i and j differ by at most one. Let $\Lambda_{r_z,k+1}(y)$ denote the error reduction obtained from estimating z with $q_{k+2}^T y, q_{k+3}^T y, \dots$. Furthermore, let $\Lambda_{r_z,k+1}^{\perp}(y)$ denote the error reduction obtained from estimating z with the random variable formed by making $q_{k+1}^T y$ uncorrelated with $q_i^T y$ for $i \neq k+1$. Then,

$$\Lambda_{e_z}(y) - \Lambda_{e_z,k}(y) = \Lambda_{r_z,k+1}(y) + \Lambda_{r_z,k+1}^{\perp}(y). \tag{3.22}$$

Since y is simply a noisy version of z, $\Lambda_{r_z,k+1}(y) \leq \Lambda_{r_z,k+1}(z)$, where $\Lambda_{r_z,k+1}(z)$ is the error reduction obtained from estimating z with $q_{k+2}^T z, q_{k+3}^T z, \dots$. Furthermore, $\Lambda_{r_z,k+1}(z) \leq \Lambda_{e_z,k}(z)$ because $\Lambda_{e_z}(z) = 0$ and $q_i^T z$ is uncorrelated with $q_j^T z$ if i and j differ by more than one. Combining the last two inequalities with (3.22) yields

$$\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y) \leq \Lambda_{e_z,k}(z) + \Lambda_{r_z,k+1}^{\perp}(y). \tag{3.23}$$

The matrix $\Lambda_{r_z,k+1}^{\perp}(y)$ in (3.23) is bounded above by the optimal error reduction for estimating z from $q_k^T y$, $q_{k+1}^T y$, and $q_{k+2}^T y$ since $\Lambda_{r_z,k+1}^{\perp}(y)$ is the error reduction for an estimator that is linear in these three functionals of y. Furthermore, $\Lambda_{r_z,k+1}^{\perp}(y)$ is bounded above by the optimal error reduction for estimating z from $p_{k-1}^T y, \dots, p_{k+2}^T y$ since q_k, q_{k+1}, and q_{k+2} are linear combinations of $p_{k-1}, \dots, p_{k+2}$. Now, write the rank-one matrix $\Lambda_{r_z,k+1}^{\perp}(y)$ as $f_k f_k^T$. Then, the latter bound on $\Lambda_{r_z,k+1}^{\perp}(y)$ implies (3.21). □

Although Proposition 3.2.1 provides a bound on $\|f_k\|^2$, the argument in the proof suggests that the bound is very weak. Recall from the proof that $f_k f_k^T = \Lambda_{r_z,k+1}^{\perp}(y)$, the error reduction obtained for estimating z from the random variable formed by making

$q_{k+1}^T y$ uncorrelated with $q_k^T y$ and $q_{k+2}^T y$. Both q_k and q_{k+2}, as vectors from a Krylov subspace generated by Λ_y, are such that $q_k^T y$ and $q_{k+2}^T y$ are significantly correlated with z. Thus, making $q_{k+1}^T y$ uncorrelated with $q_k^T y$ and $q_{k+2}^T y$ will often significantly reduce the correlation of the resulting quantity with z. As a result, $\Lambda_{r_z,k+1}^{\perp}(y)$ is typically much smaller than the error reduction for estimating z from $q_{k+1}^T y$ alone, which, in turn, is smaller than the right-hand side of (3.21). Thus, the bound on $\|f_k\|^2$ is weak, and $\Lambda_{e_z,k}(z)$, the dominant term in (3.20), could be used alone as the basis of a stopping criterion.

One of the main advantages of the bound in Proposition 3.2.1 is that the diagonal elements of $\Lambda_{e_z,k}(z)$ are easily computable. As discussed in the proof of Proposition 3.2.1, the Lanczos vectors $q_1, q_2, \ldots$ generated by Algorithm 3.1.1 not only tri-diagonalize Λ_y, they tri-diagonalize Λ_z:

$$\begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix}^T \Lambda_z \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} = T_{z,k}. \tag{3.24}$$

Let $L_{z,k}$ be the lower bi-diagonal Cholesky factor of $T_{z,k}$, and let the vectors $r_1, r_2, \ldots$ be defined by

$$\begin{bmatrix} r_1 & r_2 & \cdots & r_k \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} L_{z,k}^{-T}. \tag{3.25}$$

Then, the linear functionals of the signal, $r_1^T z, r_2^T z, \ldots$ are white. So, a simple recursion can be used to compute $\Lambda_{e_z,k}(z)$:

$$(\Lambda_{e_z,k}(z))_{ii} = (\Lambda_{e_z,k-1}(z))_{ii} - ((b_{z,k})_i)^2 \tag{3.26}$$

with the initialization

$$(\Lambda_{e_z,0}(z))_{ii} = (\Lambda_z)_{ii} \tag{3.27}$$

where $i = 1, \ldots, m$ and $b_{z,k} = \Lambda_z r_k$. Note that $b_{z,k}$ can be computed without an additional multiply by Λ_z since Algorithm 3.1.1 computes $\Lambda_z q_i$. The computations for calculating $\Lambda_{e_z,k}(z)$ are summarized as follows:

Algorithm 3.2.1. *A Method for Calculating* $(\Lambda_{e_z,k})_{ii}$.

1. *Initialize* $(\Lambda_{e_z,0}(z))_{ii} = (\Lambda_z)_{ii}$.

2. *At each iteration* k:

   (a) *compute* $b_{z,k}$ *using* q_k *and the one-step recursion specified by* $L_{z,k}^T$, *and*

   (b) *update*

$$(\Lambda_{e_z,k}(z))_{ii} = (\Lambda_{e_z,k-1}(z))_{ii} - ((b_{z,k})_i)^2. \tag{3.28}$$

Stopping Algorithm 3.1.1 when a function of $(\Lambda_{e_z,k}(z))_{ii}$ falls below some threshold has a variety of advantages and disadvantages. Although it may appear that one of the main disadvantages is the requirement that Λ_n must be a multiple of the identity, this is not the case. There is an extension to the non-white case that makes use of preconditioning ideas, as discussed in Section 3.4. In fact, the main disadvantage stems from the bound in Proposition 3.2.1 being based on the noiseless estimation problem (*i.e.* $\Lambda_n = 0$). If Λ_n is not small, the bound may not be tight. Thus, a stopping criterion based on $\Lambda_{e_z,k}(z)$ may be conservative in determining the number of iterations needed to guarantee a specified level of accuracy. On the other hand, the bound is easy to compute and provides a good indication of the fraction of error reduction that has been attained by a specific iteration.

■ 3.3 The Main Convergence Result

In this section, we state the main convergence result. It establishes a bound on the rate at which the approximation to the error variances, in exact arithmetic, converges to the optimal estimation error variances. The result leads naturally to a consideration of the two acceleration techniques discussed in the next section. The proof of the main result is left for Section 3.5.

Establishing the convergence result requires making a few assumptions concerning the estimation problem and starting vector for the algorithm. The first is that the starting vector s in Algorithm 3.1.1 is a zero-mean Gaussian random vector. This assumption is needed to guarantee the independence of uncorrelated components of s. The covariance matrix of s, Λ_s, is assumed to equal Λ_y or be proportional to the identity. As regards the estimation problem for the purposes of this section, Λ_n is not necessarily a multiple of the identity. However, we do assume that Λ_y and Λ_z have the same eigenvectors $u_1, u_2, \ldots, u_m$ and that the corresponding eigenvalues $\lambda_{y,1} \geq \lambda_{y,2} \geq \cdots \geq \lambda_{y,m}$ and $\lambda_{z,1} \geq \lambda_{z,2} \geq \cdots \geq \lambda_{z,m}$ satisfy the inequality, $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$ for some $\sigma^2 > 0$ and sequence $\bar{\lambda}_i$. Note that both of these statements would hold for $\bar{\lambda}_i = \lambda_{z,i}$ if Λ_n were $\sigma^2 I$. The conditions are stated this generally because Λ_n may not be a multiple of the identity if some of the preconditioning techniques of Section 3.4.1 are used. We also assume that the eigenvalues of Λ_y are distinct and have a relative separation $(\lambda_{y,i} - \lambda_{y,i+1})/(\lambda_{y,i+1} - \lambda_{y,m})$ that is bounded below by a constant $\lambda_{\text{sep}} > 0$. Furthermore, the $\lambda_{y,i}$ are assumed to decrease slowly enough (not faster than a geometric decay) that one can find constants $\zeta > 0$ and $0 < \Gamma < 1$ of reasonable magnitude (ζ not much larger than $\|\Lambda_y\|$) for which $1/(\lambda_{y,k}\gamma^k) < \zeta\Gamma^k$, where

$$\gamma \triangleq 1 + 2(\lambda_{\text{sep}} + \sqrt{\lambda_{\text{sep}} + \lambda_{\text{sep}}^2}). \tag{3.29}$$

This last assumption is a very weak assumption that is almost never violated. All of these assumptions concerning the estimation problem are not restrictive because they can be guaranteed using appropriate preconditioning techniques, as described in Section 3.4. The assumptions are summarized as follows.

Assumptions

1. The starting vector s in Algorithm 3.1.1 is a zero-mean Gaussian random vector, and $\Lambda_s = \Lambda_y$ or $\Lambda_s \propto I$,
2. There exist constants $\zeta > 0$ and $0 < \Gamma < 1$ such that $1/(\lambda_{y,k}\gamma^k) < \zeta\Gamma^k$,
3. Λ_y and Λ_z have the same eigenvectors,
4. There exists a constant $\sigma^2 > 0$ and a sequence $\bar{\lambda}_i$ such that $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$,
5. There exists a constant $\lambda_{\text{sep}} > 0$ such that $(\lambda_{y,i}-\lambda_{y,i+1})/(\lambda_{y,i+1}-\lambda_{y,m}) \geq \lambda_{\text{sep}} > 0$.

These assumptions lead to the main convergence result, as stated next in Theorem 3.3.1. The theorem consists of two bounds, one concerning the error variances for estimating x and one, the error variances for estimating only the measured components of x, $z = Cx$. Two bounds are given because one may need fewer iterations to obtain a good estimate of z than of x. Moreover, the rate of convergence of the error variance for estimating z is of interest since z is often a subsampled version of x.[2]

Theorem 3.3.1. *If Assumptions 1-5 hold, then*

$$\sum_{j=1}^{m}(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{jj} \leq \frac{\|s\|^2\zeta\eta\|\Lambda_x\|\|\Lambda_y\|}{\sigma^2(1-\frac{1}{\gamma^2})(1-\frac{1}{\sqrt[4]{\gamma}})}\gamma^{-k/4} + \frac{\|\Lambda_x\|}{\sigma^2}\sum_{i=k}^{m-1}(i-k+4)\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor} \tag{3.30}$$

and

$$\sum_{j=1}^{m}(\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{jj} \leq \frac{\|s\|^2\zeta\eta\|\Lambda_y\|}{(1-\frac{1}{\gamma^2})(1-\frac{1}{\sqrt{\gamma}})}\gamma^{-k/2} + \sum_{i=k}^{m-1}(i-k+4)\min\left(\frac{\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\lambda_{z,\lfloor\frac{i}{4}\rfloor}}{\sigma^2}, \bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\right), \tag{3.31}$$

where γ is given by (3.29) and η is a random variable whose statistics depend only on λ_{sep}, γ, and Γ.

The bounds in Theorem 3.3.1 provide a characterization of the difference between the optimal error variances and the computed approximation. The bounds are a sum of two terms. The first terms on the right-hand sides of (3.30) and (3.31) characterize how well the Krylov subspaces have captured the dominant components of Λ_y. The bigger λ_{sep} is, the larger γ is, and the smaller the first terms in (3.30) and (3.31) become. Thus, the more separated the eigenvalues (as measured by λ_{sep}), the better

[2] That the two bounds differ is a consequence of the fact that, for a given number of iterations k, we are not computing the best k linear functionals of the data for estimating x.

the algorithm will perform. The second term is a sum of bounds $\bar{\lambda}_i$ on the ratio of eigenvalues $\lambda_{z,i}/\lambda_{y,i}$. The sum is over those $\bar{\lambda}_i$ corresponding to eigenvectors of Λ_z that are not well-captured by the Krylov subspaces at step k. Note that the sum is over the more rapidly decreasing $\bar{\lambda}_{\lfloor i/4 \rfloor}\lambda_{z,\lfloor i/4 \rfloor}$ in (3.31).

The bounds are useful principally for two reasons. First, they indicate how the errors will scale as s, σ^2, $\|\Lambda_x\|$, $\|\Lambda_y\|$, and the eigenvalues of Λ_z change. In particular, note that the only dependence on the starting vector s is through the norm $\|s\|$. Thus, the performance of the algorithm does not depend strongly on s. Second, the bounds indicate that the rate of convergence can be increased by transforming the estimation problem in order to make γ big enough so that the second terms in (3.30) and (3.31) dominate. Such transformations are discussed next in Section 3.4.1.

■ 3.4 Techniques for improving convergence properties

This section presents two different techniques for improving the convergence properties of the proposed algorithm for computing error variances. These techniques can be used to guarantee convergence in the case that a given estimation problem violates any of the assumptions of Theorem 3.3.1. One can also use these techniques to increase γ so as to improve the theoretical convergence rates.

■ 3.4.1 Preconditioning

In the estimation context, preconditioning consists of determining an invertible transformation B such that estimating x from the transformed data By can be theoretically done more efficiently by the proposed algorithm than estimating x directly from y. This will be the case if the covariances of the transformed data, $B\Lambda_y B^T$, and of the transformed signal, $B\Lambda_z B^T$, satisfy Assumptions 3 and 5 of Theorem 3.3.1 but Λ_y and Λ_z don't. The convergence properties will also be improved if γ for the transformed problem is higher than for the untransformed problem. The principal novelty of the preconditioning approaches described here is that they focus on these particular goals, which are very different than those of standard CG preconditioning and differ significantly from those of preconditioning for eigenvector algorithms [71, Chapter 8]. Although the goals of the preconditioning discussed here are different than for standard CG, the implementation details are very similar. In particular, explicit specification of a transformation B is not necessarily required for preconditioning techniques because preconditioning can be implemented in such a way that only $B^T B$-vector multiplies are needed instead of B- and B^T-vector multiplies.

There are three different implementations of preconditioning, each of which is mathematically equivalent in exact arithmetic. Symmetric preconditioning simply consists of applying the Krylov subspace algorithm to estimating x from $By = BCx + Bn$. Essentially, x is estimated given linear functionals from Krylov subspaces $\mathcal{K}(B\Lambda_y B^T, Bs, k)$ applied to By. There are also left and right preconditioning techniques. The following discussion focuses on right preconditioning, and analogous statements can be made concerning left preconditioning. Right preconditioning differs from symmetric preconditioning in that it involves estimating x given linear functionals from the Krylov subspaces $\mathcal{K}(\Lambda_y B^T B, s, k)$ applied to $B^T By$. Note that this is equivalent to the estimation performed in the case of symmetric preconditioning. Although $\Lambda_y B^T B$ is not symmetric, it is self-adjoint with respect to the $B^T B$ inner product. As in Algorithm 3.1.1, we do not compute the conjugate search directions for the preconditioned estimation problem using a standard preconditioned CG iteration. Instead, we use Lanczos iterations that compute a series of $B^T B$-conjugate vectors that tri-diagonalize $B^T B \Lambda_y B^T B$, as follows:

$$\alpha_k = t_k^T \Lambda_y t_k \tag{3.32}$$

$$h_k = \Lambda_y t_k - \alpha_k q_k - \beta_k q_{k-1} \tag{3.33}$$

$$d_k = B^T B h_k \tag{3.34}$$

$$\beta_{k+1} = \sqrt{d_k^T h_k} \tag{3.35}$$

$$q_{k+1} = \frac{h_k}{\beta_{k+1}} \tag{3.36}$$

$$t_{k+1} = \frac{d_k}{\beta_{k+1}} \tag{3.37}$$

where $t_1 = B^T Bs$, $q_1 = s$, $q_0 = 0$, and $\beta_1 = 0$. The q_k are the $B^T B$-conjugate Lanczos vectors that tri-diagonalize $B^T B \Lambda_y B^T B$, and the $t_k = B^T B q_k$ tri-diagonalize Λ_y. This latter tri-diagonal matrix can be factored, as in Algorithm 3.1.1, to compute the Λ_y-conjugate search directions p_k. The only difference is that the t_k replace the q_k in (3.14) and (3.15). Moreover, one can compute the filtered backprojected search directions $b_{y,k} = \Lambda_x C^T p_k$ as a by-product. Overall, the steps of the preconditioned Krylov subspace algorithm are the same as those in Algorithm 3.1.1 except that a preconditioned Lanczos iteration replaces the normal Lanczos iteration. Note that the Lanczos method for tri-diagonalizing a left-preconditioned system is the same as the generalized Lanczos algorithm for solving generalized eigenvalue problems [64, §15.11]. What follows are some examples of preconditioners in squared up form, $B^T B$, that one can consider using in various contexts.

One choice for a preconditioner when the noise covariance Λ_n is not a multiple of the identity but is invertible is to choose $B^T B = \Lambda_n^{-1}$. This choice of preconditioner will effectively shape the noise covariance to be a multiple of the identity. The transformed data covariance, $B\Lambda_y B^T$, and signal covariance, $B\Lambda_z B^T$, will then satisfy Assumption 3. Multiplying a vector by Λ_n^{-1} is often easy because Λ_n is often diagonal.

If the noise covariance is, or has been transformed to be, a multiple of the identity, one can consider preconditioners that will maximally separate the eigenvalues of Λ_y. Such preconditioners can guarantee that the transformed data covariance, $B\Lambda_y B^T$, satisfies Assumption 5 and can increase γ to improve the bounds in Theorem 3.3.1. Note that such preconditioning will do little to change the bound $\bar{\lambda}_i$ on $\lambda_{z,i}/\lambda_{y,i}$ in Assumption 4 because the preconditioner will transform both $\lambda_{z,i}$ and $\lambda_{y,i}$. The ideal preconditioner would simply operate on the spectrum of Λ_y and force a geometric decay in the eigenvalues to the noise level σ^2. The geometric decay guarantees a constant relative separation in the eigenvalues as measured by the ratio in Assumption 5. However, operating on the spectrum is difficult because one doesn't know the eigendecomposition of Λ_y. When the rows of C are orthogonal (which is often the case in the applications mentioned in the introduction) and the eigendecomposition of Λ_x is known, one practical preconditioner is the following. Let Λ_p be a matrix whose eigenvectors are the same as those of Λ_x and whose eigenvalues decay geometrically. Then, let the preconditioner be given by $B^TB = C\Lambda_pC^T$. Although this preconditioner has worked well in practice, as described in Section 3.6, we have no theoretical results concerning the properties of the transformed estimation problem.

One can use extensions of each of the stopping criteria of Section 3.2 in conjunction with preconditioning; however, the preconditioner must satisfy certain assumptions for the extension of the noiseless-estimation stopping criterion of Section 3.2.2 to be used. What follows is a discussion of the extension and the underlying assumptions concerning the preconditioner for the right-preconditioned case. Recall that the discussion in Section 3.2.2 assumes that the noise covariance is a multiple of the identity. This assumption ensures that the Lanczos vectors tri-diagonalize both Λ_y and Λ_z so that one can compute $\Lambda_{e_z,k}(z)$ efficiently. Now, suppose one is using a preconditioning transformation B. Let $\Lambda_{n'} = \Lambda_n - (B^TB)^{-1}$. Assume that $\Lambda_{n'}$ is positive semi-definite so that it is a valid covariance matrix. Let n' be a random vector with covariance $\Lambda_{n'}$ and uncorrelated with z. Then, $z' = z + n'$ has covariance $\Lambda_{z'} = \Lambda_z + \Lambda_{n'}$. One can compute $\Lambda_{e_z,k}(z')$ efficiently because the t_k in (3.32)-(3.37) tri-diagonalize both Λ_y and $\Lambda_{z'}$. For $\Lambda_{e_z,k}(z')$ to be useful, the pseudo-signal z' should not have any significant components not in z. Note that an example of a preconditioner satisfying the above two assumptions is given by $B^TB = \Lambda_n^{-1}$. For this preconditioner, $\Lambda_{n'} = 0$; so, $\Lambda_{e_z,k}(z) = \Lambda_{e_z,k}(z')$. Thus, one can use $\Lambda_{e_z,k}(z')$ as part of a stopping criterion in conjunction with preconditioning provided that the preconditioner satisfies the two assumptions outlined above.

■ 3.4.2 Using multiple starting vectors

Another technique for improving convergence properties in the case where Λ_y has repeated eigenvalues is to use a block form of Algorithm 3.1.1. Block Krylov subspace algorithms have been developed for other computations, particularly eigendecompositions [33, §9.2.6]. The principal novelty of the algorithm we present here is the application to estimation.

Now, consider the subspace spanned by the columns of

$$[S \quad \Lambda_y S \quad \Lambda_y^2 S \quad \cdots \quad \Lambda_y^{k-1} S] \tag{3.38}$$

where S is an $m \times r$ matrix of independent identically distributed random starting vectors whose marginal statistics satisfy the restrictions for Algorithm 3.1.1 starting vectors. Denote this subspace by $\mathcal{K}(\Lambda_y, S, k)$. Then, one can consider forming

$m \times r$ matrices $P_1, \dots, P_k$ whose columns form bases for $\mathcal{K}(\Lambda_y, S, k)$ and which satisfy $P_i^T \Lambda_y P_j = \delta_{ij} I$. As for the single starting vector case in Section 3.1, the LLSE of x given the random vectors $P_1^T y, \dots, P_k^T y$ and the associated error variances can be computing using a recursion:

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + B_{y,k} P_k^T y \tag{3.39}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - \sum_{j=1}^{r} ((B_{y,k})_{ij})^2 \tag{3.40}$$

with initial conditions

$$\hat{x}_0(y) = 0 \tag{3.41}$$

$$(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii} \tag{3.42}$$

where $i = 1, \dots, l$ and $B_{y,k} = \Lambda_x C^T P_k$.

The P_i and $B_{y,i}$ can be computed using a reorthogonalized block Lanczos algorithm [33, §9.2.6]. The block Lanczos iteration generates, according to the following recursions, a sequence of orthogonal matrices Q_i that are orthogonal to each other:

$$A_k = Q_k^T \Lambda_y Q_k \tag{3.43}$$

$$H_k = \Lambda_y Q_k - Q_k A_k - Q_{k-1} R_k \tag{3.44}$$

$$Q_{k+1} R_{k+1} = H_k \quad \text{(QR factorization of } H_k\text{)} \tag{3.45}$$

where Q_1 and R_1 are a QR factorization of the starting matrix S, and $Q_0 = 0$. The Q_i block tri-diagonalize Λ_y; so, one can write

$$\begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix}^T \Lambda_y \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} = T_{y,k} \tag{3.46}$$

where $T_{y,k}$ is a block tri-diagonal matrix. Let $L_{y,k}$ be the lower block bi-diagonal Cholesky factor of $T_{y,k}$. Then, the P_i are defined by

$$\begin{bmatrix} P_1 & \cdots & P_k \end{bmatrix} \triangleq \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} L_{y,k}^{-T}. \tag{3.47}$$

Thus, the P_i can be computed from the Q_i using a one-step recursion. Moreover, the $B_i = \Lambda_x C^T P_i$ can be computed as a by-product, as with a single starting vector.

As for the single starting vector case in Section 3.1, the block Lanczos iteration must be combined with some form of reorthogonalization. Unlike the previous case, however, there are not as many methods for reorthogonalizing the block Lanczos iteration. Full orthogonalization is very common and is the method we have used. This simply recomputes H_k as

$$H_k := H_k - \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix}^T H_k \tag{3.48}$$

between steps (3.43) and (3.44)

The algorithm is summarized as follows.

Algorithm 3.4.1. *A Block Krylov Subspace Method for Static Estimation.*

1. *Initialize* $\hat{x}_0(y) = 0$, $(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}$ *for* $i = 1, \dots, l$.
2. *Generate a random* $m \times r$ *matrix* S *to initialize the block Lanczos iteration.*
3. *At each step* k,
   (a) *compute the block of search directions* P_k *and filtered backprojections* $B_{y,k}$ *using a reorthogonalized block Lanczos iteration, and*
   (b) *update*

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + B_{y,k}P_k^T y \tag{3.49}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - \sum_{j=1}^{r}((B_{y,k})_{ij})^2 \quad \textit{for } i = 1, \dots, l. \tag{3.50}$$

The advantage of using the block form is that there may be small angles between the subspaces $\mathcal{K}(\Lambda_y, S, k)$ and multiple orthogonal eigenvectors of Λ_y associated with the same repeated eigenvalue, even in exact arithmetic. This is because each of the columns of S may have linearly independent projections onto the eigenspace associated with a repeated eigenvalue. The following theorem establishes convergence rates for the block case when there may be repeated eigenvalues. It is an extension of Theorem 3.3.1 to the block case. The proof of both theorems are very similar; so, the proof of Theorem 3.4.1 is omitted here but provided in Appendix A.

Theorem 3.4.1. *Suppose that*

1. There exists a constant $\lambda_{sep,r} > 0$ *such that* $(\lambda_{y,i} - \lambda_{y,i+r})/(\lambda_{y,i+r} - \lambda_{y,m}) \geq \lambda_{sep,r}$.

2. There exist constants $\zeta > 0$ *and* $0 < \Gamma < 1$ *such that* $1/(\lambda_{y,i}\gamma_r^i) < \zeta\Gamma^i$ *where*

$$\gamma_r \triangleq 1 + 2(\lambda_{sep,r} + \sqrt{\lambda_{sep,r} + \lambda_{sep,r}^2}). \tag{3.51}$$

3. Λ_y *and* Λ_z *have the same eigenvectors,*

4. There exists a constant $\sigma^2 > 0$ *and a sequence* $\bar{\lambda}_i$ *such that* $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$

5. $(\lambda_{y,i} - \lambda_{y,i_+})/(\lambda_{y,i_+} - \lambda_{y,m})$ *is bounded away from zero, where* i_+ *is the index of the next smallest distinct eigenvalue of* Λ_y *after* i, *and*

Then,

$$\sum_{j=1}^{m}(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{jj} \leq \frac{\eta\|\Lambda_x\|\|\Lambda_y\|}{\sigma^2(1 - \frac{1}{\gamma_r^2})(1 - \frac{1}{\sqrt[4]{\gamma_r}})}\gamma_r^{-k/4} + \frac{\|\Lambda_x\|}{\sigma^2}\sum_{i=k}^{m-1}(i - k + 4)\bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \tag{3.52}$$

and

$$\sum_{j=1}^{m}(\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{jj} \leq \frac{\eta\|\Lambda_y\|}{(1-\frac{1}{\gamma_r^2})(1-\frac{1}{\sqrt{\gamma_r}})}\gamma_r^{-k/2} + \sum_{i=k}^{m-1}(i-k+4)\min\left(\frac{\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\lambda^2_{z,\lfloor\frac{i}{4}\rfloor}}{\sigma^2}, \bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\right), \quad (3.53)$$

where the statistics of the random variable η depend on the starting matrix S.

There are two key differences between the statements of Theorems 3.3.1 and 3.4.1. The first addresses the possibility of repeated eigenvalues. Specifically, the bounds in Theorem 3.4.1 depend on the eigenvalue separation through $\lambda_{\text{sep,r}}$, which measures the relative separation between eigenvalues whose indices differ by r. Thus, the proposition establishes a convergence rate in the case where there may be groups of up to r repeated or clustered eigenvalues. The second key difference is that the bounds in Theorem 3.4.1 may have a strong dependence on the starting matrix through S. This contrasts with the bounds in Theorem 3.3.1 which depend on the starting vector s only through the norm $\|s\|$. However, our numerical results have not indicated that the block algorithm's performance depends strongly on the starting matrix S.

One can use natural extensions of the preconditioning techniques and either of the stopping criteria of Section 3.2 with Algorithm 3.4.1. Thus, Algorithm 3.4.1 is a simple replacement for Algorithm 3.1.1 that can be used to obtain better convergence properties when Λ_y has repeated eigenvalues.

■ 3.5 Convergence Analysis

The bounds in Theorem 3.3.1 are proved in this section in several steps. The first few steps place bounds on the norms of the filtered backprojected conjugate search directions, $\|\Lambda_x C^T p_i\|$ and $\|C\Lambda_x C^T p_i\|$. The bounds are proved using Saad's convergence theory for the Lanczos algorithm [70]. These bounds are stated in terms of an extremum of independent random variables. The extremum arises because the starting vector affects the angles between the Krylov subspaces and the dominant components of Λ_y. However, we prove that the extremum is part of a sequence of extrema that are converging in probability to a finite random variable (η in Theorem 3.3.1). Thus, the starting vector has no strong effect on the quality of the approximation to the error variances. This result is the principal novelty of our convergence analysis. After establishing the convergence of the extrema, we prove Theorem 3.3.1.

■ 3.5.1 Bounds on the filtered backprojected search directions

One is interested in bounding the norms of the filtered backprojected search directions because the quality of the approximation to the error variances depends on the norms as follows:

$$\sum_{j=1}^{l}(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{jj} = \sum_{i=k+1}^{l} \|\Lambda_x C^T p_i\|^2 \tag{3.54}$$

$$\sum_{j=1}^{l}(\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{jj} = \sum_{i=k+1}^{l} \|C\Lambda_x C^T p_i\|^2. \tag{3.55}$$

Proposition 3.5.1. *Write the conjugate search directions in the basis of eigenvectors of Λ_y, as follows:*

$$p_i = v_{i,1}u_1 + \cdots + v_{i,m}u_m. \tag{3.56}$$

Then

$$\|\Lambda_x C^T p_i\|^2 \leq \|\Lambda_x\| \sum_{j=1}^{m} \lambda_{z,j} v_{i,j}^2, \tag{3.57}$$

and

$$\|C\Lambda_x C^T p_i\|^2 = \sum_{j=1}^{m} \lambda_{z,j}^2 v_{i,j}^2. \tag{3.58}$$

Proof. $\|\Lambda_x C^T p_i\|^2 \leq \|\Lambda_x\| \|\Lambda_x^{1/2} C^T p_i\|^2 = \|\Lambda_x\| \sum_{j=1}^{m} \lambda_{z,j} v_{i,j}^2$. This proves the first inequality. The second inequality follows from Parseval's Theorem. □

As we now show, one can bound the coefficients $v_{i,j}$ in terms of $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$, where π_i is the operator that produces the orthogonal projection onto $\mathcal{K}(\Lambda_y, s, i)$ with respect to the standard inner-product.

Proposition 3.5.2. *Write $p_i = v_{i,1}u_1 + \cdots + v_{i,m}u_m$ as in Proposition 3.5.1. Then,*

$$|v_{i+1,j}| \leq \frac{\|\Lambda_y\|^{1/2}}{\lambda_{y,j}} \frac{\|(I - \pi_i)u_j\|}{\|\pi_i u_j\|}. \tag{3.59}$$

Proof. Note that

$$\begin{aligned} \lambda_{y,j}|v_{i+1,j}| &= |p_{i+1}^T \Lambda_y u_j| \\ &= |p_{i+1}^T \Lambda_y \pi_i u_j + p_{i+1}^T \Lambda_y (I - \pi_i) u_j| \\ &= |p_{i+1}^T \Lambda_y (I - \pi_i) u_j| \end{aligned} \tag{3.60}$$

since p_{i+1} is Λ_y-conjugate to vectors in the range of π_i. Thus, $\lambda_{y,j}|v_{i+1,j}| \leq \|\Lambda_y p_{i+1}\| \cdot \|(I-\pi_i)u_j\| \leq \|\Lambda_y\|^{1/2}\|(I-\pi_i)u_j\|$ because of the Cauchy-Schwartz inequality and the fact that p_{i+1} is Λ_y-normal. The inequality in (3.59) then follows from $\|\pi_i u_j\| \leq 1$. □

The bound in Proposition 3.5.2 can be refined. In particular, a theorem due to Saad [70, Theorem 1] implies the following result concerning the ratio $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$, which we state without proof.

Theorem 3.5.1. *Let γ be defined by (3.29) and K_j, by*

$$K_j \triangleq \begin{cases} \prod_{k=1}^{j-1} \frac{\lambda_{y,k}-\lambda_{y,m}}{\lambda_{y,k}-\lambda_{y,j}}, & \text{if } j \neq 1 \\ 1 & \text{if } j = 1. \end{cases} \tag{3.61}$$

Then,

$$\frac{\|(I-\pi_i)u_j\|}{\|\pi_i u_j\|} \leq \frac{2K_j}{\gamma^{i-j}} \frac{1}{\|\pi_1 u_j\|}. \tag{3.62}$$

Recall, from the definition of angles between subspaces given in Section 3.1, that $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$ is the tangent of the angle between the Krylov subspace $\mathcal{K}(\Lambda_y, s, i)$ and the eigenvector u_j. Theorem 3.5.1 bounds the rate at which these angles decrease as the subspace dimension i increases. The bound has three components. The rate of decay is γ, the relative separation between eigenvalues as defined in (3.29). The constant in the numerator, $2K_j$, depends on the eigenvalues according to (3.61). The numerator, $\|\pi_1 u_j\|$, is the norm of the projection of the starting vector, s, onto u_j. The primary importance of the theorem is that is establishes the decay rate γ.

One can refine the bound in Proposition 3.5.2 by splitting the coefficients $v_{i,j}$ into two groups: those that are getting small by Proposition 3.5.2 and Theorem 3.5.1 and those that may be large but do not significantly affect $\|\Lambda_x C^T p_i\|$ because the corresponding eigenvalues of Λ_z are small. This idea leads to the following proposition.

Proposition 3.5.3.

$$\|\Lambda_x C^T p_{i+1}\|^2 \leq 4\|\Lambda_x\|\|\Lambda_y\| \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor -1} K_j^2 \frac{1}{\gamma^{2(i-j)}\|\pi_1 u_j\|^2} \frac{\lambda_{z,j}}{\lambda_{y,j}^2} + \|\Lambda_x\| \sum_{j=\lfloor \frac{i}{4} \rfloor}^{\infty} \frac{\lambda_{z,j}}{\lambda_{y,j}}, \tag{3.63}$$

and

$$\|C\Lambda_x C^T p_{i+1}\|^2 \leq 4\|\Lambda_y\| \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor -1} K_j^2 \frac{1}{\gamma^{2(i-j)}\|\pi_1 u_j\|^2} \frac{\lambda_{z,j}^2}{\lambda_{y,j}^2} + \sum_{j=\lfloor \frac{i}{4} \rfloor}^{\infty} \frac{\lambda_{z,j}^2}{\lambda_{y,j}}. \tag{3.64}$$

Proof. The first term in each of (3.63) and (3.64) follows immediately from Propositions 3.5.1 and 3.5.2 and Theorem 3.5.1. The second term follows from Proposition 3.5.1 and the fact that $p_{i+1}^T \Lambda_y p_{i+1} = \sum_{j=1}^{m} \lambda_{y,j} v_{i+1,j}^2 = 1$. □

The first terms in the bounds of Proposition 3.5.3 may get large if $1/(\gamma^i \|\pi_1 u_j\|^2)$ or K_j are not well-behaved. However, the standing assumptions concerning the eigenvalues of Λ_y, Λ_z, and Λ_s imply that K_j and $1/(\gamma^i \|\pi_1 u_j\|^2)$ are bounded by quantities of a reasonable magnitude, as we now show.

3.5.2 Convergence of infinite products and extrema of independent sequences

The main result regarding the convergence of infinite products and extrema of independent sequences is the following.

Proposition 3.5.4. *Let $F_i(v)$, $i = 1, 2, \dots,$ be a sequence of functions such that:*

1. $1 - F_i(v)$ *is a cumulative distribution function,* i.e. *right-continuous and monotonically increasing from zero to one,*

2. *For every interval $[V, \infty)$ over which $1 - F_i(v)$ are positive, there exists a constant $A(V)$ and an absolutely summable sequence $\bar{F}_i(V)$ such that $F_i(V) \leq \bar{F}_i(V) \leq A(V) < 1$ $\forall i$; and*

3. $\lim_{v \to \infty} \sum_{i=1}^{\infty} F_i(v) = 0$.

Then, $F(v) = \prod_{i=1}^{\infty}(1 - F_i(v))$ is a distribution function. Moreover, $F(v)$ is positive over every interval such that $1 - F_i(v)$ is positive $\forall i$.

Proof. For $F(v)$ to be a distribution function, it must be right-continuous and monotonically increasing from zero to one.

Consider the interval $[V, \infty)$. Now, $\sum_{i=1}^{I} \log(1 - F_i(v))$ is right-continuous for each I since each $F_i(v)$ is right-continuous. Furthermore,

$$\left|\log(F(v)) - \sum_{i=1}^{I} \log(1 - F_i(v))\right| = \left|\sum_{i=I+1}^{\infty} \log(1 - F_i(v))\right| \leq \left|\sum_{i=I+1}^{\infty} \log(1 - \bar{F}_i(V))\right|$$

$$= \left|\sum_{i=I+1}^{\infty} \sum_{j=1}^{\infty} \frac{\bar{F}_i^j(V)}{j}\right| \leq \left|\sum_{i=I+1}^{\infty} \frac{\bar{F}_i(V)}{1 - A(V)}\right|. \tag{3.65}$$

Since $\bar{F}_i(V)$ is absolutely summable, $\sum_{i=1}^{I} \log(1 - F_i(v))$ converges to $\log(F(v))$ uniformly for $v \in [V, \infty)$. Thus, $\log(F(v))$ and, in turn, $F(v)$ are right-continuous.

That $F(v)$ is monotonic follows from the monotonicity of the $1 - F_i(v)$. Now, $\lim_{v \to -\infty} F(v) = 0$ since $\lim_{v \to -\infty}(1 - F_1(v)) = 0$. Moreover,

$$\lim_{v \to \infty} \log(F(v)) \geq \lim_{v \to \infty} \sum_{i=1}^{\infty} \frac{-F_i(v)}{1 - A(V)} = 0, \tag{3.66}$$

where V is such that $1 - F_i(v)$ is positive over $[V, \infty)$ $\forall i$. So, $\lim_{v \to \infty} F(v) = 1$.

Furthermore, if $1 - F_i(v)$ is positive $\forall i$ over an interval $[V, \infty)$, then

$$\log(F(v)) \geq \sum_{i=1}^{\infty} \frac{-\bar{F}_i(V)}{1 - A(V)} > -\infty. \tag{3.67}$$

Hence, $F(v)$ is positive over every interval such that $1 - F_i(v)$ is positive $\forall i$. □

A particular example of such a sequence of functions $F_i(v)$ satisfying the assumptions of Proposition 3.5.4 is

$$F_i(v) = \begin{cases} 1 & v < 0 \\ (1-v)^i & 0 \leq v \leq 1 \\ 0 & v > 1. \end{cases} \tag{3.68}$$

Thus, any product of numbers converging geometrically fast towards one is bounded away from zero, and the product is continuously varying from zero to one as the geometric rate changes from one to zero. This fact is used in the proof of the following proposition, which bounds the constants K_j.

Proposition 3.5.5. *There exists a function $K(v)$ which is continuous and monotonically decreasing from infinity to one as v ranges from zero to infinity and satisfies*

$$K_j \leq K(\lambda_{sep}). \tag{3.69}$$

Proof.

$$\begin{aligned} \frac{1}{K_j} &= \prod_{k=1}^{j-1} \frac{\lambda_{y,k} - \lambda_{y,j}}{\lambda_{y,k} - \lambda_{y,m}} \\ &\geq \prod_{k=1}^{j-1} \left(1 - \left(\frac{1}{1+\lambda_{\text{sep}}}\right)^k\right) \end{aligned} \tag{3.70}$$

where the inequality follows from Assumption 5. By Proposition 3.5.4, the product is monotonically decreasing to a limit as j tends to infinity. The limit is a continuous function of λ_{sep} that varies monotonically from zero to one as λ_{sep} increases from zero to infinity. Denote the limit by $1/K(\lambda_{\text{sep}})$. Then, $K_j \leq K(\lambda_{\text{sep}})$, as desired. □

The bound on $1/(\gamma^i\|\pi_1 u_j\|^2)$ is stochastic because $\pi_1 = s^T/\|s\|$, where s is the starting vector. By Assumption 1, one can write $\|\pi_1 u_j\|^2 = \lambda_{s,j}|w_j|^2/\|s\|^2$, where $\lambda_{s,j}$ are eigenvalues of Λ_s and w_j are independent, zero mean, unit variance Gaussian random variables. Thus,

$$\frac{1}{\gamma^i\|\pi_1 u_j\|^2} \leq \|s\|^2 \max_{1\leq k\leq m} \frac{1}{\lambda_{s,k}\gamma^k|w_k|^2}, \tag{3.71}$$

for $m \geq i \geq j$. Suppose that the $\lambda_{y,k}$ satisfy

$$\frac{1}{\lambda_{y,k}\gamma^k} < \zeta\Gamma^k \tag{3.72}$$

for constants $\zeta > 0$ and $0 < \Gamma < 1$. Then, (3.72) holds for $\lambda_{s,k}$ for the same ζ and Γ if $\Lambda_s = \Lambda_y$ and for a different ζ and $\Gamma = 1/\gamma$ if $\Lambda_s \propto I$. Let

$$\mu_k = \max_{1 \leq j \leq k} \frac{\Gamma^j}{|w_j|^2}. \tag{3.73}$$

The quantity μ_k is an extremum of an independent, non-identically distributed sequence of random variables. Bounding the rate at which extrema grow is a classic problem in statistics [51]. The following result states that the μ_k don't grow without bound but converge in probability.

Proposition 3.5.6. *Suppose $w_1, w_2, w_3, \ldots$ is an independent sequence of zero mean, unit variance Gaussian random variables. Let μ_k be as in (3.73). Then, the μ_k converge in probability to a finite-valued random variable.*

Proof. First, we show the μ_k converge in distribution.

$$\mathrm{P}\{\mu_k \leq M\} = \prod_{i=1}^{k} \mathrm{P}\left\{|w_i| \geq \sqrt{\frac{\Gamma^i}{M}}\right\}. \tag{3.74}$$

Let

$$F_i(M) = \mathrm{P}\left\{|w_i| \geq \sqrt{\frac{\Gamma^i}{M}}\right\}. \tag{3.75}$$

Then,

$$F_i(M) \leq \sqrt{\frac{2}{\pi}}\sqrt{\frac{\Gamma^i}{M}}, \tag{3.76}$$

which satisfy the conditions of Proposition 3.5.4. Thus, $\lim_{k\to\infty} \mathrm{P}\{\mu_k \leq M\} = F(M)$, for some distribution function F.

To show that the μ_k converge in probability, consider the following. For $n > k$ and $\varepsilon > 0$,

$$\mathrm{P}\{\mu_n - \mu_k > \varepsilon\} = \int \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) \tag{3.77}$$

where G_k is the distribution of μ_k. Now,

$$\begin{aligned} \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} &= \mathrm{P}\left\{\max_{1 \leq j \leq n-k+1} \frac{\Gamma^j}{|w_j|^2} > \frac{\varepsilon + v}{\Gamma^{k-1}}\right\} \\ &\leq 1 - F\left(\frac{\varepsilon + v}{\Gamma^{k-1}}\right). \end{aligned} \tag{3.78}$$

Let V be such that $1 - F(V) < \varepsilon/2$ and N such that

$$1 - F\left(\frac{\varepsilon + v}{\Gamma^{k-1}}\right) < \frac{\varepsilon}{2} \quad \text{for } k \geq N. \tag{3.79}$$

For $n > k \geq N$,

$$\begin{aligned}\int \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) = \int_0^V \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) + \\ \int_V^\infty \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) \\ \leq \int_0^V \frac{\varepsilon}{2} dG_k(v) + \int_V^\infty dG_k(v) < \varepsilon.\end{aligned} \tag{3.80}$$

Thus, the μ_k satisfy the Cauchy criterion and converge in probability to a random variable whose distribution function is F [26, pp.226-7]. □

3.5.3 Proof of Theorem 3.3.1

The results of the preceding two subsections combine to form a proof of Theorem 3.3.1, as follows.

Proof. By Propositions 3.5.3 and 3.5.5,

$$\begin{aligned}\sum_{j=1}^{m} (\Lambda_{e_x}(p_1^T y, \ldots, p_k^T y))_{jj} - (\Lambda_{e_x}(y))_{jj} = \sum_{i=k+1}^{m} \|\Lambda_x C^T p_i\|^2 \\ \leq 4\|\Lambda_x\|\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}}) \zeta \mu_m \sum_{i=k}^{m-1} \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor - 1} \frac{\lambda_{z,j}}{\lambda_{y,j}^2} \frac{1}{\gamma^{(i-2j)}} + \|\Lambda_x\| \sum_{i=k}^{m-1} \sum_{j=\lfloor \frac{i}{4} \rfloor}^{m} \frac{\lambda_{z,j}}{\lambda_{y,j}},\end{aligned} \tag{3.81}$$

and

$$\begin{aligned}\sum_{j=1}^{m} (\Lambda_{e_z}(p_1^T y, \ldots, p_k^T y))_{jj} - (\Lambda_{e_z}(y))_{jj} = \sum_{i=k+1}^{m} \|\Lambda_x C^T p_i\|^2 \\ \leq 4\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}}) \zeta \mu_m \sum_{i=k}^{m-1} \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor - 1} \frac{\lambda_{z,j}^2}{\lambda_{y,j}^2} \frac{1}{\gamma^{(i-2j)}} + \sum_{i=k}^{m-1} \sum_{j=\lfloor \frac{i}{4} \rfloor}^{m} \frac{\lambda_{z,j}^2}{\lambda_{y,j}}.\end{aligned} \tag{3.82}$$

By Assumptions 4 and 2, $\lambda_{z,j}/\lambda_{y,j} \leq \bar{\lambda}_j/\sigma^2$ and $\bar{\lambda}_j/(\gamma^j \lambda_{y,j}) \leq \xi$ for a constant ξ. Moreover, $\lambda_{z,i}/\lambda_{y,j} \leq 1$, in general. Thus

$$\begin{aligned}\sum_{j=1}^{m} (\Lambda_{e_x}(p_1^T y, \ldots, p_k^T y))_{jj} - (\Lambda_{e_x}(y))_{jj} = \sum_{i=k+1}^{m} \|\Lambda_x C^T p_i\|^2 \\ \leq \frac{4\|\Lambda_x\|\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}}) \zeta \mu_m \xi}{\sigma^2 (1 - \frac{1}{\gamma^2})} \sum_{i=k}^{m-1} \frac{1}{\gamma^{i/4}} + \frac{\|\Lambda_x\|}{\sigma^2} \sum_{i=k}^{m-1} (i - k + 4) \bar{\lambda}_{\lfloor \frac{i}{4} \rfloor},\end{aligned} \tag{3.83}$$

and

$$\sum_{j=1}^{m}(\Lambda_{e_z}(p_1^T y, \ldots, p_k^T y))_{jj} - (\Lambda_{e_z}(y))_{jj} = \sum_{i=k+1}^{m} \|C\Lambda_x C^T p_i\|^2$$
$$\leq \frac{4\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}})\zeta\mu_m}{(1-\frac{1}{\gamma^2})} \sum_{i=k}^{m-1}\frac{1}{\gamma^{i/2}} + \sum_{i=k}^{m-1}(i-k+4)\min\left(\frac{\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\lambda_{z,\lfloor\frac{i}{4}\rfloor}}{\sigma^2}, \bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\right), \tag{3.84}$$

The increasing μ_m converge in probability to a random variable μ by Proposition 3.5.6. Equations (3.30) and (3.31) follow immediately from (3.83) and (3.84). □

The analysis presented here predicts actual convergence behaviors, as illustrated next with the numerical examples in Section 3.6.

■ 3.6 Numerical Examples

The following numerical examples illustrate the actual performance of the algorithm in relation to the theory of the previous sections. There are four different examples. Each one illustrates a different aspect of the theory. The estimation problems in each of the examples is different. The breadth of estimation problems provides a glimpse at the range of applicability of the Krylov subspace estimation algorithm. For each of the following problems, full orthogonalization was used, except as noted.

The results in Figure 3.1 illustrate the relationship between the actual performance of the algorithm and that predicted by Theorem 3.3.1. The estimation problem consists of estimating 1024 samples of a stationary process, x, on a 1-D torus from 512 consecutive point measurements, y. The power spectral density (PSD) of x has a geometric decay, $S_{xx}(\omega) \propto (0.3)^{|\omega|}$ and is normalized so that the variance of x is one. Depicted in Figure 3.1 are the fractions of error reduction obtained for estimating x,

$$\frac{\sum_{i=1}^{l}(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}}{\sum_{i=1}^{l}(\Lambda_x - \Lambda_{e_x}(y))_{ii}}, \tag{3.85}$$

and z,

$$\frac{\sum_{i=1}^{l}(\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{ii}}{\sum_{i=1}^{l}(\Lambda_z - \Lambda_{e_z}(y))_{ii}}, \tag{3.86}$$

where $\Lambda_n = \sigma^2 I$ for $\sigma^2 = 1$ and $\sigma^2 = 10^{-8}$. Note that the numerators in (3.85) and (3.86) are the terms bounded in Theorem 3.3.1 and that the denominators are independent of the iteration index, k. The reference values $\Lambda_{e_x}(y)$ and $\Lambda_{e_z}(y)$ are computed using direct methods in MATLAB. The numerical errors in these direct methods tend to dominate after several iterations especially for $\sigma^2 = 10^{-8}$. Note that the eigenvalues of Λ_x and Λ_z satisfy $\lambda_{x,i} \geq \lambda_{z,i} \geq \lambda_{x,l-m+i}$ as a consequence of Cauchy's

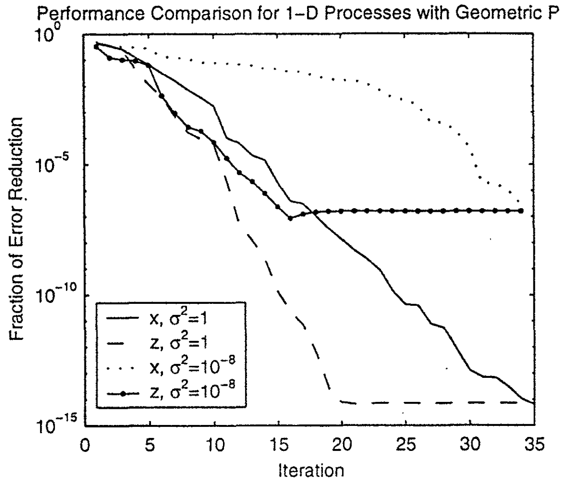


Figure 3.1. The four curves plotted here show the convergence behaviors when computing error variances for estimating two different quantities in two slightly different estimation problems. One of the quantities to be estimated is a 1-D process, x, and the other is a subsampled version of the same process, z. Both quantities are estimated from measurements consisting of z embedded in additive noise. The only difference between the two estimation problems is the variance of the noise, σ^2, which is 1 in one case and 10^{-8} in the other. The curves indicate that convergence is slower for lower σ^2 and for estimating x, as predicted by Theorem 3.3.1.

interlace theorem [40, Theorem 4.3.15] and the rows of the measurement matrix C being orthogonal. Since the PSD (collection of eigenvalues) display a two-sided geometric decay, Λ_z and, in turn, $\Lambda_y = \Lambda_z + \sigma^2 I$, may have eigenvalue multiplicities of order two. However, the plots show a geometric rate of convergence consistent with a geometrical decay of Λ_y despite the fact that the block form of the algorithm is not used. A block form is not necessary because roundoff error will introduce components of the eigenvectors of Λ_y into the Krylov subspaces that are not present in the starting vector [65, pp. 228]. Note also that, as suggested by Theorem 3.3.1, the rate of convergence is faster for the error variances at measurement locations, *i.e.* for estimates of z, than away from measurement locations, *i.e.* for estimates of all of x. The theorem also suggests that convergence is slower for smaller σ^2, which is evident in Figure 3.1. Thus, Theorem 3.3.1 accurately predicts convergence behavior.

Figure 3.2 depicts how the two stopping criteria relate to the difference between the computed approximation to the error covariance for estimating x at iteration k and the optimal error covariance, $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$. The process to be estimated is the same one previously described. The measurement locations are chosen randomly. At any given location, the chance that there is a measurement is 50% and is independent of there being a measurement at any other sample point. The measurement noise covariance matrix is a diagonal matrix whose elements vary according to the following triangle

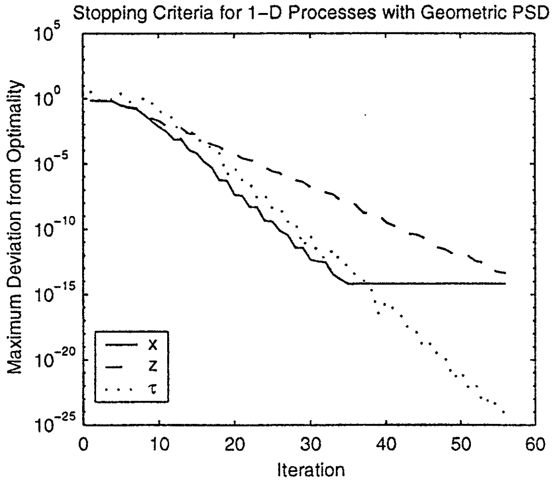


Figure 3.2. The results plotted here indicate how the computable quantities making up the two stopping criteria of Section 3.2 relate to the difference between the computed approximation to the error covariance for estimating x at iteration k and the optimal error covariance, $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$. The solid line is the maximal difference between the computed and optimal error variances for estimating x, $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. Each of the other two curves plot the quantities making up the two stopping criteria. The dashed line is the maximal error variance for estimating z, $\max_i(\Lambda_{e_z,k}(z))_{ii}$, and the dotted line is the maximum change made to the error variances at the current iteration, $\tau_{k,0}$, as defined in (3.19), for $K_{\text{win}} = 0$.

function:

$$(\Lambda_n)_{ii} = \begin{cases} 9\frac{i-1}{\lfloor m/2 \rfloor - 1} + 1 & \text{for } 1 \le i \le \lfloor m/2 \rfloor \\ 9\frac{m-i}{m - \lfloor m/2 \rfloor - 1} + 1 & \text{for } \lfloor m/2 \rfloor + 1 \le i \le m. \end{cases} \tag{3.87}$$

A whitening preconditioner, Λ_n^{-1}, is used. The figure contains plots of the maximal difference between the computed and optimal error variances for estimating x, $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. There are also plots of the two quantities making up each of the two stopping criteria. One is of the maximal error variance for estimating z, $\max_i(\Lambda_{e_z,k}(z))_{ii}$, and the other is of the maximum change made to the error variances at the current iteration, $\tau_{k,0}$ as defined in (3.19). Note that $\Lambda_{e_z,k}(z)$ is a bound on $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$, but that the rates of convergence of these two quantities are different. The $\tau_{k,0}$, on the other hand, are more erratic but decrease at a rate close to $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$. Stopping when $\tau_{k,\varepsilon_{\text{min}}}$ falls below a threshold has been the most successful criterion because the $\tau_{k,\varepsilon_{\text{min}}}$ give a good indication of the rate of decrease of $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. However, stopping when $\max_i(\Lambda_{e_z,k}(z))_{ii}$ falls below a threshold is a preferable criterion when the noise intensity is small primarily because $\max_i(\Lambda_{e_z,k}(z))_{ii}$ provides a tight bound on $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$.

A comparison among various techniques to accelerate convergence is provided in Figure 3.3. The estimation problem consists of estimating a stationary random field, x, on a 32×32 toroidal grid from point measurements, y, of equal quality taken over one

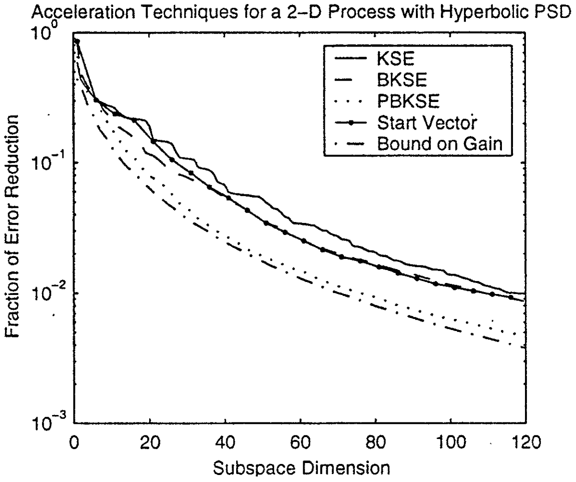


Figure 3.3. The results plotted here indicate that various acceleration techniques can be used to achieve nearly optimal performance. The curves depict the fraction of error reduction for estimating x for different methods of choosing linear functionals of the data. The figure shows the results for the standard Krylov subspace estimation algorithm (KSE), a block form with a block size of 2 (BKSE), and a preconditioned block form (PBKSE) also with a block size of 2. For comparison, the figure shows two additional curves. One (Start Vector) is of the results for Algorithm 3.1.1 modified to start with a linear combination of the first 60 eigenvectors of Λ_y. The other (Bound on Gain) is of the fraction of error reduction attained by using the optimal linear functionals of the data.

32×16 rectangle. The PSD of x is proportional to $1/(|\omega|+1)^3$ and is normalized so that the variance of x is one. The measurement noise covariance matrix, $\Lambda_n = 4I$. The plots are of the fraction of error reduction attained for estimating x, as defined by (3.85), versus the Krylov subspace dimensions. Both a right-preconditioned and block form are considered. The preconditioner has the form $C\Lambda_p C^T$, as described in Section 3.4.1. A simple block algorithm (BKSE) with a block size of 2 does not do much better than the standard algorithm (KSE). However, a preconditioned block form (PBKSE) requires considerably fewer iterations to achieve a given level of accuracy than the standard algorithm. The error reduction attained by using the optimal linear functionals of the data (referred to as "Bound on Gain" in Figure 3.3) is also plotted in Figure 3.3. The performance of PBKSE is close to the optimal performance. Figure 3.3 also shows the results of an experiment to determine whether one can gain much by picking a good starting vector. A starting vector with components in each of the first 60 eigenvectors of Λ_y was used to start a run. The results are plotted in Figure 3.3 and are comparable to those of BKSE, indicating that one does not gain much by picking a good starting vector. That the choice of starting vector should have little impact on the results is a consequence of Proposition 3.5.6.

Lastly, Figure 3.5 shows how the number of iterations grows with the region size for the problem of estimating deviations from mean sea surface temperature, x, from the satellite data, y, in Figure 3.4 [29]. The temperature deviations are estimated on a

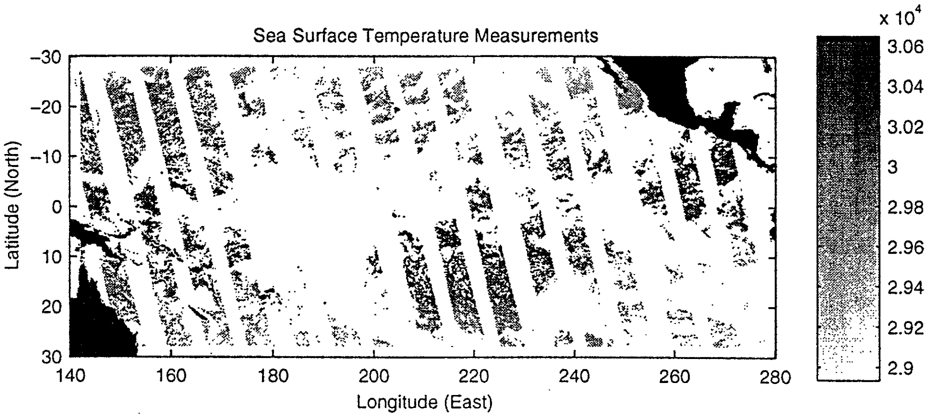


Figure 3.4. These data are satellite measurements of sea surface temperature. Measurements are taken only along satellite tracks with no obscuring cloud cover.

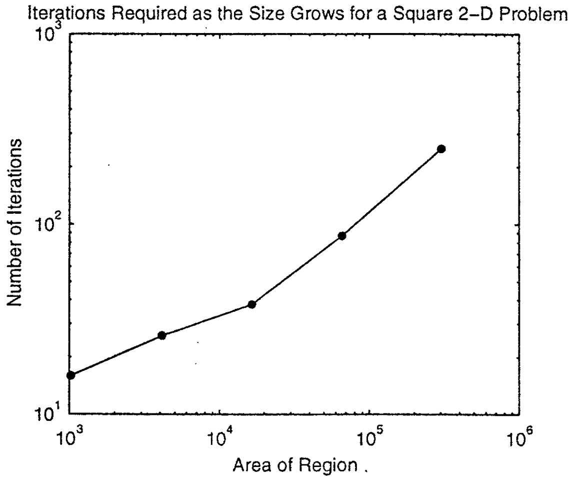


Figure 3.5. The number of iterations required for a practical 2-D problem of interest is not very large and grows no more than linearly with the area of the region of interest.

rectangular grid and are assumed to be stationary with a Gaussian-shaped covariance function. The width of the Gaussian is 60 pixels, and the height is 9×10^4. The measurements are very scattered because they only exist along the satellite tracks where there is no obscuring cloud cover (see Figure 3.4). The measurement noise covariance, $\Lambda_n = 400I$. Figure 3.5 shows how the number of iterations needed to satisfy $\tau_{k,10^{-2}} < 10^{-2}$ for $K_{\text{win}} = 8$ grows as a region of interest grows. Note that the measurement density in these regions varies from approximately $10-20\%$. The growth in the number of iterations is less than linear as the area of the region grows. One expects this behavior because one should need an increasing number of linear functionals as the region grows, but the growth should be no more than linear in the area, provided that

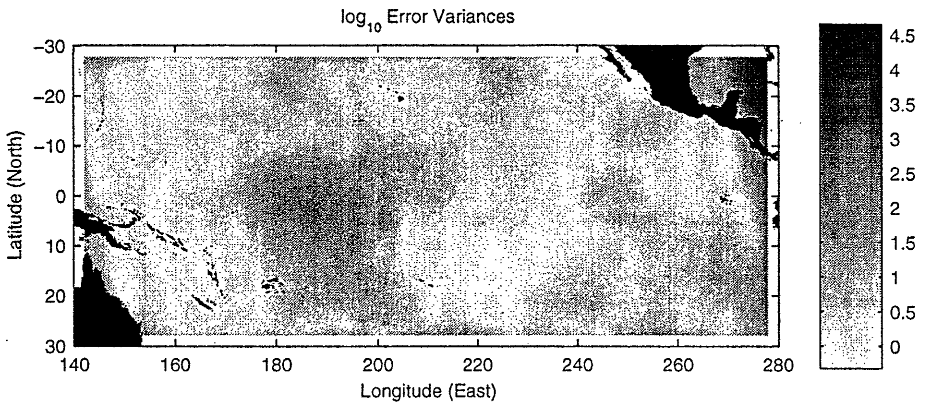


Figure 3.6. The Krylov subspace estimation algorithm generated these error variances on a 1/6-degree grid.

the process is stationary (as it is in this case). Figure 3.6 shows the error variances for estimating sea surface temperature given all 42,298 measurements in Figure 3.4. A selective orthogonalization scheme was used to generate this result (see Appendix B). Although the number of iterations is growing with problem size, the number of iterations needed for this moderately large 320,400-dimensional estimation problem is 249. That only a relatively small number of iterations were used indicates that the algorithm has found a very low rank, but very good, estimator. Hence, the algorithm described here can be used to solve high-dimensional, practical problems with relatively few iterations.

■ 3.7 Summary

In this chapter, a statistical interpretation of CG has been used to derive a Krylov subspace estimation algorithm. The algorithm computes a low-rank approximation to the linear least-squares error reduction term which can be used to recursively compute linear least-squares estimates *and* error variances. An analysis of the convergence properties explains behaviors of the algorithm. In particular, convergence is more rapid at measurement locations than away from them when there are scattered point measurements. Furthermore, the analysis indicates that a randomly generated vector is a good starting vector. The theory also suggests preconditioning methods for accelerating convergence. Preconditioning has been found to increase the rate of convergence in those cases where convergence is not already rapid.

The low-rank approximation to the error reduction term is a very useful statistical object. The computation of estimates and error variances is just one application. Another is the simulation of Gaussian random processes. Simulation typically requires the computation of the square root of the covariance matrix of the process, a potentially costly procedure. However, the Krylov subspace estimation algorithm can be adapted to generate a low-rank approximation to the square root of the covariance matrix. Yet another application is the fusion of existing estimates with those generated by additional data. The resulting fusion algorithm can also be used as the engine of a Kalman filtering routine, thereby allowing the computation of estimates of quantities evolving in time. These are the subjects of Chapters 4-6.

Chapter 4

A Krylov Subspace Method for Realization

The Krylov subspace algorithm for estimation can also be used for realization. In the realization problem, as considered here, one is given the covariance Λ_x of an l-dimensional zero-mean random vector x, which is often referred to, in this chapter, as the *process* of interest. Then, one is interested in computing two quantities. The first is the synthesis of a zero-mean random vector x' whose covariance $\Lambda_{x'}$ matches the given covariance Λ_x, either exactly or approximately. The second quantity of interest is a low-rank approximation to the covariance. That is, one is interested in computing a reasonably small number of vectors, $a_1, \ldots, a_r$ such that

$$\sum_{i=1}^{r} a_i a_i^T \approx \Lambda_x. \tag{4.1}$$

For performing both sets of computations, one would like an algorithm that is as computationally efficient as possible. In this chapter, we demonstrate how one can use a variant of the Krylov subspace algorithm for estimation to solve both aspects of the realization problem efficiently. This variant is a new approach to the realization problem.

■ 4.1 Existing Approaches to Realization

There are many existing methods for solving the realization problem. Three such are summarized here to provide a context in which to understand the proposed algorithm. The first makes use of eigendecompositions; the second, FFTs; and the third, Lanczos iterations. In each case, one is trying to realize a zero-mean random variable x with covariance Λ_x.

■ 4.1.1 Karhunen-Loève Bases

One approach to approximate realization involves solving the following optimization problem. Consider finding a linear least-squares estimate of x given x such that the estimator is of rank less than or equal to r. That is, consider finding $\hat{x}$ such that

$$\hat{x} = \underset{\{\hat{x}=Ax \mid \operatorname{rank}(A) \leq r\}}{\operatorname{argmin}} \mathrm{E}\left[\|x - \hat{x}\|^2\right]. \tag{4.2}$$

Now,

$$\begin{aligned} \mathrm{E}\left[\|x - Ax\|^2\right] &= \operatorname{Tr}(\operatorname{Cov}((I - A)x)) && (4.3)\\ &= \operatorname{Tr}((I - A)\Lambda_x (I - A)^T) && (4.4)\\ &= \operatorname{Tr}\left((I - A)U\Sigma U^T (I - A)^T\right) && (4.5) \end{aligned}$$

where $\Lambda_x = U\Sigma U^T$ is the eigendecomposition of Λ_x (Karhunen-Loève decomposition of x). Thus, $\hat{x} = U_r U_r^T x$ where the columns of U_r are the orthonormal eigenvectors corresponding to the r largest eigenvalues.

The resulting covariance of $\hat{x}$ is an optimal rank r approximation to Λ_x. This covariance is $\Lambda_{\hat{x}} = U_r \Sigma_r U_r^T$, where Σ_r is the diagonal matrix with the dominant r eigenvalues of Λ_x on the diagonal. Moreover, the product of $U_r \sqrt{\Sigma_r}$ with a random vector that has identity covariance, is a synthesis of a random vector with covariance $\Lambda_{\hat{x}}$. This is one approach to realization.

4.1.2 FFT Methods

Although one can, in principle, perform eigendecompositions to create realizations of arbitrary accuracy, even exact ones, this is not always the best approach because of the computational effort required to perform eigendecompositions. If there is special structure in the covariance matrix, one may be able to find methods for computing a realization that are more computationally efficient than methods based on computing an eigendecomposition. In particular, one may be able to use FFTs if the process to be realized consists of samples of a stationary process so that the covariance matrix Λ_x is Toeplitz.

Using FFTs for realization is similar to using FFTs for Λ_x-vector multiplication, *i.e.* convolution. First, one embeds Λ_x in a circulant embedding matrix C

$$C = \begin{pmatrix} \Lambda_x & * \\ * & * \end{pmatrix} \tag{4.6}$$

that need be no larger than $2(l-1) \times 2(l-1)$. Since C is circulant, it is diagonalized by the DFT matrix F. That is, one can write

$$C = F^* G F \tag{4.7}$$

where G is a diagonal matrix. Interpreting the product Cz as the circular convolution of z with the "impulse response" corresponding to the first row of C, one can think of the diagonal elements of G as defining the corresponding frequency response or transfer function. From (4.6) and (4.7), one then has that

$$\Lambda_x = SF^*GFS^T \tag{4.8}$$

where

$$S = \begin{pmatrix} I & 0 \end{pmatrix} \tag{4.9}$$

selects the first l components of a vector. The factorization in (4.8) amounts to zero-padding, performing a $2(l-1)$ FFT, multiplying by the transfer function G, performing an inverse FFT, and finally selecting the first l components of the result.

It would appear that one could synthesize a process by simply multiplying $SF^*\sqrt{G}F$ by a random vector w that has identity covariance because the resulting product has covariance

$$\mathrm{Cov}(SF^*\sqrt{G}Fw) = SF^*\sqrt{G}FF^*\sqrt{G}FS = \Lambda_x, \tag{4.10}$$

as desired. However, the circulant matrix C, and hence G, may not be positive semi-definite even if Λ_x is. So, one may not be able to form square roots of G. For those processes which admit positive semi-definite embeddings, however, FFT-based methods are efficient. The following theorem provides sufficient conditions for a Toeplitz covariance matrix to have a positive semi-definite circulant embedding [23, Theorem 2]. The statement is written in terms of the covariance function $K[i] = (\Lambda_x)_{1i}$.

Theorem 4.1.1. *If the values of the covariance function of an l-point random vector, $K[1], K[2], \ldots, K[l]$, form a sequence that is convex*[1]*, decreasing, and nonnegative, then the associated $2(l-1) \times 2(l-1)$ circulant matrix is positive semi-definite.*

As an example, consider the situation where the vector to be synthesized, x, consists of the increments between regularly spaced samples of fractional Brownian motion (fBm) [54] for Hurst parameter $H \in (1/2, 1)$. The increments process is stationary with covariance function

$$K[m] = \frac{\sigma^2\delta^{2H}}{2}\left(|m+1|^{2H} + |m-1|^{2H} - 2|m|^{2H}\right) \tag{4.11}$$

where δ is the sampling interval and σ^2 is a constant appearing in the definition of fBm [1]. One can verify that the covariance of the increments, $K[m]$, defined in (4.11), is convex, decreasing, and nonnegative, as follows. To verify that $K[m]$ is nonnegative, note that

$$K[m] = \frac{\sigma^2\delta^{2H}}{2}\Big(\left(|m+1|^{2H} - |m|^{2H}\right) - \left(|m|^{2H} - |m-1|^{2H}\right)\Big). \tag{4.12}$$

[1] A convex sequence is one such that for any two integers $m < n$, $\lambda K[m] + (1-\lambda)K[n] \geq K[i]$ for all $\lambda \in [0,1]$ and $m \leq i \leq n$.

Since $2H > 1$, $(|m+1|^{2H} - |m|^{2H}) \geq (|m|^{2H} - |m-1|^{2H})$; so, $K[m] \geq 0$. To verify that $K[m]$ is convex, view m as a continuous parameter greater than zero, and note that

$$\begin{aligned}\frac{dK}{dm} &= \frac{\sigma^2\delta^{2H}}{2}(2H|m+1|^{2H-1} \pm 2H|m-1|^{2H-1} - 2(2H)|m|^{2H-1}) \\ \frac{d^2K}{dm^2} &= \frac{\sigma^2\delta^{2H}}{2}(2H(2H-1)|m+1|^{2H-2} + 2H(2H-1)|m-1|^{2H-2} \\ &\quad - 2(2H)(2H-1)|m|^{2H-2}) \\ &= \frac{\sigma^2\delta^{2H}}{2}(2H(2H-1)|m+1|^{2H-2} + 2H(2H-1)|m-1|^{2H-2} \\ &\quad - 2H(2H-1)|2^{1/(2H-2)}m|^{2H-2}).\end{aligned} \tag{4.13}$$

Since $-1 < 2H - 2 < 0$, $2^{1/(2H-2)} < 1$. This implies that

$$|m+1|^{2H-2} > |2^{1/(2H-2)}m|^{2H-2}; \tag{4.14}$$

so, $d^2K/dm^2 > 0$, and $K[m]$ is convex. That K is decreasing follows from the fact that K is convex, nonnegative, and asymptotically approaching zero. Thus, FFTs can be used to synthesize important processes such as fBm increments.

Although not commonly done, one can also consider using FFT methods to form low-rank approximations to covariance matrices. One method for doing this is to pick out terms from the expansion in (4.8). That is, one forms a rank $\min(r, l)$ approximation

$$\Lambda_x \approx \sum_{j=1}^{r} (f_{i_j} S^T)^T g_{i_j} (f_{i_j} S^T) \tag{4.15}$$

where $g_{i_1} \geq g_{i_2} \geq \cdots \geq g_{i_{2(l-1)}}$ are the ordered elements of the diagonal of G and $(f_{i_j} S^T)$ are the corresponding Fourier vectors truncated by the selection matrix S. Hence, one can use FFTs to solve both aspects of the realization problem.

4.1.3 Lanczos Algorithms

Yet another approach to realization is to use Lanczos methods for approximately computing functions of matrices [12, 25, 76]. These methods approximate a function $f(\cdot)$ of a matrix A using Krylov subspace methods, specifically the Lanczos algorithm. As described in Section 3.1, the Lanczos algorithm is iterative. At step k, the method computes a set of orthonormal vectors $q_1, \ldots, q_k$ that form a basis for the Krylov subspace of dimension k generated by q_1 and A. These vectors form a matrix

$$Q_k = \begin{pmatrix} q_1 & \cdots & q_k \end{pmatrix} \tag{4.16}$$

that tri-diagonalize A, *i.e.* $Q_k^T A Q_k = T_k$, where T_k is tri-diagonal. One can perform an efficient eigendecomposition of each of the T_k to yield the factorization $S_k D_k S_k^T = T_k$. Then, the approximation of $f(A)$ at the k-th step of Lanczos is

$$f(A) \approx Q_k S_k f(D_k) S_k^T Q_k^T. \tag{4.17}$$

For the realization problem, one is interested in using this method for the square-root function, $f(\cdot) = \sqrt{\cdot}$. Specifically, the approximate synthesis of a random vector with covariance Λ_x at the k-th step is

$$Q_k S_k \sqrt{D_k} S_k^T Q_k^T w \tag{4.18}$$

for a random vector w that has identity covariance, and the approximation to Λ_x is

$$Q_k S_k D_k S_k^T Q_k^T. \tag{4.19}$$

Although this approach to computing matrix approximations and square roots has not been widely used for realization[2], it has been used in other contexts [25]. The algorithm proposed in this chapter is a variation on the standard Lanczos approach that replaces eigendecompositions of T_k with Cholesky factorizations. This makes the algorithm more recursive in nature and more akin to conjugate gradient than Lanczos.

4.2 Krylov Subspace Realization

This section presents a new approach to realization that makes use of the Krylov subspace estimation algorithm. Approximate realization is related to estimation using linear functionals in the following sense. Recall that the problem of interest is realizing a zero-mean random vector with the same second-order statistics as a zero-mean random vector x which has covariance Λ_x. Let $p_1^T x, \ldots, p_k^T x$ be linear functionals of x that whiten x, *i.e.* $\mathrm{Cov}(p_i^T x, p_j^T x) = \delta_{ij}$. The best linear estimate of x given $p_1^T x, \ldots, p_k^T x$ is

$$\hat{x}_k(x) = \sum_{i=1}^{k} (b_i)(p_i^T x), \tag{4.20}$$

where $b_i = \Lambda_x p_i$ are the filtered back-projected linear functionals (see Section 3.1). Since the $p_i^T x$ are white, one can replace them with any other sequence of white random variables $w_1, w_2, \ldots, w_k$ to obtain another random vector with the same second-order statistics,

$$x'_k = \sum_{i=1}^{k} b_i w_i. \tag{4.21}$$

The covariance of x'_k is

$$\Lambda_{x'_k} = \sum_{i=1}^{k} b_i b_i^T \tag{4.22}$$

[2] In fact, there appear to be no instances in the literature of the Lanczos method for function approximation being used for realization.

The vector x'_k is an approximate realization of x, and $\Lambda_{x'_k}$ is an approximation to Λ_x.

We compute approximate realizations by picking linear functionals from Krylov subspaces generated by the covariance of interest, Λ_x. This has the interpretation of using the Krylov subspace estimation algorithm, described in Chapter 3, to estimate x from x. The advantages of using Krylov subspaces are the same as for Krylov subspace estimation. In particular, the realization algorithm will find a good low-rank approximation to Λ_x if one exists since the Krylov subspaces generated by Λ_x are capturing more and more of the dominant modes of Λ_x as the dimension of the Krylov subspaces increases.

As for the Krylov subspace estimation algorithm, one of the advantages of the realization algorithm is that one can formulate a natural stopping criterion. The main estimation algorithm stopping criterion, discussed in Section 3.2.1, makes use of the quantity, $\tau_{k,\varepsilon_{\min}}$. This quantity measures relative changes to the error reduction in the last few iterations. When $\tau_{k,\varepsilon_{\min}}$ falls below a threshold, the estimation algorithm stops. However, $\tau_{k,\varepsilon_{\min}}$ is not necessarily the quantity to check for determining when to stop the realization algorithm. For realization, $\tau_{k,\varepsilon_{\min}}$ would examine changes made to $\Lambda_{x'_k}$ in previous iterations relative to the difference in covariances $\Lambda_{r,k} = \Lambda_x - \Lambda_{x'_k}$. Yet, $\Lambda_{r,k}$ tends to zero as k increases. Thus, $\tau_{k,\varepsilon_{\min}}$ may become large. Instead of $\tau_{k,\varepsilon_{\min}}$, one can use

$$\frac{1}{l}\sum_{i=1}^{l}(\Lambda_{r,k})_{ii} \tag{4.23}$$

as a basis for a stopping criterion. This measures the total difference in the variances between x and x'_k. It is a useful measure of the quality of approximation of x'_k and is easy to update at each iteration. Thus, one can use this Krylov subspace method to compute an approximate realization of x and a low-rank approximation to Λ_x and easily verify the quality of the approximation after each iteration. Summarizing, one has the following:

Algorithm 4.2.1. *A Krylov Subspace Method for Realization.*

1. *Initialize* $x'_0 = 0$, $(\Lambda_{r,k})_{ii} = (\Lambda_x)_{ii}$ *for* $i = 1, \ldots, l$.

2. *Generate a zero mean Gaussian random vector s with identity covariance to initialize the Krylov subspace.*

3. *Perform the following operations for each step k until* $1/l\sum_{i=1}^{l}(\Lambda_{r,k})_{ii}$ *falls below a threshold* χ*:*

   (a) *Compute the conjugate search direction p_k and filtered backprojection $b_k = \Lambda_x p_k$ using a reorthogonalized Lanczos iteration, (3.10)-(3.13), as for the Krylov subspace estimation algorithm of Section 3.1 with $C = I$ and $\Lambda_n = 0$.*

(b) *Generate an independent random number w_k and update*

$$x'_k = x'_{k-1} + b_k w_k \tag{4.24}$$

$$(\Lambda_{r,k})_{ii} = (\Lambda_{r,k-1})_{ii} - ((b_k)_i)^2 \quad \textit{for } i = 1, \ldots, l. \tag{4.25}$$

Since the realization algorithm is an extension of the estimation algorithm, one can consider applying the techniques in Section 3.4 to accelerate convergence of the estimation algorithm. In particular, one can consider using preconditioning to separate the eigenvalues, which will improve the theoretical error bounds in Theorem 3.3.1. The preconditioning strategies in Section 3.4.1 can also be used for realization. In particular, one can operate on the spectrum of the covariance matrix Λ_x if one has approximations to its eigenvectors. An example of how one can do this is presented later in Section 4.4.3

■ 4.3 Computational Complexity

This section examines the computational complexity of the four approaches to realization discussed in this chapter. The focus is on the Krylov subspace realization algorithm and the standard Lanczos iteration; however, the complexity of approaches using Karhunen-Loève bases and FFT methods are also examined.

■ 4.3.1 Krylov Subspace Realization vs. Standard Lanczos

Note that the Krylov subspace realization algorithm and the standard Lanczos iteration yield results that would be almost the same in exact arithmetic. Specifically, the covariance matrix approximation generated at step k by the Krylov subspace realization algorithm is

$$\Lambda_x Q_k T_k^{-1} Q_k^T \Lambda_x \tag{4.26}$$

where Q_k and T_k are as in Section 4.1.3. The standard Lanczos algorithm for computing matrix square roots, on the other hand, generates the following approximation at step k:

$$Q_k T_k Q_k^T. \tag{4.27}$$

These approximations are very similar.

To see this, consider running the Lanczos tri-diagonalization to completion. For any k, then, one can write

$$\Lambda_x = \begin{pmatrix} Q_k & Q_k^{\perp} \end{pmatrix} \begin{pmatrix} T_k & E^T \\ E & T_k^{\perp} \end{pmatrix} \begin{pmatrix} Q_k & Q_k^{\perp} \end{pmatrix}^T \tag{4.28}$$

where the columns of $Q_k^\perp$ are the Lanczos vectors generated after step k, $T_k^\perp$ is tridiagonal, and E is of the form

$$E = \begin{pmatrix} 0 & \cdots & 0 & * \\ 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{4.29}$$

Thus,

$$\Lambda_x Q_k T_k^{-1} Q_k^T \Lambda_x = Q_k T_k Q_k^T + Q_k^\perp E Q_k^T + Q_k E^T \left(Q_k^\perp\right)^T + Q_k^\perp E T_k^{-1} E^T \left(Q_k^\perp\right)^T. \tag{4.30}$$

So, the difference between the Krylov subspace realization algorithm approximation in (4.26) and the Lanczos iteration approximation in (4.27) is at most rank 3 since E is rank one.

The primary advantage of the Krylov subspace realization algorithm is that it allows for one to recursively update the synthesis and approximation error at each step instead of having to recompute these quantities. The recursive structure also results in a modest computational gain. The amount of reduction can be quantified, as follows, by counting the number of multiplications required to perform most of the computation. Suppose each Λ_x-vector multiply requires $\mu_x l$ multiplications and that a preconditioner is used that requires $\mu_p l$ multiplications for each preconditioning matrix-vector multiply. Suppose further that the algorithm is run for k iterations. Then, the Krylov subspace realization algorithm performs $(\mu_x + \mu_p)lk$ scalar multiplies to compute the matrix-vector products and lk^2 scalar multiplies to perform reorthogonalization. The standard Lanczos method for computing a square root performs both of these computations and must also recompute the variance differences at every iteration. This requires an additional $lk^2/2$ multiplications. The totals are provided in Table 4.1, and the parameters are summarized in Table 4.2. Thus, the Krylov subspace realization algorithm achieves a modest computational gain of $lk^2/2$ over the standard Lanczos algorithm.

	Scalar Multiplies
Krylov Subspace Realization	$(\mu_x + \mu_p)lk + lk^2$
Standard Lanczos Matrix Square Root	$(\mu_x + \mu_p)lk + (3/2)lk^2$

Table 4.1. Scalar Multiplies Required for Realization

■ 4.3.2 Karhunen Loève Bases

Any implementation of a realization algorithm using Karhunen-Loève bases, as discussed in Section 4.1.1, requires a routine for computing partial eigendecompositions.

Parameter	Description
μ_x	per sample work for Λ_x-vector multiplies
μ_p	per sample work for preconditioner multiplies
l	random vector dimension
k	number of iterations

Table 4.2. Parameters in the Realization Algorithm Computational Complexity Analysis

One of the most popular iterative routines is the Lanczos algorithm [33, Chapter 9]. This method, at step k, will have computed a basis for a Krylov subspace of dimension k generated by the matrix of interest Λ_x, and some starting vector s. Then, one computes approximate eigenvectors by selecting appropriate vectors from this Krylov subspace.

Note that such an implementation of a realization algorithm using Karhunen-Loève bases will always generate an approximation with worse mean-squared error for the same computational effort than the Lanczos algorithms for function approximation presented in Section 4.1.3. This follows from the fact that at iteration k, the function approximation approach will project the covariance Λ_x onto the entire Krylov subspace $\mathcal{K}(\Lambda_x, s, k)$ whereas the Karhunen-Loève approach will project Λ_x onto the subspace of $\mathcal{K}(\Lambda_x, s, k)$ spanned by the approximate eigenvectors. Thus, the Karhunen-Loève approach to realization is not particularly practical and is considered in this chapter only because it is optimal.

■ 4.3.3 FFT Methods

Unlike approaches for realization using Karhunen Loève bases, FFT methods may be computationally competitive with the Krylov subspace realization algorithm. The computation of an FFT method is dominated by the FFT. This can be implemented by an algorithm that is $O(l \log l)$ where l is the dimension of the random vector to be realized. Whether this is competitive or not with the Krylov subspace realization algorithm depends on the problem.

Specifically, consider two different asymptotic scenarios. In each case, suppose one is realizing samples of a continuous random process over a compact subset of $\mathbb{R}^d$ for some dimension d.

In the first case, let the size of the subset grow but keep the sampling density fixed. Then, as l increases, consider the behavior of the Krylov subspace realization algorithm. The number of linear functionals of the process, k, needed to meet a desired level of accuracy should grow linearly with l. This is a consequence of the need to use more linear functionals of the process to capture its behavior over the larger region. Since the Krylov subspace realization algorithm has complexity $O(lk^2)$, and the FFT method, $O(l \log l)$, the FFT method will become more competitive as the region size grows.

Now consider the case where the region size is fixed, but the sampling density increases. Then, as l increases, the number k of linear functionals will remain constant. Instead, one needs different linear functionals that capture the behavior on the refined grid. The Krylov subspace realization algorithm will compute the appropriate linear functionals for the grid size. In this case, the Krylov subspace realization algorithm will be less computationally intensive than the FFT method for large l.

The problem sizes at which one method becomes less intensive than the other depend on the specific problem and implementations.

■ 4.4 Numerical Examples

The performance of the Krylov subspace realization algorithm is illustrated in this section with three examples. For each example, three sets of results are provided. First, a high quality sample path generated by the Krylov subspace realization algorithm is plotted along with a sample path whose statistics exactly match the given covariance. Second, the difference between the true variances and those of the Krylov subspace realization are plotted. Since the difference between the covariances is positive semi-definite, the variances provide a good measure of the quality of the approximation. Lastly, the fraction of total mean-squared error reduction of the Krylov subspace realization is plotted versus approximation rank. For comparison, the error reduction obtained by the optimal Karhunen-Loève approach outlined in Section 4.1.1 is also plotted. All results were generated using MATLAB on a Sun workstation with a floating point precision of approximately 2×10^{-16}.

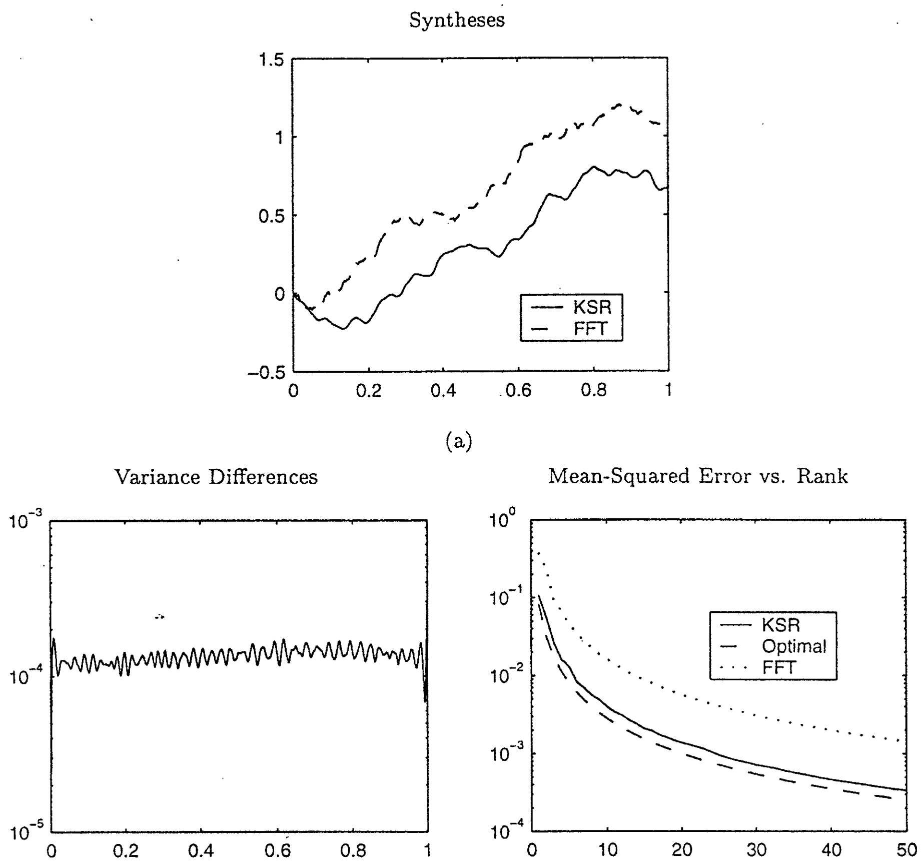
■ 4.4.1 Fractional Brownian Motion

The first example consists of realizing 1024 samples of a fractional Brownian motion (fBm) with a Hurst parameter $H = 3/4$. The covariance of fBm is given by

$$K_{xx}(s,t) = \frac{1}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right). \tag{4.31}$$

Recall from Section 4.1.2 that fBm has stationary increments, and for $H = 3/4$, the covariance matrix of the increments process can be embedded in a positive semi-definite circulant matrix which is not more than twice as large as the fBm increments covariance. Thus, one can synthesize this fBm exactly with 2048-point FFTs and also generate good finite-rank approximations to the covariance matrix. As a result, there is not necessarily a need to use a Krylov subspace method to realize this fBm. However, the problem of realizing fBm provides a good example of how the Krylov subspace realization algorithm could be used to realize a non-stationary process. The example also illustrates the algorithm's power in obtaining near-optimal low-rank covariance approximations.

Figure 4.1 presents the results. Part (a) of the figure shows sample paths generated using the exact FFT method and 50 iterations of the Krylov subspace method. Note that the one generated with the Krylov subspace method looks similar to the one generated with FFTs. One would expect this since the differences between the



(a)

(b)

(c)

Figure 4.1. These results are for realizing 1024 samples of a fBm with Hurst parameter $H = 3/4$. Part (a) shows sample paths generated with FFTs (an exact method) and with 50 iterations of the Krylov subspace realization (KSR) algorithm. The difference between the variances are plotted in part (b). Note that the true fBm variances are given by $t^{3/2}$ as t ranges from zero to one. The fraction of error reduction obtained by each method as a function of approximation rank is plotted in part (c). The optimal (KL) results are plotted for comparison.

exact and approximate variances are small, as indicated in part (b). There are two other interesting features of the variance differences. One is that, they are uniformly small. The other is that they consist mostly of high frequency oscillations indicating that higher frequency modes (and, hence, the least important ones) are the ones left out of the approximation. Again, this is expected since, as indicated in part (c), the Krylov subspace approach is picking linear functionals that are almost as good as picking the optimal ones, namely the eigenvectors. Also, note that the Krylov subspace approach does much better than the FFT-based approach outlined in Section 4.1.2 for approximating the covariance matrix at any specified rank.

4.4.2 Windowed Cosine Covariance

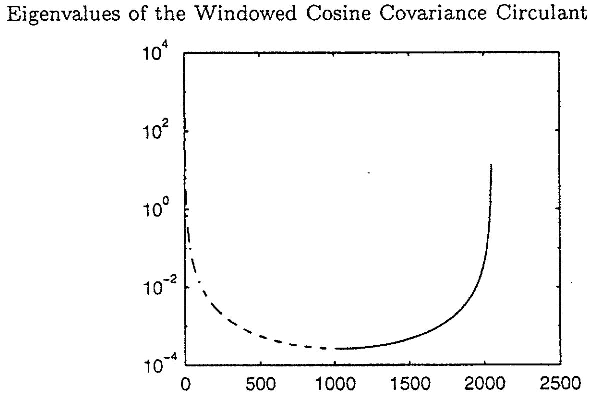


Figure 4.2. The plot shows the eigenvalues of the 2048-point circulant embedding for the covariance matrix of 1024 samples of a process with a windowed cosine covariance given by (4.32). There are both positive and negative eigenvalues. The curve plots the magnitudes. Those plotted with a solid line are negative; those, with a dashed line, positive.

For the second example, the random vector to be realized consists of 1024 samples in the unit interval of a stationary process whose covariance function, $K_{xx}(\tau)$, is a Gaussian-windowed cosine:

$$K_{xx}(\tau) = e^{-\frac{\tau^2}{2}} \cos(2\pi\tau). \tag{4.32}$$

This process is interesting because the 2048-point circulant embedding matrix has a substantial number of negative eigenvalues, as indicated in Figure 4.2. Thus, one can not use this embedding and FFTs to generate realizations.

Results using the Krylov subspace realization algorithm are plotted in Figure 4.3. Sample paths are plotted in part (a). The exact sample path is generated by forming a square root of the covariance matrix. The approximate synthesis is generated using only 14 iterations of the Krylov subspace realization algorithm. At this point, both syntheses have similar structure. This is expected because the differences between the approximate and exact realization variances are small, as indicated in part (b) of the figure. Not many iterations are needed because the eigenvalues of the covariance matrix are decreasing rapidly, and the Krylov subspace approach is near optimal, as indicated in part (c) of the figure.

4.4.3 Two-Dimensional Spherical Covariance

Lastly, consider realizing a two-dimensional isotropic random field with radial covariance

$$K_{xx}(\tau) = \begin{cases} 1 - \frac{3}{2}|\tau| + \frac{1}{2}|\tau|^3 & 0 \le |\tau| \le 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.33}$$

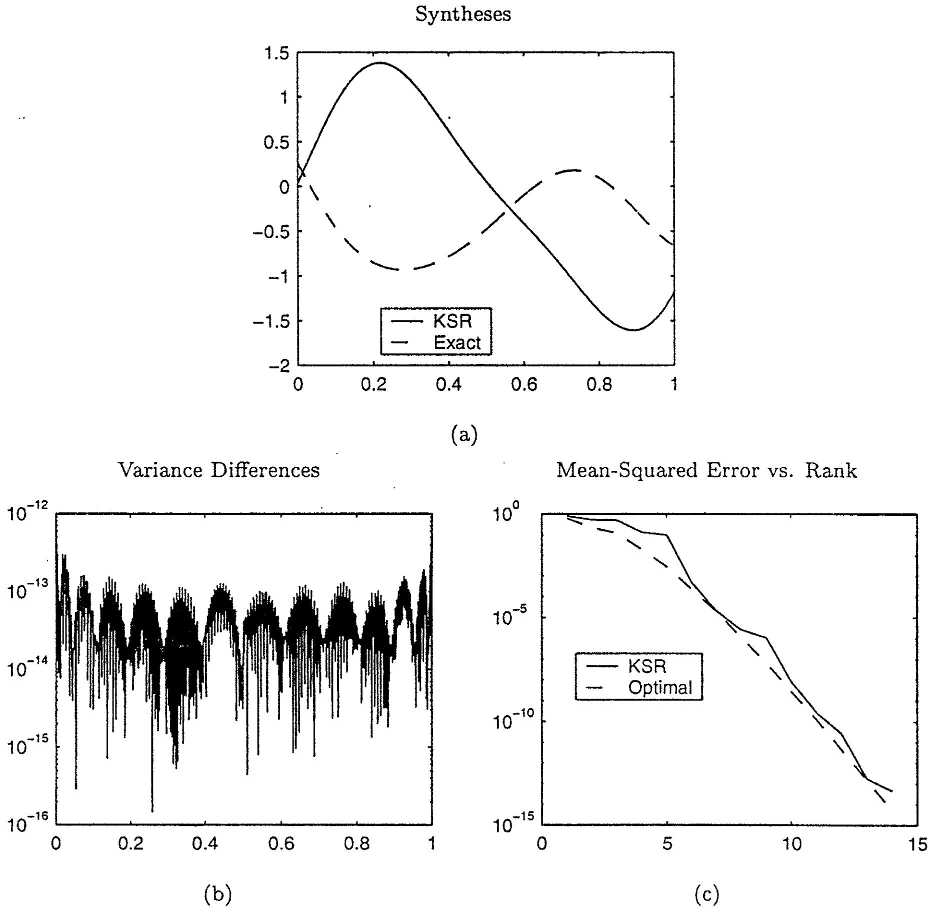


Figure 4.3. These results are for realizing 1024 samples of a process with a windowed cosine covariance given by (4.32). Part (a) shows sample paths generated with a matrix square root computation (an exact method) and with 14 iterations of the Krylov subspace realization (KSR) algorithm. The difference between the variances are plotted in part (b). The true variance of the stationary process is 1. The fraction of error reduction obtained by each method as a function of approximation rank is plotted in part (c). The optimal (KL) results are plotted for comparison.

This covariance function is known as the spherical covariance function in the geostatistical community [23, 44]. Partly due to its potential practical application, and partly, its rich structure, the spherical covariance has been used by several to characterize realization algorithms [23, 30, 77]. One can consider using two-dimensional FFTs to realize samples of a field with spherical covariance. However, if the samples are taken from a square grid that does not include the unit square, the minimal circulant embedding is not positive semi-definite. In order to demonstrate the performance of the Krylov subspace realization algorithm on a two-dimensional problem for which FFT-based methods do not apply, this section considers realizations on a 33×33 grid

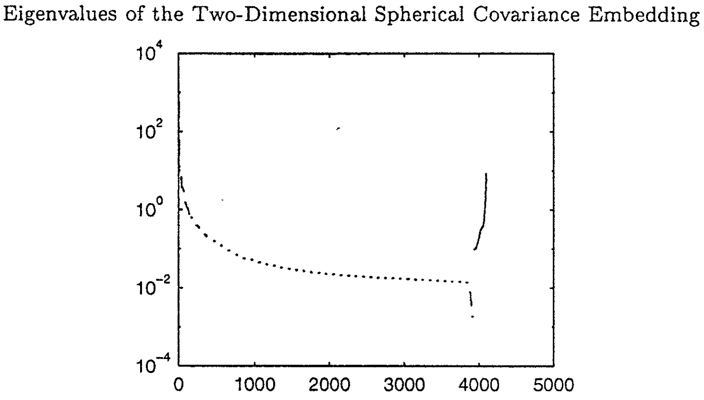


Figure 4.4. The plot shows the eigenvalues of the circulant embedding for the covariance matrix of samples on a 33 × 33 grid of a process with a two-dimensional spherical covariance given by (4.33). There are both positive and negative eigenvalues. The curve plots the magnitudes. Those plotted with a solid line are negative; those, with a dashed line, positive.

covering $[0, 32/45] \times [0, 32/45]$. The 64 × 64 two-dimensional circulant embedding of the covariance matrix of these samples has several negative eigenvalues as illustrated in Figure 4.4.

A preconditioned Krylov subspace realization algorithm is applied to this realization problem. Recall from Sections 4.2 and 3.4 that one strategy for preconditioning is to use an approximation of the eigenvectors to form a matrix that separates out the eigenvalues of Λ_x. Since Λ_x is stationary, the elements of the Fourier basis are approximate eigenvectors. Thus, one can consider using a preconditioner of the form

$$SF^* G_p F S^* \tag{4.34}$$

where S^* zero pads, F is the two-dimensional DFT operator for the zero-padded image, G_p is a diagonal matrix, F^* is the inverse two-dimensional DFT, and S selects the portion of the result of interest. Once again, the diagonal elements of G_p can be interpreted as a frequency response, which we use here to shape the preconditioning. Specifically, in order to separate out eigenvalues, the diagonal elements of G_p are chosen to be

$$G_p(f) = 50|f|^2(0.4)^{|f|} + 1 \tag{4.35}$$

where $f \in [0, 32] \times [0, 32]$ is a two-dimensional frequency vector. The first term of (4.35) tends to separate out mid-frequency eigenvalues in the covariance of the process to be realized. This is done because the high-frequency eigenvalues tend to be small, and the low-frequency modes are difficult to separate out with a FFT-based preconditioner because of edge effects. The second term of (4.35) introduces a shift so that the preconditioner doesn't have a null space.

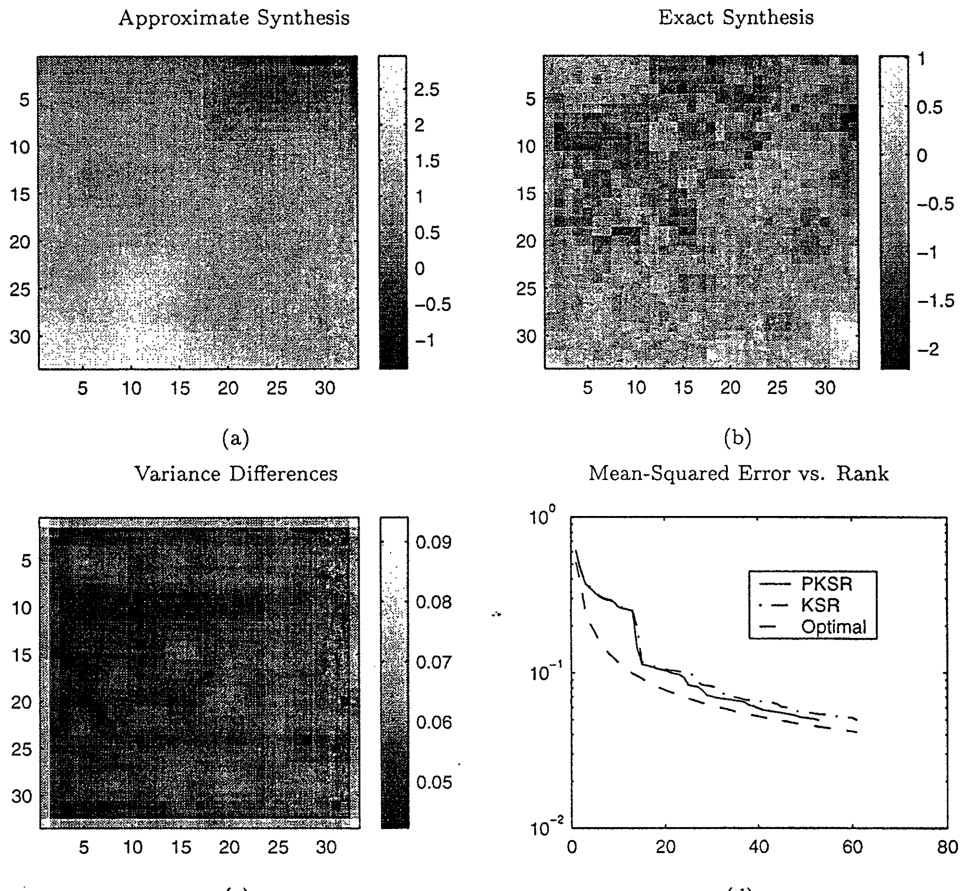


Figure 4.5. These results are for realizing samples on a 33 × 33 grid of a process with a two-dimensional spherical covariance given by (4.33). Part (a) shows a sample field generated with 14 iterations of a preconditioned Krylov subspace realization (KSR) algorithm, and part (b) shows a sample field generated with a matrix square root computation (an exact method). The difference between the variances are imaged in part (c). The true variance of the stationary process is 1. The fractions of error reduction obtained by both the preconditioned (PKSR) and un-preconditioned (KSR) Krylov subspace realization algorithm are plotted in part (d) as a function of rank. The optimal (KL) results are plotted for comparison.

Exact and approximate syntheses are pictured in parts (a) and (b) of Figure 4.5. The exact synthesis is generated by computing a matrix square-root as for the windowed-cosine covariance example. The approximate synthesis in part (a) is generated with 53 iterations of the preconditioned Krylov subspace realization algorithm. Note that the approximate realization is much smoother than the exact one. For moderate quality realizations, the Krylov subspace method tends to compute a very smooth one. This may or may not be desirable for certain applications. That is, there are applications for which one would prefer to have a rough realization with a high mean-squared error to one that has low mean-squared error and is smooth. However, for many applications, the ability of the Krylov subspace algorithm to pick a close to optimal mean-squared-error and, hence, smooth realization is desirable. Note that the differences in variances, plotted in part (c) are, for the most part, uniformly low. They ripple in the interior, indicating high-frequency terms have not been accounted for in the realization. The variance differences are also high at the edges and corners, which are apparently difficult to realize with finite-rank approximations.

Part (d) of Figure 4.5 shows how the preconditioned and un-preconditioned Krylov subspace realization algorithms compare to the optimal (KL) approach to low-rank realization. Note that the un-preconditioned algorithm is close to optimal but not as close as for the other examples. The preconditioned algorithm does better as the number of linear functionals used increases. This is because the preconditioner is primarily acting to separate out mid-frequency modes. The preconditioned and un-preconditioned algorithms are both far from optimal for ranks less than 20. The preconditioner doesn't help in this region because the modes that need to be captured are strongly influenced by edge effects, which the FFT-based preconditioner doesn't take into account. One can consider modifying the preconditioner to take the edge effects into account, but this hasn't been done here since it lies outside the scope of the thesis.

4.5 Methodological Comparisons

This chapter has presented four methods for obtaining low-rank realizations. The approach using Karhunen-Loève bases is optimal but not competitive with other methods in terms of computational complexity. The FFT-based methods tend to be the most efficient for synthesis. However, their range of applicability is limited since not every covariance matrix has a minimal circulant embedding that is positive semi-definite, as demonstrated by the examples in Sections 4.4.2 and 4.4.3. Moreover, the low-rank realizations generated by the FFT-based methods may not be as good as those generated by KL or Krylov subspace methods, as illustrated in Figure 4.1. Our proposed Krylov subspace realization algorithm can efficiently synthesize a process and compute low-rank approximations to covariance matrices provided that covariance matrix-vector multiplies can be efficiently implemented. The algorithm provides certain advantages over existing Lanczos methods for computing function of matrices. In particular, the Krylov subspace realization algorithm recursively computes the synthesis and the difference between the variances of the approximation and exact realization. The variance differences can be used as part of a stopping criterion. These differences would have to be recomputed at every step by the existing Lanczos methods for computing functions of matrices.

Chapter 5

A Krylov Subspace Method for Space-Time Estimation Problems

This chapter addresses the computational challenge of estimating physical phenomena varying in space and time from remotely sensed data. The specific example of such a problem considered in this chapter is that of estimating sea surface anomaly, the deviation of sea surface height from a mean, from data gathered by the TOPEX/ POSEIDON (T/P) altimeter. Although this specific example motivates much of the work in this chapter, the techniques we propose are widely applicable to large-scale, linear data assimilation problems.

The dynamics for the general class of problems under consideration can be written in the standard state-space form

$$x(t+1) = A(t)x(t) + w(t) \tag{5.1}$$

where $x(0)$ is zero mean with covariance Λ_x and $w(t)$ is zero mean with covariance $\Lambda_w(t)$. Typically, $x(t)$ is a vector of values of a physical quantity (such as sea surface anomaly) sampled spatially. The dynamics may incorporate a variety of mixing and transport terms. For example, a damped heat equation driven by noise w,

$$x_t = -\nabla^2 x - \alpha x + w, \tag{5.2}$$

when sampled, will lead to a dynamics matrix $A(t)$ that is a spatial discretization of $-\nabla^2 + (1-\alpha)$. Although not always necessary, we generally assume that the dynamics matrix in (5.1) is such that $\|A(t)\| \leq 1$, which ensures that the state of the unforced system does not grow without bound. This is often the case for models of physical phenomena. We also assume that the process noise $w(t)$ is uncorrelated in time but is spatially smooth. By spatially smooth, we mean that the process does not have a significant amount of white noise in it. This ensures that the smallest eigenvalues of $\Lambda_w(t)$ are much less than its largest eigenvalue (*i.e.* $\Lambda_w(t)$ has a high condition number). This assumption is critical for our proposed techniques to work. However, it is not very restrictive since the physical process being modeled is often smooth; so, the process noise must also be smooth.

The measurements we consider are linear with additive noise and so have the form

$$y(t) = C(t)x(t) + n(t) \tag{5.3}$$

where $C(t)$ is the measurement matrix and $n(t)$ is uncorrelated with $x(t)$ and has zero mean and covariance $\Lambda_n(t)$. For the T/P altimetric data, $C(t)$ consists of rows of the identity matrix because the data consist of point measurements taken only along satellite tracks. In general, $C(t)$ could take on a variety of structures. As for the process noise, we assume that $n(t)$ is temporally uncorrelated. Unlike the process noise, however, we do assume that $n(t)$ has a spatially white noise component in addition to any spatially correlated components. This ensures that the eigenvalues of $\Lambda_n(t)$ are bounded away from zero. This is not a very restrictive assumption since one rarely has almost perfect observations along a particular direction in measurement space. Thus, the class of problems we consider is very general.

Given dynamic and measurement models, we would like to produce linear least-squares estimates of $x(t)$ at each point in time given the data. The computational difficulties are two-fold. First, the problems are typically large. The state dimension is large because the state consists of samples of a process varying over space. State dimensions of $10^3 - 10^5$ and higher are common. Moreover, the measurement vectors are also typically high-dimensional for satellite remote sensing problems. Dimensions of 10-100% of the state dimension are common. The size of the problems prevents one from using standard techniques such as straightforward Kalman filter implementations [3, Section 3.3] since these require multiplication, inversion, and storage of matrices of the size of the state and measurement vector dimensions. Second, there is often a certain degree of irregularity in the estimation problem that prevents one from using Fourier methods [77, Section 2.2.3]. Specifically, the point measurements taken by satellite are often sparse and irregularly spaced. Since standard approaches to computing the desired estimates aren't appropriate because of the high measurement dimension, one must make use of efficient techniques that can exploit the structure of the problem.

We propose a method for solving the space-time estimation problems that incorporates the Krylov subspace algorithm for solving static estimation problems from Chapter 3. Recall that the Krylov subspace estimation algorithm computes both estimates *and* a representation of the error covariance matrix. The error covariance information is essential for space-time problems because it allows one to merge current state estimates with future data in an optimal manner.

Others have explored the use of Krylov subspace methods for solving Kalman filtering subproblems. In particular, Cohn, Todling, and Sivakumaran describe some approaches for using Lanczos algorithms, Krylov subspace methods for computing partial eigendecompositions, to compute reduced rank representations of the various error covariance and transition matrices [16,74,75]. These representations are used to perform some of the Kalman filtering computations more efficiently. Specifically, the prediction step of the Kalman filter, as described in the next section, is accelerated [16, pp. 64-66]. In contrast, our method focuses on using Krylov subspaces to perform *all* of the major computations in Kalman filtering. In particular, we use an algorithm similar to Lanczos for computing approximations to the prediction error covariance, and we use our Krylov subspace estimation algorithm, which is similar to conjugate gradient, for computing both updated estimates and error covariances. In both cases, the algorithms we employ are specialized to the estimation context and provide special advantages over Lanczos and conjugate gradient. In particular, the ability to simultaneously compute updated estimates and error covariances using a variant of conjugate gradient is new.

Another notable approach to solving large dynamical estimation problems is the work of Jaimoukha and Kassenally on using Krylov subspaces to solve large algebraic Riccati equations [42]. An algebraic Riccati equation is a nonlinear matrix equation whose solution is the error covariance matrix for a steady-state space-time estimation problem whose parameters do not vary in time. Jaimoukha and Kassenally consider solving large problems by projecting both the equation and the solution onto relatively low-dimensional Krylov subspaces. The methods proposed in this chapter are more widely applicable since they can be used to compute error covariance matrices for space-time estimation problems whose parameters are varying in time. Another difference between the work in this chapter and that of Jaimoukha and Kassenally is that the matrices used to generate the Krylov subspaces have different structure. In particular, they make use of the block Krylov subspaces $\mathcal{K}(A, \Lambda_w^{1/2}, k)$ (see Section 3.4.2 for an introduction to block Krylov subspaces), which can be considered reachability Gramians for the system (5.1). Note that these subspaces do not depend on the form of the measurements. In contrast, the Krylov subspaces used in this chapter do depend on the measurements, as one would expect would be necessary to compute accurate approximations for various types of measurements.

The details of how we propose to use Krylov subspace methods for solving space-time estimation problems are presented in two parts, in Sections 5.1 and 5.3. Some numerical stability issues are discussed in Section 5.2. The computational complexity is analyzed in Section 5.4. Finally, the performance of the algorithm is characterized in Section 5.5.

■ 5.1 Krylov Subspace Kalman Filtering

The starting point for the derivation of our algorithm for solving the space-time estimation problem outlined in the introduction is the standard discrete-time Kalman filter [43,47]. Recall that the Kalman filter recursively computes a sequence of estimates of $x(t)$ given data up to time t, $\hat{x}(t|t)$, termed *updated estimates*; another sequence of estimates of $x(t)$ but given data up to time $t-1$, $\hat{x}(t|t-1)$, termed *predicted estimates*; and the associated error covariances $\Lambda_e(t|t)$ and $\Lambda_e(t|t-1)$. The recursion is a two-step procedure, involving an update and prediction step at each point in time. The update is typically written in terms of the innovation

$$\nu(t) = y(t) - C(t)\hat{x}(t|t-1), \tag{5.4}$$

which is the residual in the predicted measurement given data up to time $t-1$. The covariance of the innovation,

$$\Lambda_\nu(t) = C(t)\Lambda_e(t|t-1)C^T(t) + \Lambda_n(t). \tag{5.5}$$

Each update estimates the error in the predicted estimate from the innovation and adds the correction:

$$\hat{x}(t|t) = \hat{x}(t|t-1) + \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t)\nu(t) \tag{5.6}$$

$$\Lambda_e(t|t) = \Lambda_e(t|t-1) - \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t)C(t)\Lambda_e(t|t-1). \tag{5.7}$$

Each prediction propagates the updated estimate one time step:

$$\hat{x}(t+1|t) = A(t)\hat{x}(t|t) \tag{5.8}$$

$$\Lambda_e(t+1|t) = A(t)\Lambda_e(t|t)A^T(t) + \Lambda_w(t). \tag{5.9}$$

These recursions are initialized with $\hat{x}(0|-1) = 0$, the the prior mean on $x(0)$, and $\Lambda_e(0|-1) = \Lambda_x$, a prior covariance on $x(0)$ that needs to be specified. We will perform each of the steps using the Krylov subspace methods outlined in Chapters 3 and 4.

Since the first update is a straightforward static estimation problem, one can make use of the Krylov subspace estimation algorithm, Algorithm 3.1.1 described in Section 3.1. This approach will be efficient provided that Λ_x-, $C(0)$-, and Λ_n-vector multiplies are efficient and that not too many iterations are required. After $k_u(0)$ iterations, the algorithm has computed an estimate of $x(0)$, $\hat{x}_{k_u(0)}(0|0)$; linear functionals $u_1(0), \ldots, u_{k_u(0)}(0)$; and the filtered backprojected linear functionals $r_1(0), \ldots, r_{k_u(0)}(0)$. The filtered backprojected linear functionals are then used in the subsequent prediction step.

In terms of the $r_1(0), \ldots, r_{k_u(0)}(0)$, the approximation to the update error covariance at time 0 is

$$\Lambda_x - \sum_{i=1}^{k_u(0)} r_i(0)r_i^T(0). \tag{5.10}$$

Propagating this one step ahead yields:

$$A(0)(\Lambda_x - \sum_{i=1}^{k_u(0)} r_i(0)r_i^T(0))A^T(0) + \Lambda_w(0). \tag{5.11}$$

Computing this explicitly is not feasible because of the size of the problem, nor is an explicit representation useful for subsequent update steps. Instead, we use the Krylov subspace realization algorithm, Algorithm 4.2.1 described in Section 4.2, to generate a low-rank approximation to the matrix in (5.11). Each iteration of the algorithm requires multiplying this matrix by a vector. Thus, each iteration is efficient provided that $A(0)$,

Λ_x, and $\Lambda_w(0)$-vector multiplies are efficient and $k_u(0)$ is reasonably small. The entire procedure is efficient if, in addition, the number of iterations required, $k_p(1)$, is not too large. If that is the case, then the filtered backprojected linear functionals generated by the algorithm, $f_1(1), \ldots, f_{k_p(1)}(1)$, form a low-rank approximation to the prediction error covariance:

$$\Lambda_{e,k_p(1)}(1|0) = \sum_{i=1}^{k_p(1)} f_i(1) f_i^T(1). \tag{5.12}$$

This can be used in the subsequent update step.

One can again use the Krylov subspace estimation algorithm to perform the second update. The prior covariance is the low-rank approximation to the prediction error covariance (5.12). Each iteration of the algorithm is efficient provided that $C(1)$-, $\Lambda_n(1)$-, and $\Lambda_{e,k_p(1)}(1|0)$-vector multiplies are efficient. The last multiplication is efficient provided the rank of the approximation in (5.12), $k_p(1)$, is not too large. The algorithm will generate an estimate of $x(1)$ given data up to time 1, $\hat{x}_{k_u(1)}(1|1)$; linear functionals $u_1(1), \ldots, u_{k_u(1)}(1)$; and a sequence of filtered back projected search directions $r_1(1), \ldots, r_{k_u(1)}(1)$. These search directions are used in the subsequent prediction step.

In terms of $r_1(1), \ldots, r_{k_u(1)}(1)$, the approximation to the update error covariance at time 1 is

$$\sum_{i=1}^{k_p(1)} f_i(1) f_i^T(1) - \sum_{i=1}^{k_u(1)} r_i(1) r_i^T(1). \tag{5.13}$$

Propagating this one step ahead yields:

$$A(1) \left(\sum_{i=1}^{k_p(1)} f_i(1) f_i^T(1) - \sum_{i=1}^{k_u(1)} r_i(1) r_i^T(1) \right) A^T(1) + \Lambda_w(1). \tag{5.14}$$

Again, this is too large and complicated a matrix to compute explicitly. However, one can use the Krylov subspace realization algorithm again to find a low-rank approximation to the matrix in (5.14). For each iteration of the algorithm to be efficient, $A(1)$ and $\Lambda_w(1)$-vector multiplies must be efficient. Moreover, the ranks of the two previously approximated matrices, $k_p(1)$ and $k_u(1)$, must not be too large so that vectors can be efficiently multiplied by the corresponding matrices. The filtered backprojected search directions generated by this algorithm, $f_1(2), \ldots, f_{k_p(2)}(2)$, form a low-rank approximation to the predicted error covariance:

$$\Lambda_{e,k_p(2)}(2|1) = \sum_{i=1}^{k_p(2)} f_i(2) f_i^T(2). \tag{5.15}$$

One can then continue in this manner to perform updates and predictions. The algorithm is summarized, as follows.

Algorithm 5.1.1. *Krylov subspace Kalman filter.*

1. *Initialize:*

   (a) *Update. Compute initial estimate* $\hat{x}_{k_u(0)}(0|0)$*; search directions* $u_1(0), \dots, u_{k_u(0)}(0)$*; and filtered back-projected search directions* $r_1(0), \dots, r_{k_u(0)}(0)$ *using the Krylov subspace estimation routine for prior covariance* Λ_x*, measurement matrix* $C(0)$*, and noise covariance* $\Lambda_n(0)$*.*

   (b) *Predict. Compute* $\hat{x}_{k_p(1)}(1|0) = A(0)\hat{x}_{k_u(0)}(0|0)$*. Generate filtered back-projected search directions* $f_1(1), \dots, f_{k_p(1)}(1)$ *by using the Krylov subspace realization algorithm to compute a low-rank approximation to the covariance matrix*

$$A(0)\left(\Lambda_0 - \sum_{i=1}^{k_u(0)} r_i(0) r_i^T(0)\right) A^T(0) + \Lambda_w(0).$$

2. *Repeat at each time step:*

   (a) *Update. Compute updated estimate* $\hat{x}(t|t)$*; search directions* $u_1(t), \dots, u_{k_u(t)}(t)$*; and filtered back-projected search directions* $r_1(t), \dots, r_{k_u(t)}(t)$ *using the Krylov subspace estimation routine for prior covariance* $\sum_{i=1}^{k_p(t)} f_i(t) f_i(t)^T$*, measurement matrix* $C(t)$*, and noise covariance* $\Lambda_n(t)$*.*

   (b) *Predict. Compute* $\hat{x}_{k_p(t+1)}(t+1|t) = A(t)\hat{x}_{k_u(t)}(t|t)$*. Generate filtered back-projected search directions* $f_1(t+1), \dots, f_{k_p(t+1)}(t+1)$ *by using the Krylov subspace realization algorithm to compute a low-rank approximation to the covariance matrix*

$$A(t)\left(\sum_{i=1}^{k_p(t)} f_i(t) f_i^T(t) - \sum_{i=1}^{k_u(t)} r_i(t) r_i^T(t)\right) A^T(t) + \Lambda_w(t).$$

■ 5.2 Error Analysis

The Krylov subspace method for Kalman filtering outlined in the previous section introduces approximations to exact Kalman filtering steps at various points. One would like to know how these approximations propagate through the filter. The subsequent analysis characterizes the error propagation. The theory provides some guarantees of stability and suggests how good the approximations need to be at every time step to guarantee good overall performance.

5.2.1 Predicted Error Covariance Perturbations

An important aspect in the application of the Krylov subspace method for Kalman filtering is the specification of the quality of approximation to the predicted error covariance. This is especially true for the remote sensing problems that motivate the development of the algorithm. These problems tend to have irregularly spaced, incomplete measurements of very good quality. This structure can lead to the propagation of significant approximation errors if too few iterations of the Krylov subspace method are used. The errors occur when performing an update after generating an approximation to the prediction error covariance that is too poor. This behavior is best understood by analyzing how the linear least-squares estimate and error covariance for a static estimation problem change as the prior covariance is perturbed.

Specifically, consider the static linear least-squares estimation problem of estimating x from measurements

$$y = Cx + n \tag{5.16}$$

where x and n are uncorrelated and have zero mean and covariances Λ_x and Λ_n, respectively. Now, suppose that Λ_x is perturbed additively by a matrix Δ. One can then calculate approximations to the perturbed estimate and error covariance by ignoring terms that are second-order or higher in Δ. In particular, one can approximate the inverse of $M + \Delta$ for a given matrix M by

$$M^{-1} - M^{-1}\Delta M^{-1}. \tag{5.17}$$

Now, the perturbed estimate is given by

$$\begin{aligned}(\Lambda_x + \Delta)C^T(C(\Lambda_x + \Delta)C^T + \Lambda_n)^{-1}y &\approx (\Lambda_x + \Delta)C^T\left((C\Lambda_xC^T + \Lambda_n)^{-1} - \right.\\ &\quad \left.(C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1}\right)y\\ &= \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}y +\\ &\quad \Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1}y -\\ &\quad (\Lambda_x + \Delta)C^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T \times\\ &\quad (C\Lambda_xC^T + \Lambda_n)^{-1})y\\ &\approx \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}y +\\ &\quad \Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1}y -\\ &\quad \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T \times\\ &\quad (C\Lambda_xC^T + \Lambda_n)^{-1})y.\end{aligned} \tag{5.18}$$

The first term of the final approximation in (5.18) is the correct estimate, and the other terms are the resultant perturbation. A sufficient condition to keep the error terms in

(5.18) small is to require that

$$\|\Delta\| \ll (\lambda_{\min}(\Lambda_n))^2. \tag{5.19}$$

Likewise, the perturbed error covariance is given by

$$\begin{aligned}
&(\Lambda_x + \Delta) - (\Lambda_x + \Delta)C^T(C(\Lambda_x + \Delta)C^T + \Lambda_n)^{-1}C(\Lambda_x + \Delta) \\
&\approx (\Lambda_x + \Delta) - (\Lambda_x + \Delta)C^T \times \\
&((C\Lambda_xC^T + \Lambda_n)^{-1} - (C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1})C(\Lambda_x + \Delta) \\
&= \Lambda_x - \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Lambda_x + \\
&\Delta - \Delta C^T((C\Lambda_xC^T + \Lambda_n)^{-1} - (C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1})C(\Lambda_x + \Delta) - \\
&(\Lambda_x + \Delta)C^T((C\Lambda_xC^T + \Lambda_n)^{-1} - (C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1})C\Delta - \\
&\Lambda_xC^T((C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1})C\Lambda_x \\
&\approx \Lambda_x - \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Lambda_x + \\
&\Delta - \Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Lambda_x - \Lambda_xC^T(C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta - \\
&\Lambda_xC^T((C\Lambda_xC^T + \Lambda_n)^{-1}C\Delta C^T(C\Lambda_xC^T + \Lambda_n)^{-1})C\Lambda_x.
\end{aligned} \tag{5.20}$$

For the perturbation in (5.20) to be small, one needs only that (5.19) hold again. The unusual implication of the analysis is that Δ must be small not only relative to the matrix Λ_x being perturbed, but also to the minimal noise variance represented by $\lambda_{\min}(\Lambda_n)$.

The following example illustrates how the perturbation may come about and provides some insight into the behavior of the Krylov subspace method for Kalman filtering when applied to remote sensing problems. The scenario is as follows. Suppose $X(t)$ is a stationary process on $[0, 1]$ with covariance function

$$\operatorname{Cov}(X(t), X(t+\tau)) = e^{-\tau^2}. \tag{5.21}$$

Let the vector x consist of l samples of $X(t)$ taken at intervals $\delta t > 0$

$$x = \begin{bmatrix} X(0) & \cdots & X(l\delta t) \end{bmatrix}^T. \tag{5.22}$$

Furthermore, suppose y consists of two measurements taken at consecutive sampling points with measurement noise n whose components are uncorrelated and of equal intensity σ^2, much less than one:

$$y = \begin{pmatrix} X(0) \\ X(\delta t) \end{pmatrix} + n. \tag{5.23}$$

These statistics of x and y are similar to those appearing in the update step of a Kalman filter for a remote sensing problem. In particular, the prediction errors are often smooth, and the measurements taken are pointwise and clustered in certain regions.

How the estimate of x given y behaves given these statistics is the subject of the following analysis. The intuition in this example is that, in the limit of low measurement noise, the estimator will simply extrapolate the two almost perfect measurements of x using a Taylor series expansion. The behavior of the estimator is very much like that of a linear extrapolation of two data points which are much closer to each other than to the points at which one is evaluating the extrapolation. In particular, the estimate of x will be a low-frequency function that is highly dependent on the differences between the two data points, y_0 and y_1. The details are as follows.

The covariance of the measured piece of x is

$$C\Lambda_x C^T = \begin{pmatrix} 1 & e^{-\delta t^2} \\ e^{-\delta t^2} & 1 \end{pmatrix}, \tag{5.24}$$

and the data covariance is

$$\Lambda_y = \begin{pmatrix} 1+\sigma^2 & e^{-\delta t^2} \\ e^{-\delta t^2} & 1+\sigma^2 \end{pmatrix}. \tag{5.25}$$

Let $\Delta = \varepsilon I$ be a perturbation of Λ_x, and

$$\Lambda_y' = \begin{pmatrix} 1+\sigma^2+\varepsilon & e^{-\delta t^2} \\ e^{-\delta t^2} & 1+\sigma^2+\varepsilon \end{pmatrix} \tag{5.26}$$

be the perturbed measurement covariance. Then,

$$(\Lambda_y')^{-1} = \begin{pmatrix} 1+\sigma^2+\varepsilon & -e^{-\delta t^2} \\ -e^{-\delta t^2} & 1+\sigma^2+\varepsilon \end{pmatrix} \frac{1}{1-(e^{-\delta t^2})^2+(\varepsilon+\sigma^2)(2+\varepsilon+\sigma^2)}. \tag{5.27}$$

Fix δt and σ^2 with $\sigma^2 \ll 1 - e^{\delta t^2} \ll 1$ and consider the behavior of the perturbed estimation problem as ε varies near 0. The product

$$(\Lambda_y')^{-1}y \approx \frac{1}{1-(e^{-\delta t^2})^2+(\varepsilon+\sigma^2)(2+\varepsilon+\sigma^2)} \begin{pmatrix} y_0 - y_1 \\ y_1 - y_0 \end{pmatrix}, \tag{5.28}$$

which consists of approximate first differences of the data. Note that this product is highly dependent on the value of ε through the denominator in (5.28). In particular, a perturbation $\varepsilon \sim \sigma^2$ will perturb $(\Lambda_y')^{-1}$ significantly, which, in turn, will alter the perturbed estimate of x,

$$(\Lambda_x + \Delta)(\Lambda_y')^{-1}y. \tag{5.29}$$

The effect on the estimate is minimal if

$$\|\Delta\| = \varepsilon \ll \sigma^4 = (\lambda_{\min}(\Lambda_n))^2 \tag{5.30}$$

in accordance with the previous general discussion.

Note that for $\varepsilon \sim \sigma^2$, the perturbation Δ will not significantly affect the estimate of x_0 and x_1, the two quantities being directly measured. This follows from the fact that the perturbed estimates $\hat{x}'_0$ and $\hat{x}'_1$ are given by

$$\begin{pmatrix} \hat{x}'_0 \\ \hat{x}'_1 \end{pmatrix} = C(\Lambda_x + \Delta)C^T(\Lambda'_y)^{-1}y \approx y \tag{5.31}$$

since $\varepsilon, \sigma^2 \ll 1 - e^{-\delta t^2}$. One will only observe perturbations in the estimates of components of x not directly measured. Thus, these effects are especially prominent in applications with scattered point measurements and low measurement noise.

The consequence of this analysis for applying the Krylov subspace method for Kalman filtering is that one must ensure that the perturbation in the prediction error covariance is at least kept smaller than $(\lambda_{\min}(\Lambda_n(t)))^2$, in order to ensure that perturbations in the update results are kept small. In the case for which $\Lambda_n(t) = \sigma^2 I$ at every time step, one can keep the perturbation small by stopping the Krylov subspace realization algorithm at the prediction steps when the quality of the approximation, as measured by (4.23), falls significantly below σ^2. In some cases, one may be able to exploit structure in the problem to arrive at a modified form of the Krylov subspace method for Kalman filtering that introduces less perturbation into the prediction error covariance. Such an approach is used in the oceanographic example in Section 5.5.2.

5.2.2 Filter Stability

In addition to understanding how the approximations in the Krylov subspace method for Kalman filtering affect each update step, one is interested in how the approximations propagate through the dynamics of the filter. The updated estimates obey the following dynamics

$$\hat{x}(t+1|t+1) = \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t)C^T(t)A(t)\hat{x}(t|t) + f(t) \tag{5.32}$$

where $f(t)$ is a forcing term proportional to the data. The approach taken here to understanding how the effect of approximations propagate through the filter is to examine the unforced dynamics,

$$z(t+1) = \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t)C^T(t)A(t)z(t). \tag{5.33}$$

where $z(t)$ starts off in some state $z(0)$.

There is an extensive theory built up that provides conditions for stability of the Kalman filter dynamics [43,45,46]; however, much of it does not apply to the scenario under consideration. Most of the existing stability theory focuses on exponential stability. The dynamics are exponentially stable if there exist positive constants c_1 and c_2 such that

$$\|z(t)\| \leq c_1 e^{-c_2 t}. \tag{5.34}$$

for all $z(0)$ where $z(t)$ obeys (5.33). Commonly stated sufficient conditions for exponential stability include that the system be uniformly reachable from the noise, *i.e.*, there exist positive constants T, α, t_0 such that

$$\sum_{\tau=t}^{t+T} \Phi(t+T,\tau)\Lambda_w(\tau)\Phi^T(t+T,\tau) \geq \alpha I \tag{5.35}$$

for all $t \geq t_0$, where $\Phi(t,\tau)$ is the state transition matrix of (5.1). However, the driving noise in remote sensing problems is often spatially smooth at every time. Thus, some of the eigenvalues of Λ_w may be very small. The system may be technically reachable from the noise; however, the constant α in (5.35) may have to be very close to zero. Moreover, the resulting system may technically be exponentially stable, but the decay rate c_2 in (5.34) may be very slow.

This behavior is a consequence of the estimation problem being fundamentally infinite dimensional. One needs to consider the situation in which the quantity to be estimated, $x(t)$, is not just samples of a process varying in space, but is a process varying in space. In this setting, one can precisely state the type of stability that the Kalman filter exhibits.

The specific setting is a Hilbert space framework. Let the Hilbert space X be the space in which the state of the system, $x(t)$, takes on values. This is typically a space of functions over a two or higher dimensional region, such as an L^2 space. Let the Hilbert space Y represent the measurement space, which might be be finite or infinite dimensional. The operators $A : X \mapsto X$, $C : X \mapsto Y$, $\Lambda_n : Y \mapsto Y$, and $\Lambda_w : X \mapsto X$ are all assumed to be bounded linear mappings and have the same roles as in the finite dimensional setting. In addition, the operator $\Lambda_n(t)$ is assumed to have a bounded inverse for each time t. In this framework, a symmetric operator M is considered to be positive-definite if

$$\langle z', Mz' \rangle > 0 \qquad \forall z' \neq 0, \tag{5.36}$$

and a partial ordering on symmetric positive-definite operators is given by

$$M > N \Longleftrightarrow M - N > 0. \tag{5.37}$$

The type of convergence that will be studied in this chapter is strong convergence. A sequence u_i converges to appoint u_* strongly if

$$\lim_{i\to\infty} \|u_i - u_*\| = 0. \tag{5.38}$$

Thus, a system is considered strongly stable if its state $z(t)$ converges strongly to 0, *i.e.*

$$\lim_{t\to\infty} \|z(t)\| = 0. \tag{5.39}$$

Additionally, a sequence of linear operators U_i is said to converge to a operator U_* strongly if

$$\lim_{i\to\infty} \|(U_i - U_*)z'\| = 0 \qquad \forall z' \in X. \tag{5.40}$$

The Hilbert space framework used in this chapter can capture the behavior of the Kalman filter for many types of estimation problems under consideration, but there is one important exception. That is, isolated point measurements are not necessarily included in the theory when X is an L^2 space on an open subset of a finite-dimensional real space. This is because evaluating a function at a point is not a bounded linear mapping. An example of a measurement structure that is included in this framework for such a space is the following. Suppose X is an $L^2(O)$ space where O is an open subset of a finite-dimensional real space. Moreover, suppose that measurements are taken pointwise over a set $M \subset O$:

$$y_s = x_s + n_s \qquad s \in M \tag{5.41}$$

where y is the measurement, x is the quantity to be measured, and n is noise. Then, this type of measurement is included in the framework of this chapter provided M is open. That M is open implies that no measurements are made at isolated points. Examples of measurement structures for which M is open are data of ocean state (such as sea surface temperature) taken along swaths of the ocean by a satellite.

The following result guarantees stability of the Kalman filter given lower and upper bounds on the measurement quality. The measurement quality over an interval $[t-T, t]$ is measured by the observability Grammian

$$\mathcal{I}(t, t-T) \triangleq \sum_{\tau=t-T}^{t} \Phi(\tau,t)C^*(\tau)\Lambda_n^{-1}C(\tau)\Phi(\tau,t) \tag{5.42}$$

where $\Phi(\tau, t)$, again, is the state transition operator of (5.1).

The proof of the following result and more discussion of stability issues regarding the filter dynamics are left for Chapter 6.

Theorem 5.2.1. *Suppose that the dynamics matrix is bounded above and below,* i.e. *that there exist constants $\gamma_1, \gamma_2 > 0$ such that*

$$\|A^{-1}(t)\| \leq \frac{1}{\gamma_1} \qquad \forall t \tag{5.43}$$

$$\|A(t)\| \leq \gamma_2 \qquad \forall t \tag{5.44}$$

and suppose that the system is uniformly observable, i.e. *that there exist constants $\beta_1, \beta_2, T > 0$ such that*

$$\beta_1 I \leq \mathcal{I}(t, t-T) \leq \beta_2 I \qquad \forall t \geq T \tag{5.45}$$

Moreover, suppose that the prior covariance on x is positive-definite:

$$\Lambda_x > 0, \tag{5.46}$$

and that the measurement noise covariance is bounded below, i.e. *that there exists a constant $\sigma^2 > 0$ such that*

$$\Lambda_n(t) \geq \sigma^2 I \qquad \forall t. \tag{5.47}$$

Then, the dynamics of the Kalman filter are strongly stable, i.e.

$$\lim_{t\to\infty} \|z(t)\| = 0 \tag{5.48}$$

where $z(t)$ obeys the dynamics (5.33).

Theorem 5.2.1 characterizes the propagation of the errors in the estimates but not the error covariances. The error covariances are much more difficult to analyze, and no proof of stability in their dynamics is offered here. However, we do conjecture that for time-invariant problems, the error covariances tend to a fixed point under mild conditions.

Conjecture 5.2.1. *Consider a time-invariant system,* i.e. *one such that $A(t)$, $C(t)$, $\Lambda_n(t)$, and $\Lambda_w(t)$ are constant over time. Suppose that the dynamics matrix is bounded above and below,* i.e. *that there exist constants $\gamma_1, \gamma_2 > 0$ such that*

$$\|A^{-1}(t)\| \leq \frac{1}{\gamma_1} \qquad \forall t \tag{5.49}$$

$$\|A(t)\| \leq \gamma_2 \qquad \forall t \tag{5.50}$$

and suppose that the system is uniformly observable, i.e. *that there exist constants $\beta_1, \beta_2, T > 0$ such that*

$$\beta_1 I \leq \mathcal{I}(t, t-T) \leq \beta_2 I \qquad \forall t \geq T \tag{5.51}$$

Moreover, suppose that the prior covariance on x is positive-definite:

$$\Lambda_x > 0, \tag{5.52}$$

and that the measurement noise covariance is bounded below, i.e. *that there exists a constant $\sigma^2 > 0$ such that*

$$\Lambda_n(t) \geq \sigma^2 I \qquad \forall t. \tag{5.53}$$

Then, the error covariance $\Lambda_e(t|t)$ tends to a steady-state $\Lambda_e(\infty)$ strongly, i.e.

$$\lim_{t\to\infty} \|(\Lambda_e(t|t) - \Lambda_e(\infty))z'\| = 0 \qquad \forall z' \in X. \tag{5.54}$$

Were this conjecture correct, one implication would be that perturbations in the error covariance computations would be damped out over time. Specifically, if the updated error covariance were perturbed slightly only at one time step, the perturbed resulting error covariance, $\Lambda'_e(t|t)$ would have the same limit as the unperturbed error covariance:

$$\lim_{t\to\infty} \|(\Lambda'_e(t|t) - \Lambda_e(\infty))z'\| = \lim_{t\to\infty} \|(\Lambda_e(t|t) - \Lambda_e(\infty))z'\| = 0 \qquad \forall z' \in X. \tag{5.55}$$

Thus, the error covariance calculations would damp out perturbations when the matrices in the estimation problem are time-invariant. We have, in fact, observed this in our numerical work.

■ 5.3 Smoothing

One is sometimes interested not only in the filtered estimates of $x(t)$ but also the smoothed estimates over a given period of time. The smoothed estimate $\hat{x}(t|T)$ is the estimate of $x(t)$ given data $y(s)$ for $s \in [0, T]$ for a fixed maximum time T, and $\Lambda_e(t|T)$ is the associated error covariance. One can compute $\hat{x}(t|T)$ and $\Lambda_e(t|T)$ in terms of quantities already computed by the Kalman filter as it proceeds up to time T.

The modified Bryson-Frazier smoother [10] is considered to be the most efficient recursive smoothing algorithm for many applications [9,11,49]. The smoothed estimate is written as a sum of the predicted estimate, summarizing data from the past, and another quantity $\hat{v}(t|T)$ that summarizes information from the future:

$$\hat{x}(t|T) = \hat{x}(t|t-1) + \Lambda_e(t|t-1)\hat{v}(t|T). \tag{5.56}$$

One computes $\hat{v}(t|T)$ using the following backwards recursion:

$$\hat{v}(t-1|T) = F^T(t-1)\hat{v}(t|T) + C^T(t-1)\Lambda_\nu^{-1}(t-1)\nu(t-1) \tag{5.57}$$

where $F(t)$ is the Kalman filter error dynamics matrix,

$$F(t) = A(t)(I - \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t-1)C(t)), \tag{5.58}$$

$\Lambda_\nu(t-1)$ is the innovation covariance defined by (5.5), and $\nu(t-1)$ is the innovation defined by (5.4). The recursion for $\hat{v}(t|T)$ is initialized with

$$\hat{v}(T|T) = C^T(T)\Lambda_\nu^{-1}(y(T) - C(T)\hat{x}(T|T-1)). \tag{5.59}$$

Likewise, the smoothed error variances are written as the difference between the predicted errors and terms $V(t,T)$ that summarize the reduction from the future:

$$\Lambda_e(t|T) = \Lambda_e(t|t-1) - \Lambda_e(t|t-1)V(t,T)\Lambda_e(t|t-1). \tag{5.60}$$

The following backwards recursion computes $V(t,T)$:

$$V(t-1,T) = F^T(t-1)V(t,T)F(t-1) + C^T(t-1)\Lambda_\nu^{-1}(t-1)C(t-1) \tag{5.61}$$

for the initialization

$$V(T,T) = C^T(T)\Lambda_\nu^{-1}(T)C(T). \tag{5.62}$$

One can approximate matrices in the recursion for the smoothed estimates, (5.56)-(5.59), in terms of quantities computed by the Krylov subspace Kalman filtering algorithm, Algorithm 5.1.1. In particular,

$$F(t) \approx A(t)\left(I - \left(\sum_{i=1}^{k_u(t)} r_i(t)u_i^T(t)\right)C(t)\right) \triangleq F_{k_u(t)}(t) \tag{5.63}$$

and

$$\Lambda_\nu^{-1}(t) \simeq \sum_{i=1}^{k_u(t)} u_i(t)u_i^T(t) \triangleq \Lambda_{\nu,k_u(t)}^{-1}(t) \tag{5.64}$$

where $u_i(t)$ are the search directions generated at the update step, and $r_i(t)$ are the associated filtered back-projected search directions, as described in Section 5.1. The latter approximation means that the two matrices are approximately equal when restricted to the subspace of primary interest when estimating $x(t)$. Specifically

$$\Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t) \approx \Lambda_e(t|t-1)C^T(t)\Lambda_{\nu,k_u(t)}^{-1}(t). \tag{5.65}$$

This follows from the fact that the matrix on the left in (5.65) is the gain matrix for the update step at time t, and the matrix on the right would be the approximation generated by the Krylov subspace estimation algorithm if $\Lambda_e(t|t-1) = \Lambda_{e,k_p(t)}(t|t-1)$. The approximation in (5.65) is useful provided terms depending on $C^T(t)\Lambda_\nu^{-1}(t)$ are pre-multiplied by matrices whose dominant eigenspaces are the same as those of $\Lambda_e(t|t-1)$. For the computation of $\hat{x}(t|T)$, the matrix $C^T(t)\Lambda_\nu^{-1}(t)$ in (5.57) is pre-multiplied by $\Lambda_e(t|t-1)$ in (5.56). However, earlier estimates $\hat{x}(s|T)$ for $s < t$ involve pre-multiplying $C^T(t)\Lambda_\nu^{-1}(t)$ by

$$\Lambda_e(s|s-1)F^T(s)F^T(s+1)\cdots F^T(t-1) \tag{5.66}$$

via (5.56) and the recursion (5.57). Thus, one can use the approximation (5.65) to compute smoothed estimates only if the matrices $F^T(t)$ do not significantly alter the dominant eigenspace of $\Lambda_e(t|t-1)$ and the dominant eigenspaces of $\Lambda_e(t|t-1)$ do not change significantly from time step to time step. This does appear to be the case for the examples in Section 5.5.

For such cases, one can use these approximations of terms in (5.56)-(5.59) to approximate the smoothed estimates, as follows. First, one can compute an approximation to $\hat{v}(t|T)$ with the recursion

$$\begin{aligned}\hat{v}_{k_u(t-1)}(t-1|T) = F_{k_u(t-1)}^T(t-1)\hat{v}_{k_u(t)}(t|T) + \\ C^T(t-1)\Lambda_{\nu,k_u(t-1)}^{-1}(t-1)(y(t-1) - C(t-1)\hat{x}(t-1|t-2))\end{aligned} \tag{5.67}$$

for the initialization

$$\hat{v}_{k_u(T)}(T|T) = C^T(T)\Lambda_{\nu,k_u(T)}(y(T) - C(T)\hat{x}(T|T-1)). \tag{5.68}$$

Then,

$$\hat{x}_{k_u(t)(t|T)} = \hat{x}_{k_p(t)}(t|t-1) + \left(\sum_{i=1}^{k_p(t)} f_i(t)f_i^T(t)\right)\hat{v}_{k_u(t)}(t|T) \tag{5.69}$$

is an approximation to the smoothed estimate at t given all data up to time T.

Computing an approximation to the smoothed errors can also be done in a fashion that reuses quantities already computed by the Krylov subspace Kalman filtering algorithm. However, one must perform an additional Krylov subspace computation at each time step to reduce the dimensionality of the approximation to the covariance matrix $V(t,T)$. Unfortunately, $V(t,T)$ is not approximately low-rank. However, only relatively few modes of $V(t,T)$ are needed to compute the reduction to $\Lambda_e(t,t-1)$ in (5.60) since $\Lambda_e(t|t-1)$ can be approximated by a low-rank matrix as is done for the filtering step described in Section 5.1. Let

$$V_{k_s(t)}(t|T) = \sum_{i=1}^{k_s(t)} v_i(t)v_i^T(t) \tag{5.70}$$

be the approximation to $V(t|T)$. Now, note that

$$\begin{aligned}\Lambda_e(t-1|T) = \Lambda_e(t-1|t-2) - (\Lambda_e(t-1|t-2)V(t-1,T))V^{-1}(t-1,T)\\ (V(t-1,T)\Lambda_e(t-1|t-2)),\end{aligned} \tag{5.71}$$

and that $V(t-1,T)$ can be written in terms of $V(t,T)$ using (5.61). One can approximate the reduction term in (5.71), *i.e.* the second term, by applying the Krylov subspace estimation algorithm to an estimation problem with data covariance

$$F_{k_u(t-1)}^T(t-1)V_{k_s(t)}(t|T)F_{k_u(t-1)}(t-1) + C^T(t-1)\Lambda_{\nu,k_u(t-1)}^{-1}(t-1)C(t-1) \tag{5.72}$$

and signal-data cross-covariance

$$\begin{aligned}\Lambda_{e,k_p(t)}(t|t-1)\Big(F_{k_u(t-1)}^T(t-1)V_{k_s(t)}(t|T)F_{k_u(t-1)}(t-1) +\\ C^T(t-1)\Lambda_{\nu,k_u(t-1)}^{-1}(t-1)C(t-1)\Big).\end{aligned} \tag{5.73}$$

The variances of the reduction term for this estimation problem are an approximation to those appearing in the smoothed error variance calculation (5.60).

Moreover, the $k_s(t-1)$ vectors

$$\begin{aligned}v_i(t-1) \triangleq \Big(F_{k_u(t-1)}^T(t-1)V_{k_s(t)}(t|T)F_{k_u(t-1)}(t-1) +\\ C^T(t-1)\Lambda_{\nu,k_u(t-1)}^{-1}(t-1)C(t-1)\Big)p_i,\end{aligned} \tag{5.74}$$

where p_i are the search directions generated by the Krylov subspace estimation algorithm, are such that

$$V(t-1|T) \simeq \sum_{i=1}^{k_s(t-1)} v_i(t-1)v_i^T(t-1) \triangleq V_{k_s(t-1)}(t-1|T). \qquad (5.75)$$

The approximation is made in a similar sense as for approximating $\Lambda_\nu(t)$ in (5.64). Specifically,

$$\Lambda_e(t-1|t-2)V(t-1,T)\Lambda_e(t-1|t-2) \approx \\ \Lambda_e(t-1|t-2)V_{k_s(t-1)}(t-1,T)\Lambda_e(t-1|t-2). \qquad (5.76)$$

The advantage of calculating an approximation to $V(t|T)$ in this fashion is that one can use the stopping criteria of Section 3.2 to determine the necessary number of iterations and, hence, rank of the approximation.

This approach to computing an approximation to $V(t|T)$ may not always work. Specifically, one of the assumptions of the Krylov subspace estimation algorithm, when applied to computing the reduction term in (5.71), is that $V(t-1,T)$ and

$$V(t-1,T)\Lambda_e(t-1|t-2)V(t-1,T) \qquad (5.77)$$

have the same eigenvectors. This may not always be the case, but, as discussed in Section 3.4.1, one can transform the problem using an appropriate preconditioner, to achieve convergence. For all of the examples in Section 5.5, however, no preconditioning was required.

A preconditioner that may work well is

$$\left(\sum_{i=1}^{k_p(t-1)} f_i(t)a_{RTS}^{k_p(t-1)-i+1}f_i^T(t)+I\right) \qquad (5.78)$$

for some constant $a_{RTS} > 1$. This preconditioner will tend to accentuate those modes relevant to reducing the error in $\hat{x}(t|t-1)$ by placing more weight on the filtered backprojected search directions $f_i(t)$ for smaller i. The preconditioning transformation in (5.78) has not been extensively tested, however.[1]

The smoothing algorithm is summarized, as follows.

Algorithm 5.3.1. *Krylov subspace smoother.*

1. *Initialize:*

$$\hat{v}_{k_u(T)}(T|T) = C^T(T)\Lambda_{\nu,k_u(T)}(y(T)-C(T)\hat{x}(T|T-1)). \qquad (5.79)$$

[1] A preconditioner of this form was used successfully for some preliminary attempts at processing some oceanography data. However, the data and model were poorly matched; so, the experiment is not discussed in Section 5.5.

2. *At each step, compute*

$$\begin{aligned}\hat{v}_{k_u(t-1)}(t-1|T) = F^T_{k_u(t-1)}(t-1)\hat{v}_{k_u(t)}(t|T) + \\ C^T(t-1)\Lambda^{-1}_{\nu,k_u(t-1)}(t-1)(y(t-1)-C(t-1)\hat{x}(t-1|t-2))\end{aligned} \tag{5.80}$$

   and

$$\hat{x}_{k_u(t-1)}(t-1|T) = \hat{x}_{k_p(t-1)}(t-1|t-2) + \Lambda_{e,k_p(t-1)}(t-1|t-2)\hat{v}_{k_u(t-1)}(t-1|T) \tag{5.81}$$

   for

$$F_{k_u(t)}(t) \triangleq A(t)\left(I - \left(\sum_{i=1}^{k_u(t)} r_i(t)u_i^T(t)\right)C(t)\right) \tag{5.82}$$

   and

$$\Lambda_{\nu,k_u(t)}(t) \triangleq \sum_{i=1}^{k_u(t)} u_i(t)u_i^T(t). \tag{5.83}$$

3. *Initialize,*

$$V_{k_s(T)}(T|T) = \sum_{i=1}^{k_u(T)} C^T(T)u_i(T)u_i^T(T)C(T) \tag{5.84}$$

4. *At each step,*

   (a) *Compute $v_1(t), \ldots, v_{k_s(t)}(t)$ using the Krylov subspace estimation algorithm for data covariance*

$$F^T_{k_u(t-1)}(t-1)V_{k_s(t)}(t|T)F_{k_u(t-1)}(t-1) + C^T(t-1)\Lambda^{-1}_{\nu,k_u(t-1)}(t-1)C(t-1), \tag{5.85}$$

   and signal-data cross-covariance

$$\begin{aligned}\Lambda_{e,k_p(t)}(t|t-1)F^T_{k_u(t-1)}(t-1)V_{k_s(t)}(t|T)F_{k_u(t-1)}(t-1) + \\ C^T(t-1)\Lambda^{-1}_{\nu,k_u(t-1)}(t-1)C(t-1),\end{aligned} \tag{5.86}$$

   possibly using the preconditioner

$$\left(\sum_{i=1}^{k_p(t-1)} f_i(t)a_{RTS}^{k_p(t-1)-i+1} f_i^T(t) + I\right). \tag{5.87}$$

   (b) *Compute smoothed error variances*

$$\left(\Lambda_{e,k_p(t)}(t|t-1)\right)_{ii} - \sum_{j=1}^{k_s(t)} \left(\left(\Lambda_{e,k_p(t)}(t|t-1)v_j(t)\right)_i\right)^2. \tag{5.88}$$

parameter	description
l	state dimension
m	measurement dimension
k_u	number of iterations at update step
k_p	number of iterations at predict step
k_s	number of iterations at smoothing step

Table 5.1. Parameters in the Computational Complexity Analysis

■ 5.4 Computational Complexity

The Krylov subspace methods for Kalman filtering and smoothing can provide a substantial speedup relative to standard implementations. The degree depends on the specific problem. In this section, we count the number of multiplication operations for each update, predict, and smoothing step. Recall that each of these steps is solved with a Krylov subspace iterative method. Thus, the multiplication count is a function of the number of iterations run at each update step, k_u; the number of iterations run at each predict step, k_p; and the number of each iterations run at each smoothing step, k_s in addition to the state dimension, l, and the measurement dimension, m (see Table 5.1). Each of these parameters is assumed to be constant for all time. The focus of the operation count is on terms which are cubic or higher in these parameters. Note that a standing assumption is that Λ_n-, Λ_w-, and C-vector products can all be performed using a number of multiplications that scales only linearly in either l or m.

■ 5.4.1 Comparison of Techniques

The analysis of filtering complexity is carried out in the next section. The conclusion is that the computational gain of using the Krylov subspace method over straightforward direct methods is

$$\frac{m^3/6 + 2m^2 l}{mk_u^2 + lk_p^2 + 2lk_p(k_p + k_u) + 2k_p k_u l}. \tag{5.89}$$

Thus, one has a gain if

$$k_p^2 + 2k_p(k_p + k_u) + 2k_p k_u < 2m^2 \tag{5.90}$$

and

$$k_u^2 < \frac{m^2}{6}. \tag{5.91}$$

The latter constraint implies $k_u \lessapprox 0.41m$. Since one expects $k_u \leq k_p$, a worst case scenario for constraint (5.90) is that $k_u = k_p$, in which case one requires

$$k_p \leq \sqrt{\frac{2}{7}}m \approx 0.53m. \tag{5.92}$$

	Matrix-vector products	Back orthogonalization
Update	$2lk_pk_u$	mk_u^2
Predict	$2l(k_p+k_u)k_p$	lk_p^2

Table 5.2. Matrix Multiplications Required for Krylov Subspace Kalman Filtering

If $k_u \ll k_p$, a best case scenario, then (5.90) reduces to

$$k_p \leq \sqrt{\frac{2}{3}}m \approx 0.82m. \tag{5.93}$$

In other words, one expects gains if one can obtain a good approximate estimator with a number of iterations that is significantly less than the number of measurements. Moreover, one may be able to exploit additional structure in the problem to obtain additional gains, as illustrated by the oceanographic example in Section 5.5.2.

The computation of smoothed estimates and error covariances requires an additional sweep through the data after filtering. The conclusions of the complexity analysis in Section 5.4.3 is that this additional sweep tends to dominate the computational workload. The computational gain resulting from using the Krylov subspace method for the sweep is

$$\frac{3l^3/2 + 2l^2m}{(5k_um + k_ul + 2k_pl)k_s + k_s^2l}. \tag{5.94}$$

Since a typical remote sensing problem has $k_u, k_p, k_s < m < l$, the gain is generally lower than for filtering. Gains for specific examples are worked out in Section 5.5.

■ 5.4.2 Analysis of the Filter

Consider a step of the Krylov subspace method for Kalman filtering after the initial time with no preconditioning. The two major contributions to the computational workload are matrix-vector products and back orthogonalization. The cubic terms in the number of multiplications needed for matrix-vector products and back orthogonalization for both the current update and the next predict are listed in Table 5.2. The total number of operations for a combined update and predict step is

$$mk_u^2 + lk_p^2 + 2lk_p(k_p + k_u) + 2k_pk_ul. \tag{5.95}$$

For comparison, the work in standard implementations of the Kalman filter is dominated by matrix-matrix products and matrix inversions. Recall, however, that there are efficient routines for matrix-vector products involving the Λ_n, Λ_w, and C matrices. Then, the cubic terms in the number of multiplications needed for a standard implementation can be broken down as in Table 5.3, following [24]. This leads to a total of

$$\frac{m^3}{6} + 2m^2l \tag{5.96}$$

multiplications. The ratio of (5.96) and (5.95) leads to (5.89).

	Expression	Work
$C\Lambda$	$C(t)\cdot\Lambda_e(t\|t-1)$	-
Λ_{pr}	$[C\Lambda]\cdot(C(t))^T+\Lambda_n(t)$	-
K	$A(t)\cdot[C\Lambda]^T\cdot\Lambda_{pr}^{-1}$	$m^3/6+m^2l$
$\Lambda_e(t+1\|t)$	$(A(t)\cdot\Lambda_e(t\|t-1)-K\cdot C\Lambda)\cdot(A(t))^T+\Lambda_w$	lm^2
$\hat{x}(t+1\|t)$	$A(t)\cdot(\hat{x}(t\|t-1)-K_U(C(t)\cdot\hat{x}(t\|t-1)-y(t)))$	-

Table 5.3. Number of Multiplications Required for a Standard Implementation of Kalman Filtering

5.4.3 Analysis of the Smoother

Operation	Work
$F_{k_u(t-1)}(t-1)$-multiply	$k_u(m+l)k_s$
$V_{k_s(t)}(t\|T)$-multiply	k_umk_s
$\Lambda^{-1}_{\nu,k_u(t-1)}(t-1)$-multiply	k_umk_s
$\Lambda_{e,k_p(t)}(t\|t-1)$-multiply	$2k_pl$
Back orthogonalization	k_s^2l

Table 5.4. Matrix Multiplications Required for Krylov Subspace Smoothing

The workload of the Krylov subspace method for smoothing is dominated by the matrix-vector products in (5.85) and (5.86) as well as back orthogonalization. The cubic terms in the numbers of multiplications required for these operations are listed in Table 5.4. The total is

$$(5k_um+k_ul+2k_pl)k_s+k_s^2l. \tag{5.97}$$

Operation	Work
$F^T(t-1)V(t,T)$	l^2m
$(F^T(t-1)V(t,T))F(t-1)$	l^2m
$C^T(t-1)\Lambda_\nu^{-1}(t-1)$	-
$(C^T(t-1)\Lambda_\nu^{-1}(t-1))C(t-1)$	-
$\Lambda_e(t\|t-1)V(t,T)$	l^3
$(\Lambda_e(t\|t-1)V(t,T))\Lambda_e(t\|t-1)$	$l^3/2$

Table 5.5. Number of Multiplications Required for a Standard Implementation of a Modified Bryson-Frazier Smoother

The workload for a standard implementation of the modified Bryson-Frazier smoother is dominated by the matrix-matrix products in (5.60) and (5.61). The cubic terms in the numbers of multiplications required for these operations are listed in

Table 5.5. Note that products involving the matrix

$$F(t) = A(t)(I - \Lambda_e(t|t-1)C^T(t)\Lambda_\nu^{-1}(t-1)C(t)) \tag{5.98}$$

are relatively efficient since $A(t)$- and $C(t)$-vector products can be done efficiently and part of the product in (5.98) is already formed as part of the filtering step (see Table 5.3). The total of the cubic terms in Table 5.5 is

$$3l^3/2 + 2l^2m. \tag{5.99}$$

The ratio of (5.99) and (5.97) leads to (5.94).

■ 5.5 Numerical Examples

In this section, the performance of the Krylov subspace method for Kalman filtering is characterized with two sets of numerical examples. The state being estimated propagates differently in time for each set of examples. In the first, the state propagates according to a damped heat equation, and, in the second, a Rossby wave equation. The dynamics of a damped heat equation are chosen because such dynamics have been used previously to test approximate Kalman filtering problems [36–39]. The Rossby wave equation dynamics have also been used previously to test Kalman filter methodology and are more relevant to oceanographic remote sensing [31]. Thus, these examples are interesting of themselves, and the differences between them provide an indication of the generality of the proposed method for filtering.

■ 5.5.1 Damped Heat Equation Dynamics

The dynamics of the first set of two examples obey a stochastic damped heat equation on a ring. Specifically, the state x obeys a spatially and temporally discretized version of

$$x_t = -\nabla^2 x - \alpha x + w, \tag{5.100}$$

where w is the driving noise. The discretization leads to dynamics of the form

$$x(t+1) = Ax(t) + w(t) \tag{5.101}$$

where

$$A = (1 - \alpha\delta t)I + \delta t L \tag{5.102}$$

for a time step constant δt and approximation to the Laplacian, L. The form of L used here is a simple three-point approximation that uses the kernel

$$\begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \tag{5.103}$$

to compute an approximation to the second derivative. For each of the two examples in this subsection, the damping coefficient $\alpha = 0.2$, the time step $\delta t = 0.1$, and the state dimension is 1024. The statistics of the driving noise $w(t)$ and initial state $x(0)$ differ for each example.

The first example is a simple one that illustrates some typical behaviors of the Krylov subspace method. In this example, the driving noise is stationary on the discretized ring, and the statistics do not change in time. The power spectral density $S_{ww}(\omega)$ of the noise decays exponentially with rate 0.3, *i.e.*

$$S_{ww}(\omega) \propto (0.3)^{|\omega|}. \tag{5.104}$$

The variance of the noise is set to 0.02. The initial state $x(0)$ has the same statistics as $w(t)$. Measurements are taken pointwise everywhere in space and time and are embedded in additive white noise of intensity 640.[2] The update iterations were stopped when $\tau_{k,10^{-6}} < 10^{-6}$ for $K_{\text{win}} = 8$, where $\tau_{k,10^{-6}}$ and K_{win} are defined in (3.19). The stopping criterion of Algorithm 4.2.1 was used for the prediction steps with a threshold $\chi = 10^{-4}$.

The results for this example are displayed in Figures 5.1 and 5.2. In Figure 5.1, the outputs of the Krylov subspace method are compared against exact solutions computed using FFTs. Relative mean-squared error in the filtered and smoothed estimates and error variances are plotted as a function of time step. The errors increase sharply after the first step and then level off below 1%. The increase is largely due to the fact that the dominant source of approximation error is from the prediction step. That the errors level off is a consequence of the time-invariance of the parameters in the problem and the state dynamics being stable in the sense that $\|A\| < 1$.

For each of the update, predict, and smoothing steps, the numbers of iterations required to achieve these levels of approximation are plotted in Figure 5.2. These numbers tend to remain constant over time. There is, however, a certain degree of oscillatory behavior, reflecting the interplay between the state dynamics and the predict and update approximation algorithms. One can use the median numbers of iterations and the results of Section 5.4 to calculate approximately the amount of speedup offered by the Krylov subspace method as compared to a straightforward implementation of

[2] Note that this corresponds to an intensity of 5/8 if the problem were continuous spatially. Specifically, the measurement noise n of the continuous problem would have a covariance function

$$\text{Cov}(n_u(t), n_v(t)) = \frac{5}{8}\delta(u-v). \tag{5.105}$$

Since $\delta(u-v)$ is approximated by

$$\delta(u-v) \approx l \qquad |u-v| \leq \frac{l}{2} \tag{5.106}$$

where l is the number of points in the spatial discretization, the discretized measurement noise has variance $l5/8$. For our example, $l = 1024$; so the variance of the measurement noise for the discretization is 640. This value of the noise variance was chosen so that the error variances would be at intermediate values, neither too close to 0 nor too close to the *a priori* variance of the state, $x(t)$.

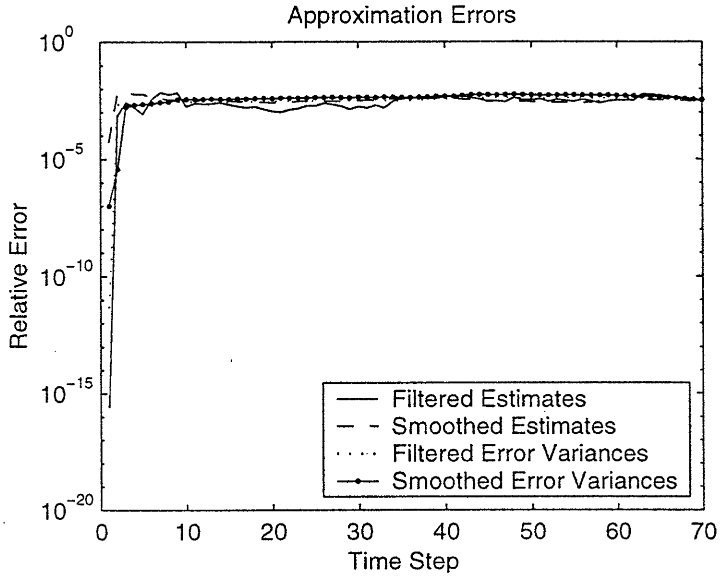


Figure 5.1. The curves plot the approximation errors for the Krylov subspace method applied to the damped heat equation problem with driving noise that has exponentially decaying power spectral density. Each curve plots the mean-squared differences between results computed using the Krylov subspace method and an exact FFT method. The four sets of results are for the filtered and smoothed estimates and error variances.

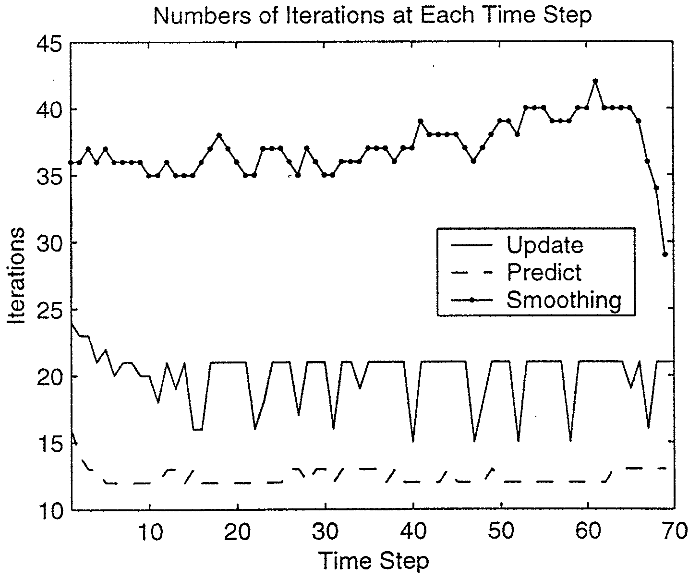


Figure 5.2. The curves plot the numbers of iterations needed to meet the stopping criteria for the Krylov subspace method applied to the damped heat equation problem with driving noise that has exponentially decaying power spectral density.

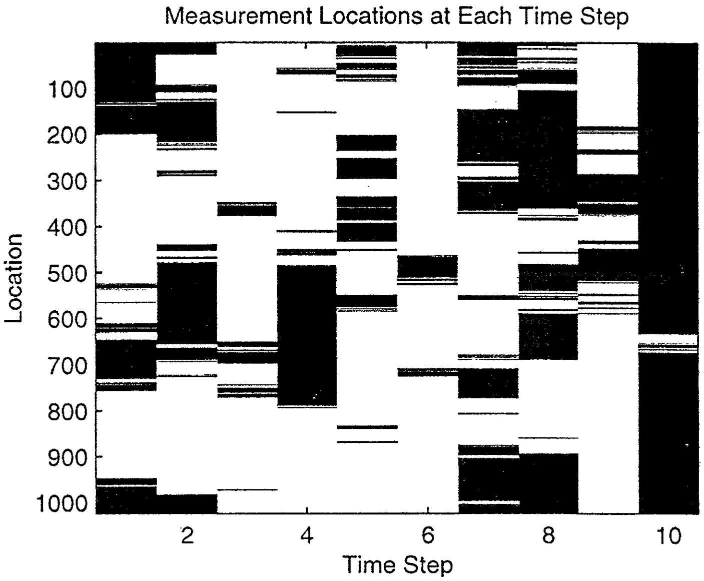


Figure 5.3. The image shows the measurement locations for the damped heat equation problem with driving noise that has a polynomially decaying power spectral density. Measurements are white.

the filter and smoother. The median numbers of iterations for the update, predict, and smoothing steps are 21, 12, and 37, respectively. The resulting speedups are 1200 for filtering and 680 for smoothing. Thus, the results for this first example indicate that the errors and iteration counts of the Krylov subspace method behave reasonably well and that the method can be substantially faster than a standard Kalman filter implementation.

The second example demonstrates some of the issues in using preconditioners to accelerate convergence of the Krylov subspace method. As for the first example, the driving noise in this second example is stationary on the discretized ring, and the statistics do not change in time. The power spectral density of the driving noise $S_{ww}(\omega)$ falls off only polynomially. Specifically,

$$S_{ww}(\omega) \propto \frac{1}{(|\omega| + c)^2}, \tag{5.107}$$

where $c = 1/1024$. The variance of the noise is set to 1. The statistics of the initial state $x(0)$ are the same as those of the driving noise. Measurements consist of scattered point measurements embedded in additive white noise. The intensity of the noise is 640. Figure 5.3 indicates the locations of the measurements at each point in time. The locations were chosen randomly in such a manner that they tend to cluster spatially, as often happens in remote sensing problems. The exact numbers of measurements are plotted in Figure 5.4. The number of Krylov subspace iterations within each update step is fixed at 20, and the number of predict iterations is fixed at 100.

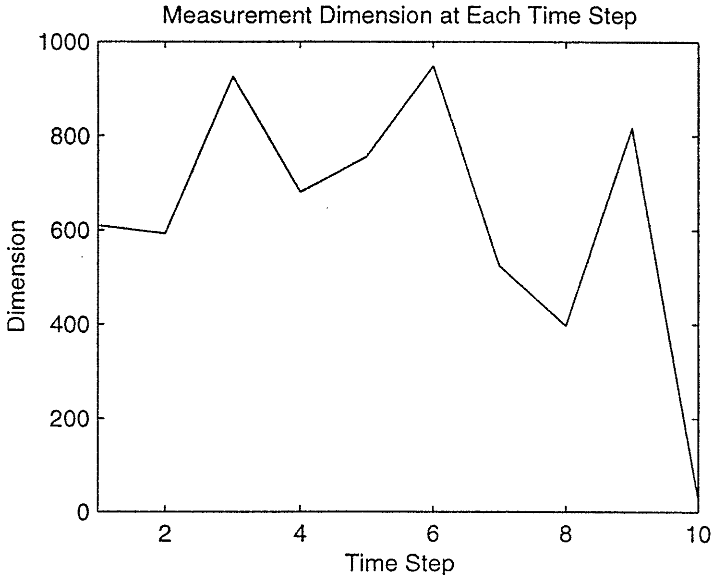


Figure 5.4. The image shows the measurement dimensions for the damped heat equation problem with driving noise that has a polynomially decaying power spectral density.

A simple update preconditioner is considered for this problem. At each time step, it has the form

$$C(t)\Lambda_{pu}C^T(t) + I \tag{5.108}$$

where $C(t)$ is the measurement matrix, *i.e.* the matrix that selects elements of a vector at measurement locations; and Λ_{pu} is a stationary covariance matrix corresponding to the power spectral density $10^8(0.5)^{|\omega|}$. As discussed previously in Section 3.4.1, the first term in (5.108) is trying to induce a geometric separation in the eigenvalues of the prediction error covariance (which acts as the prior in the update step). The shift by the identity in (5.108) ensures that no modes of the prediction error covariance get annihilated by the preconditioner.

The results in Figure 5.5 illustrate the effectiveness of this preconditioner. Results for both a preconditioned and non-preconditioned Krylov subspace method for filtering are compared to an exact calculation done with direct methods. The relative mean-squared errors of the error variances are plotted in Figure 5.5. Note that the preconditioner reduces the error by about half an order of magnitude at the initial time step but leaves the errors essentially unchanged at every subsequent time step. Thus, the preconditioner is ineffective after the first time step. There are two possible reasons for this behavior. The first is that the errors are dominated by the contributions from the predict steps. The other possible reason for the ineffectiveness of the preconditioner is that the matrix Λ_{pu} is Toeplitz. The preconditioner can separate out modes at the initial time step since the initial covariance is stationary, but the preconditioner is in-

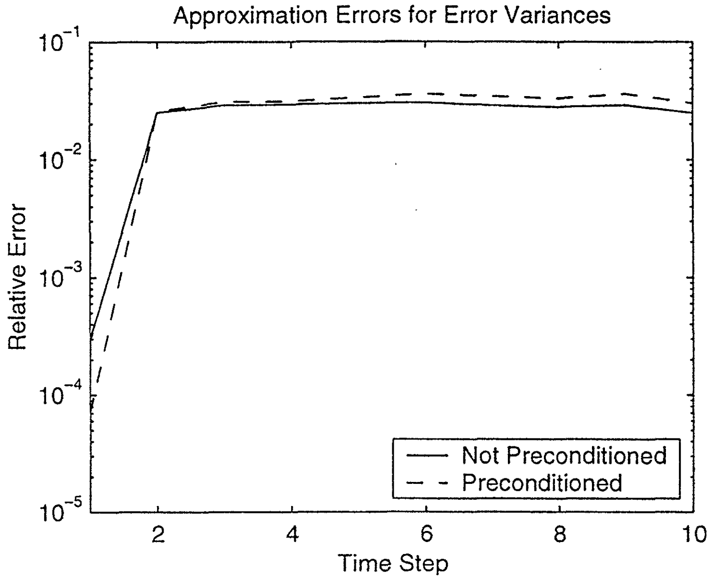


Figure 5.5. The curves plot the approximation errors for both a preconditioned and non-preconditioned Krylov subspace method applied to the damped heat equation problem with driving noise that has polynomially decaying power spectral density. Each curve plots the mean-squared differences between error variances computed using the Krylov subspace method and an exact solution computed using direct methods.

effective at subsequent steps because the prediction error covariance is non-stationary and changing from time step to time step because of the measurement structure. Although this single example is not conclusive by itself, the results suggest that update preconditioning may only be effective if the prediction errors are smaller than the gain provided by the preconditioner and the update preconditioner is adaptively matched to the measurement structure. The development of such an adaptive preconditioner is left for future work.

■ 5.5.2 Rossby Wave Dynamics

This section presents the results of applying the Krylov subspace method for Kalman filtering to the estimation of sea surface anomaly from real data gathered by the T/P altimeter. The region of interest is 25.125° – 28.875°N and 212.5° – 220°E, which lies in the Pacific, west of North America. The altimetric data over this region are displayed in Figure 5.6. Since we are interested in comparing our algorithm's results with those of direct methods, we chose the size of the region to be relatively small.

Each data point in this region is a measurement of sea surface anomaly: the height with a mean and ocean tide effects removed [27,50]. Data has been grouped into 10 day repeat cycles. Although the satellite repeats its orbit approximately every 10 days, the measurement locations from each 10 day period are a little different for various reasons.

Note that the coverage of the data is fairly sparse.

The goal of the algorithm is to assimilate the data into a model to provide an interpolated map of sea surface anomaly. The model comes from a variety of statistical and physical considerations. A variety of models and assimilation methods have been proposed for processing satellite altimetry [14, 28, 31, 73]. The model we used and describe subsequently is very similar to that used in [31]. However, the details of the model we used were chosen more to illustrate the performance of the algorithm than for their faithfulness to the data. Although the model has realistic components, the emphasis is on the algorithmic performance and not on the modeling.

The Statistical Model

We model the ocean using a linearized Rossby wave equation with stochastic forcing. This has the form

$$\frac{\partial}{\partial t}\left\{\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\frac{\rho}{S}\frac{\partial\psi}{\partial z}\right)\right\}+\beta\frac{\partial\psi}{\partial x}=w \tag{5.109}$$

where ψ is the sea surface anomaly, w is the stochastic forcing, ρ is the density, S is a parameter varying as a function of depth (z) that measures the effect of stratification in the ocean, and β is a constant that measures the effect of the Coriolis force in the latitude range of interest. A detailed discussion of the homogeneous equation's solution and properties can be found in [67, Section 6.12]. A brief summary is given here.

There are a countably infinite number of solutions to the homogeneous equation. Solutions can be written as a wave in the x-y plane with frequency σ_n whose amplitude Φ_n varies with depth:

$$\operatorname{Re} e^{ik_1x+k_2y-\sigma_n t}\Phi_n(z) \qquad n=0,1,2,\ldots. \tag{5.110}$$

The amplitude $\Phi_n(z)$ will depend on $S(z)$. The exact dependence of $\Phi_n(z)$ on $S(z)$ is not important here because we are only interested in solutions evaluated at the surface ($z=0$), which can be written as waves in the x-y plane:

$$\operatorname{Re} e^{ik_1x+k_2y-\sigma_n t} \qquad n=0,1,2,\ldots. \tag{5.111}$$

The frequency of the wave, σ_n, varies as a function of wavenumbers k_1 and k_2 according to the dispersion relation

$$\sigma_n=-\frac{\beta k_1}{k_1^2+k_2^2+1/R_n^2} \qquad n=0,1,2,\ldots \tag{5.112}$$

where R_n is a constant that depends on $S(z)$ and is termed the Rossby radius of deformation of the nth vertical mode. The exact dependence of R_n on $S(z)$ is complicated and not directly of interest here since only the values of R_n are needed to form solutions to (5.109), and estimates of the values can be found in the literature for certain n [13].

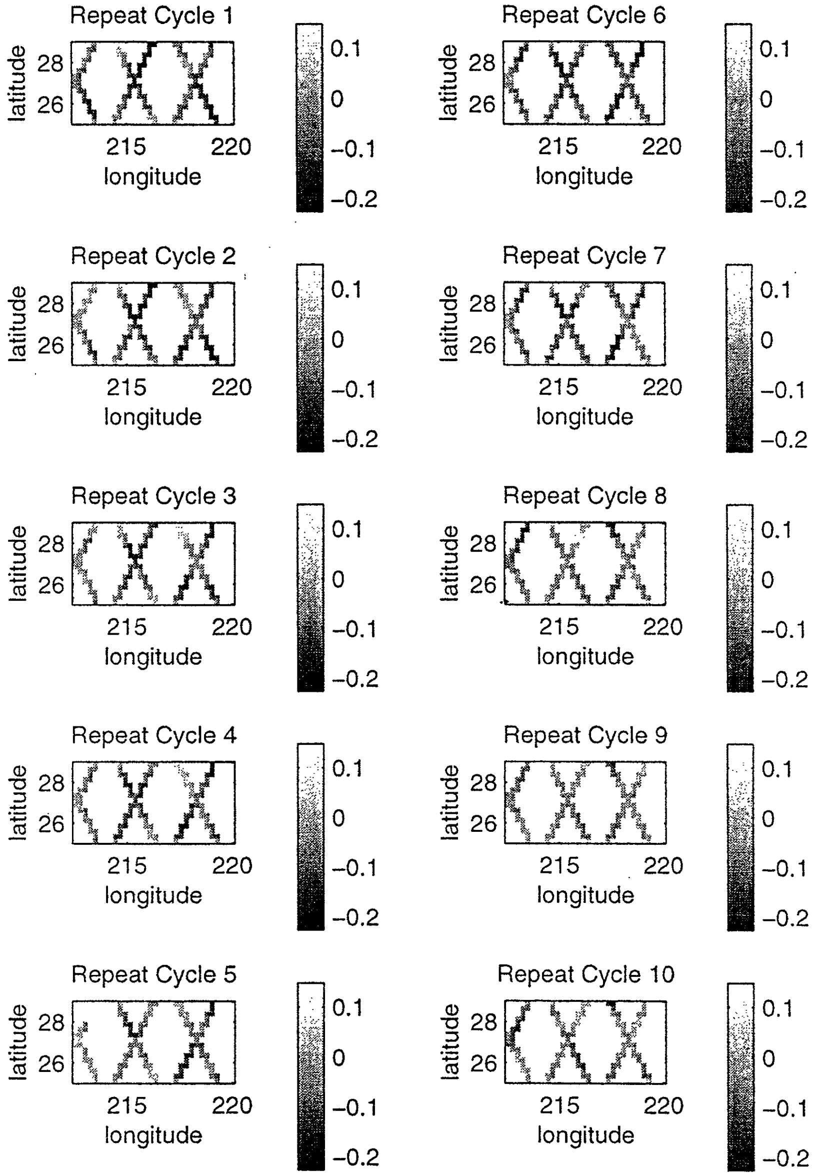


Figure 5.6. Each image shows the T/P sea surface anomaly data over a 10 day repeat cycle

In particular, the zeroth vertical mode always has an infinite radius of deformation, independent of $S(z)$, and is known as the barotropic mode. This mode is also the solution to the Rossby wave equation assuming that the pressure is linear in depth. The other modes are called baroclinic. In the region of interest, the radius of deformation for the first baroclinic mode, R_1, has been calculated from data in [13] to be approximately $35km$.

We model the ocean as a sum of two sets of plane waves, one with frequency σ_0 and the other with σ_1. Note that only the plane waves with frequency σ_0 are included in the model in [31]. The addition of plane waves with frequency σ_1 is one of the principal differences between the model used here and in [31]. In what follows, $\psi_0(x, y, t)$ denotes the contribution to the ocean surface anomaly at (x, y) and time t from those waves with frequency σ_0, and $\psi_1(x, y, t)$, with frequency σ_1. The anomaly at (x, y) and time t is $\psi_0(x, y, t) + \psi_1(x, y, t)$. The functions $\psi_0(x, y, t)$ and $\psi_1(x, y, t)$ are often written as $\psi_0(t)$ and $\psi_1(t)$, in the subsequent discussion.

Since we are interested in the ocean state at discrete time instants, we arrive at the following recursion

$$\begin{aligned} \psi_0(t + \delta t) &= A_0 \psi_0(t) + w_0(t) \\ \psi_1(t + \delta t) &= A_1 \psi_1(t) + w_1(t) \end{aligned} \tag{5.113}$$

where δt is the difference between times of interest. The operators A_0 and A_1 are all-pass filters that perform the appropriate phase shifts:

$$\frac{-\beta k_1 \delta t}{k_1^2 + k_2^2} \tag{5.114}$$

for A_0 and

$$\frac{-\beta k_1 \delta t}{k_1^2 + k_2^2 + 1/R_1^2} \tag{5.115}$$

for A_1 where k_1 and k_2 range over all wavenumbers in the x-y plane. The random vectors $w_0(t)$ and $w_1(t)$ are the stochastic forcing terms of each set of plane waves.

The statistics of the stochastic forcing terms $w_0(t)$ and $w_1(t)$, as well as the initial conditions $\psi_0(0)$ and $\psi_1(0)$, are chosen to match ocean statistics. We assume that the sea surface anomaly is stationary over the region of interest. Then, if we choose w_0, w_1, $\psi_0(0)$, and $\psi_1(0)$ to be stationary, both $\psi_0(t)$ and $\psi_1(t)$ will also be stationary. Moreover, if we set the power spectral densities (PSDs) of $\psi_0(0)$, $\psi_1(0)$, w_0, and w_1 all proportional to a common PSD $S_{\psi\psi}(\omega)$, then $\psi_0(t) + \psi_1(t)$ will also have a PSD proportional to $S_{\psi\psi}(\omega)$ because of the all-pass nature of A_0 and A_1. Since we would like $\psi_0(t) + \psi_1(t)$ to have a statistical structure similar to the ocean, we choose $S_{\psi\psi}(\omega)$ to have the same shape as the ocean PSD. Stammer has conducted studies of one-dimensional ocean spectra along satellite tracks [72]. He determined that the spectra did not vary significantly from track to track or region to region, and that the spectra obeyed different power laws over different wavenumber intervals approximately as outlined in Table 5.6. From the spectra not varying significantly from track to track, he concluded that the two-dimensional spectra are reasonably isotropic. Unfortunately these spectra are one-dimensional spectra, which are line integrals of the full two-dimensional spectra. However, noting that the asymptotic decay of a two-dimensional power law decay is also a power law of one less degree,

$$\int_{\infty}^{\infty} \frac{1}{(\sqrt{\omega_1^2 + \omega_2^2})^n} d\omega_1 \propto \frac{1}{\omega_2^{n-1}}, \tag{5.116}$$

Wavenumber (cycles/ km)	Power Law
$2.5 \times 10^{-5} - 2.4 \times 10^{-3}$	0.5
$2.4 \times 10^{-3} - 1.0 \times 10^{-2}$	3.0
$1.0 \times 10^{-2} - 8.3 \times 10^{-2}$	4.0

Table 5.6. Power Laws of One-dimensional Ocean Spectra

Wavenumber (cycles/ km)	Power Law
$2.5 \times 10^{-5} - 2.4 \times 10^{-3}$	1.5
$2.4 \times 10^{-3} - 1.0 \times 10^{-2}$	4.0
$1.0 \times 10^{-2} - 8.3 \times 10^{-2}$	5.0

Table 5.7. Power Laws of Two-dimensional Ocean Spectra

one can assume that the power laws in two dimensions are one order higher, as listed in Table 5.7. This is the two-dimensional power spectrum we chose for each of $\psi_0(0)$, $\psi_1(0)$, $w_0(t)$, and $w_1(t)$. We then normalize the overall variance so that the variance of each of $\psi_0(0)$ and $\psi_1(0)$ is half that of a steady-state ocean, $5m^2$, and the variance of each of $w_0(t)$ and $w_1(t)$ is half that of the 10-day variability in the ocean, $10^{-3}m^2$.

We reduce (5.113) to a finite-dimensional state-space model for implementation on a computer by propagating only some of the plane waves corresponding to each depth mode. Specifically, we propagate only plane waves whose wavenumbers are harmonics of a region which is double the size in each dimension of the region of interest. We propagate more than just the harmonics of the region of interest because doing so would imply a periodicity in the model. A more in-depth motivation and discussion of the implications of the approximation on the model accuracy follow the discretization details, which are provided next.

The region of interest in the discretized model is gridded to 16 × 8 pixels. The discretized states, $\psi_0'(t)$ and $\psi_1'(t)$, consist of samples of the barotropic and first baroclinic modes, respectively, over a region twice the size of the region of interest, gridded to 32 × 16 pixels. The recursion for the discretized model is given by

$$\psi_0'(t+\delta t) = A_0'\psi_0'(t) + w_0'(t) \tag{5.117}$$

$$\psi_1'(t+\delta t) = A_1'\psi_1'(t) + w_1'(t). \tag{5.118}$$

The operators A_0' and A_1' act on $\psi_0'(t)$ and $\psi_1'(t)$, by performing the same phase shifts as A_0 and A_1, respectively, on those plane waves present in the discretized model, *i.e.* the harmonics of the 32×16 grid. The covariances of the process noises, Λ_0' and Λ_1', are equal, as for the non-discretized problem. The discretized processes are chosen to be stationary on a torus so that the covariances are diagonalized by the DFT. The power spectral densities have the decays given in Table 5.7. This does not set the DC terms of the power spectral densities, however. To set the DC terms, we note that for the continuous spectrum in Table 5.7, 80% of the power is in the lowest wavenumber band.

Thus, the DC terms of the discrete spectra are set so that 80% of the power is contained in the DC terms plus those wavenumbers in the band of the slowest decay. This has the effect of modeling the real ocean variability of the very lowest wavenumbers by an aggregate DC variability.

The motivation for incorporating extract harmonics into the model and the implications on its accuracy are as follows. Consider the following state-space model of the barotropic mode that differ slightly from the previous model. Let $\psi_0''(t)$, like $\psi_0'(t)$ defined previously, be a state vector consisted of samples on a regular grid of the barotropic mode in a region twice the size of the region of interest. A discrete-time state-space model for $\psi_0''(t)$ is given by

$$\psi_0''(t+\delta t) = A_0''\psi_0''(t) + w_0''(t). \tag{5.119}$$

Here, A_0'' is the same as A_0' in the previously discussed model. In other words, A_0'' acts on $\psi_0''(t)$ by performing the same phase shifts as A_0 on those plane waves in the discretized model. However, the covariance Λ_0'' of the driving noise $w_0''(t)$ differs from Λ_0'. Specifically, suppose that the Toeplitz covariance matrix Λ_0 of samples of the driving noise $w_0(t)$ in (5.113) over the region of interest can be embedded into a positive definite circulant matrix of twice the dimension (note that this is similar to zero padding for performing convolutions with an FFT, and is often possible [23]). Then, Λ_0'' is given by this embedding matrix. The resulting state-space model has some nice properties because of the all-pass nature of A_0''. In particular, the covariance of the portion of $\psi_0''(t)$ corresponding to samples over the region of interest is exactly the same as the covariance of the samples in this region as specified by (5.113). This guarantees that one has not introduced any periodicity into the covariances over the region of interest. However, the cross-covariances between states $\psi_0''(t)$ and $\psi_0''(t')$ for $t \neq t'$ will not be exactly the same as those specified by (5.113). This is because the full infinite-dimensional state represents the ocean as an infinite plane, and non-zero anomalies far away from the region of interest may eventually propagate, according to the Rossby wave dynamics, into the region of interest. Yet, the cross-covariances not being the same for the infinite and finite models should not cause significant differences in the estimates provided by the two models. This is because data are confined to the region of interest so non-zero estimates of anomaly will tend to be confined to lie within the area captured by the finite state ψ_0''. This approach to forming a model by keeping a region twice as large in each dimension as the region of interest motivates our state-space model consisting of modes which are harmonics of the larger region. As noted previously, however, the actual process noise covariance Λ_0' used in the numerical work differs from the covariance Λ_0'' in the motivating discussion here. This is because Λ_0 is not known exactly. Only portions of the continuous power spectral densities are given. Thus, Λ_0' can be computed while Λ_0'', even if it were to exist, cannot be computed exactly.

The measurement noise model we use is motivated by that used in [31] to analyze GEOSAT data. In particular, we assume that the noise is uncorrelated from track to track. As noted in [31], this assumption is a worst-case scenario made more because it implies a Markov model than because it is realistic. Along track, we assume the presence of both long-range and short-range correlated noise components. The long-range components are meant to model satellite tracking errors that can be fairly large. As in [31], the noise is assumed stationary with a correlation function of the form:

$$n_0 e^{-|\tau|/l}. \tag{5.120}$$

We set $n_0 = 2.2m^2$ and $l = 3.1 \times 10^3 km$ to match the variance and variance of the derivatives of the noise model in [28].[3] Since each measurement is spaced approximately $7km$ apart, this yields a significantly long-range correlated noise. The short-range component we model as white noise with intensity $25cm^2$. This is set a little higher than the $9cm^2$ used for static estimation in [28] to allow for modeling errors in the dynamics.

The Numerical Results

We apply the general Krylov subspace method described in the previous sections to the problem of assimilating data into this model with only one minor modification to the method. Specifically, we use the Krylov subspace realization algorithm to compute an approximation only to the error reduction term of the prediction error covariance and not the full error covariance, which adds in the prior covariance of the state. That is, at step t, the updated error covariance is approximated by

$$\Lambda_x(t) - \sum_{i=1}^{k_p(t)} f_i(t) f_i^T(t) \tag{5.121}$$

where $\Lambda_x(t)$ is the exact covariance of $x(t)$ and $f_i(t)$ are described subsequently. The prediction error covariance at time $t+1$ is then approximated by

$$\Lambda_x(t+1) - A(t) \left(\sum_{i=1}^{k_p(t)} f_i(t) f_i^T(t) \right) A^T(t). \tag{5.122}$$

this approximation is used to perform matrix-vector multiplies in the Krylov subspace estimation algorithm at the next update step. This yields and update error covariance of

$$\Lambda_x(t+1) - A(t) \left(\sum_{i=1}^{k_p(t)} f_i(t) f_i^T(t) \right) A^T(t) - \sum_{i=1}^{k_u(t+1)} r_i(t+1) r_i^T(t+1) \tag{5.123}$$

where $r_i(t+1)$ are the filtered back projected search directions generated by the Krylov subspace estimation algorithm. The Krylov subspace realization algorithm is then run

[3] The noise model in [28] is linear. Specifically, the noise along a track is modeled as $a + bD$ where a and b are random variables and D is the distance along track.

to yield a low-rank approximation to

$$A(t)\left(\sum_{i=1}^{k_p(t)} f_i(t)f_i^T(t)\right)A^T(t)+\sum_{i=1}^{k_u(t+1)} r_i(t+1)r_i^T(t+1). \tag{5.124}$$

Then, the update error covariance at step $t+1$ is approximated by

$$\Lambda_x(t+1)-\sum_{i=1}^{k_p(t+1)} f_i(t+1)f_i^T(t+1) \tag{5.125}$$

where $f_i(t+1)$ are the filtered back projected search directions generated by the Krylov subspace realization algorithm. Note that we know the covariance of the state, $\Lambda_x(t)$, at every time in part because the state at every time step remains stationary. By not trying to approximate the whole prediction error covariance, we avoid some of the potential numerical problems caused by not computing enough terms in the approximation, as discussed in Section 5.2.1.

Other than this minor algorithmic change, the algorithm implementation is straightforward. The matrix-vector products are implemented with FFTs. Since the measurement noise is not white, we use a whitening preconditioner at the update steps, as described in Section 3.4.1. The preconditioner consists of circulant approximations to the inverse of the along-track measurement noise covariances. Thus, the preconditioner is also implemented with FFTs.

For comparison purposes, we assimilated the data into the model exactly using direct methods in MATLAB. The interpolated grid for the region of interest is 16 × 8, yielding a state dimension of 1024. The number of measurements in this region is 373 on every repeat cycle except for two, on which the number is 341 and 369. The exact smoothed estimates of sea surface anomaly (the estimates of the sum of the barotropic and first baroclinic modes) are displayed in Figure 5.7. Figure 5.8 displays the error variances of the smoothed estimates. The variances are lower along tracks. They also decrease over time in the East, presumably because the group delay for many of the modes is eastward traveling.

The exact results provide a reference against which the results of the Krylov subspace method can be compared. The Krylov subspace method was applied to exactly the same problem as the reference. The update iterations were stopped when $\tau_{k,10^{-2}} < 10^{-2}$ for $K_{\text{win}} = 8$, where $\tau_{k,10^{-2}}$ and K_{win} are defined in (3.19). Between 76 and 100 iterations were required to meet this requirement; the median number of iterations was 95.5. The stopping criterion of Algorithm 4.2.1 was used for the prediction steps with a threshold $\chi = 2 \times 10^{-2}$. Between 103 and 197 iterations were required to meet this; the median number of iterations was 163. The stopping criterion for the smoothing steps were the same as for the update steps. Between 108 and 269 iterations were required; the median number was 196. Figure 5.9 shows the relative mean squared error between the exact results and the Krylov subspace method results for the filtered

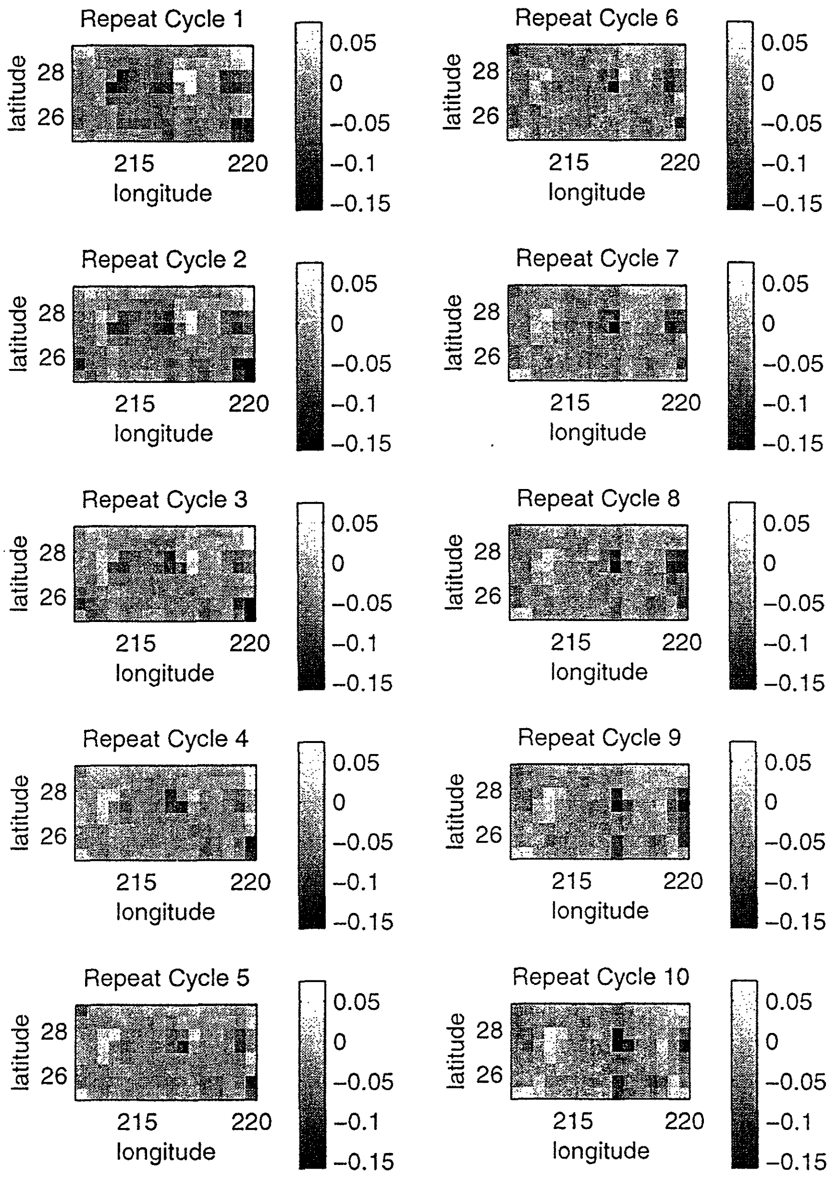


Figure 5.7. Each image shows the smoothed estimates of sea surface anomaly computed using direct methods in MATLAB

and smoothed full state estimates and error variances. The approximation errors in the estimates are no bigger than 22%, and error variances, no bigger than 1%. Although a computational error of 22% in the estimates may seem large, it is not necessarily so since there are many factors that contribute to the real quantity of interest, the error between the estimates and actual sea surface anomaly. Modeling error, in particular, is

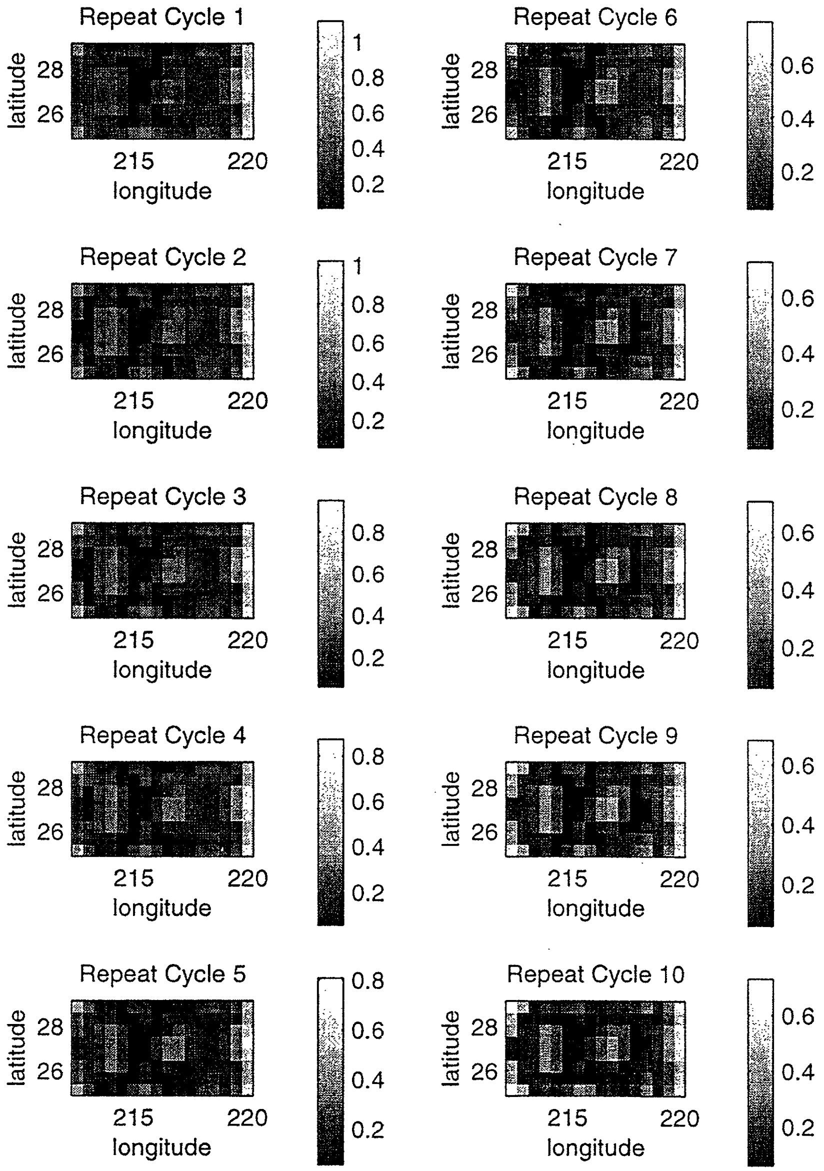


Figure 5.8. Each image shows the error variances of the estimates of sea surface anomaly computed using direct methods in MATLAB

a major factor. Moreover, note that the true error variances for the given the model, displayed in Figure 5.8, can exceed 1, giving a sense of the variability one expects in the estimates. Now, note that the filtered approximation errors in the estimates and error variances increase, at first, but then they level out. One expects this since the system is strongly stable, as described in Section 5.2.2. The smoothed approximation

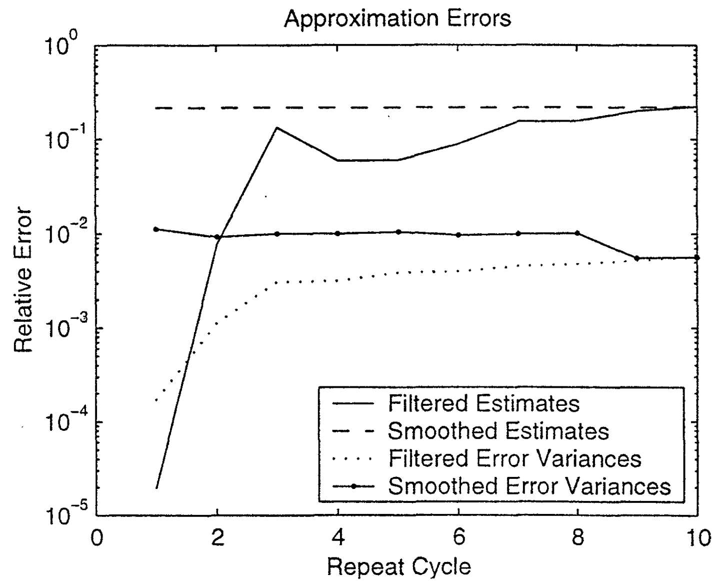


Figure 5.9. The solid line plots total relative computational errors for the filtered estimates; dashed line, smoothed estimates; dotted line, filtered error variances; and dashed-dot line, smoothed error variances.

	Matrix-vector products	Back orthogonalization
Update	$2lk_pk_u$	mk_u^2
Predict	$4lk_uk_p$	lk_p^2

Table 5.8. Matrix Multiplies Required for Krylov Subspace Kalman Filtering As Applied to the Oceanographic Data Assimilation Problem

errors do not increase significantly on the backward sweep, and thus, remain almost constant across repeat cycles. Note that the approximation errors for computing the error variances remain lower than for the estimates. Since the matrices in the estimation problem do not change to a large extent from time step to time step, one expects that, given Conjecture 5.2.1, the error covariance will tend towards a steady-state value. The dynamics of the filter will, thus, damp out perturbations in the error covariance by driving the error covariance to this steady-state. Hence, one expects computational errors in the error variances to be lower than for the estimates since the dynamics of the estimates are only strongly stable so that perturbations in their values may accumulate slowly over time.

One can obtain a rough estimate of computational gains using the analysis of Section 5.4, as follows. Note that the numbers of multiplications needed for filtering listed in Table 5.2 needs to be modified since the algorithm we used for this oceanographic data assimilation problem differs slightly from the general algorithm. Table 5.8 lists the computational work for our modified Krylov subspace method for filtering. For iteration and problem size, we'll use the median values: $k_u = 95.5$, $k_p = 163$, $l = 1024$, and $m = 3736$. Then, the approximate multiplication count per time step for the Krylov subspace method is $2lk_pk_u + 4lk_pk_u + mk_u^2 + lk_p^2 = 1.26 \times 10^8$. An exact implementation requires approximately $m^3/6 + 2m^2l = 2.94 \times 10^8$ multiplications. Thus, the speed-up is approximately a factor of 2.3. The additional number of multiplications required for the Krylov subspace method for smoothing is $(5k_um + k_ul + 2k_pl)k_s + k_s^2 l = 1.59 \times 10^8$. The exact implementation requires approximately $3l^3/2 + 2l^2m = 2.39 \times 10^9$ additional multiplies. The final speedup is 9.4 for smoothing. Although this order of magnitude speedup may not be enough to, on today's computers, feasibly assimilate global data sets into models of the moderate complexity used here, the speedup significantly stretches the capability of modern computers to solve small to moderately sized data assimilation problems.

■ 5.6 Summary

This chapter presents an approach to implementing a Kalman filter and modified Bryson-Frazier smoother insert after smoothing. using Krylov subspace methods. This approach is applicable to a wide variety of space-time estimation problems, including many from remote sensing. Our approach to assimilating real oceanographic data into a model with many realistic components yielded a speedup, particularly for smoothing.

Chapter 6

Stability of Kalman Filters for Space-Time Estimation

This chapter contains a proof of the stability of the Kalman filter for a class of space-time estimation problems. Specifically, a proof of Theorem 5.2.1 is provided. The proof requires the development of new results on the stability of infinite-dimensional linear systems. A discussion of the results and their proofs is given in this chapter.

■ 6.1 The Framework

In the framework of this chapter, the states and measurements in the estimation problem take on values in Hilbert spaces. The discussion in this chapter makes use of two different topologies, strong and weak, and associated notions of convergence [69, Section 3.11] . A sequence $\{u_i\}$ is said to converge in the strong topology, or strongly, to a point u_* if

$$\lim_{i \to \infty} \|u_i - u_*\| = 0 \tag{6.1}$$

where $\|\cdot\|$ is the standard norm induced by the inner product. A sequence $\{u_i\}$ is said to converge in the weak topology, or weakly, to a point u_* if

$$\lim_{i \to \infty} \langle v, u_i - u_* \rangle = 0 \tag{6.2}$$

for all vectors v in Hilbert space where $\langle \cdot, \cdot \rangle$ is the inner product. The discussion in this chapter also makes use of a notion of positive definiteness. Specifically, a symmetric operator M is positive definite if

$$\langle v, Mv \rangle > 0 \tag{6.3}$$

for all vectors $v \neq 0$ in the Hilbert space.

The remainder of this chapter considers two Hilbert Spaces X and Y. The states $x(t)$ take on values in X, and measurements $y(t)$ take on values in Y. The operators $A : X \mapsto X$, $C : X \mapsto Y$, $\Lambda_n : Y \mapsto Y$, and $\Lambda_w : X \mapsto X$ are all assumed to be bounded linear mappings and play the same roles as in the finite dimensional settings of (5.1) and (5.3). In addition, the operator $\Lambda_n(t)$ is assumed to have a bounded inverse for each t. The filtering equations are still given by (5.6)-(5.9) [34, p. 297]. Of principal interest is the stability of the recursion for the updated estimates. The unforced dynamics are given by

$$\hat{x}(t+1|t+1) = \Lambda_e(t|t-1)C^*(t)(C(t)\Lambda_e(t|t-1)C^*(t) + \Lambda_n(t))^{-1}C(t)A(t)\hat{x}(t|t). \tag{6.4}$$

The stability results discussed in this chapter make various assumptions concerning certain reachability and observability grammians. The reachability grammian of interest is defined by

$$\mathcal{R}(t,s) \triangleq \sum_{\tau=s}^{t-1} \Phi(t,\tau+1)\Lambda_w(\tau)\Phi^*(t,\tau+1). \tag{6.5}$$

This grammian measures how much noise has entered the state between times s and t. The observability grammian of interest is defined by

$$\mathcal{I}(t,s) \triangleq \sum_{\tau=s}^{t} \Phi(\tau,t)C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)\Phi(\tau,t). \tag{6.6}$$

This grammian measures the quality of the measurements between times s and t.

In finite dimensions, there are a variety of existing results concerning the stability of the Kalman filter. The following theorem states that the filter is exponentially stable if there are uniform upper and lower bounds on the reachability and observability grammians [43, 45, 46].

Theorem 6.1.1. *Suppose the filtering problem is finite dimensional and that there exist constants $\alpha_1, \alpha_2, \beta_1, \beta_2, T > 0$ such that*

$$\alpha_1 I \leq \mathcal{R}(t,t-T) \leq \alpha_2 I \qquad \forall t \geq T \tag{6.7}$$

$$\beta_1 I \leq \mathcal{I}(t,t-T) \leq \beta_2 I \qquad \forall t \geq T. \tag{6.8}$$

Then, there exist constants $c_1, c_2 > 0$ such that

$$\|z(t)\| \leq c_1 e^{c_2 t} \tag{6.9}$$

where $z(t)$ follows the dynamics (6.4). Moreover, the constants c_1 and c_2 are independent of the initial condition for $z(t)$.

Hager and Horowitz have considered a slightly different stability issue but in an infinite-dimensional setting [34]. Their focus is on the stability of the filter error dynamics, *i.e.* the dynamics of $e(t) = x(t) - \hat{x}(t|t)$. These are given by

$$e(t+1) = (I - \Lambda_e(t|t-1)C^*(t)(C(t)\Lambda_e(t|t-1)C^*(t) + \Lambda_n(t))^{-1}C(t))A(t)e(t). \tag{6.10}$$

The main result regards the time-invariant case and states the following [34, Theorem 9].

Theorem 6.1.2. *Consider the time-invariant case for which* $A(t) = A$, $C(t) = C$, $\Lambda_n(t) = \Lambda_n$, *and* $\Lambda_w(t) = \Lambda_w$. *Now, suppose*

1. *The measurement noise is uniformly positive definite,* i.e. *there exists a constant* $0 < \sigma_n^2 < \infty$ *such that*

$$\Lambda_n \geq \sigma_n^2 I \tag{6.11}$$

2. *The system is observable in the sense that there exists an integer* $r \geq 0$ *and constant* $0 < \alpha < \infty$ *such that*

$$\sum_{i=0}^{r} A^i C^* C (A^*)^i \geq \alpha I \tag{6.12}$$

3. *The system is reachable in the sense that exists an integer* $s \geq 0$ *and constant* $0 < \beta < \infty$ *such that*

$$\sum_{i=0}^{s} (A^*)^i \Lambda_w A^i x) \geq \beta I. \tag{6.13}$$

Then, there exist constants $c_1, c_2 > 0$ *such that*

$$\|z(t)\| \leq c_1 e^{c_2 t} \tag{6.14}$$

where $z(t)$ *follows the dynamics (6.10). Moreover, the constants* c_1 *and* c_2 *are independent of the initial condition for* $z(t)$.

The goal of this chapter is to establish stability of the Kalman filter dynamics that relies on assumptions weaker than the uniform reachability and observability criteria of (6.7), (6.8), (6.13), and (6.12). The reason for doing this is that (6.7) and (6.13) may not hold in a remote sensing space-time estimation problem. This follows from the fact that the driving noise $\Lambda_w(t)$ may be smooth. Imposing the weaker condition that the reachability grammian (6.7) be positive definite, we are able to prove strong stability of the filter. By strong stability of the filter dynamics, we mean that $z(t)$ converges strongly to 0,

$$\lim_{t \to \infty} \|z(t)\| = 0, \tag{6.15}$$

where $z(t)$ follows the dynamics (6.4). By relaxing the assumptions further and only requiring that the observability grammian (6.8) be positive definite, we are able to prove weak stability of the filter. By weak stability, we mean that $z(t)$ converges weekly to 0,

$$\lim_{t \to \infty} \langle z(t), z' \rangle = 0 \qquad \forall z' \in X, \tag{6.16}$$

where, again, $z(t)$ follows the dynamics (6.4).

The proofs of our results are extensions of those given in [43,45,46] for the finite-dimensional case rather than those in [34] for the infinite-dimensional case. In particular, we make use of Lyapunov functions. This has necessitated the development of an appropriate Lyapunov theory for infinite-dimensional linear systems. The standard proofs relating Lyapunov functions to asymptotic stability in finite dimensions [47,48] rely on closed and bounded sets being compact [47, p. 379]. This property, characteristic of all finite-dimensional topological vector spaces, does not necessarily hold in infinite dimensions [69, p. 9 and 16-18].

Many others have extended various aspects of Lyapunov theory to infinite-dimensional dynamical systems. Some references include [7,20,21,35,55–60,68]. However only two of these references consider strong or weak stability. In particular, Ross states a theorem concerning weak stability of time-invariant linear systems [68, Theorem 2.1.2]. This theorem does not apply to the dynamics of the Kalman filter because of the theorem's assumption concerning the time-invariance of the system whose stability is in question. Massera and Schäffer state a theorem concerning strong stability of time-varying systems [58, Theorem 4.1]. This theorem does not apply to the case under consideration for a more subtle reason than for Ross's theorem. Specifically, Massera and Schäffer require that the Lyapunov function, $V(x,t)$, have what they term "an infinitely small upper bound". This means that there must exist a continuous function a such that

$$V(x,t) \leq a(\|x\|) \qquad \forall x, t. \tag{6.17}$$

The existence of the infinitely small upper bound guarantees that the Lyapunov function is bounded over every closed ball centered at the origin and that the bound changes continuously with the radius of the ball. We will need to relax this restriction.

We develop these ideas more fully over the next few sections. In Section 6.2, theorems regarding the boundedness of the error covariances are stated and proved. Then, a Lyapunov theory for strong stability is developed in Section 6.3. This is applied to demonstrate strong stability of the filter in Section 6.4. Next, a Lyapunov theory for weak stability is developed in Section 6.5. Finally, this is applied in Section 6.6 to demonstrate weak stability of the filter under more relaxed restrictions than those in Section 6.4.

■ 6.2 Boundedness of the Error Covariances

The first step to proving the stability of the Kalman filter is to bound the update error covariance. The following two theorems provide such bounds given conditions on the reachability and observability grammians, (6.5) and (6.6). The development follows that of [45,46].

Theorem 6.2.1. *Suppose there exist constants $\alpha_1, \alpha_2, \beta_1, \beta_2, T \geq 0$ such that*

$$\alpha_1 I \leq \mathcal{R}(t, t-T) \leq \alpha_2 I \qquad \forall t \geq T \tag{6.18}$$

$$\beta_1 I \leq \mathcal{I}(t, t-T) \leq \beta_2 I \qquad \forall t \geq T. \tag{6.19}$$

Then, the update error covariance of the Kalman filter satisfies

$$\Lambda_e(t|t) \leq \left(\frac{1}{\beta_1} + T\frac{\beta_2^2 \alpha_2}{\beta_1^2}\right) I. \tag{6.20}$$

A proof of the upper bound in Theorem 6.2.1 for finite-dimensional systems is given in [45]. The proof extends to the infinite-dimensional setting here, without modification. The next theorem provides a lower bound on the update error covariance. The theorem statement and proof are modifications of those in [46] that take into account the errors cited in [45].

Theorem 6.2.2. *Suppose that $\forall t$ $\Lambda_w(t)$ and Λ_x have bounded inverses. Moreover, suppose that there exist constants $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, T \geq 0$ such that*

$$\alpha_1 I \leq \mathcal{R}(t, t-T-1) \leq \alpha_2 I \qquad \forall t \geq T+1 \tag{6.21}$$

$$\beta_1 I \leq \mathcal{I}(t-1, t-T) \leq \beta_2 I \qquad \forall t \geq T \tag{6.22}$$

$$\|A(t)^{-1}\| \leq \frac{1}{\gamma_1} \qquad \forall t \tag{6.23}$$

$$\|A(t)\| \leq \gamma_2 \qquad \forall t. \tag{6.24}$$

Then, the update error covariance of the Kalman filter satisfies

$$\Lambda_e(t|t) \geq \left(\frac{\alpha_1^2 \gamma_1}{\alpha_1 \gamma_1 + T\alpha_2^2 \beta_2} + \beta_2\right) I. \tag{6.25}$$

Proof. Consider the system

$$x'(t+1) = A^{-*}x'(t) + A^{-*}(t)C^*(t)w'(t) \tag{6.26}$$

$$y'(t) = x'(t) + n'(t) \tag{6.27}$$

where $\mathrm{Cov}(w'(t)) = \Lambda_n^{-1}(t)$, $\mathrm{Cov}(n'(t)) = \Lambda_w^{-1}(t-1)$, and the system is initialized at time 0 with no measurement and $\Lambda_{x'} = \Lambda_x^{-1}$. The reachability and observability grammians for this system are

$$\mathcal{R}'(t,s) = \sum_{\tau=s}^{t-1} \Phi^{-*}(t, \tau+1)\Phi^{-*}(\tau+1, \tau)C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)\Phi^{-1}(\tau+1, \tau)\Phi^{-1}(t, \tau+1) \tag{6.28}$$

$$\mathcal{I}'(t,s) = \sum_{\tau=s}^{t} \Phi^{-1}(\tau, t)\Lambda_w^{-1}(\tau-1)\Phi^{-*}(\tau, t). \tag{6.29}$$

One can rewrite the reachability grammian as

$$\begin{aligned}\mathcal{R}'(t,s) &= A^{-*}(t-1)\left(\sum_{\tau=s}^{t-1}\Phi^*(\tau,t-1)C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)\Phi(\tau,t-1)\right)A^{-1}(t-1)\\ &= A^{-*}(t-1)\,\mathcal{I}(t-1,s)A^{-1}(t-1),\end{aligned} \tag{6.30}$$

and the observability grammian as

$$\begin{aligned}\mathcal{I}'(t,s) &= \sum_{\tau=s}^{t}\Phi(t,\tau)\Lambda_w^{-1}(\tau-1)\Phi^*(t,\tau)\\ &= \sum_{\tau=s-1}^{t-1}\Phi(t,\tau+1)\Lambda_w^{-1}(\tau)\Phi^*(t,\tau+1)\\ &= \mathcal{R}(t,s-1).\end{aligned} \tag{6.31}$$

By (6.21), (6.22), and (6.23),

$$\frac{\beta_1}{\gamma_2}I \leq \mathcal{R}'(t,t-T) \leq \frac{\beta_2}{\gamma_1} \qquad \forall t \geq T \tag{6.32}$$

$$\alpha_1 \leq \mathcal{I}'(t,t-T) \leq \alpha_2 I \qquad \forall t \geq T. \tag{6.33}$$

Theorem 6.2.2 then implies that the error covariance of the new system satisfies

$$\Lambda_e'(t|t) \leq \left(\frac{1}{\alpha_1} + T\frac{\alpha_2^2\beta_2}{\alpha_1^2\gamma_1}\right)I, \tag{6.34}$$

and, thus,

$$(\Lambda_e')^{-1}(t|t) \geq \left(\frac{\alpha_1^2\gamma_1}{\alpha_1\gamma_1 + T\alpha_2^2\beta_2}\right)I. \tag{6.35}$$

Now, the recursions for the Kalman filter error covariances for the new system are

$$(\Lambda_e')^{-1}(t|t) = (\Lambda_e')^{-1}(t|t-1) + \Lambda_w(t-1) \tag{6.36}$$

$$\Lambda_e'(t+1|t) = A^{-*}\Lambda_e'(t|t)A^{-1}(t) + A^{-*}C^*(t)\Lambda_n^{-1}(t)C(t)A^{-1}(t). \tag{6.37}$$

Comparing this with the recursions for $\Lambda_e(t|t)$, the Kalman filter error covariances of the original system, one notices that

$$\Lambda_e^{-1}(t|t) = (\Lambda_e')^{-1}(t|t) + C^*(t)\Lambda_n^{-1}(t)C(t). \tag{6.38}$$

By (6.35) and (6.23),

$$\Lambda_e(t|t) \geq (\Lambda_e')^{-1} \geq \left(\frac{\alpha_1^2\gamma_1}{\alpha_1\gamma_1 + T\alpha_2^2\beta_2} + \beta_2\right)I. \tag{6.39}$$

□

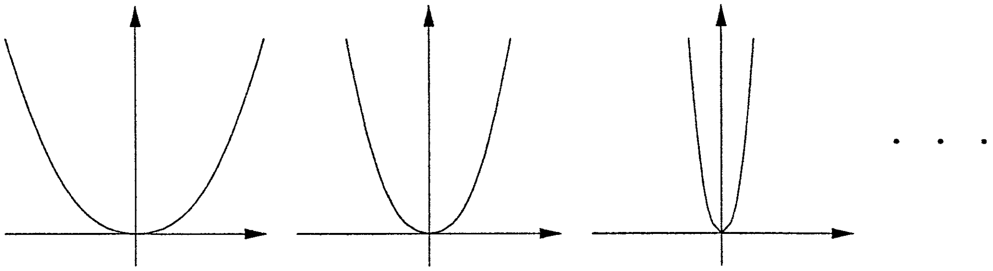


Figure 6.1. This figure illustrates a possible behavior of Lyapunov functions used for proving strong stability. Specifically, the Lyapunov function may grow unbounded in certain directions. This is illustrated here by showing coordinate slices of increasing curvature from a candidate quadratic Lyapunov function.

■ 6.3 Lyapunov Theory for Strong Stability

In finite dimensions, the Lyapunov function used to establish exponential stability of the Kalman filter dynamics is the quadratic form associated with the inverse of the update error covariance, $\Lambda_e^{-1}(t|t)$. One would like to use this same, natural Lyapunov function to establish stability for the space-time filtering problems. For these problems, however, $\Lambda_e^{-1}(t|t)$ is typically an unbounded operator. Thus, $\Lambda_e^{-1}(t|t)$ is not defined on all of X. Yet, $\Lambda_e^{-1}(t|t)$ should still work as a Lyapunov function provided one can establish descent along state trajectories.

The general situation is illustrate in Figure 6.1. The figure shows coordinate slices of increasing curvature for a candidate quadratic Lyapunov function. Since the curvature may become arbitrarily large, the Lyapunov function may take on arbitrarily large values for initial states lying in a bounded set. Thus, given only that the successive differences of the Lyapunov function along state trajectories are bounded from below, one can not expect that the state will converge to zero at a uniform rate across all initial conditions in a bounded set. This would be necessary for exponential stability. However, one does still expect the state to head to zero if trajectories are always decreasing fast enough. One method for doing this is to find a sequence of Lyapunov functions with bounded curvature that converge to the unbounded Lyapunov function. This idea is made more precise in the following theorem.

Theorem 6.3.1. *Let X be a real Hilbert space, and $\mathcal{B}(X)$, the set of bounded linear operators on X.*

Let $z(t) \in X$ evolve according to

$$z(t+1) = F(t)z(t) \tag{6.40}$$

with $F(t) \in \mathcal{B}(X)$. Consider a family of approximations to $z(t)$, $z_\sigma(t) \in X$ for $\sigma \in \mathbb{R}^+$, evolving according to

$$z_\sigma(t+1) = F_\sigma(t)z(t) \tag{6.41}$$

with $z_\sigma(0) = z(0)$ *and* $F_\sigma(t) \in \mathcal{B}(X)$ *and converging pointwise in time for all* t,

$$\lim_{\sigma\to\infty} \|z_\sigma(t) - z(t)\| = 0. \tag{6.42}$$

Now, let $U_\sigma(t) \in \mathcal{B}(X)$ *be a family of symmetric, positive definite operators, and let*

$$V_\sigma(z_\sigma(t), t) = \langle z_\sigma(t), U_\sigma(t) z_\sigma(t)\rangle \tag{6.43}$$

be the associated Lyapunov functions.

Suppose there exists a constant T*, symmetric operators* $W_\sigma(t) \in \mathcal{B}(X)$ *and a symmetric positive definite (but not necessarily bounded) operator* U *such that*

1. $V_\sigma(z_\sigma(t+1), t+1) - V_\sigma(z_\sigma(t), t) \le 0$ *and* $V_\sigma(z_\sigma(t), t) - V_\sigma(z_\sigma(t-T), t-T) \le \langle z_\sigma(t), W_\sigma(t) z_\sigma(t)\rangle$ *for all* $t \ge T$.
2. $\langle z_\sigma(t), W_\sigma(t) z_\sigma(t)\rangle \ge \eta \|z_\sigma(t)\|^2$ *for all* σ *and* $t \ge T$
3. $\lim_{\sigma\to\infty}\langle z(0), U_\sigma(t) z(0)\rangle = \langle z(0), U z(0)\rangle \qquad \forall z(0) \in D(U)$,

Then,

$$\lim_{t\to\infty} \|z(t)\| = 0 \qquad \forall z(0) \in D(U). \tag{6.44}$$

Proof. Note that for any $s \in [0, T)$,

$$0 \le V_\sigma(z_\sigma(t), t) \le V_\sigma(z(0), 0) - \sum_{\tau=1}^{\lfloor \frac{t-s}{T} \rfloor} \langle z_\sigma(\tau T + s), W_\sigma(\tau T + s) z_\sigma(\tau T + s)\rangle \qquad \forall t. \tag{6.45}$$

Thus,

$$\eta \sum_{\tau=1}^{\infty} \|z_\sigma(\tau T + s)\|^2 \le V_\sigma(z(0), 0). \tag{6.46}$$

By Fatou's lemma [78, Theorem 10.29],

$$\eta \sum_{\tau=1}^{\infty} \liminf_{\sigma\to\infty} \|z_\sigma(\tau T + s)\|^2 \le \eta \liminf_{\sigma\to\infty} \sum_{\tau=1}^{\infty} \|z_\sigma(\tau T + s)\|^2 \le \eta \liminf_{\sigma\to\infty} V_\sigma(z(0), 0). \tag{6.47}$$

Thus,

$$\eta \sum_{\tau=1}^{\infty} \|z(\tau T + s)\|^2 \le \langle z(0), U z(0)\rangle < \infty \tag{6.48}$$

and

$$\lim_{t\to\infty} \|z(t)\|^2 = 0 \qquad \forall z(0) \in D(U). \tag{6.49}$$

□

■ 6.4 Strong Stability of the Kalman Filter for Space-Time Estimation

Theorem 6.3.1 can be used to prove strong stability of the Kalman filter under the conditions stated in Theorem 5.2.1, which is restated here for convenience.

Theorem 5.2.1 *Suppose that the dynamics matrix is bounded above and below,* i.e. *that there exist constants* $\gamma_1, \gamma_2 > 0$ *such that*

$$\|A(t)^{-1}\| \leq \frac{1}{\gamma_1} \qquad \forall t \tag{6.50}$$

$$\|A(t)\| \leq \gamma_2 \qquad \forall t. \tag{6.51}$$

and suppose that the system is uniformly observable, i.e. *that there exist constants* $\beta_1, \beta_2, T > 0$ *such that*

$$\beta_1 I \leq \mathcal{I}(t, t-T) \leq \beta_2 I \qquad \forall t \geq T \tag{6.52}$$

Moreover, suppose that the prior covariance on x *is positive-definite:*

$$\Lambda_x > 0, \tag{6.53}$$

and that the measurement noise covariance is bounded below, i.e. *that there exists a constant* $\sigma^2 > 0$ *such that*

$$\Lambda_n(t) \geq \sigma^2 I \qquad \forall t. \tag{6.54}$$

Then, the dynamics of the Kalman filter are strongly stable, i.e.

$$\lim_{t \to \infty} \|z(t)\| = 0 \tag{6.55}$$

where $z(t)$ *obeys the dynamics (6.4).*

Proof. The proof of Theorem 5.2.1 primarily involves constructing a sequence of systems and associated Lyapunov functions so that Theorem 6.3.1 can be applied.

Specifically, consider adding white noise to the process noise and initial covariance, thereby shifting the covariances by a multiple of the identity. For a shift of $(1/\sigma^2)I$ with $\sigma^2 \geq 1$, the new covariances will be

$$\Lambda'_w(t) = \Lambda_w(t) + \frac{1}{\sigma^2} I \tag{6.56}$$

and

$$\Lambda'_x = \Lambda_x + \frac{1}{\sigma^2} I. \tag{6.57}$$

For this new system, all other matrices remain unaltered, *i.e.* $C'(t) = C(t), \Lambda'_n(t) = \Lambda_n(t), A'(t) = A(t)$. Let $z_\sigma(t)$ be a sequence of states propagating according to the filter dynamics for the new system, and $z(t)$ for the original system. Then, for each time point t,

$$\lim_{\sigma\to\infty} \|z_\sigma(t) - z(t)\| = 0 \tag{6.58}$$

if $z_\sigma(0) = z(0)$ by the continuous dependence on σ of the filtering dynamics (6.4). Thus, $z_\sigma(t)$ provides information about $z(t)$ for large σ. One can use Lyapunov functions and Theorem 6.3.1 to establish a precise link.

Now, since $C'(t) = C(t), \Lambda_n'(t) = \Lambda_n(t), A'(t) = A(t)$, the observability grammian of the new system, $\mathcal{I}_\sigma'(t,s)$, is the same as the original, $\mathcal{I}(t,s)$. By the assumptions of Theorem 5.2.1,

$$\beta_1 I \leq \mathcal{I}_\sigma'(t, t-T) \leq \beta_2 I \qquad \forall t \geq T. \tag{6.59}$$

The reachability grammian of the new system is also bounded above and below due to the shift (6.56). Specifically, there exists a constant α_2 such that $\forall t \geq T$.

$$\begin{aligned}
\frac{1}{\sigma^2} I &\leq \mathcal{R}_\sigma'(t, t-T) && (6.60)\\
&= \sum_{\tau=t-T}^{t-1} \Phi'(t,\tau+1)\Lambda_w'(\tau)(\Phi')^*(t,\tau+1) && (6.61)\\
&\leq \sum_{\tau=t-T}^{t} \Phi'(t,\tau+1)\Lambda_w'(\tau)(\Phi')^*(t,\tau+1) && (6.62)\\
&\leq \alpha_2 I. && (6.63)
\end{aligned}$$

Moreover, since

$$\mathcal{R}_\sigma'(t+1, t-T) = A(t) \sum_{\tau=t-T}^{t} \Phi'(t,\tau+1)\Lambda_w'(\tau)(\Phi')^*(t,\tau+1)A^*(t) \tag{6.64}$$

$\mathcal{R}_\sigma'(t+1, t-T)$ can be bounded as follows:

$$\frac{\gamma_1^2}{\sigma^2} I \leq \mathcal{R}_\sigma'(t+1, t-T) \leq \gamma_2^2 \alpha_2 I \qquad \forall t \geq T. \tag{6.65}$$

By Theorems 6.2.1 and 6.2.2, the update error covariance of the modified system, $\Lambda_{e,\sigma}'(t|t)$, is bounded above and below as follows:

$$\left(\frac{\gamma_1^5}{\sigma^2\gamma_1^3 + (T+1)\sigma^4\alpha_2^2\beta_2\gamma_2^4} + \beta_2\right) I \leq \Lambda_{e,\sigma}'(t|t) \leq \left(\frac{1}{\beta_1} + T\frac{\beta_2^2\alpha_2\gamma_2^2}{\beta_1^2}\right) I \qquad \forall t \geq T. \tag{6.66}$$

Hence, one can consider using

$$V_\sigma(x,t) = \langle x, \left(\Lambda_{e,\sigma}'(t|t)\right)^{-1} x\rangle \qquad t \geq T \tag{6.67}$$

as candidate Lyapunov functions for the dynamics of the Kalman filter for the modified system.

In order to use this sequence of systems and associated Lyapunov functions to establish strong stability of the original filter dynamics (6.4), one needs to verify each of the three major conditions of Theorem 6.3.1 for $U_\sigma(t) = (\Lambda'_{e,\sigma}(t|t))^{-1}$, $U = \Lambda_e^{-1}(T|T)$, and $W_\sigma(t) = \mathcal{I}(t, t-T)$. The verification follows. Note that in what follows, the trajectories of the Lyapunov functions are examined starting at time T, not 0.

1. In [46, p. 764], a bound on the differences of $V_\sigma(z_\sigma(t), t)$ is established in finite dimensions. The derivation of the bound holds in general Hilbert spaces. The bound states that

$$V_\sigma(z_\sigma(t+1), t+1) - V_\sigma(z_\sigma(t), t) \leq 0 \tag{6.68}$$

$$V_\sigma(z_\sigma(t), t) - V_\sigma(z_\sigma(t-T), t-T) \leq \sum_{\tau=t-T}^{t} \langle z(\tau), C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)z(\tau)\rangle. \tag{6.69}$$

Since

$$\sum_{\tau=t-T}^{t} \langle z(\tau), C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)z(\tau)\rangle = \sum_{\tau=t-T}^{t} \langle \Phi(\tau,t)z(t), C^*(\tau)\Lambda_n^{-1}(\tau)C(\tau)\Phi(\tau,t)z(t)\rangle = \langle z(t), \mathcal{I}(t,t-T)z(t)\rangle, \tag{6.70}$$

one has that

$$V_\sigma(z_\sigma(t), t) - V_\sigma(z_\sigma(t-T), t-T) \leq \langle z(t), \mathcal{I}(t,t-T)z(t)\rangle. \tag{6.71}$$

2. $\langle z_\sigma(t), \mathcal{I}(t,t-T)z(t)\rangle \geq \beta_1 \|z_\sigma(t)\|^2$ for all σ and $t \geq T$ by (6.52).

3. Finally, $\lim_{\sigma\to\infty}\langle z(T), (\Lambda'_{e,\sigma}(T|T))^{-1}z(T)\rangle = \langle z(T), \Lambda_e^{-1}(T|T)z(T)\rangle$ for all initial conditions $z(0)$. The limit follows simply from the fact that the estimator is a continuous function of σ. That $z(T) \in D(\Lambda_e^{-1}(T|T))$ follows from the fact that $z(T) \in R(\Lambda_e(T|T-1))$ by (6.4) and that $\Lambda_e^{-1}(T|T) = \Lambda_e^{-1}(T|T-1) + C^*(T)\Lambda_n^{-1}(T)C(T)$ where

$$C^*(T)\Lambda_n^{-1}(T)C(T) \leq \mathcal{I}(T,0) \leq \beta_2 I. \tag{6.72}$$

Now, Theorem 6.3.1 implies that

$$\lim_{t\to\infty} \|z(t)\| = 0 \tag{6.73}$$

where $z(t)$ follows the filter dynamics (6.4). □

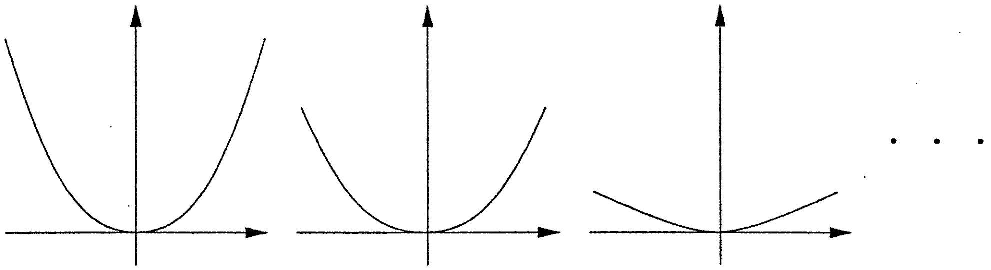


Figure 6.2. This figure illustrates the behavior of Lyapunov functions that may be used for proving weak but not strong stability. The graphs depict coordinate slices of the bounds on the negative magnitude of successive differences of a candidate Lyapunov function. These are guaranteed to be positive, but the curvature may be arbitrarily small.

6.5 Lyapunov Theorem for Weak Stability

The focus of the previous sections is on sufficient conditions for strong stability. A natural question to ask is whether one can guarantee weak stability, as defined in (6.16), under relaxed versions of the conditions in Theorems 6.3.1 and 5.2.1. In this section, a Lyapunov theory is developed for weak stability when Condition 2 in Theorem 6.3.1 is relaxed so that W_σ need only be positive definite and not bounded below by a multiple of the identity.

Figure 6.2 provides some intuition as to why W_σ being positive definite is sufficient for establishing weak stability but not strong stability of a given system. The figure depicts the quadratic form associated with W_σ, which provides a lower bound on the negative magnitude of successive differences of the Lyapunov function. Each graph in Figure 6.2 plots a different coordinate slice of the quadratic function. That the function is positive away from the origin implies that each coordinate of the state of the given system, $z(t)$, is tending to zero. Thus, one expects

$$\lim_{t\to\infty} \langle z', z(t)\rangle = 0 \qquad \forall z' \in X. \tag{6.74}$$

However, the curvature of each slice may be arbitrarily small so that $\|z(t)\|$ is not necessarily tending to zero.

These ideas are made precise in the following theorem. The statement is almost the same as that of Theorem 6.3.1. The primary difference is that the lower bound on $W_\sigma(t)$ is replaced by a weaker condition in terms of the ranges of the limiting operator $W(t)$. This is motivated by Ross's work on using Lyapunov functions to establish weak stability of time-invariant systems [68, Theorem 2.1.2].

Theorem 6.5.1. *Let X be a real Hilbert space, and $\mathcal{B}(X)$, the set of bounded linear operators on X.*

Let $z(t) \in X$ evolve according to

$$z(t+1) = F(t)z(t) \tag{6.75}$$

with $F(t) \in \mathcal{B}(X)$. Consider a family of approximations to $z(t)$, $z_\sigma(t) \in X$ for $\sigma \in \mathbb{R}^+$, evolving according to

$$z_\sigma(t+1) = F_\sigma(t)z(t) \tag{6.76}$$

with $z_\sigma(0) = z(0)$ and $F_\sigma(t) \in \mathcal{B}(X)$ and converging pointwise in time for all t,

$$\lim_{\sigma\to\infty} \|z_\sigma(t) - z(t)\| = 0. \tag{6.77}$$

Now, let $U_\sigma(t) \in \mathcal{B}(X)$ be a family of symmetric, positive definite operators, and let

$$V_\sigma(z_\sigma(t), t) = \langle z_\sigma(t), U_\sigma(t)z_\sigma(t)\rangle \tag{6.78}$$

be the associated Lyapunov functions.

Suppose there exists constants T and η; bounded symmetric positive definite operators $W_\sigma(t), W(t), G(t)$; a symmetric positive definite (but not necessarily bounded) operator U; and real-valued function $M(\cdot)$ on $D(U)$ such that

1. $V_\sigma(z_\sigma(t+1), t+1) - V_\sigma(z_\sigma(t), t) \leq 0$ *and* $V_\sigma(z_\sigma(t), t) - V_\sigma(z_\sigma(t-T), t-T) \leq \langle z_\sigma(t), W_\sigma(t)z_\sigma(t)\rangle$
2. $\lim_{\sigma\to\infty}\langle z(0), U_\sigma(t)z(0)\rangle = \langle z(0), Uz(0)\rangle \qquad \forall z(0) \in D(U)$,
3. $\langle z', U_\sigma(0)z'\rangle \leq M(z') \qquad \forall z' \in D(U)$
4. $\lim_{\sigma\to\infty}\langle z', W_\sigma(t)z'\rangle = \langle z', W(t)z'\rangle \qquad \forall z' \in X$
5. $G^2(t) = W(t)$ *and* $\overline{\bigcap_t R(G(t))} = X$, *where the bar denotes closure in the strong topology* [1]
6. $\langle z', U_\sigma(t)z'\rangle \geq \eta\|z'\|^2$ *for all σ, t and all $z' \in X$.*

Then,

$$\lim_{t\to\infty}\langle z(t), z'\rangle = 0 \qquad \forall z(0) \in D(U), z' \in X. \tag{6.79}$$

Proof. The proof is broken down into three steps.

1. The first step is to note that

$$0 \leq V_\sigma(z_\sigma(t), t) \leq V_\sigma(z(0), 0) - \sum_{\tau=1}^{\lfloor\frac{t-s}{T}\rfloor} \langle z_\sigma(\tau T + s), W_\sigma(\tau T + s)z_\sigma(\tau T + s)\rangle \leq V_\sigma(z(0), 0) \qquad \forall t. \tag{6.80}$$

[1] Note that every bounded symmetric positive definite operator has a bounded symmetric positive definite square root [69, pp. 330-331].

2. The next step is to show that

$$\lim_{t\to\infty} \langle z(t), W(t)z(t)\rangle = 0 \qquad \forall z(0) \in D(U). \tag{6.81}$$

To see this, note that by Fatou's lemma [78, Theorem 10.29] and (6.80),

$$\sum_{\tau=1}^{\infty} \liminf_{\sigma\to\infty} \langle z_\sigma, W_\sigma(\tau T + s)z_\sigma(\tau T + s)\rangle \leq \liminf_{\sigma\to\infty} \sum_{\tau=1}^{\infty} \langle z_\sigma, W_\sigma(\tau T + s)z_\sigma(\tau T + s)\rangle \tag{6.82}$$

$$\leq \liminf_{\sigma\to\infty} V_\sigma(z(0), 0). \tag{6.83}$$

for any $s \in [0, T)$. Thus,

$$\sum_{\tau=1}^{\infty} \langle z(\tau T + s), W(\tau T + s)z(\tau T + s)\rangle \leq \langle z(0), Uz(0)\rangle < \infty \tag{6.84}$$

and (6.81) follows.

3. Lastly, weak stability is established. Now, one can rewrite (6.81) as

$$\lim_{t\to\infty} \|G(t)z(t)\|^2 = 0 \qquad \forall z(0) \in D(U) \tag{6.85}$$

by Condition 5. That $G(t)z(t)$ converges to zero strongly implies it converges to zero weakly, *i.e.*

$$\lim_{t\to\infty} \langle G(t)z(t), z'\rangle = 0 \qquad \forall z' \in X, z(0) \in D(U). \tag{6.86}$$

Fix $z' \in X$. Let $\{z'_m\} \subseteq \bigcap_t R(G(t))$ be a sequence converging to z' strongly. One can do this by Condition 5. Then,

$$|\langle z(t), z'\rangle| = |\langle z(t), z' - z'_m + z'_m\rangle| \tag{6.87}$$

$$\leq |\langle z(t), z'_m\rangle| + |\langle z(t), z' - z'_m\rangle| \tag{6.88}$$

$$\leq |\langle z(t), z'_m\rangle| + \|z(t)\|\|z' - z'_m\| \tag{6.89}$$

$$\leq |\langle z(t), z'_m\rangle| + \frac{M(z(0))}{\eta}\|z' - z'_m\| \tag{6.90}$$

where the second to last step follows from the Cauchy-Schwartz inequality, and the last step follows from Conditions 3 and 6. Fix $\varepsilon > 0$. Then, there exists m_0 such that

$$\frac{M(z(0))}{\eta}\|z' - z'_{m_0}\| \leq \frac{\varepsilon}{2}. \tag{6.91}$$

Moreover, there exists a t_0 such that

$$|\langle z(t), z'_{m_0}\rangle| \leq \frac{\varepsilon}{2} \tag{6.92}$$

for all $t > t_0$ and hence

$$|\langle z(t), z'\rangle| \leq \varepsilon \tag{6.93}$$

for all $t > t_0$. Since this holds for all $\varepsilon > 0$,

$$\lim_{t\to\infty} |\langle z(t), z'\rangle| = 0 \qquad \forall z' \in D(u). \tag{6.94}$$

□

■ 6.6 Weak Stability of the Kalman Filter for Space-time Estimation

Theorem 6.5.1 can be used to establish weak stability of the Kalman filter when the observability condition (6.52) of Theorem 5.2.1 is weakened. The condition (6.52) is replaced with two mild restrictions on the observability grammian $\mathcal{I}(t,s)$. The first requirement is that there exists a T for which $\mathcal{I}(t,t-T)$ is positive definite for all $t \geq T$. This merely requires that there exist measurements of the entire state over an interval of time T. These measurements need not have a lower bound on their quality, however, because $\mathcal{I}(t,t-T)$ is not required to be bounded from below by a multiple of the identity. The other principal restriction is that

$$\overline{\bigcap_{t\geq T} R((\mathcal{I}(t,t-T))^{1/2})} = X \tag{6.95}$$

hold. This is a fairly mild restriction. In particular, the range of $(\mathcal{I}(t,t-T))^{1/2}$ is dense if $\mathcal{I}(t,t-T)$ is positive definite [69, Theorem 12.12b]. Moreover, the ranges of $(\mathcal{I}(t,t-T))^{1/2}$, for all times t, will overlap significantly in many cases. In particular, this will hold for satellite measurements, which tend to repeat on cycles dictated by the orbit of the satellite. Thus, the following theorem can provide a useful characterization of Kalman filter stability when measurements are not of uniform quality.

Theorem 6.6.1. *Suppose that there exist constants $\beta_2, \gamma_1, \gamma_2 T$ such that*

$$0 < \mathcal{I}(t,t-T) \leq \beta_2 I \qquad \forall t > T \tag{6.96}$$

$$\gamma_1 I \leq A(t) \leq \gamma_2 I \qquad \forall t \tag{6.97}$$

and that the square-root of $\mathcal{I}(t,t-T)$ satisfies

$$\overline{\bigcap_{t\geq T} R((\mathcal{I}(t,t-T))^{1/2})} = X. \tag{6.98}$$

Moreover, suppose that the prior covariance on x is positive definite:

$$\Lambda_x > 0. \tag{6.99}$$

Then, the dynamics of the Kalman filter are weakly stable, i.e.

$$\lim_{t\to\infty} \langle z(t), z' \rangle = 0 \qquad \forall z' \in X \tag{6.100}$$

where $z(t)$ obeys the dynamics (6.4).

Proof. The proof of Theorem 6.6.1 makes use of the same sequence of systems and associated Lyapunov functions used in the proof of Theorem 5.2.1.

The differences between the proofs of Theorems 5.2.1 and 6.6.1 start with the bounds on the update error covariance for the modified system. Specifically, by Theorem 6.2.2, the update error covariance of the modified system, $\Lambda'_{e,\sigma}(t|t)$, is bounded below as follows:

$$\left(\frac{\gamma_1^5}{\sigma^2\gamma_1^3 + (T+1)\sigma^4(\alpha'_2)^2\beta_2\gamma_2^2} + \beta_2 \right) I \leq \Lambda'_{e,\sigma}(t|t). \tag{6.101}$$

Moreover, the error covariance $\Lambda'_{e,\sigma}(t|t)$ is always bounded above by the prior covariance of the state. Hence, one can consider using

$$V_\sigma(x,t) = \langle x, \left(\Lambda'_{e,\sigma}(t|t)\right)^{-1} x \rangle \qquad t \geq T \tag{6.102}$$

as candidate Lyapunov functions for the dynamics of the Kalman filter for the modified system.

In order to use this sequence of systems and associated Lyapunov functions to establish strong stability of the original filter dynamics (6.4), one needs to verify each of the six major conditions of Theorem 6.3.1 for $U_\sigma(t) = (\Lambda'_{e,\sigma}(t|t))^{-1}$, $U = \Lambda_e^{-1}(T|T)$, $W_\sigma(t) = W = \mathcal{I}(t, t-T)$, $G(t) = (\mathcal{I}(t, t-T))^{1/2}$, $M(z') = \max_{\sigma\geq 1}\langle z', (\Lambda'_{e,\sigma}(T|T))^{-1}, z'\rangle$, and $\eta = \beta_2$. The verification follows. Note that, in what follows, the trajectories of the Lyapunov function are examined starting at time T, not 0.

1. As in the proof of Theorem 5.2.1,

$$V_\sigma(z_\sigma(t+1), t+1) - V_\sigma(z_\sigma(t), t) \leq 0 \tag{6.103}$$

$$V_\sigma(z_\sigma(t), t) - V_\sigma(z_\sigma(t-T), t-T) \leq \langle z(t), \mathcal{I}(t, t-T) z(t) \rangle. \tag{6.104}$$

2. Also for the same reasons as in the proof of Theorem 5.2.1,

$$\lim_{\sigma\to\infty} \langle z(T), (\Lambda'_{e,\sigma}(T|T))^{-1} z(T) \rangle = \langle z(T), \Lambda_e^{-1}(T|T) z(T) \rangle \tag{6.105}$$

and $z(T) \in D(\Lambda_e^{-1}(T|T))$ for all initial conditions $z(0)$.

3. Now,

$$\max_{\sigma\geq 1}\langle z', (\Lambda'_{e,\sigma}(T|T))^{-1}, z'\rangle \tag{6.106}$$

exists for all $z' \in D(\Lambda_e^{-1}(T|T))$ by continuity of $(\Lambda'_{e,\sigma})^{-1}$ as a function of σ and the existence of a limit as σ tends to infinity. Thus,

$$\langle z', (\Lambda'_{e,\sigma}(T|T))^{-1}z'\rangle \leq \max_{\sigma\geq 1}\langle z', (\Lambda'_{e,\sigma}(T|T))^{-1}, z'\rangle \qquad \forall z' \in D(\Lambda_e^{-1}(T|T)). \tag{6.107}$$

4. $\lim_{\sigma\to\infty}\langle z', \mathcal{I}_(t, t-T)z'\rangle = \langle z', \mathcal{I}(t, t-T)z'\rangle$ identically.

5. $\overline{\bigcap_{t\geq T} R((\mathcal{I}(t,t-T))^{1/2})} = X$ by the assumptions of Theorem 6.6.1

6. $\langle z', (\Lambda'_{e,\sigma}(t|t))^{-1}z'\rangle \geq \beta_2\|z'\|^2$ for all $t \geq T$ by (6.101).

Now, Theorem 6.5.1 implies that

$$\lim_{t\to\infty}\langle z(t), z'\rangle = 0 \qquad \forall z' \in X \tag{6.108}$$

where $z(t)$ follows the filter dynamics (6.4). □

■ 6.7 Summary

The set of theorems presented in this chapter establish stability of the Kalman filter under restrictions mild enough that they apply to scenarios arising in remote sensing. In particular, the numerical examples in Section 5.5 have positive definite driving noise and measurement structures such that the observability grammian is bounded below by a multiple of the identity. Thus, the Kalman filters for these problems are strongly stable. In the dual case for which the measurements are of poorer quality but the observability grammian is still positive definite, the Kalman filter would be weakly stable.

Chapter 7

Conclusions and Open Problems

The contributions of this thesis center on the development of three algorithms as well as a Lyapunov theory for studying the stability of infinite-dimensional linear systems. Although each of these algorithms and the Lyapunov theory have been extensively researched and discussed in this thesis, there remain several open problems. These contributions and open problems are discussed in the remainder of the chapter in more detail.

7.1 Contributions

The major contributions for each chapter are summarized, as follows.

7.1.1 Krylov Subspace Estimation Algorithm

One of the major contributions of this thesis is the derivation of a Krylov subspace algorithm for linear least-squares estimation. The algorithm is a variant of the conjugate gradient algorithm that simultaneously computes both linear least-squares estimates and error variances. Moreover, there are two natural criteria that can be used for determining when to stop the algorithm.

The convergence of the Krylov subspace estimation algorithm is extensively analyzed in this thesis. The convergence analysis has contributed not only to a better understanding of the algorithm but also to a fundamental characterization of random processes. In particular, the analysis has necessitated the development of new results concerning the extrema of sequences of independent random variables. The final conclusions of the convergence analysis, however, are very specific to, and provide important insight into, the Krylov subspace estimation algorithm. Specifically, they suggest two different methods for accelerating convergence. The first is to use preconditioning, and the second, multiple starting vectors.

The Krylov subspace estimation algorithm, in its standard, preconditioned, and block forms, has been extensively characterized with numerical examples in this thesis. The examples include ones with both synthetic and real data. The conclusion is that the Krylov subspace estimation algorithm is an efficient method for simultaneously computing both linear least-squares estimates and error variances for many problems. Moreover, use of preconditioning or a block form may decrease the computational work required to achieve a given level of accuracy.

7.1.2 Krylov Subspace Realization Algorithm

This thesis also presents a Krylov subspace algorithm for realization. It is a variant of the Krylov subspace estimation algorithm that can simultaneously compute a sample path of a given Gaussian stochastic process as well as a low-rank approximation to the covariance matrix of the given process. The convergence analysis of the Krylov subspace estimation algorithm applies to the realization algorithm, as well. As a consequence of this, one can use preconditioning to accelerate the convergence of the Krylov subspace realization algorithm.

The realization algorithm is characterized in this thesis both analytically and numerically. The analytical characterization consists primarily of the comparison of computational work between the Krylov subspace realization algorithm and other realization algorithms. The algorithms used for comparison include FFT-based approaches as well as iterative approaches. The Krylov subspace realization algorithm is also compared against these alternative algorithms in numerical experiments. The conclusion is that, in many cases, the Krylov subspace realization algorithm is an efficient method for simultaneously computing sample paths and low-rank approximations to the covariance matrix of a given process.

7.1.3 A Krylov Subspace Method for Space-Time Estimation

The problem of space-time estimation is also addressed in this thesis. The Krylov subspace estimation and realization algorithms have been combined to form recursive algorithms for both filtering and smoothing. The algorithms are modifications of the standard Kalman filtering and modified Bryson-Frazier smoothing algorithms. As such, they simultaneously compute both estimates and error variances for, respectively, either the filtering or smoothing problems.

The computational cost and stability of the Krylov subspace algorithms for filtering and smoothing are analyzed in this thesis. The stability of the algorithms is analyzed so as to determine how the approximations made to the Kalman filtering and modified Bryson-Frazier smoothing algorithms propagate through the recursions. The conclusion is that, by following some guidelines on the numbers of update and predict iterations, one can keep the perturbations small. As long as the numbers of iterations required to meet these guidelines is not excessively large, the cost analysis in this thesis demonstrates that the Krylov subspace algorithms provide a computational gain over standard implementations of the Kalman filter and modified Bryson-Frazier smoother.

Numerical examples using both synthetic and real oceanographic data are used to characterize the Krylov subspace algorithms. The synthetic data are generated by a relatively simple heat equation model. The oceanographic data are of sea surface anomaly, the deviations of sea surface height from a mean. The model that is used in this thesis for studying the sea surface anomaly data is based on models developed by others for analyzing similar data. We model the ocean as propagating in time according to a

Rossby wave equation and being forced by stationary noise whose power spectral density obeys power laws derived from data. The Krylov subspace algorithms successfully processed both this oceanographic data as well as the synthetic data. However, the computational gains are much greater for the synthetic data. The conclusion is that the Krylov subspace algorithms provide a computational gain which may be substantial but depends on the details of the model.

■ 7.1.4 Stability of Kalman Filters for Space-Time Estimation

Studying the stability of the Krylov subspace algorithms for filtering and smoothing necessitates the study, in this thesis, of the stability of infinite-dimensional Kalman filters. The types of stability studied are strong and weak stability. This is done by developing a Lyapunov theory for studying strong and weak stability in infinite-dimensional linear systems. The theorems proved in this thesis provide sufficient conditions for such stability. Specifically, the existence of sequences of Lyapunov functions satisfying specified properties can demonstrate strong or weak stability.

There exist sequences of Lyapunov functions satisfying the appropriate conditions needed for establishing stability of Kalman filters for certain types of space-time estimation problems. In particular, if sufficient measurements of good quality exist that the observability Grammian is bounded away from zero, the Kalman filter may be strongly stable. However, if the observability Grammian is only positive definite, the Kalman filter, may only be weakly stable.

■ 7.2 Open Problems

There are number of open problems. A sampling follows, organized by topic.

■ 7.2.1 Krylov Subspace Estimation

Local Functionals of the Data

One of the factors limiting the computational gain of the Krylov subspace estimation algorithm is that the linear functionals, $p_1^T, p_2^T, \ldots$, used to reduce the dimensionality of the data are not local (sparse) when measurements are taken pointwise. The lack of locality is the main reason that the computational workload grows as mk^2, where m is the dimension of the data and k is the number of iterations. This workload growth rate is large if k is proportional to m. Local linear functionals would reduce this workload. Moreover, this is a natural step to take from an estimation-theoretic point of view: the estimate at a point should depend mostly on the data taken near it.

One potential solution to this problem would be to break down the estimation problem into a set of overlapping subproblems. Specifically, points at which one is computing estimates are divided up into contiguous, disjoint regions. Each of these estimation regions is then matched to a set of measurements needed to form estimates within the estimation regions to the desired level of accuracy. Although the estimation regions are disjoint, the sets of measurements associated with them, may not be. Provided the sets of measurements are strictly proper subsets of all of the data, this procedure will force the linear functionals used in each subproblem to be local. Then, as the overall estimation problem grows, the computational workload will only grow linearly with the problem size, provided the size of the subproblems can remain fixed to achieve the desired level of accuracy. This and other possibly more efficient solutions are subjects for future research.

Statistics of Constants in the Convergence Analysis

The convergence analysis in this thesis provide bounds on the computational error of the Krylov subspace estimation problem in terms of a finite random variable η, whose statistics are not studied in this thesis. The finiteness of η ensures that these bounds decrease as the number of iterations increase. Moreover, one can compute convergence rates for a problem that are independent of η. However, there may be variability in the performance of the algorithm from run to run for a given estimation problem, depending on the initial starting vector. The degree of variability is reflected in the statistics of η. Thus, computing, or just bounding, the mean and variance would be useful. This is left as an open problem.

Matrix-Vector Products

The application of the Krylov subspace estimation algorithm to a particular estimation problem requires the development of an efficient routine for multiplying vectors by the prior covariance matrix. The development of such routines is an open research problem. When the prior covariance is stationary, one can use FFTs to implement the matrix-vector products, as was done for the results in this thesis. Moreover, if the process is stationary in local regions, one may be able to use local cosine bases to implement the matrix-vector products, using the decomposition in [53]. If the prior is truly non-stationary, developing a routine for computing matrix-vector products is much more challenging.

Preconditioning

Preconditioning can accelerate the convergence of the Krylov subspace estimation algorithm. A methodology for developing effective preconditioners is an open research topic, however. The simple preconditioners discussed in this thesis were not developed in a methodical fashion, and the performance could be much improved upon. Developing effective preconditioners is especially important for estimation problems with spatial priors since there may be many repeated or clustered eigenvalues as a consequence of the covariance having some type of invariance (*e.g.* it may be isotropic).

Data Sets

The Krylov subspace estimation algorithm can be applied to the processing of many different types of data. Determining to what types of data the algorithm can be applied is another open research topic. One type of data that the algorithm may be especially suited for processing is hyperspectral imagery [41]. Hyperspectral imagery consists of electromagnetic intensity measurements typically taken over 2-dimensional fields (such as a patch of the earth). Associated with each pixel in the image is a vector of measurements (typically in the hundreds). The elements of the measurement vector at a pixel are measurements of electromagnetic intensity taken over disjoint narrow spectral bands for a small patch in the 2-dimensional field. Processing hyperspectral imagery often requires data reduction because of the large volumes of data. Thus, the Krylov subspace estimation algorithm may be well suited to estimating quantities from hyperspectral imagery.

■ 7.2.2 Space-Time Estimation

Numbers Of Iterations for Predict and Update Steps

One of the observations acquired from repeatedly running the Krylov subspace algorithms for space-time estimation is that the total computational error tends to be dominated by the approximations made at only one of the following three recursive steps: predicting, updating, or smoothing. Thus, the Krylov subspace algorithms for filtering and smoothing may do more work than is necessary to get the desired level of computational accuracy, depending on the stopping criteria used. For example, if the Krylov subspace algorithm for Kalman filtering is run to obtain excellent prediction approximations but only moderately accurate update approximations, extra work is being done at the prediction steps with no gain in overall accuracy of the filtered estimates and error variances. Hence, one would like to know how to specify stopping criteria for each step so as to minimize the total number of iterations, summed over time and steps. Some analysis is done in Section 5.2.1. However, one would like a more complete set of guidelines.

Nonlinear Time Dynamics

Although the space-time estimation problems considered in Chapter 5 are linear, there are many space-time estimation problems that involve nonlinear dynamics. In particular, most of the problems in meteorology and oceanography have a nonlinear component. If a nonlinear space-time estimation problem can be approximately solved with an extended Kalman filter, however, one may be able to adapt the Krylov subspace techniques for space-time estimation to approximately solve the nonlinear estimation problem. Specifically, the extended Kalman filter computes a sequence of filtered estimates by solving a set of linear estimation problems arising from linearized dynamics. Each of these linear estimation problems could be approximated using the Krylov subspace techniques in Chapter 5. However, the approximation errors may grow quickly in time depending on the nonlinear dynamics in the space-time estimation problem. The feasibility of such a Krylov subspace approach is a topic for future research.

Statistics of the Noise Driving Rossby Waves

The statistics of the noise driving Rossby waves in the model of Section 5.5.2 are derived from altimeter data of sea surface height. These data capture physical phenomena occurring at a wide range of spatial and temporal scales. One would like to alter the statistics of the driving noise so that the resulting model is only accounting for Rossby waves at specified spatial and temporal scales. The result of altering the statistics should be a less complex model that would allow the Krylov subspace algorithms for space-time estimation to process more data for the same computational effort. How to develop appropriate statistics is an open question.

7.2.3 Lyapunov Theory for Infinite-Dimensional Linear Systems

Converses

Chapter 6 introduces a Lyapunov theory for infinite-dimensional linear systems. Specifically, Theorems 6.3.1 and 6.5.1 detail sufficient conditions for a linear system to be, respectively, strongly and weakly stable. Most other Lyapunov theorems in the literature have converses. That is, the conditions are not only sufficient but also necessary, under mild restrictions. Theorems 6.3.1 and 6.5.1 may also have converses. To prove these, however, one may need to restrict the notions of stability. Specifically, one may need to consider systems that are not only strongly or weakly stable but also have state trajectories, $z(t)$, that sum absolutely so that

$$\sum_{\tau} \|z(\tau)\| < \infty \tag{7.1}$$

or

$$\sum_{\tau} |\langle z(\tau), z' \rangle| < \infty \qquad \forall z', \tag{7.2}$$

respectively. Formulating detailed converses and proving them is left for future research.

Continuous-Time Theory

All of the results in Chapter 6 are for the discrete-time setting. Versions of all of these results may exist in a continuous-time setting. However, the proofs would become much more technical. In particular, the type of derivative would need to be specified. The natural derivative may be the weak one, *i.e.* the derivative $\dot{z}(t)$ of the function $z(t)$ is the vector that satisfies

$$\langle \dot{z}(t), z' \rangle = \frac{d}{dt}\langle z(t), z' \rangle \qquad \forall z'. \tag{7.3}$$

As for the converse theorems, formulating detailed continuous-time theorems, and proving them is left for future research.

Robustness of Space-Time Filters

In this thesis, the stability of space-time filters has motivated the study of general infinite-dimensional linear systems. The theory developed in this thesis establishes the stability of a class of space-time filters. A stable system, in this analysis, refers to one whose unforced state tends to zero irrespective of the initial condition. However, one is also interested in the robustness of space-time filters. Specifically, one is interested in determining whether the state, when driven by bounded perturbations, remains bounded over all time. One may be able to establish robustness using the stability theory developed in this thesis. How to do this is an open problem.

Appendix A

Proof of Convergence for the Block Krylov Subspace Estimation Algorithm

The proof of the block Krylov subspace estimation algorithm convergence result, Theorem 3.4.1, is almost the same as for the standard Krylov subspace estimation algorithm convergence result, Theorem 3.3.1. The primary difference is in the bound on $\|(I-\pi_i)u_j\|$, the norm of the residual of the projection of the eigenvector u_j onto the ith Krylov subspace. For the standard Krylov subspace estimation algorithm, the bound is given by Theorem 3.5.1. Saad generalized Theorem 3.5.1 to the block case [70, Theorem 5].

■ A.1 Angles Between Eigenspaces and Krylov Subspaces

In the generalization of Theorem 3.5.1, there is the need to define and redefine some notation. In particular, the projection operators π_i, for the block case, project onto the ith Krylov subspace, $\mathcal{K}(\Lambda_y, S, i)$, of dimension ir. The vector $\hat{u}_i$ is the vector in S such that

$$(\hat{u}_i, u_j) = \delta_{ij} \qquad \text{for } j = i, i+1, \ldots, i+r-1. \tag{A.1}$$

That is, $\hat{u}_i$ is the residual of projecting u_i onto S and orthogonalizing it against the projections of $u_{i+1}, \ldots, u_{i+r-1}$:

$$\hat{u}_i \propto \pi_1 u_i - U_i \left(U_i^T U_i\right)^{-1} \pi_1 u_i \tag{A.2}$$

where

$$U_i = \begin{pmatrix}\pi_1 u_{i+1} & \cdots & \pi_1 u_{i+r-1},\end{pmatrix} \tag{A.3}$$

and

$$\|\hat{u}_i\| = \|\pi_1 u_i\|^2 - (\pi_1 u_i)^T U_i \left(U_i^T U_i\right)^{-1} U_i^T \pi_1 u_i. \tag{A.4}$$

(Note that $\|\hat{u}_i\|$ may get very large if the projections of r consecutive eigenvectors onto S get close together.) Using this notation, one can write the generalization of Theorem 3.5.1, as follows.

Theorem A.1.1. *Let γ_r be defined by (3.51), and K_j by*

$$K_j \triangleq \begin{cases} \prod_{i \in d_j} \frac{\lambda_{y,i}-\lambda_{y,m}}{\lambda_{y,i}-\lambda_{y,j}} & \text{if } j \neq 1 \\ 1 & \text{if } j = 1 \end{cases} \tag{A.5}$$

where d_i is the set of indices of the first $j-1$ distinct eigenvalues. Then,

$$\frac{\|(I-\pi_i)\,u_j\|}{\|\pi_i u_j\|} \leq \frac{2K_j}{\gamma_r^{i-j}} \|u_i - \hat{u}_j\|\,. \tag{A.6}$$

Theorem A.1.1, like Theorem 3.5.1, provides a bound on $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$, the tangent of the angle between the Krylov subspace $\mathcal{K}(\Lambda_y, S, i)$ and the eigenvector u_j. This bound in (A.6) has three components. The rate of decay as i increases is given by γ_r. The constant in the numerator, $2K_j$, depends on the distribution of distinct eigenvalues as specified by (A.5). The third term, $\|u_i - \hat{u}_j\|$ depends on the starting matrix S, as already discussed.

■ A.2 Proof of Convergence for the Block Case

Now, the proof in Section 3.3 of Theorem 3.3.1, establishing convergence for the standard Krylov subspace estimation algorithm, extends to the block case with minimal changes. First, each of Propositions 3.5.1 and 3.5.2 holds for the block case; only the meaning of π_i has changed. Moreover, Proposition 3.5.3 also holds in the block case with $1/\|\pi_1 u_j\|^2$ replaced by $\|u_j - \hat{u}_j\|^2$. The bound on K_j in Proposition 3.5.5 also extends to the block case, but the proof is slightly different; so it is stated separately here.

Proposition A.2.1. *There exists a function $K(v)$ which is continuous and monotonically decreasing from infinity to one as v ranges from zero to infinity and satisfies*

$$K_j \leq K(\lambda_{sep}). \tag{A.7}$$

Proof.

$$\begin{aligned} \frac{1}{K_j} &= \prod_{k \in d_j} \frac{\lambda_{y,k}-\lambda_{y,j}}{\lambda_{y,k}-\lambda_{y,m}} \\ &\geq \prod_{k \in d_j} \left(1 - \left(\frac{1}{1+\lambda_{\mathsf{sep},+}}\right)^k\right) \end{aligned} \tag{A.8}$$

where the inequality follows from Condition 5 of Theorem 3.4.1. The remainder of the proof is the same as that of Proposition 3.5.5. □

Now, the inequalities in Section 3.5.3 that finish the proof of Theorem 3.3.1 also finish the proof of Theorem 3.4.1. However, μ_m must be replaced by

$$\max_{1 \leq j \leq m} \frac{1}{\gamma^j} \|u_j - \hat{u}_j\| . \tag{A.9}$$

Proposition 3.5.6 establishes an extreme value result that shows that the statistics of μ_m depend only on the eigenvalue decay rates and not on the size of the problem. No similar result exists for (A.9) since $\hat{u}_j$ is a very complicated random variable. Thus, the constant η appearing in Theorem 3.4.1 may depend on problem size resulting in a strong dependence on the starting matrix S for large problems. This was never observed in numerical results, however. Thus, a conjecture is that (A.9) is a well-behaved extremum that converges in distribution with increasing m for constant eigenvalue decay rates.

Appendix B

Orthogonalization Techniques

For each of the algorithms presented in this thesis, a Lanczos iteration, given by (3.10)-(3.13), is used to compute orthonormal bases, $\{q_1, \ldots, q_k\}$, for the Krylov subspaces $\mathcal{K}(\Lambda_y, s, k)$. Unfortunately, the Lanczos iteration has some poor numerical properties. In particular, the bases tend to lose their orthogonality as k increases in finite-precision arithmetic. A variety of schemes have been proposed for altering the Lanczos iteration so as to maintain orthogonality of the q_i [17, 18, 61, 62, 64, 65]. Full orthogonalization is discussed briefly in Sections 3.1 and 3.4.2. The presentation is expanded upon here, and a second scheme, selective orthogonalization is also put forth. These two schemes are the ones that have been used to generate the results in this thesis. The discussion of these schemes is followed by the details of a test used to detect loss of orthogonality (breakdown) in the context of the statistical applications addressed in this thesis.

■ B.1 Full Orthogonalization

The simplest orthogonalization technique is to remove components of $q_1, \ldots, q_k$ from h_k between the computations in (3.11) and (3.12). Specifically,

$$h_k := h_k - \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T h_k \tag{B.1}$$

where := denotes reassignment. The q_i, themselves, are not reorthogonalized since it is assumed that the Lanczos iteration with the added full orthogonalization step (B.1) will yield bases that remain orthogonal. The technique tends to work very well. The additional number of multiplications required by the kth step is approximately mk^2 for $q_i \in \mathbb{R}^m$.

This technique extends to the block case. The block Lanczos iteration is given by (3.43)-(3.45). Full orthogonalization consists of subtracting out components of $Q_1, \ldots, Q_k$ from H_k as follows:

$$H_k := H_k - \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix}^T H_k. \tag{B.2}$$

Full orthogonalization is the only method used in this thesis to guarantee orthogonality in the block algorithm.

Full orthogonalization is only slightly different for left- and right-preconditioned systems. The difference is a consequence of the different inner-products being used to formulate the preconditioned algorithms. For a preconditioning transformation B, B^TB inner-products are used in right-preconditioned algorithms, and $(B^TB)^{-1}$ inner-products are used in left-preconditioned algorithms. The following focuses on right-preconditioned standard Lanczos iterations; although all of the concepts extend to the left-preconditioned and block forms of the algorithm.

Recall that the right-preconditioned Lanczos iteration is given by (3.32)-(3.37). Full orthogonalization is performed between steps (3.33) and (3.34) as follows:

$$h_k := h_k - \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T d_k \tag{B.3}$$

where $d_k = B^TBh_k$. For the preconditioned algorithm, one must recompute d_k by either performing an additional orthogonalization or B^TB matrix-vector product. The recomputation is necessary because d_k is used in a subsequent Lanczos step. The additional orthogonalization would be performed as follows:

$$d_k := d_k - \begin{bmatrix} t_1 & \cdots & t_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T d_k \tag{B.4}$$

where $t_i = B^TBq_i$. Note that this second orthogonalization requires an additional $mk^2/2$ multiplies at step k for $q_i \in \mathbb{R}^m$ as compared to $\mu_p mk$ multiplies for an additional matrix-vector product. The more efficient approach depends on μ_p and the number of iterations required. If k grows with the problem size, however, computing the additional matrix-vector product is asymptotically more efficient; so this approach is assumed when calculating computational complexity in the thesis.

Thus, one can use full orthogonalization to guarantee numerical stability of the Lanczos iteration and any of the preconditioned block variants.

B.2 Selective Orthogonalization

One may not need to orthogonalize the h_k against all of the q_i for $i = 1, \ldots, k$. In exact arithmetic, the h_k would be orthogonal to each of the q_i, and in finite-precision arithmetic, the h_k are orthogonal to a significant subspace of the span of the q_i. This subspace is specified by the following theorem due to Paige [64, Section 13-4].

Theorem B.2.1. *At step k of the standard Lanczos iteration (without orthogonalization) given by (3.10)-(3.13), let $T_k = Q_k^T \Lambda_y Q_k$ be the tri-diagonalization of Λ_y, and let s_i be the eigenvectors of T_k. Then,*

$$(Q_k s_i)^T q_{k+1} \leq \frac{g_i}{\beta_{ki}} \tag{B.5}$$

where $\beta_{ki} = \beta_k s_{ki}$ and g_i is a floating point error.

Theorem B.2.1 provides bounds on the inner products between the next Lanczos vector q_{k+1} and vectors in the span of the columns of Q_k in terms of the computable quantities β_{ki} and an unknown floating point error g_i. Although there exist bounds on the g_i, Parlett notes that these bounds are not tight but that typically g_i is of the order of $\varepsilon_{\text{float}}\|\Lambda_y\|$ where $\varepsilon_{\text{float}}$ is the floating point precision for the computations [64, p. 268].

One can use Theorem B.2.1 to determine those linear combinations of the q_i that need to be removed from $h_k \propto q_{k+1}$ at each step. Specifically, let

$$O_k = \{i \in \{1, \ldots, k\} : |\beta_{ki}| < \varepsilon_{\text{orth}}\sqrt{\varepsilon_{\text{float}}}\theta_{\max}\} \tag{B.6}$$

where β_{ki} and $\varepsilon_{\text{float}}$ are as in Theorem B.2.1, $\theta_{\max}$ is the maximum eigenvalue of T_k and serves as an approximation to $\|\Lambda_y\|$, and $\varepsilon_{\text{orth}}$ is a parameter of the algorithm. Then, O_k indexes linear combinations $Q_k s_i$ that need to be removed from h_k between steps (3.11) and (3.12):

$$h_k := h_k - \sum_{i \in O_k} (Q_k s_i)(Q_k s_i)^T h_k. \tag{B.7}$$

One can also use selective orthogonalization with the preconditioned Lanczos iterations. In the case of right preconditioning, this amounts to deflating h_k between steps (3.33) and (3.34) by

$$h_k := h_k - \sum_{i \in O_k} (Q_k s_i)(Q_k s_i)^T d_k \tag{B.8}$$

where O_k is the same as in (B.6); although, $\theta_{\max}$ now serves as an approximation to $\|\Lambda_y\|_{B^TB}$. The validity of this technique follows from a straightforward extension of Theorem B.2.1, as follows.

Theorem B.2.2. *At step k of the preconditioned Lanczos iteration (without orthogonalization) given by (3.32)-(3.37), let $T_k = Q_k^T B^T B \Lambda_y B^T B Q_k$ be the tri-diagonalization of $B^T B \Lambda_y B^T B$, and let s_i be the eigenvectors of T_k. Then,*

$$(Q_k s_i)^T B^T B q_{k+1} \leq \frac{g_i}{\beta_{ki}} \tag{B.9}$$

where $\beta_{ki} = \beta_k s_{ki}$ and g_i is a floating point error.

As in Theorem B.2.1, the bound in Theorem B.2.2 is in terms of a computable quantity, β_{ki} and an unknown floating point error. Although no extensive study of the properties of g_i has been made, g_i is conjectured to be of the order of $\varepsilon_{\text{float}}\|\Lambda_y\|_{B^TB}$ in analogy to the assumptions concerning floating point error for the standard Lanczos iteration. Also note that Theorem B.2.2 is specific to right preconditioning, but a similar result holds for left preconditioning.

Selective orthogonalization is particularly advantageous when convergence of the Krylov subspace algorithms is slow. This follows from the fact that, in exact arithmetic for the standard Lanczos iteration,

$$\|\Lambda_y Q_k s_i - \theta_i Q_k s_i\| = |\beta_{ki}| \tag{B.10}$$

where θ_i is the ith eigenvalue of T_k [64, p. 266]. In other words, a newly computed h_k will not be orthogonal to large fractions of the kth Krylov subspace if and only if almost k eigenvectors of Λ_y are well-approximated in that Krylov subspace. The latter holds when convergence is rapid. Thus, selective orthogonalization offers advantages when convergence is slow. The same is true for the preconditioned algorithms since (B.10) generalizes. In particular, for the right-preconditioned algorithm,

$$\left\|\Lambda_y \left(B^T B\right) Q_k s_i - \theta_i Q_k s_i\right\|_{B^T B} = \left|\beta_{ki}\right|. \tag{B.11}$$

Although selective orthogonalization can be very advantageous, it also has a disadvantage. Namely, the q_i may lose orthogonality after many iterations because the orthogonalization is not being done precisely enough. This is usually not an issue, though, when striving for only moderately low levels of error, on the order of a few per cent.

■ B.3 Breakdown Test

The Lanczos iteration may breakdown even if an orthogonalization scheme is used. This is especially true in the estimation setting when the Krylov subspaces are generated by the covariance of the data Λ_y, which is often a shifted matrix, *i.e.* of the form $\Lambda_y = \Lambda_z + \sigma^2 I$ for some covariance Λ_z. When the eigenvalues of the signal covariance, Λ_z, fall off rapidly, Λ_z may be effectively low-rank in finite-precision arithmetic. In particular, the Krylov subspaces generated by Λ_y may effectively include the column space of Λ_z at some Krylov subspace dimension $k \ll m$. If the Lanczos iteration is continued past this point, the algorithm will generate results dominated by roundoff error. Thus, one must use a test to determine when this situation has occurred.

A simple test used for the results in this thesis can be described as follows. Recall that for the standard Lanczos recursion, $\Lambda_y Q_k = Q_k T_k$ where T_k is the tri-diagonal matrix,

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix} \tag{B.12}$$

whose entries are given by (3.10)-(3.13). Now, the column space of Λ_z is contained in the kth Krylov subspace if $\Lambda_y q_k \propto q_k$, *i.e.* if $\beta_k \approx 0$ relative to the norm of T_k. Thus, the Krylov subspace algorithms for estimation presented in this thesis are terminated if

$$\frac{\beta_{k+1}}{\theta_{\max}} < \varepsilon_{\text{breakdown}} \varepsilon_{\text{float}} \tag{B.13}$$

where θ_{max} is the maximum eigenvalue of T_k, $\varepsilon_{\text{float}}$ is the floating point precision, and $\varepsilon_{\text{breakdown}}$ is a parameter of the algorithm. The same test can be used for the preconditioned and block (with $\|R_{k+1}\|$ replacing β_{k+1}) forms. For all of the results in this thesis, $\varepsilon_{\text{breakdown}}$ was set to 10.

■ B.4 Summary

Both full and selective orthogonalization were used to generate results in this thesis. However, selective orthogonalization for $\varepsilon_{\text{orth}} = 200$ was used for most results. Unless otherwise noted, this was the technique used. The breakdown test of the previous section for $\varepsilon_{\text{breakdown}} = 10$ was employed throughout the thesis.

Bibliography

[1] R. J. Barton and H. V. Poor. Signal detection in fractional Gaussian noise. *IEEE Transactions on Information Theory*, 34(5):943–959, September 1988.

[2] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky. Modeling and estimation of multiresolution stochastic processes. *IEEE Transactions on Information Theory*, 38(2):766–784, March 1992.

[3] A. F. Bennett. *Inverse Methods in Physical Oceanography.* Cambridge University Press, New York, 1992.

[4] A. F. Bennett. Inverse methods and data assimilation. Lecture Notes from the 1999 Summer School at the College of Oceanic and Atmospheric Sciences of Oregon State University, 1999.

[5] A. F. Bennett, B. S. Chua, and L. M. Leslie. Generalized inversion of a global numerical weather prediction model. *Meteorology and Atmospheric Physics*, 60:165–178, 1996.

[6] A. F. Bennett, B. S. Chua, and L. M. Leslie. Generalized inversion of a global numerical weather prediction model II: Analysis and implementation. *Meteorology and Atmospheric Physics*, 62:129–140, 1997.

[7] A. Bensoussan, G. DaPrato, M. C. Delfour, and S. K. Mitter. *Representation and Control of Infinite Dimensional Systems.* Birkhäuser, Boston, 1992.

[8] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–236, 1974.

[9] G. J. Bierman. A comparison of discrete linear filtering algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, AES-9(1):28–36, 1973.

[10] G. J. Bierman. Fixed interval smoothing with discrete measurements. *International Journal of Control*, 18(1):65–75, 1973.

[11] G. J. Bierman. Sequential square root filtering and smoothing of discrete linear systems. *Automatica*, 10:147–158, 1974.

[12] C. Cabos. Evaluation of matrix functions with the block Lanczos algorithm. *Computers and Mathematics with Applications*, 33(1/2):45–57, 1997.

[13] D. B. Chelton, R. A. DeSzoeke, M. G. Schlax, K. El Naggar, and N. Siwertz. Geographical variability of the first baroclinic Rossby radius of deformation. *Journal of Physical Oceanography*, 28:433–460, March 1998.

[14] D. B. Chelton and M. G. Schlax. Global observations of oceanic Rossby waves. *Science*, 272:234–238, 12 April 1996.

[15] K. Chou. *A Stochastic Modeling Approach to Multiscale Signal Processing*. PhD thesis, MIT, May 1991.

[16] S. E. Cohn and R. Todling. Approximate data assimilation schemes for stable and unstable dynamics. *Journal of the Meteorological Society of Japan*, 74(1):63–75, 1996.

[17] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Programs*, volume 2. Birkhäuser, Boston, 1985.

[18] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Theory*, volume 1. Birkhäuser, Boston, 1985.

[19] A. da Silva and J. Guo. Documentation of the Physical-space Statistical Analysis System (PSAS) part I: The conjugate gradient solver version PSAS-1.00. DAO Office Note 96-02, Data Assimilation Office, Goddard Laboratory for Atmospheres, NASA, February 1996. ftp://dao.gsfc.nasa.gov/pub/office_notes/on9602.ps.Z.

[20] R. Datko. An extension of a theorem of A. M. Lyapunov to semi-groups of operators. *Journal of Mathematical Analysis and Applications*, 24:290–295, 1968.

[21] R. Datko. Extending a theorem of A. M. Liapunov to Hilbert space. *Journal of Mathematical Analysis and Applications*, 32:610–616, 1970.

[22] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[23] C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal of Scientific Computation*, 18(4):1088–1107, July 1997.

[24] P. Van Dooren and M. Verhagen. Condensed forms for efficient time-invariant Kalman filtering. *SIAM Journal on Scientific and Statistical Computing*, 9(3):516–530, May 1988.

[25] V. Druskin and L. Knizhnerman. Extended Krylov subspaces: Approximation of the matrix square root and related functions. *SIAM Journal on Matrix Analysis and Applications*, 19(3):755–771, July 1998.

[26] R. M. Dudley. *Real Analysis and Probability.* Chapman & Hall, New York, 1989.

[27] C.K. Shum et al. Accuracy assesment of recent ocean tide models. *Journal of Geophysical Research*, 102(C11):25173–25194, November 15 1997.

[28] P. Fieguth, D. Menemenlis, T. Ho, A. Willsky, and C. Wunsch. Mapping Mediterranean altimeter data with a multiresolution optimal interpolation algorithm. *Journal of Atmospheric and Oceanic Technology*, 15:535–546, April 1998.

[29] P. W. Fieguth, M. R. Allen, and M. J. Murray. Hierarchical methods for global-scale estimation problems. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pages 161–164, May 1998.

[30] A. B. Frakt. *Internal Multiscale Autoregressive Processes, Stochastic Realization, and Covariance Extension.* PhD thesis, MIT, August 1999.

[31] P. Gaspar and C. Wunsch. Estimates from altimeter data of barotropic Rossby waves. *Journal of Physical Oceanography*, 19(12):1821–1844, December 1989.

[32] A. George. *Computer Solution Of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, N.J., 1981.

[33] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1996.

[34] W. W. Hager and L. L. Horowitz. Convergence and stability properties of the discrete Riccati operator equation and the associated optimal control and filtering problems. *SIAM Journal of Control and Optimization*, 14(2):295–312, February 1976.

[35] W. Hahn. *Theory and Application of Liapunov's Direct Method.* Prentice-Hall, Englewood Cliffs, 1963.

[36] T. Ho. *Multiscale Modeling and Estimation of Large-Scale Dynamics Systems.* PhD thesis, MIT, September 1998.

[37] T. T. Ho, P. W. Fieguth, and A. S. Willsky. Multiresolution stochastic models for the efficient solution of large-scale space-time estimation problems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3097–3100, 1996.

[38] T. T. Ho, P. W. Fieguth, and A. S. Willsky. Computationally efficient multiscale estimation of large-scale dynamic systems. In *International Conference on Image Processing*, pages 274–278, 1998.

[39] T. T. Ho, P. W. Fieguth, and A. S. Willsky. Recursive multiscale estimation of space-time random fields. In *International Conference on Image Processing*, pages 867–870, 1999.

[40] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[41] A. Ifarraguerri and C. I. Chang. Unsupervised hyperspectral image analysis with projection pursuit. *IEEE Transactions on Geoscience in Remote Sensing*, 38(6):2529–2538, November 2000.

[42] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM Journal on Numerical Analysis*, 31(1):227–251, 1994.

[43] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[44] A. G. Journel and C. T. Huijbregts. *Mining Geostatistics.* Academic Press, New York, 1978.

[45] J. J. Deyst Jr. Correction to 'conditions for asymptotic stability of the discrete minimum-variance estimator'. *IEEE Transactions on Automatic Control*, AC-18(5):562–563, October 1973.

[46] J. J. Deyst Jr and C. F. Price. Conditions for asymptotic stability of the discrete minimum-variance linear estimator. *IEEE Transactions on Automatic Control*, AC-13:702–705, December 1968.

[47] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "second method" of Lyapunov i: Continuous-time systems. *Transactions of the ASME: Journal of Basic Engineering*, 82:371–393, 1960.

[48] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "second method" of Lyapunov ii: Discrete-time systems. *Transactions of the ASME: Journal of Basic Engineering*, 82:394–400, 1960.

[49] P. G. Kaminski and Arthur E. Bryson Jr. Discrete square root smoothing. In *AIAA Guidance and Control Conference*, August 14-16 1972. AIAA Paper No. 72-877.

[50] C. King, D. Stammer, and C. Wunsch. The CMPO/MIT TOPEX/POSEIDON altimetric data set. Technical Report 30, MIT, 1994.

[51] M. R. Leadbetter, G. Lindgren, and H. Rootzen. *Extremes and related properties of random sequences and processes.* Springer-Verlag, New York, 1983.

[52] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wilely & Sons, Inc., 1969.

[53] S. Mallat, G. Papanicolau, and Z. Zhang. Adaptive covariance estimation of locally stationary processes. *Annals of Statistics*, 26(1):1–47, February 1998.

[54] B. B. Mandelbrot and J. W. Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Review*, 10(4), October 1968.

[55] J. L. Massera. Contributions to stability theory. *Annals of Mathematics*, 64(1):182–206, July 1956.

[56] J. L. Massera and J. J. Schäffer. Linear differential equations and functional analysis I. *Annals of Mathematics*, 67(3):517–573, May 1958.

[57] J. L. Massera and J. J. Schäffer. Linear differential equations and functional analysis II: Equations with periodic coefficients. *Annals of Mathematics*, 69(1):88–104, January 1959.

[58] J. L. Massera and J. J. Schäffer. Linear differential equations and functional analysis III: Lyapunov's second method in the case of conditional stability. *Annals of Mathematics*, 69(3):535–574, May 1959.

[59] J. L. Massera and J. J. Schäffer. Linear differential equatioins and functional analysis IV. *Mathematische Annalen*, 139:287–342, 1960.

[60] L. Massera. Erratum: Contributions to stability theory. *Annals of Mathematics*, 28(2):202, September 1958.

[61] C. C. Paige. Computational variants of the Lanczos method for the eigenproblem. *Journal of the Institute for Mathematics and its Applications*, 10:373–381, 1972.

[62] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *Journal of the Institute of Mathematics and its Applications*, 18:341–349, 1976.

[63] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[64] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood-Cliffs, 1980.

[65] B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation*, 33(145):217–238, January 1979.

[66] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., San Mateo, 1988.

[67] J. Pedlosky. *Geophysical Fluid Dynamics.* Springer, New York, 1987.

[68] A. P. Ross. Stability and stabilizability of infinite dimensional linear systems via Liapunov's direct method. In *24th Midwest Symposium on Circuits and Systems*, pages 579–582, Albuquerque, June 29-30 1981.

[69] W. Rudin. *Functional Analysis.* McGraw-Hill, 2nd edition, 1991.

[70] Y. Saad. On the rates of convergence of the Lanczos and the block-Lanczos method. *SIAM Journal of Numerical Analysis*, 17(5):687–706, October 1980.

[71] Y. Saad. *Numerical methods for large eigenvalue problems.* Manchester University Press, Manchester, 1992.

[72] D. Stammer. Globa characteristics of ocean variability estimated from regional TOPEX/ POSEIDON altimeter measurements. *Journal of Physical Oceanography*, 27:1743–1769, August 1997.

[73] D. Stammer and C. Wunsch. The determination of the large-scale circulation of the Pacific Ocean from satellite altimetry using model Green's functions. *Journal of Geophysical Research*, 101(C8):18409–18432, August 15 1996.

[74] R. Todling and S. E. Cohn. Suboptimal schemes for atmosperic data assimilation based on the Kalman filter. *Monthly Weather Review*, 122:2530–2557, November 1994.

[75] R. Todling, S. E. Cohn, and N. S. Sivakumaran. Suboptimal schemes for retrospective data assimilation based on the fixed-lag Kalman smoother. *Monthly Weather Review*, 126:2274–2286, August 1998.

[76] H. A. Van Der Vorst. An iterative solution method for solving $f(a)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix a. *Journal of Computational and Applied Mathematics*, 18:249–263, 1987.

[77] Irving W. W. *Multiscale Stochastic Realization and Model Identification with Applications to Large-Scale Estimation Problems.* PhD thesis, MIT, September 1995.

[78] R. L. Wheeden and A. Zygmund. *Measure and Integral.* Marcel Dekker, New York, 1977.

[79] G. Xu, Y. Cho, and T. Kailath. Application of fast subspace decomposition to signal processing and communication problems. *IEEE Transactions on Signal Processing*, 42(6):1453–1461, June 1994.

[80] G. Xu and T. Kailath. Fast estimation of principal eigenspace using Lanczos algorithm. *SIAM Journal on Matrix Analysis and Applications*, 15(3):974–994, July 1994.

[81] G. Xu and T. Kailath. Fast subspace decomposition. *IEEE Transactions on Signal Processing*, 42(3):539–551, March 1994.

[82] G. Xu, H. Zha, G. Golub, and T. Kailath. Fast algorithms for updating signal subspaces. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 41(8):537–549, August 1994.