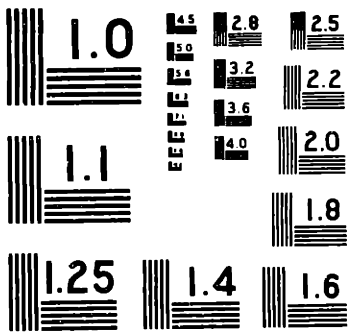
1997



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963

Multiresolution Statistical Modeling with Application to Modeling Groundwater Flow

by

Michael M. Daniel

B.S. Electrical Engineering and Computer Science
University of California at Berkeley, 1990

S.M. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1993

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 1997

Signature of Author: _____
Dept. of Electrical Engineering and Computer Science
January 29, 1997

Certified by: _____
Alan S. Willsky
Professor of EECS
Thesis Supervisor

Accepted by: _____
Arthur C. Smith
Professor of EECS
Chair, Committee for Graduate Students

Multiresolution Statistical Modeling with Application to Modeling Groundwater Flow

by

Michael M. Daniel

Submitted to the Department of Electrical Engineering
and Computer Science
on January 31, 1997 in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy
in Electrical Engineering and Computer Science

Abstract

The development of accurate mathematical models describing the flow of groundwater is an important problem due to the prevalence of contaminants in or near groundwater supplies. An important parameter of these models is hydraulic conductivity, which describes the ability of the subsurface geology to conduct water flow. Because hydraulic conductivity is a function of the earth's subsurface, direct measurements can be made only at a relatively small number of locations. Instead, one must rely on indirect measurement sources, which supply observations of conductivity at different locations and resolutions. An important problem is to estimate the hydraulic conductivity function from all available data, and to characterize the remaining uncertainty. The class of multiscale processes introduced in [Chou et al., 1994] appears to be well-suited to this problem, since these processes can be estimated efficiently at every scale at which they are modeled; the multiscale estimator also provides an uncertainty measure for the estimates. However, this multiscale framework has some limitations that must be overcome before it can be applied to general data fusion problems. First, because all of the previous applications have focused on the measurement and estimation of the finest-scale process, arbitrary nonlocal properties of interest (e.g., coarse-resolution measurements of hydraulic conductivity) have not been represented within the multiscale framework. Second, the class of stochastic processes that is well modeled by low-order tree models has not been fully characterized. To address the first limitation, this thesis (1) extends multiscale realization theory so that coarse-scale variables can represent particular nonlocal properties to be measured or estimated and (2) applies these realization algorithms to the estimation of hydraulic conductivity from sparse measurements made at different resolutions. To partially address the second limitation, the multiscale processes are shown to provide natural approximations of fractional Brownian motion. These approximations are based on the observation that the multiscale realization problem can be considerably simplified when modeling random processes that are statistically self-similar and/or have stationary increments.

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

Contents

Acknowledgments

> As this wave from memories flows in, the city soaks it up like sponge and expands. A description of Zaira as it is today should contain all Zaira's past. The city, however, does not tell its past, but contains it like the lines of a hand, written in the corners of the streets, the gratings of the windows, the banisters of the steps, the antennae of the lightning rods, the poles of the flags, every segment marked in turn with scratches, indentations, scrolls.
>
> *Marco Polo describing the City of Zaira*
> *Italo Calvino*
> *Invisible Cities*

Roughed up in qualifying exams, blessed by great friendships, aged in years, nicked by mediocre research, published, and perished, I too have gained a few scratches, lines, and indentations after too many years at MIT. I will now try thank those who have provided me the memories that I will take away from here. Whether or not my accomplishments would have been possible without these people is irrelevant. I only know that the experience was made more pleasurable, and certainly less stressful, than it undoubtedly could have been.

First of all, I thank my parents, who provided unselfishly for me, visited me on numerous occasions, and accepted my many years in graduate school with stoicism. (Yes, Mom, I now plan to visit L.A. more frequently.) Any success I had in the past or will have in the future is undoubtedly their handiwork; any consistent failures we can chalk up to bad genes. I also thank my new family members—Dad and Mom Jaggi, Anju, and Preeti—for providing moral support and a incredibly good time in Toledo last September.

I had the pleasure of working very closely and becoming friends with two MIT faculty members, Alan Willsky and Dennis McLaughlin. All of the interesting work in this thesis is either directly or indirectly their contribution and my polished regurgitation. I thank Alan, in particular, for providing me with funding, travel to foreign lands, mentorship, and loads of patience. (I never missed another meeting.) I hope our collaboration continues beyond my graduation.

I also must thank David Rossi and Schlumberger-Doll Research for providing me with three summers of refuge from Building 35. Dave is a pleasure to work with, provided me with many interesting problems, applied very little pressure, but unfortunately never learned to play soccer.

Thanks to Sanjoy Mitter for always taking time to talk with me, for providing useful advice, and for reading this voluble tome.

These acknowledgments would be incomplete without thanking Clem Karl and Mitch Trott for tirelessly supporting the computer systems on which this work was

done. Their hard work undoubtedly allowed me to accomplish much more ... or at least gave me more free time to play with Netscape.

I must also thank the members of the Stochastic Systems Group that I have been fortunate enough to share offices and bad jokes with. (In no particular order) Paul for great laughs and unforgettable self-psycho-analyses. Eric and Marc for showing me the ropes as my first office-mates. Bill for providing the foundation of work that allowed me to graduate. Big Dog Fosgate for losing every bet—Refuse to Win—and playing soft in the post. Mike for the most entertaining door at MIT. Mickey for getting my foot in the door. Rachel for many many favors. Ben for getting my coffee each day, cream, no sugar. AJ for supporting USC through the "good years". Ilya for answering every question with a question. Cedric for "will it ever end" (it appears it has), and for the four food groups (Salt, Sugar, Fat, and Caffeine). Hot for answering every question with a question (C-T-C-T-C-T...). Austin for providing many useful suggestions regarding my thesis writing. Hamid for no longer asking me if I'm happy. Jun for providing post-doctoral advice. Dewey for competing courageously with Cedric in the bad diet contest, but losing after taking a bite from an apple this week. I have also been fortunate to become good friends with other LIDS members, including Mike Branicky, Stefano Casadei, Francesca Villoresi, Sekhar Tatikonda, and Steve Patek.

Thanks to my fellow cesspool colleagues, John Buck and Andy Singer, for providing many lively conversations and helping to produce a great book [8]. Will anyone buy it? Who cares.

Thanks to all the members of the Aero-Astro Soccer team. This last year was especially enjoyable. Thanks also to Hydros Water Polo players; I'll really miss those near drowning experiences.

Thanks to Nella and Emily for keeping the place clean and providing many friendly conversations.

Thanks to the Cambex Corporation for rescuing me from MIT for two summers when I really needed it.

Thanks to Joan Goody for providing me with reasonable rent in the best part of town.

And finally to my wife, Seema, who allows me to look back on this whole experience and know it was worth it.

Notational Conventions

Symbol	Definition
General Notation	
$a:b$	the row vector $[a, a+1, \ldots, b-1, b]$
U^T	the transpose of the matrix U
$U(i,j)$	the element in the i-th row and j-th column of U
$U(a:b, c:d)$	the $(b-a+1)$-by-$(d-c+1)$ submatrix of U composed of rows a through b and columns c through d of U
$U(a:b, :)$	the $(b-a+1)$-row submatrix of U composed of rows a through b and all columns of U
$U(:, c:d)$	the $(d-c+1)$-column submatrix of U composed of all rows and columns c through d of U
$\lvert f \rvert$	a vector (function) obtained by taking the absolute value of the vector (function) f
$\lVert f \rVert$	the 2-norm of the function or vector f
$\langle g, f \rangle$	the inner-product of vectors (functions) g and f
δg	an infinitessimal perturbation of the function g
$\frac{\partial \mathcal{F}(g(x))}{\partial g}$	the Fréchét derivative of the functional operator $\mathcal{F}$ with respect to the funtion $g(x)$
∇	the gradient or Jacobian operator
$\nabla\cdot$	the divergence operator
$f \perp y$	f and y are uncorrelated, i.e., $E[(f-m_f)(y-m_y)^T] = 0$
$f \stackrel{\mathcal{P}}{=} g$	f and g have identical probability distributions
$f \sim \mathcal{N}(m_f, P_f)$	the vector f is normally distributed with mean m_f and covariance P_f
$f \sim (m_f, P_f)$	the vector f is has mean m_f and covariance P_f
$e(f, \hat{f})$	the error of the estimate $\hat{f}$, i.e., $e = f - \hat{f}$
$E[f]$	the expected value of f
$E[f \mid y]$	the expected value of f conditioned upon y
$\hat{E}[f \mid y]$	the LLSE estimate of f from y
$\hat{f}$	the estimate of f, which in most contexts is the LLSE estimate
m_f	the expected value (mean) of f, $E[f]$

Symbol	Definition
General Notation (cont ...)	
M	the dimension of the measurement vector
N	the dimension of the vector to be estimated
$\mathcal{O}(m^p)$	denotes that a performance measure is asymptotically bounded by a polynomial of order p
$\mathcal{O}(m^p n^q)$	denotes that a performance measure is asymptotically bounded by a bivariate polynomial of order p in m and q in n
$p_f(F)$	the probability density function for the random vector f
$p_{f\|y}(F \mid Y)$	the probability density function for the random vector f conditioned upon y
$p_{f,y}(F, Y)$	the joint probability density function for the random vector f and y
P_f	the covariance of f, i.e., $E[(f - m_f)(f - m_f)^T]$
$P_{f\|y}$	the covariance of f conditioned upon y, i.e., $E[(f - m_f)(f - m_f)^T \mid y]$
P_{fy}	the cross-covariance of f and y, i.e., $E[(f - m_f)(y - m_y)^T]$
R	the covariance of the measurement noise
v	the measurement noise
y	the measurement vector
Groundwater Hydrology	
f	log hydraulic conductivity: $\ln(K)$
h	hydraulic head (m)
K	hydraulic conductivity (m/s)
n	effective porosity
q	specific dischage vector (m/s)
Q	rate of volumetric water inputs per unit volume (s^{-1})
T	transmissivity (m^2/s)
t	time in seconds, travel time in seconds
t_{cp}	travel time to a control plane
v	the velocity field (m/s)
x	the spatial coordinate
$(\mathbf{x}_1, \mathbf{x}_2)$	the spatial coordinates in for two dimensional flow

Symbol	Definition
Multiscale Models	
A_s, Q_s	the auto-regression parameter and process noise covariance for the transition from node $s\overline{\gamma}$ to node s
$d(s)$	the dimension of the state at node s
f_s	the set of finest-scale elements descending from node s
f_s^c	the set of fintest-scale elements which does not descend from node s
$f_{s\alpha_{q+1}}$	f_s^c
$F_s, \overline{Q}_s$	the auto-regression parameter and process noise covariance for the backwards model
$m(s)$	scale, i.e., the distance from the root node to node s
$P_{z(s)}$	the covariance of $z(s)$
q_s	the number of children descending from node s
s	the node index for a tree process
$s\alpha_i$	the i-th child of node s, $i = 1, \ldots, q_s$
$s\overline{\gamma}$	the parent of node s
$s \wedge t$	the common ancestor of s and t that is at the finest scale
S_s	the set of nodes descendent from and including node s
S_s^c	the complement of S_s
τ_i	node at which the nonlocal function $g_i^T f$ is represented
$z(0)$	the state at the root node of the tree
$z(s)$	the state at node s
$\hat{z}(s)$	the LLSE estimate of $z(s)$

List of Figures

Chapter 1

Introduction

This thesis focuses on the application of the multiscale statistical framework introduced in [19, 20] to the estimation of random processes measured by different instruments. In particular, the framework is applied to problems in which the measurement data may be of very different types and, in particular, convey information about the random phenomenon at very different scales. One such problem is determining the parameters of partial differential equations (PDEs) that describe the flow of groundwater in the earth's subsurface. Perhaps the parameter that most significantly influences the flow of groundwater is hydraulic conductivity, which relates pressure differentials to flow rates. Because hydraulic conductivity is a property of the earth's subsurface, it can be measured directly only in a small number of locations. Instead, one must rely on indirect measurement sources, each supplying observations of conductivity at different locations and resolutions. The application of the multiscale framework to such a data fusion problem, however, is not straightforward. The class of multiscale random processes defined in [19] have been successfully applied to a problems in remote sensing [34], image processing and analysis [33, 60, 63], and parameter estimation [35]. But for all of these problems, both the measurements and the variables to be estimated are at the finest scale of the multiresolution process. The coarser scale variables only ensure that the multiresolution process has the statistical structure that allows for efficient processing algorithms. If the multiscale framework is to be applied to data fusion problems that involve the measurement and/or estimation of nonlocal functions of the phenomenon of interest, the existing tools for the realization of multiscale stochastic processes will have to be extended. Two of the main contributions of this thesis are

- extending multiscale realization theory so that the contents of coarse-scale variables can be specified, and
- applying these realization algorithms to the estimation of hydraulic conductivity and travel times associated with advective flow.

The extensions to multiscale realization theory are in many significant ways derived from and motivated by results and realization algorithms in [51]. The application of the multiresolution framework to hydraulic conductivity estimation is a direct extension of the methods in [69].

A third contribution, somewhat disjoint from the first two, is approximating fractional Brownian motion (fBm) with multiresolution stochastic processes. Two of these approximations are based on the random midpoint displacement and wavelet methods for synthesizing fBm [36], and the resulting multiscale models are direct extensions of the models provided in [62] and [35]. A more general approach is also provided, which allows for approximating fBm to arbitrary fidelity. These approximations are derived from the statistical self-similarity and stationary increments of fBm.

In the following section, the three major problems motivating this thesis are covered in more detail. In Section 1.2, the contents and contributions of each chapter are summarized.

1.1 Problems Addressed

1.1.1 Data Fusion using Multiscale Models

The primary subject of this thesis is to develop methods for the assimilation of data from different measurement sources that supply information about the phenomenon of interest at different resolutions and locations. Problems requiring the estimation of random processes or random fields from measurement data of very different types arise in a variety of contexts. Two notable areas are remote sensing and geophysical applications, for which spatially distributed random fields are to be estimated for a variety of purposes ranging from the simple production of maps of quantities like rainfall distributions to the estimation of spatial quantities to be used in the analysis of complex geophysical processes like ocean currents and subsurface fluid flow.

Geophysical phenomena such as these are typically not accessible to dense, uniform measurement, and one generally must rely on a variety of measurement sources of very different types in order to obtain enough spatial coverage to produce reliable estimates. Furthermore, while some of these measurements may be taken at individual points in the field—e.g., rain gauges, ocean measurements from ships, measurements of subsurface properties in boreholes—these measurements are typically sparse, irregularly sampled, and inadequate by themselves. Consequently, they must be fused with measurements that are indirect and provide nonlocal measurements of the phenomenon of interest over areas that are not adequately covered by the localized point measurements. These indirect observations are usually of varying resolution. An example sensor fusion problem with multiresolution measurements is the estimation of precipitation, which is used for numerical weather prediction (NWP). Precipitation can be measured with rain gauges, radar sensors, and microwave and infrared satellites. The rain gauges provide point samples of precipitation at select locations, while the infrared satellites provide broad but coarse resolution coverage. Climatologists have long recognized that no single measurement source is sufficient for reliable precipitation estimates, and instead all measurements must be incorporated [46, 81]. Another geophysical system requiring the assimilation of heterogeneous measurements is the analysis of ocean currents. Ocean currents are measured with a variety of sensors, including floating buoys, acous-

tic travel times, satellite altimetry, and direct and indirect observations of temperature and salinity. While the floating buoys can observe fine-scale fluctuations in the ocean currents, their coverage is limited. More comprehensive coverage, albeit at a coarser resolution and limited to the ocean surface, is given by the satellite data. How to fuse the many different measurements in order to produce the most reliable descriptions of ocean currents is a very active research topic [41].

The application considered in this thesis is the estimation of hydraulic conductivity for characterizing groundwater flow. Accurately describing the flow of fluids in the earth's subsurface is important due to the prevalence of contaminated soils in or near groundwater supplies. An accurate description of groundwater flow requires an accurate description of hydraulic conductivity, which is a property of the subsurface geology known to be an important determinant of groundwater flow. Geologic properties like hydraulic conductivity can be measured directly only at select well locations. Indirect observations are supplied by tracer travel times [47], pump tests [9, 29, 53, 66, 75, 76] acoustic wave propagation (seismics) [24, 47, 86], and measurements of fluid properties like hydraulic head [1, 23, 74]. These observations differ in spatial resolution and support, and each is related to hydraulic conductivity by a physical equation, i.e., a PDE. As illustrated in Chapter 3, point samples of hydraulic head are essentially observations of a coarse-scale derivative of hydraulic conductivity and are nonlocal in the sense that each head sample is sensitive to the entire conductivity field to be estimated. Again, no single measurement source can provide a reliable estimate of hydraulic conductivity, and all available measurements should be used [23, 47, 68].

A fundamental problem is to develop methods for the fusion of such disparate measurement sources, a difficult problem given the nonlocal nature of at least some of the measurement data. Moreover, there are several other features of such geophysical problems that add to the challenge. First, and most importantly, the functions to be estimated are multidimensional, requiring either computationally efficient algorithms or descriptions in terms of a manageable number of parameters. Secondly, there are often very strong reasons to think about describing phenomena at multiple scales, both because the underlying phenomena in these applications exhibit variability over wide ranges of scales and because the available data may support statistically meaningful estimation at different resolutions in different regions, depending on the coverage and nature of the available measurements. Thirdly, there is generally a strong need for the computation not only of estimates of phenomena but also of error variances for these estimates so that their significance can be assessed.

A variety of methods for fusing measurements in such contexts have been used over the years (see [69] for a review of many of these), but it is fair to say that computational complexity, especially if error variances are desired, remains a significant and limiting challenge. Several other researchers have attempted to make use of the multiscale nature of the problem by using wavelet decompositions in order to overcome computational limitations, for example [7, 18, 58, 72]. However these efforts do not address all of the issues of interest here as they either focus on using wavelets to obtain estimates but

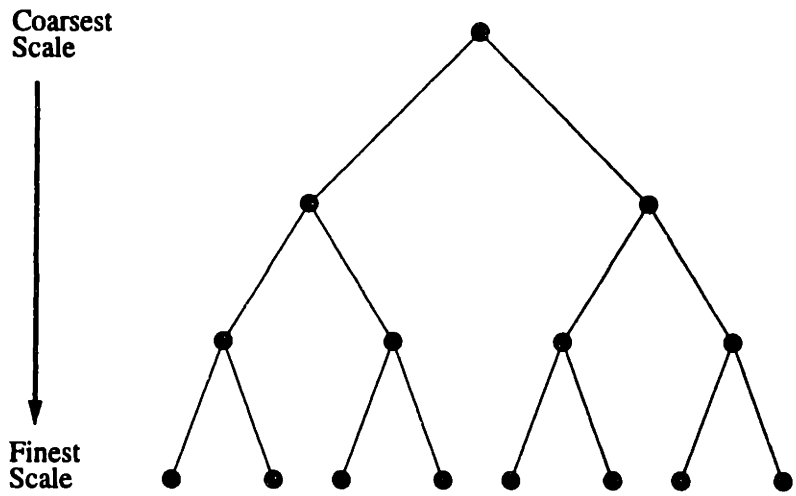


Figure 1.1. A binary tree used to index a random process at multiple resolutions.

not error statistics [7, 18, 58], require regular measurements so that wavelet transforms can be applied [72], or admit only very special nonlocal measurements, namely those that correspond to the explicit direct measurements of wavelet coefficients at particular scales [7]. In contrast, the approach that we develop here computes estimates and error statistics, is directly applicable to arbitrary measurement sets, and allows us to use a wide variety of prior statistical models to describe the statistical variability of the phenomenon.

The approach of the multiscale framework is to develop models for random processes and fields within the class introduced in [19]. These models describe random phenomena using tree structures for which each level of the tree represents a different resolution of the phenomenon of interest. An example of a binary tree used to index a multiresolution process is illustrated in Figure 1.1. Analogous to 1D autoregressive models which evolve recursively in time, these multiscale models evolve recursively in scale. The utility of this class of models is twofold. First, the class has been shown to provide useful models for a handful of random processes and fields, such as 1D Markov processes and 2D Markov random fields (MRFs) [62] and self-similar processes that can be used to model natural phenomena arising in geophysics [34, 35]. Second, and most importantly, just as the Markov property associated with 1D autoregressive models leads to a highly efficient estimation algorithm (the Kalman filter), the multiscale models satisfy a Markov property in scale and space which leads to an efficient estimation algorithm. Also, the multiscale estimator automatically, i.e., with no additional computations, produces estimation error covariances. Moreover, the efficiency of this algorithm does not require regular data and in particular can accommodate arbitrarily spaced measurements.

As noted, the multiscale framework has been successfully applied to a number of problems, but all of this work has been focused on the finest scale of the multiscale rep-

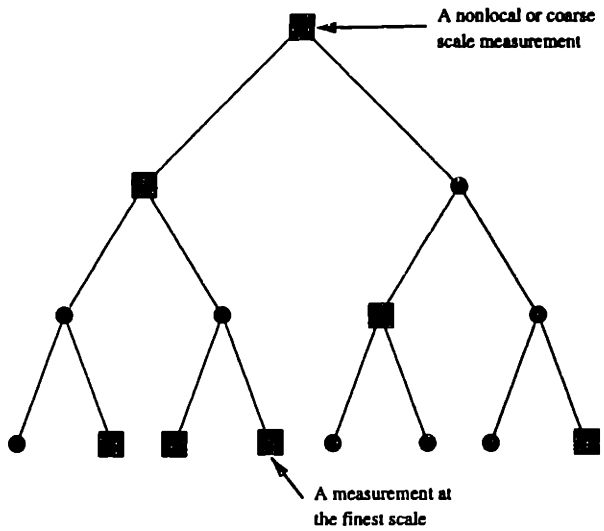


Figure 1.2. An example of the set of variables that can be measured and incorporated by the multiscale estimator. The nodes at which variables are measured are indicated by the shaded squares.

resentation. That is, in modeling a random phenomenon in this framework the objective has been to ensure that the finest scale of the model has a desired statistical structure. Also, in estimation applications, all of the measurements considered have been at the finest level of representation, i.e., they have corresponded to point measurements of the phenomenon. In that context, the variables captured at higher (coarser) levels in the multiscale representation are simply abstract variables that admit efficient algorithms by satisfying the Markov property of multiscale tree processes. Nevertheless, these algorithms actually allow measurements and produce estimates at these coarser scales. In fact, the measurements incorporated by the estimator can be noisy linear functions of any variable on the tree. An example measurement set is illustrated in Figure 1.2. The measured variables, denoted by the shaded nodes, can be arbitrarily distributed on the tree, but each measurement must be in the form

$$y(s) = C_s\, z(s) + v(s)\,, \tag{1.1}$$

where $z(s)$ is the variable at tree node s and $v(s)$ is uncorrelated measurement noise. The multiresolution estimator produces the LLSE estimate of each variable $z(\cdot)$ on the tree from any set of measurements in the form of Eq. (1.1).

For data fusion problems, the coarser-scale variables of the multiscale models must be able to represent the nonlocal weighted averages of the phenomenon of interest. These nonlocal averages correspond either to variables that are measured through indirect measurements or to variables that must be estimated. Once a multiscale model is found with the proper coarse-scale variables and the desired statistics for the finest-scale process, the multiscale estimator can be applied to the fusion of measurements at different resolutions. One of the contributions of this thesis is to show exactly how such multiscale models can be realized.

The second contribution of this thesis is to apply the multiscale framework to problems in groundwater hydrology. We first consider the estimation of hydraulic conductivity from head and conductivity measurements. Because head samples provide nonlocal measurements of hydraulic conductivity through the PDE describing groundwater flow, head samples can be represented as the coarse-scale variables in a multiscale process for hydraulic conductivity. Given that the finest-scale process represents samples of hydraulic conductivity, the multiscale estimator can then be used to estimate hydraulic conductivity (at multiple scales) from measurements of both head and conductivity.

For most field experiments, one has access to more than just head and conductivity measurements. Another common source of measurements are the times that particles take to flow between two locations in the reservoir. Travel times directly measure the velocity function over the path between the two points. Because advective velocity is also a function of hydraulic conductivity, travel time measurements serve as another nonlocal measurement of hydraulic conductivity. Furthermore, because the multiscale estimator produces estimates of the multiscale process at each node on the tree, the multiscale framework can also be used to compute *estimates* of travel times between points in the aquifer. Estimates of travel times are necessary for many EPA studies [98, 99].

1.1.2 Statistically Self-Similar Processes and Fractional Brownian Motion

Another contribution of this thesis is the development of multiscale models for statistically self-similar processes. Self-similar processes occur frequently in nature [4, 57] and are often well modeled by random processes with power spectral densities that behave as $1/f^{\alpha}$. For example, $1/f$ processes can be used to describe average temperature distributions [45, 57], annual flow rates in rivers [45], the noise in vacuum tubes and electrical components [57], biological time series like heartbeats [96], economic time series like the Dow Jones Industrial Average [96], and traffic in communications networks [94]. Processes with $1/f$-like power spectra are also used to generate images that model real world objects like clouds and mountain ranges [4, 89]. These processes possess two common characteristics, statistical self-similarity and long-range dependence. Also, many of the processes are nonstationary. A popular model that possesses these two characteristics is the class of fractional Brownian motions [64], which are a generalization of Brownian motion.

Fractional Brownian motions are defined as the zero-mean Gaussian random processes with statistically stationary and self-similar increments. The first models for fBm [64] were fractional integrals of white Gaussian noise, but such nonlinear integrals are not useful for synthesizing or processing (estimating, smoothing, and the like) fBm. Developing useful models for fBm and other $1/f$ processes has been a very active area of research, as noted in [56, 96]. Most practical approaches approximate fBm with models that lead to efficient synthesis or processing algorithms. Two such approximations are random midpoint displacement and wavelet-based representations. The random mid-

point displacement algorithm is a popular tool for approximately synthesizing fBm [4]. Another useful tool is the wavelet synthesis equation, since the wavelet transform has been shown to approximately whiten fBm [36] and $1/f$ processes in general [96]. In addition, Wornell [96] also used the wavelet-based framework for accomplishing signal processing tasks like the estimation of fBm from uniformly sampled and noisy measurements.

Both the midpoint displacement and the wavelet-based approximations lead to synthesis algorithms consisting of a progression from coarse to fine scales, adding successively finer details at each step. The details for the midpoint displacement algorithm are perturbations about interpolated values, while the details for the wavelet-based model are the detail coefficients of the wavelet transform. Because both synthesis algorithms are analogous to the multiscale autoregression, the random midpoint displacement and wavelet-based approximations are naturally represented by multiscale processes [35, 62]. A major advantage of representing these algorithms within the multiscale framework is the ability to take advantage of the efficient processing and synthesis algorithms. However, the approximations to fBm given by directly mapping the midpoint displacement and wavelet-synthesis algorithms to tree models have some undesirable properties. First, both of these approaches do not account for the inherent nonstationarity of fBm. Specifically, for both the displacement and the wavelet-based models developed in the literature, the process noise is assumed to have constant variance at any scale of the tree. For the wavelet-based approximation proposed in [35], this assumption leads to a finest-scale process with stationary variance, unlike the variance of fBm which grows polynomially in time. More importantly, because the process noise of these two models represents either the displacements or the wavelet detail coefficients, assuming that the displacements (detail coefficients) are completely uncorrelated leads to large approximation errors in the covariance of the finest-scale process. For instance, the wavelet-based multiscale model (using Haar wavelets) produces sample paths with noticeable discontinuities. These artifacts can be distracting if used for synthesis and misleading if used for estimation. Fortunately, as we will develop, much of the correlation among the displacements (detail coefficients) can be captured in multiscale models without increasing the state dimensions of the models.

While the multiscale models based on random midpoint displacement and wavelet synthesis can be enhanced, there might also be models more tailored to the particulars of multiscale tree processes. In fact, multiscale tree models appear to be a natural framework for representing statistically self-similar processes, since trees themselves are geometrically self-similar. We show in this thesis how statistical self-similarity, in conjunction with stationary increments, leads to a fundamental approach for approximating fBm with multiscale trees. This analysis will also apply to the multiscale modeling of other self-similar processes.

■ 1.2 Contributions and Organization

In what follows, the contents of each chapter are briefly summarized and tied to the overall contributions of this thesis.

Chapter 2, Background: Estimation Theory and Multiscale Processes
This chapter first reviews the properties of optimal Bayesian estimators and linear least-squared error (LLSE) estimators. This review includes the computational complexity of standard implementations of LLSE estimators, as well as how to handle singular measurement covariances and nonlinear measurement equations. The second half of the chapter is devoted to the class of multiscale models first described in [19, 20]. A number of example processes are provided, including internal models for Markov Random Fields and external models for process with approximately $1/f$ power spectra. The chapter concludes with a discussion of the realization theory developed in [51], which is used throughout this thesis.

Chapter 3, Groundwater Flow and Hydraulic Conductivity Estimation
This chapter summarizes the partial differential equations (PDEs) commonly used to describe the flow of water within the earth's subsurface, and then discusses the problem of calibrating these equations. The focus is on estimating hydraulic conductivity, an important parameter of the flow equations, from samples of conductivity and hydraulic head. In particular, we show how the head measurement equation can be linearized, leading to a nonlocal observations of the hydraulic conductivity function. This linearization can be used to develop an approximate LLSE estimator of hydraulic conductivity, and is the linearization required by the Gauss-Newton solution to the maximum a posteriori estimator. This approach to conductivity estimation is quite general, and has been used to incorporate measurements of contaminant concentrations [84].

Chapter 4, Extensions of Multiscale Realization Theory
This chapter focuses on extensions of the class of processes that can be realized within the framework of [19, 20]. The major contribution is to show how nonlocal functions of the phenomenon of interest can be represented by the coarser-scale variables of a multiscale tree process. The first approach presented is to extend the realization algorithm presented in [51], which specifies only the statistics of the finest-scale process, so that desired nonlocal functions of the finest-scale process can be represented at the coarser-scale nodes on the tree. While instructive, this approach is computationally infeasible for large problems. As an alternative, a method is presented for augmenting the states of multiscale models with the desired linear functions of the finest-scale process. The augmentation must preserve the statistical integrity of the original multiscale model, i.e., it must preserve the Markov property and the finest-scale statistical structure of the process. This algorithm is applied to an example estimation problem—the estimation of hydraulic conductivity from head and conductivity for one-dimensional flow—that

illustrates how one might choose the nodes at which the nonlocal functions are to be represented. The choice of nodes affects the state dimensions of the augmented multiscale model. Finally, an approximate state augmentation algorithm is presented that allows one to control the increases in state dimensions in return for sacrificing statistical accuracy.

Chapter 5, Multiscale Modeling and Estimation of Hydraulic Conductivity
This chapter applies the multiscale framework to the estimation of hydraulic conductivity in 2D from measurements of head and conductivity. First, the effect of head measurements on conductivity estimates is discussed in some detail. Then the state augmentation algorithm of Chapter 4 is used to represent linearized head measurements at the coarser scales of multiscale models for 2D hydraulic conductivity. The multiscale estimator in these examples implements an approximation to the optimal LLSE estimator. Next, the head measurements are relinearized about current estimates of hydraulic conductivity, leading to an iterative algorithm for which the state augmentation algorithm of Chapter 4 must be applied at each iteration. The resulting estimate is the Gauss-Newton implementation of the maximum a posteriori estimator, and it provides a significant improvement over the approximate LLSE estimator only when the variations in the head function due to variations in hydraulic conductivity are significant when compared to the head measurement noise.

Chapter 6, Travel Time Measurements and Estimation
The chapter extends the work in Chapter 5 to include travel-time measurements. First, a linearization of travel time with respect to hydraulic conductivity is developed which allows travel times to be represented at the coarser scales of multiscale tree models. The multiscale framework can then be used estimate hydraulic conductivity from conductivity, head, and travel-time measurements. Next, the multiscale framework is used to estimate conditional distributions for travel times. These densities are then compared to the conditional distributions generated by Monte-Carlo Simulations.

Chapter 7, Modeling and Estimation of Fractional Brownian Motion
A number of multiscale models that approximate fractional Brownian motion are presented. First, the random midpoint displacement and wavelet synthesis algorithms for synthesizing fBm are discussed and then represented within the multiscale framework. The multiscale framework not only allows one to take advantage of efficient processing and synthesis algorithms, but also allows one to develop more accurate representations of fBm by accounting for local correlations among displacements or detail coefficients. The second part of the chapter develops more general results for the multiscale modeling of statistically self-similar processes and for processes with stationary increments. These results are then applied to approximating fBm, and are shown to provide very accurate representations even when the dimensions of the state variables are small.

Chapter 8, Contributions, Limitations, and Potential Solutions
A brief summary of the major contributions of the thesis is provided. We then point out the limitations of the contributions, the problems that should be addressed, and some possible solutions. Finally, alternative approaches to multiscale realization are suggested.

Chapter 2

Background: Estimation Theory and Multiscale Processes

This chapter provides an introduction to the multiscale stochastic framework, which is one of the primary subjects of this thesis. To motivate the framework and to justify its application, a brief review of standard estimation theory is provided, focusing particularly on linear least-square error (LLSE) estimators. The shortcomings associated with standard implementations of estimators are noted, along with the problems encountered when the measurements are nonlinear functions of the variables to be estimated. The multiscale framework can be seen as a method for extending LLSE estimators to problems of very large dimension, as is common for the estimation of two-dimensional random processes, or problems which are naturally decomposed into multiple resolutions. The multiscale framework consists of the following: a class of stochastic processes which are indexed on a tree and which evolve recursively from coarse to fine scale, a collection of tools for realizing such processes to have a desired statistical structure, an efficient estimation algorithm analogous to the Kalman filter, a probabilistic model for the estimation errors, and a likelihood calculator.

2.1 Estimation Theory

The generic estimation problem is to estimate a vector[1] f from a vector of observations y. This problem is covered in detail in [79, 87, 95] and is briefly reviewed here. The difficulty is that the observations usually contain incomplete information about f. For instance, the dimension of f may be larger than the dimension of y, the observations may be noisy or degraded, and the relationship between y and f may be ill-conditioned[2], i.e., very large variations in f lead only to small variations in y. If the dimension of f is much larger than that of y, additional information or assumptions about f must be supplied to the estimator. Ill-conditioning requires that the estimator of f be chosen

[1] For this section, we assume that both f and y are members of finite-dimensional vector spaces. For some of the applications encountered in this thesis, the unknown variable is a function, e.g., $f(x)$ for $x \in \mathbb{R}$, and thus lives in an infinite-dimensional vector space. How to extend the results of this chapter to such problems is described in Chapter 3.

[2] A thorough discussion of ill-conditioned (ill-posed) problems is given by Tikhonov in [91].

such that small perturbations in y, either from numerical or measurement errors, do not lead to large changes in the estimate. One method for dealing with these problems is the Bayesian framework.

Bayesian estimators require the specification of two probability densities, $p_f(F)$ and $p_{y|f}(Y \mid F)$. The former density is known as the prior, since it represents knowledge about f prior to the acquisition of any measurements. The latter density is the probability density of the measurements conditioned on perfect knowledge of f. In practice this density is usually specified implicitly by a measurement equation, e.g.,

$$y = H(f) + v\,, \tag{2.1}$$

where v is the measurement noise. If the joint density of v and f is known, then $p_{y|f}(Y \mid F)$ can in theory be derived from the measurement equation.

In the Bayesian framework, the solution to the estimation problem is

$$p_{f|y}(F \mid Y) = \frac{p_{y|f}(Y \mid F)\, p_f(F)}{p_y(Y)}\,. \tag{2.2}$$

The conditional probability density $p_{f|y}(F \mid Y)$ contains all the information (and uncertainty) about f after conditioning on the observations y. The problem is that specifying such a density is usually impractical for very large dimensional problems. Instead, estimators like the mean, mode, or median of the conditional density are computed.

The mean, mode, and median of the conditional density $p_{f|y}(F \mid Y)$ are part of a class of estimators known as optimal Bayesian estimators. Optimal Bayesian estimators are given by specifying a cost function that quantifies preferences for different estimation errors. The optimal estimate is the one which minimizes the expected cost of the estimation error. If $\hat{f}(y)$ is the estimate of f as a function of the measurement vector y, then the estimation error is given by

$$e(f, \hat{f}(y)) \triangleq f - \hat{f}(y)\,. \tag{2.3}$$

Given a cost function $C(e)$, the optimal Bayesian estimate is given by

$$\hat{f}(y) = \arg\min_{g(\cdot)} \iint C\big(e(F, g(Y))\big)\, p_{f,y}(F, Y)\, dF\, dY \tag{2.4}$$

This leads to the much simpler condition [95]

$$\hat{f}(y) = \arg\min_{a} \int C\big(e(f, a)\big)\, p_{f|y}(F \mid y)\, dF\,. \tag{2.5}$$

Note that the optimal Bayesian estimator only depends on the cost function and the *conditional* density for f, which should not be surprising given that this conditional density contains all the information about f after conditioning on the measurements.

2.1.1 Least-Squares Estimation

If the error covariance is a suitable metric for measuring the performance of an estimator, then a natural cost function is $C(e) = \|e\|^2$. Substituting this cost function into Eq. (2.5) and differentiating with respect to a yields

$$\hat{f}_{\mathrm{BLSE}}(y) = E[f \mid y] , \tag{2.6}$$

which is the mean of the conditional density $p_{f|y}(F \mid Y)$. The covariance of this conditional density, which is also the covariance of the estimation error conditioned on y, is given by [3]

$$P_{e|y}(y) = E\Big[\big(f - E[f \mid y]\big)\big(f - E[f \mid y]\big)^T \Big| y\Big] , \tag{2.7}$$

where we have made use of the zero bias of the Bayes' least-squares error (BLSE) estimator. Note that the covariance of the estimation error will in general be a function of the observation vector y. The unconditional error covariance is given by taking the expectation over all values of y to yield

$$P_{\mathrm{BLSE}} = E\big[e(f, \hat{f}_{\mathrm{BLSE}})\, e(f, \hat{f}_{\mathrm{BLSE}})^T\big] = E\big[P_{f|y}(y)\big] , \tag{2.8}$$

and is therefore not a function of y.

The BLSE estimator has a number of interesting properties.

- $\hat{f}_{\mathrm{BLSE}}$ is unbiased.
- The error $e\big(f, \hat{f}_{\mathrm{BLSE}}(y)\big)$ is orthogonal to all functions of the data.
- $E\big[C(e)\big] = \mathrm{trace}[P_{\mathrm{BLSE}}]$
- $P_{\mathrm{BLSE}} - P \leq 0$, where P is the error covariance of any other estimator and the inequality means that the matrix difference is negative semidefinite.

The problem with general Bayesian estimators is that the estimator $\hat{f}_{\mathrm{BLSE}}(y)$ can be a very complicated nonlinear operator which is difficult to compute. This estimator also requires complete knowledge of the conditional density, which is impractical to compute for many problems. In such cases one usually settles for suboptimal estimators which are easier to implement and analyze.

Linear Least-Squares Estimation

Keeping the least-squares cost function, one means for obtaining a suboptimal estimator is to limit the estimator to be a linear (affine) function of the data, i.e.,

$$\hat{f}(y) = Ay + b .$$

[3] In this thesis, the mean and covariance of a random vector x will be denoted by m_x and P_x, respectively. Likewise, the cross-covariance of x and y will be denoted by P_{xy} and the covariance of x after conditioning on y is $P_{x|y}$.

Substituting this representation into Eq. (2.5) and differentiating with respect to A and b yields the following:

$$\hat{f}_{\text{LLSE}}(y) = m_f + P_{fy}P_y^{-1}(y - m_y) \tag{2.9a}$$

$$\begin{aligned} P_{\text{LLSE}} &= P_e \\ &= P_f - P_{fy}P_y^{-1}P_{fy}^T \end{aligned} \tag{2.9b}$$

where m_f is the mean of f, P_{fy} is the cross-covariance of f and y, and P_y is the covariance of y. The estimator in Eq. (2.9a) is known as the linear least-squares error (LLSE) estimator. Note that both the LLSE estimator and the estimation error covariance **depend only on first- and second-order statistics of f and y for any joint probability density of f and y.** This property is crucial, since it can simplify considerably the estimation problem. Some other properties of LLSE estimators are the following:

- $\hat{f}_{\text{LLSE}}(y)$ is unbiased.
- The estimation error is orthogonal to all linear (affine) functions of the data.
- $P_{\text{LLSE}} - P_{e_{\text{L}}} \leq 0$, where $P_{e_{\text{L}}}$ is the error covariance of any linear estimator and P_{LLSE} is the LLSE estimation error defined in Eq. (2.9b). However, remember that the linear estimator is in general sub-optimal with respect the optimal Bayesian estimator.
- $P_{e|y} = P_{\text{LLSE}}$.

The last property simply states that the conditional estimation error covariance does not depend on the value of y.

LLSE Estimation for Linear Measurements

In order to implement the LLSE estimator, the covariance matrices P_{fy} and P_y, as well as the mean m_y, must be computed. Closed-form expressions for these moments are difficult to compute in general, yet there exist simple expressions when the measurement equation is linear. Specifically, consider the measurement equation

$$y = Hf + v\,, \tag{2.10}$$

where H is an M-by-N dimensional matrix. Assume for simplicity that v is zero mean, has covariance R, and is uncorrelated with f. The mean and cross-covariances can then be computed in closed form, which leads to the following LLSE estimator equations:

$$\hat{f}(y) = m_f + P_f H^T(HP_fH^T + R)^{-1}(y - Hm_f)\,, \tag{2.11a}$$

$$P_e = P_f - P_f H^T(HP_fH^T + R)^{-1}HP_f\,. \tag{2.11b}$$

These equations can be expressed alternatively as

$$\hat{f}(y) = m_f + (P_f^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1} (y - H m_f), \tag{2.12a}$$

$$P_e^{-1} = P_f^{-1} + H^T R^{-1} H. \tag{2.12b}$$

The relative merits of Eqs. (2.11) and (2.12) depend on M and N, the dimensions of the measurement vector and the vector to be estimated.

Consider the number of computations required to implement each of the LLSE estimators. Assuming all the matrices in Eqs. (2.11) and (2.12) are full—except the measurement noise covariance, which is assumed to be diagonal throughout this thesis—and that direct methods are used to compute the matrix inverses, then Eq. (2.11a) requires $\mathcal{O}(M^3+N^2M)$ computations while Eq. (2.12a) requires $\mathcal{O}(N^3)$ computations. Note that the cubic terms in the asymptotic complexities are due the LU factorizations required for the matrix inversions. Equation (2.11a) will be much more efficient to implement for large problems when $M \ll N$. This scenario is likely when one desires estimates of f at a finer resolution than can be supported by the data. For $N \ll M$, Eq. (2.12a) is much more efficient to implement. In any case, the number of computations required to implement the estimator and compute the estimator covariance becomes prohibitive as both M and N grow large.

Iterative methods can be used to speed up the computation of the estimates. For instance, Eq. (2.11a) requires only $\mathcal{O}(M^2 + N^2M)$ computations when the matrix $(HP_fH^T + R)$ is symmetric positive definite, since the conjugate gradient algorithm [5] can be employed. However, these computational savings will not apply to the computation of the error covariances, since the savings provided by the iterative method only apply to the computation of a single vector $(HP_fH^T + R)^{-1}(y - m_y)$, not to the computation of the entire matrix $(HP_fH^T + R)^{-1}$.

Now consider the storage requirements of the LLSE estimators, which are the same for either set of equations. The estimate $\hat{f}$ requires only N storage elements. However, a covariance matrix requires N^2 storage elements, unless the random vector is stationary. By a stationary random vector we mean that f contains samples of a stationary, discrete-index random process or field, in which case the covariance of a single sample with all other samples completely describes the entire covariance matrix. For storing the estimator covariance, even if P_f is stationary, P_e will generally be nonstationary and thus require N^2 storage elements. These storage requirements are impractical for large problems—$N = 10^4$ implies that approximately 1 Gigabyte of storage are required for double-precision arithmetic. Therefore, if LLSE estimators are to be used for solving large-dimensional estimation problems while also calculating estimator covariances, alternative frameworks must be employed which account for both the growth in estimator complexity and the storage requirements for the estimator covariance.

An alternative implementation of Eq. (2.12a) is given for specific priors P_f and measurement matrices H. Consider implementing

$$(P_f^{-1} + H^T R^{-1} H)(\hat{f}(y) - m_f) = H^T R^{-1} (y - H m_f), \tag{2.13}$$

which is just a reformulation of Eq. (2.12a). If f is a wide-sense Markov random field (MRF), it can be modeled implicitly as [25]

$$Af = w,$$

where A is a symmetric local operator in the form of a discretization of an elliptic PDE and w has covariance A; thus, $P_f^{-1} = A$. Remember that R is diagonal, so that $H^T R^{-1} H$ will also be diagonal if the measurements are point observations of individual elements in f. In this case, the matrix operator on the left-hand side of Eq. (2.13) has the same sparsity (locations of the nonzero elements) as does A. Therefore, a direct solver like nested dissection [39, 40] can be used to implement the estimator in $\mathcal{O}(N^{3/2})$ computations. Furthermore, the elements of P_e that lie in the nonzero locations required to store the LU factorization of A can be computed with little extra computation [28, 30]. Note that these elements include the diagonal of P_e, which contains the variances of the individual elements of the estimator. However,

- the entire matrix P_e is still infeasible to compute for large-dimensional problems, so that the variance of particular linear functions of $\hat{f}$ cannot be computed without significant additional computations;
- the efficiency of the MRF implementation diminishes when a significant number of nonlocal measurements are incorporated in the estimate, since the factorization of the matrix $(P_f^{-1} + H^T R^{-1} H)$ will no longer be sparse.

The need to incorporate measurements of such nonlocal functions, or to estimate and characterize the uncertainty of such nonlocal functions, will be demonstrated in the application chapters. The multiscale framework described in Section 2.2 will be the tool used in this thesis to overcome these problems.

LLSE Estimation and Gaussian Random Variables

For Gaussian random variables, which are characterized completely by their mean and covariance, the LLSE estimator is optimal in a Bayes' least-squares sense. In this thesis we will use the notation

$$f \sim \mathcal{N}(m_f, P_f)$$

to denote that a random vector f is a Gaussian (normal) random variable with mean m_f and covariance P_f, while $f \sim (m_f, P_f)$ will simply denote that f has the same mean and covariance but is not necessarily Gaussian.

If the measurement vector y and the unknown vector f are jointly Gaussian, then the LLSE estimate of f from y is equal to the Bayes' least-squares estimate of f. Therefore, we have that $\hat{f}_{\mathrm{BLSE}}(y) = \hat{f}_{\mathrm{LLSE}}(y)$ and $P_{e|y}$ is independent of the value of y. Furthermore, the conditional density $p_{f|y}(F \mid Y)$ is Gaussian, with mean $\hat{f}$ and covariance P_e, so that the LLSE estimator covariance completely summarizes the uncertainty in f after conditioning on y.

Due to the equality of the BLSE estimator with the LLSE in the jointly Gaussian case, it is tempting to approximate the sub-optimality of the LLSE estimator using the deviation of f and y from jointly Gaussian random variables. However, because there exist non-Gaussian estimators for which the BLSE and LLSE estimators coincide [95], this reasoning can lead to faulty conclusions. An example of non-Gaussian y is given in Section 2.1.2.

Singular Estimation

For many problems, the measurements are highly correlated and have covariances which are extremely ill-conditioned or even singular. In this case, implementing the LLSE estimator in Eq. (2.9) requires some care, since numerical round-off errors in the matrix inversion can lead to large errors in the estimator. A solution is to replace y with a linear function $u = Ly$ that has a well-conditioned covariance, where u retains most of the information in y used to estimate f.

The condition number of a matrix is given by computing its singular value decomposition (SVD) [43]. Because covariance matrices are symmetric, their singular value decompositions have the form

$$P_y = USU^T ,$$

where U is orthogonal and S is diagonal and positive semidefinite. The diagonal elements of S are the singular values, which are in descending order, i.e., $S(k,k) \geq S(k+1,k+1)$. For symmetric matrices, the singular value decomposition is equivalent to the eigenvalue decomposition, i.e., S contains the eigenvalues of P_y. However, the SVD is often preferred for numerical reasons, e.g., the diagonal elements of S returned by a SVD are always non-negative when P_y is positive semidefinite, whereas the values returned by an eigenvalue decomposition may be slightly negative.

For the covariance matrix of y, the singular values correspond to the variances of particular linear combinations of y. Namely,

$$\mathrm{var}\left[U(:,k)^T y\right] = S(k,k) .$$

The precision of finite precision arithmetic is limited by the ratio of $S(M,M)$ to $S(1,1)$, so that values of $S(k,k)$ which fall below a certain threshold should be treated as zero. These linear combinations of y can be discarded in forming u. A standard threshold is given by [48]

$$\sigma^2_{\min} = \epsilon * S(1,1) * M ,$$

where ϵ is the numerical precision of the computer, $S(1,1)$ is the largest singular value, and M is the dimension of y. If the matrix S has m singular values greater than or equal to $\sigma^2_{\min}$, then setting $L = U(1:m,:)^T$ yields

$$\begin{aligned} P_u &= L(USU^T)L^T , \\ &= \Sigma , \end{aligned}$$

where $\Sigma = \text{diag}\big(S(1,1), S(2,2), \ldots, S(m,m)\big)$. The estimation problem is now one of estimating f from an observation vector u with dimension m. This estimator follows as

$$\hat{f}(y) = m_f + P_{fy}L^T(LP_yL^T)^{-1}(u - Lm_y)\,, \tag{2.14a}$$

$$P_e = P_f - P_{fy}L^T(LP_yL^T)^{-1}LP_{fy}^T\,. \tag{2.14b}$$

This modified estimator is used in implementing many of the algorithms described in this thesis. Note that the matrix inversions in Eq. (2.14) will not encounter any numerical conditioning problems. The estimator in Eq. (2.14a) will be approximately equal to the exact (using infinite precision arithmetic) LLSE estimator as long as the linear functions of y which are discarded do not contain significant information[4] about f.

■ 2.1.2 Nonlinear Measurements

Optimal estimates, including LLSE estimates, are more difficult to compute when the relationship between f and y is nonlinear, i.e., when the observation function $H(\cdot)$ is nonlinear. In this section, we describe how to compute an approximation to the LLSE estimator, and how this approximation is equivalent to a single iteration of the Gauss-Newton implementation of the maximum *a posteriori* (MAP) estimate when f is Gaussian. The need to consider nonlinear measurements will be evident in Chapter 3.

Consider measurements generated by Eq. (2.1) when $H(\cdot)$ is nonlinear. In order to compute the LLSE estimate of f from y, the covariances P_y and P_{fy} must be computed. However, these matrices are in general very difficult to compute when H is nonlinear. One possible solution is to linearize $H(f)$ about some vector f_0, and then apply the LLSE estimator to the linearized problem to obtain an approximation of the true LLSE estimate. A Taylor series expansion of $H(f)$ about f_0 yields

$$H(f) = H(f_0) + \nabla H(f_0)\,(f - f_0) + \mathcal{O}(|f - f_0|^2)\,, \tag{2.15}$$

where $\nabla H(f_0)$ is the Jacobian of $H(f)$ evaluated at f_0. Substituting Eq. (2.15) into Eq. (2.1), the measurement equation becomes

$$\underbrace{y - H(f_0) + \nabla H(f_0)\,f_0}_{\mathcal{Y}(f_0)} = \nabla H(f_0)\,f + \underbrace{v + \mathcal{O}(|f - f_0|^2)}_{\nu}\,. \tag{2.16}$$

The problem now is to estimate f from the measurement vector $\mathcal{Y}(f_0)$. Assuming $v \sim (0, R)$ and $v \perp f$, the covariances $P_{\mathcal{Y}}$ and $P_{f\mathcal{Y}}$ can be approximated as

$$P_{f\mathcal{Y}} \approx P_f\,\nabla H(f_0)^T \quad \text{and} \quad P_{\mathcal{Y}} \approx \nabla H(f_0)\,P_f\,\nabla H(f_0)^T + R \tag{2.17}$$

[4]The elements of y which contain significant information about f are not necessarily those with large variances. The real problem is to determine the subspace of y which most significantly effects the LLSE estimate of f. The complementary subspace can be discarded when forming the projection Ly. In other words, both P_{fy} and P_y need to be considered simultaneously when P_y is numerically ill-conditioned. Such a complete solution to the singular estimation problem is beyond the scope of this discussion.

by ignoring the higher-order terms of $(f - f_0)$. (The effect of these higher-order terms can be approximated by increasing the measurement error covariance to account for the linearization error in $\mathcal{V}$.)

Using the linearized measurement equation and the approximate cross-covariances, an approximation to the LLSE estimate can then be derived using either Eq. (2.11) or Eq. (2.12). For instance, Eq. (2.12a) yields

$$\hat{f} = m_f + \left(P_f^{-1} + \nabla H(f_0)^T R^{-1} \nabla H(f_0)\right)^{-1} \nabla H(f_0)^T R^{-1} \left(\mathcal{Y}(f_0) - \nabla H(f_0)\, m_f\right). \tag{2.18}$$

However, remember that the estimator covariances given by Eqs. (2.11b) and (2.12b) are only approximations to the true covariance of the estimate, since they assume that $P_{\mathcal{Y}}$, $P_{f\mathcal{Y}}$, and $m_{\mathcal{Y}}$ are computed exactly.

Now consider the case when f and v are jointly Gaussian. We already noted that for f and y jointly Gaussian, the LLSE estimator is equal to the BLSE estimator. Also, because the conditional probability density for $p_{f|y}(F \mid Y)$ is Gaussian, the BLSE and LLSE estimates are also equal to the MAP estimate, which is the mode (or location of maximum value) of the conditional density. However, when $H(\,)$ is nonlinear, a Gaussian distribution for f and v does not guarantee that y is Gaussian, or that the distribution of f conditioned on y is Gaussian. The conditional distribution of f is given by

$$p_{f|y}(F|y) = c \exp\left(-\frac{1}{2} J(F)\right), \tag{2.19}$$

$$J(f) = \left(y - H(f)\right)^T R^{-1} \left(y - H(f)\right) + (f - m_f)^T P_f^{-1} (f - m_f)$$

where c is a constant independent of f. Note that this density is generally not Gaussian when $H(\cdot)$ is nonlinear.

The maximization of Eq. (2.19) is equivalent to the following minimization

$$\hat{f}_{\text{MAP}}(y) = \arg\min_f J(f)$$

There are many ways to minimize $J(f)$; one method is to solve for the extremal points of $J(f)$, which are given by those f for which $\nabla J(f) = 0$. Equivalently, these f satisfy

$$P_f^{-1}(f - m_f) = \nabla H(f)^T R^{-1} \left(y - H(f)\right). \tag{2.20}$$

A solution to Eq. (2.20) can be found using an iterative algorithm which successively linearizes the function $H(f)$ about the present value of the MAP estimate. This algorithm is known as the Gauss-Newton iteration [69]. Assuming that $\hat{f}_k$ is the present value of the MAP estimate, the linearization gives

$$P_f^{-1}(f - m_f) = \nabla H(\hat{f}_k)^T R^{-1} \underbrace{\left(y - H(\hat{f}_k) + \nabla H(\hat{f}_k)\, \hat{f}_k\right.}_{\mathcal{Y}(\hat{f}_k)} \left. - \nabla H(\hat{f}_k)\, f\right).$$

Setting f to $\hat{f}_{k+1}$ and rearranging gives

$$\hat{f}_{k+1} = m_f + \left(P_f^{-1} + \nabla H(\hat{f}_k)^T R^{-1} \nabla H(\hat{f}_k)\right)^{-1} \nabla H(\hat{f}_k)^T R^{-1} \left(\mathcal{Y}(\hat{f}_k) - \nabla H(\hat{f}_k) m_f\right). \quad (2.21)$$

The value to which this sequence converges is taken as the MAP estimator. Note that Eq. (2.21) reduces to the LLSE estimator when H is linear. Also, the linearized approximation to the LLSE estimator can be seen to be equivalent to a single iteration of the Gauss-Newton minimization. In the remaining section we return our focus to LLSE estimators based on linear measurement equations. We will return to nonlinear measurements in Chapter 3.

■ 2.2 Multiscale Stochastic Models and Estimation

This section is broken into two parts. First, the class of tree-indexed multiscale processes is defined using an autoregression in scale. Second, an estimator based on the Markov property of multiscale trees is summarized.

■ 2.2.1 Multiscale Models on Trees

The class of multiscale random processes introduced in [17, 19, 20] is indexed by the nodes of trees organized into scales. The coarsest scale is indexed by the root node, while the finest scale is indexed by the set of leaf nodes. The root node and leaf nodes are illustrated for a binary tree in Figure 2.1a. The multiscale process defined on a tree consists of a set of random vectors $z(s)$, one for each node s on the tree. The *scale* of node s, which we denote by $m(s)$, is the distance between node s and the root node of the tree. To describe the relationship between the process at neighboring scales, first define $\overline{\gamma}$ to be the upward (in scale) shift operator, so that $s\overline{\gamma}$ denotes the parent of any node s. The relationship between s and $s\overline{\gamma}$ is illustrated in Figure 2.1b. The class of multiscale processes considered in this paper satisfies the following autoregression *in scale*

$$z(s) = A_s\, z(s\overline{\gamma}) + w(s)\,, \quad (2.22a)$$

$$w(s) \sim (0, Q_s)\,, \quad (2.22b)$$

where $z(s)$ is the process value at node s, $z(s\overline{\gamma})$ is the process value at node $s\overline{\gamma}$, and $w(s)$ is the process noise. Equation (2.22) defines an autoregression from coarse to fine scale, where the term $A_s z(s\overline{\gamma})$ can be viewed as an interpolation from the coarser scale and $w(s)$ is the finer-scale detail. The autoregression is initialized at the root node $s = 0$ by

$$z(0) \sim (0, P_0)\,. \quad (2.23)$$

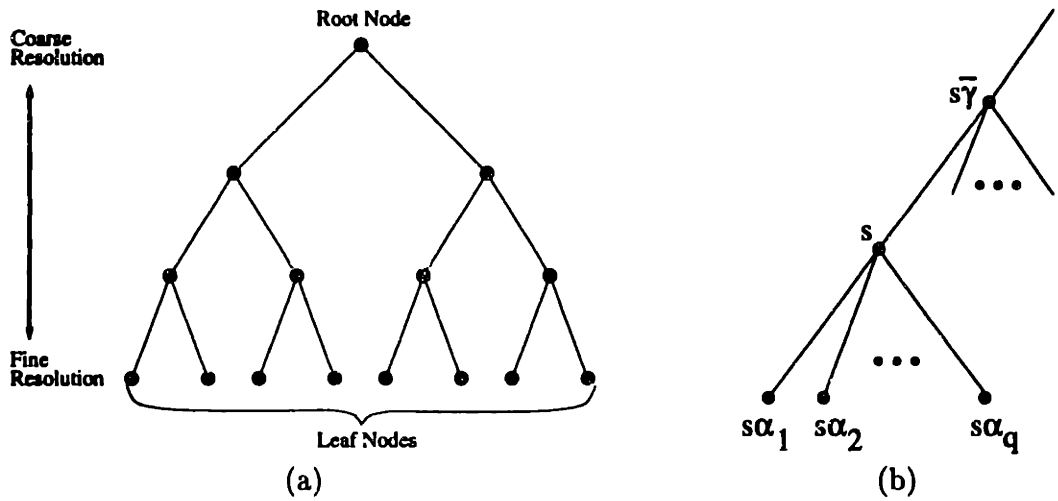


Figure 2.1. (a) A binary tree used to index a random process at multiple resolutions. (b) The local labeling of the $q+1$ nodes connected to node s.

Since $z(0)$ and $w(s)$ are zero-mean, every process value $z(s)$ will be a zero-mean[5] random vector.

The process noise $w(s)$ is assumed to be a white-noise process uncorrelated across scale and space and also uncorrelated with the root node state, i.e., $E[w(s)\,z(0)^T] = 0$. The whiteness of the process noise implies that the second-order moments of a multiscale tree model are characterized completely by P_0—the root node covariance—and the autoregression parameters A_s and Q_s for all nodes $s \neq 0$. (A_s and Q_s are not defined for $s = 0$.) Remember, for optimal linear estimators, that only the first-order and second-order moments of a random variable are required. More importantly, the whiteness of the process noise leads to a Markov property similar to the Markov property for 1D autoregressive processes driven by white noise. Specifically, note that any node s, with q_s defined to be the number of children of node s, partitions the tree into q_s+1 subsets of nodes (see Figure 2.1b): $\mathcal{S}_{s\alpha_1}$, ..., $\mathcal{S}_{s\alpha_{q_s}}$, and $\mathcal{S}_s^c$, where[6]

$$\mathcal{S}_s \triangleq \text{set of nodes descendent from and including node } s$$

$$\mathcal{S}_s^c \triangleq \text{complement of } \mathcal{S}_s$$

$$s\alpha_i \triangleq \text{child of node } s, \quad i = 1, \ldots, q_s\,.$$

We will also find it useful to write $\mathcal{S}_{s\alpha_{q_s+1}} \triangleq \mathcal{S}_s^c$. The Markov property of multiscale tree processes is that, conditioned on the state $z(s)$, the q_s+1 sets of states partitioned

[5]The zero-mean assumption is made for simplicity and is easily relaxed by adding a deterministic term to Eq. (2.22a) or Eq. (2.23).

[6]The only exceptions are the finest resolution leaf nodes which have no children and the coarsest resolution root node which has no parent.

by node s are conditionally uncorrelated. More formally,

$$E\big[z(r)\,z(t)^T \,|\, z(s)\big] = E\big[z(r)\,|\,z(s)\big]\,E\big[z(t)|\,z(s)\big]^T,$$

for all $r \in \mathcal{S}_{s\alpha_i}$, $t \in \mathcal{S}_{s\alpha_j}$, $i \neq j$, and $(i,j) \in [1, q_s+1] \times [1, q_s+1]$. Because this Markov property is a function only of the second-order moments of the process, it is often called a *wide-sense* property. The Markov property is strict-sense if the process is Gaussian [25], in which case the $q_s + 1$ sets of states partitioned by s are conditionally independent. A proof of the Markov property is provided in Appendix A.

Because of the Markov property of multiscale trees, the process value $z(s)$ is will be referred to as the *state* at node s. This terminology is an extension of that for 1D Markov processes. For $q = 1$, the Markov property reduces to the standard notion of Markovianity for discrete-time Markov processes. The state for these 1D processes contains all the information about past values of the process such that the past and present are conditionally uncorrelated.

As noted, the second-order statistics can be determined directly from the root node covariance and the model AR parameters A_s and Q_s. The state covariances satisfy a Lyapunov equation which evolves in scale,

$$P_{z(s)} = A_s P_{z(s\overline{\gamma})} A_s^T + Q_s\,. \tag{2.24}$$

The cross-covariance $P_{z(s)\,z(t)}$ is given by first defining the state transition matrix

$$\Phi(s,t) = \begin{cases} I\,, & s = t\,, \\ A_s\Phi(s\overline{\gamma},t)\,, & m(s) > m(t)\,, \\ \Phi(s,t\overline{\gamma})A_t^T\,, & m(t) > m(s)\,, \end{cases} \tag{2.25}$$

in direct analogy to the state transition matrix used for standard time-series models. Using the whiteness of the process noise, we obtain

$$P_{z(s)\,z(t)} = \Phi(s, s\wedge t) P_{z(s\wedge t)} \Phi(s\wedge t, t)\,, \tag{2.26}$$

where $s \wedge t$ is defined to be the common ancestor of s and t for which $m(s \wedge t)$ is largest. (Remember that $m(s)$ increases as the scale becomes finer.)

■ 2.2.2 The Multiscale Estimator and Error Model

The Markov property of the multiscale processes leads to an efficient algorithm for computing the LLSE estimate of $z(\cdot)$ at every node on the tree. Each measurement incorporated by the estimator is a noise-corrupted observation of $z(\cdot)$ at an individual node of the tree, i.e.,

$$y(s) = C_s z(s) + v(s)\,, \qquad v(s) \sim (0, R_s) \tag{2.27}$$

where $E\big[v(s)v(t)^T\big] = 0$ for $s \neq t$ and the measurement noise is uncorrelated with $z(\cdot)$ at all nodes on the tree. Because the fine-scale nodes of the multiscale process will

generally contain only local, fine scale information about the phenomenon modeled on the multiscale tree, fine-scale measurements of the phenomenon of interest will generally be modeled as observations of fine-scale nodes. Likewise, measurements at coarse-scale nodes will generally be equivalent to measurements of coarse-resolution or nonlocal functions of the finest-scale process.

The multiscale estimator [19], which is detailed in Appendix B, is a generalization of the Kalman filter and Rauch-Tung-Striebel smoother [82] for dynamic systems in time, i.e., processes given by Eq. (2.22) for a tree with $q = 1$. The first sweep of the estimator is a recursion from fine to coarse scale, which is then followed by a recursion from coarse to fine scale. The multiscale estimator returns $\hat{z}(\cdot)$, the LLSE estimate of the state at each node on the tree. As a by-product, the multiresolution estimator also produces the estimation error covariance $P_{e(s)}$ at every node, where the estimation error is defined to be $e(s) \triangleq z(s) - \hat{z}(s)$.

While there are numerous variations on the estimator equations provided in Appendix B, the number of computations for any estimator is basically a cubic function of the dimension of the state at each node. For example, if a tree has N nodes and the state dimension is constant and equal to d for each node, then the estimator requires $\mathcal{O}(Nd^3)$ computations to compute $\hat{z}(\cdot)$ and $P_{e(\cdot)}$ at each node[7]. For most processes of interest, the state dimension at the coarser scale nodes is not independent of the number of finest-scale nodes, and thus a crucial question for the efficiency of the multiscale implementation is how rapidly the state dimensions of the coarser scale nodes grow with N. For example, if $d = N^{2/3}$, then the number of computations grows cubicly with N. Another important question is for which stochastic processes and measurement equations do the state dimensions grow sufficiently slowly such that the multiscale estimator can be employed to solve very large problems.

While the multiscale estimator described in Appendix B computes the error covariance $P_{e(s)}$ for the estimate at each node, the "off-diagonal" terms of the full error covariance matrix—those terms $P_{e(s)e(t)}$ for $t \neq s$—are not computed. These terms can be computed individually using the error model for $e(s)$. As first derived in [61] and summarized in Appendix B, the errors of the multiscale estimator satisfy a multiscale autoregression in the form of Eq. (2.22), and the parameters of this autoregression follow directly from the estimator equations. This model can be used in conjunction with Eq. (2.26) to compute the cross-covariance between errors at different nodes.

The multiscale error model can also be used to compute sample paths of the multiscale process conditioned on the measurements. After estimation, the multiscale process can be decomposed as

$$z(s) = \hat{z}(s) + e(s). \tag{2.28}$$

The error $e(s)$ represents the uncertainty in the LLSE estimate $\hat{z}(s)$, and it can be generated independently of $\hat{z}(s)$ thanks to the orthogonality of the LLSE estimation

[7]Note that a tree with N_f nodes nodes at the finest scale has only $\mathcal{O}(N_f)$ nodes, and thus also requires only $\mathcal{O}(N_f d^3)$ computations to be estimated.

error. As demonstrated in Appendix B, the multiscale error model can be used to *efficiently* generate sample paths of $e(s)$ from a Gaussian distribution with the same second-order moments (covariance) as the error process. In the case when $z(s)$ and $y(s)$ are jointly Gaussian, $e(s)$ is also Gaussian, and the higher-order moments also are correctly represented. For each error process generated, a sample path of $z(s)$ is also generated. These samples are called *conditional simulations*, since the distribution from which these samples are drawn is equal to the probability distribution of $z(\cdot)$ conditioned on $y(\cdot)$ in the jointly Gaussian case. In Chapter 6, we show that the ability to quickly generate many sample paths is a useful property of the multiscale framework.

■ 2.3 Multiscale Realization Theory

If the multiscale framework is to be used as an alternative framework for performing LLSE estimation and related statistical analysis, then one must be able to build or *realize* multiscale models which have a particular probabilistic structure. The realization problem is to select the parameters of the autoregression, Eq. (2.22), and the parameters of the measurement equation, Eq. (2.27), such that the multiscale process $z(\cdot)$ and the measurements $y(s)$ have a desired joint probability distribution. (For LLSE estimators, we are only concerned about second-order moments, i.e., cross-covariances.) A number of multiscale models have been developed and successfully applied to applications in remote sensing [34] and computer vision [60]. For these models the focus is on the finest scale of the multiscale process, since only the finest-scale process is estimated and all the measurements are point samples of the finest-scale process. Two of these models are summarized in the following subsections. Also, the general realization algorithm described in [49] for realizing a multiscale process with an arbitrary finest-scale covariance structure is described. This subsection is also used to motivate the more general realization problem in which both measurements are provided at multiple resolutions and estimates are required at multiple resolutions. Extending the current methodology to handle the general realization problem is the subject of Chapter 4.

■ 2.3.1 Internal Multiscale Models

An important class of multiscale models are those for which every variable $z(\cdot)$ is a linear function of the finest-scale process, where the finest-scale process is given by those variables $z(t)$ for which t is a leaf node. Namely, if $f \sim (0, P_f)$ is a random vector containing all the finest-scale variables, then

$$z(s) = V_s f \tag{2.29}$$

for every node s. These models are called internal models [49], and each linear function $V_s f$ will be referred to as an *internal variable* and each matrix V_s as an *internal matrix*. For example, consider the case in which f contains samples of a continuous-index random process or field. These samples are generally the finest-scale at which the process is to be estimated or analyzed, so that the elements of f can be mapped to the finest-scale

nodes of the tree process. If the tree model is internal, then every variable will be a linear function of f.

Consider the set of multiscale tree models which have a desired finest-scale covariance P_f. The element of this set which leads to the most efficient estimation algorithm may not be an internal model. In other words, there may exist an external (not internal) multiscale tree model for which the finest-scale process has covariance P_f and the state dimensions are less than or equal to those of any internal model which has the same finest scale covariance [49]. However, internal models are useful for a number of reasons. For one, the difference in state dimension between the optimal multiscale model (measured in terms of estimator computations) and the optimal internal model will likely be negligible. Second, internal models are simple to analyze. For instance, the parameters P_0, A_s, and Q_s of an internal multiscale model can be expressed completely in terms of the internal matrices V_s and the covariance of f. Specifically, substituting Eq. (2.29) evaluated at $s = 0$ into $P_0 = E\left[z(0)\, z(0)^T\right]$ yields

$$P_0 = V_0 P_f V_0{}^T . \tag{2.30}$$

The parameters A_s and Q_s can then be computed by noting that Eq. (2.22a) is just the optimal prediction of $z(s)$ based on $z(s\overline{\gamma})$, plus the associated prediction error, i.e.,

$$z(s) = E\left[z(s) \,|\, z(s\overline{\gamma})\right] + w(s) . \tag{2.31}$$

Using standard equations from LLSE estimation, the model parameters follow as

$$A_s = P_{z(s)z(s\overline{\gamma})}\, P_{z(s\overline{\gamma})}^{-1} , \tag{2.32a}$$

$$Q_s = P_{z(s)} - P_{z(s)z(s\overline{\gamma})}\, P_{z(s\overline{\gamma})}^{-1}\, P_{z(s\overline{\gamma})z(s)} . \tag{2.32b}$$

Finally, the covariances in Eq. (2.32) follow from Eq. (2.29) as

$$P_{z(s)} = V_s P_f V_s{}^T , \tag{2.33a}$$

$$P_{z(s)z(s\overline{\gamma})} = P_{z(s\overline{\gamma})z(s)}^T = V_s P_f V_{s\overline{\gamma}}{}^T . \tag{2.33b}$$

The construction of an internal model generally consists of three steps: (i) mapping the components of f to leaf nodes of the tree, which also determines V_s for each of the finest-scale nodes; (ii) specifying the internal matrices V_s at the coarser-scale nodes; and (iii) computing the model parameters using Eqs. (2.30) and (2.32). Step (i) is generally straightforward, although the mapping can be affected by the nonlocal measurements to be represented (and the nonlocal functions of f to be estimated) at coarser-scale nodes. Also, as we have seen, step (iii) is generally straightforward once the covariances in Eq. (2.33) have been computed. Consequently, the core of constructing internal realizations is determining the internal matrices V_s and the resulting covariances in Eq. (2.33). Remember that the internal matrices cannot be chosen arbitrarily, since the resulting internal variables $z(s)$ must satisfy the Markov property of multiscale

trees. Also, the matrices V_s should have minimal row dimension, since the dimension of the resulting variables $z(\cdot)$ determines the efficiency of the multiscale estimator and likelihood calculator.

As discussed in [49], internal multiscale realizations can, in principle, be constructed for a finest-scale random process f with any desired covariance P_f. However, for an arbitrary P_f the ranks of the resulting internal matrices, which equal the dimensions of the corresponding state vectors $z(s)$, may be quite large and thus negate the computational advantage of the tree model. Fortunately, as developed in [35, 49, 62], there are large classes of processes for which either exact or adequately approximate multiscale realizations can be constructed that have sufficiently low dimension to make the multiscale formalism quite attractive. In the next subsection, such models for wide-sense Markov processes and MRFs [62] are described and will later be used to illustrate our methodology. Other examples of internal models are provided in Chapters 4 and 7. An external model is provided in Section 2.3.3.

■ 2.3.2 1D and 2D Wide-Sense Markov Processes

Because Markov processes and Markov random fields (Markov processes in 2D) are used for illustrative purposes throughout this thesis, this subsection describes them in detail. A discrete-time process $f[k]$ is a *bilateral* Markov process [25] if, conditioned on the values of $f[k]$ at the boundaries of any interval $I = [k_1, k_2]$, $k_2 > k_1$, the process inside the interval is uncorrelated with $f[k]$ outside the interval. The width of these boundaries depends on the order, n, of the process. To be more precise, define the following:

$$\begin{aligned} f_b &\triangleq \{f[k] \,|\, k \in [k_1 - n, k_1 - 1] \cup [k_2 + 1, k_2 + n]\}\,, \\ \widetilde{f}[k] &\triangleq f[k] - \widehat{E}\big[f[k] \,|\, f_b\big]\,, \\ \widehat{E}\big[f \,|\, y\big] &\triangleq \text{the LLSE estimate of } f \text{ from } y. \end{aligned}$$

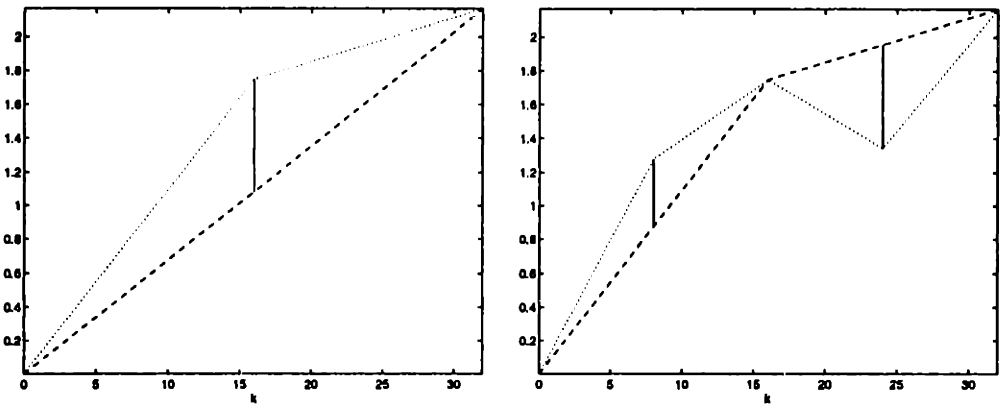
The vector f_b contains $f[k]$ at the boundaries of I for an n-th order process, while $\widetilde{f}[k]$ is the uncertainty in $f[k]$ after conditioning[8] on the boundary values. Then $f[k]$ is said to be n-th order bilateral Markov if

$$E\big[\widetilde{f}[l]\,\widetilde{f}[m]\big] = 0 \qquad \text{for all } l \in I \text{ and } m \notin I\,.$$

Similar to the boundary values, define f_p to contain n consecutive samples of $f[k]$, i.e., $f_p = \{f[k], f[k+1], \ldots, f[k+n-1]\}$. An n-th order ***unilateral*** Markov process is one for which

$$E\big[\check{f}[l]\,\check{f}[m]\big] = 0 \qquad \text{for all } l < k \le m\,,$$

[8] In this section, "conditioning f on y" refers to the effect of the reduction in uncertainty in f given by LLSE estimation; the LLSE estimation error process $\widetilde{f}[k]$ has less variance than the prior variance for $f[k]$.



(a) (b)

Figure 2.2. Steps one (a) and two (b) of the midpoint displacement algorithms for synthesizing Brownian motion. The dashed line provides the interpolation (LLSE estimate) of the process from the present boundary values, the solid line is the displacement of the midpoint(s), and the dotted line is the new interpolation.

where $\breve{f}[l] = f[l] - \widehat{E}[f[l] \mid f_p]$, i.e., conditioned on the n "present" values of $f[k]$, the past and future are uncorrelated. While not every bilateral Markov process is a unilateral Markov process, every unilateral process is a bilateral [25], so that any method for the multiscale modeling of bilateral Markov processes applies equally well to unilateral Markov processes.

The multiscale models described in [62] are based on the midpoint displacement algorithm for synthesizing Brownian motion [22]. The basic idea behind the midpoint displacement algorithm is that, given the values of a Markov process at the boundaries of any interval, the midpoint value of this interval can be synthesized independently of any values outside the interval. As an example, consider a first-order Markov process on the interval $[0, N]$. Given $f[0]$ and $f[N]$, then the midpoint value $f[k_0]$, where the "midpoint" k_0 can in fact be anywhere in the interval $[1, N-1]$, can be written as

$$f[k_0] = \widehat{E}[k_0 \mid f_b] + \widetilde{f}[k_0], \tag{2.34a}$$

$$= P_{f[k_0],f_b} P_{f_b}^{-1} f_b + \widetilde{f}[k_0], \tag{2.34b}$$

where $f_b = [f[0]\ f[N]]^T$. The second equality in Eq. (2.34) follows from standard LLSE formulas. The covariance matrices in Eq. (2.34b), as well as the covariance of $\widetilde{f}[k_0]$, are given by the statistics of the Markov process. Equation (2.34) can be interpreted as an interpolation from the boundary values plus a displacement $\widetilde{f}[k_0]$, as illustrated in Figure 2.2a for a sample path of Brownian Motion on the interval $[0, 32]$.

Once $f[k_0]$ is determined, we then have the boundary values of the two intervals $I_1 = [0, k_0]$ and $I_2 = [k_0, N]$. The values of the process at the midpoints of these two

intervals can again be generated by an interpolation and a displacement, i.e.,

$$f[k_1] = \widehat{E}\left[f[k_1] \,|\, f[0], f[k_0]\right] + \widetilde{f}[k_1], \tag{2.35a}$$

$$f[k_2] = \widehat{E}\left[f[k_2] \,|\, f[k_0], f[N]\right] + \widetilde{f}[k_2], \tag{2.35b}$$

for any "midpoints" $k_1 \in [1, k_0 - 1]$ and $k_2 \in [k_0 + 1, N - 1]$. More importantly, the two displacements $\widetilde{f}[k_1]$ and $\widetilde{f}[k_2]$ are uncorrelated due to Markovianity, and thus can be generated independently when the process is Gaussian. An example of the interpolation and displacement associated with these two samples is illustrated in Figure 2.2b.

After the first two steps of the midpoint displacement synthesis, $f[k]$ has been computed at the endpoints of four intervals. The synthesis process continues recursively by generating the midpoint values of the four intervals, each of which can be generated independently. In what follows, we describe how this recursive process can be represented by a multiscale autoregression. To simplify notation, assume that $N = 2^M$. Choosing $k_0 = N/2$ as the first midpoint, the state at the root node is given by

$$z(0) = \begin{bmatrix} f[0] \\ f[N/2] \\ f[N] \end{bmatrix}.$$

Modeling the process on a binary tree, the process values at the two descendents of the root node can also be chosen to contain three samples of $f[k]$. Namely, choose

$$z(0\alpha_1) = \begin{bmatrix} f[0] \\ f[N/4] \\ f[N/2] \end{bmatrix} \quad \text{and} \quad z(0\alpha_2) = \begin{bmatrix} f[N/2] \\ f[3N/4] \\ f[N] \end{bmatrix}.$$

The process noise generated when transitioning from scale $m = 0$ to scale 1, $w(0\alpha_1)$ and $w(0\alpha_2)$, will contain $\widetilde{f}[N/4]$ and $\widetilde{f}[3N/4]$, respectively. From Markovianity, these two vectors are uncorrelated with each other, and from the orthogonality of the LLSE they are uncorrelated with $z(0)$. This process can be continued recursively until the variables at a given level of the tree represent the entire interval of interest, as illustrated in Figure 2.3a for $N = 8$.

The multiscale models for 1D Markov processes are internal multiscale models, since each state $z(\cdot)$ simply contains samples of $f[k]$. Therefore, the parameters for the scale recursive autoregression follow from Eq. (2.32). However, note that the covariance matrices in Eq. (2.33) can be computed without explicitly computing P_f, the covariance of the entire Markov process which is to be represented at the finest scale of the tree. The ability to compute the model parameters without explicitly forming P_f is especially important for modeling 2D random fields [62].

There is considerable flexibility in modeling 1D Markov processes with the multiscale autoregression. The trees do not have to be binary, and the state dimension can vary from node to node, as illustrated in Figure 2.3c. (See [62] for further discussion.) However, the general procedure for developing an internal multiscale model of a Markov

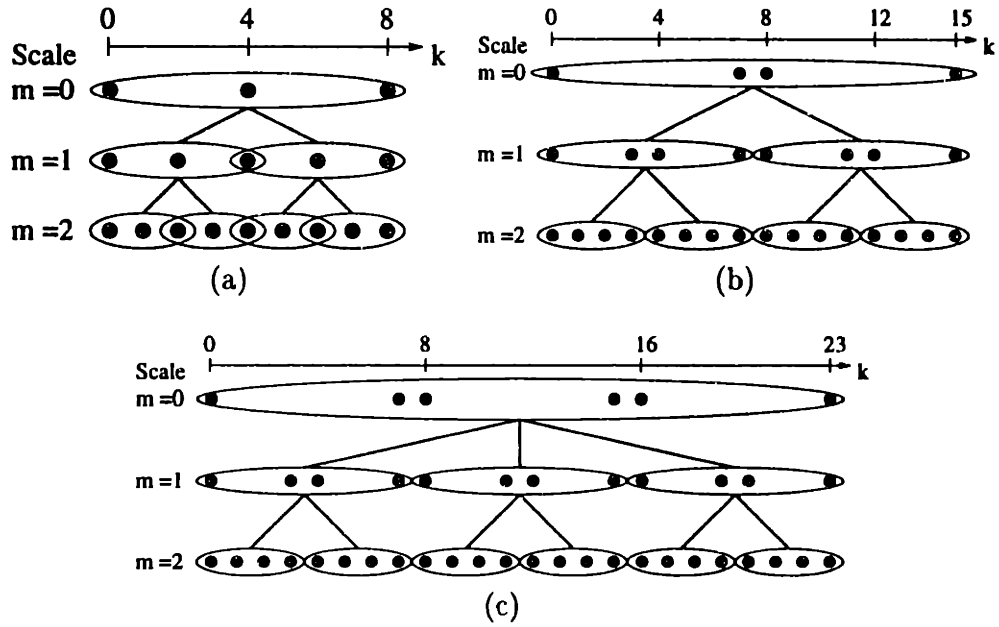


Figure 2.3. Multiscale models for first-order Markov processes, where each ellipse represents the samples of the Markov process which comprise the state $z(s)$ at a single node s. For each model, the state $z(s)$ is confined to sample values of the Markov process within the interval I_s. (a) A binary tree with a state dimension of three, (b) a binary tree with state dimension four, and (c) a tree where the nodes have varying state dimension and varying numbers of children.

process remains the same, i.e., forming states as samples of subintervals of the Markov process and then deducing the model parameters from Eqs. (2.30) and (2.32). This flexibility also holds in 2D, where the midpoint displacement algorithm can be generalized to develop internal multiscale models for Markov random fields [62]. A MRF is the 2D generalization of a 1D bilateral process. Namely, a wide-sense MRF is a 2D random process $f[i,j]$ for which the values of f in any connected set Ω are uncorrelated with the values of f outside this set when conditioned on the values of f on the boundary of Ω. Analogous to the multiscale models for 1D Markov processes, the states of the multiscale models for MRFs contain the values of f on the boundaries of subregions of the entire domain on which f is defined. In other words, if $\{f[i,j] \,|\, (i,j) \in \Omega_s\}$ is the finest-scale MRF descendent from node s, then $z(s)$ contains the values of $f[i,j]$ on the boundaries of subregions which cover Ω_s. The width of the boundaries is proportional to the order[9] of the MRF.

The root node variable $z(0)$ for a first-order or second-order MRF is shown in Figure 2.4a. The root node variable partitions the domain of interest into four conditionally uncorrelated subregions, $\{R_i\}_{1 \le i \le 4}$. The root node will thus have four children, each corresponding to one of the four sub-regions. These subregions can be further partitioned to determine the state variables at finer scales of the tree. Once all of the state

[9]The order of a Markov random field is defined in [14] using a hierarchy of neighborhoods.

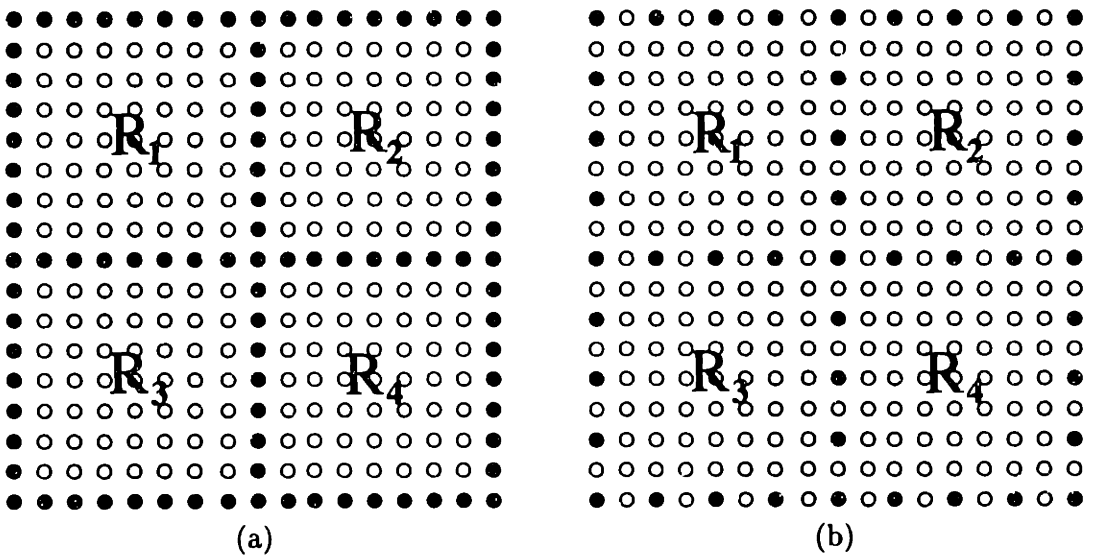


Figure 2.4. (a) The elements of the MRF process contained in the root node variable of the tree are illustrated with black circles (•), while the white circles (o) contain the rest of the process. (b) The root node variable for an approximate MRF tree model.

variables have been determined, the model parameters again follow from Eqs. (2.30) and (2.32).

The major difference between the multiscale models for 1D Markov processes and those for MRFs is that the state dimensions for MRFs grow with the size of the domain of the finest-scale process. The dimension of a state in a multiscale model of 1D Markov processes depends only on the order of the process and the number of children, q_s, descending from node s. For 2D MRFs, the dimension of the state $z(s)$ at a node s corresponding to some 2D region is proportional to the linear dimension of the boundary of that region. For example, if the finest-scale process is an N-by-N MRF, then the root node state illustrated in Figure 2.4a will have dimension $6N - 9$. Because the computational complexity of the multiscale estimator increases cubicly with each state dimension, an exact multiscale representation of an MRF will require at least $\mathcal{O}(N^3)$ computations to estimate. If the state dimension becomes prohibitively large, one possibility is to use the multiscale framework to approximate the statistics of the MRF. One such approximation is illustrated in Figure 2.4b, where only every other sample of the state from the exact model is represented at the root node of the approximate model. Such approximations are useful when the samples of the MRF are highly correlated over small spatial scales, so that the MRF process is effectively bandlimited. However, this and similar heuristic approximations are by no means optimal in terms of minimizing the state dimension required to match a desired level of fidelity in the covariance of the finest scale process [62, 70]. Generating such optimal approximations is in fact an open problem, and was the motivation for the Canonical Correlations based algorithms provided in [49].

2.3.3 Self-Similar and $1/f$-like Processes

Stochastic processes with self-similar distributions have been shown to be useful models for a wide range of physical phenomena [96]. Because these processes appear self-similar when analyzed at different scales, they seem to be a natural starting point for building useful classes of multiscale models. An important class of self-similar processes are those with power spectral densities with power law behavior, i.e., $P(j2\pi f) \sim f^{-\gamma}$. These processes are called $1/f$ processes.

A multiscale model proposed in [18] with $1/f$-like behavior is given by a tree with autoregression parameters

$$A_s = 1, \quad Q_s = \sigma^2 2^{-\gamma m(s)}, \tag{2.36}$$

where $P_0 = Q_0$. The dimension of the tree variables is equal to one, while the number of children per node and the number of scales can be chosen to suit the individual application. The process noise variance decays geometrically with scale, which results in a finest-scale process whose power spectrum can be shown to decay approximately as $1/f^{\gamma}$. While this model has been successfully applied as a prior model for both optical flow fields [60] and ocean-surface height interpolation [34], a few comments are in order.

- Like many $1/f$ processes, the finest-scale process is nonstationary, so that traditional notions of power spectral density do not apply.
- Realizations of the finest-scale process have visible discontinuities.
- The model is external, i.e., not internal.

The discontinuities can lead to troubling artifacts which do not represent the behavior of the corresponding physical phenomenon. However, if the density of the measurements at the finest scale is relatively large, the artifacts may not appear in the estimates. Also, as shown in Chapter 6, there are other applications for which the discontinuous behavior of the estimates is not necessarily an issue. While the model being external is by no means a drawback, note that no multiscale model with scalar state variables and nontrivial noise variances is internal; the reason is that no linear combination of the finest-scale states will exactly recover a coarser-scale state on the tree. Therefore, no internal multiscale model with scalar variables can be realized to have the same covariance as that for the finest-scale process defined by Eq. (2.36).

To overcome the limitations of the scalar model, higher-order processes can be developed. A class of multiscale models based upon self-similar $1/f$ processes, in particular, fractional Brownian motion, is the subject of Chapter 7.

2.3.4 Canonical Correlations

The methods used to realize the multiscale models for Markov processes and fBm are tailored to the particular characteristics of those processes and will not apply to more

general distributions. For a more general problem, assume that one is given the covariance of the vector f which is to be modeled at the finest scale of a multiscale process. In other words, the variables at the finest scale of the tree must have covariance P_f. The question whether or not it is possible to realize such a multiscale process is not the right one, for the single scale process $z(0) \sim (0, P_f)$ is a multiscale model with the correct finest-scale covariance. (Also, the multiple-scale process consisting of $z(s) = f$ at every node and the proper elements of f at the leaf nodes is also such a multiscale process, albeit with zero process noise.) However, the trivial model $z(0) \sim (0, P_f)$ amounts to a standard implementation of the normal equations, which is exactly what we were trying to avoid with the multiscale framework. A more useful problem is to design an optimal multiscale model, say the one which minimizes the sum of the cubes of the state dimensions

$$c(d) = \sum_{s \in \mathcal{S}_0} d(s)^3 , \tag{2.37a}$$

$$d(s) \triangleq \text{dimension of } z(s) , \tag{2.37b}$$

which is a meaningful measure of the efficiency of the multiscale framework. In general, for an exact multiscale representation of a process f with arbitrary covariance, the dimension of $z(0)$ will be proportional to the number of elements in the finest-scale process. If the number of elements is large, then again the solution of this realization problem does not allow one to circumvent the problems posed by a standard implementation of the normal equations.

This growth in the dimension of the coarser-scale variables with the number elements at the finest scale also holds for fBm. However, we have already pointed out that useful, low-dimensional approximations can be developed. For large problems, approximations not only make sense but they are absolutely necessary if one is interested in quantifying the uncertainty of the estimates. One general approximate realization problem is then to minimize $c(d)$ subject to $\|P_f - P_f^a\| < \epsilon$, where P_f^a is the covariance of the finest-scale process, $\|\cdot\|$ is a suitable norm quantifying the difference between two covariances, and ϵ is the maximum error tolerance. An alternative realization problem is to minimize $\|P_f - P_f^a\|$ subject to $c(d)$ less than some specified value. The solution to either of these problems has proved elusive. However, a very solid foundation was laid [49]. This realization algorithm has the following properties:

- the first step is to design the tree graph and map f to the finest-scale nodes; this step is done somewhat arbitrarily;
- for each node, a maximum state dimension λ_s is specified, i.e., $d(s) \leq \lambda_s \; \forall \; s \in \mathcal{S}_0$;
- the error in the covariance of the finest-scale process is minimized in a myopic, local sense.

The algorithm makes use of Canonical Correlations [2]. In the following, we show how Canonical Correlations is used, describe the nature of the myopic error minimization,

and finally target specific areas for improvement. The following tools are used throughout this thesis.

Because the variables of multiscale trees conditionally decorrelate subsets of random vectors, a useful realization tool is the ability to find the minimal set of information which decorrelates multiple random vectors. We first analyze the problem of decorrelating two random vectors. A fundamental question is what minimal set of information is required to decorrelate two random vectors ξ_1 and ξ_2. We will assume, as in [49], that this information must be a linear combination of $\xi^T = [\xi_1^T \ \xi_2^T]$. The first step is to develop a measure of the correlation between two random vectors. The generalized correlation coefficient used in [49] is

$$\overline{\rho}(\xi_1,\xi_2) \triangleq \max_{g_1,g_2} \frac{E\left[(g_1^T(\xi_1 - E[\xi_1]))(g_2^T(\xi_2 - E[\xi_2]))\right]}{\mathrm{var}(g_1^T\xi_1)^{1/2}\mathrm{var}(g_2^T\xi_2)^{1/2}}, \tag{2.38a}$$

$$= \max_{\left\{\substack{g_1^T P_{\xi_1} g_1 = 1 \\ g_2^T P_{\xi_2} g_2 = 1}\right\}} g_1^T P_{\xi_1\xi_2} g_2 \,. \tag{2.38b}$$

Also define the conditional correlation as

$$\overline{\rho}(\xi_1,\xi_2 \,|\, y) \triangleq \overline{\rho}(\widetilde{\xi}_1,\widetilde{\xi}_2)\,,$$
$$\widetilde{\xi}_i \triangleq \xi_i - \widehat{E}[\xi_i \,|\, y]\,.$$

To find the vector $W\xi$ of minimal state dimension such that $\overline{\rho}(\xi_1,\xi_2 \,|\, W\xi) = 0$, Canonical Correlations can be used. The Canonical Correlations decomposition is given by the following theorem [49]:

Theorem 1 *There exist matrices T_1 and T_2 such that*

$$\underbrace{\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}}_{T} \begin{bmatrix} P_{\xi_1} & P_{\xi_1\xi_2} \\ P_{\xi_2\xi_1} & P_{\xi_2} \end{bmatrix} \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}^T = \begin{bmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{bmatrix}, \tag{2.39}$$

where $T_i \in \mathbb{R}^{m_i \times n_i}$ and $D \in \mathbb{R}^{m_1 \times m_2}$ has the form

$$D = \begin{bmatrix} \widehat{D} & 0 \\ 0 & 0 \end{bmatrix}.$$

The matrix $\widehat{D}$ a positive-definite diagonal matrix $\widehat{D} = \mathrm{diag}(d_1, d_2, \ldots, d_{m_{12}})$ with $d_j \in (0,1]$ and elements ordered in descending order, i.e., $1 \geq d_1 \geq d_2 \geq \ldots \geq d_{m_{12}} > 0$.

The proof of this theorem is rather straightforward [49, 50]. The following theorem, called Proposition 6 in [49], allows one to determine the linear function of ξ with dimension less than or equal to λ that maximally decorrelates the two vectors ξ_1 and ξ_2.

Theorem 2 *Define λ to be the maximal number of elements used for decorrelation, and define $\mathcal{M}_\lambda$ to be the set of matrices with λ or fewer rows (and with a number of columns given by the context). For $i = 1, 2$ and $0 \leq \lambda < m_{12}$,*

$$\begin{aligned}\min_{W \in \mathcal{M}_\lambda} \overline{\rho}(\xi_1, \xi_2 \,|\, W\xi) &= \min_{W_i \in \mathcal{M}_\lambda} \overline{\rho}(\xi_1, \xi_2 \,|\, W_i \xi_i) \\ &= \overline{\rho}\Big(\xi_1, \xi_2 \,\Big|\, T_i\big(1 : \min(\lambda, m_{12}), :\big)\Big) \\ &= \begin{cases} d_{\lambda+1}\,, & \lambda < m_{12} \\ 0\,, & \lambda \geq m_{12} \end{cases}\end{aligned}$$

This theorem has a number of implications. First, note that instead of conditioning upon linear functions of ξ, one can condition upon linear functions of either ξ_1 or ξ_2 alone without adding to the dimension of the conditioning information. Second, the linear combination of ξ_1 (ξ_2) required for the decorrelation is given by the first m_{12} rows of T_1 (T_2) from Eq. (2.39). Third, the theorem provides a method for optimally decorrelating a pair of random vectors when the dimension constraint, λ, is less than m_{12}, the dimension required for complete decorrelation. Finally, since Canonical Correlations—the computation of either T_1 or T_2 in Eq. (2.39)—is the method used for decorrelating two random vectors, it is worthwhile to speak about the number of computations required to compute these matrices. As shown in [49], the computation of T in Eq. (2.39) requires the SVD of the three matrices P_{ξ_1}, P_{ξ_2}, and $P_{\xi_1\xi_2}$. Thus computing T_1 or T_2 requires $\mathcal{O}(n^3)$ computations if ξ has n elements.

The multiscale realization algorithm given in [49] uses Canonical Correlations to produce internal multiscale models that have a desired covariance at the finest scale. When no constraints are made upon coarser-scale variables, the internal variables can be chosen such that

$$z(s) = W_s f_s\,, \tag{2.40}$$

where f_s is defined to be the components of f represented at the finest-scale nodes descending from node s. In other words, the state at node s is a function only of its descendents at the finest scale. Under this assumption the Markov property reduces to the following: the $q_s + 1$ vectors

$$\widetilde{f}_{s\alpha_i} \triangleq f_{s\alpha_i} - \widehat{E}\big[f_{s\alpha_i} \,|\, W_s f_s\big]\,, \qquad i = 1, \ldots, q_s + 1\,, \tag{2.41}$$

are mutually uncorrelated. The vector $f_{s\alpha_{q+1}}$ is defined to be the vector of finest-scale elements that do not descend from node s. For this reason we sometimes find it useful to denote this vector as $f_{s^c} \triangleq f_{s\alpha_{q+1}}$, since f_{s^c} contains the finest-scale variables $\mathcal{S}_{s^c}$. Thus the realization problem boils down to solving for the linear combinations $W_s f_s$ that conditionally decorrelate the $q_s + 1$ sets of finest-scale variables partitioned by node s. The approach taken in [49] is to compute these matrices independently, which is why the algorithm is called myopic.

In order to derive W_s, we must first define the correlation between multiple random vectors $\xi_1, \xi_2, \ldots, \xi_q$. From [49] we have

$$\overline{\rho}(\xi_1, \ldots, \xi_q) \triangleq \max_{i \neq j} \overline{\rho}(\xi_i, \xi_j). \tag{2.42}$$

This function is a natural extension of the correlation between two random vectors, and a similar extension holds for the conditional correlation. To determine the linear combination of $\xi^T = [\xi_1^T, \ldots, \xi_q^T]$ that conditionally decorrelates the q sub-vectors ξ, consider the following theorem which is stated as Proposition 5 in [49]:

Theorem 3 *For $i = 1, 2$ and for all matrices W_i,*

$$\overline{\rho}(\xi_1, \xi_2 \,|\, W_i \xi) \leq \overline{\rho}(\xi_1, \xi_2). \tag{2.43}$$

In other words, conditioning on linear combinations of either ξ_1 or ξ_2 cannot increase the correlation between them. The state $W_s f_s$ can thus be formed by first finding for each $i = 1, \ldots, q_s$ the linear combination $W_{s\alpha_i} f_{s\alpha_i}$ which decorrelates $f_{s\alpha_i}$ from $f_{s\alpha_i^c}$. These linear combinations can be stacked columnwise into a single linear function of f_s. Using Eq. (2.43), this linear function can be shown to conditionally decorrelate the $q_s + 1$ vectors $f_{s\alpha_i}$. The model parameters then follow from Eqs. (2.30) and (2.32), making the proper substitution of $W_s f_s$ for $V_s f$.

While the algorithm of [49] provides a solid foundation for building multiscale realizations to have arbitrary covariances for the finest-scale process, there are still some significant obstacles to overcome.

- The myopic algorithm, which computes each matrix W_s independently, requires q Canonical Correlations factorizations of the finest-scale covariance for each node. With N elements at the finest scale, this implies $\mathcal{O}(qN^4)$ computations to compute the matrices W_s. This computational complexity applies even to approximate realizations.

- When the matrices W_s are computed independently, there is no way to enforce that approximations are done consistently, i.e., so that each variable $z(s)$ has discarded roughly the same set of information as its parent, $z(s\overline{\gamma})$.

- The repeated, sequential application of Canonical Correlation used to compute each W_s can lead to sub-optimal state dimensions, i.e., there may exist linear combinations of f_s with fewer elements but which also accomplish the desired decorrelation.

- The sub-optimality of such myopic approximations is often manifested by the design of an external model. (The parameters chosen by Eqs. (2.30) and (2.32) do not guarantee an internal model.)

- The approximations yielded by Canonical Correlations decompositions and the generalized correlation function $\overline{\rho}(\cdot)$ often lead to discontinuous artifacts near the boundaries of the vectors decorrelated.

- The generalized correlation function is not sensitive to the variances of ξ_1 and ξ_2, so that elements of these vectors which are highly correlated but which have little energy may be placed in the state variable of an approximate realization in favor of more significant components.

- The algorithm is focused entirely upon the finest scale process. For example, nonlocal measurements cannot be modeled at coarse-scale nodes.

Some of these items are addressed in this thesis. For example, the last item is the focus of Chapter 4. There is also the question of how broad a class random phenomena can be effectively modeled as a white-noise driven autoregression in scale. While this question is not the explicit focus of this thesis, it is addressed indirectly in both the multiscale modeling of Chapters 4-7 and the applications of Chapters 5-6

Chapter 3

Background and Preliminaries: Groundwater Flow and Hydraulic Conductivity Estimation

The development of accurate mathematical models describing the flow of groundwater is an important problem due to the prevalence of contaminants in or near groundwater supplies. Groundwater, which is stored in and travels through porous regions of the earth's subsurface, is a major source of fresh water. The potability of groundwater is often threatened by industrial and organic contaminants like pesticides and hydrocarbons. The presence of these impurities necessitates accurate predictions of how and when drinking supplies will be affected. A number of mathematical models, usually in the form of partial differential equations (PDEs), have been developed to describe the flow of groundwater and its place in the hydrogeologic cycle [6, 27, 38, 65]. The parameters of these equations are spatially distributed functions that describe the effect of the subsurface geology on the flow of groundwater. Because the earth's subsurface cannot be measured directly, except in select locations, knowledge of these parameters is limited either in spatial resolution or in spatial coverage. A problem, then, is to compute estimates of the parameters based on limited information and to characterize the uncertainty in these estimates. Uncertainty measures are necessary if any reasonable conclusions are to be drawn from the resulting groundwater models.

The goal of this chapter is to summarize the problem of estimating parameters of groundwater flow models and to motivate the application of the multiscale framework in Chapters 5 and 6. We begin by summarizing some of the equations commonly used to describe groundwater flow and transport. Hydraulic conductivity is an important parameter of these equations, and can be estimated from a number of measurement sources. We show how the Bayesian framework of Chapter 2 can be used to estimate hydraulic conductivity from measurements of head and conductivity. The problem with such estimators, however, is that standard parameterizations of the conductivity function lead either to poor representations of the finer scale measurements or to a prohibitively large number of parameters. We conclude with a discussion of the utility of parameterizations with multiple resolutions.

3.1 Equations of Groundwater Flow

Most equations of groundwater flow are based on conservation of mass and empirical laws. One such law is Darcy's Law [27], which relates the geology and changes in fluid potential to the volumetric flow of water. Specifically, Darcy's Law is

$$q(x) = -K(x)\nabla h(x), \tag{3.1}$$

where q is the specific discharge (m/s), K is the hydraulic conductivity (m/s), and h is the hydraulic head (m). The specific discharge is the volumetric water flux per unit area. Hydraulic head is a measure of the gravitational and pressure potential of the groundwater. Darcy's law essentially states that water flows downhill, hence the minus sign in Eq. (3.1), at a rate proportional to the gradient of hydraulic head. The rate of proportionality is *defined* as the hydraulic conductivity, which must be nonnegative. Hydraulic conductivity typically ranges from 10^{-2} to 10^{-8} m/s for sedimentary rocks, but can range as low as 10^{-9} to 10^{-13} m/s for clays [27]. Because hydraulic conductivity has such a large dynamic range, and because a limited number of studies have shown K to be log-normally distributed [69], it is often more convenient to work with log-conductivity in lieu of conductivity. In this case Darcy's Law becomes

$$q(x) = -e^{f(x)}\nabla h(x), \tag{3.2}$$

where $f = \ln K$ is log-conductivity.

Combining Darcy's Law with conservation of mass and assuming that the aquifer[1] is in steady-state yields

$$-\nabla \cdot \left(e^{f(x)}\nabla h(x)\right) = Q(x), \tag{3.3}$$

where Q represents the water source rate per unit volume, and hence has units s^{-1}. For steady-state flow, the source function Q typically represents recharge from rainfall or runoff. Note that steady-state does not imply no flow in the aquifer, but only means that the hydraulic head function does not vary over time. Equation (3.3) can be modified to account for transient flow, which can be induced by pumping or injecting fluids in wellbores. Pump tests are in fact an important tool for measuring local values of hydraulic conductivity, especially for petroleum engineering applications which naturally yield such measurements [9, 29, 66, 75, 76, 77].

Note that Eq. (3.3) is exactly analogous to the equation for electrostatic potential, where Q would be replaced by electric charges, K would be replaced by electric permittivity, and h would be replaced by electrostatic potential. Equation (3.3) is also used to model steady-state temperature distributions [11].

While Eq. (3.3) describes the flow of fluids in three dimensions, the same equation is used to describe fluid flow in fewer dimensions, such as one- or two-dimensional flow. In this case, either the variables of Eq. (3.3) are assumed to be constant over

[1] An aquifer is a permeable and contiguous geologic unit which contains water in its pore space.

one or two of the spatial dimensions, or the function $f(x)$ is some aggregate measure of log-conductivity given by averaging over one of the spatial distinctions. For 2D flow averaged vertically, the aggregate hydraulic conductivity function is called transmissivity [27]. In this thesis, i.e., in Chapters 5 and 6, only 1D and 2D flow are considered, but we will still refer to $f(x)$ as log-conductivity without explicitly defining whether it is an average value or just constant over the other spatial dimensions.

The steady-state flow equation does not have a unique solution, i.e., a unique mapping from Q to h, unless boundary conditions are specified. Two commonly used boundary conditions are Dirichlet and Neumann conditions. Define Ω to be the domain of interest and $\partial\Omega$ to be its boundary. Dirichlet conditions are in the form

$$h(x) = h_b(x) \qquad x \in \partial\Omega \tag{3.4}$$

where h_b is a known function of the boundary. Given these boundary conditions, then $h(x)$ can be determined uniquely everywhere in Ω [54]. Neumann conditions are in the form

$$-\left(e^{f(x)}\nabla h(x)\right) \cdot \hat{n}(x) = q_b(x) \tag{3.5}$$

and can be specified in lieu of Dirichlet conditions on parts[2] of $\partial\Omega$. The function $\hat{n}(x)$ is the unit normal vector to the boundary of Ω at x, and thus $q_b(x)$ is the net water flux across the boundary at x. Dirichlet and Neumann conditions also serve a practical role in groundwater modeling. When streams or lakes form the boundaries of aquifers, then Dirichlet conditions can represent these boundaries. Furthermore, highly impermeable ($K \approx 0$) geologic barriers can be modeled by Neumann conditions with $q_b = 0$; these boundaries are called no flux boundaries.

The forward problem in groundwater flow is the determination of h given Q, K, and appropriate boundary conditions. Only under very special circumstances can h be determined analytically, meaning that numerical methods must be employed. The most common numerical methods, finite-element and finite-difference, replace Eq. (3.3) and the boundary conditions by a linear system of equations

$$Au = b\,,$$

where u is a discrete representation of h, b is a discrete representation of Q and the right-hand side of the boundary conditions, and A represents the differential operator $-\nabla\cdot K\nabla$ and the left-hand side of the boundary conditions. Because this differential operator is self-adjoint and positive-definite, A will generally be symmetric and positive definite. Also, because the differential operator is local, A will be sparse. Sparse, positive-definite systems of equations can be solved efficiently, even when the dimension of u is very large, using iterative methods like conjugate gradients [5] or nested dissection [40].

[2] If $h_0(x)$ is a function which satisfies Eqs. (3.3) and (3.5), then $h_0(x)+c$ for any constant c will also satisfy these equations.

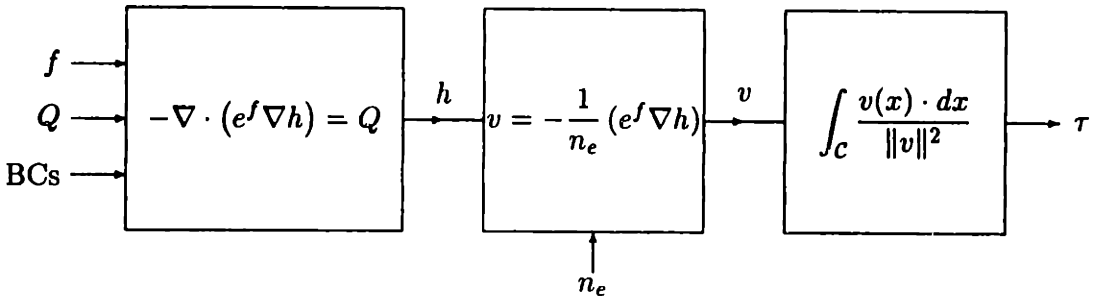


Figure 3.1. The determination of travel time from hydraulic conductivity.

The efficiency of the conjugate gradient method is limited by the condition number of A, which is a function of the variability in K [80].

For most groundwater problems, the forward solution alone is not of particular interest. One is usually more interested in variables like the velocity field or the time for a particle to travel a particular distance in the aquifer, both of which can be derived from the forward solution. The velocity of water in steady-state is given by [27]

$$v(x) = \frac{1}{n_e(x)} q(x), \tag{3.6}$$

where n_e is the effective porosity. Groundwater typically travels on the order of a foot per year; a typical value of effective porosity for sedimentary rocks is 0.05 [27], and porosity typically has much less spatial variation than does conductivity. Assuming the porosity function is well known or can be effectively estimated, then $v(x)$ follows directly from the forward solution. Ignoring diffusion, the velocity field can then be used to determine how long is required for a contaminant to travel from one point to another in the aquifer. If C is the path from point x_0 to point x_1 determined by $v(x)$, then the time to travel between these two points is given by

$$t_{\text{cp}} = \int_C \frac{v(x) \cdot dx}{\|v\|^2}. \tag{3.7}$$

The process of calculating t_{cp} from Q, K, and the boundary conditions is illustrated in Figure 3.1. A procedure for estimating travel time in an aquifer is provided in Chapter 6.

■ 3.2 Hydraulic Conductivity Estimation

The solution of the groundwater flow equation, i.e., the forward problem, requires a complete specification of the hydraulic conductivity function $K(x)$, the source input

$Q(x)$, and appropriate boundary conditions. The problem is that these functions can never be known perfectly, since the earth's subsurface can be observed directly only at select locations. For the applications considered in this thesis, we will assume that the source function and boundary conditions are known perfectly, so that only the hydraulic conductivity function in unknown. This assumption is common [1, 23, 38, 86, 68, 97, see Table 1] and can be justified on two accounts. One, and most important, all of the estimators considered in this thesis can be readily extended to the case in which Q and the boundary conditions are not known perfectly and must be estimated in conjunction with hydraulic conductivity. Second, the source function and boundary conditions can sometimes be estimated separately from hydraulic conductivity within a reasonable degree of accuracy. For example, the source function Q can be determined from soil moisture content, rain gauges, and satellite measurements [46, 81]. Also, the boundary conditions can sometimes be inferred from geologic boundaries or known water tables like streams and lakes [69].

The estimation of hydraulic conductivity has been a subject of intense interest for the last thirty years. The petroleum community proposed automated methods for hydraulic conductivity estimation[3] in the 1960's [52, 53]. For petroleum reservoirs[4] in which fluids had been extracted over an extended period of time, petroleum engineers sought to infer the hydraulic conductivity function based upon changes in pressure (head) over time. The process of choosing a conductivity function which reproduces the pressure histories recorded at wells is sometimes referred to as history matching, and much work has been devoted to this problem [9, 13, 16, 29, 66, 75, 76, 77, 78, 88]. As numerical models for simulations of petroleum reservoirs have become more sophisticated [3], so has the problem of estimating their parameters become more difficult. Reservoir modelers now must rely upon a variety of measurement sources, including seismograms [24], electro-magnetic tomography [92], and nuclear magnetic resonance. While groundwater hydrology has benefited greatly from this work, the estimation of hydraulic conductivity in the context of groundwater flow is different for two major reasons. First, forcing (pumping) the aquifer is usually not allowed, so as to minimize the displacement of contaminants. Second, the groundwater problem is on a much smaller spatial (and monetary[5]) scale. Groundwater studies can involve aquifers with vertical extents of only a few meters [98], while typical petroleum reservoirs extend to depths of thousands of meters. Hydraulic conductivity estimation in the groundwater community also differs in its heavy use of statistical methods [10, 23, 68, 74, 98]. The use of statistical methods has also been summarized in a number of recent review papers [42, 69, 97]. The utility of statistical methods for estimating hydraulic conductivity is

[3]In the context of petroleum engineering, permeability is usually used in lieu of hydraulic conductivity. The two parameters are related by a straightforward change of dimensions.

[4]The porous geologic body in which hydrocarbons are stored is often referred to as a reservoir rather than an aquifer, although the geologic properties of the two bodies are usually identical.

[5]While the petroleum business is much larger than the business of environmental cleanup, the Superfund project administered by the EPA presently adds one billion dollars annually to its endowment and has identified 35,000 potential hazardous waste sites. (Source: EPA)

widely recognized, since computing an estimate of hydraulic conductivity is not an end in itself [98], but a means for using the resulting flow model to draw more high-level inferences. For instance, one might need to estimate the time it takes the contaminant to travel from point x_0 to point x_1. The uncertainty in such an estimate is directly related to the uncertainty in the estimated hydraulic conductivity function. (See Chapter 6 for further discussion of the travel time problem.)

Estimating hydraulic conductivity is difficult for two primary reasons — the natural variability of hydraulic conductivity and the difficulty in measuring it. Hydraulic conductivity is known to vary by orders of magnitude over spatial scales from centimeters to kilometers [38]. The only way to directly measure hydraulic conductivity is to extract a small piece of the earth, apply a head differential, and then measure the resulting flow. Such core sampling is rarely done at more than a few well locations; instead, a small amount of water is usually injected into the earth at select locations in a wellbore, after which local conductivity values can be inferred from changes in head. To infer values of the hydraulic conductivity function away from the wells, indirect measurements must be used. One indirect measurement is the seismogram, which determines stratigraphic (structural) layers of the geologic environment by measuring the reflection of acoustic waves off geologic boundaries. Because seismograms have a coarse resolution that is typically larger than the vertical extent of an aquifer, they are usually only used to determine stratigraphic layering and the natural boundaries of an aquifer. Also, seismograms are difficult to correlate with values of hydraulic conductivity [47]. The two indirect measurement sources considered in this thesis are measurements of hydraulic head and tracer tests. Hydraulic head measurements are also taken at wellbores, but they are not sensitive to point values of conductivity [38]. Instead, they supply coarse-resolution information about conductivity and are related to the conductivity function by Eq. (3.3). Tracer tests measure the time a "marked" particle takes to travel from one wellbore to another, and thus measure the fluid velocity between wellbores — see Equation (3.7).

If the hydraulic conductivity function is to be estimated from direct (fine scale) measurements of hydraulic conductivity, measurements of head, and tracer tests, then a number of problems must be addressed.

- The measurements are sparse, and therefore do not fully constrain the conductivity function.
- The measurements are at multiple resolutions.
- The measurements are typically corrupted by measurement noise.
- The head measurements are ill-conditioned.
- The head samples and tracer tests are nonlinearly related to the conductivity function by a PDE.

The sparsity of the measurements means that we cannot hope to estimate the conductivity function at all spatial scales. Instead, additional information must be supplied, often in the form of a prior distribution for $f(x)$. Another way to constrain the estimate, which is necessary if computational methods are to be employed, is to restrict the estimate to lie in a finite-dimensional subspace $\text{span}\{\phi_j(x)\}_{1\leq j\leq N}$ [69, 97]. In this case, the estimator satisfies

$$\hat{f}(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x) . \tag{3.8}$$

The choice of functions ϕ_j is an important problem. These functions must be chosen such that the measurements can be accurately represented; for example, if a slug test measures the average value of conductivity over some small region, one of the functions ϕ_j could be chosen to be constant over this region and zero elsewhere. In this case, however, the function space cannot be chosen to have constant resolution, otherwise N will be prohibitively large. A discussion of how to choose the functions ϕ_j based upon the measurement resolutions and locations is supplied in Section 3.6. The measurement noise can also be accounted for using a statistical estimator. Explicitly accounting for measurement noise is especially important when the measurements are ill-conditioned. The ill-posedness of estimating conductivity from head measurements, as well as how to use Eq. (3.3) to relate samples of head to the conductivity function, are discussed in the following section. Bayesian estimators are then applied to conductivity estimation in Section 3.5. A motivation for using the multiresolution estimator is discussed in Section 3.6.

3.3 Relating Head Measurements to Hydraulic Conductivity

Because travel times are not considered until Chapter 6, we focus here on the estimation of hydraulic conductivity from direct measurements conductivity and head measurements. Assuming the direct measurements — core samples and slug tests — are functions of point values of conductivity, the direct measurements can be represented as

$$y_i^f = f(x_i^f) + v_i^f , \quad i = 1, \ldots , M_f \tag{3.9}$$

where the noise vector v_i^f represents not only measurement error but scale mismatches between the support of the measurement source and that of a point [69]. The superscript f is used to distinguish the variables of the conductivity measurements from those of the head measurements. The head measurements are in the form

$$y_i = h(x_i) + v_i , \quad i = 1, \ldots , M_h . \tag{3.10}$$

To understand how head measurements constrain the conductivity function, first note that the estimation of hydraulic conductivity from head measurements alone is an

ill-posed problem. Even if the head function is known everywhere on Ω, the conductivity function $f(x)$ cannot be determined uniquely without specifying a value of $f(x)$ on each streamline of the flow field [67]. If a pumping well is present and the flow rate of the well is known, then $f(x)$ is implicitly specified along each streamline emanating from the well (using Darcy's law and the knowledge of $h(x)$). Even more importantly in our context, point values of head are relatively insensitive to local variations in hydraulic conductivity. To illustrate this insensitivity, consider the following two-dimensional flow scenario:

- Ω is the unit square, $(x_1, x_2) \in [0,1] \times [0,1]$;
- $Q = 0$ on Ω;
- the boundary conditions are $h = 1$ for $x_1 = 0$, $h = 0$ for $x_1 = 1$, and $dh/dx_2 = 0$ for $x_2 = 0$ and $x_2 = 1$.

For $f = 0$ on Ω, the corresponding head function is plotted in Figure 3.2. For $f = 0$ the specific discharge vector $q(x)$ has magnitude one and points in the positive x_1 direction. Now consider the log-conductivity function plotted in Figure 3.3a, which is a sample path of a zero-mean random field. The corresponding head function is plotted in Figure 3.3b. Note that, despite the large variability of $f(x)$, the head function is still rather smooth and exhibits much less variation. This smoothing is due to the integration of $e^{f(x)}$ implied by Eq. (3.3) in order to compute $h(x)$. In the context of estimating $f(x)$ from measurements of $h(x)$, this smoothing means that the effects of measurement noise must be accounted for, otherwise small perturbations in the head samples will be confused with large variations in conductivity.

We now develop a more rigorous description of the relationship between samples of $h(x)$ and the function $f(x)$, so that measurements of hydraulic head can be used to estimate the hydraulic conductivity function. In Section 2.1.2, we showed how both the LLSE estimator and the MAP estimate can be implemented by linearizing Equation (2.1). Recall that the LLSE estimate in this case is based upon an approximate measurement model. To express Eq. (3.10) in a form analogous to Eq. (2.1), recall that the mapping from $f(x)$ to $h(x)$ is unique when the source function and boundary conditions are known. Thus we can write

$$h = \mathcal{F}_h(f)\,,$$

where $\mathcal{F}_h$ is the forward operator mapping f to h. The pointwise evaluation of this mapping can be written

$$h(x_i) = \mathcal{F}_h(f, x_i)\,, \tag{3.11}$$

which stresses that the sample $h(x_i)$ is a function of conductivity over the entire domain Ω. Using this mapping, the measurement equation becomes

$$y_i = \mathcal{F}_h(f, x_i) + v_i\,, \tag{3.12}$$

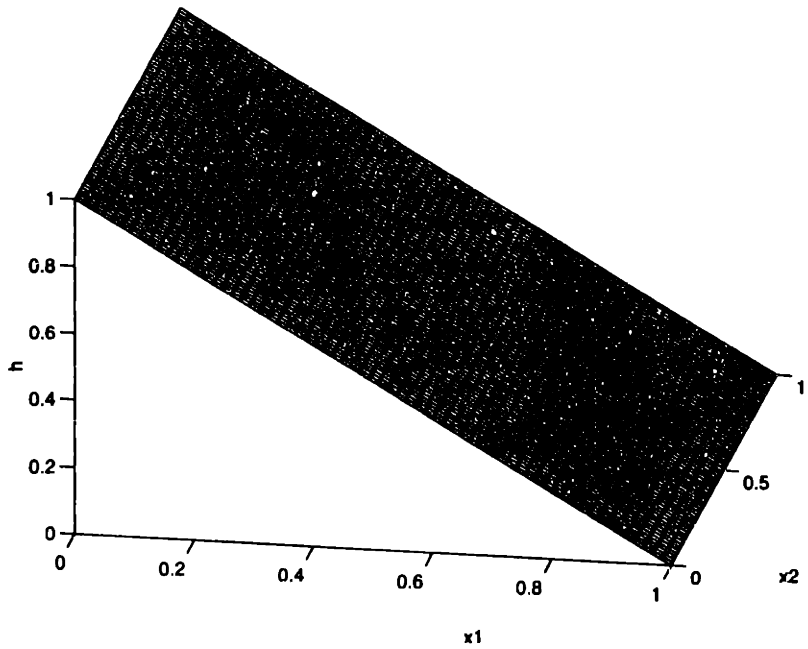


Figure 3.2. The head function for 2D flow when log-conductivity is constant and equal to zero.

which looks much more like Eq. (2.1). One difference is that f in Eq. (2.1) is a vector, whereas f in Eq. (3.12) is a function. We now show how to compute the linearization of this measurement equation.

Much like the Taylor Series expansion given in Eq. (2.15), a Taylor series expansion exists for the functional operator in Eq. (3.11). For an expansion about the function $f_0(x)$, the Taylor series has the form [73]

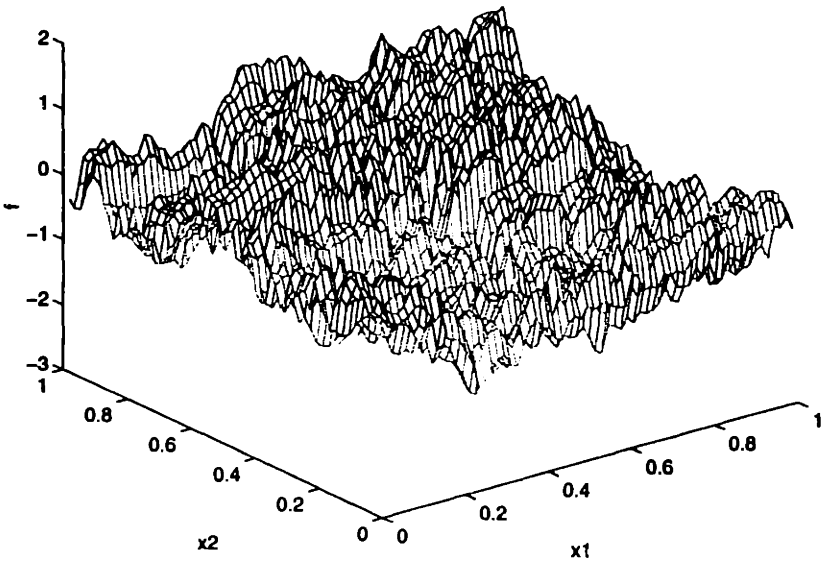
$$\mathcal{F}_h(f, x_i) = \mathcal{F}_h(f_0, x_i) + \langle \frac{\partial \mathcal{F}_h(f_0, x_i)}{\partial f}, f - f_0 \rangle + \text{h.o.t.} \tag{3.13}$$

where $\mathcal{F}_h(f_0, x_i)$ is the head value at x_i when the conductivity function is equal to f_0, $\partial \mathcal{F}_h(f_0, x_i)/\partial f$ is the Fréchét derivative of $\mathcal{F}_h(f, x)$ evaluated at $(f, x) = (f_0, x_i)$, and "h.o.t." denotes terms of higher order in $(f - f_0)$. Note that the Fréchét derivative is a function on Ω and that $\langle \cdot, \cdot \rangle$ denotes the inner product

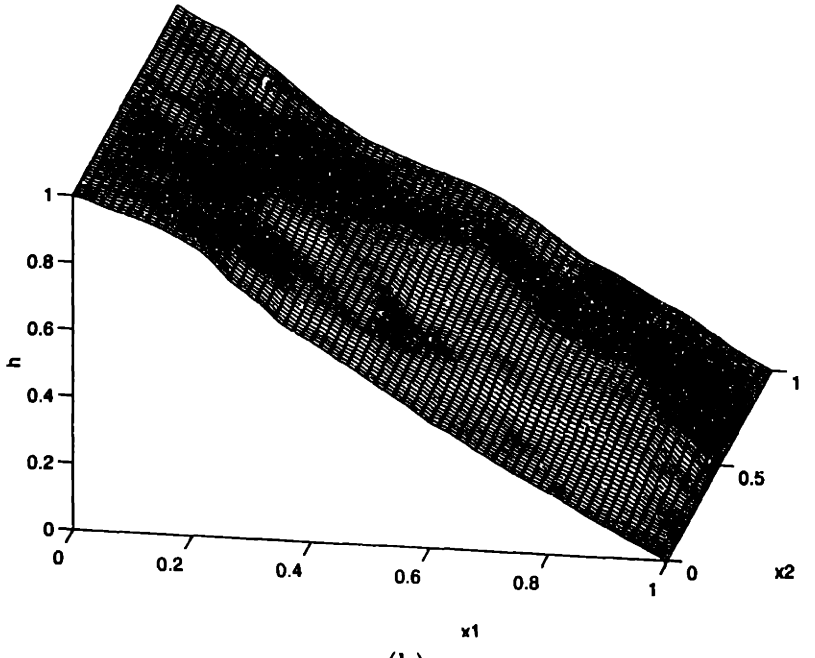
$$\langle h, f \rangle = \int_\Omega h^*(x) f(x)\, dx\,. \tag{3.14}$$

Substituting Eq. (3.13) into Eq. (3.12) yields the linearized measurement equation

$$\underbrace{y_i - \mathcal{F}_h(f_0, x_i) + \langle \frac{\partial \mathcal{F}_h(f_0, x_i)}{\partial f}, f_0 \rangle}_{\mathcal{Y}_i(f_0)} = \langle \frac{\partial \mathcal{F}_h(f_0, x_i)}{\partial f}, f \rangle + v_i + \text{h.o.t} \tag{3.15}$$



(a)



(b)

Figure 3.3. For two-dimensional flow, (a) a log-conductivity function with mean zero and (b) the corresponding head function.

where the higher-order terms can be ignored if the linearization is sufficiently accurate. Note the similarity between this measurement equation and Eq. (2.16). Note also that this linearization is completely general, and can be extended to measurements of other indirect observations of f, like travel times. The only constraint is that one must be able to compute the Fréchét derivative.

Ignoring questions of existence, methods for calculating the Fréchét derivative generally fall into two categories — perturbation and adjoint methods [26]. Perturbation methods assume that the function f has already been projected onto the finite-dimensional subspace $\text{span}\{\phi_j(x)\}_{1\le j\le N}$, i.e., $f = \sum_{j=1}^{N} \alpha_j \phi_j$. In this case, a finite-difference approximation can be used to calculate the sensitivity of $h(x_i)$ to each variable α_j. This finite-difference is given by

$$\frac{\partial \mathcal{F}_h(f_0, x_i)}{\partial \alpha_j} \approx \frac{\mathcal{F}_h(f_0 + \Delta\alpha_j\, \phi_j, x_i) - \mathcal{F}_h(f_0, x_i)}{\Delta\alpha_j}, \qquad j = 1, \ldots, N. \tag{3.16}$$

If there are M_h head measurements and $\underline{h}$ is a vector containing the corresponding head samples, then the finite-difference approximations together yield an approximation of the Jacobian[6] $\nabla_\alpha \underline{h}$. For our purposes, there are two significant drawbacks to the perturbation approach. First and foremost, the Jacobian requires $N + 1$ forward simulations of the flow equations, independent of the number of head measurements. These forward simulations can be quite costly if N is large. Second, the derivatives computed by the finite-difference approximation are sensitive to the magnitude of each $\Delta\alpha_j$. Adjoint methods overcome both of these drawbacks.

Adjoint methods derive a linearization directly from the underlying partial differential equations and then use Green's functions to compute the Fréchét derivative. The first step is to determine from the flow equations how a small perturbation in f, call it δf, relates to the change in the head function δh. Assume that Dirichlet conditions are imposed upon $\partial\Omega_D$ and Neumann conditions are imposed upon $\partial\Omega_N$, where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. Also define $h_0 = \mathcal{F}_h(f_0)$. For a small perturbation in conductivity about the conductivity function f_0, the first variation in head can be shown to satisfy (see Appendix C)

$$-\nabla \cdot e^{f_0} \nabla \delta h = \nabla \cdot e^{f_0}\, \delta f\, \nabla h_0, \qquad x \in \Omega \tag{3.17a}$$

$$\delta h = 0, \qquad x \in \partial\Omega_D \tag{3.17b}$$

$$-e^{f_0} \nabla \delta h \cdot \hat{n} = e^{f_0}\, \delta f\, (\nabla h_0 \cdot \hat{n}), \qquad x \in \partial\Omega_N \tag{3.17c}$$

which is a linear relationship between δf and δh. The Fréchét derivative is the function $g(x_i, x')$ for which $\delta h(x_i) = \langle g, \delta f \rangle$. This function can be derived using Green's function theory [44]. The Green's function is defined as the response of the adjoint of Eq. (3.17) to a unit impulse. Because the steady-state flow equation is self-adjoint, the Green's

[6] We will use subscripts for the Jacobian/gradient operator ∇ whenever the variables to which the differentiation is made is unclear.

function satisfies

$$-\nabla_{x'} \cdot e^{f_0(x')} \nabla_{x'} G(x_i, x' \mid f_0) = \delta(x_i - x'), \qquad x' \in \Omega \tag{3.18a}$$

$$G(x_i, x' \mid f_0) = 0, \qquad x' \in \partial\Omega_D \tag{3.18b}$$

$$-e^{f_0(x')} \nabla_{x'} G(x_i, x' \mid f_0) \cdot \hat{n} = 0, \qquad x' \in \partial\Omega_N \tag{3.18c}$$

where the notation $G(x_i, x' \mid f_0)$ emphasizes that the Green's function depends upon the "point" of linearization f_0. The Green's function representation of $\delta h(x_i)$ follows as

$$\delta h(x_i) = \int_\Omega G(x_i, x' \mid f_0) \, \nabla_{x'} \cdot \left(e^{f_0(x')} \, \delta f(x') \, \nabla_{x'} h_0(x') \right) dx' + \int_{\partial\Omega_N} G(x_i, x' \mid f_0) \, e^{f_0(x')} \left(\nabla_{x'} h_0(x') \cdot \hat{n}(x') \right) \delta f(x') \, dx' . \tag{3.19}$$

To express $\delta h(x_i)$ as an inner product of the Fréchét derivative and δf, we invoke Green's theorem and assume for notational simplicity that $\delta f = 0$ on $\partial\Omega$; this leads to

$$\delta h(x_i) = \int_\Omega e^{f_0(x')} \left(\nabla_{x'} G(x_i, x' \mid f_0) \cdot \nabla_{x'} h_0(x') \right) \delta f(x') \, dx' , \tag{3.20a}$$

$$\frac{\partial \mathcal{F}_h(f_0, x_i)}{\partial f}(x') = e^{f_0(x')} \left(\nabla_{x'} G(x_i, x' \mid f_0) \cdot \nabla_{x'} h_0(x') \right) . \tag{3.20b}$$

The advantage of the adjoint approach is that only $M_h + 1$ forward simulations are required for M_h head measurements.[7] Because the number of head measurements is usually much smaller than the number of variables α_j used to represent the conductivity estimate, i.e., $M_h \ll N$, the adjoint method leads to dramatic computational savings for large values of N. Another advantage of the adjoint method is that it does not depend upon the choice of basis functions ϕ_j. Examples of the Fréchét derivative for 1D and 2D flow are provided in the following section.

■ 3.4 Examples of Fréchét Derivatives for Head Measurements

To understand how head measurements impact estimates of hydraulic conductivity, it is worthwhile to consider some example Fréchét derivatives. While the Fréchét derivative in Eq. (3.20b) appears fairly innocuous, it varies with

- the spatial dimension of the flow (only 1D and 2D flow are considered in this thesis),

[7] Actually, one forward simulation of the flow equation is required for h_0, and M_h simulations of the adjoint of the flow equation are required for the M_h functions $G(x_i, x' \mid f_0)$. However, the flow equation is self-adjoint, meaning that it is equal to its adjoint [44].

- the boundary conditions,
- the point of linearization f_0,
- the source function Q, and
- the location of the head measurements with respect to the boundaries of Ω and any pumping or injection wells.

1D Flow

First consider the case of 1D flow with $f_0 = 0$, $Q = 0$, and Ω equal to the unit interval $[0, 1]$. For the boundary Dirichlet conditions $h(0) = 1$ and $h(1) = 0$, the Fréchét derivatives for $x_i = i/8$, $i = 1, 4, 7$, are plotted in Figure 3.4. Note that the Fréchét kernels are nonzero over the entire unit interval, indicating that each head measurement is sensitive to the entire conductivity function $f(x)$. In other words, head samples are non-local, coarse-scale functions of conductivity. Also note that the Fréchét derivatives are positive upstream from the point of measurement and are negative downstream from the point of measurement. The reason is that if conductivity is increased (relative to f_0) upstream from x_i, then the difference $h(0) - h(x_i)$ will be less than the difference $h_0(0) - h_0(x_i)$. Since the value of h at $x = 0$ is fixed, this implies and increase in $\delta h(x_i)$. The reason for the negative downstream values is similar. Another interesting feature is that each of the Fréchét kernels integrates to zero, meaning that head measurements in this flow scenario cannot be used to determine DC values of conductivity.

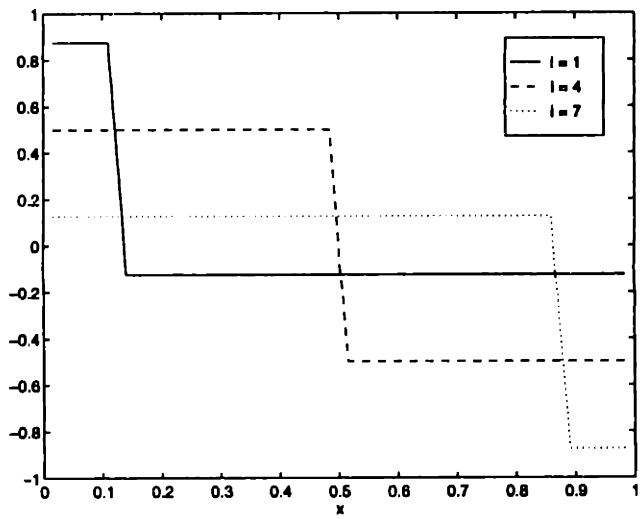


Figure 3.4. The Fréchét derivatives at $x_i = i/8$, $i = 1, 4, 7$, for the 1D flow equation when linearized about the log-conductivity function $f_0 = 0$. The boundary conditions are $h(0) = 1$ and $h(1) = 0$.

To see the effect of the choice of boundary conditions upon the Fréchét derivative, consider the same flow conditions but with $h(0) = 1$ replaced by the Neumann condition

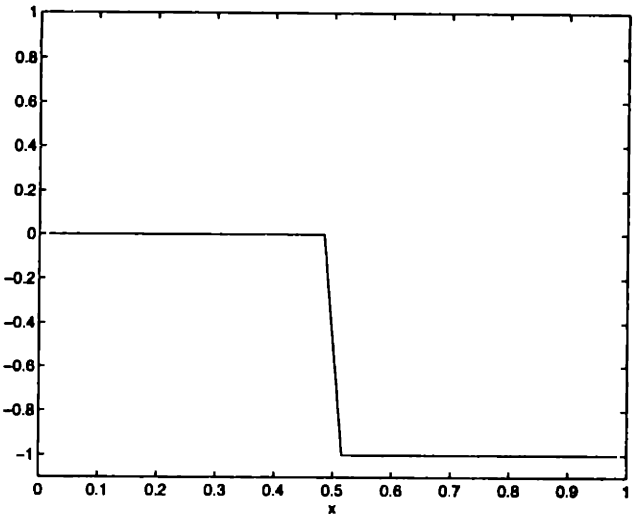


Figure 3.5. The Fréchét derivative at $x_i = 1/2$ for the 1D flow equation when linearized about the log-conductivity function $f_0 = 0$. The boundary conditions are $dh/dx = -1$ at $x = 0$ and $h(1) = 0$.

$dh/dx = -1$ at $x = 0$. For a head measurement at $x_i = 1/2$, this flux condition leads to the Fréchét derivative plotted in Figure 3.5. The Fréchét derivative is now nonzero only for downstream locations, because the head value at $x = 0$ is no longer constrained. The head values in the scenario are insensitive to upstream values of conductivity. However, unlike the previous scenario, the Fréchét kernel does not integrate to zero and the head measurements can be used to estimate the average value of f.

To illustrate the effect of the point of linearization on the Fréchét derivative, consider the original 1D flow scenario ($h(0) = 1$) but with $f_0(x) = 0.5\sin(2\pi x)$. The Fréchét derivative, as well as the original Fréchét derivative for $f_0 = 0$ are plotted in Figure 3.6. The shape of the function f_0 affects the shape of the Fréchét derivative, which would be expected from inspecting Eq. (3.20b).

2D Flow

The Fréchét derivatives for 2D flow appear quite different from those for 1D flow, in part because flow is less constrained in 2D. Consider 2D flow on the unit square $\Omega = [0,1] \times [0,1]$ with $f_0 = 0$, $Q = 0$, and the following boundary conditions[8]: $h = 1$ for $\mathbf{x}_2 = 1$, $h = 0$ for $\mathbf{x}_2 = 0$, and $dh/d\mathbf{x}_1 = 0$ for $\mathbf{x}_1 = 0$ and $\mathbf{x}_1 = 1$. This flow scenario is illustrated in Figure 3.7, which notes that $dh/d\mathbf{x}_1 = 0$ is equivalent to specifying no flow across the boundaries $\mathbf{x}_1 = 0$ and $\mathbf{x}_1 = 1$. The Fréchét derivatives evaluated at $x_i = (0.5, 0.5)$ and $(0.5, 0.08)$ are plotted in Figures 3.8a and 3.8b, respectively. A number of similarities between these functions and the Fréchét derivatives for 1D flow should be noted. First, the 2D Fréchét derivative is essentially positive for upstream

[8]For two-dimensional flow, the two spatial dimensions will be denoted in boldface as $\mathbf{x}_1$ and $\mathbf{x}_2$ in order to distinguish them from the measurement locations x_i.

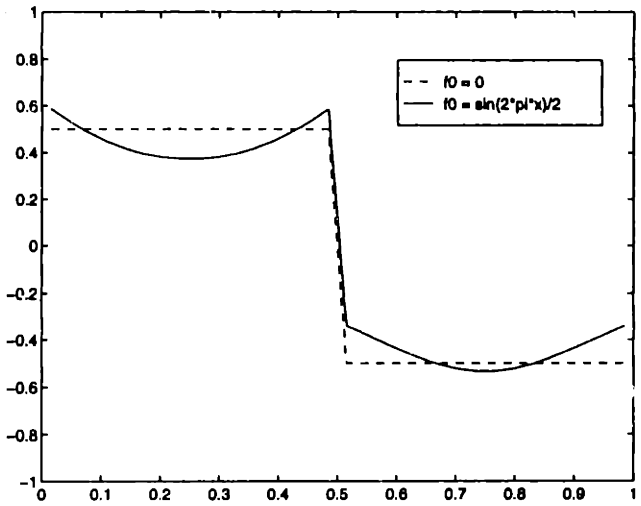


Figure 3.6. The Fréchét derivative at $x_i = 0.5$ for the 1D flow equation when linearized about the log-conductivity function $f_0 = 0.5\sin(2\pi x)$. The boundary conditions are $h(0) = 1$ and $h(1) = 0$. The Fréchét derivative for $f_0 = 0$ is provided by the dashed line.

locations and negative for downstream locations. Secondly, the Fréchét kernel integrates to zero over Ω, meaning that (under this scenario) head measurements alone cannot be used to determine the DC values of conductivity. In fact, the inner product of the Fréchét derivative in Figure 3.8 and f is a coarse-scale derivative of f. However, a major difference between the 1D and 2D functions is that, while the 2D Fréchét derivative is nonlocal, it is more localized than that for the 1D flow equation. Another notable difference is that the 2D Frechet derivative is relatively shift-invariant, even when evaluated near the boundaries of Ω.

To illustrate the effect of a nonzero source function on the Fréchét derivative, consider 2D flow on the unit square with a pumping well at $x_s = (0.5, 0.5)$ and $h = 0$ on the boundary of Ω. The well is idealized as the point source $Q(x) = -\delta(x - x_s)$. Again use $f_0 = 0$. Figure 3.9a illustrates the Fréchét derivative for a head measurement located just outside the wellbore. This function is entirely different from the functions displayed in Figure 3.8. The reason[9] is that, whenever an aquifer is pumped at a known rate, the conductivity value at the well can be inferred directly from the slope of h at the wellbore (using Darcy's Law). Measuring a single value of h does not provide the slope at the wellbore, but this slope can be estimated from knowledge that head is equal to zero on the boundary of Ω. Thus, the head samples near the wellbore measure conductivity near the wellbore; the region of support of the head measurement is not a point, however, as evidenced by the plot in Figure 3.9a, again due to the inability to exactly determine the gradient of head from a single head measurement. (There are

[9]The difference between the Fréchét derivative in Figure 3.9a and that in Figure 3.8a cannot be explained by the change in boundary conditions, since the Fréchét derivative would have similar form even if the boundary conditions were not changed.

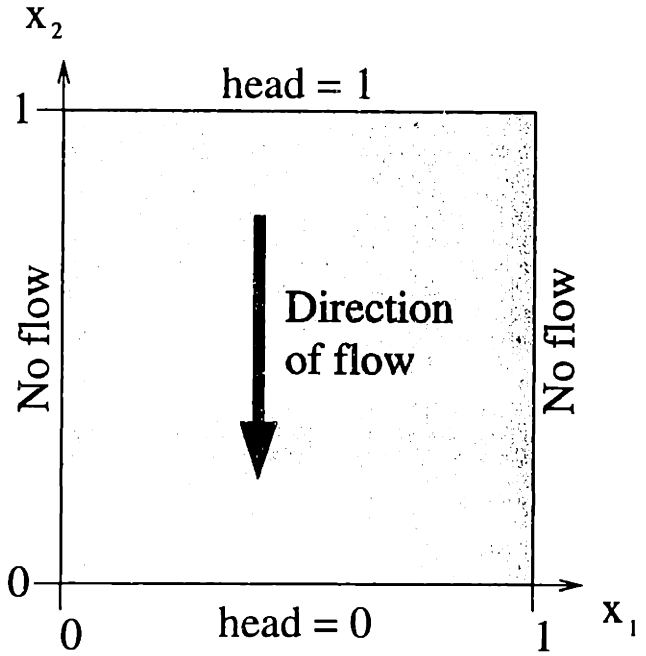
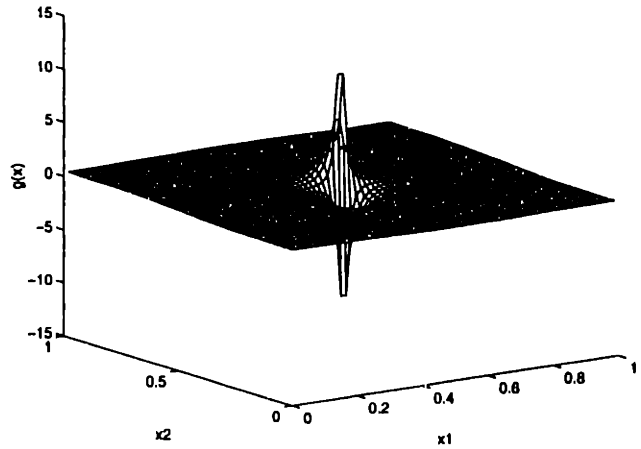


Figure 3.7. An illustration of the boundary conditions used to calculate the Fréchét derivatives in Figure 3.8.

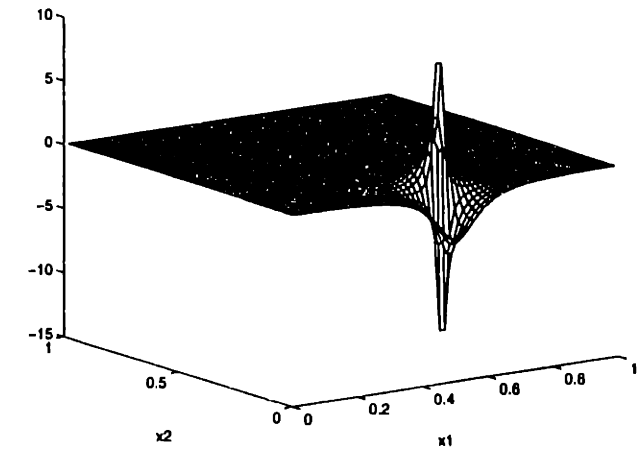
also numerical errors due to the finite-difference implementation of the flow equation.)

While the Fréchét derivative does not vary significantly with respect to its distance from the boundary of Ω, it does vary significantly with respect to its distance from pumping and injection wells. This variation is illustrated in Figure 3.9b, which displays the Fréchét derivative for a head measurement located at $x_i = (0.25, 0.25)$, which is well outside the wellbore. The Fréchét derivative is dramatically different from that in Figure 3.9a. The reason is that the gradient of head at significant distances from the wellbore can no longer be used to infer local values of conductivity. In fact, the Fréchét derivative in Figure 3.9b is similar in shape to a superposition of two (rotated) Fréchét derivatives from Figure 3.8. As the measurement location moves farther from the well, the effect of the forcing reduces and the Fréchét derivative becomes similar to that when there is no source term.

To summarize, the information supplied by head measurements about the conductivity function depends heavily upon the flow scenario, e.g., whether or not the aquifer is forced ($Q \neq 0$), what boundary conditions are assumed. Furthermore, the information supplied by the Fréchét derivative also depends upon the location of the measurement, the accuracy of the linearization, and the point of linearization f_0. All of these factors should be kept in mind when interpreting the conductivity function estimates computed in Chapters 5 and 6.

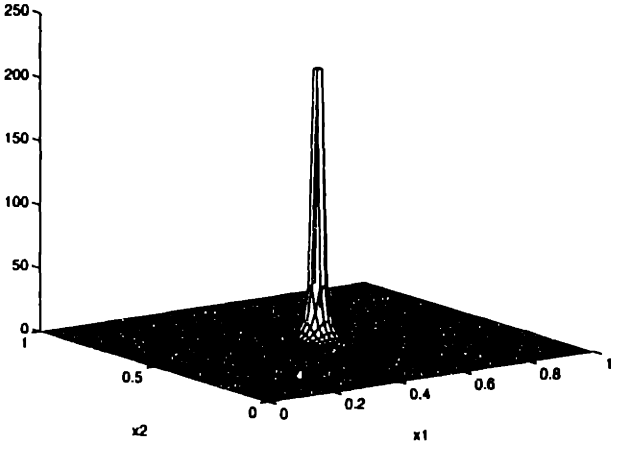


(a)

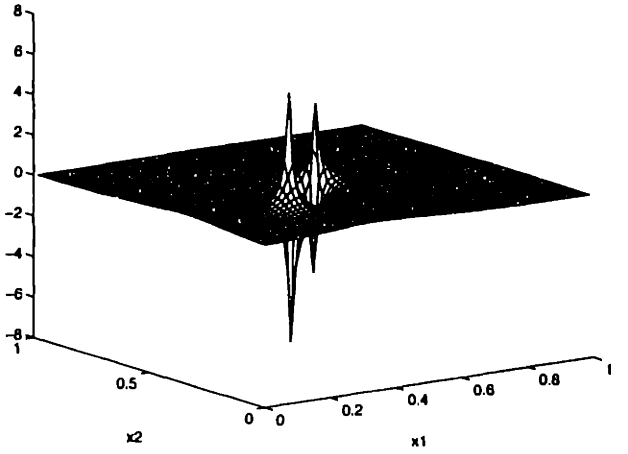


(b)

Figure 3.8. The Fréchét derivative for the 2D flow equation at $x_i = (0.5, 0.5)$ and (0.5,0.08) when linearized about the log-conductivity function $f_0 = 0$. The boundary conditions are illustrated in Figure 3.7.



(a)



(b)

Figure 3.9. The Fréchét derivative for the 2D flow equation when forced with a pumping well in the middle of Ω. The measurement is placed (a) near the pumping well at $(0.5, 0.5)$ and (b) away from the pumping well at $(0.25, 0.25)$.

■ 3.5 Estimating Hydraulic Conductivity from Measurements of Conductivity and Head

When introducing the Bayesian estimation framework in Chapter 2, the unknown to be estimated was assumed to be a finite-dimensional vector. The conductivity function, however, lies in an infinite-dimensional vector space. Bayesian estimators of $f(x)$, based upon Gaussian prior distributions, can be developed [69]. These estimators are conceptually important, since they are independent of any finite parameterization[10] of $f(x)$. However, in order to (numerically) compute an estimate, the conductivity function can only be represented by a finite number of elements. We show that, when $f(x)$ is restricted to a finite-dimensional vector space, the estimators described in Chapter 2 can be directly applied to conductivity estimation.

Assume that the conductivity function (or more importantly the estimate) is restricted to be a member of the function space $\text{span}\{\phi_j(x)\}_{1\le j\le N}$. This membership implies that

$$f(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x), \tag{3.21}$$

$$= \Phi(x)\,\underline{\alpha}, \tag{3.22}$$

for some vector $\underline{\alpha}^T = [\alpha_1, \ldots, \alpha_N]$, where $\Phi(x) = [\phi_1(x), \ldots, \phi_N(x)]$. The unknown function f is thus replaced by an unknown vector $\underline{\alpha}$. To use Bayesian estimators, a prior model must be specified for $\underline{\alpha}$.

To estimate $\underline{\alpha}$, the head and conductivity measurements must be expressed as functions of $\underline{\alpha}$. Consider first how to represent the conductivity measurements of Eq. (3.9). Recall that these measurement equations are somewhat of an idealization, since point values of conductivity can never be measured; instead, conductivity is measured over some local area. If conductivity is relatively constant over this area and we choose $\phi_i(x)$ to have the same support as the i-th conductivity measurement, then we obtain

$$y_i^f = \alpha_i + v_i^f, \qquad i = 1, \ldots, M_f \tag{3.23}$$

for each of the M_f conductivity measurements. Note that the measurement noise v_i^f may also model scale errors due to the mismatch between ϕ_i and true support of the i-th conductivity measurement [69].

The head measurements can be expressed in terms of $\underline{\alpha}$ by substituting $f = \Phi\underline{\alpha}$ into Eq. (3.12). The linearization follows directly from Eq. (3.15). First note that the point of linearization is no longer a function, but a vector $\underline{\alpha}_0$ for which $f_0 = \Phi\underline{\alpha}_0$. Plugging $f = \Phi\underline{\alpha}$ and $f = \Phi\underline{\alpha}$ into Eq. (3.15) and ignoring the higher-order terms yields

$$\mathcal{Y}_i(\Phi\underline{\alpha}_0) = g_i(\underline{\alpha}_0)^T\underline{\alpha} + v_i, \qquad i = 1, \ldots, M_h \tag{3.24}$$

[10]By *finite parameterization* of a function we are simply referring to the finite number of variables chosen to represent the function.

where

$$g_i(\underline{\alpha}_0)^T = \left[\langle \frac{\partial \mathcal{F}_h(\Phi\underline{\alpha}_0, x_i)}{\partial f}, \phi_1 \rangle, \ldots, \langle \frac{\partial \mathcal{F}_h(\Phi\underline{\alpha}_0, x_i)}{\partial f}, \phi_N \rangle \right]. \tag{3.25}$$

Equations (3.23) and (3.24) are in exactly the same form as required by the approximate implementation of the LLSE estimator for nonlinear measurements at described in Section 2.1.2. This linear estimator is also equivalent to a single iteration of the Gauss-Newton optimization of the MAP density $p(\underline{\alpha}\,|\,y)$. Both of these estimators are implemented in Chapter 5. Once the estimate $\widehat{\underline{\alpha}}$ has been computed, the conductivity estimate follows as $\hat{f} = \Phi\widehat{\underline{\alpha}}$. The error covariance is similarly derived.

■ 3.6 Choice of the Conductivity Parameterization

Given that the functions $\phi_j(x)$ must be chosen before the conductivity estimate can be computed, it is important to consider how these functions should be chosen. Usually these basis functions are chosen at a fixed scale and each function is a translation of a single generating function [98]. For instance, the parameterization used by a zeroth-order finite-element method (FEM) approximation in 1D is

$$\phi_j(x) = \phi\left(\frac{x - j\Delta x}{\Delta x}\right), \quad \text{where } \phi(x) = \begin{cases} 1, & |x| \leq 1/2, \\ 0, & \text{otherwise.} \end{cases}$$

Each function is a pulse of width Δx. The scale of the basis functions is chosen such that the finest-scale measurements, direct measurements of conductivity in this case, can be accurately represented. If the scale of the finest-scale measurement is small relative to the size of the domain, then the number of variables (N) can be very large, especially for multidimensional flow. In terms of computing the estimate, the large number of variables means that the $M_h + 1$ forward simulations required to compute the Fréchét derivatives can be prohibitively large. The same also becomes true of the estimator and estimation error equations.

With a choice of basis functions that have uniform and fine resolution, the estimate is generally "over-parameterized" in regions where there are no fine-scale measurements. In the context of conductivity estimation, these regions correspond to the areas in between wellbores. One solution, known in some contexts as upscaling, is to choose a coarser resolution than that of the direct measurements and to somehow account for the scale errors inherent in representing the finer-scale measurements. A more natural approach would be to choose a parameterization with space-varying resolution, where the resolution is allocated according to that which can be resolved by the measurements. Such an allocation of resolution is a difficult problem, especially when the measurements are nonlinear and there is no clear method for choosing the parameterization *a priori*. We will return to this problem in Chapters 5 and 6 when using multiscale models to estimate properties of groundwater aquifers.

Chapter 4

Extensions of Multiscale Realization Theory

This and the following chapters discuss extensions to the multiscale realization theory summarized in Section 2.3. The utility of the multiscale framework hinges upon the ability to realize low-order multiscale stochastic processes in the form of Eq. (2.22) that capture the correct joint second-order statistics between the process to be estimated and the measurements. In principle, Canonical Correlations theory can be used to realize multiscale models that have desired second-order statistics at the finest scale. While it provides a solid foundation for multiscale realization, the algorithm presented in [49, 51] has a number of drawbacks. (See Section 2.3.4 for a more thorough discussion.) Of these, the following are addressed in this chapter.

- Arbitrary nonlocal measurements cannot be incorporated.
- The estimation error variances are only computed for the variables $z(s)$. The error variances for other nonlocal functions of the finest-scale process will require significant additional computations.
- Canonical Correlations realizations are generally quite inefficient, requiring a number of SVD decompositions of the desired covariance for the finest-scale process.
- The approximate realizations based on Canonical Correlations have undesirable properties, e.g., the approximate realization algorithm requires roughly the same number of computations as does the exact realization algorithm and the resulting models are inconsistent.

We focus upon the first two problems, i.e., incorporating and estimating arbitrary nonlocal properties within the multiscale framework, and touch upon the latter two within this context.

Most of the multiscale applications thus far have focused on measuring and estimating the finest-scale process [33, 34, 35, 51, 60, 62, 61, 70], which makes perfect sense for applications like image processing that are characterized by dense measurements at the finest-scale of interest. For these applications, the coarser-scale tree variables only need to satisfy the Markov property of multiscale trees, which in turn leads to the

efficiency of the multiscale estimator described in Appendix B. Estimates of coarse-scale variables have been used in [60, 81], but without a careful explanation of what is represented by the coarse scales. To measure or estimate specific nonlocal functions of the random phenomenon of interest will require placing additional constraints upon the coarser scale variables of the tree process. In other words, the coarser-scale variables of the realized multiscale model will have to contain specific nonlocal functions in addition to satisfying the Markov property.

This chapter begins by describing the multiscale realization problem in the context of measuring or estimating nonlocal parameters. We then extend the realization algorithm of [49] to allow the modeling of particular nonlocal functions at coarse scales of tree processes. Because this algorithm is inefficient, it is of little practical interest. However, it suggests an efficient algorithm for building nonlocal functions into existing multiscale trees. For instance, assuming that a multiscale model already exists that has the desired second-order statistics at the finest scale, the variables of this tree can be augmented so that coarser-scale variables contain desired nonlocal functions of the finest-scale process. We then discuss the realization of multiscale models with approximate statistics, which are necessary for implementing large-dimensional estimation problems. Extensions of the multiscale framework for $1/f$ processes are the subject of Chapter 7.

■ 4.1 Measuring and Estimating Nonlocal Properties

As shown in Eq. (2.27), any measurement to be incorporated by the multiscale estimator must be in the form

$$y(s) = C_s z(s) + v(s)\,. \qquad v(s) \sim (0, R_s) \tag{4.1}$$

The implications of this measurement model are most clearly illustrated for internal models. Define f to be a vector containing the finest-scale variables of an internal multiscale model. The measurement equations in this case are restricted to the form

$$y(s) = C_s V_s f + v(s)\,, \qquad v(s) \sim (0, R_s) \tag{4.2}$$

meaning that only measurements

$$y(s) = H_s f + v(s) \tag{4.3}$$

for which H_s is in the row space of an individual internal matrix V_s can be incorporated by the multiscale estimator. If the internal matrices at coarser scales are chosen only to satisfy the multiscale Markov property, then the multiscale estimator will be unable to incorporate a large class of nonlocal measurements. For instance, given one of the multiscale models for 1D Markov processes described in Chapter 2, the multiscale estimator cannot incorporate a measurement of the average value of the finest-scale process, since no single state contains the average value of f.

Another reason for reassessing the role of the coarse-scale variables in multiscale models is the need to estimate particular coarse-resolution functions of the phenomenon

of interest. As a motivating example, consider the estimation of hydraulic conductivity, where f is a vector representing the hydraulic conductivity at the finest resolution of interest. This resolution might correspond to the resolution of the finest-scale measurement. However, because these fine resolution samples are sparsely distributed over the region of interest, a more coarse-resolution estimate is justified, at least in regions far from any fine-scale measurements. In this case the multiscale model, along with the estimation error variances, could in theory be used to select the optimal distribution of resolution in the estimate. (In selecting an optimal resolution, we are obviously making a trade-off between resolution and estimation error variance [71].) The only problem is that the coarser-scale variables of the multiscale process must contain specific coarse-resolution representations of the conductivity function.

As another example of coarse-resolution estimation, consider the estimation of the travel time in Eq. (3.7). In this case, estimating the conductivity function is only an intermediate step in obtaining an estimate of travel time. If travel time t_{cp} is a linear function of conductivity[1], then we can write

$$t_{\text{cp}} = g^T f \,,$$

meaning that the travel time estimate can be derived from the finest-scale conductivity estimate $\hat{f}$ as $\hat{t}_{\text{cp}} = g^T \hat{f}$. However, we are also interested in the uncertainty of this estimate. The travel time estimation error is equal to

$$\text{var}[\,\hat{t}_{\text{cp}}\,] = g^T P_e \, g \,, \qquad P_e = E\big[(f - \hat{f})(f - \hat{f})^T\big] \,,$$

which requires complete knowledge of the covariance of the hydraulic conductivity estimation error. The multiscale estimator only provides the error variance for the estimates of the each of the finest-scale variables, which corresponds to block-diagonal elements of P_e. The remaining entries can be derived from the multiscale error model [61], but such a computation is prohibitively expensive for large-dimensional f.

4.2 General Method for Realization of Internal Models

The multiscale realization algorithm based upon Canonical Correlations summarized in Section 2.3.4 is part of the more general framework for multiscale realization described in [49]. In this section, we will describe this more general framework and show how it can be readily extended to allow for the modeling of particular nonlocal functions by the coarse-scale variables.

Assume that we are only interested in internal multiscale models. The first step of the realization algorithms described in [49] is to specify a tree structure and to map the vector $f \sim (0, P_f)$ to the finest-scale of the tree. Since the statistics of only the finest-scale process are specified, each tree variable can be restricted to a function of its

[1] As shown in Chapter 3, travel time is a nonlinear and nonlocal function of hydraulic conductivity, but we will show in Chapter 6 how this relationship can be linearized.

finest-scale descendents, i.e.,

$$z(s) = W_s f_s \,. \tag{4.4}$$

The intuitive reason is that the process indexed on the subtree descending from node s serves only to realize the finest-scale process f_s. In this case, the q_s sets of variables descending from node s, $\{z(t) \,|\, t \in \mathcal{S}_{s\alpha_i}\}$ for $i = 1, \ldots, q_s$, are each linear functions only of $f_{s\alpha_i}$. Namely, for all nodes $t \in \mathcal{S}_{s\alpha_i}$, $z(t)$ is a linear function of $f_{s\alpha_i}$. For the other set of variables partitioned by node s, $\{z(t) \,|\, t \in \mathcal{S}^c_{s\alpha_i}\}$, each element must be a linear function only of f_{s^c} and $z(s)$; otherwise, these variables would be correlated with the process noise descending from node s. This implies that the Markov property to be satisfied by multiscale tree models for which only the finest-scale covariance is fixed reduces to the following: the $q_s + 1$ vectors

$$\widetilde{f}_{s\alpha_i} \triangleq f_{s\alpha_i} - \widehat{E}\big[f_{s\alpha_i} \,|\, W_s f_s\big] \,, \qquad i = 1, \ldots, q_s + 1 \,, \tag{4.5}$$

are mutually uncorrelated.

The realization algorithm for such internal multiscale models reduces to the following sequential steps:

(a) map the process $f \sim (0, P_f)$ to the finest-scale variables of a tree;

(b) at each node, find a matrix W_s that decorrelates the $q_s + 1$ vectors in Eq. (4.5);

(c) compute the multiscale model parameters from Eqs. (2.30) and (2.32).

A proof that this algorithm leads to multiscale models with the desired finest-scale covariance is provided in Appendix D. Canonical correlations can be used to compute the internal matrices W_s, but Canonical Correlations is not specific to this approach.

Assume we now desire that the coarse-scale variables of the multiscale models contain specific nonlocal functions of f. Namely, assume that each linear function $g_i^T f$, $i = 1, \ldots, L$, is to be placed at the node τ_i. If no restrictions are placed upon the vectors g_i and the nodes τ_i, then we can no longer assume that the tree variables are only functions of their finest-scale descendents. Instead, the internal variables will take the more general form $z(s) = V_s f$. In addition to satisfying the Markov property, these internal variables must be chosen such that

$$c_i^T V_{\tau_i} = g_i^T \tag{4.6}$$

for some vector c_i, $1 \le i \le L$.

When the state variables are constrained as in Eq. (4.6), the variables at the nodes τ_i can no longer be assumed to be linear functions of f_s. For instance, if $\tau_1 \neq 0$, and $g_1^T f$ is the average value of the finest-scale process, then $z(\tau_i)$ must be a function of the entire finest-scale process. In this case, the multiscale Markov property is not equivalent to the decorrelation of the variables in Eq. (4.5). The correspondence between Eq. (4.5)

and the Markov property is only true when all the variables at nodes in $\mathcal{S}_s$ are linear functions of f_s. To determine an analogous sufficient condition for the Markov property when the constraints in Eq. (4.6) are imposed, first define

$$h_s = G_s f \tag{4.7}$$

to be vector of the linear functionals $g_i^T f$ for which $\tau_i \in \mathcal{S}_s$, i.e., all the linear functionals placed at nodes descending from or equal to s. Also define the vector

$$b_s = \begin{bmatrix} f_s \\ h_s \end{bmatrix} . \tag{4.8}$$

Note that the only requirements on the variables in the subtree $\mathcal{S}_s$ are that f_s be generated at the finest scale and that h_s be generated at designated nodes in $\mathcal{S}_s$. Thus $z(s)$ only needs to be a function of b_s, i.e.,

$$z(s) = \mathcal{W}_s b_s . \tag{4.9}$$

Equation (4.9) is a direct analogy of Eq. (4.4). Assuming the internal variables satisfy Eq. (4.9), all of the variables at nodes in $\mathcal{S}_s$ will be linear functions only of b_s. The Markov property of the multiscale tree in this case reduces to the following: the $q_s + 1$ vectors

$$\tilde{b}_{s\alpha_i} = b_{s\alpha_i} - \hat{E}\left[b_{s\alpha_i} \mid \mathcal{W}_s b_s\right] , \qquad 1 \le i \le q_s + 1 \tag{4.10}$$

are mutually uncorrelated, where h_{q_s+1} is a vector of all the linear functionals $g_i^T f$ for which $\tau_i \in \{\mathcal{S}_s^c \cup s\}$. Equation (4.10) is analogous to Eq. (4.5).

One problem with the internal variables $\mathcal{W}_s b_s$ that conditionally decorrelate the vectors in Eq. (4.10) is that they are not guaranteed to contain the linear functionals placed at node s, i.e., $g_i^T f$ for $\tau_i = s$. However, the reason that the vector b_{q_s+1} contains these linear functionals is that they can be added to the vectors $\mathcal{W}_s b_s$ without introducing any correlation among the vectors in Eq. (4.10).

An algorithm for realizing a multiscale tree with covariance P_f at the finest scale and with states $z(\tau_i)$ for $1 \le i \le L$ satisfying $E\left[\left(c_i^T z(\tau_i)\right) f\right] = g_i^T P_f$ for some vector c_i^T—see Eq. (4.6)—follows as:

(a) map the process $f \sim (0, P_f)$ to the finest-scale variables of a tree; map the linear functionals $g_i^T f$ to the nodes τ_i;

(b) at each node s, find a matrix $\mathcal{W}_s$ that decorrelates the $q_s + 1$ vectors in Eq. (4.10); from this matrix, the internal variable $z(s)$ is given by

$$z(s) = \begin{bmatrix} \mathcal{W}_s b_s \\ h(s) \end{bmatrix} , \tag{4.11a}$$

$$= V_s f , \tag{4.11b}$$

where $h(s)$ is a vector of all $g_i^T f$ for which $\tau_i = s$; this internal variable satisfies the Markov property and there exists a vector c_i^T for each $\tau_i = s$ such that $c_i^T z(s) = g_i^T f$;

(c) compute the multiscale model parameters from Eqs. (2.30) and (2.32), where V_s is derived from $\mathcal{W}_s$, b_s, and $h(s)$.

These three steps are directly analogous to the three steps given for the finest-scale realization algorithm. Again, Canonical Correlations can be used to compute matrices $\mathcal{W}_s$ with near minimal row dimensions, but Canonical Correlations is not central to the approach.

While this realization algorithm is quite general in its approach and the resulting multiscale models can be used to incorporate arbitrary nonlocal measurements, there are a number of drawbacks. Those drawbacks associated with the Canonical Correlations implementation—in particular, computational inefficiency for both exact and approximate realizations—also apply to the more general realization algorithm. The more general algorithm also suffers from not being able to take advantage of the possible stationarity of P_f. (As shown in [49], the FFT can be used to significantly reduce the number of computations required for the Canonical Correlations implementation when P_f is stationary.) The more general drawback associated with Canonical Correlations based realizations is that they are completely general and in no way account for the particular features of the prior covariance P_f and the measurement kernels g_i. As shown in Section 2.3.2 for 1D Markov processes and 2D Markov Random Fields and in Chapter 7 for fractional Brownian motion, the internal variables $W_s f_s$ can sometimes be determined without significant computations. In this case, the only modeling to be done is to "adjust" the internal variables $W_s f_s$ so that they also account for the nonlocal functions $g_i^T f$. This procedure is the subject of Section 4.3.

Another aspect of the multiscale realization problem not addressed in this section is the choice of nodes τ_i. Remember that the overall goal of any multiscale realization is to achieve some optimal trade-off between the state dimensions and the fidelity (in terms of second-order statistics) of the multiscale model. As shown in the following section, this trade-off is affected by the choice of nodes τ_i, e.g., for exact multiscale models the computational cost function in Eq. (2.37) will change depending upon the choice of nodes τ_i. The problem is to determine the set of nodes τ_i which minimizes the computational costs (of the associated multiscale estimator or likelihood calculator) for any desired level of statistical accuracy. Note that if the placement of the nonlocal linear functionals had no effect upon the state dimensions of the multiscale model, then all the nonlocal functions could simply be mapped to the finest scale of the multiscale tree, in which case the algorithm of [49] could be used. Namely, we would map the vector $q = [f^T, g_1^T f, g_2^T f, \ldots, g_L^T]^T$ to the finest scale of a multiscale tree; since the covariance of q is determined by P_f and the kernels g_i, the algorithm for realizing the statistics of the finest-scale process could be directly applied.

■ 4.3 State Augmentation

For the multiscale models of 1D and 2D Markov processes described in Section 2.3.2 (and for the multiscale model for fractional Brownian motion described in Chapter 7), the

internal variables which exactly (or approximately) decorrelate the vectors in Eq. (4.5) can be specified without computation. To expand the set of functions represented by these tree models, so that specific nonlocal measurements can be incorporated by the multiscale estimator, one can imagine augmenting the internal variables with additional linear functions of f and then re-computing the model parameters. However, doing this requires considerable care. In particular, the states must be augmented such that both the Markovianity of the tree process is preserved and the resulting model is internal. For example, suppose we have an internal multiscale model with variables $z(s) = W_s f_s$ and we naively augment the state of our model at a single node τ in order to include the linear function Gf. That is, suppose $\zeta(s) = z(s)$ for $s \neq \tau$ and

$$\zeta(\tau) = \underbrace{\left[\begin{array}{c|c} W_\tau & 0 \\ \hline G & \end{array}\right]}_{\mathcal{V}_\tau} f \,. \tag{4.12}$$

In general, this augmentation will destroy the Markovianity of the tree. For example, the states $z(\tau\overline{\gamma})$, $z(\tau\alpha_1)$, ... , $z(\tau\alpha_q)$ generally are **correlated** with each other after conditioning on $\zeta(\tau)$. The consequences of this correlation are that, for the multiscale model defined by Eqs. (2.30) and (2.32) with $\mathcal{V}_\tau$ substituted for V_τ, the finest-scale process will not have covariance P_f; also, the model will not be internal, i.e., the state at node τ will not be equal to a linear function of the finest-scale process.

The issue here is that the augmentation at node τ introduces some coupling among variables due to the nonlocal nature of the linear function Gf. If the correct statistics are to be maintained, and the state at node τ is to contain the desired function of the finest-scale process, the effect of the coupling must also be propagated to other nodes on the tree.

■ 4.3.1 Maintaining the Markov Property of the Internal Variables

To illustrate how augmentation can destroy the Markovianity of internal variables, consider the multiscale models for 1D Markov processes described in Section 2.3.2. Recall that the finest-scale of these multiscale models represents a 1D Markov process $f[k]$ on some interval $[0, N]$. Conditioned on $z(0)$, which contains $f[0]$, $f[N]$, and $f[k_0]$ for some $0 < k_0 < N$, the values of $f[k]$ on $[0, k_0]$ are uncorrelated with the values on the interval $[k_0, N]$. However, if the average value of $f[k]$ on $k \in [0, N]$ is added to $z(0)$, i.e., $\zeta(0)$ contains $z(0)$ as well as the average value of the finest-scale process, then $f[k]$ on $[0, k_0]$ will not be uncorrelated with $f[k]$ on $[k_0, N]$ after conditioning on $\zeta(0)$. This means that the displacements $\widetilde{f}[k_1]$ and $\widetilde{f}[k_2]$, which are defined by Eq. (2.35) and are to be represented by the process noise at the first scale of the tree, will be correlated. This implies that some additional augmentation will be required to account for the correlation.

Instead of directly augmenting $z(\tau)$ in Eq. (4.12) with Gf, another linear function must be found for which (i) the augmented variable $\zeta(\tau)$ still decorrelates the $q_\tau + 1$ sets of internal variables partitioned by node τ, and (ii) $\zeta(\tau)$ contains Gf. If all the

internal variables have the form $z(s) = W_s f_s$, recall that $z(s)$ satisfies the multiscale Markov property if and only if it conditionally decorrelates the $q_s + 1$ random vectors $\{f_{s\alpha_i}\}_{1 \le i \le q_s+1}$. Therefore, to augment $z(s)$ with a linear function of f and not alter the covariance of the finest-scale process, we need only ensure that the $q_s + 1$ vectors $\{f_{s\alpha_i}\}_{1 \le i \le q_s+1}$ remain conditionally uncorrelated. To do so we make use of the following corollary of Proposition 5 in Chapter 3 of [49].

Corollary *If the $q_s + 1$ vectors $\{f_{s\alpha_i}\}_{1 \le i \le q_s+1}$ are uncorrelated after conditioning on some linear function $V_s f$, they remain uncorrelated after conditioning on $V_s f$ and individual linear functions of $f_{s\alpha_i}$, $i = 1, \ldots, q_s + 1$.*

For example, the $q_s + 1$ vectors $\{f_{s\alpha_i}\}_{1 \le i \le q_s+1}$ are uncorrelated after conditioning on $V_s f$, $L_1 f_{s\alpha_1}$, and $L_2 f_{s\alpha_2}$, but they will generally be correlated after conditioning upon $V_s f$ and $L[f_{s\alpha_1}^T,\ f_{s\alpha_2}^T]^T$. Therefore, to add the linear functional $\langle g, f \rangle \triangleq g^T f$ to $z(\tau)$, we first define the following matrix

$$G_\tau = \begin{bmatrix} g_{\tau\alpha_1}{}^T & 0 & \cdots & 0 & 0 \\ 0 & g_{\tau\alpha_2}{}^T & 0 & & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & & \ddots & g_{\tau\alpha_{q_\tau}}{}^T & 0 \\ 0 & 0 & \cdots & 0 & g_{\tau\alpha_{q_\tau+1}}{}^T \end{bmatrix}, \tag{4.13a}$$

where the vectors $g_{\tau\alpha_i}$ are defined by

$$\langle g, f \rangle = \langle g_{\tau\alpha_1}, f_{\tau\alpha_1} \rangle + \langle g_{\tau\alpha_2}, f_{\tau\alpha_2} \rangle + \cdots + \langle g_{\tau\alpha_{q_\tau}}, f_{\tau\alpha_{q_\tau}} \rangle + \langle g_{\tau\alpha_{q_\tau+1}}, f_{\tau^c} \rangle . \tag{4.13b}$$

The variable at node τ can now be augmented as

$$\zeta(\tau) = \underbrace{\left[\begin{array}{c|c} W_\tau & 0 \\ \hline G_\tau & \end{array}\right]}_{\mathcal{V}_\tau} \underbrace{\begin{bmatrix} f_\tau \\ f_{\tau^c} \end{bmatrix}}_{f}, \tag{4.13c}$$

without altering the Markov property. Note that if g has full support, i.e., if each term $\langle g_{\tau\alpha_i}, f_{\tau\alpha_i} \rangle \not\equiv 0$, then this augmentation requires an additional $q_\tau + 1$ elements in the state at node τ. If some of these terms are zero, then a lower-dimensional augmentation is possible. Furthermore, if any of the rows of G_τ are already in the row-space of $[W_\tau\ 0]$, these elements are already available in $z(\tau)$ and need not be added. Note also that since the partitioning of f into $f_{\tau\alpha_i}$ is different for each node, one might imagine that there is a best choice for node τ in terms of minimizing the number of terms $\langle g_{\tau\alpha_i}, f_{\tau\alpha_i} \rangle$ which are nonzero and hence minimizing the dimension of the augmentation.

Define the augmented variable at each node by $\zeta(s) = \mathcal{V}_s f$, where $\zeta(s) = z(s)$ if the state at node s is not augmented. The model parameters of the augmented model

follow as

$$P_0 = P_{\zeta(0)}, \tag{4.14a}$$
$$= \mathcal{V}_0 P_f \mathcal{V}_0^T,$$
$$A_s = P_{\zeta(s)\zeta(s\bar{\gamma})} P_{\zeta(s\bar{\gamma})}^{-1}, \tag{4.14b}$$
$$Q_s = P_{\zeta(s)} - P_{\zeta(s)\zeta(s\bar{\gamma})} P_{\zeta(s\bar{\gamma})}^{-1} P_{\zeta(s\bar{\gamma})\zeta(s)}. \tag{4.14c}$$

where $P_{\zeta(s)} = \mathcal{V}_s P_f \mathcal{V}_s^T$ and $P_{\zeta(s)\zeta(s\bar{\gamma})} = \mathcal{V}_s P_f \mathcal{V}_{s\bar{\gamma}}^T$. This augmented model will have a finest-scale process with covariance identical to that of the original model.

■ 4.3.2 Maintaining an Internal Multiscale Model

The augmentation described in the preceding section, which augments the state at a single node τ, does maintain Markovianity and hence yields a model whose finest-scale process will have the desired covariance P_f. However, this model will not generally be consistent; that is, the element of the state at node τ that is intended to equal $\langle g_{\tau\alpha_i}, f_{\tau\alpha_i} \rangle$ may not be equal to $\langle g_{\tau\alpha_i}, f_{\tau\alpha_i} \rangle$. The reason for this is simple: at node τ we are attempting to **pin** a linear combination of the values descending from node $\tau\alpha_i$. In order to ensure that this value is pinned, information must be propagated from node τ all the way to its descendents at the finest scale. To illustrate this problem and to motivate its solution, consider the following example of augmenting the root node of a multiscale model for a 1D Markov process with the sum of the finest-scale process.

Example: Multiscale Modeling the Sum of a 1D Markov Process

Consider a 1D first-order Markov process $f[k]$ on the interval $[0, 15]$. A multiscale model for this process was described in Section 2.3.2, and the samples of $f[k]$ contained in each variable of this model are illustrated in Figure 2.3b. Assume that the 1D Markov process is to be estimated, using the multiscale estimator, from point measurements of $f[k]$ together with a measurement of the sum $h = \left(\sum_{k=0}^{15} f[k]\right)$ of the finest-scale process. From Section 4.3.1, we know that $z(0)$ can be augmented with the two linear functions

$$h_1 = \sum_{k=0}^{7} f[k] \quad \text{and} \quad h_2 = \sum_{k=8}^{15} f[k] \tag{4.15}$$

without altering the Markov property of the tree. Thus, if the root node variable is augmented as $\zeta(0) = [z(0)^T, h_1, h_2]^T$ and no other variables are changed, then the finest-scale process of the multiscale model derived from Eq. (4.14) will have the covariance of the 1D Markov process.

However, the element in the state at the root node which is intended to contain $h = h_1 + h_2$ will not be equal to the sum of the finest-scale process unless this value is propagated from the root node to the finest-scale. This propagation is accomplished by

constraining the scale-to-scale recursion of the multiscale model. For this 1D Markov example, this means constraining the midpoint displacements by conditioning them on the value of h generated at the root node. This conditioning is accomplished by augmenting the descendents of the root node, except for the finest-scale states which are never augmented, with h. Again, this augmentation must also preserve Markovianity. For example, consider the two children of the root node, nodes $0\alpha_1$ and $0\alpha_2$. The augmentation of these nodes is

$$\zeta(0\alpha_1) = \begin{bmatrix} z(0\alpha_1) \\ \sum_{k=0}^{3} f[k] \\ \sum_{k=4}^{7} f[k] \\ h_2 \end{bmatrix} \quad \text{and} \quad \zeta(0\alpha_2) = \begin{bmatrix} z(0\alpha_2) \\ \sum_{k=8}^{11} f[k] \\ \sum_{k=12}^{15} f[k] \\ h_1 \end{bmatrix}. \tag{4.16}$$

However, these states contain more information than is needed. For instance, $f[k]$ on the interval $[0,7]$ is uncorrelated with $f[k]$ on the interval $[8,15]$ when conditioned on $z(0\alpha_1)$. Thus the last element of $\zeta(0\alpha_1)$ in Eq. (4.16) contains no additional information about the descendents of nodes $0\alpha_1$. That is, in order to maintain consistency, and hence an internal realization, the state at node $0\alpha_1$ must only be made consistent with h_1, the component of h corresponding to the finest-scale descendents of node $0\alpha_1$. Similarly, the state at node $0\alpha_2$ must only be made consistent with h_2. As a result, the states in Eq. (4.16) can be reduced to

$$\zeta(0\alpha_1) = \begin{bmatrix} z(0\alpha_1) \\ \sum_{k=0}^{3} f[k] \\ \sum_{k=4}^{7} f[k] \end{bmatrix} \quad \text{and} \quad \zeta(0\alpha_2) = \begin{bmatrix} z(0\alpha_2) \\ \sum_{k=8}^{11} f[k] \\ \sum_{k=12}^{15} f[k] \end{bmatrix}. \tag{4.17}$$

For this simple example, the augmentation is now complete, and the parameters of the augmented multiscale model can now be derived from Eq. (4.14). The resulting model generates a finest-scale process with the desired covariance P_f and is internally consistent so that, for example, the value of $h_1 + h_2$ at the root node does exactly equal the sum of the finest-scale process.

Example: Modeling the Sum at Scale One

Even though the sum of the finest-scale process is a function of the entire finest-scale process, it can be advantageous, as will be shown in the example of Section 4.3.5, to model this value at a node other than the root node. Consider augmenting the variable at $0\alpha_1$ with h. The augmented variable which preserves the Markov property of $z(0\alpha_1)$ is

$$\zeta(0\alpha_1) = \begin{bmatrix} z(0\alpha_1) \\ \sum_{k=0}^{3} f[k] \\ \sum_{k=4}^{7} f[k] \\ h_2 \end{bmatrix}. \tag{4.18}$$

While, as argued in the previous example, the last element of $\zeta(0\alpha_1)$ is not needed for maintaining Markovianity or consistency with the nodes descending from node $0\alpha_1$, it is necessary if h is to be captured at this node.

To maintain consistency, the information contained in h_2 at node $0\alpha_1$ must be propagated to the other half of the tree, i.e., that descending from node $0\alpha_2$. To accomplish this, it is necessary that the value of h_2 be available to this part of the process; therefore, the root node must be augmented as

$$\zeta(0) = \begin{bmatrix} z(0) \\ h_2 \end{bmatrix}. \tag{4.19}$$

The state $z(0\alpha_2)$ can be augmented exactly as in Eq. (4.17). Note that because we have chosen to place the measurement at node $0\alpha_1$, $\zeta(0)$ does not need to include the sum over the "left" half of the tree, (as we do not introduce the constraint on this sum at the root node). Comparing with the augmented model in the preceding subsection, the dimension of $\zeta(0)$ in this model has been reduced while that of $\zeta(0\alpha_1)$ has been increased. The remaining states are identical in the two models. As before, having defined the states, the model parameters can be generated from Eq. (4.14).

4.3.3 An Algorithm for Augmenting Internal Multiscale Realizations

Using the previous two examples for intuition, we now present a general algorithm for adding linear functions of f to the coarser-scale variables of internal multiscale models. This algorithm applies to a much broader class of processes than those discussed in the previous section. The multiscale model can have an arbitrary number of children per node and the finest-scale process can have any desired covariance—not just that of 1D Markov process. The algorithm proceeds in two stages: (a) first, the augmented variables $\zeta(\cdot)$ are created for each node on the tree, and then (b) the model parameters are computed from Eq. (4.14) for the augmented process $z^a(\cdot)$.

The algorithm which follows is for adding a single linear functional $\langle g, f \rangle$ to the variable at node τ. This procedure can then be applied recursively to add additional linear functions. The initial step is to determine $\zeta(\tau)$. As discussed in Section 4.3.1, the augmented variable which preserves the Markov property of $z(\tau)$ is given by Eq. (4.13). The next step is to define $\zeta(\cdot)$ for the remaining nodes in the tree to guarantee that the information generated by $z^a(\tau)$ is passed consistently to the finest-scale process. First consider the nodes descendent from node τ. Since all the descendents of node τ are linear functions of f_τ, the entire process descendent from node τ is uncorrelated with f_{τ^c} when conditioned on $z(\tau)$. Thus, augmenting any variable descendent from node τ with a linear function of f_{τ^c} will have no effect upon the parameters derived from Eq. (4.14). Consequently, since the linear function $\langle g, f \rangle$ can be decomposed as

$$\langle g, f \rangle = \langle g_\tau, f_\tau \rangle + \langle g_{\tau^c}, f_{\tau^c} \rangle, \tag{4.20}$$

the variables descendent from node τ only need to be made consistent with $\langle g_\tau, f_\tau \rangle$. In fact, because of the conditioning property of $z(s)$, (where s is a descendent of τ),

the augmented variable only needs to include $\langle g_s, f_s \rangle$. This augmentation will guarantee that all the process noise added to the descendents of node τ is conditioned on $\langle g, f \rangle$. Therefore, the augmentation of $z(s)$ which preserves Markovianity and maintains consistency is

$$\zeta(s) = \begin{bmatrix} W_s \\ G_s \end{bmatrix} f_s = \mathcal{V}_s f \,, \tag{4.21a}$$

$$G_s = \begin{bmatrix} {g_{s\alpha_1}}^T & 0 & \cdots & 0 \\ 0 & {g_{s\alpha_2}}^T & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & {g_{s\alpha_{q_s}}}^T \end{bmatrix} . \tag{4.21b}$$

Now consider determining $\zeta(\cdot)$ for nodes not in $\mathcal{S}_\tau$. These nodes must be augmented to make the value of $\langle g_{\tau^c}, f_{\tau^c} \rangle$ consistent with the finest-scale process f_{τ^c}. However, if the support of g is not the entire domain, we may only need to augment a subset of the nodes in $\mathcal{S}_\tau^c$. Specifically, define the direct ancestors of τ as $\tau\overline{\gamma}$, $\tau\overline{\gamma}^2$, ..., and let σ be the ancestor closest to node τ for which

$$g^T f = {g_\sigma}^T f_\sigma \,. \tag{4.22}$$

Only nodes descendent from node σ need to be augmented, since, conditioned on $z(\sigma)$, the variables at any node outside the subtree descending from σ are uncorrelated with f_σ and hence with $g^T f$. Consider first the augmentation of a node $s \neq \tau$ on the path connecting τ and σ, (i.e., s is a direct ancestor of τ that is either node σ or a descendent of σ). As always, $\langle g, f \rangle$ can be expressed as

$$\langle g, f \rangle = \langle g_{s\alpha_1}, f_{s\alpha_1} \rangle + \langle g_{s\alpha_2}, f_{s\alpha_2} \rangle + \ldots + \langle g_{s\alpha_{q_s}}, f_{s\alpha_{q_s}} \rangle + \langle g_{s^c}, f_{s^c} \rangle \,. \tag{4.23}$$

While $\langle g_{s^c}, f_{s^c} \rangle$ is not needed at node s to maintain Markovianity, it must be included in $\zeta(s)$ to ensure that this value is passed to the state at node τ. This is a generalization of the description of $\zeta(0\alpha_1)$ in Eq. (4.18), for which the last component of the state was not required for Markovianity but was needed to have the entire linear functional available at node τ. In the more general case here, the last component is needed to have the entire linear functional available at a *descendent* of node s, (namely, node τ).

Turning to the other q_s components in Eq. (4.23), all but one must be included in $\zeta(s)$. This component corresponds to the child $s\alpha_i$ of node s for which $\tau \in \mathcal{S}_{s\alpha_i}$, i.e., the child of node s that is either node τ itself or a direct ancestor of node τ. This component can be excluded without disturbing Markovianity or consistency and can be generated at a descendent of node s. This is a generalization of the augmentation of node 0 given in Section 4.3.2, where $z(0)$ only needs to be augmented with $h_2 = \sum_{k=8}^{15} f[k]$ and not with $h_1 = \sum_{k=0}^{7} f[k]$.

The augmented variable $\zeta(s)$ is then given by

$$\zeta(s) = \underbrace{\left[\begin{array}{c|c} W_s & 0 \\ \hline G_s & \end{array}\right]}_{\mathcal{V}_s} \underbrace{\left[\begin{array}{c} f_s \\ f_{s^c} \end{array}\right]}_{f}, \tag{4.24a}$$

where the elements of $G_s f$ correspond to all of the elements on the right-hand side of Eq. (4.23) except the one not needed for the augmentation. For example, if $i = 1$ is the term not needed for the augmentation and if the elements of f_s are organized as $f_s^T = \left[f_{s\alpha_1}^T, f_{s\alpha_2}^T, \cdots, f_{s\alpha_{q_s}}^T\right]^T$, then

$$G_s = \begin{bmatrix} 0 & g_{s\alpha_2}^T & 0 & \cdots & & 0 \\ 0 & 0 & g_{s\alpha_3}^T & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & 0 & g_{s\alpha_{q_s}}^T & 0 \\ 0 & 0 & \cdots & & 0 & g_{s^c}^T \end{bmatrix}. \tag{4.24b}$$

Finally, once we have augmented each of the direct ancestors of τ up to and including σ, the descendents of these nodes must also be augmented. This is exactly the same procedure used to augment the descendents of node τ, i.e., each state is augmented with those elements necessary to maintain both Markovianity and internal consistency. The resulting overall algorithm for state augmentation can then be summarized as follows: for each node $s \in \mathcal{S}_\sigma$—recall that σ is the node closest to τ on the path from τ to the root node such that Eq. (4.22) is satisfied—and s not at the finest scale

(a) If $s = \tau$, then $\zeta(\tau)$ is given by Eq. (4.13).

(b) If $s \neq \tau$ and s is on the path from σ to τ, then $\zeta(s)$ is given by Eq. (4.24).

(c) Otherwise, $\zeta(s)$ is given by Eq. (4.21).

For $s \notin \mathcal{S}_\sigma$ or s at the finest scale, then $\zeta(s) = z(s)$. Note that if $\tau = \sigma$, i.e., if the linear function placed at node τ is a function only of f_τ, then the augmentation is simplified. Namely, $\zeta(s)$ is given by Eq. (4.21) for $s \in \mathcal{S}_\tau$, and $\zeta(s) = z(s)$ for $s \notin \mathcal{S}_\tau$.

Once the matrices $\mathcal{V}_s$ have been determined, the final step of the augmentation algorithm is to compute the model parameters from Eq. (4.14). Given the parameters of the original multiscale model, only the parameters for $s \in \{\mathcal{S}_\sigma, \sigma\overline{\gamma}\}$ need to be recomputed for the augmented model.

For adding linear functions of f, i.e., multiple linear functionals, to the state at node τ or any number of nodes of a multiscale tree, the state augmentation just described can be applied recursively to individual linear functionals. This recursive procedure can be used to represent nonlocal measurements of f or coarse resolution functions to be estimated within the multiscale framework. Note, however, that Eq. (4.14) need

only be executed once, after all the linear functionals have been incorporated into the augmented states $\zeta(\cdot)$.

Numerous examples of the augmentation algorithm are provided in Chapters 5 and 6 in the context of hydraulic conductivity and travel time estimation. (See Chapter 3 for a discussion of these parameters.)

■ 4.3.4 Implementation Issues for State Augmentation

The feasibility of the state augmentation algorithm of course depends on the number of computations and storage elements required to implement Eq. (4.14). Given all the necessary covariance matrices $P_{\zeta(s)}$ and $P_{\zeta(s)\zeta(s\overline{\gamma})}$, computing the autoregression parameters requires roughly $\mathcal{O}(Nd^3)$ computations (assuming the state variables have the same dimension). More to the point, the number of computations required to implement Equation (4.14) when the augmented state covariances are known is of the same order as the number of computations required to implement the multiscale estimation of the augmented tree model. Because the number of computations required for the multiscale estimator is a function only of the augmented variable dimensions, we turn our attention to the formation of the covariance matrices $P_{\zeta(s)}$ and $P_{\zeta(s)\zeta(s\overline{\gamma})}$. The dimensions of the augmented variables are the subject of the next subsection.

When the vector f has very large dimension, the primary obstacle to implementing Eq. (4.14) is the formation of the covariance matrices $P_{\zeta(s)}$ and $P_{\zeta(s)\zeta(s\overline{\gamma})}$. The reason is that P_f cannot be explicitly stored in memory when the dimension of f is large. For instance, if the finest-scale process corresponds to a 2D field of 64-by-64 elements, P_f has approximately 17 million elements, making it infeasible to derive $P_{\zeta(s)}$ directly from stored matrices P_f and $\mathcal{V}_s$. For these large problems, the augmented covariances must computed implicitly, i.e., without explicitly storing P_f. The ability to compute these matrices implicitly, as well as the method used, will depend upon the particular application. As a demonstration, consider the examples in Chapters 5 and 6, where the finest scale process is a stationary Markov Random Field and the variables at coarser scales are augmented with nonlocal functions $g_i^T f$. From Section 2.3.2, we know that the original variables (before augmentation) for multiscale models of MRFs correspond to samples of the MRF on boundaries of the 2D domain of interest. Partition $\zeta(s)$ as $\zeta(s) = [z(s)^T,\ \theta(s)^T]^T$, where $z(s)$ contains the samples of the MRF and $\theta(s) = G_s f$ contains the linear functions added to the variable at node s. The covariance $P_{\theta(s)}$ can be determined by sampling the covariance function of the Markov random field. The other elements of $P_{\zeta(s)}$ can be determined using the stationarity of f. Namely, if $g^T f$ corresponds to an element of $\theta(s)$, then $g^T P_f$ can be computed using the FFT and zero padding determined by the correlation length of the MRF. The computation of $G_s P_f$ thus requires $\mathcal{O}(d_\theta N \log N)$ computations when the number of finest-scale elements (the dimension of f) is equal to N and the dimension of θ is equal to d_θ. The computation of $P_{\theta(s)}$ from $G_s P_f$ requires and additional $d_\theta N$ computations, while $P_{\theta(s)z(s)}$ is given by sampling the columns of $G_s P_f$ which correspond to the boundary samples in $z(s)$. Thus we have computed $P_{\zeta(s)}$ in essentially $\mathcal{O}(d(s)N \log N)$ computations and $\mathcal{O}(d(s)N)$

storage elements. The cross-covariances $P_{\zeta(s)\zeta(s\overline{\gamma})}$ can be formed similarly. The lesson to keep in mind, however, is that the feasibility of the augmentation algorithm hinges upon the ability to perform such implicit calculations, and there undoubtedly exist applications for which the covariances in Eq. (4.14) cannot be computed efficiently.

Another implementation issue is that the augmented state variables may have singular or ill-conditioned covariances. In this case, the augmented variables must be reduced before executing Eq. (4.14). One method for reducing the augmented variables is to use the singular estimators described in Chapter 2. The parameter A_s in Eq. (4.14) is such that $A_s\zeta(s\overline{\gamma})$ is the LLSE estimate of $\zeta(s)$ from $\zeta(s\overline{\gamma})$, while Q_s is the corresponding error covariance. If $\zeta(s\overline{\gamma})$ has an ill-conditioned covariance matrix, then the singular estimator in Eq. (2.14) can instead be employed. Namely, before computing the model parameters, the augmented states can be reduced to

$$\zeta_2(s) = L_s\zeta(s)\,, \tag{4.25}$$

where L_s removes the components of $\zeta(s)$ that have variances smaller than the machine precision. This projection should have little effect upon the resulting model, since the components discarded are near zero. The model parameters for the well-conditioned model are

$$P_0 = L_0 V_0 P_f V_0^T L_0^T\,, \tag{4.26a}$$

$$A_s = L_s P_{\zeta(s)\zeta(s\overline{\gamma})} L_{s\overline{\gamma}}^T (L_{s\overline{\gamma}} P_{\zeta(s\overline{\gamma})} L_{s\overline{\gamma}}^T)^{-1}\,, \tag{4.26b}$$

$$Q_s = L_s P_{\zeta(s)} L_s - L_s P_{\zeta(s)\zeta(s\overline{\gamma})} L_{s\overline{\gamma}}^T (L_{s\overline{\gamma}} P_{\zeta(s\overline{\gamma})} L_{s\overline{\gamma}}^T)^{-1} L_{s\overline{\gamma}} P_{\zeta(s\overline{\gamma})\zeta(s)} L_s^T\,. \tag{4.26c}$$

Note that both $P_{\zeta(s)}$ and L_s are required for the computation of the parameters (A_s,Q_s) and all $(A_{s\alpha_i},Q_{s\alpha_i})$. This implies that $P_{\zeta(s)}$ and L_s will be required when computing the parameters at scales $m(s)$ and $m(s)+1$. If $P_{\zeta(s)}$ and L_s are not stored when computing the model parameters for scale $m(s)$, one must be very careful when computing the model parameters for scale $m(s)+1$ (which involves computing both $P_{\zeta(s)}$ and L_s again). Very small perturbations in the coefficients of $P_{\zeta(s)}$ can lead to very large differences in the matrix L_s. For reasons of consistency, the use of different projection matrices L_s for calculating the coefficients at neighboring scales will lead to significant errors in the statistics of the realized model.

■ 4.3.5 Performance of the Augmented Multiscale Processes

The utility of the multiscale framework is the ability to efficiently provide statistical analysis in the form of optimal estimates and error covariances. Assuming the model parameters can be computed efficiently, the remaining question is how the state augmentations affect the computational efficiency of multiscale estimator. Remember that the number of computations required by the multiscale estimator increase cubicly with the dimension of each state of the tree. For each linear functional $\langle g, f\rangle$ placed at node τ by the algorithm of Section 4.3.3, the state at each node s in the subtree descending from τ will increase by q_s elements—unless τ is a descendent of σ, where σ is defined

in Section 4.3.3, in which case the state dimension at node τ increases by $q_\tau + 1$ and the state dimension at node σ increases by $q_\sigma - 1$. While the effect of this increase is insignificant when adding a single linear functional of f, the effect will be problematic when a large number of linear functionals must be added. Therefore, an important problem is to manage the dimension of the states in the augmented multiscale model. There are basically three methods for reducing the dimension of the state variables. First, as mentioned in the previous subsection, the state dimensions can be reduced whenever the augmented variables $\zeta(s)$ have ill-conditioned covariances. Second, as demonstrated in the following example, the nodes τ_i can be chosen intelligently so as to minimize the effect of the augmentation. Third, and most importantly, approximate multiscale models which sacrifice statistical accuracy for computational efficiency can be employed. Some methods for approximation are discussed in Section 4.4.

One-Dimensional Flow

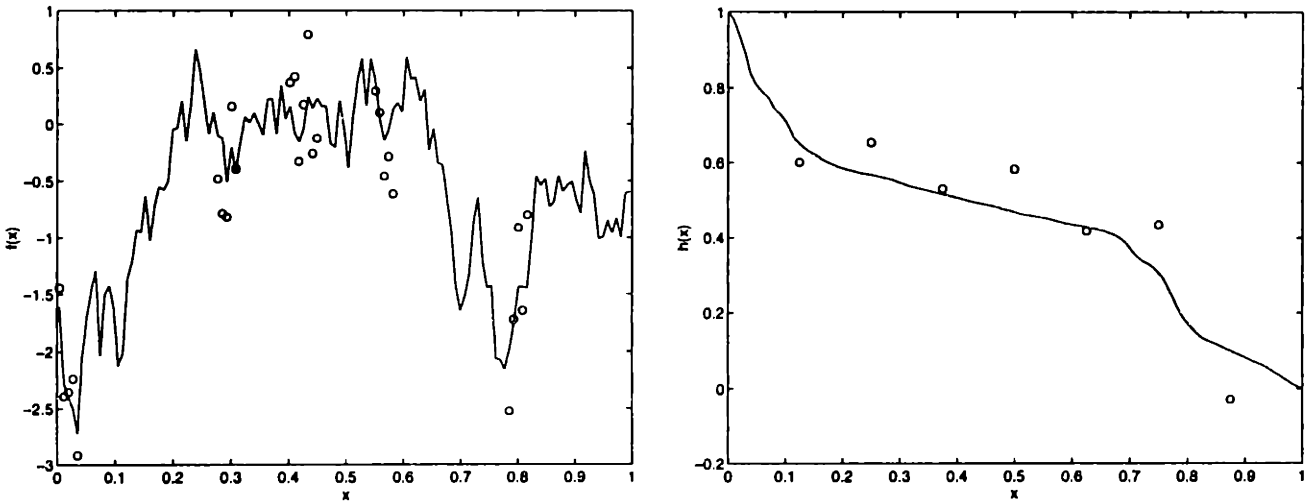
The problem of optimal node placement is best illustrated by example. We consider the estimation of hydraulic conductivity for 1D flow from measurements of conductivity and head. (See Chapter 3 for a description of the hydraulic conductivity estimation problem.)

For steady-state flow in 1D, consider estimating log-conductivity on the interval $x \in [0,1]$. The boundary conditions are $h(0) = 1$ and $h(1) = 0$. Assume that $f(x)$ is a 1D first-order Markov process with zero mean and covariance

$$E\big[f(x)f(x+r)\big] = e^{-5|r|}. \tag{4.27}$$

Samples of this 1D Markov process can be mapped to the finest scale of one of the multiscale models described in Section 2.3.2. In particular, assume a binary tree with six scales, four samples per state, and $N = 128$ elements at the finest scale. (For $N = 16$, the samples of the Markov process represented by each variable of the multiscale model are illustrated in Figure 2.3b.) A sample path of f and the corresponding head function are illustrated in Figure 4.1, along with the noisy point measurements. The head function is observed at $x_i = i/8$, $i = 1, \ldots, 7$. The Fréchét derivatives of the head measurements are illustrated in Figure 3.4, i.e., the head samples are linearized about the mean of the conductivity function, $m_f = 0$.

The variables of the multiscale representation of the 1D Markov process can be augmented so that all of the head measurements are modeled at the root node of the tree. (The measurement support node σ is equal to the root node for all of the linearized head measurements.) The estimate $\hat{f}(x)$ of the finest-scale process of the multiscale model is given in Figure 4.2. The multiscale estimator also computes the estimation estimation error variances $E\big[\big(\hat{f}(x) - f(x)\big)^2\big]$, which are included in Figure 4.2 in the form of confidence intervals. The confidence intervals are equal to the LLSE estimate plus or minus a single standard deviation of the estimation error. As would be expected, most of the true conductivity function lies within the confidence interval.



(a) (b)

Figure 4.1. (a) A sample path of the log-conductivity function, and (b) the corresponding head function. The noisy measurements are indicated by o's.

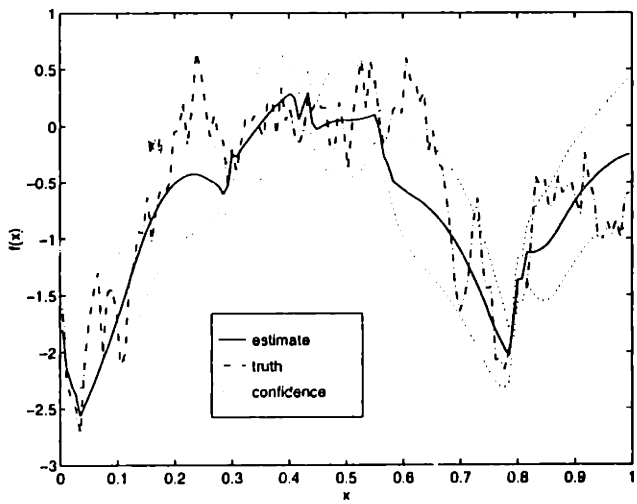


Figure 4.2. LLSE estimate of the log-conductivity function (solid line) along with the one standard deviation confidence intervals (dotted lines). The true conductivity function is also provided (dashed line).

A closer look at the state augmentation for the head measurements illustrates how the state dimensions of the augmented model can be reduced, even when no approximations are made. Assume that all seven head measurements are placed at the root node of the tree by recursively applying the algorithm of Section 4.3.3. The seven linearized head measurements are represented by the inner products $\langle g^i, f\rangle$, where $\langle g^i, f\rangle$ is the 128 sample Riemann sum approximation of $\int_{x=0}^{1} g(x_i, x \mid f_0) f(x)\, dx$. Using a naive application of the augmentation algorithm, the dimension of each state of the multiscale tree will increase by fourteen. However, all of these dimensions can be reduced. To see this, first note that the seven Fréchét derivatives are piecewise constant with a discontinuity at the corresponding measurement sample x_i; therefore, each Fréchét derivative is equal to a linear combination of the eight local averages

$$a_i = \int_{(i-1)/8}^{i/8} f(x)\, dx\,, \qquad i = 1, \ldots, 8\,.$$

Since the state $z(0)$ can be augmented with each of these local averages without destroying the Markov property of this state, this variable only needs to be augmented with eight elements. Secondly, $z(0\alpha_1)$ only needs to be augmented with $a_1, \ldots, a_4$, and $z(0\alpha_2)$ only needs to be augmented with $a_5, \ldots, a_8$. Finally, note that the discontinuities of all seven Fréchét derivatives lie at the boundaries of the eight finest-scale intervals partitioned by the four nodes at scale $m = 2$. Over each of these intervals, the Fréchét derivatives are constant, and thus linearly dependent. This "local linear dependence" means that $\langle g_s^i, f_s\rangle \propto \langle g_s^j, f_s\rangle$ for all nodes s at scales $m(s) > 2$. Therefore, the augmentation of any state $z(s)$ for $m(s) \geq 2$ is given by the two local averages over the two finest-scale intervals descendent from node s. The resulting augmentation is illustrated in Figure 4.3. The seven measurements are thus incorporated with only a minor increase in the state dimension, especially at the finer scale nodes. These increases are considerably less than would be predicted from a repeated application of the algorithm of Section 4.3.3, and are due to the local linear dependence of the kernels g^i over the finest-scale intervals partitioned by the nodes of the tree. Thus one can imagine modifying the structure of the tree models, i.e., tailoring the descendents of each node, to maximize this linear dependence and minimize the effect of the augmentation on the estimation algorithm.

Another way to reduce the effect of the state augmentation on the multiscale estimator is to distribute the measurements at various nodes on the tree, even though σ may be identical for each measurement. One problem with placing all the measurements at a single node is that the dimension of this node can become quite large, and the computations required by the multiscale estimator increase cubicly with each state dimension. For this example, keep $\langle g^4, f\rangle$ at the root node, but place $\langle g^2, f\rangle$ and $\langle g^6, f\rangle$ at nodes $0\alpha_1$ and $0\alpha_2$ and place $\langle g^i, f\rangle$ for $i = 1, 3, 5, 7$ at the four nodes at scale $m = 2$. In this case, by repeatedly applying the algorithm of Section 4.3.3 and also accounting for local linear dependence, the dimension of the state at the root node increases by only two, the states at scales $m = 1$ and $m = 2$ increase by three, and the remaining

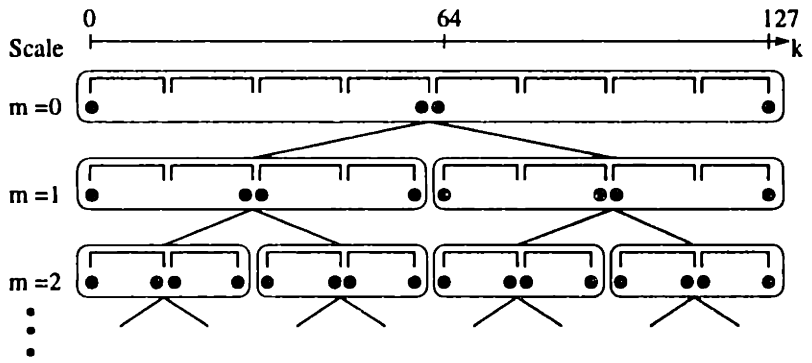


Figure 4.3. The states of the first three scales of the multiscale model for a 1D Markov process after the inclusion of the seven linear functionals illustrated in Figure 3.4. The brackets in each state represent local averages of the finest scale process.

states for scales $m > 2$ increase by 2. Thus a redistribution of the coarse-resolution functionals leads in this case to nontrivial computational savings.

In general, the "localization" of the nonlocal measurement kernels is rarely exact but instead approximate. For most applications, the problem will be to choose the measurement nodes τ_i such that the resulting multiscale model minimizes the associated estimator complexity for a given tolerance of the statistical accuracy of the multiscale model. The approximations made in the augmentation will depend on both the measurement kernels and the covariance (prior) of the finest scale process. For example, if the finest-scale covariance has very long correlation lengths, we might first discard fine-scale fluctuations in the measurement kernels before representing them on the tree.

■ 4.4 Approximate Realization Algorithms for Internal Models

All of the multiscale models discussed thus far are for modeling exactly the second-order statistics of the finest-scale process and any nonlocal functions represented at coarser-scale nodes. In this section, we first describe how the augmentation algorithm of the previous section can be applied to internal multiscale models that only approximately capture the desired second-order statistics. Next, we describe how to manage the increase in state dimension due to augmentation by developing approximate multiscale models. The key property of these approximate models is that they are *consistent.*

■ 4.4.1 State Augmentation for Approximate Multiscale Models

For internal multiscale models that exactly capture the joint statistics of the finest-scale process and the nonlocal functions to be estimated or measured, the internal variables $z(s) = W_s b_s = V_s f$ exactly decorrelate the $q_s + 1$ vectors in in Eq. (4.10). For approximate multiscale models, this decorrelation is only approximate, i.e.,

$$\overline{p}(b_{s\alpha_1}, \ldots, b_{s\alpha_q}, b_{s\alpha_{q+1}} \mid V_s f) > 0\,, \tag{4.28}$$

where $\overline{\rho}$ is defined in Section 2.3.4. There are basically two methods—heuristic and optimal—for realizing multiscale models with approximate statistics. Heuristic methods are useful when the internal variables $V_s f$ for the exact models can be determined with little or no computation. An example is the multiscale model for MRFs described in Section 2.3.2. As illustrated in Figure 2.4a, the internal variable at node s for the exact model consists of the boundaries of the finest-scale process descending from node s. However, if the correlation distances of the MRF are more than a few samples long, then the samples along any boundary will be highly correlated. Thus, one can decrease the sampling rate along the boundaries, as illustrated in Figure 2.4b, while still approximately decorrelating the subsets of the random field at the finest scale. A similar multiscale model for a finest-scale process with Gaussian covariance is provided in [70].

An optimal approach to realizing approximate multiscale models is to choose the internal variables $V_s f$ so that errors in the model's statistics are minimized for some fixed cost function, e.g., the number of computations required by the multiscale estimator. By statistical errors, we mean errors in the multiscale model representation of the joint statistics of the finest-scale process and the nonlocal functions to be estimated or measured. This problem has not been satisfactorily solved, but some foundation was provided in [49]. The approach taken in [49] is to fix the dimensions of the internal variables[2], and then to choose V_s so as to minimize $\overline{\rho}$ in Eq. (4.28). Because the internal matrices V_s are chosen independently, the algorithm is called "myopic".

To augment the variables of these approximate multiscale models with arbitrary linear functions of the finest-scale process, we make use of Theorem 3 in Chapter 2. As shown in [49], this theorem leads to the following corollary.

Corollary *Assume that the linear combination $V_s f$ satisfies*

$$\overline{\rho}(b_{s\alpha_1}, \ldots, b_{s\alpha_q}, b_{s\alpha_{q+1}} \mid V_s f) = \varepsilon\,, \tag{4.29}$$

then for any linear combination $R_{s\alpha_i} b_{s\alpha_i}$, $i = 1, \ldots, q_s + 1$, the conditional correlation satisfies

$$\overline{\rho}(b_{s\alpha_1}, \ldots, b_{s\alpha_q}, b_{s\alpha_{q+1}} \mid V_s f, R_{s\alpha_i} b_{s\alpha_i}) \leq \varepsilon\,. \tag{4.30}$$

In other words, augmenting the internal variables $z(s) = V_s f_s$ with additional linear functions of $b_{s\alpha_i}$ (for individual values of i) will not increase the correlation among the vectors $b_{s\alpha_i}$. Because $b_{s\alpha_i}$ includes $f_{s\alpha_i}$, the augmentation algorithm described in the previous section can be used to add linear functions to approximate internal models. Fortunately, the augmentation can only increase the statistical accuracy of the multiscale model.

[2]Recall that [49] provides multiscale models for which only the finest-scale covariance are specified.

■ 4.4.2 Approximation Algorithm

In this section we discuss how to control the increase in the state dimensions due to augmentation. More generally, we show how to consistently reduce the state dimensions of multiscale models. The approximation which follows amounts to discarding elements of the tree variables which are either insignificant or which must be removed to have a manageable state dimension. However, just as care must be taken when augmenting the variables of internal multiscale models, care must be taken when discarding elements of these variables. In particular, the elements must be discarded so that the resulting multiscale model is *consistent.*

To define the consistency of multiscale models, consider the relationship between $z(s)$ and its children. Assume for now that $z(s) = W_s f_s$, which is the case if only the finest-scale covariance is specified. The autoregression yields

$$\begin{bmatrix} z(s\alpha_1) \\ z(s\alpha_2) \\ \vdots \\ z(s\alpha_{q_s}) \end{bmatrix} = \underbrace{\begin{bmatrix} A_{s\alpha_1} \\ A_{s\alpha_2} \\ \vdots \\ A_{s\alpha_{q_s}} \end{bmatrix}}_{A} z(s) + \begin{bmatrix} w(s\alpha_1) \\ w(s\alpha_2) \\ \vdots \\ w(s\alpha_{q_s}) \end{bmatrix}. \tag{4.31}$$

Define $z_\perp(s)$ and $z_\parallel(s)$ by

$$\begin{aligned} z(s) &= z_\parallel(s) + z_\perp(s), \\ z_\parallel(s) &= A^T\lambda, \\ A\, z_\perp(s) &= 0. \end{aligned}$$

The vector $\lambda = (AA^T)^{-1}A\, z(s)$ follows from these relations. The component $z_\perp(s)$ is not used in the prediction of the children of node s. Because $z_\perp(s)$ must be uncorrelated with the variables at any descendents of the children $s\alpha_i$, $i = 1, \ldots, q_s$, it can be removed from $z(s)$ without affecting the statistics of the finest-scale process or the estimation of the finest-scale process from observations of $z(s)$. An example of such inconsistency was provided in Section 4.3.2 by augmenting the root node with the average value of the finest-scale process. The problem in this example was that the descendents of the root node did not propagate the average value at the root node to the finest-scale descendents. The solution was to augment the **descendents** of $z(0)$ with local averages so as to make the model consistent. Analogously, we should expect that removing elements of $z(s)$ will require the removal of elements of the **ancestors** of $z(s)$ so as to make the model consistent. A consistent model is one with no redundant elements, i.e., $z_\perp(s) = 0$. In other words, all the elements of the state $z(s)$ are used to predict its children.

This consistency requirement can be used to reduce the state dimensions of the models produced by the Canonical Correlations algorithm of [49]. Because the internal variables $W_s f_s$ are chosen independently, there is no reason to believe that the

resulting model will be consistent. This inconsistency leads to unnecessarily large state dimensions. The unnecessary elements can be removed using the following steps:

- for each node s at scale $m(s) = M-1$, where M is the finest scale, set $z(s) = z_{\|}(s)$.
- compute the multiscale model autoregression parameters (A_s, Q_s) for each node $s \in \{t \,|\, m(t) = M\}$;
- repeat this process for scales $m = M-2, \ldots, 0$.

The final result provides the autoregression parameters of a consistent multiscale model. Note that the algorithm must begin at the finest-scale nodes of the tree. The reason is that the consistency requirements for the state at node s are given in terms of its children $s\alpha_i$; thus, the variables at the child nodes have to be determined first.

Now we can describe how to consistently reduce the state dimensions of multiscale models that have been augmented using the algorithm of Section 4.3.3. Assume that we have the augmented internal variables $\zeta(s) = \mathcal{V}_s f$. The consistency requirements are similar to those just described for multiscale models focusing on the finest scale process; the only difference is that some of the augmented portion of $\zeta(s)$ must be ignored when enforcing consistency. Remember from Section 4.3 that, for each nonlocal functional $g^T f$ placed at node s, the variables at node s and nodes descending from node s must be augmented with linear functions derived from $g^T f$. However, as argued in Section 4.3.3, the descendents only need to be augmented to be made consistent with $\langle g_s, f_s \rangle$; the component $\langle g_{s^c}, f_{s^c} \rangle$, if it is nonzero, is ignored. The reason is that $\langle g_{s^c}, f_{s^c} \rangle$ is generated at ancestors of node s, so the descendents of node s are not affected by this function. Therefore, the consistency requirements for the variable at node s do not include the components of the functionals placed at node s that are functions of f_{s^c}. To be more precise, first rewrite $\zeta(s)$ as

$$\zeta(s) = \begin{bmatrix} \mathcal{V}_s^a f \\ G_s^a f_{s^c} \end{bmatrix}, \tag{4.32}$$

$$= \mathcal{V}_s f, \tag{4.33}$$

where $G_s^a f_{s^c}$ contains the components $\langle g_{s^c}, f_{s^c} \rangle$ for the nonlocal functions placed at node s, i.e., all $g_i^T f$ for which $\tau_i = s$. The consistency requirement is that every linear combination of $\mathcal{V}_s^a f$ be used in the prediction of the children $\zeta(s\alpha_i)$, $i = 1, \ldots, q_s$. Those components not used are discarded from $\zeta(s)$.

To further reduce the dimensions of the augmented variables, we can systematically discard elements. There are a number of possible criteria for deciding what information to discard, e.g.,

- retain only the $J(s)$ linear combinations of $\zeta(s)$ that correspond to the $d(s)$ largest eigenvalues of $P_{\zeta(s)}$, or
- retain only the linear combinations of $\zeta(s)$ that correspond to eigenvalues of $P_{\zeta(s)}$ greater than ϵ_s, or

- retain only the $d(s)$ linear combinations of $\zeta(s)$ that maximally decorrelate $\zeta(s\alpha_1)$, $\ldots$, $\zeta(s\alpha_q)$.

The latter criterion is preferred, since this reduction is invariant to multiplying the elements of $\zeta(s)$ by scalars. In any case, the only problem is to ensure that any such reduction of the variables is done consistently. A fine-to-coarse scale approximation algorithm follows as:

- for every node at scale $m(s) = M - 1$, make the augmented state $\zeta(s)$ consistent with the variables at scale M; remember that only $\mathcal{V}_s^a f$ needs to be made consistent with the children of $\zeta(s)$; call $\zeta_{\|}(s)$ the new consistent variables at scale $M - 1$;
- reduce the $\zeta_{\|}(s)$ at scale $M - 1$ according to the particular reduction criterion, e.g., discard the linear combinations of $\zeta_c(s)$ corresponding eigenvalues less than ϵ_s;
- update the multiscale model autoregression parameters (A_s, Q_s) for each node $s \in \{t \,|\, m(t) = M\}$;
- repeat this process for scales $m = M - 2, \ldots, 0$.

The only trick is to keep track of how the particular nonlocal functions to be represented at coarser-scale nodes are affected by the variable reduction steps. Note that the consistency of $\zeta(s)$ is enforced before the variable reduction, which in turn ensures that the reduced variable is consistent with its children.

Chapter 5

Multiscale Modeling and Estimation of Hydraulic Conductivity

In this chapter, the multiscale framework is used to estimate hydraulic conductivity from measurements of both hydraulic conductivity and hydraulic head. The practical importance of, and problems associated with, hydraulic conductivity estimation were discussed in Chapter 3. The problems stem primarily from the natural variability of hydraulic conductivity and the difficulty in measuring it. Hydraulic conductivity is known to vary significantly over many spatial scales [38], exhibit long-range dependencies that increase with the scale of observation [38], and require nonstationary models to be characterized statistically [98, 99]. The only way to measure hydraulic conductivity directly is to extract samples from the earth's subsurface. These observations are expensive and limited in spatial coverage; instead, one must rely on indirect observations that range from analyzing the reflection of acoustic waves across geologic boundaries to measuring the change in fluid pressure that results from injecting fluids into the side of a wellbore. Because these measurements are sparse and limited in spatial resolution, it is impossible to determine the hydraulic conductivity function at every scale of variation. Instead, additional information must be supplied, usually in the form of a prior probability distribution and/or a finite-dimensional parameterization of the hydraulic conductivity function.

As noted in Chapter 3, the spatial resolution of the conductivity parameterization is usually chosen to be constant [98]. Furthermore, this constant resolution is chosen as close as is computationally feasible to the resolution of the finest-scale measurement. This approach can lead to a prohibitively large number of parameters to be estimated, even when the number of measurements is small. The class of multiscale models provides an alternative parameterization. Because the phenomenon of interest is modeled at multiple resolutions, the estimate can have resolution that varies according to the distribution of the resolution in the measurements. Furthermore, using the algorithms provided in Chapter 4, coarse resolution and nonlocal measurements can be modeled and naturally incorporated at coarse scale nodes of the multiscale tree. These multiscale processes capture the relationship between the measurements made at different scales and the conductivity function at each scale represented on the tree.

This chapter focuses on the estimation of hydraulic conductivity for 2D flow from observations of head and conductivity. This problem has been studied by many others, e.g., [1, 68, 74, 97]. Other indirect observations of hydraulic conductivity, such as tracer tests and contaminant concentrations, are usually available in field studies [38, 69]. The incorporation of these measurements can lead to considerable improvements in the accuracy of the log-conductivity estimate. However, computing conductivity estimates from numerous measurement sources, which usually provide observations at different resolutions, is a difficult problem and no consistent and computationally feasible framework exists for assimilating these various measurement sources. The multiscale framework is one possibility, and, although only head and conductivity measurements are considered in this chapter, the general approach that we describe applies to these broader classes of measurements. (In particular, in Chapter 6 we demonstrate how tracer-test measurements can be incorporated within the multiscale framework.) This chapter begins in Section 5.1 with a discussion of the influence of head measurements on conductivity estimates. These results demonstrate that the contribution of head measurements to hydraulic conductivity estimates depends heavily on the experimental setup, e.g., the geometry of the head and conductivity sampling, the existence of pumping and injection wells, and the nature of the boundary conditions. In Section 5.2, the multiscale framework is applied. In particular, the nonlocal head measurements are represented as point measurements of coarse-scale variables, while the conductivity measurements are represented as point measurements of the finest-scale process. The head measurements are incorporated into the multiscale framework using the state augmentation algorithm of Chapter 4. The examples of Section 5.2 demonstrate that nontrivial data fusion problems can be solved within the multiscale framework. In Section 5.3, the estimates of hydraulic conductivity are used to re-linearize the head measurements. This re-linearization is then incorporated into an iterative algorithm that computes the Gauss-Newton maximization of the posterior probability density. If the maximization produces a global extremum, then the MAP estimate is produced.

■ 5.1 The LLSE Estimation of Conductivity from Head Measurements

Before applying the multiscale framework, we first analyze the influence of head measurements on LLSE estimates of hydraulic conductivity. The aquifer is assumed to be in steady state, so that the hydraulic head is related to conductivity by

$$\nabla \cdot (K \nabla h) = Q . \tag{5.1}$$

This equation is discussed in detail in Chapter 3. In this chapter, only two-dimensional flow is considered. Two-dimensional flow describes horizontal flow in an aquifer of limited vertical extent or flow in an aquifer that has little conductivity variation in the vertical direction. In either case, the conductivity function to be estimated is a vertical average of conductivity, which is called transmissivity in the groundwater literature [27, 38]. For simplicity of exposition, we will not make this distinction.

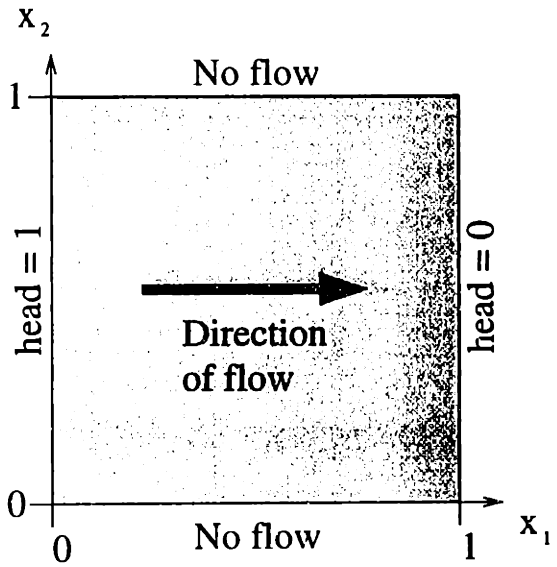


Figure 5.1. An illustration of the boundary conditions for flow in the $\mathbf{x}_1$ direction.

Assume for the following examples that log-conductivity is a Markov Random Field with zero mean and covariance

$$E\big[f(x)f(x+r)\big] = \sigma^2 e^{-|r|^T d}, \tag{5.2}$$

where $|r|^T = [|r_1|, |r_2|]$. The parameters d_1 and d_2 are the inverse correlation distances in the $\mathbf{x}_1$ and $\mathbf{x}_2$ directions, respectively. For now, fix $\sigma^2 = 1$ and $d = [3/2, 3/2]$. We will analyze the problem of estimating N_f-by-N_f evenly spaced samples of f on the unit square $\Omega = [0,1] \times [0,1]$. The head measurements are linearized about $f_0 = 0$, the mean conductivity, using the adjoint method described in Section 3.3. Also assume that the head measurement errors have variance $\sigma_h^2 = 0.005$.

For the first example, assume that $Q = 0$ and that the boundary conditions are given by the following: $h = 1$ along $\mathbf{x}_1 = 0$, $h = 0$ along $\mathbf{x}_1 = 1$, and the water flux normal to the boundaries $\mathbf{x}_2 = 1$ and $\mathbf{x}_2 = 0$ is equal to zero. These boundary conditions are illustrated in Figure 5.1. The log-conductivity function to be estimated is plotted in Figure 5.2 and the corresponding head function is plotted in Figure 5.3a. Head measurements are provided at the fifteen locations marked in Figure 5.3b. All of these measurements are in the vicinity of the line $\mathbf{x}_1 = 0.5$. Assume for all of the examples in this section that the head measurement noise has variance $\sigma_h^2 = 0.005$.

The LLSE estimate of log-conductivity from the head measurements in Figure 5.3b and the associated error variances of each log-conductivity sample are plotted in Figure 5.4. Remember that the LLSE estimate and error covariance are actually only approximations based on the linearization of the head measurement equation; therefore, one must be careful in drawing any conclusions about uncertainty reduction from the error variance plots. The most striking feature of the estimate in Figure 5.4 is

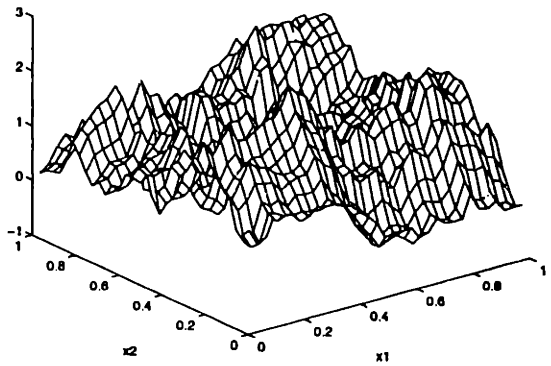


Figure 5.2. A log-conductivity function generated assuming that $\sigma^2 = 1$ and $d = [3/2, 3/2]$ in Eq. (5.2).

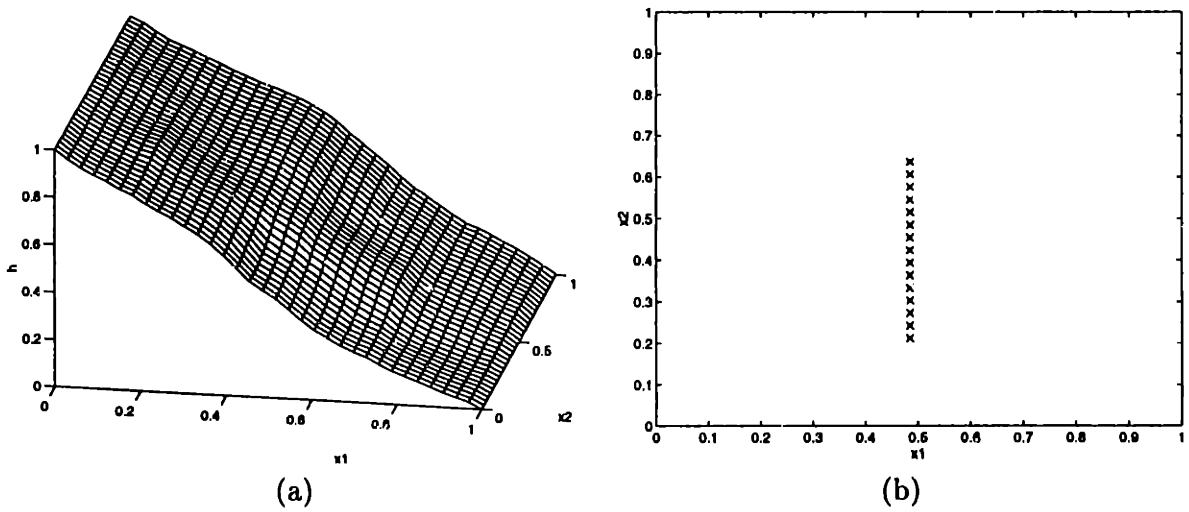


Figure 5.3. (a) The head function produced by the log-conductivity function in Figure 5.2 and the boundary conditions in Figure 5.1. (b) Fifteen head measurements located along a single value of x_1.

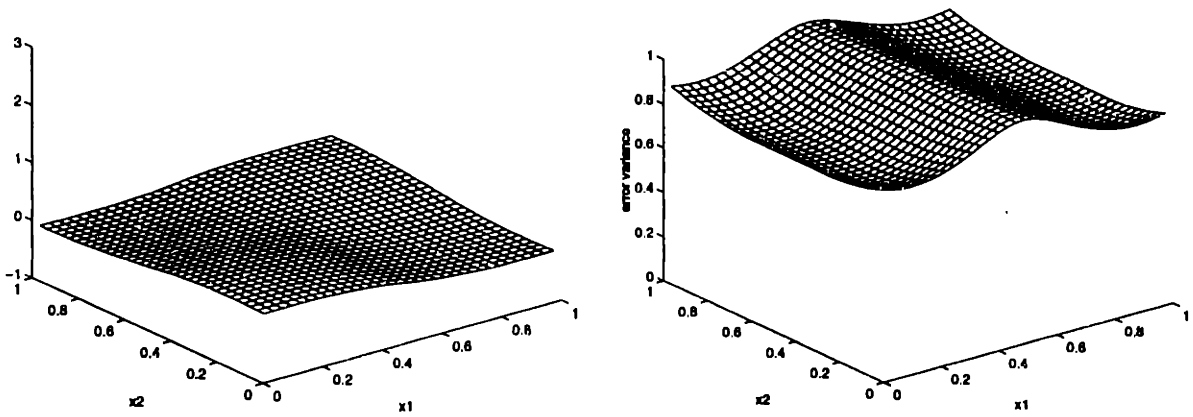


Figure 5.4. (a) The LLSE estimate of log-conductivity from the head measurements at the locations in Figure 5.3b and (b) the corresponding estimation error variances.

the lack of variation in the estimate, even near $\mathbf{x}_1$. There are two reasons for this. First, as noted in Section 3.4, the head measurements for this scenario are insensitive to conductivity values near the measurement location. This insensitivity is manifested in the estimation error variances near $\mathbf{x}_1 = 0.5$, which are close to the prior variance of $\sigma^2 = 1$. The other reason, also noted in Section 3.4, is that head measurements in this scenario provide no information about the DC value of log-conductivity. All of the DC information in the estimate is supplied by the prior mean of zero.

An explanation for the insensitivity of the head samples to absolute values of conductivity is provided by the following analysis [67]. The steady-state flow equation for $Q = 0$ is

$$\nabla \cdot (K \nabla h) = 0 .$$

If the head function $h(x)$ is known everywhere on the domain of interest, then the flow equation is equal to a first-order ODE for K along each streamline, i.e.,

$$\frac{d}{ds}(a(x)\, K(x)) = 0 ,$$

where s is the direction along the streamline and $a(x)$ is a function of $h(x)$. The streamlines follow from the vector field ∇h, which is known. However, to uniquely determine K along each streamline requires an initial condition or final condition for K. Thus, even when h is known everywhere in the domain of interest, the problem of estimating K is ill-posed.

For the second example, consider the fifteen head samples at the locations in Figure 5.5. Unlike the previous measurement geometry, these measurements provide information about ∇h in the direction of flow. The estimate based on these head measurements and the corresponding error variances are illustrated in Figure 5.6. This

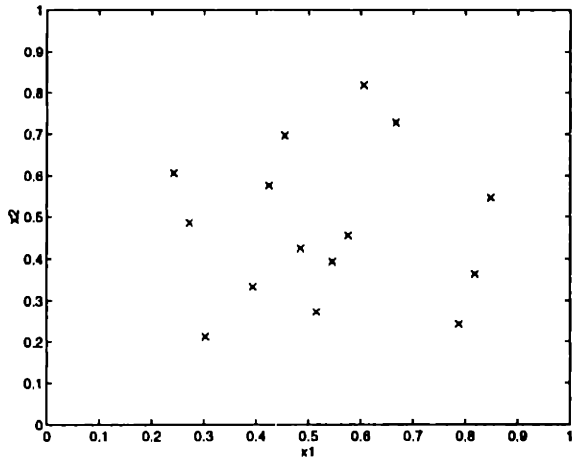


Figure 5.5. The locations of fifteen head samples.

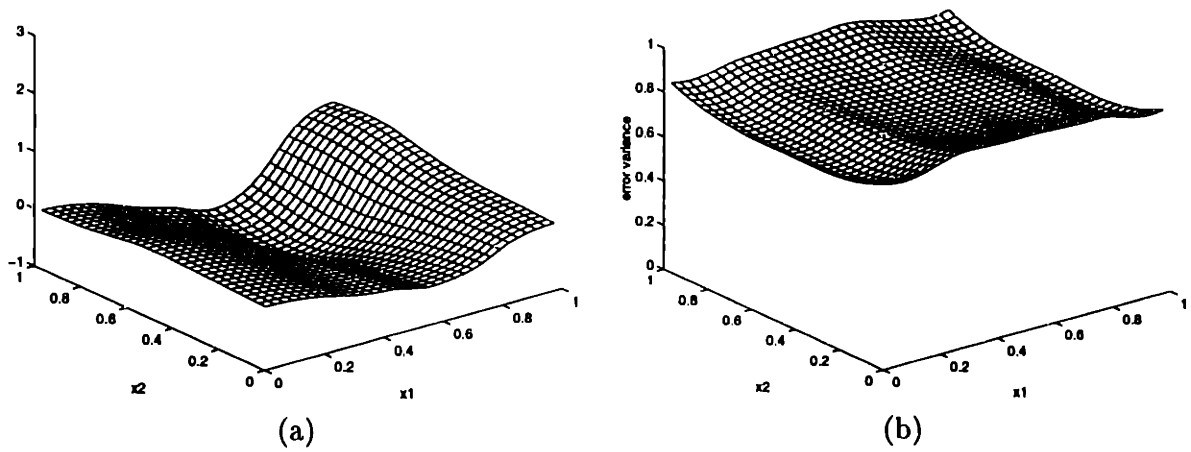


Figure 5.6. (a) The LLSE estimate of log-conductivity from the head measurements at the locations in Figure 5.5 and (b) the corresponding estimation error variances.

estimate has considerably more variation than the estimate in Figure 5.4, including a correct prediction of large conductivity values near $x = (1, 1)$; also, the error variances are generally smaller than those in Figure 5.4b. However, the estimate of $f(x)$ is still very poor and the reduction in uncertainty is still quite small.

The influence of the head measurements changes considerably when $Q \neq 0$ and Q is strong enough to considerably influence flow behavior in the aquifer. Consider again the estimation of conductivity assuming that head is known everywhere in the domain of interest, but that $Q = Q_0 \delta(x - x_s)$, which is a point source idealization of an injection ($Q_0 > 0$) or pumping ($Q_0 < 0$) well. If the source strong enough, then all streamlines will originate at $x = x_s$. Assuming Q_0 is known, then K can be determined at the

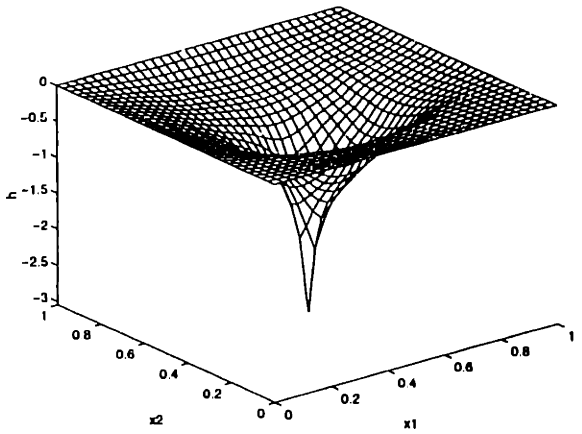


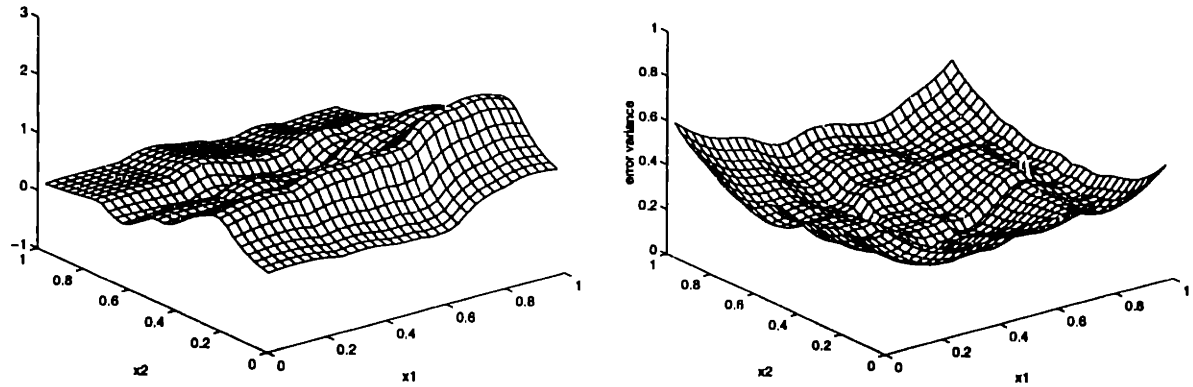
Figure 5.7. The head function for a pumping well at $x_s = (0.5, 0.5)$.

wellbore from Darcy's Law[1]. Since the wellbore is the origination of each streamline, K can be determined everywhere in the reservoir. Thus, head measurements should have significant influence on conductivity estimates when the source function is known and exerts influence on flow in the aquifer.

For the third example, consider the case when Q is a pumping well located at $x_s = (0.5, 0.5)$ and is the primary determinant of flow over the domain of interest, Ω. For the boundary conditions, assume that $h = 0$ along the entire boundary of Ω. These boundary conditions are chosen so that every point in the aquifer is crossed by a streamline originating from $x_s = (0.5, 0.5)$, which will allow us to illustrate the scenario in which head measurements maximally constrain the conductivity function. The head function is illustrated in Figure 5.7. Assume again that the head samples are located at the points illustrated in Figure 5.5. The LLSE estimate of hydraulic conductivity and the corresponding error variances are shown in Figure 5.8. Note that the reduction in uncertainty, as measured by the error variance, is much greater than for the two previous examples. Also, the log-conductivity estimate has much more structure. However, the head measurements alone are still unable produce a very accurate estimate of the conductivity function. This problem will be remedied somewhat when conductivity measurements are included along with the head measurements.

In the following section, the multiscale framework is applied to the estimation of hydraulic conductivity from head and conductivity samples. The introduction of conductivity measurements will generally decrease the influence of the head measurements. Head measurements will provide the most influence in regions where no conductivity samples are present. Also, head measurements, unlike conductivity measurements, can constrain the conductivity estimates in the interwell regions where no measurements are made.

[1]Here we have made the assumption that the wellbore has nonzero radius, so that the head gradient is finite at the wellbore.



(a) (b)

Figure 5.8. (a) The LLSE estimate of log-conductivity from measurements of the head function in Figure 5.7 at the locations in Figure 5.5. (b) The corresponding estimation error variances.

■ 5.2 Applying the Multiscale Framework to Conductivity Estimation

In Section 4.3.5, the multiscale framework was applied to the estimation of hydraulic conductivity for one-dimensional flow. A considerably more challenging problem is the estimation of hydraulic conductivity for two-dimensional flow. To apply the multiscale framework to the estimation of 2D conductivity from measurements of head and conductivity requires

- the specification of multiscale stochastic process that describes the conductivity function at all scales of interest and
- the representation of both the head and conductivity measurements as point observations of the multiscale process.

If the finest scale of the multiscale tree represents samples of hydraulic conductivity, then the conductivity measurements can be represented as point observations of the finest-scale process. However, as discussed in Section 3.3, samples of hydraulic head are nonlocal and nonlinear functions of conductivity. To apply the multiscale framework, the head measurements must be linearized about some conductivity function f_0. The adjoint method for computing a linear relationship between point values of head and the conductivity function is described in Chapter 3. Because the linearized head measurements are nonlocal, they are most naturally modeled at the coarser scales of the multiscale process. In this section, the state augmentation algorithm of Section 4.3 is used to represent the nonlocal head measurements at the coarser scales of the tree.

The use of the state augmentation algorithm implies that an internal multiscale model already exists for the conductivity function. In this chapter, the finest-scale samples of the log-conductivity function are modeled (a priori) as discrete-index Markov

Random Fields. Multiscale models for MRFs were described in Section 2.3.2. The variables of these models consist of dense samples along the boundaries of the finest-scale regions partitioned by each node—see Figure 2.4. We again assume that $f(x)$ has zero mean covariance given by Eq. (5.2). The N_f-by-N_f rectangularly spaced samples of this process form a discrete-index Markov random field[2]. As shown in Section 2.3.2, if $N_f = 2^{M+1}+1$, then these samples are naturally mapped to the finest scale of a quad-tree with M scales. However, we must emphasize that the multiscale framework is in no way restricted to such a simple process, or even to MRFs. Markov Random Fields are only chosen because the corresponding multiscale processes are easily visualized.

We now consider a number of example applications of the multiscale framework. For each example, both the performance of the multiscale framework (in terms of computational efficiency, accuracy in computing the LLSE, etc.) and the results of the conductivity estimation will be analyzed.

■ 5.2.1 Example: Horizontal Flow, Head and Conductivity Samples at Identical Locations

For this example, consider estimating 33-by-33 ($N_f = 33$) samples of log-conductivity on the unit square $\Omega = [0,1] \times [0,1]$. The steady-state flow satisfies Eq. (5.1) with $Q = 0$ and the boundary conditions are illustrated in Figure 5.1, which implies flow in the $\mathbf{x}_1$ direction. The covariance of the log-conductivity samples is given by sampling Eq. (5.2) with $\sigma^2 = 0.5$ and $d = [3/2, 3/2]$.

The measurements consist of twenty pairs of head and conductivity samples, located at the points shown in Figure 5.9. These samples might correspond to the measurements provided in twenty wells at a common depth or geologic stratum. The measurement noises for the head and conductivity measurements have variance $\sigma_h^2 = 0.001$ and $\sigma_f^2 = 0.1$, respectively.

The N_f-by-N_f finest-scale samples of conductivity are mapped to variables at the finest scale of a quad-tree with $M = 4$ scales. The conductivity measurements are easily incorporated as point measurements of the finest-scale process. The nonlocal head measurements (linearized about $f_0 = 0$) can be represented at coarser scales of the tree using the state augmentation algorithm of Section 4.3. The four head samples nearest $x = (0.5, 0.5)$ are represented at the root node. Each of the remaining head measurements is represented at the node at scale $m = 1$ whose finest-scale descendents include the quadrant of Ω in which the head measurement is located.

Because the linearized head measurements are nonlocal (the Fréchét derivatives are nonzero for all Ω), the state dimensions at each node on the tree should increase by $20q$ ($q = 4$) at each node after applying the augmentation algorithm of Section 4.3. However, before the model parameters are computed from Eqs. (2.30) and (2.32), the state dimensions can be reduced by accounting for probabilistic dependence among the elements of each internal variable. (See Section 4.3 for a more thorough discussion.)

[2]In general, sampling a continuous-index MRF does not produce a discrete-index MRF [15].

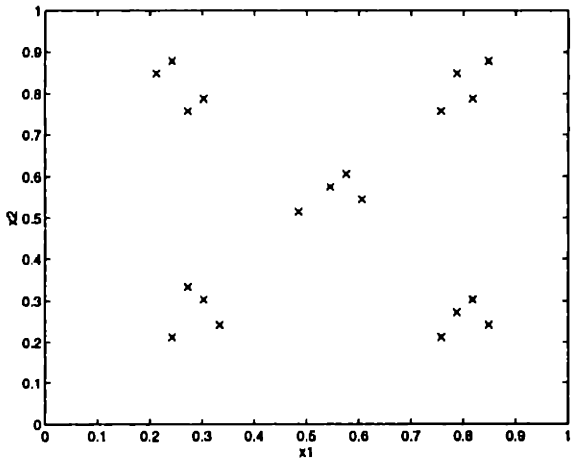


Figure 5.9. Each × corresponds to location at which both the head and conductivity function are sampled.

The dimensions of variables at scales zero and one of the multiscale tree model for the aforementioned MRF increase by forty. While less than eighty (4*20), these increases in the state dimensions are still significant and lead to large increases in the number of computations required by the multiscale estimator. The problem is that we have modeled the head measurements exactly, even though large additional reductions in the state dimensions can likely be achieved without significant degradation of the statistical relationships. One such approximate algorithm was discussed in Section 4.4, but it is not implemented in this thesis.

The total complexity required by the multiscale estimation algorithm is approximately 500 M-flops, while a direct implementation of the LLSE estimation and error covariance equations requires approximately 250 M-flops. This comparison, however, is somewhat meaningless, since the two problems are very different. The multiscale algorithm produces an estimate and error variance for the conductivity process at multiple scales, while the standard LLSE estimate produces only a finest-scale estimate and the corresponding complete error covariance matrix. However, this is the largest-sized problem for which the normal equations for standard LLSE estimation can be implemented on our computer. Storing just the covariance matrix of the finest-scale process requires $33^4 = 1.2$ million storage elements. If the linear dimension of the domain is doubled, storing the covariance in double precision arithmetic would require $8 * (2 * 33)^2 = 150$ megabytes. Fortunately, this $\mathcal{O}(N_f^4)$ growth in complexity does not hold for the multiscale estimator. Thus the real advantage of the multiscale algorithm will be for larger-sized domains, for which the standard implementation of the LLSE normal equations requires too many storage elements.

The log-conductivity function to be estimated and the corresponding head function are illustrated in Figures 5.10 and 5.11. The LLSE estimate of the process at the finest scale of the multiscale tree is illustrated in Figure 5.12, while the estimation

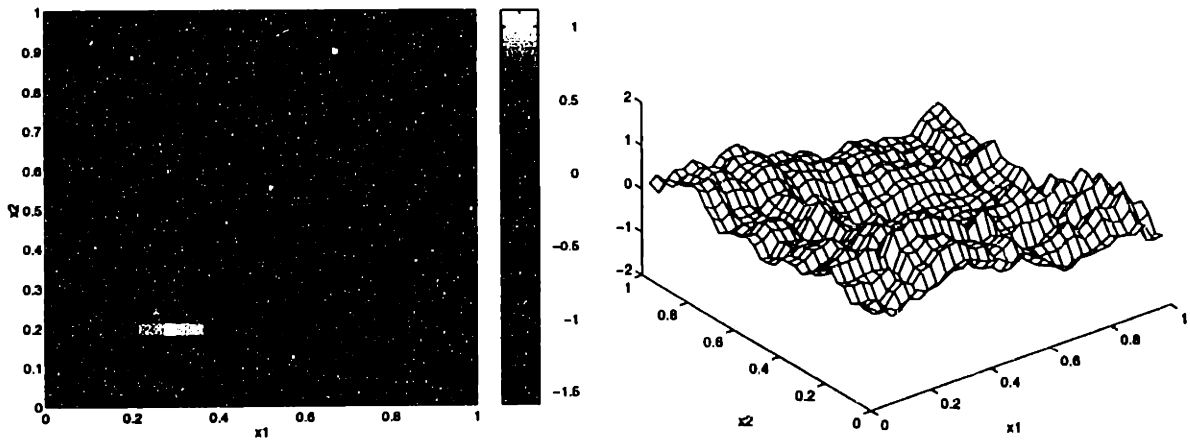


Figure 5.10. A sample path of the log-conductivity function, plotted in (a) gray scale and (b) using a mesh plot.

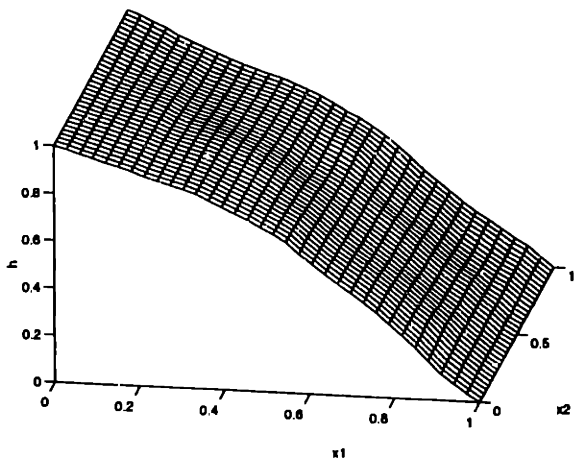


Figure 5.11. The head function produced by the conductivity function in Figure 5.10 and the boundary conditions in Figure 5.1.

error variances are illustrated in Figure 5.13. Recall that the unconditional variance of log-conductivity is 0.5. Note that the estimate has fine-scale variations only near the conductivity measurements.

■ 5.2.2 Example: Flow in a Vertical Slice

While the previous example considered flow in the horizontal plane, we now present an example[3] for which $\mathbf{x}_2$ corresponds to depth. Again consider estimating 33-by-33 samples of log-conductivity on the unit square $\Omega = [0,1] \times [0,1]$, but assume an anisotropic

[3]This example is drawn directly from [21].

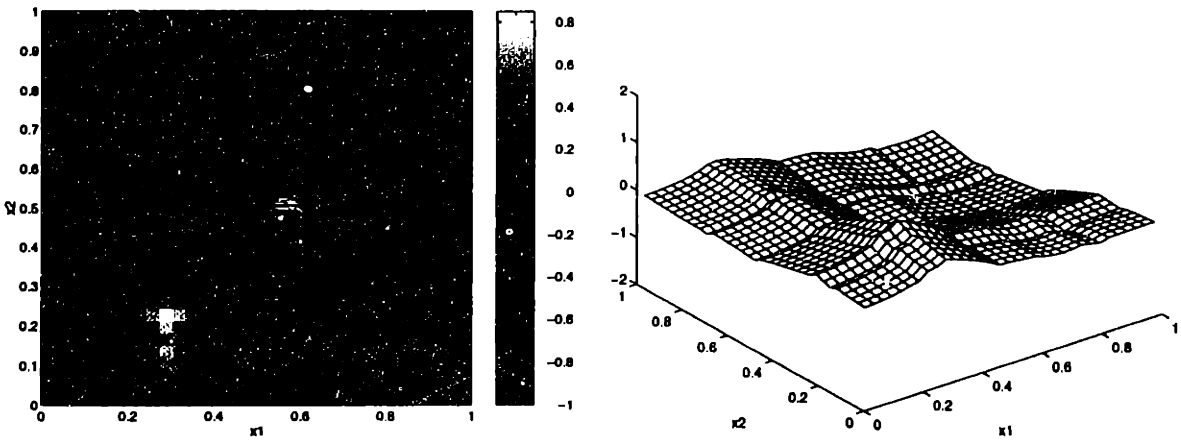


Figure 5.12. The LLSE estimate of the log-conductivity function in Fig. 5.10: (a) gray scale image, (b) mesh plot.

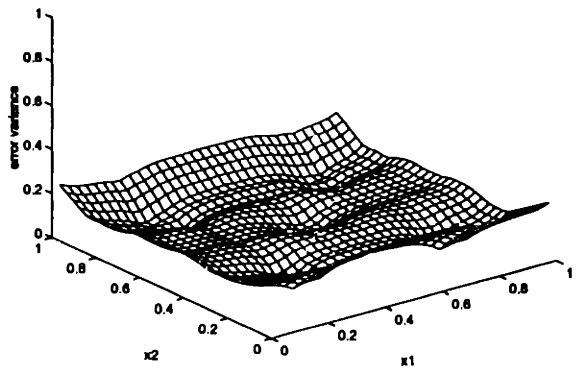


Figure 5.13. The variance of the estimation errors associated with the log-conductivity estimate in Figure 5.12.

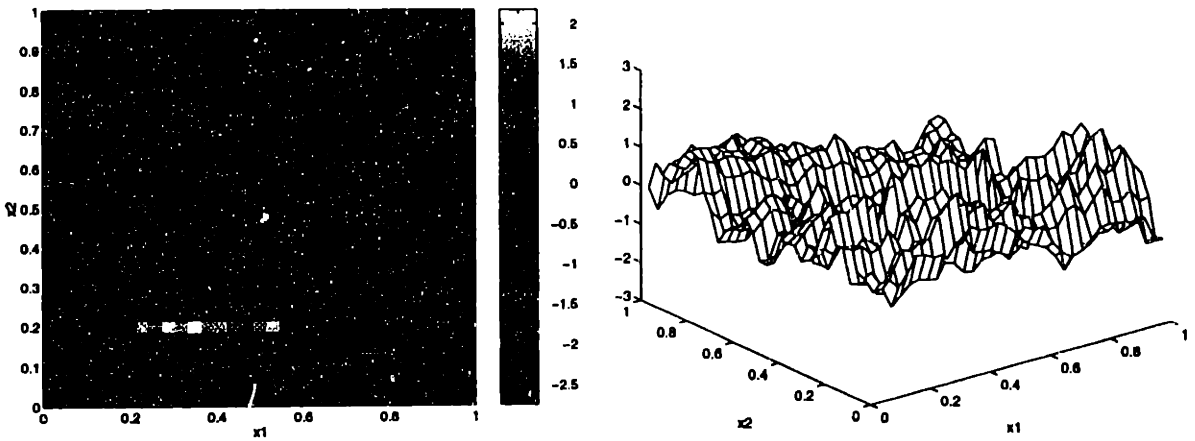


Figure 5.14. A sample path of the log-conductivity function plotted in (a) gray scale and (b) mesh plot.

covariance function with $d = [5/3, 6]$ and $\sigma^2 = 1$ in Eq. (5.2). This covariance implies that the log-conductivity function has stronger correlation in the horizontal direction ($\mathbf{x}_1$) than in the vertical direction ($\mathbf{x}_2$), which is typical of groundwater aquifers that arise from sedimentary deposition. A sample path of this discrete-index MRF is plotted in Figure 5.14.

Flow is again given by Eq. (5.1) with $Q = 0$ and the boundary conditions illustrated in Figure 5.1. The head function is plotted in Figure 5.15a. The measurements provided are the 123 conductivity samples and the 20 head samples illustrated in Figure 5.15b. The head and conductivity samples are no longer assumed to be at the same locations. The measurement geometry is meant to simulate measurements along fully or partially penetrating wells. The measurement noises for the head and conductivity measurements again have variances $\sigma_h^2 = 0.001$ and $\sigma_f^2 = 0.1$, respectively.

The LLSE estimates of the finest-scale process and the associated error variances are plotted in Figures 5.16 and 5.17, respectively. Note that the error variances decrease much more significantly in regions of dense conductivity sampling than in regions of dense head sampling, illustrating the minimal influence of the head measurements when $Q = 0$. More importantly for multiscale modeling, note that the conductivity estimate has fine-scale variations only where such variations can be inferred from the data, e.g., near the line $\mathbf{x}_1 = 0.12$ in Figure 5.16. This suggests reducing the number of parameters used to describe the conductivity estimate in areas where the estimate is smooth. Namely, there is no reason to model hydraulic conductivity to very fine resolution if such variables cannot be justifiably estimated from the data. The advantage of a multiscale parameterization is that the finest resolution at which the process is modeled can be varied in different regions of the aquifer according to the distribution of resolution supplied by the measurements. The only problem is how to determine the distribution

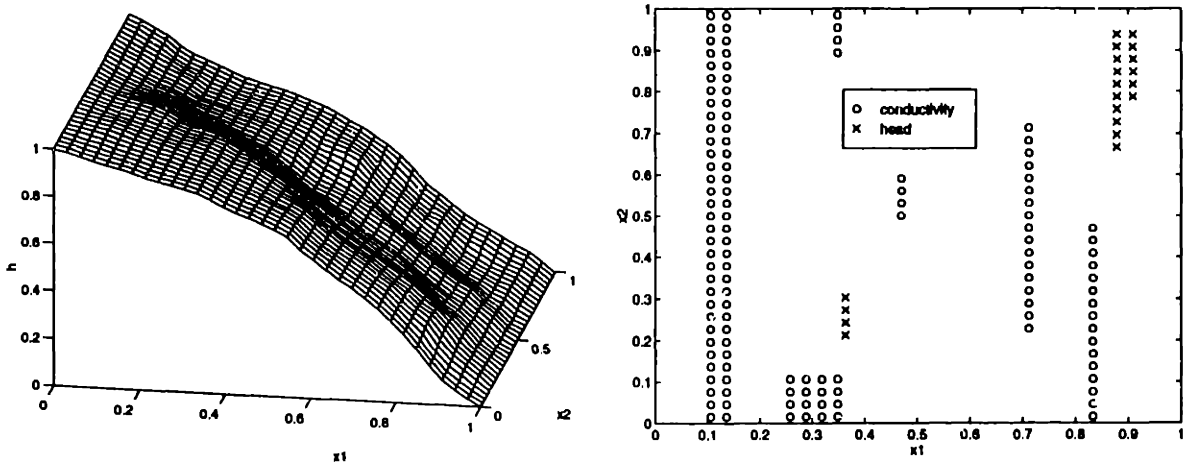


Figure 5.15. (a) The head function produced by the log-conductivity function in Figure 5.14 and the boundary conditions in Figure 5.1. (b) The locations of the conductivity measurements (o's) and head measurements (×'s).

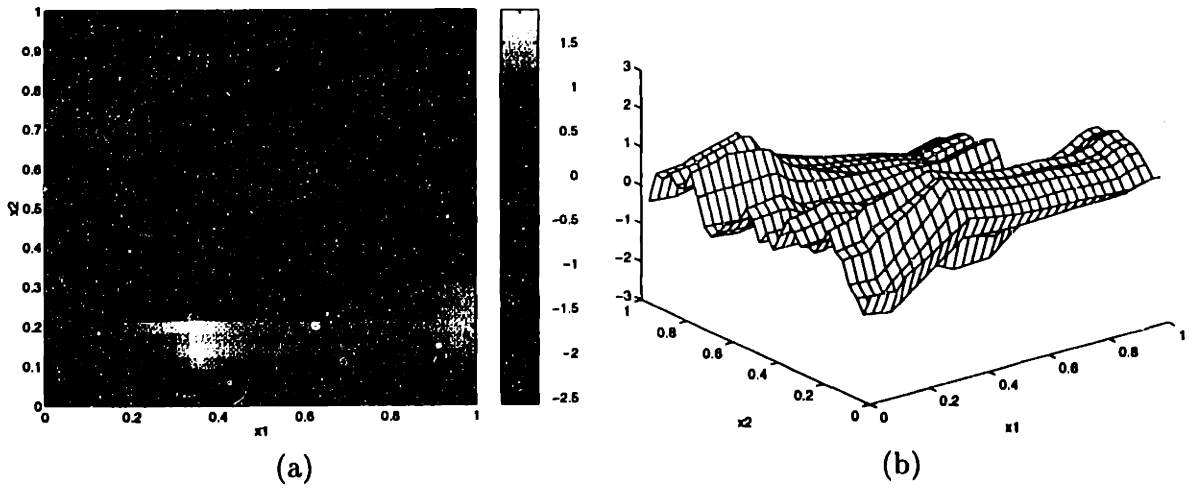


Figure 5.16. The LLSE estimate of the log-conductivity function in Fig. 5.14: (a) gray scale image, (b) mesh plot.

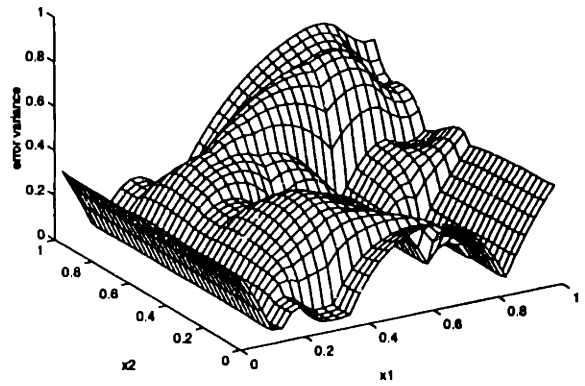


Figure 5.17. The variance of the estimation errors associated with the log-conductivity estimate in Figure 5.16.

of resolution a priori, before realizing the multiscale model. Such modeling is beyond the scope of this thesis, but is absolutely necessary if the multiscale framework (at least in the capacity presented in this thesis) is to be successfully applied to large data fusion problems. Also, within a full nonlinear optimization, where for each iteration the head measurements are linearized about the current conductivity estimate, this approach might eventually be used to reduce the number of parameters used to re-linearize the head measurements, and hence reduce the number of computations required for the conductivity estimation. The re-linearization of the head measurements is discussed in the following section.

For this example, the multiscale estimator requires slightly fewer computations (380 M-flops) than does a direct solution to the normal equations (470 M-flops) for producing the LLSE estimate and the corresponding error variances. Compared to the previous example, the number of computations required for the multiscale estimator has decreased. The reason for this decrease is that the twenty head measurements used in this example are highly correlated, so that the increase in state dimensions from augmentation is smaller once probabilistic dependence is accounted for. The increase in the number of computations required for the normal equations is due to the increase in the number of conductivity measurements. An advantage of the multiscale estimator is that the number of measurements has little effect on number of computations required. Note that the overall sampling density of the finest-scale process for our 2D example is low, as illustrated in Figure 5.15b, so the multiscale framework will compare more favorably as the number of finest-scale measurements increases.

■ 5.3 MAP Estimation: Nonlinear Optimization

The LLSE estimators described in the previous sections, whether implemented using the multiscale framework or standard solutions to the normal equations, are based

on linearizations of the head measurement equation. Because the covariance of the head measurements and the cross-covariance between head and conductivity functions are approximated from the linearization, the estimates and error covariances are approximations of the exact LLSE estimator. (See Section 2.1.2 for a more complete discussion of approximations to LLSE estimators when the measurements are nonlinear.) In all of the previous examples, the zero function ($f_0 = 0$) is used as the point of linearization. Define $\hat{f}$ to be the estimate based on this linearization. Since we expect $\|f - \hat{f}\| < \|f - f_0\|$, the linearization should generally be improved if $\hat{f}$ rather than the zero function is used as the point of linearization. The newly linearized head measurement equation can then be used to obtain an approximation of another LLSE estimate. This process can be repeated iteratively, and the general algorithm (expressed in terms of the the normal equations) is summarized by Eq. (2.21). As noted in Section 2.1.2 (see [69] for details), this iterative procedure is the Gauss-Newton solution of the MAP estimator; in other words, this procedure is the Gauss-Newton minimization of the cost function $J(f)$ defined in Eq. (2.19).

The Gauss-Newton solution of the MAP estimator can be implemented within the multiscale framework, but there are a number of issues to be considered. For instance, at each iteration the head measurements are linearized about the present estimate of the finest-scale conductivity function, $\hat{f}_k$. As $\hat{f}_k$ changes, so do the Fréchét derivatives of the head samples. Since the Fréchét derivatives change, the state augmentation algorithm of Section 4.3 must be re-applied at each iteration. For K total iterations, this will imply both K implementations of Eqs. (2.30)-(2.32) and K calculations of the Fréchét derivatives, which can be quite costly if K is large. An important question, then, is how fast the Gauss-Newton iteration converges.

■ 5.3.1 Example: Horizontal Flow, Head and Conductivity Samples at Identical Locations

Consider again the example discussed in Section 5.2.1. To initialize the Gauss-Newton iteration, set $\hat{f}_0 = 0$. The multiscale framework, in conjunction with the augmentation algorithm of Section 4.3, will be used to successively compute estimates of the log-conductivity function. Note that $\hat{f}_1$ was computed in Section 5.2.1. At each successive iteration, the head measurements are linearized about the most recently computed estimate of the finest-scale process, $\hat{f}_k$. Once these Fréchét derivatives have been computed, the multiscale model parameters can be updated using the augmentation algorithm of Section 4.3. The estimates then follow from an application of the multiscale estimation algorithm.

The convergence of the Gauss-Newton iteration is plotted in Figure 5.18, where $\hat{f}_k$ is the estimate of the finest-scale process after the k-th iteration. The estimate after six iterations and the difference $\hat{f}_6 - \hat{f}_1$ are plotted in Figure 5.19. Note from Figure 5.18 that the Gauss-Newton estimate converges geometrically as a function of k to its final value $\hat{f}_\infty$, and that the difference $\|\hat{f}_{k+1} - \hat{f}_k\|$ is quite small even for $k = 3$. The convergence of the sequence $\hat{f}_k$, however, does not guarantee that this sequence

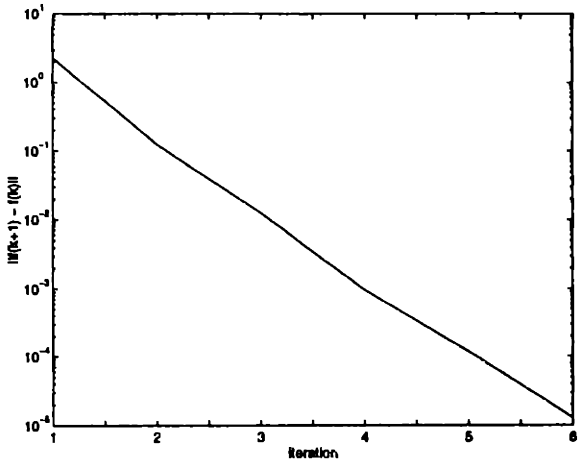


Figure 5.18. The convergence of the Gauss-Newton iteration measured in terms of $\|\hat{f}_{k+1} - \hat{f}_k\|$ for $k = 1, 2, \ldots, 6$.

converges to the global maximum of the posterior density [69], nor does it guarantee that $\hat{f}_\infty$ is actually an improvement over $\hat{f}_1$. For this particular example, we have

$$\begin{aligned} \|\hat{f}_6 - f\| &= 6.43 \\ \|\hat{f}_1 - f\| &= 6.44 \\ \|\hat{f}_{\text{kriged}} - f\| &= 6.78 \\ \|f\| &= 17.0 \end{aligned}$$

where $\hat{f}_{\text{kriged}}$ is the LLSE estimate[4] of conductivity based on only conductivity measurements. The kriged estimate is computed to show that the head measurements improve the log-conductivity estimate, at least in terms of the ℓ_2 norm. Also, the Gauss-Newton iteration converges to an estimate of log-conductivity that is slightly better than $\hat{f}_1$.

These results are typical for log-conductivity functions with small variance σ^2. Namely, the sequence $\hat{f}_k$ converges rapidly to an improved estimate of log-conductivity that is slightly more accurate than $\hat{f}_1$. Furthermore, the rate of convergence of the iteration is geometric. The amount that the Gauss-Newton estimate is an improvement over the single iteration estimate (based on the linearization f_0) depends on the value of the head measurement noise v_h. If v_h is large, the head measurements will supply inaccurate information about log-conductivity, so that an improved linearization of the head measurements will not necessarily lead to an improved estimate.

Note that we did not display the convergence of $J(\hat{f}_k)$ as a function of k for two reasons. First, the computation of this quantity is very expensive, as it requires the implicit inversion of the covariance of the entire finest-scale log-conductivity function.

[4]Kriging is a term commonly used in geostatistics to refer to the computation of the LLSE estimate of a random process from samples of that process [23], although kriging sometimes refers only to the case in which measurement noise is zero.

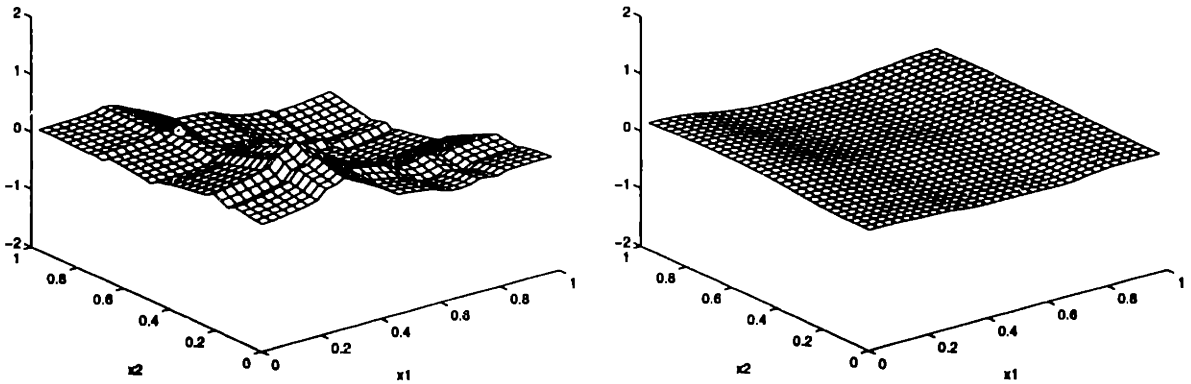


Figure 5.19. (a) The estimate of the log-conductivity function after $K = 6$ iterations, (b) the difference between this estimate and the single iteration estimate plotted in Figure 5.12.

Second, the measurement covariance cannot be computed exactly due to the nonlinearity of the head measurements. Using the linearized measurement variance R_k in place of R can lead to a sequence $J(\hat{f}_k)$ that increases with k.

To see how well the Gauss-Newton iteration performs for log-conductivity functions with large variance, consider the same example but with $\sigma^2 = 10$. A sample path of the log-conductivity function is plotted in Figure 5.20. The sequence $\|\hat{f}_{k+1} - \hat{f}_k\|$ is plotted in Figure 5.21. Note that this convergence rate is much slower than the convergence rate plotted in Figure 5.18. In fact, the general trend is that the sequence $\|\hat{f}_{k+1} - \hat{f}_k\|$ converges more slowly as σ^2 increases, and in some instances will not converge[5] to an asymptotic estimate $\hat{f}_\infty$.

The implication in terms of total computations is that larger variances of log-conductivity will generally require more iterations of the Gauss-Newton optimization. The benefit, however, is the Gauss-Newton iteration generally provides greater improvements over the single iteration estimate $\hat{f}_1$ when σ^2 is large. For this example,

$$\begin{aligned} \|\hat{f}_{14} - f\| &= 29.7 \\ \|\hat{f}_1 - f\| &= 42.3 \\ \|\hat{f}_{\text{kriged}} - f\| &= 66.4 \\ \|f\| &= 76.1 \end{aligned}$$

so that $\hat{f}_{14}$ is a significant improvement over $\hat{f}_1$. Also, compared to the $\sigma^2 = 1$ example, the Gauss-Newton iteration provides a more significant improvement over the single

[5]Some empirical results on the convergence of the Gauss-Newton iteration are provided in [100], while Chavent [12] provides conditions under which the nonlinear optimization has a unique and well-posed minimum.

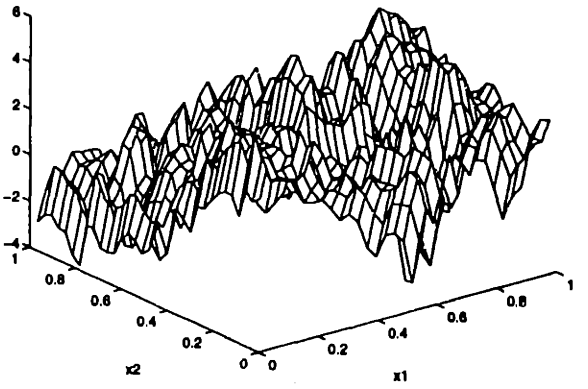


Figure 5.20. The log-conductivity function produced by $\sigma^2 = 10$.

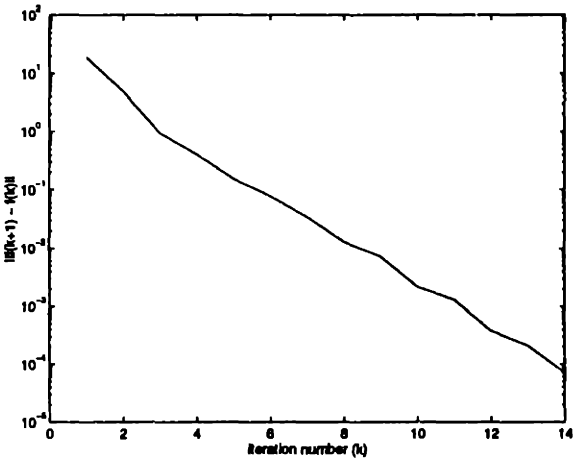


Figure 5.21. The convergence of the Gauss-Newton iteration measured in terms of $\|\hat{f}_{k+1} - \hat{f}_k\|$ for $k = 1, 2, \ldots, 14$ for the example in which $\sigma^2 = 10$.

iteration estimate, even when the differences $\|\hat{f}_k - f\|$ are normalized by $\|f\|$. Finally, note that the estimates $\hat{f}_k$ for this example are significant improvements over the estimate $\hat{f}_{\text{kriged}}$ based on only conductivity measurements. The reason is that for head measurements of fixed quality (σ_h^2 constant), the value of these measurements is greater for higher-variance log-conductivity functions. In other words, the signal-to-noise ratio of the head measurements has increased.

Chapter 6

Travel Time Measurements and Estimation

This chapter analyzes travel times in two-dimensions under steady-state flow conditions, i.e., when the head function and velocity fields do not change with time. The travel time is simply the time it takes a particle to travel from one region of the aquifer to another. Travel times arise as quantities of interest when using tracer tests or analyzing contaminant plumes. If the effects of particle displacement due to diffusion and kinematic dispersion are ignored, travel times are completely determined by the velocity field of the aquifer. Because the velocity fields are determined by hydraulic conductivity (and other hydrologic parameters like porosity and recharge rates), tracer tests can be used to infer the hydraulic conductivity function. For other applications, such as EPA performance analyses, the problem is to estimate rather than measure travel time. In this case, uncertainties in hydraulic conductivity must be propagated to uncertainties in particle travel times.

In a manner similar to the incorporation of head measurements described in Chapters 3 and 5, travel-time measurements can be used to condition hydraulic conductivity. Travel times due to advective transport (no diffusion) are functions of groundwater velocities, which follow from Darcy's Law as

$$v(x) = -\frac{1}{n(x)} K(x) \nabla h(x), \tag{6.1}$$

where $n(x)$ is the effective porosity function. (Effective porosity describes the volumetric percentage of the rock in which fluid is able to flow.) Porosity usually has much less variability than hydraulic conductivity, and is assumed constant or slowly varying in most practical applications [27, 31, 85]. Since head is a function of conductivity, it is clear from Eq. (6.1) that velocity is also a function of conductivity, and in fact is completely determined by conductivity if the boundary conditions, recharge rate, and porosity are known. Thus travel times can serve as measurements of the conductivity function. For example, in [47] tracer tests were incorporated along with cross-well seismograms to estimate the coarse-scale distribution of hydraulic conductivity. Acoustic waves are useful for determining lithographic boundaries, but must be coupled with other measurements in order to uniquely determine the coarse-scale hydraulic conductivity of the individ-

ual regions. In [37], measurements of hydraulic head and contaminant concentrations were used to estimate parameters of the log-conductivity distribution. The utility of incorporating travel-time measurements for the estimation of hydraulic conductivity is also suggested in [31]: "Tracer data are known to be a good source of information on aquifers' heterogeneity: data such as concentration and travel times provide information [that] is vital for identifying large features of the conductivity field that act as flow paths and barriers." Thus, not only might travel-time measurements reduce the uncertainty in the equations of groundwater flow, but they also have the potential to supply information about the conductivity field that is qualitatively different from the information supplied by head and conductivity measurements.

Another application of travel-time analysis is for EPA performance analyses [85, 98], which require a probabilistic distribution for the travel times associated with groundwater flow equations. For instance, the Waste Isolation Pilot Plant is a region in the subsurface of New Mexico currently being evaluated for its suitability as a repository for transuranic wastes generated by the DOE [98, 99]. The problem imposed on the DOE by the EPA is to determine a probability distribution for the time that a leaked particle would take to travel outside the Pilot Plant region when the only available data are hydraulic head and conductivity measurements.

In this chapter, we present a unified approach to the estimation of travel time and the incorporation of travel-time measurements. Analogous to the linearization of head measurements given in Chapter 3 and Appendix C, a linearization of travel times with respect to the log-conductivity function is described in Section 6.1. This linearization is then built into a multiscale model for hydraulic conductivity using the state augmentation algorithm of Chapter 4. If head measurements are also represented by the multiscale model, then, as shown in Section 6.2, the multiscale framework can be used to fuse measurements of conductivity, head, and travel times into an estimate of hydraulic conductivity. Secondly, the exact same model can also be used to determine the conditional distribution of travel time from measurements of head and conductivity. These conditional distributions are presented in Section 6.3, where they are compared to the distributions generated by conditional simulations—the Monte Carlo approach.

■ 6.1 A Linearized Relationship between Travel Time and Log-Conductivity

Travel times due to advective flow were discussed in Section 3.1. Define $x(t)$ to be the location of a particle of interest at time t, and assume that the location of the particle at $t = 0$ is known with certainty. This particle will flow along the streamline that passes through $x(0)$. If C is the streamline that passes through $x(0)$, the time for the particle to travel from $x(0)$ to the end of this streamline is

$$t = \int_C \frac{v(x) \cdot dx}{\|v\|^2}, \tag{6.2}$$

where the differential dx is always in the direction tangent to the streamline C. This equation is identical to Eq. (3.7).

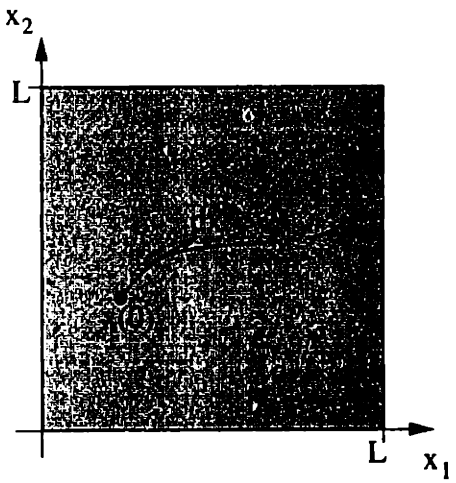


Figure 6.1. The travel of a particle along a streamline originating at $x(0)$ to the line $\mathbf{x}_1 = L$.

In this chapter, we do not consider the travel time between $x(0)$ and another point, but instead analyze the travel time from $x(0)$ to a control plane $\mathbf{x}_1 = L$, as illustrated in Figure 6.1. (Recall that boldface is used to distinguish the two spatial components $\mathbf{x}_1$ and $\mathbf{x}_2$ from the measurement locations x_i.) Define t_{cp} to be the travel time to the control plane. As noted in [85], this travel time is a natural description for any application in which one is concerned with the total mass of solute discharging through a plane. Implicit in this description of travel time is the assumption that water generally flows in the $\mathbf{x}_1$-direction, so that all possible streamlines originating from $x(0)$ pass through $\mathbf{x}_1 = L$.

The goal of this section is to develop a linearized relationship between travel time to a control plane and log-conductivity. This linear relationship will be in the form

$$t_{cp} \approx t_{cp}(f_0) + \int_\Omega g_t(x \mid f_0)\,(f(x) - f_0(x))\,dx\,, \tag{6.3}$$

where f_0 is the log-conductivity function used as the point of linearization and $t_{cp}(f_0)$ is the time to travel from $x(0)$ to the control plane when log-conductivity is equal to f_0. Our linearized analysis of travel time follows the linearized analysis provided in [85], only we do not make the assumption that ∇h_0 is a constant. (If ∇h_0 is equal to a constant, then the travel time $t_{cp}(f_0)$ is for travel in a straight line.) Also, the analysis of [85] is solely in terms of second-order moments, whereas we will develop an explicit functional relationship between t_{cp} and log-conductivity.

The first step for linearizing travel time is to develop a linearized relationship between velocity and log-conductivity. Decompose the head and log-conductivity functions as

$$f(x) = f_0(x) + \delta f(x)\,, \tag{6.4a}$$

$$h(x) = h_0(x) + \delta h(x) + \text{h.o.t}\,, \tag{6.4b}$$

where h_0 is the head function when the log-conductivity is f_0, and h is the head function when log-conductivity is f. Both h and h_0 satisfy the same boundary conditions. In Equation (6.4b), δh is the first-order perturbation of head, which is a linear function of δf. The exact relationship between δh and δf was described in Section 3.3. Substituting Equation (6.4) into Eq. (6.1) and ignoring the higher-order terms in Eq. (6.4b) yields

$$v \approx -\frac{1}{n} e^{f_0+\delta f} \nabla(h_0 + \delta h), \tag{6.5a}$$

$$= -\frac{1}{n} e^{f_0+\delta f} \nabla h_0 - \frac{1}{n} e^{f_0+\delta f} \nabla \delta h, \tag{6.5b}$$

$$\approx -\frac{1}{n} e^{f_0}(1+\delta f) \nabla h_0 - \frac{1}{n} e^{f_0}(1+\delta f) \nabla \delta h, \tag{6.5c}$$

$$\approx -\frac{1}{n} e^{f_0} \nabla h_0 - \frac{1}{n} e^{f_0} \nabla h_0 \, \delta f - \frac{1}{n} e^{f_0} \nabla \delta h, \tag{6.5d}$$

where the last two approximations are made by discarding terms that are not first-order in δf and δh. Thus, velocity can be approximated as

$$v(x) \approx U(x) + u(x), \tag{6.6a}$$

$$U(x) = -\frac{1}{n} e^{f_0} \nabla h_0, \tag{6.6b}$$

$$u(x) = -\frac{1}{n} e^{f_0} \left(\nabla h_0 \, \delta f + \nabla \delta h\right), \tag{6.6c}$$

where $U(x)$ is the "background velocity" due to the conductivity function f_0 and $u(x)$ is the perturbation in velocity due to δf. Since δh is linear in δf, the approximation to the velocity perturbation is linear in δf. In this case, Equation (6.6) describes a linearized relationship between log-conductivity and velocity.

To linearize travel time, consider the following equation

$$x(t) = x(0) + \int_{t'=0}^{t} v\big(x(t')\big)\, dt'. \tag{6.7}$$

This is not necessarily a useful definition for particle location, since $x(t)$ appears on both the left and right-hand sides of the equation. However, the path $x(t)$ can be approximated by the path determined by the background velocity $U(x)$. Define C_0 to be the streamline originating from $x(0)$ and terminating at $\mathbf{x}_1 = L$ according to the velocity function $U(x)$, so that $C_0(t)$ is the position of the particle at time t according to the background velocity. The location of the particle can be approximated by substituting $C_0(t)$ for $x(t)$ into the right-hand side of Eq. (6.7). This substitution gives

$$x(t) \approx x(0) + \int_{t'=0}^{t} v\big(C_0(t')\big)\, dt'. \tag{6.8}$$

Since $\mathbf{x}_1(t_{\text{cp}}) = L$, the first component of this equation is

$$L - \mathbf{x}_1(0) \approx \int_{t'=0}^{t_{\text{cp}}} v_1\big(C_0(t')\big)\, dt', \tag{6.9a}$$

$$\approx \int_{t'=0}^{t_{\text{cp}}} U_1\big(C_0(t')\big)\, dt' + \int_{t'=0}^{t_{\text{cp}}} u_1\big(C_0(t')\big)\, dt', \tag{6.9b}$$

where $v_1(x)$, $U_1(x)$, and $u_1(x)$ are velocities in the $\mathbf{x}_1$-direction. Note that $t_{\text{cp}}(f_0)$ is the travel time to the control plane when the velocity field is $U(x)$, i.e., $\big[C_0(t_{\text{cp}}(f_0))\big]_1 = L$. Define $\delta t_{\text{cp}} = t_{\text{cp}} - t_{\text{cp}}(f_0)$. Substituting $t_{\text{cp}} = t_{\text{cp}}(f_0) + \delta t_{\text{cp}}$ into Equation (6.9b) gives

$$L - \mathbf{x}_1(0) = \underbrace{\int_{t'=0}^{t_{\text{cp}}(f_0)} U_1\big(C_0(t')\big)\, dt'}_{L - \mathbf{x}_1(0)} + \int_{t'=t_{\text{cp}}(f_0)}^{t_{\text{cp}}(f_0)+\delta t_{\text{cp}}} U_1\big(C_0(t')\big)\, dt'$$

$$+ \int_{t'=0}^{t_{\text{cp}}(f_0)} u_1\big(C_0(t')\big)\, dt' + \int_{t'=t_{\text{cp}}(f_0)}^{t_{\text{cp}}(f_0)+\delta t_{\text{cp}}} u_1\big(C_0(t')\big)\, dt' \tag{6.10}$$

The last term on the right-hand side of Eq. (6.10) is certainly second-order in δt_{cp} and δf, and thus is discarded to yield

$$\int_{t'=t_{\text{cp}}(f_0)}^{t_{\text{cp}}(f_0)+\delta t_{\text{cp}}} U_1\big(C_0(t')\big)\, dt' = -\int_{t'=0}^{t_{\text{cp}}(f_0)} u_1\big(C_0(t')\big)\, dt'. \tag{6.11}$$

The right-hand side of Eq. (6.11) is linear in δf, but the left-hand side is linear in δt_{cp} only if $U_1(x)$ is constant in the neighborhood of $C_0(t_{\text{cp}}(f_0))$. For our linearization, we will define an aggregate velocity $\overline{U}_1$ that describes $U_1(x)$ in this neighborhood, so that the left-hand side of Eq. (6.11) is approximated by $\overline{U}_1 \delta t_{\text{cp}}$. The linear relationship between δt_{cp} and δf follows as

$$\delta t_{\text{cp}} = -\frac{1}{\overline{U}_1} \int_{t'=0}^{t_{\text{cp}}(f_0)} u_1\big(C_0(t')\big)\, dt' \tag{6.12a}$$

$$u_1(x) = -\frac{1}{n} e^{f_0(x)} \left(\delta f(x) \frac{\partial h_0(x)}{\partial \mathbf{x}_1} + \frac{\partial}{\partial \mathbf{x}_1} \delta h(x) \right). \tag{6.12b}$$

Recall that $\delta h(x) = \langle g_h(x, x' \,|\, f_0), \delta f(x') \rangle$, where $g_h(x, x' \,|\, f_0)$ is the Fréchét derivative defined in Section 3.3.

To numerically compute the kernel $g_t(x \,|\, f_0)$ in Eq. (6.3), we employ the following algorithm:

(a) integrate Eq. (6.12a) using trapezoidal integration, and

(b) for each velocity sample $u_1(x_i)$ used in the integration, compute the contribution from $\frac{\partial}{\partial \mathbf{x}_1} \delta h(x_i)$ using the Fréchét derivatives from Chapter 3.

The complexity of this computation increases linearly with the number of time samples used for the integration. For all of the examples described in this chapter, one can obtain a very good approximation of $g_t(x \,|\, f_0)$ using a small number, e.g., sixteen, of time samples.

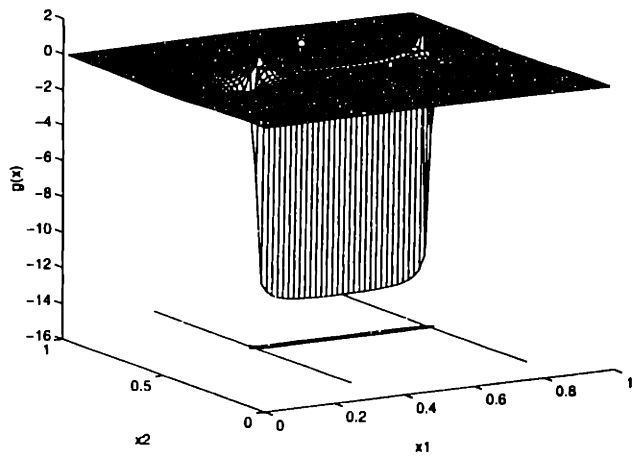


Figure 6.2. The kernel $g_t(x \mid f_0)$ that provides a linearized relationship between travel time and log-conductivity for $f_0 = 0$. The contour lines of the function are projected onto the plane $z = -16$.

■ 6.1.1 Example Linearizations

One advantage of our approach to travel-time analysis is that the kernels $g_t(x \mid f_0)$ are computed explicitly, which allows one to determine the influence of travel-time measurements on hydraulic conductivity estimates. As an example, consider the flow conditions illustrated in Figure 5.1, which lead to flow in the $\mathbf{x}_1$-direction. Consider the travel time from $x(0) = (0.25, 0.5)$ to the plane $\mathbf{x}_1 = 0.75$. For this and the following examples, assume that the effective porosity is constant and equal to $n = 0.2$. For $f_0 = 0$, the kernel $g_t(x \mid f_0)$ is plotted[1] in Figure 6.2. This function shows that the travel-time perturbation $\langle g_t, f - f_0 \rangle$ is most sensitive to log-conductivity values on the line from $x(0)$ to $(0.75, 0.5)$. This line is the path $\mathcal{C}_0$, starting at $x(0)$ and terminating at $\mathbf{x}_1 = 0.75$, determined by the background velocity. Therefore, if this linear approximation is applied to travel-time measurements, the linearized measurement equations will supply accurate information only when $\mathcal{C}$, the true travel path, is close to $\mathcal{C}_0$. Also note that $g_t(x \mid f_0)$ is negative along the path $\mathcal{C}_0$, as should be expected, since an increase in log-conductivity should lead to decreased travel times. The positive values of $g_t(x \mid f_0)$, located just off the path $\mathcal{C}_0$, are due to the second term in Eq. (6.12b).

For a second example, consider the linearization about the log-conductivity function $f_0 = \sin(2\pi\mathbf{x}_1)\sin(2\pi\mathbf{x}_2)$. The conductivity function $K_0 = e^{f_0}$ is plotted in Figure 6.3a. The background velocity field and flow path $\mathcal{C}_0$ are illustrated in Figure 6.3b. This flow

[1]Note from the linearization in Eq. (6.12) that $g_t(x \mid f_0)$ will have impulses along the path $\mathcal{C}_0$ due to the first term in Eq. (6.12)b. However, because we work with a discrete representation of conductivity, where each element of the discrete-index conductivity field corresponds to an aggregate value of $f(x)$ over a small region of area $\Delta\mathbf{x}_1$-by-$\Delta\mathbf{x}_2$, the kernel that is calculated is a convolution of $g_t(x \mid f_0)$ with pulses of area $\Delta\mathbf{x}_1$-by-$\Delta\mathbf{x}_2$. The locations of these pulses correspond to the locations of the local areas represented by each element of the discrete-index conductivity function.

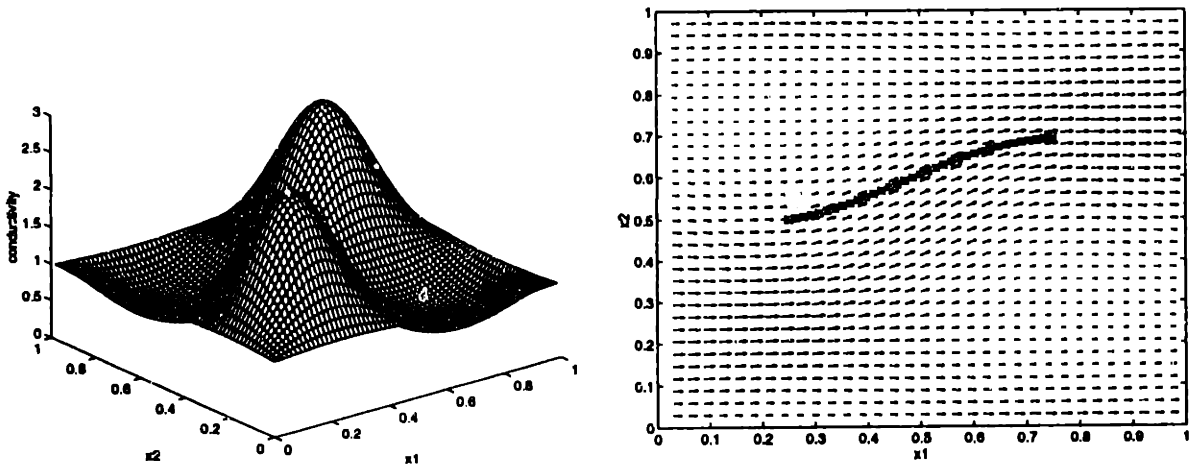


Figure 6.3. (a) The conductivity function $K_0 = \exp(\sin(2\pi x_1)\sin(2\pi x_2))$ and (b) the corresponding background velocity field. The streamline that begins at $x(0) = (0.25, 0.5)$ and terminates on x_1 is also illustrated with *'s.

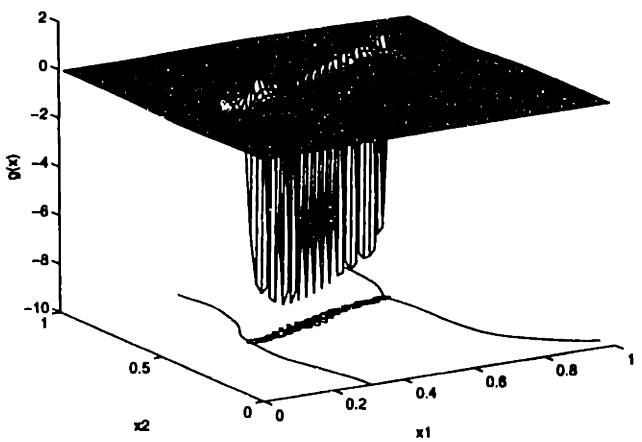


Figure 6.4. The kernel $g_t(x \mid f_0)$ that provides a linearized relationship between travel time and log-conductivity for $f_0 = \sin(2\pi x_1)\sin(2\pi x_2)$. The contour lines of the function are projected onto the plane $z = -10$.

path terminates on the control plane $x_1 = 0.75$ at $x = (0.75, 0.69)$. The kernel $g_t(x \mid f_0)$ is plotted in Figure 6.4. Note again that the travel-time perturbation $\langle g_t, f - f_0 \rangle$ is most sensitive to log-conductivity values on the path C_0. (The lack of a distinct ridge in $g_t(x \mid f_0)$ along the path C_0 is an artifact of the MATLAB function used to plot $g_t(x \mid f_0)$ from discrete samples, and is not a feature of the function itself.)

σ^2	ρ	
	Head BC	Flux BC
0.01	0.98	0.94
0.1	0.97	0.91
0.5	0.89	0.85
1.0	0.82	0.77
5.0	0.39	0.48
10.0	0.19	0.32

Table 6.1. Estimates of the cross-correlation between δt_{cp} and Δt_{cp} based on 4000 sample paths of log-conductivity. The boundary conditions are either flux or head conditions at $x_1 = 0$, $h = 0$ at $x_1 = 1$, and no flux conditions elsewhere. The linearization is based on $f_0 = 0$.

■ 6.1.2 The Accuracy of the Linearization

To characterize the accuracy of the travel-time linearization, we calculate the cross-correlation between $\Delta t_{cp} = t_{cp} - t_{cp}(f_0)$ the approximation $\delta t_{cp} = \langle g_t, f - f_0 \rangle$ when $f_0 = 0$. The cross-correlation, defined by

$$\rho = \frac{E\Big[\big(\Delta t_{cp} - E[\Delta t_{cp}]\big)\big(\delta t_{cp} - E[\delta t_{cp}]\big)\Big]}{\mathrm{var}[\Delta t_{cp}]^{1/2}\mathrm{var}[\delta t_{cp}]^{1/2}},$$

is a useful measure of how well δt_{cp} approximates Δt_{cp}, assuming that the variance of these two random variables is identical.

Assume that log-conductivity is zero-mean and has the covariance given by Eq. (5.2) with $r_1 = r_2 = 3/5$. For the flow conditions, assume the boundary conditions provided in Figure 5.1. The first column of Table 6.1 provides the variance of the log-conductivity function. Note that the variance of K is related to the variance of f by

$$\sigma_K^2 = e^{\sigma^2}(e^{\sigma^2} - 1), \tag{6.13}$$

assuming that f has zero mean. Thus an increase in σ^2 from 0.1 to 5.0 corresponds to the much larger increase in conductivity variance from $\sigma_K^2 = 0.12$ to 2.2×10^4. The second column of Table 6.1 provides estimates of ρ based on 4000 sample paths of f. As would be expected, the cross-correlation decreases with increasing log-conductivity variance. For a second example, replace the head boundary condition at $x_1 = 0$ with the flux condition $q = [1.0,\ 0.0]^T$. Note that both flow scenarios yield identical head functions and velocity fields when the log-conductivity is equal zero. Again, the cross-correlation decreases with increasing log-conductivity variance. For both scenarios, the linearization becomes quite poor for $\sigma^2 \geq 5.0$, which is quite similar to results for the accuracy of the head function linearization cited in [38]. However, using measurements of head and conductivity to condition log-conductivity and improve the linearization allows the time-travel linearization to be applied to measurements of travel time even when $\sigma^2 \geq 5.0$.

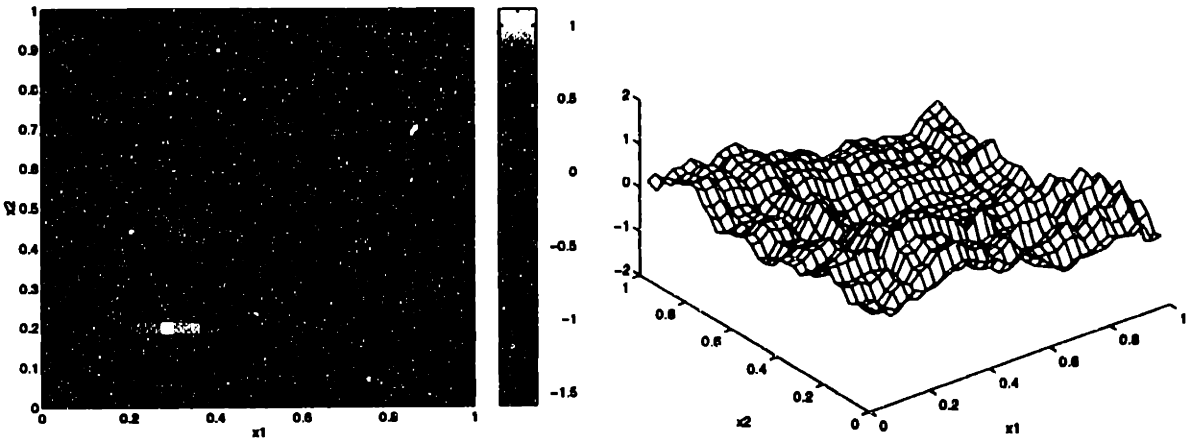


Figure 6.5. A sample path of the log-conductivity function, plotted in (a) gray scale and (b) using a mesh plot.

■ 6.2 Estimation of Hydraulic Conductivity

Now that we have described how to approximate travel-time in the form of Eq. (6.3), travel-time measurements can be incorporated within the multiscale (or any other LLSE) framework. The purpose of this section is to analyze the effect of travel-time measurements on hydraulic conductivity estimation, and also to demonstrate that the multiscale framework can be used to fuse three very different types of measurements into an optimal estimate of hydraulic conductivity. However, we do not address some of the significant multiscale modeling issues, such as developing low-order approximate multiscale implementations of this data fusion problem. Such issues are deferred to future work.

We now build on the example of Section 5.2.1 that was later revisited in Section 5.3.1. The true path is illustrated by the dashed line in Figure 6.6, which is a replica of Figure 5.10. The locations of the head and conductivity measurements were illustrated in Figure 5.9. For the travel-time measurement, assume that a particle is released from $x(0) = (0.25, 0.5)$, and the time to travel to the control plane $\mathbf{x}_1 = 0.75$ is measured. Assume that the measurement noise has very small variance ($\sigma_t^2 = 10^{-8}$) and assume $n = 0.2$ for the effective porosity. The path traveled by the particle is illustrated in Figure 6.6 along with the locations of the head and conductivity measurements. The path traveled by the particle when $f_0 = 0$ is also illustrated, and is the straight line from $x(0) = (0.25, 0.5)$ to $x = (0.75, 0.5)$. The true travel time is $t_{cp} = 0.1493$ units, while the background travel time is $t_{cp}(f_0) = 0.1$ units. This indicates that there must be a region of negative log-conductivity on the path from $x(0)$ to the control plane.

The linearized travel-time measurement is represented at the root node of the multiscale model described in Section 5.2.1, and leads to an increase of three in the dimension of the state at the root node. The finest-scale estimate of log-conductivity is illustrated

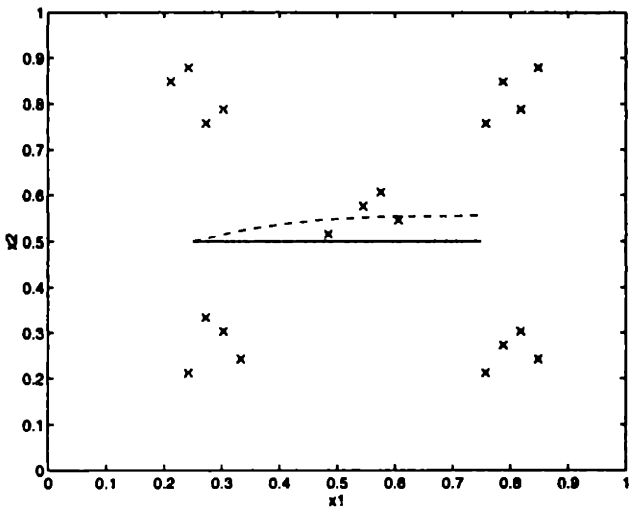
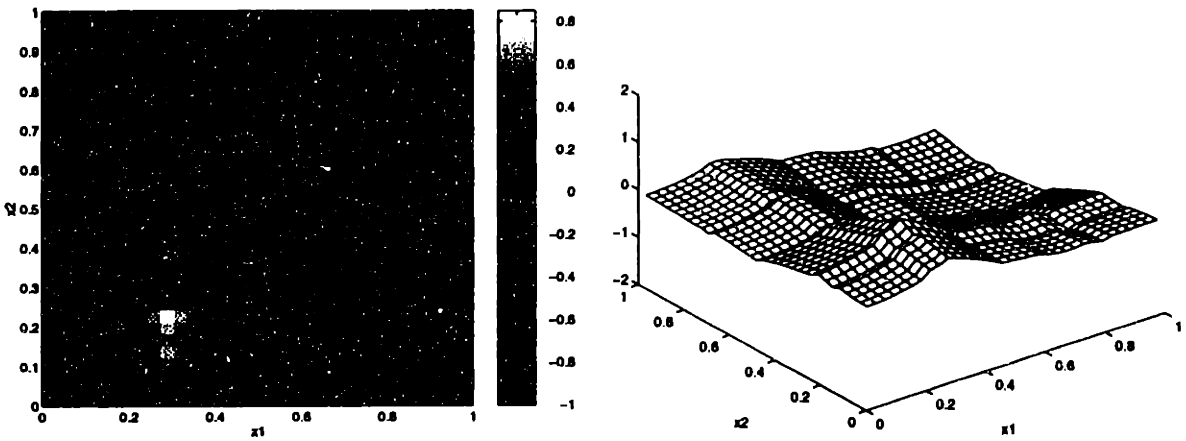


Figure 6.6. Each × corresponds to location at which both the head and conductivity function are sampled. The dashed line corresponds to the particle travel path, C, while the solid line is the path C_0 traveled according to the background velocity.



(a) (b)

Figure 6.7. The LLSE estimate of the log-conductivity function in Fig. 6.5: (a) gray scale image, (b) mesh plot.

in Figure 6.7, and the corresponding error variances are illustrated in Figure 6.8. The difference between the estimate in Figure 5.12 and the estimate in Figure 6.7 is illustrated in Figure 6.9. As would be expected from the kernel in Figure 6.2, the difference between the two estimates is greatest along C_0, the straight line from $x(0) = (0.25, 0.5)$ to $x = (0.75, 0.5)$. Also, the effect of the travel-time measurement is to decrease the log-conductivity along this path to account for the increased travel time.

Extending the results of Section 5.3.1, the Gauss-Newton iteration can be used to approximate the MAP estimate of log-conductivity from the measurements of conduc-

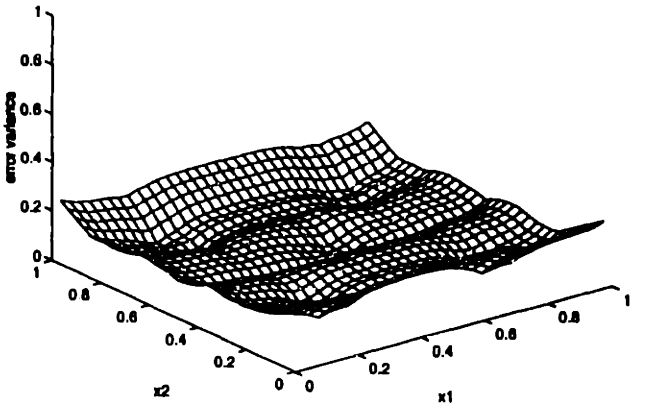


Figure 6.8. The variance of the estimation errors associated with the log-conductivity estimate in Figure 6.7.

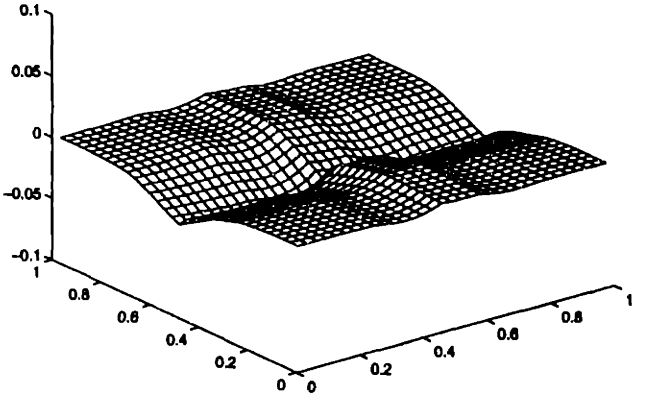


Figure 6.9. The effect of the travel-time measurement, illustrated as the difference between the log-conductivity estimate in Figure 6.7 and the estimate in Figure 5.12, which is based only on head and conductivity measurements.

tivity, head, and travel time. Again the problem is to assess the trade-off between the improvements provided by the MAP estimates (over the linearized LLSE estimate with $f_0 = 0$) and the increase in computations. The Gauss-Newton iteration is implemented using the multiscale framework and the linearization discussed in the previous section. The estimate $\hat{f}_1$ produced by the first iteration of this algorithm is identical to the estimate shown in Figure 6.7. Also, as in the example in Section 5.3.1, the Gauss-Newton iteration converges very quickly to the asymptotic value. Also, this asymptotic value appears to be the global maximum of the conditional density for log-conductivity[2], i.e., the MAP estimate. After six iterations, the errors in the finest-scale log-conductivity estimates are

$$\|\hat{f}_6 - f\| = 6.20$$
$$\|\hat{f}_1 - f\| = 6.26$$
$$\|f\| = 17.0$$

This minor improvement is consistent with the results of Section 5.3, where it was shown that the improvement provided by the Gauss-Newton estimate is generally small for $\sigma^2 < 1$, yet can be significant for large log-conductivity variances.

It is worthwhile to consider the path traveled when log-conductivity is equal to the estimate produced by the Gauss-Newton iteration. The true travel path and that for $f = \hat{f}_6$ are plotted in Figure 6.10. While the two paths are similar, recall from the previous section that the travel-time linearization is highly localized around values of log-conductivity on the path used as the point of linearization. Therefore, large differences between the path used as the point of linearization and the true path can diminish the contribution of the travel-time measurements to the conductivity estimate. Such large deviations will occur when the log-conductivity variance is large, say for $\sigma^2 \geq 5.0$. For these problems, the difficulty in locating the true path is due primarily to the nature of the travel-time measurements. Measurements of t_{cp} provide no information about the value of x_2 at which the particle crosses the control plane. However, knowing the location along x_2 in addition to the travel time would significantly constrain the number of paths leading to the measured travel time. Whether or not such information is available is beyond the scope of this thesis, but the linear approach provided in the previous section certainly can be extended to this class of travel-time measurements.

■ 6.3 Conditional Travel-Time Analysis

The previous section demonstrated that the multiscale framework can be used to estimate hydraulic conductivity from measurements of head, conductivity, and travel

[2]We tested for the global maximum using a large number of simulations and intelligent guesses for the initial log-conductivity estimate. Also, the Gauss-Newton iteration appears to converge in general to the global (MAP) estimate when $\sigma^2 \leq 0.5$. How the convergence is affected by increasing σ^2 (and changing the measurement geometry, measurement noise, conductivity correlation length, etc.) is beyond the scope of this thesis.

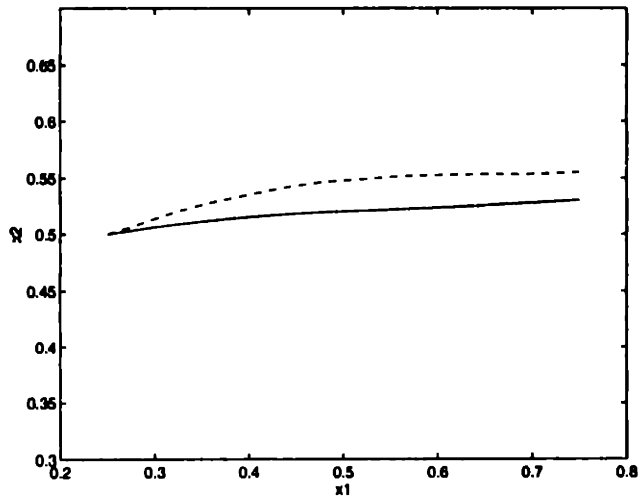


Figure 6.10. The dotted line corresponds to the true travel path, C, while the solid line is the path traveled according to the log-conductivity function produced by the Gauss-Newton algorithm.

time. The linearized travel-time measurement was represented at the root node of a multiscale model for hydraulic conductivity. For some applications, such as those described in [31, 85, 98], travel time measurements are not available, and the problem is to determine a distribution for travel time conditioned on measurements of head and conductivity. Because the multiscale estimator produces estimates and estimation error variances at every scale of the multiscale process, the multiscale framework can also be used to characterize conditional distributions of travel time. For example, consider the conductivity estimation problems discussed in Section 6.2. The multiscale models described in these examples can also be used to estimate mean and variance of the time to travel from $x(0) = (0.25, 0.5)$ to the control plane $\mathbf{x}_1 = 0.75$, conditioned on the head and conductivity measurements of these examples.

Before analyzing conditional travel-time distributions, it worthwhile to say a few words about unconditional travel-time distributions. Most importantly, travel-time distributions are essentially log-normal when conductivity is log-normal. This observation has been verified by extensive numerical simulation, but can also be argued from the relationship between velocity and log-conductivity, which from Darcy's Law is

$$v = -\frac{1}{n} e^f \nabla h . \tag{6.14}$$

The distance Δx traveled in a small time Δt is

$$\Delta x = \left(-\frac{1}{n} e^f \nabla h \right) \Delta t . \tag{6.15}$$

Taking the magnitude of both sides and solving for Δt yields

$$\Delta t = \frac{\|\Delta x\|}{\|\nabla h\|} n e^{-f} . \tag{6.16}$$

Assume now that the distance travelled, $\|\Delta x\|$, is fixed and we wish to determine the effect of variations in f on travel time. Since variations in $h(x)$ are relatively smooth compared to variations in $f(x)$, we can assume to first-order that ∇h remains relatively constant in comparison to e^f. Therefore, the time step Δt will be log-normal when conductivity is log-normal. Travel times are the sum of such time steps. When the log-conductivity field is highly correlated, a significant portion of each travel path will be in a region of relatively constant log-conductivity, such as a high-conductivity zone. In this case, we would expect the total travel time to be approximately log-normal. As the ratio of the length of the travel path to the correlation length of log-conductivity increases, the time steps that make up the total travel time become more independent. Therefore, travel time should be more normally distributed for large ratios of the length of the travel path to the correlation length of log-conductivity. These intuitive arguments have been supported by all of our numerical simulations.

The correlation length of log-conductivity also affects the variance of travel time. Namely, the variance of the travel-time distribution increases as the correlation length in the direction of flow increases. This relationship also is intuitive, since strong correlations in the direction of flow will increase the probability that the conductivity function has elongated regions of very large or very small conductivity, which will disperse the travel-time distribution.

We now use the multiscale model described in the previous section for the estimation of travel time from measurements of head and conductivity. Assume the same experimental set-up as in Section 6.2, except that we now assume that the travel time measurement is not available and instead we wish to estimate it. The LLSE estimate of $\delta t_{\rm cp}$, call it $\delta\hat{t}_{\rm cp}$, can be determined from the estimate of $z(0)$, since $\delta t_{\rm cp}$ is represented at the root node of the multiscale process. The (approximate) LLSE estimate of $t_{\rm cp}$ is then

$$\hat{t}_{\rm cp} = t_0 + \delta\hat{t}_{\rm cp}\,. \tag{6.17}$$

Note that $\delta\hat{t}_{\rm cp}$ can also be determined from the inner product of the estimate of the finest-scale process—illustrated in Figure 5.12—and the kernel $g_t(x \mid f_0)$. However, to derive the associated estimation error variance from $g_t(x \mid f_0)$ would require the error *covariance* matrix for the entire finest-scale process. Recall that we seek a conditional distribution for travel time, not just an estimate. That the multiscale estimator also produces the error variance for each state variable is the whole motivation in this example for modeling travel time at the root node.

For this example, the following results are obtained from the estimate and error variance of the root node:

$$\hat{t}_{\rm cp} = 0.1460\,, \tag{6.18a}$$

$$E[(t_{\rm cp} - \hat{t}_{\rm cp})^2] = (0.0215)^2\,. \tag{6.18b}$$

Recall that the true travel time is $t_{\rm cp} = 0.1493$ units, and the background travel time is $t_{\rm cp}(f_0) = 0.1$ units. For this particular example, the multiscale estimator does a very

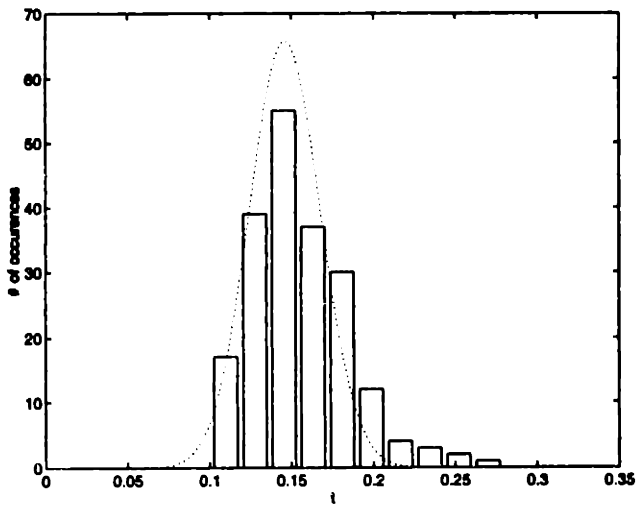


Figure 6.11. The histogram for travel time produced by two-hundred conditional simulations of log-conductivity and as many implementations of the flow equation. The dotted line is the (appropriately normalized) PDF of a normal random variable with mean 0.1460 and standard deviation 0.0215.

good job of estimating travel time, since the true travel time is well within a standard deviation of the estimation error.

A more brute force but widely used method for computing the conditional distribution of travel time, which can be used to verify the results of the multiscale analysis, is the Monte-Carlo approach. The basic idea is to generate a number of realizations of the conditional distribution for log-conductivity, to compute the travel time associated with each conductivity function, and then to generate a histogram representing the conditional distribution of travel time. The advantage of this approach is that it is not based on a linear approximation of travel time. To implement this approach, one must be able to efficiently generate many conditional simulations of log-conductivity and to solve the 2D flow equation for each log-conductivity function. Unlike most approaches to conditional simulation, which require a computationally intensive factorization of the entire estimation error matrix, the multiscale estimator produces a multiscale model for the estimation errors [61]. (See Appendix B for more details.) Once the parameters of the multiscale error model are computed, conditional simulations of log-conductivity can be rapidly generated. The bottleneck in computations, then, is the implementation of the 2D flow equations. For the previous example, two-hundred conditional simulations of log-conductivity lead to the histogram for travel time plotted in Figure 6.11. The mean and median of this histogram are 0.1549 and 0.1514, which are both slightly higher than the true travel time, but the true travel time is well within a standard deviation of the mean of this distribution.

The histogram in Figure 6.11 can also be used to verify the travel-time distribution implied by the multiscale estimator. The normal distribution with mean $\hat{t}_{cp}$ and standard deviation 0.0215 is superimposed on the histogram using dotted lines. As is evident

from the figure, the two distributions are very similar, but the histogram appears to be more log-normal than normal. This log-normality is again due to the log-normality of the conditional distribution for conductivity, the variance of which is illustrated in Figure 5.13. That the conditional distribution implied by the multiscale model is close to the true conditional distribution is because log-normal random variables with small variances are approximately normal. An open question is how to use the linearized travel-time equation to develop accurate conditional distributions for travel time when the variance of the conditional distribution for log-conductivity is large. One potential solution is relinearization, possibly using multiple paths of linearization at each iteration.

Chapter 7

Modeling and Estimation of Fractional Brownian Motion

A wide variety of physical phenomena are described by random processes whose power spectral densities behave as $1/f^{\alpha}$. For example, $1/f$ processes can be used to describe average temperature distributions [45, 57], annual flow rates in rivers [45], the noise in vacuum tubes and electrical components [57], biological time series like heartbeats [96], economic time series like the Dow Jones Industrial Average [96], and traffic in communications networks [94]. Processes with $1/f$-like power spectra are also used to generate images that model real world objects like clouds and mountain ranges [4, 89]. These processes possess two common characteristics, statistical self-similarity and long-range dependence. A popular model that possesses these two characteristics is the class of fractional Brownian motions [64], which are a generalization of Brownian motion. In this chapter, we focus on multiscale representations of fractional Brownian motion (fBm).

Fractional Brownian motions are defined as the zero-mean Gaussian random processes with statistically stationary and self-similar increments. The first models for fBm [64] were fractional integrals of white Gaussian noise, but such nonlinear integrals are not useful for synthesizing or processing (estimating, smoothing, and the like) fBm. An alternative is to approximate fBm with models that lead to efficient synthesis or processing algorithms. Two such approximations are random midpoint displacement and wavelet-based representations. The random midpoint displacement algorithm, which was described in Section 2.3.2 as a method for synthesizing Brownian motion, is a popular tool for approximately synthesizing fBm [4]. Because the the wavelet transform has been shown to approximately whiten fBm [36, 96], wavelets are another useful tool for approximately synthesizing fBm. In addition, Wornell [96] also used the wavelet-based framework for accomplishing signal processing tasks like the estimation of fBm from uniformly sampled and noisy measurements.

Both the midpoint displacement and the wavelet-based approximations lead to synthesis algorithms consisting of a progression from coarse to fine scales, adding successively finer details at each step. Because the algorithms are analogous to the multiscale autoregression, the random midpoint displacement and wavelet-based approximations

are naturally represented by multiscale processes, as shown in [62] and [35], respectively. The advantage of representing fBm within the multiscale framework is the ability to take advantage of the efficient processing and synthesis algorithms. In this chapter, we present a number of multiscale models that approximately represent fBm. We first show that the multiscale models based on midpoint displacement and the wavelet transform can be easily enhanced, without increasing the order[1] of the models, to more accurately represent fBm. Next, we provide higher-order multiscale models that improve the accuracy of the approximation at the expense of increased computational complexity. One higher-order model combines the midpoint displacements and wavelet coefficients into a single multiscale model. Finally, a more general algorithm is provided that takes advantage of the self-similarity and stationary increments properties of fBm to yield multiscale models that approximate the statistics of fBm to any desired accuracy. Extensions to these models are suggested in Chapter 8.

■ 7.1 Fractional Brownian Motion

Fractional Brownian motion (fBm) was first proposed in [64] as a class of random processes with strong interdependence between distant samples. A fractional Brownian motion (fBm) is a Gaussian process with zero mean and covariance[2] [64]

$$E\big[x(t)x(s)\big] = \frac{\sigma^2}{2}\left[|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right], \qquad 0 < H < 1. \tag{7.1}$$

This class of processes is completely characterized by $E\big[x(1)^2\big] = \sigma^2$ and the Hurst exponent H. The covariance and variance functions are plotted in Figure 7.1 for $\sigma = 1$ and three values of H. Fractional Brownian motion is statistically self-similar in the sense that

$$x(at) \stackrel{\mathcal{P}}{=} a^H x(t), \qquad a > 0, \tag{7.2}$$

where $\stackrel{\mathcal{P}}{=}$ denotes equality in (finite-dimensional) distribution. While fBm is nonstationary, its power spectral density is well-defined over any finite bandwidth observation window as [96]

$$S_x(f) = \frac{c}{f^{2H+1}}, \qquad f_1 < f \le f_2, \tag{7.3}$$

for some constant c and any two positive frequencies $0 < f_1 < f_2$. The nonstationarity of the process is evidenced near $f = 0$, where Eq. (7.3) implies infinite power in the low-frequency components of $x(t)$.

[1] An n-th order tree model is one for which all the variables have dimension n or less.

[2] To be consistent with the literature on fBm, the function $x(t)$ with independent variable t is chosen to represent fBm. However, note that in the rest of this thesis, because many of the applications involve estimating random processes that are functions of space, x is chosen as the independent variable.

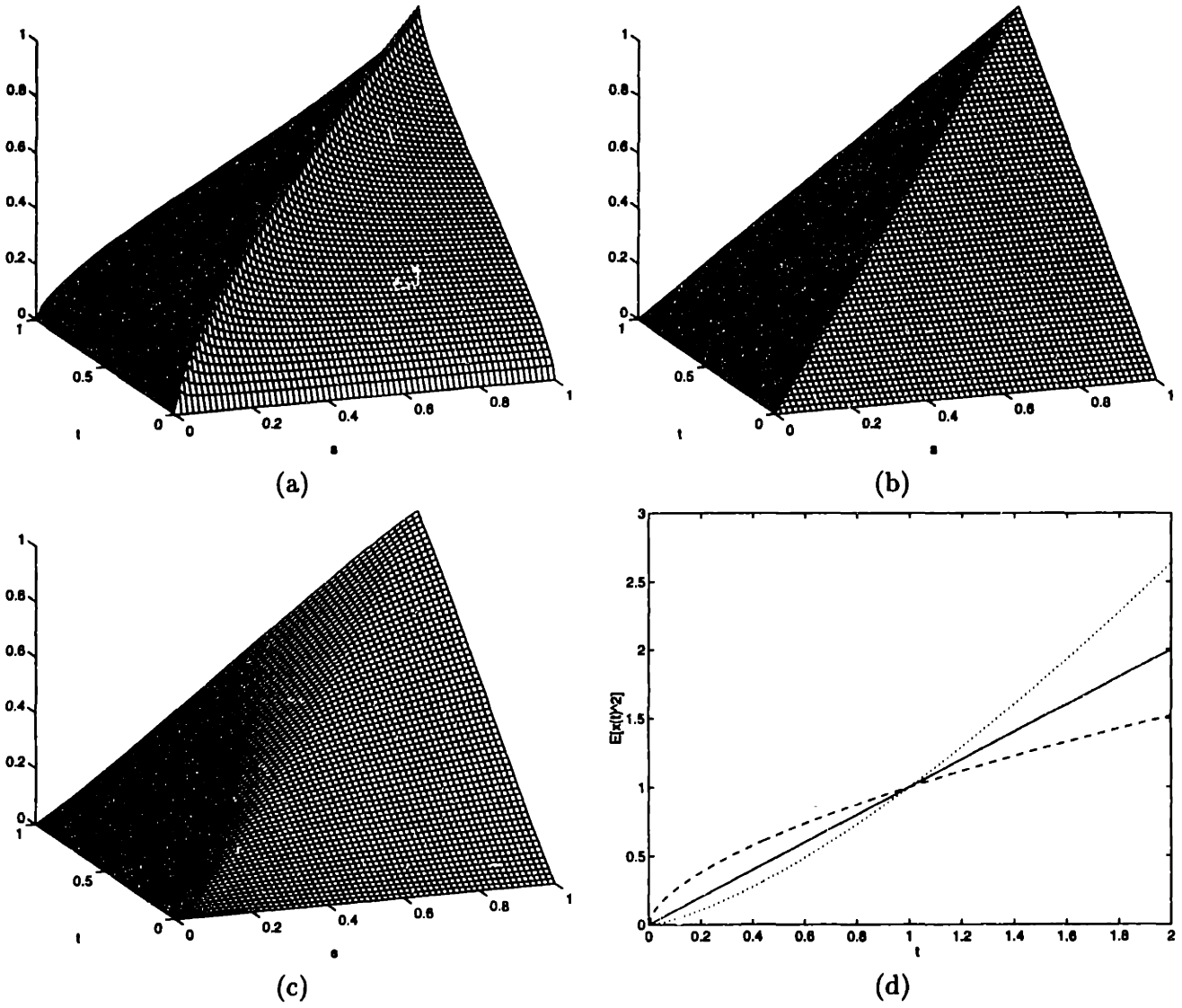


Figure 7.1. The covariance functions of fBm for $\sigma = 1$ and (a) $H = 0.3$, (b) $H = 0.5$, and (c) $H = 0.7$. (d) The variance function of fBm for $\sigma = 1$ and $H = 0.3$ (dashed line), $H = 0.5$ (solid line), and $H = 0.7$ (dotted line).

The nonstationary covariance function in Eq. (7.1) masks a more elegant definition of $x(t)$ in terms of its increments process. A process $x(t)$ is an fBm with Hurst exponent $0 < H < 1$ if and only if

- $x(t)$ is Gaussian and has zero mean,
- $x(t)$ has stationary increments, meaning that

$$E\Big[\big(x(t+s) - x(t)\big)^2\Big] = g(s)\,,$$

  where $g(s)$ is called the *structure function*,
- and $x(t)$ has self-similar increments, i.e., $g(as) = a^{2H}g(s)$.

The self-similarity of the increments implies that the structure function $g(s)$ must have the form $g(s) = \sigma^2|s|^{2H}$. The covariance function in Eq. (7.1) follows from this structure function and the stationary increments property. Fractional Brownian motion reduces to Brownian motion for $H = 0.5$. Sample paths illustrating the self-similarity of fBm are provided in Figure 7.2. Figure 7.2a plots a sample path of $x(t)$ on the interval $t \in [0,1]$ for $H = 0.3$, while Figure 7.2b plots another sample path of $2^{-H}x(t)$ on the interval $t \in [0,2]$ for $H = 0.3$. Note that the plot of Figure 7.2b effectively compresses the interval $[0,2]$ by a factor of two to match the unit interval in Figure 7.2a. The similarity between the two plots is a manifestation of the statistical self-similarity. Figures 7.2c-d are analogous plots for $H = 0.7$. Again, one can see the self-similarity of fBm. Note that fBm for $H < 0.5$ has more fine-scale energy than Brownian motion, while fBm for $H > 0.5$ has less fine-scale variation than Brownian motion.

The stationary and self-similar increments properties are important attributes of fBm. However, fBm does not have independent increments[3], except for the special case when $H = 1/2$. For $H = 1/2$, $x(t)$ is a Markov process. This distinction is important, because not only are the increments of Brownian motion correlated for $H \neq 1/2$, they are strongly correlated over large distances. This strong interdependence is what makes the modeling and analysis of fBm such a difficult problem.

A discrete-time version of fBm can be derived from samples of $x(t)$. If $x[n] \triangleq x(n\Delta t)$, then $x[n]$ is a zero-mean Gaussian process with covariance function [55]

$$E\big[x[n]x[m]\big] = \frac{\sigma^2(\Delta t)^{2H}}{2}\left(|n|^{2H} + |m|^{2H} - |n-m|^{2H}\right), \tag{7.4}$$

and the variance of the increments process is $E\big[(x[n+m] - x[n])^2\big] = \sigma^2(\Delta t)^{2H}|m|^{2H}$. The increments process is stationary and self-similar for compressions or expansions of the time axis by powers of two, i.e., $E\big[(x[n+2^k m] - x[n])^2\big] = (2^k)^{2H}E\big[(x[n+m] - x[n])^2\big]$ for positive integer values of k. Discrete-time fBm can be described in terms of a first-order autoregression

$$x[n] = x[n-1] + w[n]\,, \qquad x[0] = 0\,, \tag{7.5}$$

[3]Kasdin states erroneously, and likely by mistake, that fBm has independent increments [56, p.817].

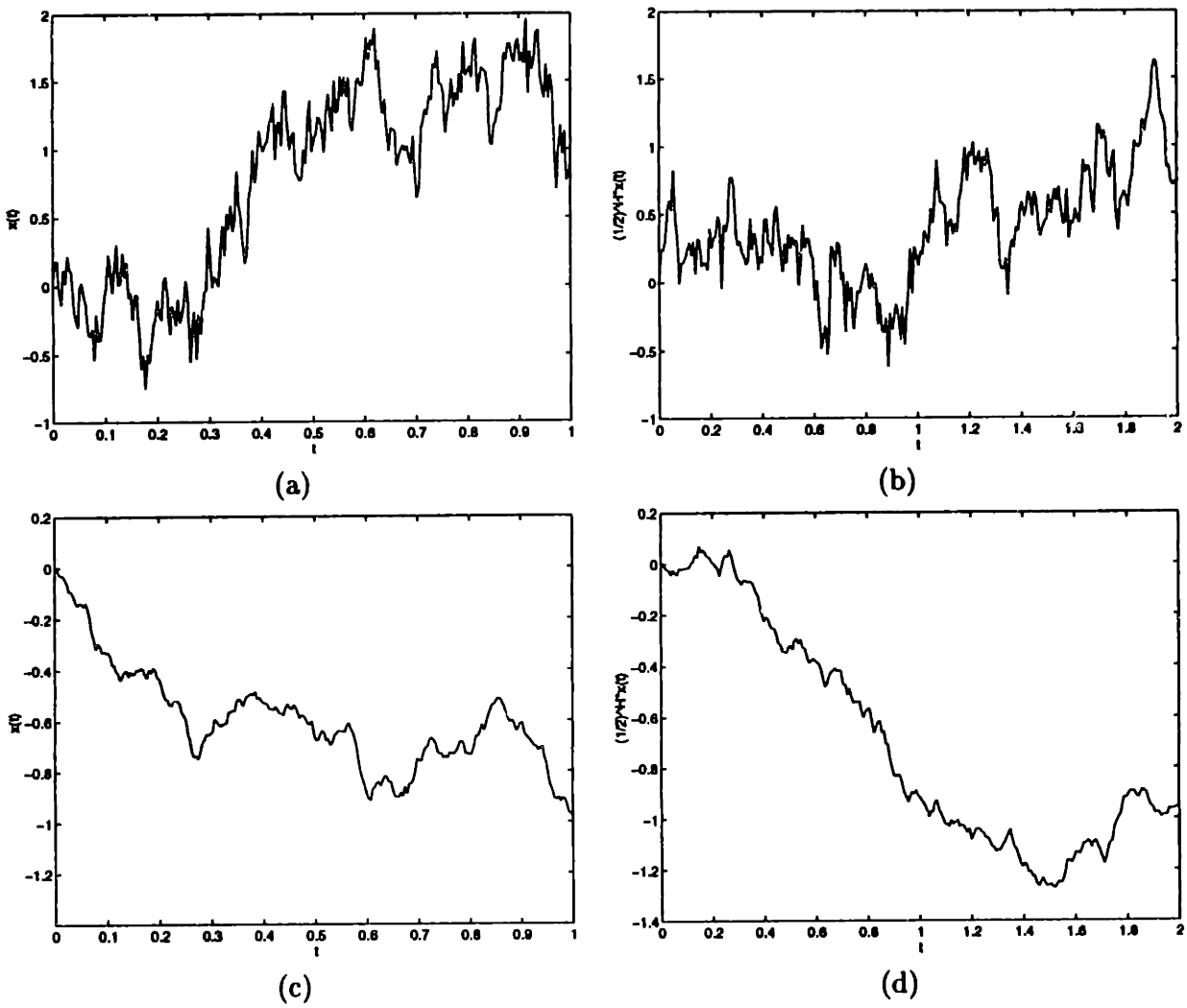


Figure 7.2. (a) A sample path of fBm ($H = 0.3$, $\sigma = 1$) on the interval $t \in [0, 1]$, and (b) A sample path of fBm ($H = 0.3$, $\sigma = 1$) scaled by $2^{-0.3}$ on the interval $t \in [0, 2]$. Plots (c) and (d) are the analogous graphs for $H = 0.7$ and a scaling of $2^{-0.7}$ in (d).

where the first-order increments process $w[n]$, also known as discrete fractional Gaussian noise [55], has covariance

$$\begin{aligned} r[m] &= E\big[w[n+m]w[n]\big] \\ &= \frac{\sigma^2(\Delta t)^{2H}}{2}\left(|m+1|^{2H} + |m-1|^{2H} - 2|m|^{2H}\right). \end{aligned} \tag{7.6}$$

The stationarity of $w[n]$ follows from the stationary increments property of fBm. Since fBm is Markov for $H = 1/2$, $w[n]$ is white for $H = 1/2$. For $H > 1/2$, $w[n]$ is positively correlated with the other increments; for $H < 1/2$, $w[n]$ is negatively correlated with other increments.

While the discrete-time version of fBm is derived directly from samples of fBm, one must be careful in applying any of results from continuous-time to the discrete-time processes, and vice versa. The reason is that sampling fBm leads to significant aliasing for $H < 1/2$, since there is significant energy in the high-frequency components of fBm for $H < 1/2$.

■ 7.1.1 Random Midpoint Displacement for Brownian Motion

As discussed in Section 2.3.2, the random midpoint displacement algorithm synthesizes Brownian motion by generating samples at successively finer sampling intervals [4]. Midpoint displacement follows from the Markovianity of Brownian motion, which holds that, *conditioned* on any two samples $x(t_1)$ and $x(t_2)$ for $t_1 < t_2$, the samples $x(t)$ on the interval $t \in (t_1, t_2)$ are independent of those outside the interval. The implication is that if samples of Brownian motion have been generated for times $t_0, t_1, \ldots, t_N$, then the midpoints of the N intervals partitioned by these samples can be generated independently.

While the midpoint displacement algorithm was discussed in Section 2.3.2, we review it here for Brownian motion in order to understand how it can be applied to fractional Brownian motion. Assume that we want to synthesize a path of Brownian motion on the unit interval $t \in [0, 1]$. The endpoint $x(0)$ is always zero and the endpoint $x(1)$ can be generated from a sample of the distribution $\mathcal{N}(0, \sigma^2)$. The midpoint $x(1/2)$ can be decomposed as

$$x(1/2) = E\big[x(1/2) \,\big|\, x(1), x(0)\big] + \tilde{x}(1/2)\,. \tag{7.7}$$

Because Brownian motion is Gaussian, $E\big[x(1/2) \,\big|\, x(1), x(0)\big]$ is the linear least-squares error (LLSE) estimate of $x(1/2)$ from $x(1)$ and $x(0)$. Also, $\tilde{x}(1/2)$ is the estimation error, which must be independent of $x(1)$ and $x(0)$. Using standard LLSE estimation equations, the elements of Eq. (7.7) follow as

$$E\big[x(1/2) \,\big|\, x(1), x(0)\big] = \frac{1}{2}x(1)\,, \tag{7.8a}$$

$$\mathrm{var}[\tilde{x}(1/2)] = \sigma^2/4\,. \tag{7.8b}$$

The midpoint $x(1/2)$ can be expressed as an interpolation—Eq. (7.8a)—and a displacement $\tilde{x}(1/2)$ that can be generated independently of $x(1)$ and $x(0)$. This interpolation and displacement is illustrated in Figure 2.2a. Note that Equations (7.7) and (7.8) are the same as Eq. (2.34), since $\widehat{E}[x \mid y] = E[x \mid y]$ for jointly Gaussian random variables.

The next step of the displacement algorithm is to generate the midpoints $x(1/4)$ and $x(3/4)$. These midpoints can be decomposed as

$$x(1/4) = E[x(1/4) \mid x(1/2), x(0)] + \tilde{x}(1/4), \tag{7.9a}$$

$$x(3/4) = E[x(3/4) \mid x(1), x(1/2)] + \tilde{x}(3/4). \tag{7.9b}$$

This set of equations is identical to Eq. (2.35). Using both the independent increments and the independence of Bayes' least-squares estimation errors from conditioning information, the displacements $\tilde{x}(1/4)$ and $\tilde{x}(3/4)$ are independent of each other and the samples $x(0)$, $x(1/2)$, and $x(1)$. Again invoking the LLSE estimation formulas, Eq. (7.9) becomes

$$x(1/4) = \frac{1}{2}x(1/2) + \tilde{x}(1/4), \qquad \operatorname{var}[\tilde{x}(1/4)] = \sigma^2/8, \tag{7.10a}$$

$$x(3/4) = \frac{1}{2}\left(x(1) + x(1/2)\right) + \tilde{x}(3/4), \qquad \operatorname{var}[\tilde{x}(3/4)] = \sigma^2/8. \tag{7.10b}$$

To compute the midpoint displacements at finer intervals, the LLSE estimator can be repeatedly applied. However, we can also make use of the statistical self-similarity of Brownian motion. Self-similarity implies that

$$P_{x(2^{-(m+1)})} = \left(\frac{1}{2}\right)^2 P_{x(2^{-m})},$$

$$P_{x(2^{-(m+1)})x(2^{-m})} = \left(\frac{1}{2}\right)^2 P_{x(2^{-m})x(2^{-(m-1)})}.$$

Applying these relationships recursively to Eq. (7.10) yields

$$x(2^{-(m+1)}) = \frac{1}{2}x(2^{-m}) + \tilde{x}(2^{-(m+1)}), \tag{7.11a}$$

$$\operatorname{var}[\tilde{x}(2^{-(m+1)})] = \sigma^2\left(\frac{1}{2}\right)^{m+2}, \tag{7.11b}$$

$$x(3/2^{m+1}) = \frac{1}{2}\left(x(2^{-(m-1)}) + x(2^{-m})\right) + \tilde{x}(3/2^{m+1}), \tag{7.11c}$$

$$\operatorname{var}[\tilde{x}(3/2^{m+1})] = \sigma^2\left(\frac{1}{2}\right)^{m+2}. \tag{7.11d}$$

Furthermore, the independent increments property of Brownian motion means that the interpolation and displacement at any given scale are shift-invariant, i.e.,

$$E[x(2^{-(m+1)} + \Delta t) \mid x(2^{-m} + \Delta t), x(\Delta t)] = \frac{1}{2}\left(x(2^{-m} + \Delta t) + x(\Delta t)\right), \tag{7.12a}$$

$$\operatorname{var}[\tilde{x}(2^{-(m+1)} + \Delta t)] = \sigma^2\left(\frac{1}{2}\right)^{m+2}, \tag{7.12b}$$

for any $\Delta t \geq 0$. In other words, the interpolation at the midpoint is always the average of the endpoints, and the variance of the displacement is constant for a given scale m and decreases geometrically with increasing scale m. The shift-invariance for Brownian motion greatly simplifies the midpoint displacement algorithm, since the interpolation is constant across all scales and shifts, and the variance of the displacement varies geometrically with scale. However, as shown in the next section, the shift-invariance does not apply to fBm for $H \neq 1/2$, since fBm for $H \neq 1/2$ does not have independent increments.

■ 7.1.2 Random Midpoint Displacement for fBm

As shown in [4], random midpoint displacement can be used to generate sample paths of a distribution that is approximately equal to the distribution of fBm. To illustrate this algorithm, and to highlight the approximations made, assume again that we want to synthesize fBm on the unit interval $t \in [0, 1]$. As for Brownian motion, the first step is to set $x(0) = 0$ and then generate $x(1)$ from a sample of the distribution $\mathcal{N}(0, \sigma^2)$. If the midpoint of the unit interval is decomposed as in Eq. (7.7), the LLSE estimator equations provide

$$E\big[x(1/2) \,\big|\, x(1), x(0)\big] = \frac{1}{2}x(1) \quad \text{and} \quad \operatorname{var}\big[\tilde{x}(1/2)\big] = \sigma^2\left[\left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^2\right], \quad (7.13)$$

where $\tilde{x}(1/2)$ is independent of $x(1)$ and $x(0)$. Using the self-similarity of fBm, we can then derive

$$E\big[x(2^{-(m+1)}) \,\big|\, x(2^{-m}), x(0)\big] = \frac{1}{2}x(2^{-m}), \quad (7.14a)$$

$$\operatorname{var}\big[\tilde{x}(2^{-(m+1)})\big] = \sigma^2\left(\frac{1}{2}\right)^{2mH}\left[\left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^2\right]. \quad (7.14b)$$

Equation (7.14a) represents an interpolation of the midpoint of the interval $[0, 2^{-m}]$ from the endpoints $x(0)$ and $x(2^{-m})$, while $\tilde{x}(2^{-(m+1)})$ is the displacement.

The midpoint displacement algorithm for fBm given in [4] is based upon Equation (7.14), and is directly analogous to the midpoint displacement algorithm for Brownian motion. Assume that the samples $x(i\Delta t_{m-1})$, where $\Delta t_{m-1} = (1/2)^m$, are known for $i = 0, \ldots, 2^m$. These samples might have been synthesized at previous iterations of the midpoint displacement algorithm, and they are illustrated by the solid circles in Figure 7.3. The samples at the midpoints of the intervals $[k\Delta t_{m-1}, (k+1)\Delta t_{m-1}]$ are

$$t_k = (2k+1)\,\Delta t_m\,, \qquad k = 0, \ldots, 2^m - 1\,,$$

where $\Delta t_m = \Delta t_{m-1}/2$. These samples are illustrated by the gray circles in Figure 7.3. For the algorithm in [4], each sample $x(t_k)$ is synthesized as an interpolation from the nearest samples $x(t_k - \Delta t_m)$ and $x(t_k + \Delta t_m)$, plus a displacement whose variance is

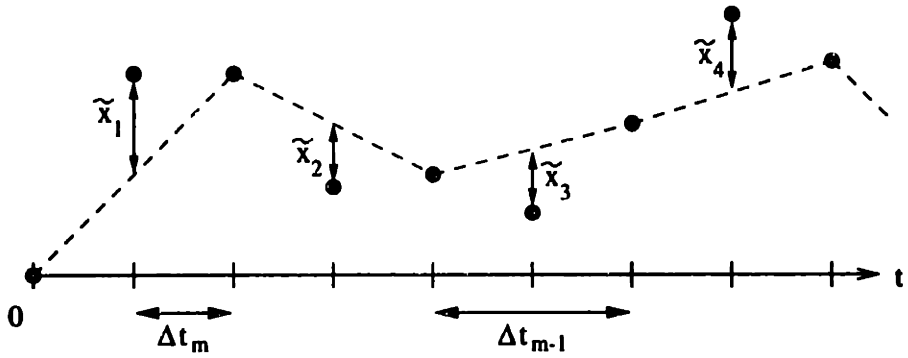


Figure 7.3. The m-th iteration of the midpoint displacement algorithm. The samples generated after the first $(m-1)$ iterations are illustrated by the solid •'s spaced by Δt_{m-1}. The samples generated at the m-th iteration are the gray circles. The linear interpolation is illustrated by the dashed lines, while the variance of the displacements $\tilde{x}_k = \tilde{x}\big((2k+1)\Delta t_m\big)$ is given by Eq. (7.15b).

given by Eq. (7.14b). This interpolation and displacement are

$$x(t_k) = \underbrace{\frac{1}{2}\left(x(t_k + \Delta t_m) + x(t_k - \Delta t_m)\right)}_{\text{interpolation}} + \tilde{x}(t_k), \tag{7.15a}$$

$$\text{var}[\tilde{x}(t_k)] = \sigma^2 \left(\frac{1}{2}\right)^{2mH} \left[\left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^{2}\right]. \tag{7.15b}$$

All of the displacements $\tilde{x}(t_k)$ are assumed to be independent. The interpolation of Eq. (7.15a) is illustrated in Figure 7.3 by the dashed lines, while the newly synthesized samples are represented by the gray circles. The result is that $x(t)$ is generated for $t = i\Delta t_m$, $i = 0, 1, \ldots, 2^{m+1}$. Finer sampling intervals can be achieved by repeating this process.

This algorithm, while simple and efficient, is an exact representation of fBm only for $H = 1/2$. For $H \neq 1/2$, two approximations are made. First, the interpolation and displacement given by Eq. (7.15) is statistically correct only for $H = 1/2$ or $k = 0$. (Note that Eq. (7.15) reduces to Eq. (7.14) for $k = 0$, i.e., when one of the endpoints is $x(0)$.) Unless $k = 0$ or $H = 1/2$,

$$E\left[x(t_k) \,\middle|\, x(t_k + \Delta t_m), x(t_k - \Delta t_m)\right] \neq \frac{1}{2}\left(x(t_k + \Delta t_m) + x(t_k - \Delta t_m)\right), \tag{7.16}$$

meaning that the conditional expectation is not equal to a simple average of the endpoint samples. Furthermore, the variance of the prediction error $\tilde{x}(t_k)$ varies as a function of k, and is equal to the variance in Eq. (7.15b) only for $k = 0$ or $H = 1/2$. The second approximation made by the midpoint displacement algorithm of [4] is that the displacements $\tilde{x}(t_k)$ are not independent. For instance, conditioned on $x(0)$, $x(1/2)$, and $x(1)$, the displacements $\tilde{x}(1/4)$ and $\tilde{x}(3/4)$ are correlated when $H \neq 1/2$. In Section 7.2, we propose multiscale models that improve upon these two approximations by using statistically accurate descriptions of the interpolation and displacement.

■ 7.1.3 Wavelet Decompositions

Because fBm is both non-stationary and self-similar, the wavelet transform seems to provide a natural decomposition of fBm. The (orthonormal) wavelet synthesis of fBm is given by

$$x(t) = \sqrt{2^J} \sum_{k=-\infty}^{\infty} a_J[k]\, \phi(2^J t - k) + \sum_{m=J}^{\infty} \sqrt{2^m} \sum_{k=-\infty}^{\infty} d_m[k]\, \psi(2^m t - k), \qquad (7.17)$$

where $\psi(t)$ is the wavelet function, $\phi(t)$ is the scaling function, $d_m[k]$ are the detail coefficients, and $a_J[k]$ are the approximation coefficients at scale J. The index k is the spatial offset and m is the scale index.[4]

As shown in [96], fBm is made stationary when filtered by any ideal bandpass filter. The wavelet function $\psi(t)$ is the impulse response of a bandpass filter, albeit not ideal. Because $1/f$ processes are made stationary when filtered by ideal bandpass filters [96], it should not be surprising that the detail coefficients at any given scale m are stationary [36]. The variance of the stationary process at scale m is [36]

$$\mathrm{var}\big[d_m[k]\big] = \frac{\sigma^2}{2} V(H, \psi) \left(\frac{1}{2}\right)^{(2H+1)m}, \qquad (7.18)$$

where $V(H, \psi)$ is a constant that depends on only the Hurst exponent and the wavelet function. The variance of the wavelet detail coefficient decreases geometrically with scale at a rate proportional to H, as do the displacements in random midpoint displacement.

If the wavelet coefficients are to be used to approximate fBm, the detail coefficients must be sufficiently close to independent. Mutual independence leads to efficient synthesis algorithms, since the detail coefficients can be generated independently. As shown in [36, 90, 96], the detail coefficients at any given scale are independent only for $H = 1/2$. For $H \neq 1/2$, the correlation between detail coefficients at any given scale decays asymptotically as [36]

$$E\big[d_m[k]\, d_n[l]\big] \sim \mathcal{O}\big(|k - l|^{2(H-R)}\big), \qquad (7.19)$$

where R is the number of vanishing moments of the wavelet function, i.e., the regularity of the wavelet function. The correlation between detail coefficients at different scales behaves similarly, i.e., they are approximately uncorrelated if H and R are chosen appropriately. Note that even for $H = 1/2$ the detail coefficients at different scales are not all uncorrelated.

The wavelet decomposition can be used to approximately synthesize fBm if the wavelet detail coefficients are assumed to all be independent [36]. Also, because the

[4]Note that the scale index m increases for detail coefficients corresponding to finer resolution wavelet functions. This convention is made to be consistent with the definition of m for the multiscale tree models, but m is the negative of the scale index used in [36].

wavelet transform is an approximate Karhunen-Loève transform of fBm, it can be used to efficiently estimate fBm from dense samples of $y(t) = x(t) + v(t)$, where the $v(t)$ is white noise with constant variance [96]. While this approach is quite attractive, especially since the wavelet decomposition can be applied to $1/f$ processes other than fBm, there are some drawbacks. First, unless the wavelet regularity R is large enough, the fBm synthesized assuming independent detail coefficients will have a number of artifacts due to correlation among the detail coefficients. Also, one must deal with the boundary effects which arise from a wavelet with large R and a finite-length data interval. Finally, there are applications for which the measurements are not dense, are not of the finest scale, or have space-varying measurement noise. An alternative model for fBm, based on the Haar wavelet but using the multiscale stochastic models of [19], is proposed in the following section and overcomes many of the aforementioned difficulties.

■ 7.2 Low-Order Multiscale Models and fBm

In this section, low-order multiscale tree models that approximate the statistics of fBm are developed. The advantage of the multiscale framework is that multiscale tree processes can be efficiently estimated and synthesized [19, 61]. The midpoint displacement algorithm and wavelet synthesis equation, which consist of adding finer and finer details at each step, can be represented by a multiscale autoregression. After defining these tree models, we show how the statistical approximations of fBm can be improved by

- for the midpoint displacement, using a statistically correct formula for the interpolation and displacement, and
- for the wavelet-based model, accounting for local correlations among the detail coefficients.

The displacements and detail coefficient of these enhanced models no longer have constant variance at each scale. (The interpolation for the enhanced displacement model also varies across a given scale.) Numerous examples are provided to illustrate the performance of these enhanced models.

■ 7.2.1 Improved Midpoint Displacement

The random midpoint displacement algorithm of Section 7.1.2 can be directly translated into a multiscale autoregression satisfying Eq. (2.22). This multiscale model for fBm is identical in form to the multiscale model for Markov processes summarized in Section 2.3.2. Brownian motion is Markov, so these models directly apply for $H = 1/2$. For $H \neq 1/2$, the models are approximate in that the finest-scale covariance is only approximately equal to that of fBm. To be more specific, assume that we are interested in modeling fBm on the unit interval, and that $\Delta t_M = (1/2)^{M+1}$ is the finest sampling interval of interest. The $(2^{M+1} + 1)$ samples $x(n\Delta t_M)$ can be synthesized from $M + 1$

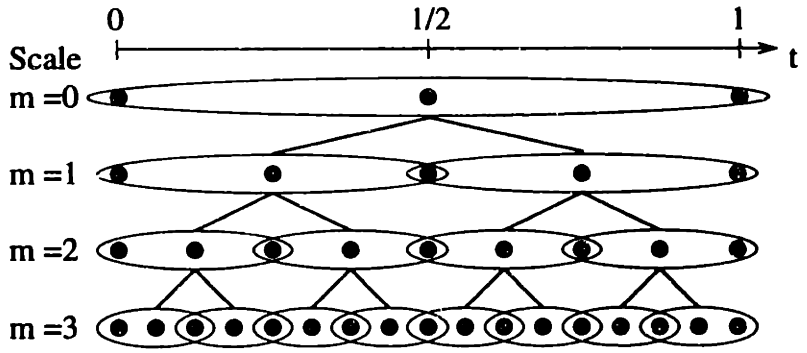


Figure 7.4. A multiscale tree representation of sampled fBm for $\Delta t_M = 1/16$. Each ellipse illustrates the samples contained by a tree variable, e.g., $z(0) = [x(0),\ x(1/2),\ x(1)]^T$.

iterations of the midpoint displacement algorithm given by Eq. (7.15). The first iteration generates $x(0)$, $x(1/2)$, and $x(1)$, which can be mapped to the root node of the tree. Using a binary tree, the $2^{m+1}+1$ samples of $x(t)$ generated by completion of the m-th iteration ($m = 0, \ldots, M$) can be mapped to the 2^m nodes at the m-th scale of the tree. This mapping is illustrated in Figure 7.4 for $\Delta t_M = 1/16$, i.e., $M = 3$. Each variable $z(s)$ contains three samples of $x(t)$, with the finest scale variables containing three consecutive samples spaced by Δt_M. The states at the first two scales of the binary tree are given by $z(0) = [x(0),\ x(1/2),\ x(1)]^T$, $z(0\alpha_1) = [x(0),\ x(1/4),\ x(1/2)]^T$, and $z(0\alpha_2) = [x(1/2),\ x(3/4),\ x(1)]^T$. The multiscale autoregression that implements the midpoint displacement of Eq. (7.15) is given by

$$z(s\alpha_1) = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1 & 0 \end{bmatrix} z(s) + w(0\alpha_1), \tag{7.20a}$$

$$z(s\alpha_2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} z(s) + w(0\alpha_2), \tag{7.20b}$$

$$Q_{0\alpha_1} = Q_{0\alpha_2} = \mathrm{diag}\left(0,\ \sigma^2 \left(\frac{1}{2}\right)^{2H(m(s)+1)} \left[\left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^2\right],\ 0\right). \tag{7.20c}$$

The process noise variances are obtained from Eq. (7.15b). For $H = 1/2$, the tree model will have the correct finest-scale covariance [62]. We refer to the multiscale model defined by Eq. (7.20) the **simple displacement model.**

For $H \neq 1/2$, the distribution of the process at the finest scale of the tree will differ from that of sampled fBm. One reason for the discrepancy, as mentioned in Section 7.1.2, is that the displacements are not truly independent for $H \neq 1/2$, whereas the multiscale process noise is independent by assumption. For instance, consider the generation of $z(0\alpha_1)$ and $z(0\alpha_2)$ from $z(0)$, which is the autoregression from scale $m = 0$ to $m = 1$. For this transition, Eq. (7.20) represents Eq. (7.15) for $t_k = 1/4$ and $3/4$ and

$\Delta t_M = 1/4$, i.e.,

$$E[x(1/4) \mid x(1/2), x(0)] = \frac{1}{2} x(1/2), \tag{7.21a}$$

$$E[x(3/4) \mid x(1), x(1/2)] = \frac{1}{2} \left(x(1) + x(1/2)\right), \tag{7.21b}$$

$$\text{var}[\widetilde{x}(1/4)] = \text{var}[\widetilde{x}(1/4)], \tag{7.21c}$$

$$= \sigma^2 \left(\frac{1}{2}\right)^{2H} \left[\left(\frac{1}{2}\right)^{2H} - \left(\frac{1}{2}\right)^2\right]. \tag{7.21d}$$

The process noise $w(0\alpha_1)$ represents $\widetilde{x}(1/4)$ and $w(0\alpha_2)$ represents $\widetilde{x}(3/4)$. By assumption, these process noise values are independent; however, the true variables $\widetilde{x}(1/4)$ and $\widetilde{x}(3/4)$ are correlated for $H \neq 1/2$, meaning that the multiscale model must introduce some errors in modeling the covariance of fBm.

The other reason noted in Section 7.1.2 for the discrepancy between the covariance at the finest scale of the simple displacement model and the covariance of fBm is that the interpolation and displacement should vary across a given scale. The sample $x(t_k)$ generated according to Equation (7.15) will have variance $\sigma^2|t_k|^{2H}$ only when $k = 0$ or $H = 1/2$. For $H \neq 1/2$, the statistics of the process $x(t)$ on the interval $t \in [t_k - \Delta t_m, t_k + \Delta t_m]$, after conditioning on the samples at the endpoints, depend on the location k of the interval. To derive an interpolation and displacement that produces the correct variance for each sample $x(t_k)$, the following LLSE prediction equations can be used

$$E[x(t_k) \mid X(t_k)] = P_{x(t_k)X(t_k)} P_{X(t_k)}^{-1} X(t_k), \tag{7.22a}$$

$$X(t_k) \triangleq [x(t_k + \Delta t_m), x(t_k - \Delta t_m)]^T, \tag{7.22b}$$

$$\text{var}[\widetilde{x}(t_k)] = P_{x(t_k)} - P_{x(t_k)X(t_k)} P_{X(t_k)}^{-1} P_{X(t_k)x(t_k)}, \tag{7.22c}$$

$$\widetilde{x}(t_k) \triangleq x(t_k) - E[x(t_k) \mid X(t_k)]. \tag{7.22d}$$

The covariance matrices follow from sampling Eq. (7.1).

The tree model representing fBm can be improved without changing the interpretation of the state variables from those of the simple displacement model. Namely, we assume that each $z(s)$ at scale $m(s)$ represents three samples of $x(t)$ spaced by $\Delta t_m = (1/2)^{m+1}$. The model parameters based on the optimal prediction of the samples in $z(s)$ from those in $z(s\overline{\gamma})$ are given by Eqs. (2.30) and (2.32). Call the resulting model the **enhanced displacement model**. For this model, the process noise representing the midpoint displacements is no longer constant across any scale (unless $H = 1/2$), and the interpolation is no longer the simple average of the endpoints. However, the model will still be an approximate representation of fBm, since the internal variables $z(s) = [x(t_k - \Delta t_m), x(t_k), x(t_k + \Delta t_m)]^T$ do not exactly decorrelate fBm on the three intervals $[t_k - \Delta t_m, t_k]$, $[t_k, t_k + \Delta t_m]$, and $[0, t_k - \Delta t_m] \cup [t_k + \Delta t_m, 1]$.

The two approximations of fBm, the simple displacement model defined by Eq. (7.20) and the enhanced displacement model based on optimal prediction, can be compared

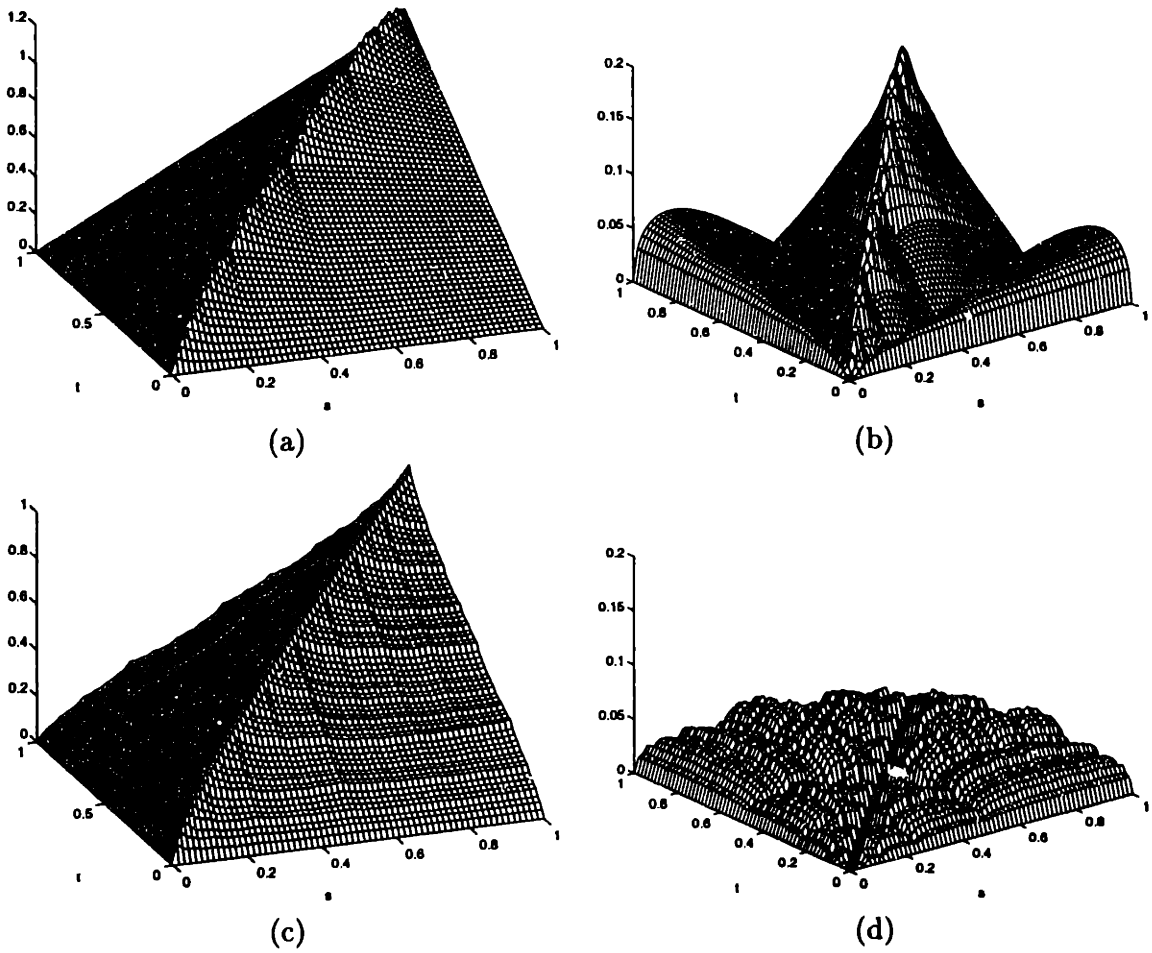


Figure 7.5. A comparison of the simple displacement and enhanced displacement models approximating fBm for $H = 0.3$ and $\sigma^2 = 1$. (a) The finest-scale covariance of the simple displacement model and (b) the absolute value of the difference between Eq. (7.4) and the finest-scale covariance. (c) The finest-scale covariance of the enhanced displacement model and (d) the absolute value of the difference between Eq. (7.4) and the finest-scale covariance.

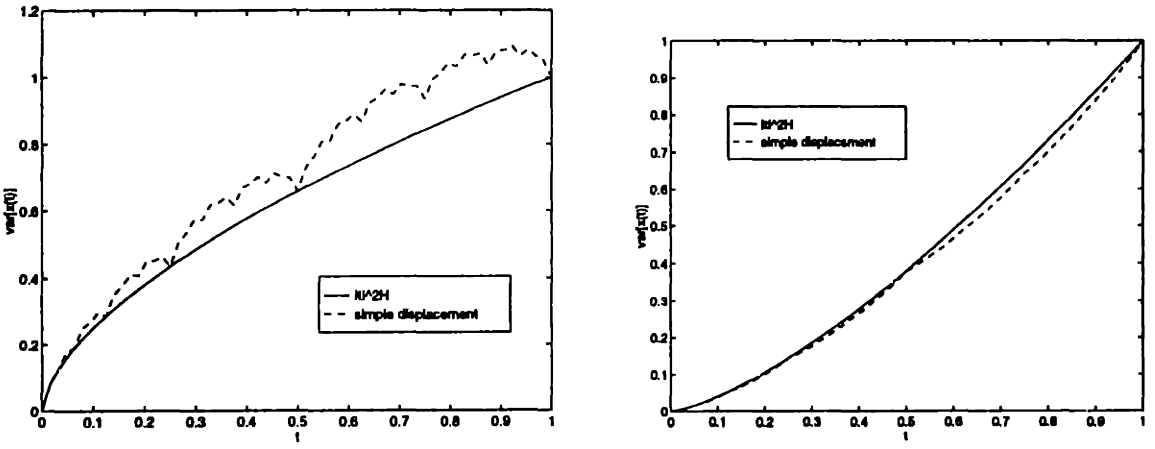


Figure 7.6. The variance $E[x(t)^2] = |t|^{2H}$ of fBm for (a) $H = 0.3$ and (b) $H = 0.7$ is shown by the solid lines. The variance of the process at the finest scale of the simple displacement model is illustrated by dashed lines. The variance of the enhanced displacement model is exact and equal to the solid lines.

to illustrate the effect of the enhancement. Because the finest-scale processes of these two models are meant to represent fBm, the covariance at the finest scale of the two models can be compared with the exact covariance of fBm. Consider fBm with $H = 0.3$ and $\sigma^2 = 1$ on the unit interval $t \in [0, 1]$. Figure 7.5a illustrates the covariance at the finest scale of the simple displacement model, while Figure 7.5b is the absolute value of the difference between the finest-scale covariance and the fBm covariance in Eq. (7.4). Figures 7.5c and 7.5d are the analogous figures for the enhanced displacement model. The variances of the two models are compared to the exact variance $|t|^{0.6}$ in Figure 7.6a. Note that the errors in the finest-scale covariance of the enhanced model are significantly smaller than those of the simple displacement model, especially along the diagonal. In fact, the variances of the samples represented at the finest scale of the enhanced multiscale model are identical to samples of $|t|^{0.6}$. The reason is that the variances of the finest-scale elements of any multiscale model computed from Eqs. (2.30) and (2.32) will be exact, i.e., equal to the diagonal of P_f. The finest-scale covariances of the simple and enhanced displacement models are compared for $H = 0.7$ in Figure 7.6b and Figure 7.7. Again, the enhanced displacement model is an improvement over the simple displacement model, especially in terms of representing the variance of fBm. However, the errors in general are much smaller for $H = 0.7$ than for $H = 0.3$. The reason is that, the larger the value of H, the smaller the variances of the displacements at fine-scales. Therefore, even if the displacement variances are modeled incorrectly, they will have less influence on the covariance of the finest-scale process.

The bottom line is that both of the displacement models produce better approximations of fBm for $H > 1/2$ than for $H < 1/2$, but the enhanced model is more accurate for all values of H.

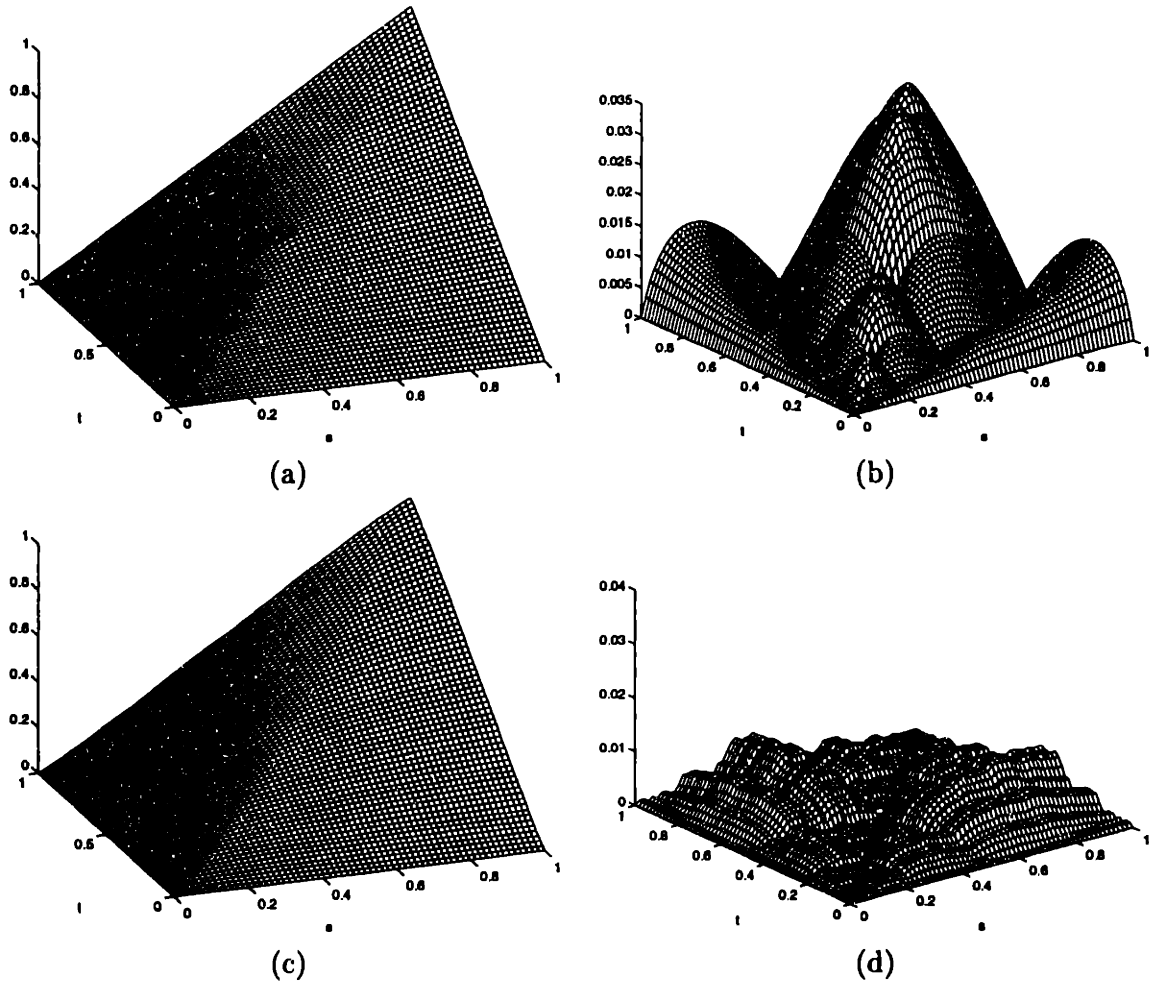


Figure 7.7. A comparison of the simple displacement and enhanced displacement models approximating fBm for $H = 0.7$ and $\sigma^2 = 1$. (a) The finest-scale covariance of the simple displacement model and (b) the absolute value of the difference between Eq. (7.4) and the finest-scale covariance. (c) The finest-scale covariance of the enhanced displacement model and (d) the absolute value of the difference between Eq. (7.4) and the finest-scale covariance.

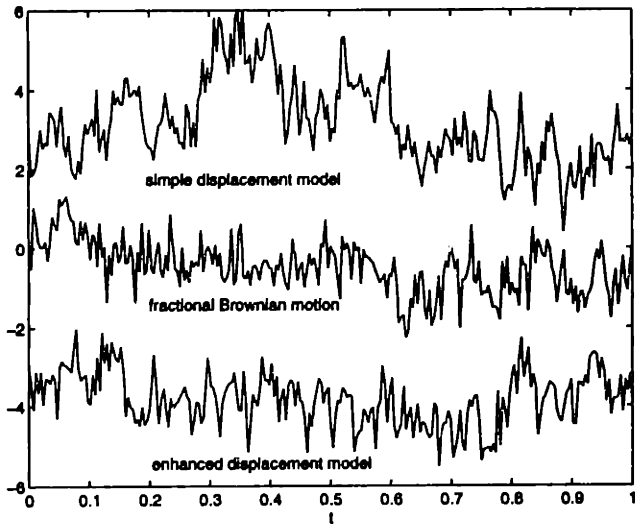


Figure 7.8. Sample paths of fBm, random midpoint displacement, and modified displacement for $H = 0.1$. A DC value of three is added to the midpoint displacement sample path and a DC value of three is subtracted from the modified displacement sample path to allow comparison.

To see how errors in the finest-scale covariance are manifested in the sample paths of the two models, consider $\sigma^2 = 1$ and $H = 0.1$. A small value of H is chosen to maximize the chance that artifacts in the covariances will be visible from sample paths. Figure 7.8 shows sample paths at the finest-scales of the two multiscale models, as well as a sample path of fBm. While the sample path of the enhanced displacement model appears similar to the sample path of fBm, the sample path of the simple displacement model has larger small-scale variations due to the incorrect variances used at small scales by the random midpoint displacement algorithm.

Approximating fBm within the multiscale framework allows one to accomplish a number of signal processing tasks. Consider the estimation of $x(t)$ from noisy measurements. A sample path of fBm (for $H = 0.3$ and $\sigma^2 = 1$) is illustrated in Figure 7.9a. Consider estimating $x(t)$ for $t \in [0, 1]$ from sparse and noisy measurements near the endpoints of the interval. The noisy measurements can be represented as $y(t_i) = x(t_i) + v_i$, where $\text{var}[v_i] = 0.05$, and they are illustrated by o's in Figure 7.9a. The LLSE estimate of $x(t)$ based on the exact fBm prior model is illustrated by the solid line in Figure 7.9b. The estimates (of the finest-scale processes) produced by the two multiscale models are given by the dashed (simple displacement model) and dotted (enhanced displacement model) lines. As would be expected from the covariances in Figure 7.5, the enhanced displacement model more closely approximates the estimates based on the fBm prior distribution. However, the finest-scale estimates produced by both models are well within a standard deviation of the estimation error for the exact LLSE estimate.

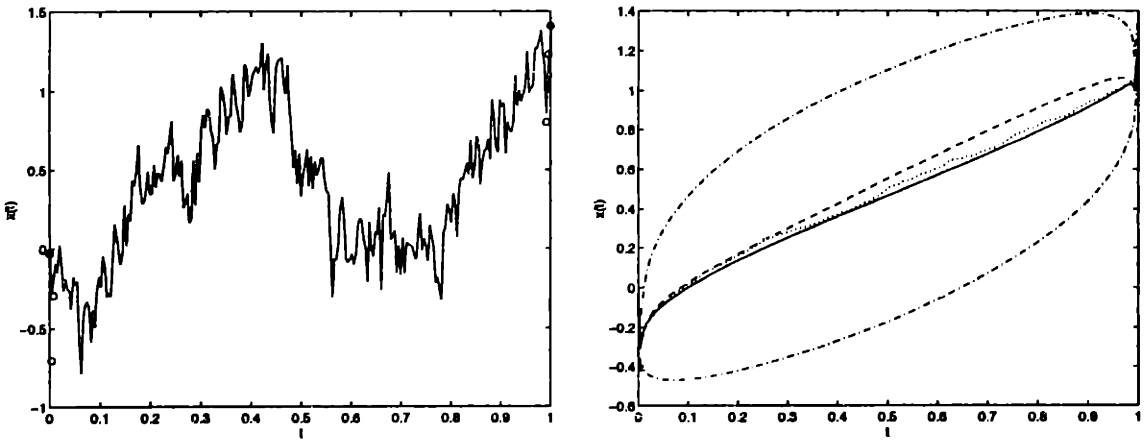


Figure 7.9. (a) A sample path of fBm for $H = 0.3$ and $\sigma^2 = 1$. Noisy measurements are represented by o's. (b) The exact LLSE estimate (solid line), the estimates produced by the two multiscale models (dashed and dotted lines), and the one standard deviation error bars for the exact LLSE estimate (dash-dot line).

■ 7.2.2 Improved Wavelet-Based Models

The approximate independence of the detail coefficients for a wavelet transform of fBm can be used to derive another multiscale representation of fBm. The basic idea is to map the detail coefficients at scale m—the variables $d_m[k]$ in Eq. (7.17)—to the process noise at scale m of a binary tree model. Because the process noise of multiscale tree models is white, these multiscale models will be approximate. For the Haar wavelet

$$\psi(t) = \begin{cases} 1, & 0 \le t < 1/2, \\ -1, & 1/2 \le t < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{7.23}$$

the multiscale tree model has a particularly simple form [35]. Because the wavelet functions $\psi(2^m t - k)$ at any scale m are non-overlapping for all values of k, each sample $x(t)$ depends on only one detail coefficient at each scale. Furthermore, the value of $x(t)$ on any interval $[k/2^m, (k+1)/2^m]$ depends only on the approximation coefficient $a_m[k]$, the detail coefficient $d_m[k]$, and the detail coefficients at scales finer than m. This leads to a multiscale tree model with each variable $z(s)$ containing a detail and a approximation coefficient at scale $m(s)$. As shown in [35], the autoregression for this model is

$$z(s) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \text{mod}(s) \\ 0 & 0 \end{bmatrix} z(s\overline{\gamma}) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix} w_d(s)}_{w(s)}, \tag{7.24a}$$

$$\text{mod}(s) \triangleq \begin{cases} 1, & s \text{ is a left descendent}, \\ -1, & s \text{ is a right descendent}, \end{cases} \tag{7.24b}$$

when s is not a leaf node. Because $z(s)$ represents an approximation coefficient and a detail coefficient at a particular scale, i.e., $z(s) = [a_m[k],\ d_m[k]]^T$ for some value of k, $w_d(s)$ must represent $d_m[k]$. The transition to the finest scale is simply

$$z(s) = \frac{1}{\sqrt{2}} [1 \mod (s)] z(s\overline{\gamma}), \tag{7.25}$$

so that $z(s)$ at the finest scale represents an approximation coefficient at scale M, which for the Haar basis is equal to $2^{M/2}$ times the average value of $x(t)$ over some interval of width $1/2^M$.

As shown in [36, 96], the variance of the detail coefficients for the continuous-time wavelet transform of fBm obey Eq. (7.18). These variances decrease geometrically with increasing scale, but are constant for a given scale m. Equation (7.18) can be used for the variances of the process noise in Eq. (7.24), since each $w_d(s)$ represents a detail coefficient $d_m[k]$. This relationship implies

$$Q_s = \begin{bmatrix} 0 & 0 \\ 0 & \text{var}[w_d(s)] \end{bmatrix}, \tag{7.26a}$$

$$\begin{aligned} \text{var}[w_d(s)] &= \text{var}[d_{m(s)}[k]] \\ &= \frac{\sigma^2}{2} V(H, \psi) \left(\frac{1}{2}\right)^{(2H+1)m(s)}. \end{aligned} \tag{7.26b}$$

The finest scale process of the multiscale model defined by Eqs. (7.24)-(7.26) is an approximation of fBm. There are two reasons for the approximation. First, the detail coefficients are correlated. Because the process noise of multiscale models is assumed to be uncorrelated, the correlation among the detail coefficients cannot be captured by this model. Second, the finest-scale process represents approximation coefficients, i.e., local averages, of $x(t)$. This will lead to approximation errors if, for example, the measurements are of samples of $x(t)$. Representing these measurements at the finest scale of the multiscale tree will lead to measurement errors due to the scale mismatch between the samples measured and the local averages represented at the finest scale. This mismatch will be largest for small H, when there is significant fine-scale energy in fBm. We now discuss multiscale models that address these two approximation.

Local correlations among the detail coefficients can be represented within the multiscale framework to more accurately model the statistics of fBm. Each state $z(s)$ of the multiscale model defined by Eqs. (7.24)-(7.26) represents an approximation coefficient and, if $m(s) \neq M$, a detail coefficient at scale $m(s)$ of the Haar wavelet transform. A more accurate model retains this definition of the state variables, but computes (A_s, Q_s) so that the multiscale autoregression is the optimal prediction of $z(s)$ from $z(s\overline{\gamma})$. These model parameters follow from Eqs. (2.30) and (2.32) when P_f is the covariance of the approximation coefficients at the finest scale, scale $m = M$. Assume that $d_m[k]$ is the detail coefficient represented by $z(s)$. While $w(s)$ in the multiscale model defined by Eqs. (7.24)-(7.26) represents $d_m[k]$, the process noise in the model based on optimal prediction represents $d_m[k]$ *conditioned* on the detail and approximation coefficient

represented by $z(s\overline{\gamma})$, i.e.,

$$Q_s = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{var}[\tilde{w}_d(s)] \end{bmatrix}, \tag{7.27a}$$

$$\mathrm{var}[\tilde{w}_d(s)] = \mathrm{var}\left[d_{m(s)}[k]\right] - E\left[d_{m(s)}[k]z(s\overline{\gamma})^T\right] P^{-1}_{z(s\overline{\gamma})} E\left[z(s\overline{\gamma})d_{m(s)}[k]\right]. \tag{7.27b}$$

Therefore, this multiscale model will capture "local" correlations among the detail coefficients represented by states at neighboring nodes in the tree. Since these are the strongest correlations among the detail coefficients [36], the optimal prediction model will do a better job of approximating the statistics of fBm than does the model with process noise variance given by Eq. (7.26).

The second approximation arises only when one wishes to model samples rather than local averages of fBm. If the finest-scale of the multiscale tree is to represent samples of fBm, then, if the model is to be internal, the state variables can no longer be the coefficients of the continuous-time wavelet transform. As shown in [35], the state variables $z(s)$ can represent coefficients of the discrete-time Haar wavelet transform rather than those of the continuous-time transform. The multiscale model proposed in [35] retains the autoregression of Eqs. (7.24)-(7.25). However, the variances of the detail coefficients given by Eq. (7.26) only apply to the continuous-time wavelet transform. In [35], Fieguth showed, at least for the Haar wavelet transform, that the variances of the discrete-time detail coefficients can significantly differ from those in Eq. (7.26). Fieguth replaced the variances of the detail coefficients in Eq. (7.26) with those of the discrete-time Haar wavelet transform. The variance of the discrete-time detail coefficients is again constant at any given scale, since the wavelet transform of fBm yields a stationary process of detail coefficients at any given scale, but they do not adhere to the strict geometric decay of Eq. (7.26). The deviation from strict geometric decay is greatest for $H \ll 1/2$, when aliasing due to the sampling of fBm is greatest. Call the resulting model the **simple wavelet model**. When samples of fBm are to represented at the finest scale of a multiscale tree model, this model is an improvement over the model defined by Eqs. (7.24)-(7.26), especially for small H. However, the simple wavelet model does not account for any correlations among the detail coefficients. An obvious improvement is to continue to represent coefficients of the discrete-time wavelet transform as the state variables on the tree, but to instead use the optimal prediction equations to compute the multiscale autoregression parameters. These parameters follow from Eqs. (2.30) and (2.32) when P_f is the covariance of the samples represented at the finest scale of the tree. Call this multiscale model the **enhanced wavelet model.**

We can now compare the simple and enhanced wavelet models to determine the effect of representing local correlations in the detail coefficients. The simple wavelet models are shown in [35] to be very useful for estimating the Hurst exponent H from samples of fBm. However, these models are not entirely appropriate for synthesizing or estimating fBm. The covariance at the finest scale of the two multiscale models is illustrated in Figure 7.10 for $H = 0.3$ and $H = 0.7$. The covariances of the simple wavelet model are quite different from the covariances provided in Figure 7.1. Furthermore, the

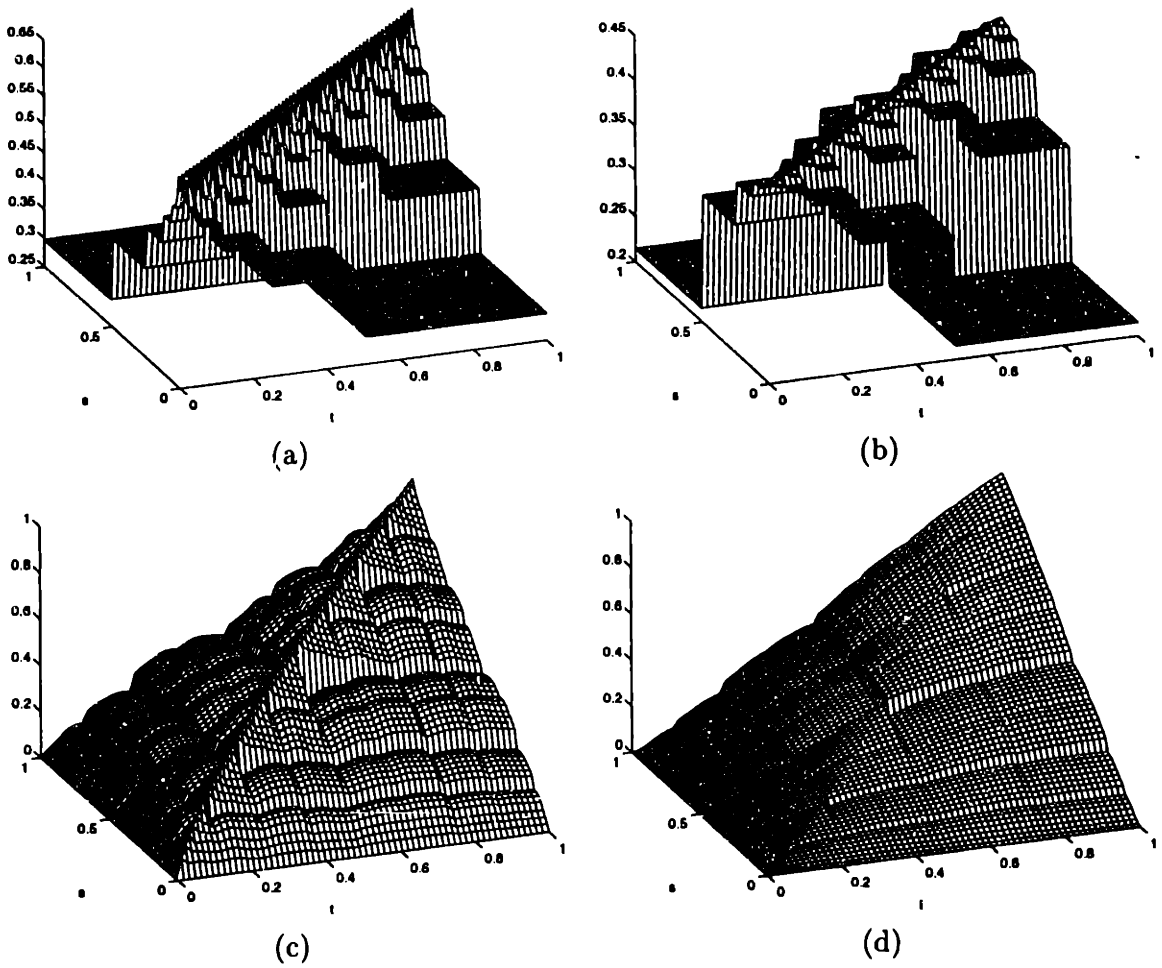


Figure 7.10. The covariance at the finest scale of the simple and enhanced wavelet models: (a) the simple wavelet model for $H = 0.3$ and (b) $H = 0.7$, and the enhanced wavelet model for (c) $H = 0.3$ and (d) $H = 0.7$.

discontinuities in the covariances of Figures 7.10a and 7.10b will lead to discontinuities in estimates or sample paths of the finest-scale process. The covariances at the finest scale of the enhanced wavelet model are illustrated in Figures 7.10c and 7.10d for $H = 0.3$ and $H = 0.7$, respectively. The enhanced wavelet model does a much better job of approximating the covariance of fBm than does the simple wavelet model, without increasing the dimension of the states in the model. The difference between the models is especially noticeable along the diagonal of the covariance matrices, where the enhanced model provides the exact variance of fBm and the simple model provides a constant variance. The variance of the simple wavelet model is constant because the variance of the process noise is constant across any scale.

To see the artifacts introduced by the simple wavelet model, consider the synthesis of the finest-scale process. Sample paths are illustrated in Figure 7.11 for both $H =$

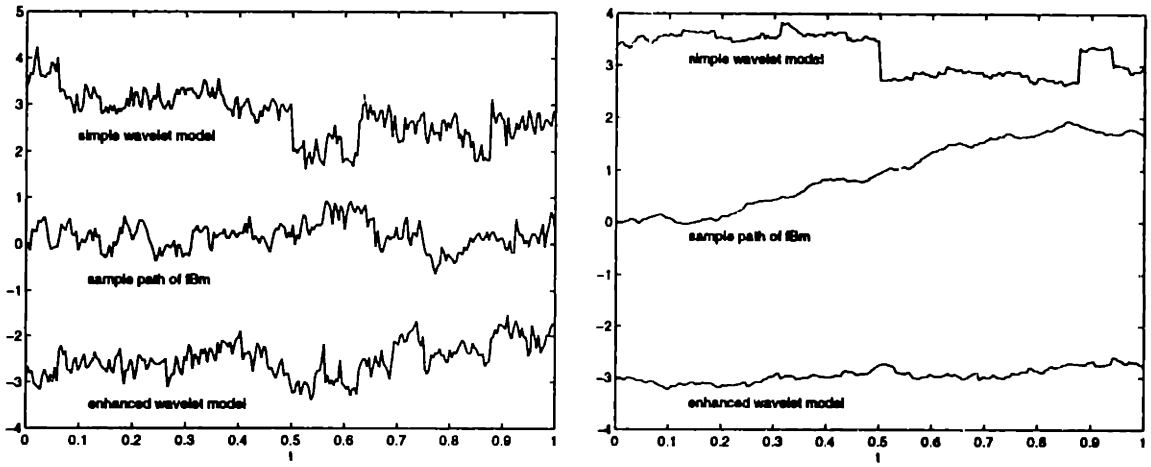


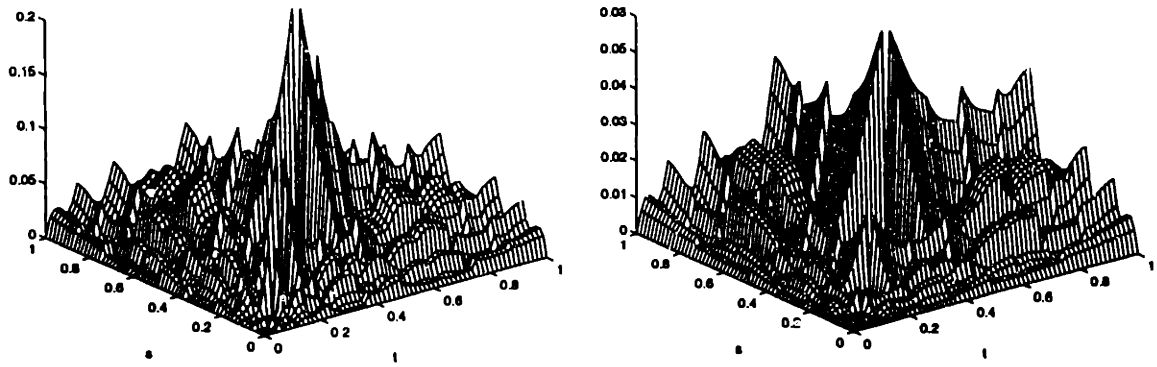
Figure 7.11. Sample paths of fBm and of the finest scale of simple and enhanced wavelet models for (a) $H = 0.3$ and (b) $H = 0.7$. A DC value of three is added to the simple wavelet model sample paths and a DC value of three is subtracted from enhanced wavelet model sample paths.

0.3 and $H = 0.7$. The sample paths generated by this simple wavelet model have distracting artifacts due to the "blockiness" of the covariance function. The artifacts are especially noticeable for $H > 1/2$, as illustrated by the discontinuities in the top signal of Figure 7.11b. These artifacts will also appear in estimates of the finest-scale process from sparse measurements, since such estimates will require interpolation from the prior covariance.

To compare the enhanced wavelet model to the enhanced displacement model, compare Figure 7.12 to Figures 7.5d and 7.7d. The enhanced displacement models generally provide more accurate representations of fBm, especially for $H = 0.7$. Also, for $H = 1/2$, the enhanced displacement model is exact and the enhanced wavelet model is only approximate. However, a direct comparison is not entirely fair, since the dimensions of the states is three for the displacement model, while the dimensions of the states in the wavelet model is two. In the following section, we propose a model that is a synthesis of the displacement and wavelet models.

■ 7.3 Higher-Order Multiscale Models for fBm

Given the multiscale models for approximating fBm described Section 7.2, a natural question is how to develop higher-order multiscale models, i.e., models with larger state dimensions, that more closely approximate the statistics of fBm. Some higher-order models immediately come to mind. For example, a multiscale model could have states that represent approximation and detail coefficients of a more regular (smooth) wavelet transform. The advantage of using more regular wavelet functions is that they lead to detail coefficients that are closer to independent than those of the Haar transform [36].



(a) (b)

Figure 7.12. The absolute difference between the exact covariance of sample fBm and the covariance at the finest scale of the enhanced wavelet model for (a) $H = 0.3$ and (b) $H = 0.7$.

The correlation among the detail coefficients is the source of the approximation errors in the finest-scale covariances of the wavelet-based multiscale models. The disadvantage is that the states of multiscale models based on more regular wavelets will have dimensions larger than two due to the overlap of the wavelet functions at any given scale. In other words, because samples of $x(t)$ will depend on the value of more than one wavelet coefficient at any given scale, the multiscale variables have to represent more than one detail coefficient.

Another higher-order multiscale model that more accurately represents fBm is given by adding more Haar detail coefficients to the variables of the multiscale tree. The enhanced wavelet model described in Section 7.2.2 represents a single detail and approximation coefficient at each state, so that the only correlations between detail coefficients represented by the multiscale model are those between detail coefficients represented at neighboring nodes on the tree. Any approximation errors in the covariance of the finest-scale process are due to correlations in the detail coefficients not represented by the model. Thus, the accuracy of the approximation will increase when more detail coefficients are represented by the each state.

The implementations of either of these higher-order multiscale models is rather straightforward, but they are not considered in this chapter. Instead, we first consider a higher-order model that is the synthesis of the wavelet and displacement models. Recall that the approximations in the multiscale model based on midpoint displacement are most significant for $H < 1/2$, since the energy in the displacements is largest for $H < 1/2$. In contrast, the Haar wavelet detail coefficients are more strongly correlated for $H > 1/2$ than for $H < 1/2$. A multiscale model that synthesizes the enhanced displacement and wavelet models should overcome the drawbacks of the two individual models. The basic idea is to combine the states of the displacement and wavelet models, and then to compute the model parameters according to optimal prediction, i.e.,

according to Equations (2.30) and (2.32).

To be more specific, assume that we wish to model N evenly spaced samples of fBm at the finest scale of the multiscale tree. Let $x[n] = x(n\Delta t)$ for $1 \leq n \leq N$ be the N samples. For notational simplicity, assume that that $N = 2^{M+2}$, so that we can choose a binary tree with M scales and four samples of $x[n]$ represented at each of the 2^M finest-scale nodes. This mapping to the finest scale is illustrated in Figure 7.13 for $M = 2$. For each node s, the finest-scale descendents of s represent $x[n]$ on an interval of width $4 \cdot 2^{M-m(s)}$ samples, i.e., $[4k\,2^{M-m(s)} + 1,\ 4(k+1)2^{M-m(s)}]$ for some value $0 \leq k \leq 2^{m(s)}$. The discrete-time Haar approximation and detail coefficients with support on the interval $[4k\,2^{M-m(s)} + 1,\ 4(k+1)2^{M-m(s)}]$ are denoted by $a_m[k]$ and $d_m[k]$. The variable of the multiscale tree that combines samples and wavelet coefficients follows as

$$z(s) = \begin{bmatrix} x[n_1] \\ x[n_2] \\ x[n_3] \\ x[n_4] \\ a_m[k] \\ d_m[k] \end{bmatrix}, \tag{7.28}$$

$$n_1 \triangleq 4k\,2^{M-m(s)} + 1\,,$$
$$n_2 \triangleq (4k+2)\,2^{M-m(s)}\,,$$
$$n_3 \triangleq (4k+2)\,2^{M-m(s)} + 1\,,$$
$$n_4 \triangleq 4(k+1)2^{M-m(s)}\,,$$

where n_1 and n_2 are the endpoints of the interval descending from $s\alpha_1$ and n_3 and n_4 are the endpoints of the interval descending from $s\alpha_2$. The samples represented by each state are illustrated in Figure 7.13 for $M = 2$. Note that $z(s)$ contains four samples of the finest-scale process, rather than the three samples used for the midpoint displacement model. The four samples are used only for symmetry. Because $x[n_2]$ and $x[n_3]$ are consecutive samples and will be highly correlated, removing either $x[n_2]$ or $x[n_3]$ has minimal effect on the finest-scale statistics of the resulting tree model, (as we have verified in simulations). The parameters of the tree model can be calculated using Eqs. (2.30) and (2.32). We refer to the resulting multiscale model the **endpoint-average model**.

The improved accuracy, in terms of representing fBm, of the endpoint average model over the enhanced displacement and wavelet models can be seen from Figure 7.14. Figure 7.14 shows the absolute value of the difference between the covariance at the finest scale of the endpoint average model and the exact covariance in Eq. (7.4). Comparing Figure 7.14a with Figures 7.5d and 7.12a, we see that the covariance errors are smallest for the endpoint average model when $H = 0.3$. Comparing Figure 7.14b with Figures 7.7d and 7.12b, we can conclude again that the covariance errors are smallest

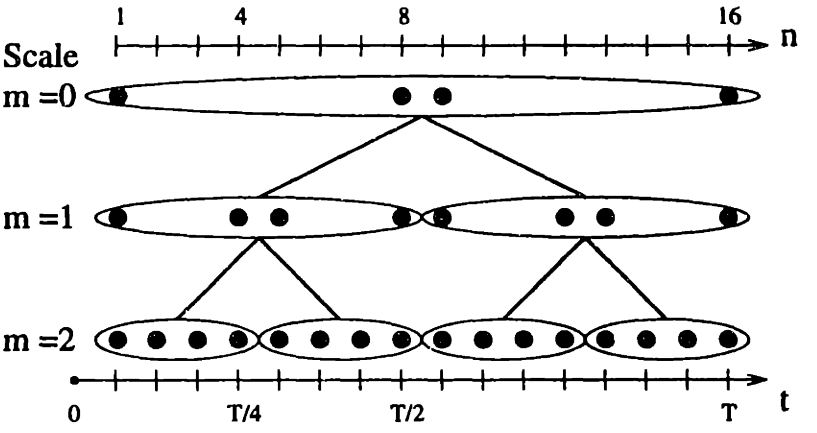


Figure 7.13. For sixteen samples of $x[n]$ represented at the finest scale, the states $z(s)$—represented by ellipses—of the multiscale model contain the endpoints of the two finest-scale intervals descendent from node s. For the endpoint-average model, the appropriate Haar wavelet transform coefficients must also be added to the states.

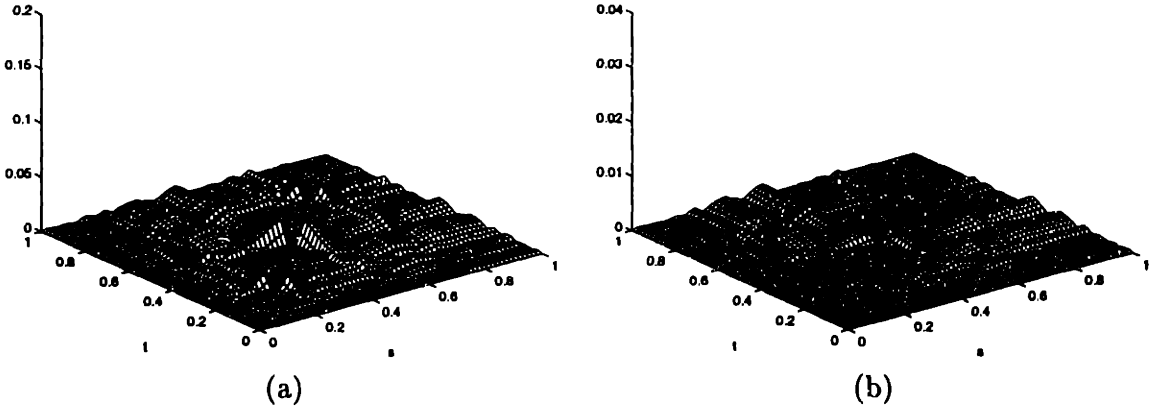


Figure 7.14. The absolute difference between Eq. (7.4) and the covariance of the finest-scale process of the "endpoint-average" multiscale model model for (a) $H = 0.3$ and (b) $H = 0.7$.

for the endpoint average model when $H = 0.7$. In both cases the maximum errors are reduced significantly using this higher-order multiscale model. The superiority of the endpoint-average model should not be surprising, since it is guaranteed by the Corollary in Section 4.4. Namely, while the finest-scale processes $f_{s\alpha_1}$, $f_{s\alpha_2}$, and f_{s^c} partitioned by node s are not completely uncorrelated after conditioning on either the endpoints or the detail coefficients in Eq. (7.28), the two combined can only decrease the decorrelation as measured by the correlation function $\overline{\rho}$ defined by Eq. (2.42).

While the endpoint-average model improves on the multiscale models discussed in Section 7.2, it is at the cost of increased state dimensions and hence a decrease in efficiency. The real question, however, is how close the endpoint-average model is

to the optimal tree model with state dimensions of six, i.e., the sixth-order multiscale process that most accurately models fBm. We do not attempt to completely answer this question, but instead propose a systematic approach for designing multiscale tree models that are in some sense optimal. These models are based on Canonical Correlations analysis, but use the statistical self-similarity and stationary increments of fBm to greatly simplify the number of computations required for the design of the multiscale models. As an example, the sixth-order multiscale approximation of fBm returned by the method we now describe has covariance errors of roughly an order of magnitude less than those of the endpoint-average model.

■ 7.3.1 A Canonical Correlations Realization for fBm

In the remainder of this chapter, a general method for realizing multiscale models of fBm is presented. The goal is to develop an efficient realization algorithm that provides an optimal trade-off between the state dimensions of the multiscale process and the statistical fidelity at the finest scale. This method is based on Canonical Correlations, but makes use of the statistical self-similarity and stationary increments property of fBm. The statistical self-similarity leads to a relationship between internal variables at neighboring scales that allows us to simplify the multiscale modeling of any statistically self-similar process. We next show how the stationary increments property leads to further simplifications and the efficient design of multiscale models for fBm. The basic result is that the Canonical Correlations decomposition of the finest-scale covariance matrix need only be computed at a small and fixed number of times, rather than a number of decompositions proportional to the number of finest-scale nodes.

To illustrate the implications of self-similarity for multiscale realization, first consider an example. Assume that samples of a random process $x(t)$ are to be represented by the nodes at the finest scale of a binary multiscale tree. Also assume that we are only interested in modeling $x(t)$ on the interval $[0,T]$. Recall that f_s is always defined as vector of the finest-scale process descendent from node s. From Section 2.3.4, the internal variables can be restricted to the form

$$z(s) = W_s f_s\,, \tag{7.29}$$

where f_s contains the finest-scale descendents of node s. The vector f_s represents $x(t)$ on some interval $[t_1, t_3]$. The internal variable $z(s)$ is chosen such that the correlation among the three vectors $f_{s\alpha_1}$, $f_{s\alpha_2}$, and f_{s^c} is minimized. The minimization is over a fixed dimension $d(s)$ for $z(s)$ and the correlation is measured by $\overline{\rho}(f_{s\alpha_1}, f_{s\alpha_2}, f_{s^c} \mid W_s f_s)$. Using Canonical Correlations, the internal variable $z(s)$ is determined in two steps [49].

1. Determine the linear combination $T_1 f_{s\alpha_1}$ that decorrelates $f_{s\alpha_1}$ from $f_{s\alpha_1^c}$. As illustrated in Figure 7.15a, this amounts to decorrelating samples of $x(t)$ on the interval $[t_1, t_2]$ from those on $[t_1, t_2]^c$, which is the complement of the interval $[t_1, t_2]$.

2. Determine the linear combination $T_2 f_{s\alpha_2}$ that decorrelates $f_{s\alpha_2}$ from $f_{s\alpha_2^c}$. This amounts to decorrelating samples of $x(t)$ on the interval $[t_2, t_3]$ from those on $[t_2, t_3]^c$.

Both T_1 and T_2 can be computed using Canonical Correlations decompositions of P_f. The internal matrix W_s then follows as[5]

$$z(s) = \underbrace{\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}}_{W_s} \underbrace{\begin{bmatrix} f_{s\alpha_1} \\ f_{s\alpha_2} \end{bmatrix}}_{f_s}. \tag{7.30}$$

The problem with this approach is that each internal variable is computed independently, requiring $\mathcal{O}(N)$ applications of Canonical Correlations for a tree with N finest-scale nodes.

To take advantage of the self-similarity of fBm, a particular mapping of the samples of $x(t)$ to the finest scale of the tree must be chosen. Namely, the mapping must be chosen such that for any node s (not at the finest scale) there exists a node at scale $m(s) + 1$ whose finest-scale descendents are self-similar to the descendents of node s. If the descendents of node s represent $x(t)$ on the interval $[t_1, t_3]$, then the descendents of some node τ at scale $m(\tau) = m(s) + 1$ must represent the interval $[t_1/2, t_3/2]$. This interval is a compression of $[t_1, t_3]$ by a factor of two. The relationship between nodes s and τ is illustrated in Figure 7.15. As a specific example, consider Figure 7.13. The finest-scale interval descending from the root node is $(0, T]$, while the interval descending from node $0\alpha_1$ is $(0, T/2]$.

For a binary tree with M scales, ignoring the discretization effects, the mapping of $x(t)$ that leads to this self-similar relationship between the variables at neighboring scales is the following: each finest-scale node must represent an interval of fixed width $T/2^M$, where the intervals are nonoverlapping and neighboring nodes represent consecutive intervals. For example, two finest-scale nodes with a common parent will represent the intervals $(0, T/2^M]$ and $(T/2^M, T/2^{M-1}]$. The parent of these two nodes has finest-scale descendents representing $(0, T/2^{M-1}]$, which is an expansion of $(0, T/2^M]$ by a factor of two. Once the finest-scale intervals have been chosen, the form of the internal variable $z(\tau)$ can be approximately derived from that of $z(s)$. When $x(t)$ is statistically self-similar, the process on the intervals represented at descendents of node s are self-similar to those descendent from node τ. Thus, were it not for the discrete nature of the finest-scale process, the internal matrix W_τ would be identical to W_s. (Recall that W_s and W_τ are the internal matrices with d rows that minimize $\overline{\rho}(f_{s\alpha_1}, f_{s\alpha_2}, f_{s^c} \mid W_s f_s)$ and $\overline{\rho}(f_{\tau\alpha_1}, f_{\tau\alpha_2}, f_{\tau^c} \mid W_\tau f_\tau)$, respectively.) However, because of discretization, f_τ has one-half the number of elements of f_s, and one must be careful when deriving W_τ from W_s. In what follows, we show how an approximation of W_τ can be derived from W_s, and more generally how all the internal matrices at some scale $m + 1$ can be derived

[5]One issue that is not addressed here is how to allocate the dimension $d(s)$ between $T_1 f_{s\alpha_1}$ and $T_2 f_{s\alpha_2}$.

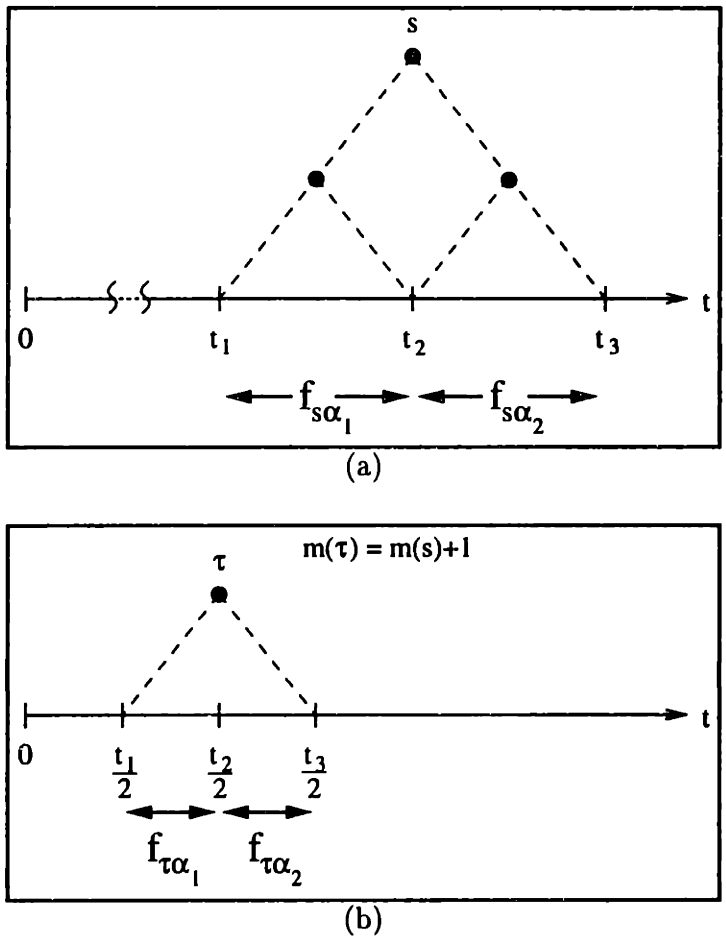


Figure 7.15. (a) The finest-scale descendents of node s represent samples of $x(t)$ on the interval $[t_1, t_3]$, while the finest-scale descendents of $s\alpha_1$ and $s\alpha_2$ represent $x(t)$ on $[t_1, t_2]$ and $[t_2, t_3]$, respectively. (b) A node τ at scale $m(\tau) = m(s) + 1$ has finest-scale descendents that represent the compressed interval $[t_1/2, t_3/2]$.

from the internal matrices at scale m. This leads to significant computational savings in the realization of internal multiscale models that have self-similar processes at the finest-scale.

To show how the internal variables at scale $m+1$ are related to those at scale m when $x(t)$ is statistically self-similar, first consider the decorrelation of two *intervals* of $x(t)$. The correlation of two intervals of $x(t)$ can be defined analogous to the correlation between two vectors defined by Eq. (2.38). Define

$$\mathcal{L}[t_1,t_2] \triangleq \text{the set of bounded linear functionals of } x(t) \text{ on the interval } [t_1,t_2].$$

Recall that the correlation between two scalar random variables is

$$\rho(u,v) = \frac{E\big[(u-m_u)(v-m_v)\big]}{\sigma_u \sigma_v}, \tag{7.31}$$

where σ_u is the standard deviation of u. The correlation conditioned on y is

$$\rho(u,v\,|\,y) = \frac{E\big[(u-m_u)(v-m_v)\,|\,y\big]}{\sigma_{u|y}\,\sigma_{v|y}}. \tag{7.32}$$

The correlation between $x(t)$ for $t \in [t_1,t_2]$ and $x(t)$ for $t \in [t_3,t_4]$ is defined as

$$\overline{\rho}\big(x,[t_1,t_2],[t_3,t_4]\big) \triangleq \max_{\substack{\{\ell_1 \in \mathcal{L}[t_1,t_2]\} \\ \{\ell_2 \in \mathcal{L}[t_3,t_4]\}}} \rho\big(\ell_1(x),\ell_2(x)\big), \tag{7.33}$$

and the correlation conditioned on y is

$$\overline{\rho}\big(x,[t_1,t_2],[t_3,t_4]\,|\,y\big) \triangleq \max_{\substack{\{\ell_1 \in \mathcal{L}[t_1,t_2]\} \\ \{\ell_2 \in \mathcal{L}[t_3,t_4]\}}} \rho\big(\ell_1(x),\ell_2(x)\,|\,y\big). \tag{7.34}$$

Note that, due to the homogeneity of linear operators, $\rho\big(\ell_1(\alpha x),\ell_2(\beta x)\big) = \rho\big(\ell_1(x),\ell_2(x)\big)$ for any two real scalars α and β. Also, because conditioning on γy is equivalent to conditioning on y for any scalar γ, we have

$$\rho\big(\ell_1(\alpha x),\ell_2(\beta x)\,|\,\gamma y\big) = \rho\big(\ell_1(x),\ell_2(x)\,|\,y\big). \tag{7.35}$$

This implies that

$$\overline{\rho}\big(\alpha x,[t_1,t_2],[t_3,t_4]\,|\,\gamma y\big) = \overline{\rho}\big(x,[t_1,t_2],[t_3,t_4]\,|\,y\big) \tag{7.36}$$

for any two scalars α and γ.

Define $x_a(t) \triangleq x(at)$ for an $a > 0$. We will use the following theorem to relate internal variables at neighboring scales when the finest-scale process is statistically self-similar.

Theorem 4 *Assume that the linear functional $\ell \in \mathcal{L}[t_1, t_2]$ satisfies*

$$\ell = \arg \min_{\ell_0 \in \mathcal{L}[t_1,t_2]} \overline{\rho}\big(x, [t_1, t_2], [t_3, t_4] \,\big|\, \ell_0(x)\big) . \tag{7.37}$$

For any statistically self-similar process $x(t)$ and any scalar $a > 0$, the bounded linear functional $\hat{\ell}(x) \in \mathcal{L}[at_1, at_2]$ that satisfies $\hat{\ell}(x) = \ell(x_a)$ must also satisfy

$$\hat{\ell} = \arg \min_{\hat{\ell}_0 \in \mathcal{L}[at_1,at_2]} \overline{\rho}\big(x, [at_1, at_2], [at_3, at_4] \,\big|\, \hat{\ell}_0(x)\big) . \tag{7.38}$$

Proof: Recall that $x_a(t) \stackrel{\mathcal{P}}{=} a^H x(t)$. For any two linear functionals $\ell_1 \in \mathcal{L}[t_1, t_2]$ and $\ell_2 \in \mathcal{L}[t_3, t_4]$, self-similarity implies that $\rho(\ell_1(x), \ell_2(x)) = \rho(\ell_1(x_a), \ell_2(x_a))$, since

$$\begin{aligned} E\big[\ell_1(x)\ell_2(x)\big] &= a^{-2H} E\big[\ell_1(x_a)\ell_2(x_a)\big] , \\ \operatorname{var}\big[\ell_1(x)\big] &= a^{-2H} \operatorname{var}\big[\ell_1(x_a)\big] , \\ \operatorname{var}\big[\ell_2(x)\big] &= a^{-2H} \operatorname{var}\big[\ell_2(x_a)\big] . \end{aligned}$$

Riesz's Lemma [83] can be used to show that for any bounded linear functional $\ell(x)$, there exists a function $g(t)$ such that

$$\ell(x) = \int g(t)x(t)\,dt .$$

Applying a change of variables to such integrals, there exist linear functionals $\hat{\ell}_1 \in \mathcal{L}[at_1, at_2]$ and $\hat{\ell}_2 \in \mathcal{L}[at_3, at_4]$ such that $\hat{\ell}_1(x) = \ell_1(x_a)$ and $\hat{\ell}_2(x) = \ell_2(x_a)$. The existence of these linear functionals leads to

$$\begin{aligned} \max_{\substack{\{\ell_1 \in \mathcal{L}[t_1,t_2]\} \\ \{\ell_2 \in \mathcal{L}[t_3,t_4]\}}} \rho\big(\ell_1(x), \ell_2(x) \,\big|\, \ell(x)\big) &= \max_{\substack{\{\ell_1 \in \mathcal{L}[t_1,t_2]\} \\ \{\ell_2 \in \mathcal{L}[t_3,t_4]\}}} \rho\big(\ell_1(a^H x), \ell_2(a^H x) \,\big|\, \ell(a^H x)\big) , \\ &= \max_{\substack{\{\ell_1 \in \mathcal{L}[t_1,t_2]\} \\ \{\ell_2 \in \mathcal{L}[t_3,t_4]\}}} \rho\big(\ell_1(x_a), \ell_2(x_a) \,\big|\, \ell(x_a)\big) , \\ &= \max_{\substack{\{\hat{\ell}_1 \in \mathcal{L}[at_1,at_2]\} \\ \{\hat{\ell}_2 \in \mathcal{L}[at_3,at_4]\}}} \rho\big(\hat{\ell}_1(x), \hat{\ell}_2(x) \,\big|\, \hat{\ell}(x)\big) , \end{aligned}$$

where the last equality follows from $\hat{\ell}(x) = \ell(x_a)$. Therefore,

$$\overline{\rho}\big(x, [t_1, t_2], [t_3, t_4] \,\big|\, \ell(x)\big) = \overline{\rho}\big(x, [at_1, at_2], [at_3, at_4] \,\big|\, \hat{\ell}(x)\big) .$$

The result of the theorem follows.
Q.E.D.

Theorem 4 can be extended to open intervals or unions of intervals, e.g., $[t_1, t_2)$ or $(t_1, t_2) \cup (0, t_1/2)$ in lieu of $[t_1, t_2]$. Theorem 4 also applies if a vector of linear functionals is substituted for $\ell(x)$.

Theorem 4 basically states that the linear functionals that maximally decorrelate two intervals of a statistically self-similar process can also be used to determine the linear functionals that maximally decorrelate any common expansions or contractions of these intervals. For example, return to the example illustrated in Figure 7.15. Assume for the moment that we are interested in modeling fBm on $[0,\infty)$, i.e., $T = \infty$. Under this assumption, the complement of the interval $[t_1, t_2]$ is $[t_1, t_2]^c = [0, t_1) \cup (t_2, \infty)$. Also assume that $z(s)$ contains the set of linear functionals that

- maximally decorrelates $x(t)$ on $[t_1, t_2]$ from $x(t)$ on $[t_1, t_2]^c$ and
- maximally decorrelates $x(t)$ on $[t_2, t_3]$ from $x(t)$ on $[t_2, t_3]^c$.

Using Theorem 4, we can also determine from these linear functionals in $z(s)$ the internal variable $z(\tau)$ that

- maximally decorrelates $x(t)$ on $[t_1/2, t_2/2]$ from $x(t)$ on $[t_1/2, t_2/2]^c$ and
- maximally decorrelates $x(t)$ on $[t_2/2, t_3/2]$ from $x(t)$ on $[t_2/2, t_3/2]^c$.

To derive $z(\tau)$ from $z(s)$, note that the i-th element of $z(s)$ can be expressed as

$$\ell_i(x) = \int_{t_1}^{t_3} g_i(t)x(t)\,dt \tag{7.39}$$

for some function $g_i(t)$. The corresponding element in $z(\tau)$ follows from Theorem 4 as $\hat{\ell}_i(x) = \ell_i(x_a)$, i.e.,

$$\ell_i(x_a) = \int_{t_1}^{t_3} g_i(t)x(at)\,dt\,, \tag{7.40a}$$

$$= \int_{at_1}^{at_3} \big(g_i(t/a)/a\big)x(t)\,dt\,, \tag{7.40b}$$

$$= \hat{\ell}_i(x)\,. \tag{7.40c}$$

Since $a = 1/2$ for this example, the kernel defining the i-th element of $z(\tau)$ is $2g_i(2t)$, a time-compression and magnitude-expansion of $g_i(t)$ by a factor of two. However, remember that conditioning on $2g_i(2t)$ is equivalent to conditioning on $g_i(2t)$, so that only the time-compression is important in the derivation of $z(\tau)$.

These results might lead one to believe that, given the internal variables for all the nodes at a single scale m, the internal variables at all other nodes can be derived directly from those at scale m. There are two problems with this line of reasoning. First, the finest-scale represents samples of $x(t)$, so that the compression given by Eq. (7.40) cannot be directly applied. Second, the interval $[0, T]$ modeled at the finest scale of the tree is finite, so that $[t_1/2, t_2/2]^c$ is not related to $[t_1, t_2]^c$ by a simple compression. The following discusses how to overcome these problems, and how the results lead to an algorithm for modeling fBm.

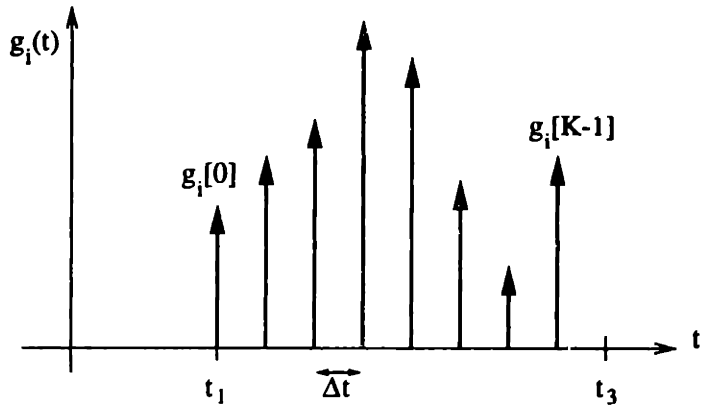


Figure 7.16. A function of impulse spaced by Δt on the interval $[t_1, t_3]$. The area of the impulse at $t_1 + k\Delta t$ is $g_i[k]$, $0 \leq k \leq K-1$, where $K = (t_3 - t_1)/\Delta t$.

The Effect of Sampling on Theorem 4

Using Eq. (7.40) to derive $z(\tau)$ from an internal variable at scale $m(\tau) - 1$ assumes that the internal variables are linear functionals of the continuous-time process $x(t)$. However, because the process at the finest-scale of a multiscale trees usually represents samples of $x(t)$, the internal variables will instead be linear functions of *samples* of $x(t)$. In this case, Equation (7.40) cannot be used to determine the internal variable that corresponds statistically to a compression of $z(s)$. If $z(s)$ represents linear functions of samples of $x(t)$ spaced by Δt, then each kernel $g_i(t)$ will have the form

$$g_i(t) = \sum_{k=0}^{K-1} g_i[k]\, \delta(t - k\Delta t - t_1)\,, \tag{7.41}$$

which is illustrated in Figure 7.16. The impulses of $g_i(2t)$, however, will be spaced by $\Delta t/2$. Half of these samples are not represented by the finest-scale process. Therefore, we must either choose an external model (by representing at $z(\tau)$ samples that are not represented at the finest scale), or we can approximate $z(\tau)$ in terms of $z(s)$.

To approximate $z(\tau)$ in terms of $z(s)$, it is worthwhile to first consider multiscale models for which the finest-scale represents local averages of $x(t)$ over non-overlapping intervals $[k\Delta t, (k+1)\Delta t]$. (These local averages are equal, within a constant factor, to approximation coefficients of the Haar wavelet transform.) For these models, W_τ can be derived exactly from W_s. Define

$$\phi(t) = \begin{cases} 1, & 0 \leq t < \Delta t, \\ 0, & \text{otherwise.} \end{cases} \tag{7.42}$$

Then $\langle x(t), \phi(t - k\Delta t)\rangle$ is the "average" value of $x(t)$ over the interval $[k\Delta t, (k+1)\Delta t]$.

Each element of $z(s)$ will be expressible as

$$\ell_i(x) = \sum_{k=0}^{K-1} g_i[k] \langle x(t - k\Delta t - t_1), \phi(t - k\Delta t - t_1) \rangle , \tag{7.43a}$$

$$= \underline{g}_i^T f_s , \tag{7.43b}$$

where f_s contains the average values represented at descendents of node s and $\underline{g}_i$ is a vector containing $g_i[k]$. From Eq. (7.40), the corresponding element of $z(\tau)$ is

$$\hat{\ell}_i(x) = \sum_{k=0}^{K-1} 2g_i[k] \langle x(2t - k\Delta t - t_1), \phi(2t - k\Delta t - t_1) \rangle . \tag{7.44}$$

Using $\phi(t) = \phi(2t) + \phi(2t - \Delta t)$,

$$\hat{\ell}_i(x) = \sum_{k=0}^{K/2-1} 2 \left(g_i[2k-1] + g_i[2k] \right) \langle x(t - k\Delta t - t_1/2), \phi(t - k\Delta t - t_1/2) \rangle , \tag{7.45a}$$

$$= \underline{h}_i^T f_\tau , \tag{7.45b}$$

where f_τ contains the average values represented at descendents of node τ. The k-th element of the vector $\underline{h}_i$ is $2\big(g_i[2k-1] + g_i[2k]\big)$.

When the finest scale of the multiscale tree represents samples of the self-similar process $x(t)$, Equation (7.45) can be used as an approximate method for deriving states at scale $m+1$ from the corresponding states at scale m. The approximation is given by replacing the terms $\langle x(t - t_k), \phi(t - t_k) \rangle$ in Eqs. (7.43) and (7.45) with samples of $x(t)$. For fBm, the accuracy of this approximation will depend on the sampling interval Δt and the Hurst exponent H. For small values of H, i.e., H near zero, the statistics of samples of fBm differ most from the statistics of local averages of fBm. The reason is that fBm has significant fine-scale energy for small values of H, so that low sampling rates lead to significant aliasing. Therefore, we would expect for a fixed Δt that the approximation given by Eq. (7.45) to be worst for $H \ll 1/2$, (or, alternatively, that a smaller Δt is needed for smaller H).

The Finite Interval Problem

The second problem with using Eq. (7.40) to determine $z(\tau)$ from $z(s)$ is that an infinite interval was assumed to be represented at the finest scale of the tree. To illustrate the source of the problem, again consider the example illustrated in Figure 7.15. In using Eq. (7.40) to derive $z(\tau)$ from $z(s)$, an implicit assumption was that $[t_1/2, t_2/2]^c$ is a compression of $[t_1, t_2]^c$ and that $[t_2/2, t_3/2]^c$ is a compression of $[t_2, t_3]^c$. If $[0, T]$ is the interval to be represented a the finest scale of the tree, then $[t_1, t_2]^c = [0, t_1) \cup (t_2, T]$ and $[t_1/2, t_2/2]^c = [0, t_1/2) \cup (t_2/2, T]$; the compression relationship will only hold if $T = \infty$. Since T is always finite, the derivation of $z(\tau)$ from Eq. (7.40) will always be

approximate. As an example, consider $z(0)$ and its descendent $z(0\alpha_1)$ in Figure 7.13. The finest-scale descendents of $z(0)$ represent samples of $x(t)$ on the interval $(0,T]$, while the descendents of $z(0\alpha_1)$ represent samples of $x(t)$ on the interval $(0,T/2]$. This interval is a compression of the interval $(0,T]$ by a factor of two. But the complements of these two intervals are not related by a compression. The internal variable $z(0)$ must conditionally decorrelate samples of $x(t)$ on the interval $(0,T/2]$ from those on $(0,T/2]^c = (T/2,T]$. The internal variable $z(0\alpha_1)$ must conditionally decorrelate samples of $x(t)$ on the interval $(0,T/4]$ from those on $(0,T/4]^c = (T/4,T]$. (In addition, $z(0\alpha_1)$ must conditionally decorrelate samples of $x(t)$ on the interval $(T/4,T/2]$ from those on $(T/4,T/2]^c = (0,T/4] \cup (T/2,T]$.) Even though the descendents of $z(0)$ and $z(0\alpha_1)$ are related by a simple compression by a factor of two, there is no such relationship between the intervals decorrelated by the two internal variables.

If the internal variable at node $z(0\alpha_1)$ is to be derived from $z(0)$, one possible solution is to compute $z(0)$ assuming that $(0,T]^c = (T,2^{m_0}T]$ for some positive integer value of m_0. In other words, $z(0)$ will conditionally decorrelate samples of $x(t)$ on the intervals $(0,T/2]$, $(T/2,T]$, and the additional interval $(T,2^{m_0}T]$. For $m_0 = 1$, these intervals are an expansion of the three intervals partitioned by node $0\alpha_1$; therefore, $z(0)$ can be used to derive $z(0\alpha_1)$, say according to Eq. (7.45), so that $z(0\alpha_1)$ decorrelates samples of $x(t)$ on the intervals $(0,T/4]$, $(T/4,T/2]$, and $(T/2,T]$. To compute internal variables for scales $m \geq 2$ from $z(0)$, larger values of m_0 can be used. The obvious trade-off to be made is between the decorrelation supplied by $z(0\alpha_1)$—and decorrelation supplied by the internal variables at finer scales that are also derived from $z(0)$—and the additional computations required to compute $z(0)$.

Another manifestation of the finite interval is that half of the internal variables at scale $m+1$ will have no ancestors at scale m for which the results of Theorem 4 apply. For example, consider node $z(0\alpha_2)$ in Figure 7.13. The finest-scale descendents of node $0\alpha_2$ are samples of $x(t)$ on the interval $(T/2,T]$. However, there is no node at the previous scale whose finest-scale descendents represent the interval $(T,2T]$. The internal variable $z(0\alpha_2)$ can, of course, be computed using a direct application of Canonical Correlations, but the application of similar Canonical Correlations required for analogous variables at finer scales will lead to an algorithm requiring $\mathcal{O}(N)$ Canonical Correlations, where $N = 2^M$ is the number of nodes at the finest scale of the tree.

To overcome this problem, we can instead invoke the stationary increments property of fBm to argue that the internal matrices are approximately constant, i.e., "shift-invariant", across any scale of the tree. To see how the stationary increments property of fBm leads to internal matrices that are effectively shift-invariant, consider the decorrelation of samples of $x(t)$ on two consecutive intervals. The sampled process is denoted by $x[n]$, and the intervals considered are $n \in [m, m+m_1-1]$ and $n \in [m+m_1, m+2m_1-1]$.

The vectors representing these two intervals are

$$x_m = \begin{bmatrix} x[m] \\ x[m+1] \\ \vdots \\ x[m+m_1-1] \end{bmatrix} \quad \text{and} \quad X_m = \begin{bmatrix} x[m+m_1] \\ x[m+m_1+1] \\ \vdots \\ x[m+2m_1-1] \end{bmatrix}. \tag{7.46}$$

The linear function $T_1 x_m$ of dimension d which minimizes $\overline{\rho}(x_m, X_m \,|\, T_1 x_m)$ is given by the Canonical Correlations decomposition of the covariance matrix

$$P_x = \begin{bmatrix} P_{x_m} & P_{x_m X_m} \\ P_{X_m x_m} & P_{X_m} \end{bmatrix}. \tag{7.47}$$

If the eigenvalue (or SVD) decompositions of the matrices P_{x_m} and P_{X_m} are given by

$$P_{x_m} = A_1 S_1 A_1^T,$$
$$P_{X_m} = A_2 S_2 A_2^T,$$

the matrix T_1 is given by the first d rows of the matrix U^T, where $U\Sigma V^T$ is the singular value decomposition of the matrix $S_1^{-1/2} A_1^T P_{x_m X_m} A_2 S_2^{-1/2}$. The terms $S_1^{-1/2} A_1^T$ and $S_2^{-1/2} A_2^T$ effectively normalize the variances of the vectors x_m and X_m. The source of the correlation between x_m and X_m is contained in $P_{x_m X_m}$.

The matrix $P_{x_m X_m}$ can be simplified using the stationary increments property. Namely, using Eq. (7.5), there exists an invertible m_1-by-m_1 transformation Q such that

$$y_m = Q x_m = \begin{bmatrix} x[m] \\ w[m+1] \\ w[m+2] \\ \vdots \\ w[m+m_1-1] \end{bmatrix}, \quad \text{and} \quad Y_m = Q X_m = \begin{bmatrix} x[m+m_1] \\ w[m+m_1+1] \\ w[m+m_1+2] \\ \vdots \\ w[m+2m_1-1] \end{bmatrix}. \tag{7.48}$$

The cross-covariance between x_m and X_m can then be transformed to

$$Q P_{x_m X_m} Q^T = \left[\begin{array}{c|ccc} E[x[m]x[m+m_1]] & E[x[m]w[m+m_1+1]] & \cdots & E[x[m]w[m+2m_1-1]] \\ \hline E[x[m+m_1]w[m+1]] & & & \\ \vdots & & P_w & \\ E[x[m+m_1]w[m+m_1-1]] & & & \end{array}\right] \tag{7.49}$$

where the matrix P_w is stationary due the the stationary increments property of fBm, i.e., $w[n]$ is a stationary process. The (i,j)-th entry of P_w is equal to $r[i-j]$, where

r is defined in Eq. (7.6). Note that $E[x[m]w[m+m_1+n]]$ is zero only for $H = 1/2$, and for $H \neq 1/2$ it is a function of m. A similar observation applies to the first column of Eq. (7.49). However, as demonstrated in the following paragraph, the Canonical Correlations decomposition based on this cross-covariance varies very little with m for $m \gg m_1$.

The matrix T_1 returned by Canonical Correlations will essentially be a function of the cross covariance $E[y_m Y_m^T]$ given in Eq. (7.49). However, recall that Canonical Correlations first normalizes by the standard deviation of each element in x_m and X_m, so that T_1 is invariant to any scaling of $x[m]$ or $x[m+m_1]$. If we divide the first element of y_m by the standard deviations of $x[m]$ and divide the first element of Y_m by the standard deviations of $x[m+m_1]$, then the $(1,1)$ entry of the covariance matrix in Eq. (7.49) is

$$\frac{E[x[m]x[m+m_1]]}{\mathrm{var}[x[m]]^{1/2}\mathrm{var}[x[m+m_1]]^{1/2}} = \frac{\frac{1}{2}(|m|^{2H} + |m+m_1|^{2H} - |m_1|^{2H})}{|m|^H |m+m_1|^H}. \tag{7.50}$$

For $m \gg m_1$ and $0 < H < 1$, this term tends to one. The other terms along the first row of $E[y_m Y_m^T]$ are

$$\begin{aligned}\frac{E[x[m]w[m+n]]}{\mathrm{var}[x[m]]^{1/2}} &= \frac{E[x[m]x[m+n]] - E[x[m]x[m+n-1]]}{\mathrm{var}[x[m]]^{1/2}}, \\ &= \frac{1}{2}\left[\left|\frac{(m+n)^2}{m}\right|^H - \left|\frac{(m+n-1)^2}{m}\right|^H - \left|\frac{n^2}{m}\right|^H + \left|\frac{(n-1)^2}{m}\right|^H\right].\end{aligned}$$

for $m_1 + 1 \leq n \leq 2m_1 - 1$. For $m \gg m_1$ and $0 < H < 1$, these terms all tend to zero. Similar analysis shows that of the terms

$$\frac{E[x[m+m_1]w[m+n]]}{\mathrm{var}[x[m+m_1]]^{1/2}} \tag{7.52}$$

also tend to zero for increasing m. Hence for large m relative to the size of the interval m_1, we have the following approximation

$$E[y_m Y_m^T] \approx \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & P_w \end{array}\right]. \tag{7.53}$$

Because this matrix is independent of the value of m, the matrix T_1 returned by Canonical Correlations will be independent of the location of the interval $[m, m+m_1-1]$. This independence can be used to determine internal variables at a given scale from other internal variables at a given scale. In fact, as shown in the following examples, all of the internal variables at a given scale can be derived from a single internal variable.

Algorithms for Approximating fBm

Theorem 4, along with Eq. (7.40), shows in principle how any internal variable at scale m can be used to derive one of the internal variables at scale $m+1$, provided the mapping of fBm to the finest scale is done appropriately. In reality, one must also account for

- the discrete nature of the finest scale process and
- the finite interval represented at the finest scale.

The discrete nature of the finest-scale process can be handled using an approximation based on the exact relationship between internal variables at neighboring scales when the finest-scale process represents local averages of $x(t)$—see Eq. (7.45). The finite interval is handled by assuming that all of the internal variables at a given scale are given by the same internal matrix.

To describe an algorithm for realizing multiscale approximations of fBm, consider modeling N samples of fBm on the interval $(0,T]$ at the finest scale of a binary tree. If we assume $N = 2^{M+2}$, then we can use an M scale tree with four samples represented at each finest-scale node. (See Figure 7.13 for an example of the finest-scale mapping when $M = 2$.) The sampling interval is $\Delta t = T/N$. We would like to be able to realize the corresponding multiscale model using a minimum number of Canonical Correlations. This number should also be independent of N. An algorithm requiring just two Canonical Correlations decompositions is given by considering and interval of length T embedded within the larger interval $[0, 2T]$, and then using shift-invariance to derive a multiscale representation of the process on $[0,T]$. The two Canonical Correlations required are the following:

- compute the linear combination $T_1 f_1$ that maximally decorrelates samples of $x(t)$ on the interval $(T/2, T]$ from those on $(0, T/2] \cup (T, 2T]$, where f_1 contains the samples on $(T/2, T]$, and
- compute the linear combination $T_2 f_2$ that maximally decorrelates samples of $x(t)$ on the interval $(T, 3T/2]$ from those on $(0, T] \cup (3T/2, 2T]$, where f_2 contains the samples on $(T, 3T/2]$.

Note that we have chosen two intervals, $(T/2, T]$ and $(T, 3T/2]$, that are sufficiently far from the boundaries of the interval entire $[0, 2T]$. The reason is that most of the internal variables in the tree must decorrelate some interval $(t_1, t_2]$ from $(t_1, t_2]^c = (0, t_1] \cup (t_2, T]$, where neither $(0, t_1]$ nor $(t_2, T]$ is empty. (One of these intervals will be empty only when the finest-scale descendent of a node includes either $x(\Delta t)$ or $x(T)$.) Thus, using both shift-invariance and self-similarity, all of the internal variables can be reasonably approximated from $T_1 f_1$ and $T_2 f_2$. To be more specific, the variable at the root node,

using shift-invariance, is given by

$$z(0) = \underbrace{\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}}_{W_0} \begin{bmatrix} f_{0\alpha_1} \\ f_{0\alpha_2} \end{bmatrix}. \tag{7.54}$$

The variables at finer scales can be derived from W_0 using Eq. (7.45) and shift-invariance.

The only drawback of this algorithm is that, because the number of computations required for a Canonical Correlations decomposition grows cubicly with the number of samples to be decorrelated, doubling the size of the interval from $(0,T]$ to $(0,2T]$ increases the number of computations for a single decomposition by eight. A more efficient algorithm is the following:

1. Compute $z(0)$ as usual, i.e., use Canonical Correlations to decorrelate samples of $x(t)$ on $(0,T/2]$ from those on $(T/2,T]$. This requires one application of Canonical Correlations to the N-by-N covariance matrix P_f.

2. Compute $z(0\alpha_2)$ as usual, which requires two applications of Canonical Correlations to P_f. Use shift-invariance to compute $z(0\alpha_1)$.

3. Compute the internal variable at the "third" node of scale $m = 2$, i.e., the node s whose finest-scale descendents represent the interval $(T/2,3T/4]$. Two applications of Canonical Correlations yield an internal variable of the form

$$z(s) = \underbrace{\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}}_{W_s} \begin{bmatrix} f_{s\alpha_1} \\ f_{s\alpha_2} \end{bmatrix}. \tag{7.55}$$

4. The remaining variables at scale two follow from shift-invariance. The variables at finer scales can be derived from W_s using Eq. (7.45) together with shift-invariance at each scale.

Note that this algorithm requires five applications of the Canonical Correlations to an N-by-N covariance matrix rather than two applications to a $2N$-by-$2N$ covariance matrix. The model parameters follow from Eqs. (2.30) and (2.32).

Example Multiscale Models

To justify Eq. (7.45) and the assumption of shift-invariance, we can compare the multiscale model produced by this algorithm to that produced by the "myopic" Canonical Correlations algorithm, which computes every internal variable independently. For $\sigma^2 = 1$ and the unit interval ($T = 1$), the covariance errors provided by these two approaches are compared in Figure 7.17. The dimensions of the states for all the models illustrated in Figure 7.17 are six, except for the finest-scale states which have dimension four. An interesting result is that the "efficient" realization algorithm, which

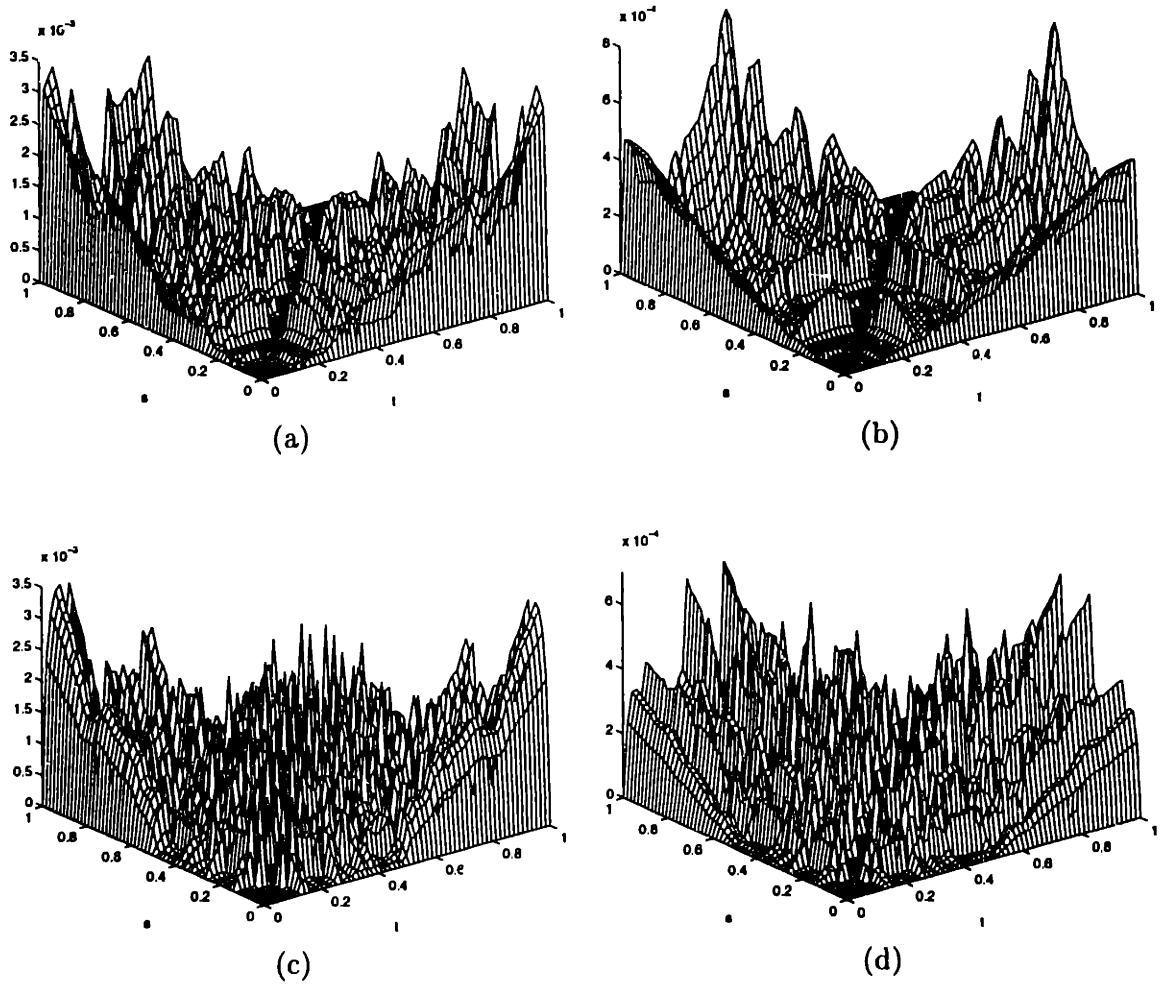
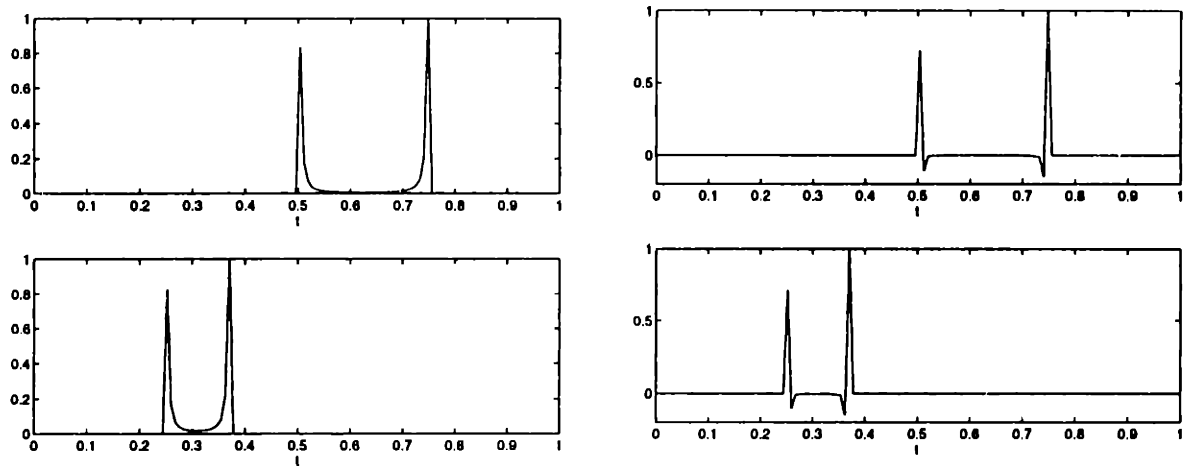


Figure 7.17. The absolute value of the difference between Eq. (7.4) and the covariance of the finest-scale process produced by the "myopic" Canonical Correlations realization for (a) $H = 0.3$ and (b) $H = 0.7$; the absolute value of the errors in the covariance of the finest-scale of the "efficient algorithm" for (c) $H = 0.3$ and (d) $H = 0.7$. The dimension of the states not at the finest scale, for both models, is six.

uses Canonical Correlations to compute directly only three internal variables, compares favorably with the "myopic" realization algorithm. Namely, the errors illustrated in Figures 7.17c and 7.17d are roughly equal in magnitude to the errors illustrated in Figures 7.17a and 7.17b, which are the covariance errors at the finest scale of the "myopic" model. Remember that the errors along the diagonal are zero for both models, since the variance errors are always zero for any model computed from Eqs. (2.30) and (2.32).

Recall that the algorithm we have just described uses three Canonical Correlations, plus the approximate self-similarity and stationarity of the matrices W_s, to determine the states at each scale. The reason that our algorithm does so well compared to

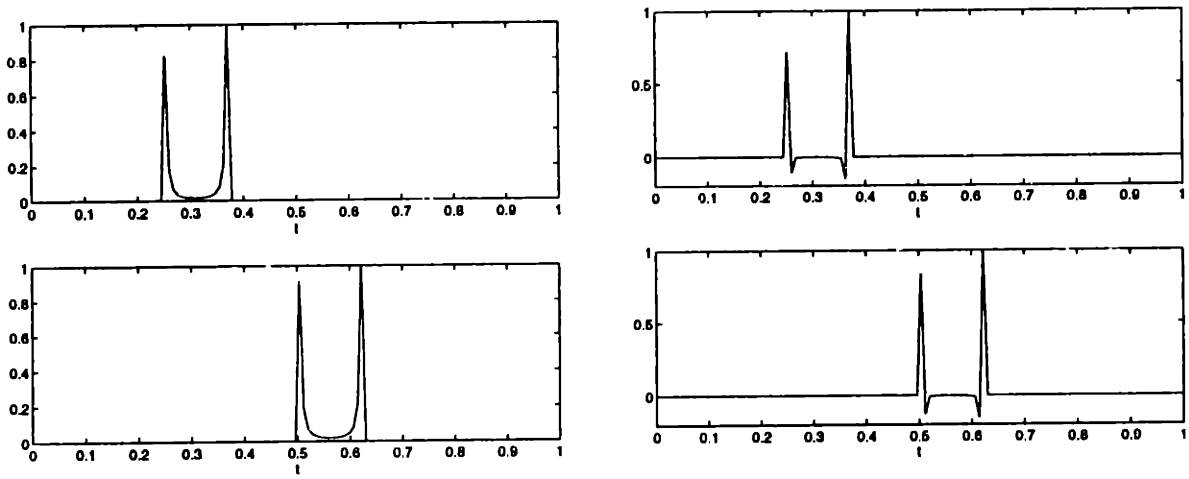


(a) (b)

Figure 7.18. For (a) $H = 0.3$ and (b) $H = 0.7$, the linear functional of the finest-scale process that maximally decorrelates (top) $(T/2, 3T/4]$ from $(T/2, 3T/4]^c$ and (bottom) $(T/4, 3T/8]$ from $(T/4, 3T/8]^c$.

exhaustive Canonical Correlations is that the functionals returned by exact Canonical Correlations very closely follow the approximation we are making. Specifically, examples illustrating the near self-similarity and shift-invariance of the linear functionals returned by Canonical Correlations are illustrated in Figures 7.18 and 7.19, respectively. First consider self-similarity. The variable $z(0\alpha_2)$ at scale $m = 1$ must decorrelate samples of $x(t)$ on the interval $(T/2, 3T/4]$ from those on $(T/2, 3T/4]^c$. There is a corresponding internal variable at scale $m = 2$ that decorrelates samples on $(T/4, 3T/8]$ from samples on $(T/4, 3T/8]^c$. The linear functional $g^T f$, where f is a vector representing the finest scale process, that maximally decorrelates samples on $(T/2, 3T/4]$ from those on $(T/2, 3T/4]^c$ is illustrated in the top of Figure 7.18a for $H = 0.3$. (Note that g, not $g^T f$ is plotted.) The plot in the bottom of Figure 7.18a is the linear functional that maximally decorrelates samples on $(T/4, 3T/8]$ from those on $(T/4, 3T/8]^c$. The analogous linear functionals for $H = 0.7$ are plotted in Figure 7.18b. For both values of H, the linear functionals are essentially related by a compression of the time-axis by a factor of two. To check the "shift-invariance" assumption, we can compare the linear functional that decorrelates $(T/4, 3T/8]$ from $(T/4, 3T/8]^c$ to the linear functional that decorrelates $(T/2, 5T/8]$ from $(T/2, 5T/8]^c$. These two functionals are compared in Figures 7.19a and 7.19b for $H = 0.3$ and $H = 0.7$, respectively. For both values of H, the two functionals are very closely related by a shift of the time axis, which justifies our assumption of shift-invariance.

It is worthwhile to note that the internal variables of any multiscale model based on Canonical Correlations change form with the value of H, which is not true for the displacement and wavelet multiscale models proposed earlier. Also, the first few



(a) (b)

Figure 7.19. For (a) $H = 0.3$ and (b) $H = 0.7$, the linear functional of the finest-scale process that maximally decorrelates (top) $(T/2, 3T/4]$ from $(T/2, 3T/4]^c$ and (bottom) $(T/4, 3T/8]$ from $(T/4, 3T/8]^c$.

elements of the internal variables for the Canonical Correlations realization are similar to the variables of the displacement model in that the functions are weighted heavily at the endpoints of the intervals, yet they are also nonlocal, as are the wavelets.

Another interesting comparison is between Figures 7.17c-d and Figures 7.14a-b. Figures 7.14a-b display the covariance errors at the finest scale of the endpoint-average model with states consisting of four samples and two wavelet coefficients; thus, the state dimensions of the two models are the same. However, the covariance at the finest scale of the multiscale model produced by the "efficient" algorithm is more accurate than that produced by the endpoint-average model for both $H = 0.3$ and $H = 0.7$. The covariance errors for the "efficient" model are about an order of magnitude less than those for the endpoint-average models. (Note that the vertical axes in Figures 7.14 and 7.17 are not the same.) This demonstrates that the models produced by the "efficient" algorithm provide more accurate approximations of fBm than those derived from less rigorous justifications for augmenting the state dimensions.

Sample paths for the six-dimensional "efficient" models are illustrated in Figure 7.20 for $H = 0.3$ and $H = 0.7$. These sample paths have no visibly discernible artifacts.

These examples illustrate that the self-similarity and stationary increments properties of fBm can indeed be exploited for the development of internal multiscale models. The proposed algorithm provides a systematic approach to realizing multiscale models that approximate the statistics of fBm within any degree of accuracy. The only drawback is that a Canonical Correlations decomposition of an N-by-N covariance matrix is still required, which implies $\mathcal{O}(N^3)$ computations. However, we are not necessarily restricted to Canonical Correlations decompositions, nor is $\overline{\rho}(\cdot)$ the only correlation

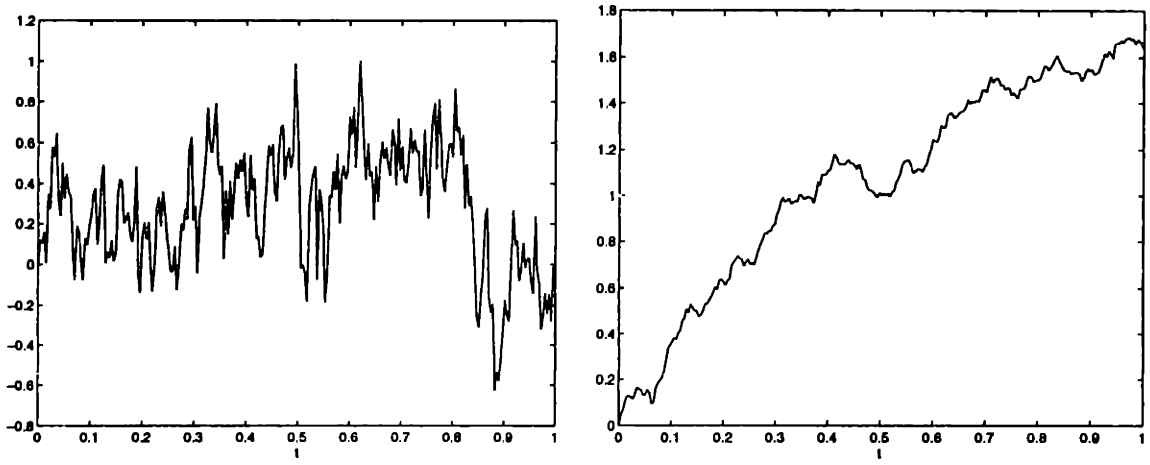


Figure 7.20. Sample paths produced by the multiscale models based on Eq. (7.45) and shift-invariance for (a) $H = 0.3$ and (b) $H = 0.7$.

measure that is suitable for self-similar processes, i.e., there are other correlation measures that "satisfy" Theorem 4. Also, we can likely use interpolation to derive internal variables at coarser scales from those at finer scales, leading to more efficient algorithms. These issues, as well as the development of multiscale models for the broader class of self-similar processes, are briefly discussed in Chapter 8.

Chapter 8

Contributions, Limitations, and Potential Solutions

The first section of this chapter summarizes the major contributions of this thesis. In Section 8.2, we point out the limitations of our approaches, the problems that must be addressed, and some possible solutions. Finally, alternative approaches to multiscale realizations are described in Section 8.3.

■ 8.1 Summary of Contributions

Multiscale Modeling of Nonlocal Properties

One of the major contributions of this thesis is the development of a methodology for realizing multiscale stochastic processes that not only have a desired statistical structure at the finest scale, but also represent particular nonlocal functions of the random phenomenon at coarser scales of the multiscale process. Before this thesis, the coarser scale variables of multiscale tree processes were chosen primarily to allow efficient processing of the finest-scale variables. However, because the multiscale estimator and likelihood calculator can incorporate observations at all scales on the tree, and because the multiscale estimator produces by default the LLSE estimate and error variance of every variable on the tree, there is sufficient motivation for representing particular nonlocal functions of the phenomenon of interest at the coarser scale nodes. Doing so allows one to incorporate measurements made at different resolutions, and to estimate and produce error variances for aggregate properties of the phenomenon.

The first approach presented was a rather straightforward extension of the realization algorithm presented in [51]. While the internal (or state) variables used in [51] are the linear functions that decorrelate subsets of the finest-scale process, the internal variables produced by the algorithm of Section 4.2 also decorrelate the particular coarse-scale or nonlocal functions that are to be represented at coarse scales of the tree. If the state variables are computed independently using Canonical Correlations, the number of computations required by this algorithm will be overwhelming for large-sized problems. However, the basic principles behind this approach apply to any multiscale realization, such as the augmentation algorithm described in Section 4.3 or

the fine-to-coarse algorithm outlined in Section 8.3.

To overcome the computational burdens associated with Canonical Correlations, we presented a method that augments the states of internal multiscale models with the desired nonlocal functions of the finest-scale process. The idea is basically to build on multiscale models, like the multiscale models for fBm described in Chapter 7, that already represent the finest-scale statistics of the phenomenon of interest. The state augmentation is not straightforward, i.e., there is more involved than just adding the desired nonlocal functions to the corresponding coarse-scale variables. For one, the Markov property must be preserved. Secondly, the descendents of the coarse-scale variable that is augmented must also be augmented to ensure that the nonlocal functions generated at the coarse-scale node are passed consistently to the finest-scale process. Once the state variables have been augmented, the parameters of the augmented multiscale model can be computed using Eqs. (2.30) and (2.32).

An interesting feature of the state augmentation algorithm is that *any* linear function of the finest-scale process can be added to the variable at any node of the internal multiscale model. The natural question is, given a set of nonlocal properties to be measured or estimated within the multiscale framework, at which nodes should these functions be represented. The obvious answer is the set of nodes that, after the augmented model has been computed, leads to the most efficient processing algorithms. But this does not tell us how to choose the optimal set of nodes. An example was provided to illustrate that the choice of nodes at which the nonlocal functions are placed does effect the state dimension of the augmented model, but no general solution is presented. The problem is that the nodes at which the nonlocal functions are placed will have the most significant impact (on the state dimensions of the augmented model) when approximations can be tolerated.

The real power of the multiscale framework is based on efficient processing algorithms. These algorithms are functions of the state dimensions of the multiscale tree; for example, the number of computations required by the multiscale estimator grows cubicly with each state dimension. If a large number of nonlocal functions are added to the variables at coarse scales of the process, the increased state dimensions will destroy the utility of the multiscale framework. In these cases, one should be willing to sacrifice statistical accuracy for computational efficiency. In fact, for many problems one should be able to significantly reduce the state dimensions before incurring meaningful errors in the statistics of the model. In this spirit, we proposed an approximate state augmentation algorithm based on compression and consistency. The compression involves removing the elements of each state variable deemed to be least significant. Consistency requires that any elements removed from the fine-scale variables should also be removed from the coarser-scale variables, since these elements cannot be consistently passed from the coarser-scale variables to the finest scale of the process. This consistency requirement implies that the approximation algorithm should proceed from fine to coarse scales. In fact, we will return to consistency in Section 8.3, showing how it can form the heart of an efficient fine-to-coarse realization algorithm.

Multiscale Modeling and Estimation of Hydraulic Conductivity

The second contribution of this thesis is to apply the multiscale framework to a non-trivial data fusion problem in which the measurements are provided at different spatial resolutions. For our example, we considered the estimation of hydraulic conductivity in 2D from measurements of head, conductivity, and travel times. The basic approach is to linearize the head and travel-time measurements about some log-conductivity function, and then to represent the observed head and travel-time values as variables at coarser scales on the tree. For the examples provided in this thesis, samples of log-conductivity are represented at the finest scale of a multiscale tree and the finest-scale process is a Markov Random Field. Head and travel-time measurements are added to the coarse-scale variables using the algorithm of Section 4.3.

Before incorporating travel-time measurements, we first focused on the estimation of conductivity from head and conductivity measurements only. The first step was to analyze the effect of head measurements on conductivity estimates. The effectiveness of the head measurements in constraining the conductivity function depends in large part on the the particular flow scenario, i.e., the form and uncertainty of the boundary conditions, the locations of the head measurements, and whether the aquifer is forced by a known function Q, e.g., a pumping or injection well. The existence of a pumping well was shown to significantly improve the effectiveness of the head measurements in regions where the pumping governs flow behavior. The signal-to-noise ratio of the head measurements, i.e., the ratio of the variance of the head function to the variance of the head measurement errors, is also a significant determinant of the effectiveness of the head measurements.

Next, the multiscale framework was applied to two hydraulic conductivity estimation problems. For both problems, the head measurements were linearized about the conductivity function $f_0 = 0$. For the first problem, the head and conductivity measurements were at the same twenty locations. The state augmentation algorithm was used to represent the head measurements at coarse scales of the tree, and the augmented multiscale tree incurred modest increases in the state dimensions—about forty at the two coarsest scales. Note that, since the head measurements are based on a linear approximation, the augmented multiscale model only approximates the cross-covariance between the head measurements and the log-conductivity samples at finest scale. Consequently, the estimate and error variances produced by the multiscale estimator are approximations of the optimal LLSE estimator. Nevertheless, in spite of the linearization, incorporating the head measurements in addition to the conductivity measurements generally leads to a decrease in the log-conductivity estimation error. For the second example, the head and conductivity function were measured at separate locations. These measurements included regions of both dense conductivity and dense head samples. The estimate of the finest scale of the multiscale tree, however, only has fine-scale detail in regions in which there are dense conductivity measurements. This example illustrates that head measurements reduce the estimation error, but provide only coarse-resolution information about the conductivity function.

The linearized head measurement equations can be improved by linearizing about estimates of the log-conductivity function rather than $f_0 = 0$. This leads to an iterative algorithm in which the head measurements are successively linearized about the most recent log-conductivity estimate. While this iterative algorithm generally leads to improved log-conductivity estimates, the cost is increased computational complexity. Namely, at each iteration the Fréchét derivatives must be computed for a new conductivity function. This implies $L_h + 1$ forward simulations of the groundwater flow equation for L_h head measurements. These simulations can be quite costly if the number parameters used to describe the conductivity function is large. (In these examples, the number of parameters is equal to the number of conductivity samples represented at the finest scale of tree.) Also, the augmentation algorithm must be reapplied with each new set of linearized head measurements. This repeated application can also lead to a significant increase in the number of computations. To determine the true costs of the iteration, one must answer how quickly the log-conductivity estimate converges to its final value and by how much the asymptotic value is an improvement over the estimate computed from a linearization about $f_0 = 0$. As a general rule, we show that the estimate converges more slowly for large log-conductivity variances, but that large-conductivity variances also lead to greater estimation error reductions due to re-linearization.

Using a procedure very similar to the linearization of head measurements provided in Chapter 3, we showed how measurements of travel-time to a control plane can be linearized in terms of perturbations of log-conductivity. This linearization allows allows the incorporation of travel time measurements within the multiscale framework, so that conductivity can be estimated from measurements of head, conductivity, and travel times. An example of such an estimator was provided. For some applications, travel time measurements are not available, and one is interested in estimating a conditional distribution for travel times in the aquifer. In this case the multiscale framework provides two methods for computing distributions of travel times that are conditioned on head and conductivity measurements. One is a Monte-Carlo method, which involves using the multiscale error model to compute numerous conditional simulations of the log-conductivity function. The advantage of this method is that it does not depend on the linearization of travel time, but it also requires numerous implementations of the 2D flow equation. The second approach is to build the travel-time variable directly into the multiscale model, just as we did for the multiscale model that incorporates travel time measurements. The key to this approach is to be able to describe travel time as a linear function of hydraulic conductivity. If travel time is then represented as a coarse-scale variable of the multiscale process, the multiscale estimator will produce both the estimate and error variance of travel time, i.e., the conditional distribution assuming that travel time is a Gaussian random variable. Both approaches were demonstrated and compared, and the deviation between the two is largest for large variances in the conditional distribution for log-conductivity, since the travel-time linearization breaks down for large conductivity variances. Also, for large conductivity variances, the con-

ditional distribution of travel time is no longer approximately Gaussian, but is better represented as log-normal random variable.

Multiscale Modeling of Fractional Brownian Motion

The third major contribution of this thesis is the construction of a class of multiscale models that approximate fractional Brownian motion (fBm). The first set of approximate models is based on the random midpoint displacement and wavelet synthesis algorithms for synthesizing fBm. The midpoint displacement algorithm (wavelet synthesis) makes the implicit assumption that the displacements (wavelet detail coefficients) are completely uncorrelated. However, the displacements and detail coefficients are correlated, and the correlations are most strong among local values. For instance, any detail coefficient is most strongly correlated with those coefficients that have common support and are at neighboring scales and time indices. By capturing these correlations within the multiscale framework, not only is fBm more accurately represented, but one can also take advantage of the efficient processing and synthesis algorithms. Also, because multiscale models are nonstationary, the multiscale autoregression can account for the fact that the interpolated value at the midpoint of two samples is not necessarily the average of these two samples. (This observation has implications only for the multiscale model based on random midpoint displacement.) The interpolation depends on the distance between the samples, the location of the samples, and the value of the Hurst exponent. The two resulting multiscale approximations of fBm were shown to be quite accurate, especially considering that the state dimensions are equal to two for the wavelet-based multiscale model and three for the midpoint-displacement-based multiscale model.

The second approach to modeling fBm is more systematic, and also uncovered some deeper issues associated with multiscale modeling. This approach is based on a deeper understanding of the implications of statistical self-similarity and stationary increments in the context of multiscale models. (Recall that fractional Brownian motion is both statistically self-similar and has stationary increments.) When the process to be modeled at the finest scale of a multiscale tree is self-similar, we showed that, in the absence of discretization effects, the linear functionals that define each variable of the resulting multiscale tree are related to linear functionals at finer scales by compressions of the "time" axis. Thus, the variables at any scale of the tree can be determined from the variables at either coarser or finer scales by expansions and compressions of the time axis. Also, if the stationary increments property of the finest-scale process is invoked, the linear functionals defining the variables at any scale can be approximated by the linear functionals of a single variable at that scale. These results were used to efficiently compute an approximate multiscale model for fBm that is very accurate, even when the dimensions of the state variables are small.

■ 8.2 Limitations and Problems to be Addressed

We now provide some commentary on the research just summarized. These comments are meant to serve two primary purposes. First and foremost, they will hopefully make the reader aware of any difficulties that may arise when implementing the methods described in this thesis. Secondly, we suggest areas that require more investigation, and perhaps entirely new ways of approaching the problems. Like the previous section, this section is organized according to the three main contributions of the thesis.

Multiscale Modeling and Nonlocal Properties

Two approaches were presented for constructing multiscale models that represent particular nonlocal functions at coarser scales. The first, described in Section 4.2, is completely general, i.e., places no restrictions on the statistics of the multiscale process. However, the current implementation of this algorithm is based on Canonical Correlations, meaning that it is computationally infeasible for large problems. The alternative approach provided in Section 4.3 is based on augmenting internal multiscale models whose internal variables are known or can be easily computed. While more efficient than the general approach, there are still some limitations that should be addressed.

- The number of classes of random fields and processes for which the internal variables are known or can be computed easily is still limited primarily to Gauss-Markov Random Fields and $1/f$-like processes.
- The effectiveness of the state augmentation depends on the computation and storage costs associated with implementing Eq. (4.14).

The first limitation is a manifestation of a more fundamental issue—the need to investigate what classes of stochastic processes are naturally modeled by auto-regressions in scale. For example, what class of processes can be modeled by third order binary tree models, and what are the properties of these models. We saw in Chapter 7 that fractional Brownian motion is well approximated by a number of different multiscale models with state dimensions less than or equal to three. A surprising result is that the state variables of these models can vary quite widely, from the samples used by the midpoint deflection model to the local averages used by the wavelet model. This diversity would lead one to believe that there is considerable flexibility in the multiscale models that can realize a particular statistical structure, even when the state dimensions are fixed. A related question is what growth in the state dimensions of the coarsest scale variables is required in order to model random fields that have relatively smooth covariance functions. The multiscale models for MRFs used throughout this thesis are not appropriate for very large problems, since the dimensions of the states at coarser scales grow linearly with the linear dimension of the domain of interest. The MRF models were used in this thesis primarily as a vehicle to illustrate our methods.

The difficulty in implementing Eq. (4.14) is computing the covariance matrices $P_{\zeta(s)}$ and $P_{\zeta(s)\zeta(s\overline{\gamma})}$. For the augmented variable $\zeta(s)$, the covariance is given by

$$P_{\zeta(s)} = \mathcal{V}_s P_f \mathcal{V}_s^T , \tag{8.1}$$

Note that $\zeta(s) = \mathcal{V}_s f$ contains[1] both the original internal variables and the nonlocal functionals to be represented at node s or its ancestors. The problem with computing $P_{\zeta(s)}$ is twofold. First, for large-sized random fields, P_f cannot be stored explicitly. Second, for the components of $\zeta(s)$ corresponding to nonlocal functions of f, the number of computations required to compute the corresponding elements of $P_{\zeta(s)}$ can be very large. Both of the problems are overcome if the finest-scale process is stationary. For example, even if every element of $\zeta(s)$ is nonlocal, $\mathcal{V}_s P_f$ can be computed in $\mathcal{O}(d(s)N_f \log N_f)$ computations using an FFT approximation, where $d(s)$ is the dimension of $\zeta(s)$ and N_f is the dimension of f. The result will require $d(s)N_f$ storage elements, and then an additional $d(s)^2 N_f$ computations to compute $P_{\zeta(s)}$. Similarly, the cross-covariance $P_{\zeta(s)\zeta(s\overline{\gamma})}$ will require $d(s)d(s\overline{\gamma})N_f$ additional computations. Computing the model parameters when P_f is nonstationary will likely require a different approach, tailored to the statistics of the finest-scale process.

One can also make a more fundamental criticism of any multiscale modeling framework based on Eq. (4.14) (or Eqs. (2.30) and (2.32)). Namely, the modeling is completely controlled by the statistics of the finest-scale process. A primary motivation for using the multiscale framework is to let coarser scale variables account, at least approximately, for the correlations between variables separated by large distances. Defining every variable in terms of a linear function of the finest-scale process, and computing all of the model parameters from the entire finest-scale covariance P_f, however, will likely require too much effort to exactly or approximately capture correlations over large distances. An alternative modeling approach is suggested in Section 8.3, but the bottom line is to be able to realize multiscale models without ever having to operate on the entire finest-scale covariance matrix. The $1/f$-like models proposed by [19, 34] are in this spirit, since the finest-scale covariance is never specified explicitly; instead, the process noise covariances are chosen to achieve a desired power-spectral-density.

Another issue not fully addressed in the development of the state augmentation algorithm is the placement of the nonlocal functions. The problem is to choose the set of nodes at which the nonlocal functions will be represented that minimizes the resulting increase in estimator complexity. If the nonlocal functions are simple summations of local averages of the finest-scale process, as for the nonlocal functions in the example of Section 4.3.5, the optimal choice of nodes can be chosen. The reason is that all of these functionals can be expressed in terms of small number of localized basis elements. The state augmentation can then be expressed in terms of the simpler problem of representing each of these local basis elements. For more general functions, such as the linearized head measurements and travel-time measurements of Chapters 5 and 6, the choice of nodes at which to represent these functions is no longer obvious. For the

[1]Recall that f in this context is the vector representing the process at the finest scale of the tree.

example of Section 5.2.1, the head measurements were placed at the nodes for which the finest-scale descendents of that node represent the log-conductivity perturbations that account for most of the variation in the head measurement. (The linearized head sample $\delta h(x_i)$ is most sensitive to conductivity perturbations in a neighborhood centered about x_i.) This choice works well when most of the head measurements are located away from the boundaries partitioned by the coarser scale nodes. For instance, if all the head measurements were on the quadrant boundaries partitioned by the root node, then all of these measurements would be represented at the root node. Similarly, this method will also place travel-time measurements at the root node when the starting point $x(0)$ and the control plane are in different quadrants of the aquifer. Representing all of the nonlocal measurements at the root node, or at the very coarse-scale nodes, will lead to an unacceptably large increase in the dimensions of the coarse scale variables. One possible solution, when log-conductivity is modeled as an MRF, is to adjust the boundaries partitioned by the coarser scale nodes according to the locations of the measurements.

One reason why the problem of placing the nonlocal functionals was not fully addressed in this thesis is that the placement of the nonlocal functions will have the most significant impact on the complexity of the resulting multiscale estimator when an approximate multiscale model is realized. Namely, when the state augmentation does not have to be exact, there is considerably more flexibility in the linear functions to be represented at the individual state variables. In this case, the root node might contain a few nonlocal functions of the finest-scale process that provide coarse approximations to the nonlocal measurements. A method for constructing approximate models from the state augmentation algorithm was provided in Section 4.4. This approximation is based on discarding the information in $\zeta(s)$ that is least valuable in terms of predicting its children. This algorithm also ensures that the resulting multiscale model is consistent, i.e., every element of the state at node s must be used in the prediction of its children. While this approximation is likely very effective at reducing the state dimensions of the resulting multiscale model and simultaneously controlling modeling errors, there are still two significant problems that should be addressed. First, the full covariance matrices $P_{\zeta(s)}$ and $P_{\zeta(s)\zeta(s\overline{\gamma})}$ must be computed, no matter how large the dimensions of the augmented variables, before the approximation is made. Secondly, the approximation is not necessarily optimal in terms of minimizing estimation errors. The ultimate goal of the multiscale model should be to represent as accurately as possible the cross-covariances between the variables to be estimated and the variables that are measured, since these cross-covariances are the only information required by LLSE estimators. These considerations should somehow be reflected in any approximate multiscale realization used for estimation.

Multiscale Modeling for Data Fusion and Estimation of Hydraulic Conductivity

In Chapters 5 and 6, a number of examples were presented of the estimation of conductivity from measurements of conductivity, head, and travel time. For all of these examples, log-conductivity was modeled as a multiscale stochastic process, and the nonlocal head and travel-time measurements were added to coarser-scale variables of the tree using the state augmentation algorithm of Section 4.3. These examples were limited in the sense that

- the finest-scale log-conductivity function in each case was a Markov Random Field, and
- the joint (second-order) statistics of the finest-scale process and the nonlocal functions were modeled exactly.

The problem with exact models is that they cannot be applied to large problems, unless the log-conductivity function is naturally modeled by a multiscale model with small state dimensions. For exact multiscale models of Markov Random Fields, the number of computations required by the multiscale estimator grows cubicly with the linear dimension of the finest-scale process, and the number of storage elements required for the model parameters grows quadratically. Since the prior model is usually an abstraction and approximation of the true variation and uncertainty in the phenomenon of interest, there is sufficient motivation to accept an approximate model. Also, there is no compelling reason to exactly model the cross-covariance between a head or travel-time perturbation and log-conductivity samples at distant locations. The multiscale framework appears to be well-suited to approximating the joint-statistics of log-conductivity and measurements made at multiple resolutions, and the application of such approximate models deserves a thorough investigation.

Another intriguing property of the multiscale framework is the ability to produce estimates of log-conductivity that have spatially varying resolution. For most of the applications of the multiscale framework that preceded this work, the measurements are dense and located only at the finest scale. However, for groundwater field experiments, the available data are typically very sparse, irregularly distributed in space, and at multiple resolutions. From these measurements, an estimate of conductivity with uniform spatial resolution is usually produced, meaning that either an inordinately large number of parameters is estimated that cannot be justified by the data, or the estimate is unnecessarily smooth in regions of fine-scale measurements. However, because the multiscale estimator produces estimates of the tree process at every scale, an estimate with spatially varying resolution can be selected. The resolution can be distributed according to the resolution supplied locally by the measurements. Consider the estimate of log-conductivity, illustrated in Figure 5.16, from the head and conductivity measurements at the locations in Figure 5.15b. The estimate has fine-resolution detail only in the regions where dense conductivity measurements are supplied. In between these

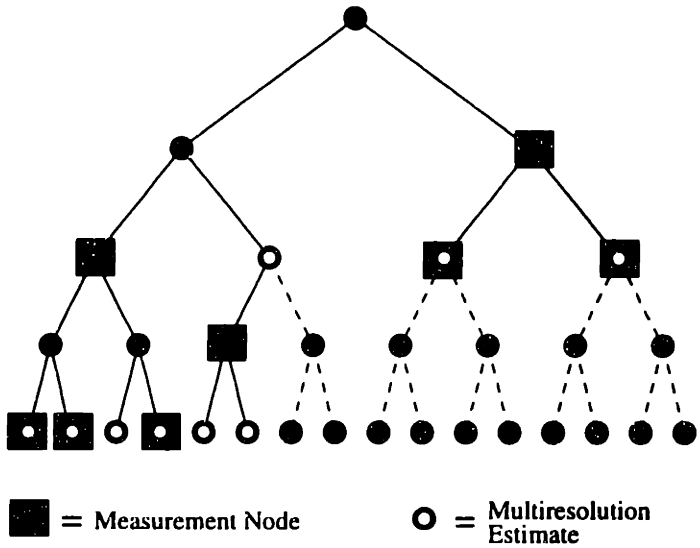


Figure 8.1. A tree model used to incorporate sparse, multiresolution measurements. The nodes that are measured are indicated by shaded boxes. The dashed lines indicate branches of the tree that do not need to be modeled if one is interested in an estimate with spatially varying resolution.

regions, because the estimate of the finest-scale process is smooth, a coarser-resolution estimate can be justified.

The advantage of a multiscale estimate of conductivity is that computation and storage costs can be reduced without significantly affecting the fidelity of the estimate. For the examples provided in Chapters 5 and 6, the finest-scale parameterization of log-conductivity was used as the linearization point for the head and travel-time measurements. Because the number of computations required to implement the 2D flow equation grows cubicly with the number parameters used for the conductivity function, a multiscale parameterization can significantly reduce these costs. A multiresolution parameterization of conductivity can also reduce the number of computations and storage elements required for the multiscale modeling and the multiscale estimator. The idea is to prune the multiscale tree by eliminating the subtree that descends from each variable that is to be included in the multiresolution estimate. For instance, consider the tree illustrated in Figure 8.1. The shaded squares indicate the locations of measurements. The dashed lines indicate branches of the tree that will likely not be needed, and thus should not be represented in the model. The only problem with this approach is that one must decide *a priori*, i.e., before constructing the model, which branches should be removed. We leave this problem to future investigations.

Independent of the multiscale framework, there are a number issues that arise in the estimation of log-conductivity from conductivity, head, and travel-time measurements that should be addressed. First, the effect of uncertainty in the boundary conditions should be investigated, since the boundary conditions are rarely, if ever, known exactly. Instead, they are usually inferred from head measurements and geologic barriers. Un-

certainty in the boundary conditions should alter the contributions of both the head and travel-time measurements. Second, uncertainty in the recharge rate (Q) should also be accounted for, since ignoring the effect of recharge on the flow field can lead to significant errors in the hydraulic conductivity estimate [99]. Third, the accuracy of the error variances provided in the examples of Chapters 5 and 6 should be understood, since these variances are based on the linear approximations of the head and travel time measurements. Fourth, we would like to understand the range of log-conductivity variances for which the Gauss-Newton algorithm converges and produces a desirable log-conductivity estimate. A related problem is to consider using multiple paths for the linearization of travel time. Finally, there is the problem of selecting the parameters of the prior distribution of log-conductivity, which for our examples are the variance and correlation lengths of the Markov Random Fields. According to [23, 38, 85], the sensitivity of most inverse methods to the values of these parameters is a major source of concern. The last problem leads one to the ultimate application—the application of the multiscale framework to a problem involving real data.

Modeling Fractional Brownian Motion

One of our more fundamental motivations for the multiscale modeling of fractional Brownian motion was to exploit the self-similar structure of a tree for modeling statistically self-similar processes. The self-similarity of fBm led to Theorem 4, which in turn led to the algorithm described in Section 7.3. This algorithm produces very accurate approximations of fBm, yet requires only a few applications of the Canonical Correlations decomposition. The only drawback is that these decompositions require the SVD of essentially the entire finest-scale covariance matrix, meaning that the number of computations grows cubicly with the number of finest-scale elements. A possible solution to this growth in complexity is to compute the coarser-scale variables from the Canonical Correlations made at a fixed, finer scale. In Section 7.3, we showed how to derive the variables at any given scale from variables at coarser scales. The basic idea follows from Theorem 4, i.e., the linear functionals that define the variables at one scale are related to those at coarser scales by a compression of the time axis. Similarly, we should be able to derive the variables at coarser scales from their descendent using an expansion of the time axis. The only difficulty is that one must account for discretization effects, since the process represented at the finest-scale of the tree is discrete. Recall that the derivation of finer-scale variables from coarser-scale variables required averaging the linear functionals that make up the coarser-scale node. Analogously, because the number of finest-scale samples descending from a node increases as the scale of the node decreases (to a coarser scale), the linear functionals derived from finer-scale variables will require some interpolation. This interpolation should be the subject of future work.

For the multiscale approximations of fBm, the linear functionals that make up the state variables are related not only by compressions of the time axis, but also by shift-invariance. The variables at any fixed scale of these models are defined by the same linear functions of their finest-scale descendents. However, because the process at the

finest-scale is non-stationary, the model parameters computed from Eq. (2.32) are not constant across any given scale. An interesting question, then, is how these model parameters are related, and what statistical model is obtained by computing only a small subset of the auto-regression parameters (A_s, Q_s) at any given scale. This would not only reduce the number of computations required for implementing Eq. (2.32), but would also reduce the number of storage elements required for the model parameters.

While the algorithm of Section 7.3 leads to accurate approximations of fBm, the goal of our research was not just to model fBm. For many applications, the polynomial growth in variance and the initial condition $\mathrm{var}[x(0)] = 0$ are undesirable, and one is more interested in the properties of self-similarity and long-range dependence. For these applications, we would like to take advantage of the self-similar structure of tree models, but not necessarily to construct the models from the finest-scale covariance of fBm. The approach taken in [96] is to implicitly bandpass fBm using the wavelet synthesis. The question for the multiscale framework is how rich a class of statistically self-similar models can be constructed with low-order multiscale models. One starting point is to consider the multiscale modeling of other $1/f$ processes, e.g., fractional Gaussian noise $(-1 > H > 0)$, which is a stationary process and corresponds to the first-order difference of discrete fBm. (Note that first-order differencing is not a bandpass operation, but the attenuation of low frequencies removes the nonstationary component of fBm.)

Another important question is whether the multiscale models developed in Chapter 7 can be naturally extended to the efficient synthesis and estimation of fractal images. Two of the most common methods for synthesizing two-dimensional fractal processes are 2D extensions of the midpoint displacement algorithm and wavelet synthesis [4, 89]. These algorithms can be represented within the multiscale framework, as were the one-dimensional analogues. Also, the algorithm of Section 7.3 can be extended to modeling 2D random fields that have structure functions $f(s) = \sigma^2|s|^{2H}$, where s is a measure of the distance between to points. The important question for these models is how the dimensions of the coarser-scale variables scale with the linear dimension of the 2D region represented at the finest-scale.

■ 8.3 Alternative Approaches to Multiscale Modeling

We now return to the problem of realizing multiscale models that approximate desired statistical relationships, which was the subject of Sections 2.3.4 and 4.2. Except for the low-order multiscale models for $1/f$-like processes described in [34, 60], all of the approaches to multiscale modeling described in this thesis are controlled by the covariance of the finest-scale process, P_f. For example, even the state augmentation algorithm of Section 4.3, which represents particular nonlocal properties at coarser-scale nodes, defines the augmented state covariances $P_{\zeta(s)}$ in terms of the finest-scale covariance, i.e.,

$$P_{\zeta(s)} = \mathcal{V}_s P_f \mathcal{V}_s^T .$$

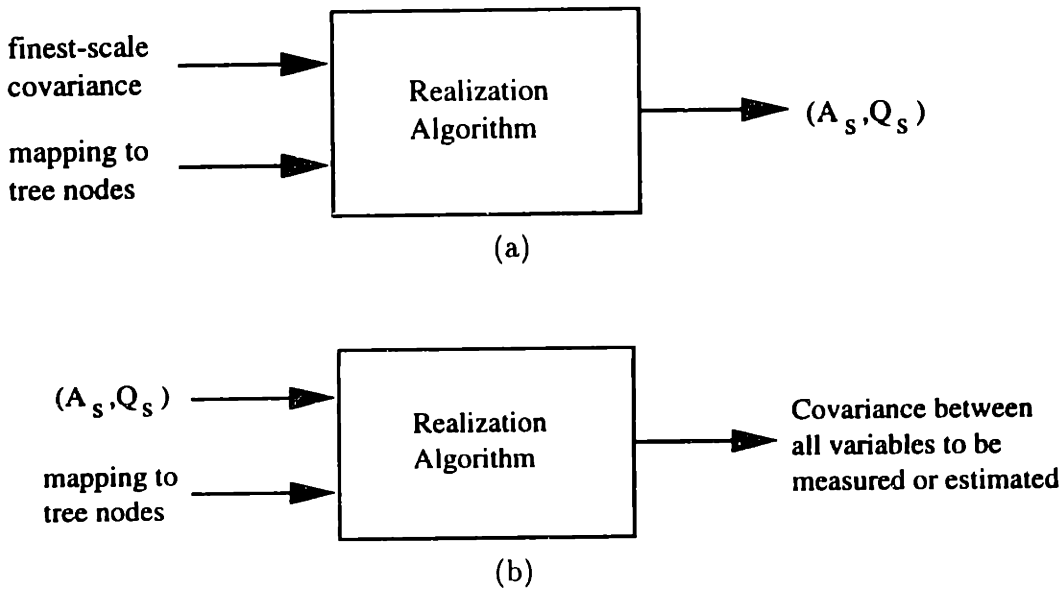


Figure 8.2. (a) A realization algorithm that accepts as inputs (i) the finest-scale covariance, P_f, (ii) the mapping of P_f to the finest-scale nodes of the tree, and (iii) the mapping of any desired nonlocal linear functions of the finest-scale process to coarser-scale variables. (b) The dual approach to the multiscale realization problem.

The reason is that the augmented coarser-scale variables are defined as linear functions of the finest-scale process, i.e., $\zeta(s) = \mathcal{V}_s f$. The general procedure of designing multiscale models from the finest-scale covariance is illustrated in Figure 8.2a.

A dual approach to multiscale modeling is to determine how the multiscale model parameters (A_s, Q_s) lead to particular cross-covariances among the state variables. This procedure is illustrated in Figure 8.2b. The advantage of this approach is that it works directly with the model parameters and therefore can greatly simplify the realization procedure. For example, the process-noise variance for the $1/f$-like processes described in [34, 60] is restricted to the form

$$Q_s = \sigma^2 2^{-\gamma m(s)}, \tag{8.2}$$

where $m(s)$ is the scale of node s. From this form for Q_s, it was shown that the finest-scale process of the resulting tree model has a power spectral density that is approximated by $1/f^{\gamma}$. In this case, providing a parametric form for the process-noise variance allows one to understand the effect of the parameter γ on the finest-scale spectrum, and also simplifies the design process if one is only interested in a multiscale model that has approximately $1/f$ spectrum. The problem in general is to understand how particular forms of the auto-regression parameters (A_s, Q_s) influence the statistics of the tree variables at the finest-scale and other scales of interest, and then to use this knowledge to design tree models that approximate the desired statistical structure.

While this dual approach to multiscale modeling is unlike any of the methods discussed in this thesis and will likely require a significant amount work to fully explore,

we can suggest a concrete alternative to the realization algorithms discussed in Sections 2.3.4 and 4.2. Recall that, for the case in which one is only interested in the statistics of the finest-scale process, each state variable $z(s)$ is set equal to the linear function of f_s—recall that f denotes the vector containing the finest-scale process and f_s is the finest-scale process descending from node s—that maximally decorrelates (with some restriction on the maximum dimension of $z(s)$) the finest-scale vectors $f_{s\alpha_1}, \ldots, f_{s\alpha_q}, f_{s^c}$. This linear function, $z(s) = W_s f_s$, can be computed using Canonical Correlations. In [49], the variable at each node is computed independently of the variables at all other nodes. There are two significant drawbacks to this approach of applying Canonical Correlations independently to each node.

- If approximation are made, they will not be done consistently.
- The Canonical Correlations decomposition effectively requires an SVD of the entire finest-scale covariance matrix for **each** variable that is calculated.

These two drawbacks can be overcome if one instead uses a fine-to-coarse realization for which

- consistency is guaranteed, i.e., the variable $z(s)$ is a linear function of its children $z(s\alpha_1), \ldots, z(s\alpha_q)$, and
- the Canonical Correlations is computed on a reduced-order covariance matrix.

This algorithm is motivated in part by the approximate state-augmentation algorithm described in Section 4.4. In this section, we showed that a fine-to-coarse sweep is essentially required to ensure that the multiscale model is consistent, i.e., that all of the information generated a coarser-scale variables is passed to the appropriate variables at finer scales. The other motivation for this algorithm can be illustrated when using Canonical Correlations to determine any variable $z(s)$ at scale $M-1$, where M is the finest scale. (Recall that the nodes at scale M are automatically determined by the mapping of f to the finest scale.) In this case, $z(s)$ conditionally decorrelates the vectors $z(s\alpha_1), \ldots, z(s\alpha_q)$ and f_{s^c}, where f_{s^c} contains all the finest-scale elements not descending from node s. The vectors $z(s\alpha_1), \ldots, z(s\alpha_q)$ will have relatively small dimensions compared to the dimension of f_{s^c}, especially if the finest-scale process has very large dimension. However, there is no need to capture the exact cross-correlations between the elements in $z(s\alpha_1), \ldots, z(s\alpha_q)$ and those in f_{s^c}, especially for those elements that are separated by large distances. Thus, the problem of determining states at scale $M-1$ becomes one of finding the linear combination of $z(s\alpha_1), \ldots, z(s\alpha_q)$ that maximally decorrelates the $q+1$ vectors

$$z(s\alpha_1),\ z(s\alpha_2),\ \ldots z(s\alpha_q),\ Lf_{s^c}$$

where L is a linear transformation that accounts only in an aggregate sense for the elements of f_{s^c} that are separated by a large distance from those represented by the

descendents of node s. If the number of rows in L is on the same order as the dimension of the vectors $z(s\alpha_1), z(s\alpha_2), \ldots, z(s\alpha_q)$, then the number of computations required for the Canonical Correlations decomposition will be greatly simplified.

Given that we have greatly simplified the calculation of the variables at scale $M-1$, the remaining question is how to do so for the variables at scale $M-2$ while also ensuring consistency. One possibility is to treat the set of variables at scale $M-1$ as the new finest-scale process with covariance $P_f^{(M-1)}$. This covariance will generally have much smaller dimension than that of P_f, and can be determined from P_f using the linear functions that define the variable at scale $M-1$. Then, each variable $z(s)$ at scale $M-2$ can be set to the linear combination of $z(s\alpha_1), z(s\alpha_2), \ldots, z(s\alpha_q)$ that maximally decorrelates $z(s\alpha_1), z(s\alpha_2), \ldots, z(s\alpha_q)$, and $LZ_{M-1}(s)$, where $Z_{M-1}(s)$ is the vector of variables at scale $M-1$ that do not descend from node s and L is a transformation that significantly reduces the dimension of $Z_{M-1}(s)$. Note that consistency is guaranteed, since each variable at scale $M-2$ is a linear combination of its descendents at scale $M-1$.

This process can be continued recursively until the root node is reached. Then the multiscale model parameters can be derived from the linear functions that define the variables at each node. This algorithm has not been implemented, and is only one of many possibilities for a fine-to-coarse realization algorithm. The questions of how well this algorithm performs and what approximations are made will be left to future research.

Appendix A

Proof of the Markov Property for Multiscale Trees

In this section, we derive the Markov property of the multiscale trees defined in Section 2.2. This property follows from the whiteness of the process noise in Eq. (2.22) and $E[z(0)w(s)^T] = 0$. (Recall we assume that $z(0)$ and $w(s)$ are zero mean.)

Proof: Define $(x \,|\, y)$ to be the LLSE error of estimating x from y, i.e.,

$$(x \,|\, y) = x - \hat{E}[x \,|\, y] \,.$$

For the purposes of this proof, it is important to note that the covariance of $(x \,|\, y)$ is equal to the covariance of x conditioned on y.

(i) To show that the variables in $S_{s\alpha_i}$ for $i = 1, \ldots, q$ are conditionally uncorrelated, first note for any node $t \in S_{s\alpha_i}$ that

$$\big(z(t) \,|\, z(s)\big) = \sum_{l \,\in\, \text{path from } s\alpha_i \text{ to } t} \Phi(t,l) w(l)$$

where $\Phi(t,l)$ is the state transition matrix from node t to node l. Thus we have

$$\big(z(t) \,|\, z(s)\big) = L_t \boldsymbol{w}_{s\alpha_i}$$

for some matrix L_t, where $\boldsymbol{w}_{s\alpha_i}$ is a vector containing the process noise $w(l)$ for all $l \in S_{s\alpha_i}$. By the whiteness of the process noise, the q vectors $\boldsymbol{w}_{s\alpha_i}$ are mutually uncorrelated, and the q subsets of states descending from s are thus conditionally uncorrelated.

(ii) Now we show that for any $t \in S_{s^c}$ and $r \in S_s$ that $\big(z(t) \,|\, z(s)\big)$ and $\big(z(r) \,|\, z(s)\big)$ are uncorrelated. Since $\big(z(r) \,|\, z(s)\big) = L_t \boldsymbol{w}_{s\alpha_i}$ for some $i = 1, \ldots, q$, we only need to demonstrate that $\big(z(t) \,|\, z(s)\big)$ and $\boldsymbol{w}_{s\alpha_i}$ are uncorrelated for all $i = 1, \ldots, q$.

$$\begin{aligned} (z(t) \,|\, z(s)) &= z(t) - \hat{E}[z(t) \,|\, z(s)] \\ &= z(t) - P_{ts} P_s^{-1} z(s) \end{aligned}$$

The state $z(t)$ is a function of $z(0)$ and the process noise on the path from node 0 to node t, while $z(s)$ is a function of $z(0)$ and the process noise on the path from node 0 to node s. Thus both $z(s)$ and $z(t)$ are uncorrelated with $w_{s\alpha_i}$ for all $i = 1, \ldots, q$, and so must be $\big(z(t) \,|\, z(s)\big)$. **Q.E.D.**

Appendix B

Multiscale Estimation Equations and Error Model

This section includes the algorithm for the multiscale estimator derived in [17, 19]. This algorithm has also been described in numerous other places, e.g., [32, 59], and is included here only for completeness. The equations for the estimation error model developed in [61] are also included, as they are required for the conditional simulations generated in Chapter 6.

Consider the problem of estimating the states $z(s)$ of a tree process given measurements in the form of Eq. (2.27). As for optimal estimators of 1D time-series models, like the Rauch-Tung-Striebel estimator [82], an important conceptual tool is the use of *backwards* Markov models. Since the forwards multiscale models are defined as an autoregression from coarse to fine scale, the backwards models are defined from fine to coarse scale. Using the results of [93], the multiscale models in the form of Eq. (2.22) satisfy the following backwards model [19]

$$z(s\overline{\gamma}) = F_s z(s) + \overline{w}(s)\,, \tag{B.1a}$$

$$\overline{w}(s) \sim (0, \overline{Q}_s)\,, \tag{B.1b}$$

where the parameters of the backwards model are given by

$$\begin{aligned} F_s &= P_{z(s\overline{\gamma})z(s)} P^{-1}_{z(s)}\,, \\ &= P_{z(s\overline{\gamma})} A_s^T P^{-1}_{z(s)}\,, \end{aligned} \tag{B.1c}$$

$$\begin{aligned} \overline{Q}_s &= P_{z(s\overline{\gamma})} - F_s\, P_{z(s)z(s\overline{\gamma})}\,, \\ &= P_{z(s\overline{\gamma})}\big(I - A_s^T P^{-1}_{z(s)} A_s P_{z(s\overline{\gamma})}\big)\,. \end{aligned} \tag{B.1d}$$

Note that these equations simply represent the LLSE estimate of $z(s\overline{\gamma})$ from $z(s)$, as can be seen by comparing Eq. (B.1) to Eq. (2.9) with $\overline{w}(s)$ as the estimation error. The backwards model differs slightly from the forwards model in that the process noise $\overline{w}(s)$ is not uncorrelated with all other process noise. The results of [93] only guarantee that $\overline{w}(s)$ is uncorrelated along any path to the root node.

To simplify the multiscale estimator equations, we first define

$$\begin{aligned}
\hat{z}(s) &\triangleq \text{the LLSE estimate of } z(s) \text{ given all } y(s)\,,\\
e(s) &\triangleq z(s) - \hat{z}(s)\,,\\
Y_s &\triangleq \{y(\sigma) \,|\, \sigma \in \mathcal{S}_s, \sigma \neq s\}\,,
\end{aligned}$$

and

$$\begin{aligned}
\hat{z}(s \,|\, \sigma) &\triangleq \text{the LLSE estimate of } z(s) \text{ from } Y_\sigma \text{ and } y(\sigma)\,,\\
P(s \,|\, \sigma) &\triangleq \text{the covariance of the error } z(s) - \hat{z}(s \,|\, \sigma)\,,\\
\hat{z}(s \,|\, \sigma+) &\triangleq \text{the LLSE estimate of } z(s) \text{ from } Y_\sigma\,,\\
P(s \,|\, \sigma+) &\triangleq \text{the covariance of the error } z(s) - \hat{z}(s \,|\, \sigma+)\,.
\end{aligned}$$

The estimation algorithm can be broken into two steps, analogous to the two sweeps of the Rauch-Tung-Striebel smoother. The only difference is that the sweeps for the multiscale algorithm proceed in scale, from the leaf nodes to the root node, and then from the root node to the leaf nodes. This two sweep algorithm allows "communication" between all nodes on the tree.

The Upwards Sweep

Assume for simplicity that all of the leaf nodes of the tree are at scale $m(s) = M$. Then for all s such that $m(s) = M$, define

$$\hat{z}(s \,|\, s+) = 0\,, \tag{B.2a}$$
$$P(s \,|\, s+) = P_{z(s)}\,. \tag{B.2b}$$

The need for this initialization will be apparent for the measurement update step below.

The upwards sweep, which calculates the LLSE estimate $z(s \,|\, s)$ and the corresponding error covariance is given by the following: for $m = M, M-1, \ldots, 0$,

1. Measurement update

$$\hat{z}(s \,|\, s) = \hat{z}(s \,|\, s+) + K_s\big(y(s) - C_s\hat{z}(s \,|\, s+)\big) \tag{B.3a}$$
$$P(s \,|\, s) = (I - K_sC_s)P(s \,|\, s+) \tag{B.3b}$$
$$= (I - K_sC_s)P(s \,|\, s+)(I - K_sC_s)^T + K_sR_sK_s^T \tag{B.3c}$$
$$K_s = P(s \,|\, s+)C_s^T\big(C_sP(s \,|\, s+)C_s^T + R_s\big)^{-1} \tag{B.3d}$$

2. Upward prediction, for $m \neq 0$

$$\hat{z}(s \,|\, s\alpha_i) = F_{s\alpha_i}\hat{z}(s\alpha_i \,|\, s\alpha_i) \tag{B.4a}$$
$$P(s \,|\, s\alpha_i) = F_{s\alpha_i}P(s\alpha_i \,|\, s\alpha_i)F_{s\alpha_i}^T + \overline{Q}_{s\alpha_i} \tag{B.4b}$$

3. Merge of predictions, for $m \neq M$

$$\hat{z}(s \mid s+) = P(s \mid s+) \sum_{i=1}^{q_s} P^{-1}(s \mid s\alpha_i)\hat{z}(s \mid s\alpha_i) \tag{B.5a}$$

$$P(s \mid s+) = \left[(1-q_s)P_{z(s)}^{-1} + \sum_{i=1}^{q_s} P^{-1}(s \mid s\alpha_i)\right]^{-1} \tag{B.5b}$$

Note that the measurement update at the root node produces the optimal estimate $\hat{z}(0) = \hat{z}(0 \mid 0)$. Steps 1 and 2 are exactly analogous the Kalman filter update and prediction steps. The only difference is the merge step, which is necessary for fusing the multiple predictions of a state $z(s)$ from its children.

The Downwards Sweep

The downwards sweep, which computes the optimal estimates $\hat{z}(s)$ and their associated error covariances $P_{e(s)}$, fuses the results of the upwards sweep with those of the downwards sweep. Namely, for $m = 1, \ldots, M$,

$$\hat{z}(s) = (I - J_s F_s)\,\hat{z}(s \mid s) + J_s \hat{z}(s\overline{\gamma}), \tag{B.6a}$$

$$P_{e(s)} = P(s \mid s) - J_s\left[P(s\overline{\gamma} \mid s) - P(s\overline{\gamma})\right]J_s^T, \tag{B.6b}$$

$$J_s = P(s \mid s)F_s^T P^{-1}(s\overline{\gamma} \mid s), \tag{B.6c}$$

where the sweep is initialized by $\hat{z}(0) = \hat{z}(0 \mid 0)$ and $P_{e(s)} = P(0 \mid 0)$.

After completing the downward sweep, the LLSE estimate of state $z(s)$ has been computed. An important feature of the multiscale estimation algorithm is that the LLSE error covariance $P_{e(s)}$ is also provided for no additional computational cost. These covariances allow one to assess the fidelity of the estimates in terms of their second-order statistics. To compute the cross covariances $P_{e(s)e(t)}$ for $s \neq t$ requires additional computations and the use of the multiscale estimation error model provided.

Numerical Stability

While the equations for the multiscale estimator are relatively straightforward, they assume that many of the state and estimation error covariances are nonsingular and well-conditioned. In particular, Eqs. (B.3d), (B.5b), and (B.6c) all require the inversion of matrices which in practice can be singular or ill-conditioned. These singularities can be overcome using more a robust implementation of the LLSE estimator, such as the SVD-based implementation provided in Section 2.1.1. Note also that we have included the Joseph stabilized form for the measurement update covaraince in Eq (B.3c), which ensures that the covariance $P(s \mid s)$ is numerically positive semidefinite.

Estimation Error Model

The estimation errors $e(s) = z(s) - \hat{z}(s)$ can also be modeled by a multiscale model whose parameters are by-products of the estimator equations [61]. Namely, the estimation errors satisfy the following autoregression in scale

$$e(s) = J_s e(s\overline{\gamma}) + \widetilde{w}(s)\,, \tag{B.7a}$$

where

$$\widetilde{Q}_s = E\left[\widetilde{w}(s)\widetilde{w}(s)^T\right] \tag{B.7b}$$

$$= P_{e(s)} - J_s P_{e(s\overline{\gamma})} J_s^T \tag{B.7c}$$

$$= P(s\,|\,s) - \underbrace{P(s\,|\,s)F_s^T P^{-1}(s\overline{\gamma}\,|\,s)}_{J_s} F_s P(s\,|\,s)\,, \tag{B.7d}$$

This autoregression is initialized by $e(0) \sim \left(0, P_{e(0)}\right)$, and J_s is computed as a by-product of the estimator in Eq. (B.6c).

The error model allows for the computation of cross-covariances $P_{e(s)e(t)}$ using Eq. (2.26). The error model is also useful for conditional simulation, i.e., for generating sample paths of the multiscale process conditioned upon the measurements. Because the estimate $\hat{z}(s)$ is uncorrelated with the error process, samples from the error process can be computed independently of $\hat{z}(s)$, and then added to $\hat{z}(s)$ to form samples of the conditional distribution. The generation of a sample path of $e(s)$ requires the generation of samples of $e(0)$ and the process noise $\widetilde{w}(s)$. The distribution of $e(0)$ and $\widetilde{w}(s)$ is known exactly if $z(s)$ and $y(s)$ are jointly Gaussian. In this case, the covariances $P_{e(0)}$ and $\widetilde{Q}_s$ can be either Cholesky[1] or square-root factored to yield the relationships

$$e(0) = U_0 w_0\,, \tag{B.8}$$

$$\widetilde{Q}_s = U_s w_s\,, \tag{B.9}$$

where the vectors w_s are white Gaussian noise and can be generated rather easily. If $z(s)$ and $y(s)$ are not jointly Gaussian, then this method can be used to generate sample paths with the same second-order statistics as $z(s)$ conditioned on $y(s)$.

[1]The danger of the Cholesky factorization is that $P_{e(0)}$ and $\widetilde{Q}_s$ will in many cases be numerically singluar or negative definite.

Appendix C

Variational Method for Linearizing the Flow Equation

This section demonstrates how to compute the first variation of hydraulic head. The first variation of hydraulic head is the component of head which is linearly related to a perturbation in the conductivity function. This variation allows us to establish a linear relationship between head and conductivity for small perturbations in conductivity.

The complete and nonlinear relationship between head and conductivity is given by

$$-\nabla \cdot \left(e^{f(x)} \nabla h(x)\right) = Q(x), \qquad x \in \Omega \tag{C.1a}$$

$$h(x) = h_b(x), \qquad x \in \partial\Omega_D \tag{C.1b}$$

$$-e^{f(x)} \nabla h(x) \cdot \hat{n}(x) = q_b(x), \qquad x \in \partial\Omega_N \tag{C.1c}$$

where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. This equation can be written more compactly as $\mathcal{A}(f,h) = b$, where

$$\mathcal{A}(f,h) = \begin{bmatrix} -\nabla \cdot \left(e^f \nabla h\right) \\ h \\ -e^f \nabla h \cdot \hat{n} \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} Q \\ h_b \\ q_b \end{bmatrix}. \tag{C.2}$$

Note that $\mathcal{A}(f,h) = b$ is the implicit representation of the forward operator $h = \mathcal{F}_h(f)$.

We can expand the implicit differential operator about the conductivity function f_0 by introducting a perturbation $f = f_0 + \delta f$. If $h_0 = \mathcal{F}_h(f_0)$ and δh is defined to be the resulting head perturbation, then these perturbation must satisfy $\mathcal{A}(f_0+\delta f, h_0+\delta h) = b$. Writing out this system of equations gives

$$\mathcal{A}(f_0 + \delta f, h_0 + \delta h) = \begin{bmatrix} -e^{\delta f}(\nabla \cdot e^{f_0} \nabla h_0) - e^{f_0+\delta f}(\nabla^2 \delta h + \nabla f_0 \cdot \nabla \delta h + \nabla h_0 \cdot \nabla \delta f + \nabla \delta h \cdot \nabla \delta f) \\ h_0 + \delta h \\ -e^{f_0+\delta f} \nabla (h_0 + \delta h) \end{bmatrix}. \tag{C.3}$$

The first variation is a linear relationship between δf and δh, so we can ignore the higher-order terms. Also, by making the substitutions $\mathcal{A}(f_0, h_0) = b$ and $e^{\delta f} = 1 + \delta f$

we obtain

$$-\nabla \cdot e^{f_0} \nabla \delta h - e^{f_0} \nabla h_0 \cdot \nabla \delta f - \delta f \, (\nabla e^{f_0} \cdot \nabla h_0) = 0 \,, \qquad x \in \Omega \qquad \text{(C.4a)}$$

$$\delta h = 0 \,, \qquad x \in \partial\Omega_D \qquad \text{(C.4b)}$$

$$-e^{f_0} (\nabla \delta h + \delta f \, \nabla h_0 + \delta f \nabla \delta h) \cdot \widehat{n} = 0 \,. \qquad x \in \partial\Omega_N \qquad \text{(C.4c)}$$

Ignoring the higher-order terms and combining terms yields the following linear reala-tionship between δh and δf

$$-\nabla \cdot e^{f_0} \nabla \delta h = \nabla \cdot e^{f_0} \, \delta f \, \nabla h_0 \,, \qquad x \in \Omega \qquad \text{(C.5a)}$$

$$\delta h = 0 \,, \qquad x \in \partial\Omega_D \qquad \text{(C.5b)}$$

$$-e^{f_0} \nabla \delta h \cdot \widehat{n} = e^{f_0} \, \delta f \, (\nabla h_0 \cdot \widehat{n}) \,. \qquad x \in \partial\Omega_N \qquad \text{(C.5c)}$$

Appendix D

Proof of the Multiscale Realization Algorithm

This section demonstrates that the multiscale models produced by the realization algorithm of Section 4.2 have the desired finest-scale covariance. Remember that a multiscale model is defined in terms of the root node covariance P_0 and the autoregression parameters A_s and Q_s. Thus the proof must work directly with these parameters.

The assumption made by the following proof is that the multiscale model is internal. Recall that a model whose parameters are derived from Eqs. (2.30) and (2.32) need not be internal (see Section 4.4, even when the internal matrices W_s are chosen to satisfy the Markov property. However, since one can always choose a set of internal matrices W_s for which the resulting model is internal, we focus only on this case. The extension to external models is rather straightforward, and simply involves partitioning the states into two components, one which is a linear function of the finest scale process and the other which is uncorrelated with all other variables on the tree.

For notational simplicity, we will assume that the trees are binary with $M+1$ scales and 2^M nodes at the finest scale. The proof is inductive. For $M = 1$, the tree has three nodes, with $z(0)$ at the root node and $z(1)$ and $z(2)$ at the finest scale. Assume that f_1 is mapped to node one and f_2 is mapped to node two, where $f = [f_1^T,\ f_2^T]^T$. The parameters of the tree are

$$P_0 = W_0 P_f W_0^T\,, \tag{D.1a}$$

$$A_i = (W_i P_f W_0^T)(W_0 P_f W_0^T)^{-1}\,, \qquad i = 1,2 \tag{D.1b}$$

$$Q_i = (W_i P_f W_i^T) - (W_i P_f W_0^T)(W_0 P_f W_0^T)^{-1}(W_i P_f W_0^T)^T\,, \qquad i = 1,2 \tag{D.1c}$$

where W_0 is chosen according to the Markov property and $W_i f = f_i$, $i = 1,2$. Straightforward calculations show that $P_{z(i)} = W_i P_f W_i^T$ for $i = 1,2$. The cross-covariance

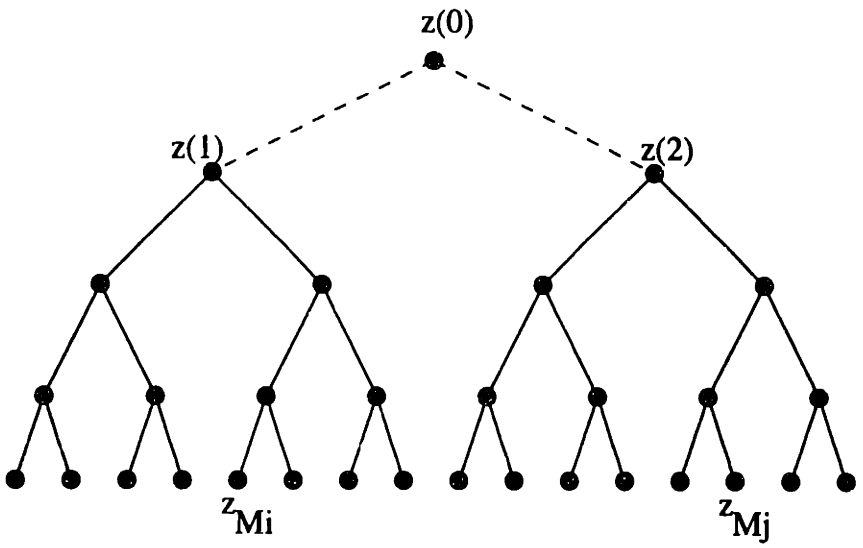


Figure D.1. The fusion of two binary trees into one tree with $M = 4$. The dashed lines denote the branches added to the fused tree.

between the two finest-scale variables follows as

$$P_{z(1)z(2)} = A_1 P_{z(0)} A_2^T , \tag{D.2a}$$

$$= (W_1 P_f W_0^T) P_{z(0)}^{-1} (W_2 P_f W_0^T)^T , \tag{D.2b}$$

$$= W_1 \underbrace{P_f W_0^T (W_0 P_f W_0^T)^{-1} W_0 P_f}_{P_f - P_{(f \mid W_0 f)}} W_2^T , \tag{D.2c}$$

where $(f \mid W_0 f)$ is equal to f conditioned on $W_0 f$. Since f_1 and f_2 are uncorrelated after conditioning upon $W_0 f$, we have

$$P_{z(1)z(2)} = W_1 P_f W_2^T , \tag{D.3a}$$

$$= P_{f_1 f_2} . \tag{D.3b}$$

Now assume that the vector $f = [f_1^T, f_2^T]^T$ is the process to be mapped to the finest scale of an $(M+1)$-scale binary tree. For the inductive step, we can also assume that we are given two multiscale processes with M scales, where the finest scale of the first tree has the distribution $f_1 \sim (0, P_{f_1})$ and the finest-scale of the second tree has the distribution $f_2 \sim (0, P_{f_2})$. The $(M+1)$-scale tree is formed by fusing the two trees as illustrated in Figure D.1. The root node of the first M-scale becomes $z(1)$, while that of the second M-scale becomes $z(2)$.

We must now show that the $(M+1)$-scale tree has the proper second-order statistics at the finest scale. By assumption, the finest scales of each of the two subtrees have the proper auto-covariances P_{f_1} and P_{f_2}. Define z_{Mi} to be the i-th variable at the finest scale of the tree, where $1 \leq i \leq 2^{M-1}$ indexes the the variables of the first (left)

subtree and $2^{M-1}+1 \leq i \leq 2^M$ indexes the the variables of the second (right) subtree. Also define D_i to be the matrix which satisfies $z_{Mi} = D_i f$. We only need to show that $E[z_{Mi} z_{Mj}^T] = D_i P_f D_j^T$ for all $1 \leq i \leq 2^{M-1}$ and $2^{M-1}+1 \leq j \leq 2^M$. From the autoregression parameters, we have

$$E[z_{Mi} z_{Mj}^T] = \left(\prod_{l \in P_i} A_l \right) (A_1 P_{z(0)} A_2^T) \left(\prod_{k \in P_j} A_k \right)^T , \tag{D.4}$$

where P_i is that set of nodes on the path from the i-th node at scale M to node one, excluding node one. Similarly, P_j is that set of nodes on the path from the j-th node at scale M to node two, excluding node two. These state transition matrices can be also be represented as

$$\prod_{l \in P_i} A_l = E[(D_i f) z(1)^T] P_{z(1)}^{-1} , \tag{D.5a}$$

$$= (D_i P_f [W_1 \; 0]^T) P_{z(1)}^{-1} , \tag{D.5b}$$

$$\prod_{k \in P_j} A_k = E[(D_j f) z(2)^T] P_{z(2)}^{-1} , \tag{D.5c}$$

$$= (D_j P_f [0 \; W_2]^T) P_{z(2)}^{-1} , \tag{D.5d}$$

where the second and fourth equalities follow from the assumption that the model is internal. Substituting Eq. (D.5) into Eq. (D.4) yields

$$E[z_{Mi} z_{Mj}^T] = D_i \underbrace{P_f [W_1 \; 0]^T (W_1 P_{f_1} W_1^T)^{-1} [W_1 \; 0]^T P_f}_{P_f - P_{(f \mid W_1 f_1)}} [0 \; W_2]^T (W_2 P_{f_2} W_2^T)^{-1} [0 \; W_2] P_f D_j^T , \tag{D.6}$$

where we have used $A_1 P_{z(0)} A_2^T = [W_1 \; 0]^T P_f [0 \; W_2]^T$, which follows from the analysis of the two-scale tree. The matrix $P_{(f \mid W_1 f_1)}$ is the covariance of f after conditioning on $W_1 f_1$. Since f_1 and f_2 are uncorrelated after conditioning on $W_1 f_1$, Eq. (D.6) reduces to

$$E[z_{Mi} z_{Mj}^T] = D_i \underbrace{P_f [0 \; W_2]^T (W_2 P_{f_2} W_2^T)^{-1} [0 \; W_2] P_f}_{P_f - P_{(f \mid W_2 f_2)}} D_j^T , \tag{D.7a}$$

$$= D_i P_f D_j^T , \tag{D.7b}$$

where $(f \mid W_2 f_2)$ is f after conditioning on $W_2 f_2$. **Q.E.D**

Bibliography

[1] S. Ahmed and G. de Marsily. Cokriged estimation of aquifer transmissivity as an indirect solution of the inverse problem: a practical approach. *Water Resources Res.*, 29(2):521–530, February 1993.

[2] H. Akaike. Markovian representations of stochastic processes by canonical variables. *SIAM J. of Control*, 13(1), January 1975.

[3] K. Aziz and S. Settari. *Petroleum Reservoir Simulation.* Applied Science, London, 1979.

[4] M. F. Barnsley, R. L. Devaney, B. B. Mandelbrot, H.-O. Peitgen, D. Saupe, and R. F. Voss. *The Science of Fractal Images.* Springer-Verlag, 1988.

[5] R. Barrett et al. *Templates for the Solution of Linear Systems.* SIAM, 1995.

[6] J. Bear. *Dynamics of Fluids in Porous Media.* American Elsevier, New York, 1972.

[7] K. E. Brewer and S. W. Wheatcraft. *Wavelets in Geophysics*, chapter Inculding multi-scale information in the characterization of hydraulic conductivity distributions, pages 213–248. Academic Press, 1994.

[8] J. R. Buck, M. M. Daniel, and A. C. Singer. *Computer Explorations in Signals and Systems.* Prentice-Hall, Upper Saddle River, NJ, 1997.

[9] J. H. Butler and W. Liu. Pumping tests in nonuniform aquifers: the radially asymmetric case. *Water Resources Res.*, 29(2):259–269, Feb. 1993.

[10] J. Carrera and S. P. Neuman. Estimation of aquifer parameters under transient and steady-state conditions: parts 1-3. *Water Resources Res.*, 22(2):199–242, 1986.

[11] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids.* Oxford U. Press, Oxford, 1959.

[12] G. Chavent. On the theory and practice of non-linear least-squares. *Adv. Water Resources*, 14(2), 1991.

[13] G. Chavent and M. Dupuy. History matching by use of optimal theory. *SPE Journal*, pages 74–86, February 1975.

[14] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *IEEE Transactions on ASSP*, 33(4):959–963, 1985.

[15] R. Chellappa and A. Jain, editors. *Markov Random Fields.* Academic Press, 1993.

[16] W. H. Chen, G. R. Gavalas, Seinfeld J. H., and M. L. Wasserman. A new algorithm for automatic history matching. *SPE Journal*, pages 593–608, December 1974.

[17] K. C. Chou. *A stochastic modeling approach to multiscale signal processing.* PhD thesis, M.I.T., May 1991.

[18] K. C. Chou and A. S. Willsky. A multi-resolution, probabilistic approach to two-dimensional inverse conductivity problems. *Signal Processing*, 18, 1989.

[19] K. C. Chou, A. S. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Trans. Automat. Contr.*, 39(3):464–478, 1994.

[20] K. C. Chou, A. S. Willsky, and R. Nikoukhah. Mulitscale systems, Kalman filters, and Riccati equations. *IEEE Trans. Automat. Contr.*, 39(3):479–491, 1994.

[21] M. M. Daniel and A. S. Willsky. A multiresolution methodology for signal-level fusion and data assimilation with applications to remote sensing. *Proc. of the IEEE*, January 1997.

[22] M. H. A. Davis. *Linear Estimation and Stochastic Control.* Chapman and Hall, London, 1977.

[23] J. P. Delhomme. Spatial variability and uncertainty in groundwater flow parameters: a geostatistical approach. *Water Resources Res.*, 15(2):269–280, April 1979.

[24] L. R. Denham. Seismic interpretation. *Proc. of the IEEE*, 72(10):1255–1265, October 1984.

[25] H. Derin and P. A. Kelly. Discrete-index Markov-type random processes. *Proc. IEEE*, Oct. 1989.

[26] A. H. Dogru and J. H. Seinfeld. Comparison of sensitivity coefficient calculation methods in automatic history matching. *SPE Journal*, pages 551–557, Oct. 1981.

[27] P. A. Domenico and F. W. Schwartz. *Physical and Chemical Hydrogeology.* Wiley, 1990.

[28] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices.* Oxford Univ., 1987.

[29] R. C. Earlougher. *Advances in Well Test Analysis.* Society of Petroluem Engineers, New York, 1977.

[30] A. M. Erisman and W. F. Tinney. On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3), 1975.

[31] S. Ezzedine and Y. Rubin. A geostatistical approach to the conditional estimation of spatially distributed solute concentration and notes on the use of tracer data in the inverse problem. *Water Resources Research*, 32(4):853–861, April 1996.

[32] P. W. Fieguth. *Application of Multiscale Estimation to Large Scale Multidimensional Imaging and Remote Sensing Problems.* PhD thesis, M.I.T., June 1995.

[33] P. W. Fieguth, W. C. Karl, and A. S. Willsky. Efficient multiresolution counterparts to variational methods for surface reconstruction. *Submitted to Computer Vision and Image Understanding*, 1995.

[34] P. W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch. Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDEN satellite altimetry. *IEEE Trans. Geosci. Remote Sensing*, 33(2):280–292, March 1995.

[35] P. W. Fieguth and A. S. Willsky. Fractal estimation using models on multiscale trees. *IEEE Trans. Sig. Proc.*, pages 1297–1300, May 1996.

[36] P. Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Trans. Info. Theory*, 38(2):910–917, March 1992.

[37] R. M. Gailey, A. S. Crowe, and S. M. Gorelick. Coupled process parameter estimation and prediction uncertainty using hydraulic head and concentration data. *Advanced Water Resources*, 14(5):301–313, 1991.

[38] L. G. Gelhar. *Stochastic Subsurface Hydrology.* Prentice Hall, New York, 1993.

[39] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 1973.

[40] A. George and J. W. Liu. *Computer Solution of Large and Sparse Positive Definite Systems.* Prentice Hall, Englewood Cliffs, NJ, 1981.

[41] M. Ghil and P. Malanotte-Rizzoli. *Advances in Geophysics*, volume 33, chapter Data assimilation in meteorology and oceanograpy, pages 141–266. Academic Press, 1991.

[42] T. R. Ginn and J. H. Cushman. Inverse methods for subsurface flow: a critical review of stochastic techniques. *Stochastic Hydrology and Hydraulics*, 4:1–26, 1990.

[43] G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins, Baltimore, 1990.

[44] M. D. Greenberg. *Application of Green's Functions in Science and Engineering.* Prentice-Hall, 1971.

[45] J. M. R. Hosking. Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research*, 20(12):1898–1908, December 1984.

[46] G. J. Huffman, R. F. Adler, B. Rudolf, U. Schneider, and P. R. Keehn. Global precipitation estimates based on a technique for combining satellite-based estimates, rain gauge analysis, and NWP model precipitation information. *Journal of Climate*, 8:1284–1295, 1995.

[47] D. W. Hyndman, J. M. Harris, and S. M. Gorelick. Coupled seismic and tracer test inversion for aquifer property characterization. *Water Resources Res.*, 30(7):1965–1977, July 1994.

[48] Mathworks Inc. *Matlab User's Guide.* Natick, MA, 4.0 edition, August 1992.

[49] W. W. Irving. *Multiscale stochastic realization and model identification with applications to large-scale estimation problems.* PhD thesis, MIT, August 1995.

[50] W. W. Irving. Thoughts on vacuous posturing. Private Communication, 1996.

[51] W. W. Irving and A. S. Willsky. A canonical correlations approach to multiscale stochastic realization. *Submitted to IEEE Trans. on Image Proc.*, 1996.

[52] P. Jacquard and C. Jain. Permeability distribution from field pressure data. *SPE Journal*, pages 281–294, December 1965.

[53] H. O. Jahns. A rapid method for obtaining a two-dimensional reservoir description from well pressure response data. *SPE Journal*, pages 315–327, December 1966.

[54] F. John. *Partial Differential Equations.* Springer-Verlag, fourth edition, 1982.

[55] L. M. Kaplan and Kuo C.-C. J. Fractal estimation from noisy data via discrete fractional Gaussian noise (DFGN) and the Haar basis. *IEEE Trans. on Signal Processing*, 41(12):3554, December 1993.

[56] N. J. Kasdin. Discrete simulation of colored noise and stochastic processes and $1/f^{\alpha}$ power law noise generation. *Proc. of the IEEE*, 83(5):802–827, May 1995.

[57] M. S. Keshner. $1/f$ noise. *Proc. of the IEEE*, 70(3):212–218, March 1982.

[58] J. Liu. A multiresolution method for distributed parameter estimation. *SIAM Journal on Scientific Computing*, 14(2):389–405, March 1993.

[59] M. R. Luettgen. *Image processing with mulitscale stochastic models.* PhD thesis, M.I.T., May 1993.

[60] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Trans. Image Proc.*, 3(1):41–64, January 1994.

[61] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Multiscale smoothing error models. *IEEE Trans. Automatic Control*, 40(1):173–175, January 1995.

[62] M. R. Luettgen, W. C. Karl, A. S. Willsky, and R. R. Tenney. Multiscale representations of Markov random fields. *IEEE Trans. Signal Proc.*, 41(12):3377, December 1993.

[63] M. R. Luettgen and A. S. Willsky. Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Trans. Image Processing*, 4(2):194–207, February 1995.

[64] B. Mandelbrot and J. van Ness. Fractional Brownian motions, fractional noises, and applications. *SIAM Review*, 10:422–437, 1968.

[65] G. de Marsily. *Quantitative Hydrogeology: Groundwater Hydrology for Engineers.* Academic Press, San Diego, CA, 1986.

[66] C. S. Matthews and D. G. Russell. *Pressure Buildup and Flow Tests in Wells.* Society of Petroleum Engineers, New York, 1967.

[67] D. B. McLaughlin. Personal conversation, 1996.

[68] D. B. McLaughlin and L. B. Reid. Estimating continuous aquifer properties from field measurements : the inverse problem for groundwater flow and transport. In *Computational Methods in Water Resources*, pages 777–784. Kluwer Academic, 1994.

[69] D. B. McLaughlin and L. R. Townley. A reassessment of the groundwater inverse problem. *Water Resources Research*, 32(5):1131–1161, 1996.

[70] D. Menemenlis, P. W. Fieguth, C. Wunsch, and A. S. Willsky. A fast optimal interpolation algorithm for mapping hydrographic and other oceanographic data. *Submitted to Journal of Geophysical Research*, 1996.

[71] W. Menke. *Geophysical Data Analysis: Discrete Inverse Theory.* Academic Press, 1984.

[72] E. L. Miller. A multiscale approach to sensor fusion and the solution of linear inverse problems. *Applied and Comp. Harmonic Anal.*, 2:127–147, 1995.

[73] R. E. Moore. *Computational Functional Analysis.* Ellis Harwood Ltd. (John Wiley & Sons), Chichester, 1985.

[74] S. P. Neuman, G. E. Fogg, and Jacobsen E. A. A statisitical approach to the inverse problem of aquifer hydrology, 2, case study. *Water Resources Res.*, 16:33–58, 1980.

[75] D. S. Oliver. The averaging process in permeability estimation from well test data. *SPE Formation Evaluation*, pages 319–324, September 1990.

[76] D. S. Oliver. Estimation of radial permeability distribution from well test data. *SPE Formation Evaluation*, pages 290–296, December 1992.

[77] D. S. Oliver. Influence of nonuniform transmissivity on storativity and drawdown. *Water Resources Res.*, 29(1):169–178, January 1993.

[78] D. S. Oliver. Incorporation of transient pressure data into reservoir characterization. *In Situ*, 18(3):243–275, 1994.

[79] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, second edition, 1984.

[80] T. Patera. Introduction to computational fluid dynamics. 2.274 Supplemental Class Notes (MIT), fall 1995.

[81] I. Primus. Scale-recursive estimation of precipitation using remote sensing data. Master's thesis, M.I.T., June 1996.

[82] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamics systems. *AIAA Journal*, 3(8), August 1965.

[83] M. Reed and B. Simon. *Functional Analysis.* Academic Press, 1990.

[84] L. B. Reid. *A functional inverse approach for three-dimensional characterization of subsurface contamination.* PhD thesis, Mass. Inst. of Tech., June 1996. Dept. of Civ. Engin.

[85] Y. Rubin and G. Dagan. Conditional estimation of solute travel time in heterogeneous formations: impact of transmissivity measurements. *Water Resources Research*, 28(4):1033–1040, April 1992.

[86] Y. Rubin, G. Mavko, and J. Harris. Mapping permeability in heterogeneous aquifers using hydrologic and seismic data. *Water Resources Research*, 28(7):1809–1816, July 1992.

[87] K. S. Shanmugan and A. M. Breipohl. *Random Signals: Detection, Estimation, and Data Analysis.* John Wiley & Sons, 1988.

[88] G. E. Slater and E. J. Durrer. Adjustment of reservoir simulation models to match field performance. *SPE Journal*, pages 295–305, September 1971.

[89] M. A. Stoksik, R. G. Lane, and D. T. Nguyen. Practical synthesis of accurate fractal images. *Graphical Models and Image Processing*, 57(3):206–219, May 1995.

[90] A. H. Tewfik and M. Kim. Correlation structure of the discrete wavelet coefficients of fractional Brownian motion. *IEEE Trans. Info. Theory*, 38(2):904–909, March 1992.

[91] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems.* Halsted Press (Wiley), New York, 1977.

[92] C. T. Verdín and T. M. Habashy. An approach to nonlinear inversion with applications to cross-well em tomography. In *SEG International Meeting*, Washington, D.C., 1993.

[93] G. Verghese and T. Kailath. A further note on backwards Markovian models. *IEEE Trans. Info. Theory*, 25(1):121–4, January 1979.

[94] W. Willinger, M. Taqqu, W. E. Leland, and D. V. Wilson. Self-similarity in high-speed packet traffic: analysis and modeling of ethernet traffic measurements. *Statistical Science*, 10:67–85, 1995.

[95] A. S. Willsky, G. W. Wornell, and J. H. Shapiro. Stochastic processes, detection and estimation. 6.432 Supplemental Class Notes (MIT), fall 1995.

[96] G. W. Wornell. Wavelet-based representations for the $1/f$ family of fractal processes. *Proc. of the IEEE*, 81(10):1428–1450, 1993.

[97] W. W. Yeh. Review of parameter identification procedures in groundwater hydrology: the inverse problem. *Water Resources Res.*, 22(2):95–108, 1986.

[98] D. A. Zimmerman, C. L. Axness, G. de Marsily, M. G. Marietta, and C. A. Gotway. *Parameter Identification and Inverse Problems in Hydrology, Geology, and Ecology*, chapter Results from a comparison of geostatistical inverse techniques for groundwater flow, pages 105–120. Kluwer Academic, 1996.

[99] D. A. Zimmerman et al. A comparison of seven geostatistically-based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow. *Submitted to Water Resources Research*, August 1996.

[100] X. Zou, I. M. Navon, M. Berger, K. H. Phua, T. Chlick, and F. X. Le Diment. Numerical experience with limited memory quasi newton and truncated newton methods. *SIAM Journal on Optimization*, 3(3):582–608, 1993.

Silver Oaks, Cabernet Sauvignon, 1985