Preface

The difference between theory and practice is smaller in theory than it is in practice. –folklore

We make discoveries about reality by examining the difference between theory and practice. There is a well-developed *theory* about the difference between theory and practice, and it is called “geophysical inverse theory”. In this book we investigate the *practice* of the difference between theory and practice. As the folklore tells us, there is a big difference. There are already many books on the theory, and often as not, they end in only one or a few applications in the author’s specialty. In this book on practice, we examine data and results from many diverse applications. I have adopted the discipline of suppressing theoretical curiosities until I find data that requires it (except for a few concepts at chapter ends).

Books on geophysical inverse theory tend to address theoretical topics that are little used in practice. Foremost is probability theory. In practice, probabilities are neither observed nor derived from observations. For more than a handful of variables, it would not be practical to display joint probabilities, even if we had them. If you are data poor, you might turn to probabilities. If you are data rich, you have far too many more rewarding things to do. When you estimate a few values, you ask about their standard deviations. When you have an image making machine, you turn the knobs and make new images (and invent new knobs). Another theory not needed here is singular-value decomposition.

In writing a book on the “practice of the difference between theory and practice" there is no worry to be bogged down in the details of diverse specializations because the geophysical world has many interesting data sets that are easily analyzed with elementary physics and simple geometry. (My specialization, reflection seismic imaging, has a great many less easily explained applications too.) We find here many applications that have a great deal in common with one another, and that commonality is not a part of common inverse theory. Many applications draw our attention to the importance of two weighting functions (one required for data space and the other for model space). Solutions depend strongly on these weighting functions (eigenvalues do too!). Where do these functions come from, from what rationale or estimation procedure? We’ll see many examples here, and find that these functions are not merely weights but filters. Even deeper, they are generally a combination of weights and filters. We do some tricky bookkeeping and bootstrapping when we filter the multidimensional neighborhood of missing and/or suspicious data.

This book is not devoid of theory and abstraction. Indeed it makes an important new contribution to the theory (and practice) of data analysis: multidimensional autoregression via the helical coordinate system.

The biggest chore in the study of “the practice of the difference between theory and practice" is that we must look at algorithms. Some of them are short and sweet, but other important algorithms are complicated and ugly in any language. This book can be printed without the computer programs and their surrounding paragraphs, or you can read it without them. I suggest, however, you take a few moments to try to read each program. If you can *write* in any computer language, you should be able to *read* these programs well enough to grasp the concept of each, to understand what goes in and what should come out. I have chosen the computer language (more on this later) that I believe is best suited for our journey through the “elementary” examples in geophysical estimation.

Besides the tutorial value of the programs, if you can read them, you will know exactly how the many interesting illustrations in this book were computed so you will be well equipped to move forward in your own direction.

Age and treachery will always overcome youth and skill. –anonymous

1998 is my sixth year of working on this book and much of it comes from earlier work and the experience of four previous books. Last year I was joined by a student, Sergey Fomel. Fomel has given me the gift of designing a much needed object-oriented style for the computer code, and converting all of it to a more modern language. After I discovered the helix idea and its wide-ranging utility, he adapted all the relevant examples in the book to use it. If you read Fomel’s programs, you will see effective application of that 1990’s revolution in coding style known as “object orientation.”

Beyond the coding, Fomel brought the mathematical level to a much higher standard. He has also inspired me with many fruitful ideas of his own and as we continue, our work will become more difficult to disentangle.

Overview

This book is about the estimation and construction of geophysical maps. Geophysical maps are used for exploration, most commonly for petroleum and mineral resources, but also for water, archeology, lost treasure, graves, and environmental pollution.

Here we follow physical measurements from a wide variety of geophysical sounding devices to a geophysical map, a 1-, 2-, or 3-dimensional Cartesian mesh that is easily transformed to a graph, map image, or computer movie. A later more human, application-specific stage (not addressed here) interprets and annotates the maps; that stage places the “$\times$” where you will drill, dig, dive, or merely dream.

Map estimation is a subset of “geophysical inverse theory,” itself a kind of “theory of how to find everything.” In contrast to “everything,” maps have an organized structure (covariance) that makes their estimation more concrete and visual, and leads to the appealing results we find here.

Geophysical sounding data used in this book comes from acoustics, radar, and seismology. Sounders are operated along tracks on the earth surface (or tracks in the ocean, air, or space). A basic goal of data processing is a map that shows the earth itself, not an image of our data-acquisition tracks. We want to hide our acquisition footprint. Increasingly, geophysicists are being asked to measure changes in the earth by comparing old surveys to new ones. Then we are involved with both the old survey tracks and new ones, as well as technological changes between old sounders and new ones.

To enable this book to move rapidly along from one application to another, we avoid applications where the transform from model to data is mathematically complicated, but we include the central techniques of constructing the adjoint of any such complicated transformation. By setting aside application-specific complications, we soon uncover and deal with universal difficulties such as: (1) irregular geometry of recording, (2) locations where no recording took place and, (3) locations where crossing tracks made inconsistant measurements because of noise. Noise itself comes in three flavors: (1) drift (zero to low frequency), (2) white or steady and stationary broad band, and (3) bursty, i.e., large and erratic.

Missing data and inconsistant data are two humble, though universal problems. Because they are universal problems, science and engineering have produced a cornucopia of ideas ranging from mathematics (Hilbert adjoint) to statistics (inverse covariance) to conceptual (stationary, scale-invariant) to numerical analysis (conjugate direction, preconditioner) to computer science (object oriented) to simple common sense. Our guide through this maze of opportunities and digressions is the test of what works on real data, what will make a better image. My logic for organizing the book is simply this: Easy results first. Harder results later. Undemonstrated ideas last or not at all, and latter parts of chapters can be skimmed.

Examples here are mostly nonseismological although my closest colleagues and I mostly make maps from seismological data. The construction of 3-D subsurface landform maps from seismological data is an aggressive industry, a complex and competitive place where it is not easy to build yourself a niche. I wrote this book because I found that beginning researchers were often caught between high expectations and concrete realities. They invent a new process to build a novel map but they have many frustrations: (1) lack of computer power, (2) data-acquisition limitations (gaps, tracks, noises), or (3) they see chaotic noise and have difficulty discerning whether the noise represents chaos in the earth, chaos in the data acquisition, chaos in the numerical analysis, or unrealistic expectations.

People need more practice with easier problems like the ones found in this book, which are mostly simple 2-D landforms derived from 2-D data. Such concrete estimation problems are solved quickly, and their visual results provide experience in recognizing weaknesses, reformulating, and moving forward again. Many small steps reach a mountain top.

Scaling up to big problems

Although most the examples in this book are presented as toys, where results are obtained in a few minutes on a home computer, we have serious industrial-scale jobs always in the backs of our minds. This forces us to avoid representing operators as matrices. Instead we represent operators as a pair of subroutines, one to apply the operator and one to apply the adjoint (transpose matrix). (This will be more clear when you reach the middle of chapter 2.)

By taking a function-pair approach to operators instead of a matrix approach, this book becomes a guide to practical work on realistic-sized data sets. By realistic, I mean as large and larger than those here; i.e., data ranging over two or more dimensions, and the data space and model space sizes being larger than about 10^5 elements, about a 300×300 image. Even for these, the world's biggest computer would be required to hold in random access memory the $10^5 \times 10^5$ matrix linking data and image. Mathematica, Matlab, kriging, etc, are nice tools but[1] it was no surprise when a curious student tried to apply one to an example from this book and discovered that he needed to abandon 99.6% of the data to make it work. Matrix methods are limited not only by the size of the matrices but also by the fact that the cost to multiply or invert is proportional to the third power of the size. For simple experimental work, this limits the matrix approach to data and images of about 10^3 elements, a low-resolution 30×30 image.

[1] I do not mean to imply that these tools cannot be used in the function-pair style of this book, only that beginners tend to use a matrix approach.

The Loptran computer dialect

Along with theory, illustrations, and discussion, I display the programs that created the illustrations. To reduce verbosity in these programs, my colleagues and I have invented a little language called Loptran, that is readily translated to Fortran 90. I believe readers without Fortran experience will comfortably be able to *read* Loptran, but they should consult a Fortran book if they plan to *write* it. Loptran is not a new language compiler but a simple text processor that expands concise scientific language into the more verbose expressions required by Fortran 90.

The name Loptran denotes Linear OPerator TRANslator. The limitation of Fortran 77 overcome by Fortran 90 and Loptran is that we can now isolate natural science application code from computer science least-squares fitting code, thus enabling practitioners in both disciplines to have more ready access to one anothers intellectual product.

Fortran is the original language shared by scientific computer applications. The people who invented C and UNIX also made Fortran more readable by their invention of Ratfor. We have taken the good ideas from Ratfor which gives Loptran much the syntax of modern languages like C++ and Java. We adopted Ratfor[2] to Fortran 90 (Bob Clapp) and added a compact notation to facilitate linear operators (Sergey Fomel). Loptran is a small and simple adaptation of well-tested languages, and translates to one. Loptran is, however, new in 1998 and is not yet well tested. In this book, I avoid special features of Fortran to make everyone comfortable with Loptran as a generic algorithmic language that some people may wish to translate to other languages such as Matlab, Java, or C++.

We provide the Loptran translator free. It is written in another free language, PERL, and therefore should be available free to nearly everyone. If you prefer not to use Ratfor90 and Loptran, you can find on the WWW[3] the Fortran 90 version of all the codes in this book.

Reproducibility

Earlier versions of this series of electronic books were distributed on CD-ROM. The idea is that each computed figure in the book has in its caption a menu allowing the reader to burn and rebuild the figures (and movies). This idea persists in the Web book versions (as do the movies) except that now the more difficult task of installing the basic Stanford libraries is the obligation of the reader. Hopefully, as computers mature, this obstacle will be less formidable. Anyway, these libraries are also offered free on our web site.

Preview for inverse theorists

People who are already familiar with “geophysical inverse theory” may wonder what new they can gain from a book focused on “estimation of maps.” Given a matrix relation $\mathbf{d} = \mathbf{F}\mathbf{m}$

[2] http://sepwww.stanford.edu/sep/bob/src/ratfor90.html
[3] http://sepwww.stanford.edu/sep/prof/gee/Lib/

between model $\mathbf{m}$ and data $\mathbf{d}$, common sense suggests that practitioners should find $\mathbf{m}$ in order to minimize the length $||\mathbf{r}||$ of the residual $\mathbf{r} = \mathbf{F}\mathbf{m} - \mathbf{d}$. A theory of **Gauss** suggests that a better (minimum variance, unbiased) estimate results from minimizing the quadratic form $\mathbf{r}'\sigma_{rr}^{-1}\mathbf{r}$, where σ_{rr} is the noise **covariance matrix**. I have never seen an application in which the noise **covariance matrix** was given, but practitioners often find ways to estimate it: they regard various sums as ensemble averages.

Additional features of inverse theory are exhibited by the partitioned matrix

$$\mathbf{d} \quad = \quad \begin{bmatrix} \mathbf{d}_{\text{incons}} \\ \mathbf{d}_{\text{consis}} \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{m}_{\text{fit}} \\ \mathbf{m}_{\text{null}} \end{bmatrix} \quad = \quad \mathbf{F}\mathbf{m} \tag{1}$$

which shows that a portion $\mathbf{d}_{\text{incons}}$ of the data should vanish for any model $\mathbf{m}$, so an observed nonvanishing $\mathbf{d}_{\text{incons}}$ is inconsistent with *any* theoretical model $\mathbf{m}$. Likewise the $\mathbf{m}_{\text{null}}$ part of the model space makes no contribution to the data space, so it *seems* not knowable from the data.

Simple inverse theory suggests we should minimize $||\mathbf{m}||$ which amounts to setting the **null space** to zero. Baysian inverse theory says we should use the model **covariance matrix** σ_{mm} and minimize $\mathbf{m}'\sigma_{mm}^{-1}\mathbf{m}$ for a better answer although it would include some nonzero portion of the null space. Never have I seen an application in which the model-covariance matrix was a given prior. Specifying or estimating it is a puzzle for experimentalists. For example, when a model space $\mathbf{m}$ is a signal (having components that are a function of time) or, a stratified earth model (with components that are function of depth z) we might supplement the fitting goal $\mathbf{0} \approx \mathbf{r} = \mathbf{F}\mathbf{m} - \mathbf{d}$ with a "minimum wiggliness" goal like $dm(z)/dz \approx 0$. Neither the model **covariance matrix** nor the null space $\mathbf{m}_{\text{null}}$ *seems* learnable from the data and equation (0.1).

In fact, both the **null space** and the model **covariance matrix** can be estimated from the data and that is one of the novelties of this book. To convince you it is possible (without launching into the main body of the book), I offer a simple example of an operator and data set from which your human intuition will immediately tell you what you want for the whole model space, including the null space.

Take the data to be a sinusoidal function of time (or depth) and take $\mathbf{B} = \mathbf{I}$ so that the operator $\mathbf{F}$ is a delay operator with truncation of the signal shifted off the end of the space. Solving for $\mathbf{m}_{\text{fit}}$, the findable part of the model, you get a back-shifted sinusoid. Your human intuition, not any mathematics here, tells you that the truncated part of the model, $\mathbf{m}_{\text{null}}$, should be a logical continuation of the sinusoid $\mathbf{m}_{\text{fit}}$ at the same frequency. It should not have a different frequency nor become a square wave nor be a sinusoid abruptly truncated to zero $\mathbf{m}_{\text{null}} = \mathbf{0}$.

Prior knowledge exploited in this book is that unknowns are functions of time and space (so the **covariance matrix** has known structure). This structure gives them predictability. Predictable functions in 1-D are tides, in 2-D are lines on maps (linements), in 3-D are sedimentary layers, and in 4-D are wavefronts. The tool we need to best handle this predictability is the multidimensional "prediction-error filter" (PEF), a central theme of this book.