# Inverse Methods in the Natural Sciences

D. L. Hysell
Earth and Atmospheric Sciences
Cornell University

September 14, 2018

# Contents

# Chapter 1

# Introduction

The cornerstone of scientific inquiry as every student of science and engineering learns from the beginning is the scientific method. The core of the scientific method is the formulation of a hypothesis to account for some observed phenomenon. The hypothesis is put to experimental tests. Depending on the outcome of the tests, the hypothesis is either discarded or refined and subjected to further testing. The most successful hypothesis or theory is the one that is best supported by the preponderance of experimental data. When two competing theories account for the data equally well, the preference will be for the one which makes the fewest assumptions. No theory is perfect, and neither are experimental data, which inevitably suffer from different degrees of incompleteness, distortion, and bias. Deciding what constitutes good agreement between theory and experiment is perhaps the second greatest difficulty in science, superseded only by the difficulty in forming new hypotheses in the first place!

Formally, the scientific method might be expressed using the following equation:

$$G(m) + e \quad = \quad d \tag{1.1}$$

Here, $m$ represents the known, controlled conditions under which the experiments are performed. It could for example be a column vector with entries corresponding to the state variables as they were during the experiments. We will refer to $m$ as the "model." to be consistent with other work in the discipline, although the existence of some kind of physical or computational model is not implied here. Back to the formula, $G$ is the theory or hypothesis that predicts the experimental outcomes or "data," $d$, which occupy the right side of the equation. Once again, $d$ could be a column vector representing all the experimental measurements. The theory or hypothesis $G$ is a mapping between the controlled conditions and the experimental outcomes. If the theory is a linear mapping, then $G$ would be a matrix. Note also that $m$ and/or $d$ could equally well be continuous functions, with $G$ supplying the appropriate mapping.

For example, Galileo studied how long it takes for balls to roll down planes inclined at different angles. For this problem, $m$ could be a vector of angles, $d$ the corresponding vector of travel times, and $G$ the intervening theory which involves the gravitational constant. (Galileo did not know about moments of inertia.) Allowances for inevitable errors in the experimental data are made with the introduction of the $e$ term. While the particular values of $e$ associated with any experimental trial are unknown a priori, any good scientific experiment involves some specification of its statistical properties.

The scientific method involves finding a theory $G$ that maps the controlled conditions $m$ into the data $d$ to an accuracy consistent with the anticipated laboratory errors $e$. In estimation theory, $G$ specifies the system, and so the scientific method is a system identification problem. (In estimation theory, $m$ is the state of the system, $e$ is the observation noise, and $d$ is the observable.) This is the problem scientists and engineers are usually trained to solve.

Once an acceptable system $G$ is found, predicting new data $d$ for new sets of conditions $m$ becomes a trivial exercise. This is called solving the direct problem, that is, moving directly from left to right through (1.1). The reward for the successful solution of the system identification problem is the direct problem, which can be applied repeatedly and reproducibly without the need for more analysis.

However, for every science and engineering professional using the scientific method and working on the system

identification problem, there are a multitude of professionals at any given time working on the inverse problem [1]. Except for frontier areas of research, the fundamental theories required to predict new experimental data are substantially in place and validated. The primary task for most scientists and engineers working routinely is to use the established body of theory to interpret new data as they become available. In the inverse problem, one begins with data $d$ and some prescription of the statistical properties of $e$ and moves from right to left through (1.1), trying to ascertain the physical conditions $m$ that prevailed during the experiment or measurement. The inverse problem is generally much more difficult to solve than the direct problem and sometimes as difficult as the system identification problem.

Only a small proportion of science and engineering students receive explicit training in inverse problems.

Inverse problems are pervasive and lurk behind nearly every task in science and engineering. For example, every scientific instrument has an instrument function which maps the "true" conditions sampled by the instrument to its actual output. The instrument function could represent some kind of spatio-temporal average, hysteresis, distortion, or bias. In the context of (1.1), the instrument function is $G$ and the true environmental conditions are $m$. Accounting for an instrument function is an example of an inverse problem and is one that accompanies any kind of measurement. The instrument function of a camera, particularly one that is in motion, gives rise to image blurring. and mitigating blurring in imagery is another classic inverse problem. Still another example is the widespread procedure of data smoothing. We can think of smoothed data as the model, unsmoothed data as data, departures from smoothness as errors, and the hypothesis $G$ as simply the identity. In this context, smoothing procedures become less ad hoc, more rigorous.

The inverse problem is generally more difficult to solve than the direct problem for three reasons. First, there is the question of existence. While a system may guarantee a mapping from actual conditions to idealized, error-free measurements, there is no guarantee of a reciprocal or inverse mapping from actual measurements (with errors) back to any set of experimental conditions, back to any model. For example, consider the problem of finding the line that passes through three non-collinear points. Second, there is the question of uniqueness. That is, there may be multiple models that correspond to the same set of measurements. That the data are a function of the model does not imply that the model is a function of the data. An example in this case is the problem of finding the parabola that passes through two points. Although seemingly counter-intuitive, existence and uniqueness issues are not mutually exclusive. For example, finding a line that passes through three data points has no solutions when the points are not collinear, one solution when they are, and multiple solutions when all the points are coincident.

Finally, there is the question of stability. For a system $G$ to be practically useful, it should be tolerant of small uncertainties in the controlled conditions or model $m$. That is, small relative variations in $m$ should map into even smaller relative uncertainties in $d$. If not, the scientific method would be hard to apply in practice, since small defects in the laboratory setting would tent to produce large model-data discrepancies and obscure the merits of essentially correct hypotheses. When moving through (1.1) from left to right, fluctuations should diminish. Galileo may never have worked out the law of gravitation had the dynamics of rolling balls depended gravely on their exact shape and uniformity. All the theories we know have, in some sense, been selected for this criterion!

Conversely, when moving through (1.1) from right to left, there is a tendency for relative fluctuations in the data to induce even larger relative fluctuations in the inferred model. When small but inevitable fluctuations in the experimental data give rise to excessively large fluctuations in the inferred model, the inverse problem is said to be unstable. This turns out to be a common feature of many inverse problems of interest. For example, consider the problem of finding the line that passes through a few, nearly-coincident points. Discrete inverse problems which are unstable are termed "ill conditioned" while unstable continuous inverse problems are termed "ill posed." Stability is often the main obstacle to solving inverse problems and has led many past investigators to abandon the attempt.

Instability is not a feature of the data but rather one of the system, although certain data will expose inherent instability more than others. Similar comments hold for existence and uniqueness issues. To understand and manage existence, uniqueness, and stability quantitatively, we must investigate the inverse problem more deeply.

[1] Author-manufactured statistic

## 1.1 Inverse problem classification

Inverse problems present themselves in a number of different forms. Linear and nonlinear problems involve linear and nonlinear mappings between the model and the data, respectively. Nonlinear problems are generally solved iteratively, with the problem being converted to a linear one (through a process called "linearization") at each iteration. We therefore begin with the consideration of linear problems, relegating nonlinear ones to chapters 8 and beyond.

Linear problems have two essential properties – superposition and scaling:

$$G(m_1 + m_2) = G(m_1) + G(m_2) \tag{1.2}$$
$$G(\alpha m) = \alpha G(m) \tag{1.3}$$

where $m_1$ and $m_2$ are different models and $\alpha$ is a constant. In linear systems, it is trivial to generalize the model and the data so as to represent vectors, vectors of vectors, etc.

Inverse problems can also be cast in terms of discrete or continuous variables or a combination. The general form of a linear discrete problem is:

$$\underline{\underline{G}}\,\underline{m} + \underline{e} = \underline{d} \tag{1.4}$$

where $\underline{\underline{G}}$ is a matrix and $\underline{m}$, $\underline{d}$, and $\underline{e}$ are vectors. We regard these vectors as real vectors belonging to a Hilbert space. If the number of data and model elements is $n$ and $m$, respectively, then $\underline{d} \in \mathbb{R}^n$, $\underline{m} \in \mathbb{R}^m$, and $\underline{\underline{G}} \in \mathbb{R}^{n \times m}$. (Later, we will generalize results for a complex vector spaces as needed.)

The general form for a continuous linear inverse problem is:

$$\int_a^b G(s,x)m(x)\,dx + e(s) = d(s) \tag{1.5}$$

which is a Fredholm integral equation of the first kind. Here, we see that $G(s,x)$ plays the role of an averaging kernel. If the model and data can be related through another linear operator $L$ such that $L(d(x) - e(x)) = m(x)$ (i.e. the inverse of the original problem), then $G(s,x)$ can be seen as the Green's function for that operator. A common example of such a problem is a convolution integral:

$$\int_{-\infty}^{\infty} G(s-x)m(x)\,dx + e(s) = d(s) \tag{1.6}$$

Indeed, deconvolution is a prototypical inverse problem, one of the primary motivations for our study. Other kinds of integral transfomrs include Fourier, which has a close relationship to convolution, Hilbert, Abel, and Radon transforms, to name a few. Many common integral transforms in science and engineering have well-known inverses.

A special case of the Fredholm equation arises when $G(s,x) = 0$ for $x > s$. The removal of the unnecessary part of the integral results in the Volterra equation of the first kind:

$$\int_a^s G(s,x)m(x)\,dx + e(s) = d(s) \tag{1.7}$$

such that the independent variable $s$ becomes the upper limit of integration. We will see that this problem is particularly amenable to inverse methods so long as the size of the error term is modest. The Volterra equation also becomes a convolution integral if the argument of the kernel function is replaced by $s - x$. This form of the convolution operator is closely related to the product rule for Laplace transforms.

Inverse problems can also be classified as being either deterministic or probabilistic. The discussion thus far has regarded the model and data as deterministic quantities to be estimated on the basis of different imperatives. However, the model and the data can also be regarded as random variables. In that case, the objective becomes the estimation of the probability density function of either. Although the two perspectives appear to be quite different, the results are essentially equivalent in the case of normally distributed random variables, as will be seen.

## 1.2 Existence, uniqueness, and stability

A solution to the inverse problem will not exist if the theory admits no mapping from the measured data back to the model. In the case of discrete problems, we can think of $\underline{\underline{G}}$ as having a column space that does not span $\mathbb{R}^n$ and therefore may not contain the given data vector. No model vector can reproduce the data vector in that case. In the case of continuous problems, we can think of $G(s,x)$ as not spanning the function space of $d(s)$. For example, if $G(s,x)$ in (1.5) has no $s$ dependence, then solutions for $m(x)$ only exist when $d(s)$ is a constant.

The solution will not be unique if the theory admits multiple mappings from the measured data back to the model. In the discrete case, we can think here of $\underline{\underline{G}}$ as having a nontrivial nullspace such that $\underline{\underline{G}}(\underline{m} + \alpha \underline{m}_\circ) = \underline{\underline{G}}\underline{m} = \underline{d}$, which is to write that multiples of vectors in the nullspace can be added to the model without affecting the data prediction. For an example of the continuous case, consider what happens for $G(s,x) = f(s)\sin(m\pi x)$ with $a = 0, b = 1$. The functions $\sin(n\pi x)$ are orthogonal on the interval [0,1]. One could therefor add any linear combination of such functions with $n \neq m$ to any candidate solution for $m(x)$ without altering the data prediction.

Both conditions can be true at the same time, i.e., some components of the data may not map to a model at all whereas other components may have multiple mappings. This occurs in the discrete case when system matrices are rank deficient.

Finally, datasets that differ only slightly may map to very different models. This last condition signifies instability and implies that small measurement errors will telegraph to large model errors. Models with extravagant features may be necessary to account for even small, spurious features in the data. This phenomenon underlies the perils of what is sometimes called "fitting the noise," attempting to assign physical significance to it. For the case of square matrices $\underline{\underline{G}}$, instability occurs when the matrix is nearly singular. A more general criterion applicable to system matrices that are not square will be developed in successive chapters.

For continuous problems, consider that

$$\lim_{k \to \pm\infty} \int_a^b G(s,x)\exp(ikx)\,dx \quad = \quad 0 \tag{1.8}$$

for all Riemann-integrable functions $G(s,t)$ on [a,b]. This is the Riemann-Lebesgue Lemma. What this implies is that one may be able to add highly oscillatory functions to candidate model solutions without greatly affecting the data prediction. Direct problems consequently tend to be somewhat immune to oscillatory model fluctuations,to the good fortune of laboratory scientists. Conversely, inverse problems will have a tendency to produce model estimates with spurious "ringing." Suppressing this ringing without unduly compromising the solutions is the goal of the data analyst..

A useful analogy regarding instability can be drawn between inverse problems and conspiracy theories which arise for the following reasons: 1) Life is messy. 2) Information is flawed. 3) It is always possible to find a story that accounts for the anecdotal information, such as it is, better than the true one. Such a story may contain outlandish characters and plots as are necessary to account for certain pieces of spurious or "noisy" information. By analogy, 1) measurements are complicated, 2) data are flawed as a rule, and 3) it is generally possible to find models that are more consistent with measured data than the true model. Such models may nonetheless be very inaccurate and need to be depreciated.

As will be seen in subsequent chapters of this text, the existence problem can be reconciled by seeking model solutions that are the most consistent with the available data by some measure. The uniqueness problem can be addressed by seeking solutions which are most consistent with preferences imposed from outside, thereby reducing the space of candidate solutions. Solutions which violate certain conservation laws, for example, could be ruled out, as could solutions that exhibit unjustifiable complexity. The same strategy will also be used to mitigate instability, with spurious features in the model being suppressed unless they have unambiguous support in the data. Both strategies will work together in concert. The goal will be to find model solutions that account for the data acceptably well without including extraordinary features rather than accounting for the data as closely as possible.

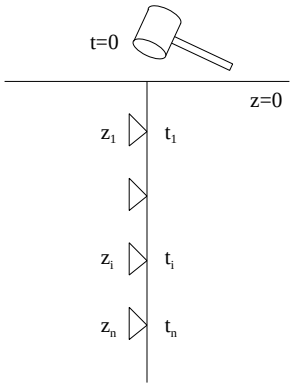


Figure 1.1: Diagram of a vertical seismic profiling experiment.

## 1.3 Some examples

The prevalence of inverse problems is best demonstrated by example. Below, a number of example problems drawn from the natural sciences are outlined. These include discrete and continuous, linear and nonlinear problems. Some problems can be stated simply while others require more background and motivation. Even simple problems can be highly troublesome, however. Throughout the course of this text, the various problems will be reexamined and addressed.

### 1.3.1 Vertical seismic profiling

A vertical seismic profiling experiment is diagrammed in Figure 1.1. A borehole is sunk into the ground starting at a depth $z = 0$. Within the borehole are placed $n$ seismic sensors at different positive depths $z_i$, $i = 1, \ldots, n$. At time $t = 0$, an impulse at the surface launches mechanical waves which propagate downward. The passage of the wave front is then detected by sensors at different depths $z_i$ at positive times $t_i = t(z_i)$.

The group speed of the wave front depends on depth, and the physical property under scrutiny is the "slowness" versus depth, $s(z)$, which is the time it takes for the wave front to propagate downward a distance of one unit. The time it takes for the wave front to be detected at an arbitrary depth $z$ is then:

$$t(z) = \int_0^z s(\xi)\, d\xi \tag{1.9}$$

$$= \int_0^\infty s(\xi) H(z-\xi)\, d\xi \tag{1.10}$$

where $H(z)$ is the Heaviside step function, introduced here to put the direct problem into the form of (1.5). According to the fundamental theorem of calculus, it is clear that $s(z) = dt(z)/dz$, suggesting one approach for inverting experimental data. This would be an acceptable strategy in the absence of noise or experimental uncertainty. In practice, differentiation will tend to exacerbate fluctuations due to noise and uncertainty, just as integration tends to suppress fluctuations (by the Riemann-Lebesgue lemma). Simple differentiation may therefore turn out to be impractical.

An complication arises from the fact that travel-time data are acquired only at certain discrete depths $z_i$, raising questions about how the derivatives are to be calculated. The problem is complicated further if the distances between the sensors are nonuniform. One way to proceed would be to break the spatial domain into $m$ equally-spaced intervals $\delta\xi$ wide and to discretize the model so that its elements $s_j$, $j = 1, \ldots, m$, are simply collocated with these intervals. In matrix form, the problem becomes:

$$t = Hs + e \tag{1.11}$$

where $t, e \in \mathbb{R}^n$, $s \in \mathbb{R}^m$, and $H \in \mathbb{R}^{n \times m}$. Here, an expression for experimental error has been reintroduced to the problem formally. The elements of $H_{ij}$ are $\delta\xi$ for $z_i > j\delta\xi$ and zero otherwise and represent the distance traveled by a

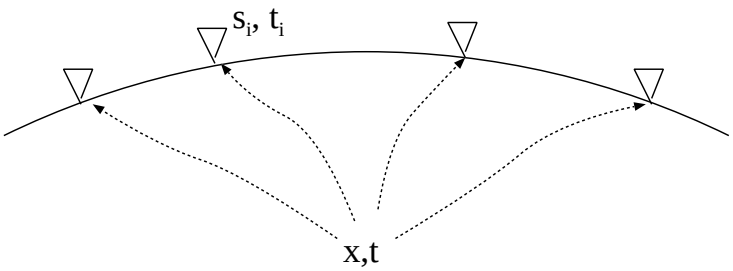


Figure 1.2: Diagram of a generic seismography experiment.

ray passing through the $j$th spatial interval on its way to the $i$th seismic recorder. This is obviously a linear problem, made so by the simple, one-dimensional geometry of the experiment.

Note that the vectors and matrix in (1.11) have been written without underbars. This will be the convention throughout the remainder of the text.

### 1.3.2 Multi-dimensional seismography

A variant of the seismic data interpretation problem is illustrated in Figure 1.2. This time, $n$ seismic receivers are stationed at different places on the earth's surface. The hypo center of an earthquake is located at some coordinate in two- or three-dimensional space $x$ at at time $t$. Receivers located at position $s_i$, $i = 1, \ldots, n$, detect the resulting mechanical wavefront at time $t_i$. The problem this time is to determine $x$ and $t$ which comprise the model vector. The elements of the data vector are the $t_i$, $i = 1, \ldots, n$.

This problem is essentially more difficult than the vertical seismic profiling problem. Even in the event that the slowness is everywhere uniform and the seismic waves travel along straight paths radially away from the hypo center, the relationship between the hypo center coordinates and the detection times is governed by the Pythagorean theorem and is therefore nonlinear. There is consequently no matrix representation for the mapping between $m$ and $d$. Estimating $m$ will involve an iterative procedure starting from an initial guess for the hypo center coordinates. Whether the iteration will converge could in principle depend on the quality of the initial guess for the solution.

Allowances for spatial variation in the slowness and for refraction complicate the direct problem considerable but do not necessarily make the inverse problem intractable. One approach is the following. The slowness throughout the medium could be specified in terms of a finite set of discrete parameters, perhaps the amplitudes of appropriately chosen basis functions or splines.

Numerous rays could be sent in all directions from a candidate hypo center space-time coordinate. The trajectories of the rays could be governed by the equations of geometric optics, physical optics, or another appropriate model. Those rays passing very near the seismic recorders would covey predictions about wavefront arrival times. Iteratively, the hypo center could then be moved and the raytracing re-performed until the predicted and measured arrival times demonstrate satisfactory agreement. In the case that the slowness throughout the medium are not known a priori, their parametrization too could be updated iteratively in pursuit of optimal agreement with measurement. Uniqueness of solution would undoubtedly be an issue, requiring the imposition of outside preferences for closure.

### 1.3.3 Travel-time tomography

Keeping with the theme of material slowness estimates, we turn to the example of travel-time tomography. Figure 1.3 shows a regular assembly of blocks, each with its own distinct but individually uniform travel time. The size of each block is $L \times L$. Consider mechanical waves traveling along straight lines in such a way that the total propagation delay time $T$ is only affected by the blocks through which each ray passes. The figure illustrates this with rays numbered one through ten – one ray for each row, one for each column, a ray that passes only through the block in the upper-left

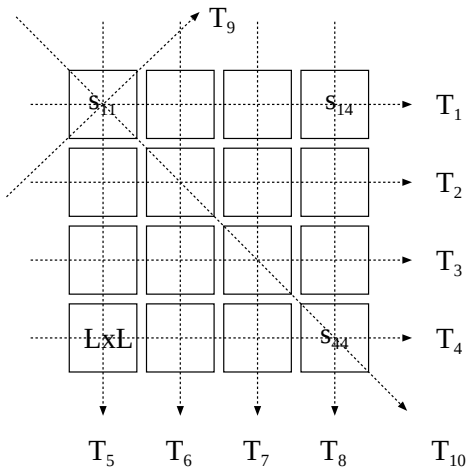


Figure 1.3: Diagram of a travel-time tomography experiment.

corner, and a final ray that passes through the main diagonal. The elements of the data vector for this problem are the ten measured travel times. The elements of the model vector are the sixteen block slownesses.

The system describing this linear experiment and can be expressed as:

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{10} \end{bmatrix} = L \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \\ \vdots \\ s_{14} \\ s_{21} \\ \vdots \\ s_{44} \end{bmatrix} \tag{1.12}$$

The problem is not invertible directly as posed. There are ten equations implied by (1.12) but sixteen unknowns. The problem is clearly under determined. Even if more ray paths were incorporated in the problem and the system matrix in (1.12) were square, there is no guarantee that the underlying equations would be linearly independent. Even if the system matrix were non-singular and invertible, there is still the possibility that it could be nearly singular by some measure. This would invite potentially large variations in the slowness estimates arising from even small errors in the travel-time measurements.

So solutions to the travel-time tomography problem posed in Figure 1.3 are not unique. However, it certainly seems as though the slowness of the block in the upper-left corner of the system should be uniquely determined and that this should have some benefit for the determination of the slowness along the first row and column. Given knowledge of $T_9$, permissible values of $T_1$, $T_5$, and $T_{10}$ are limited (to values grater than $T_9$ times a constant). If these conditions are not met, then no solution to the problem may exist. This implies that the system is over determined in part while also being under determined. What is needed is a mechanism for exposing which parts of the system are over determined and which are under determined and then for coping with both. We will see that singular value decomposition is just the tool for accomplishing this.

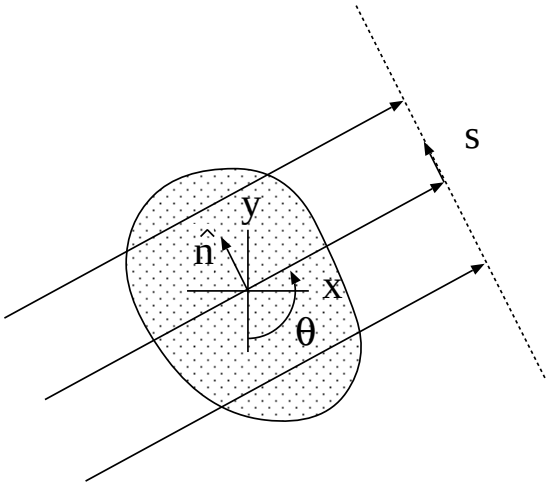


Figure 1.4: Geometry of a CAT scan.

### 1.3.4 Computed axial tomography – CAT scans

A closely-related kind of problem involves interrogating the interior of bodies using X-ray absorption. As X-rays propagate along straight line paths through materials, they are absorbed according Lambert's law: $dI/d\xi = -FI$, where $I$ is the intensity of the rays, $F$ is the absorption cross section at some location on the path, and $\xi$ is a ray path element. The solution has the form $I = I_\circ \exp(-\int F d\xi)$. Expressed as a decibel (dB) quantity, the intensity therefore decreases linearly with distance along the path in proportion with the absorption rate $F$. This is essentially the same problem as in travel-time tomography, only with intensity reduction (in minus dB) replacing total travel time and absorption replacing slowness.

Figure 1.5 shows an experimental configuration for estimating the absorption cross section within a body in two dimensions. Parallel X-rays are beamed through the body at an angle $\theta$ as shown, and the terminal intensity of the rays is detected and recorded on a screen as a function of position $s$. By rotating the apparatus around the body, data can be collected as a function of $\theta$ and $s$. Those data are related to the absorption within the body by

$$d(\theta, s) \quad = \quad \iint F(x,y)\delta(x\cos\theta + y\sin\theta - s)\,dx\,dy \tag{1.13}$$

where the integral is over all two-dimensional space and the Dirac delta function confines the contributions to the integral to a single ray path. Note that the third spatial dimension can be incorporated trivially into CAT scans by considering one cross-sectional slice at a time.

The transformation in (1.13) has the form of (1.5) and is known as a Radon transform, after X-ray pioneer Johann Radon. The datasets produced in this way are known as sinograms because an individual point target in the body understudy produces a sine wave in $s$ — $\theta$ space. This is easily seen by replacing $F(x,y)$ in (1.13) with $\delta(x-x_\circ)\delta(y-y_\circ)$, in which case $d(\theta,s)$ is only nonzero when $s = r_\circ \sin(\theta+\theta_\circ)$, $r_\circ$ and $\theta_\circ$ being the polar coordinates associated with $x_\circ$ and $y_\circ$. So, a sinogram is a superposition of sine waves, each uniquely indicating the position of a point-target absorber within the body.

With the help of some properties of Fourier transforms, it will later be shown to be possible to invert the Radon transform analytically. As with a Fourier transform then, the Radon transform is continuous and has a continuous inverse that can be expressed explicitly. The solution found through inverse transformation exists and is unique, and the algorithm is well-posed and stable. In practice, modern sinograms are acquired using digital imaging technology on a discrete grid, and so the Radon transform and its inverse need to be formulated in discrete form. The problem is common to most contemporary experimental fields, and so methods of discretizing continuous problems will have to be treated in some detail.

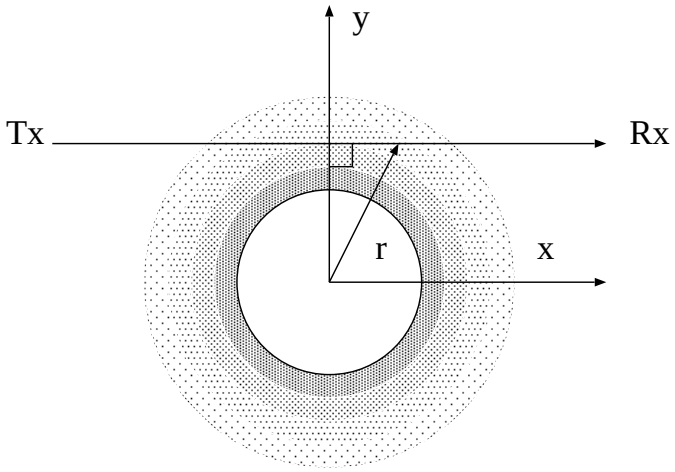


Figure 1.5: Radio occultation experiment geometry.

### 1.3.5 Abel transform and its inverse

The primary difference between the travel-time tomography and CAT scan examples was the experimental geometry. It is illustrative to consider a third geometrical variant on the problem, as is illustrated in Figure 1.5. We consider here a radio occultation experiment, wherein a source and a detector are connected by a straight-line ray passing through a medium which is circularly or spherically symmetric. The source could be a radio transmitter on a spacecraft, the detector being a radio receiver on a separate spacecraft. The medium could be the Earth's atmosphere or ionosphere, each having an index of refraction that varies with altitude alone to zeroth order. The receiver measures the group delay of the radio signal which is indicative of a line-integrated property in either the atmosphere (water vapor content) or the ionosphere (electron number density).

Without loss of generality, a ray path can be described by its impact parameter $y$. As the satellites move, the impact parameter changes. The data acquired by the receiver are then of the form

$$d(y) = \int_{-\infty}^{\infty} m(r)\,dx|_y \tag{1.14}$$

$$= 2\int_{0}^{\infty} m(r)\,dx|_y \tag{1.15}$$

$$= 2\int_{y}^{\infty} \frac{m(r)r\,dr}{\sqrt{r^2-y^2}} \tag{1.16}$$

where it is assumed that the physical quantity under study $m(r)$ decreases more quickly with altitude than $r^{-1}$. While (1.16) has been derived in the plane, it is equally applicable in a three-dimensional geometry when the variable $y$ should be replaced with $s=\sqrt{y^2+z^2}$. The integral transform thus derived is known as the Abel transform.

The Able transform has an explicit inverse which can be written as:

$$m(r) = -\frac{1}{\pi}\int_{r}^{\infty} d'(y)\frac{dy}{\sqrt{y^2-r^2}} \tag{1.17}$$

where the prime denotes differentiation with respect to the argument. To show that this is the inverse, begin by applying integration by parts to the Abel transform itself, noting that the boundary terms vanish:

$$d(y) = -2\int_{y}^{\infty} m'(r)\sqrt{r^2-y^2}\,dr \tag{1.18}$$

$$d'(y) = 2y\int_{y}^{\infty} \frac{m'(r)}{\sqrt{r^2-y^2}}\,dr \tag{1.19}$$

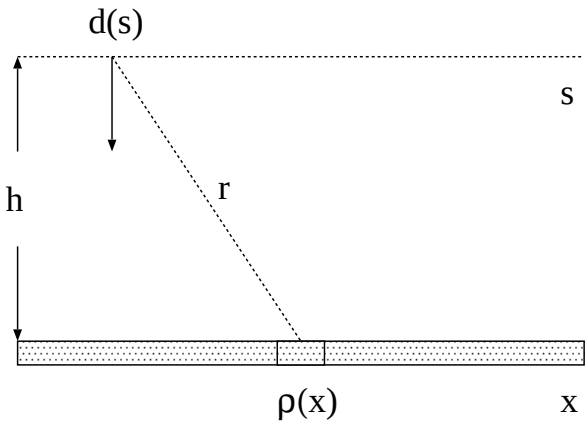


Figure 1.6: Diagram of a gravity anomaly investigation.

where it can be noted too that the term associated with differentiation with respect to an integral boundary at $r = y$ also vanishes. Substituting this result into (1.17) yields (after replacing $r$ with $\tilde{r}$ to avoid confusion):

$$m(\tilde{r}) = -\frac{1}{\pi}\int_{\tilde{r}}^{\infty}\int_{y}^{\infty}\frac{2y}{\sqrt{(y^2-\tilde{r}^2)(r^2-y^2)}}m'(r)\,drdy \tag{1.20}$$

$$= -\frac{1}{\pi}\int_{\tilde{r}}^{\infty}\int_{\tilde{r}}^{r}\frac{2y}{\sqrt{(y^2-\tilde{r}^2)(r^2-y^2)}}m'(r)\,dydr \tag{1.21}$$

$$= \int_{\tilde{r}}^{\infty}(-1)m'(r)dr \tag{1.22}$$

$$= m(\tilde{r}) \tag{1.23}$$

And so the Abel transform joins the Fourier and Radon transforms and many others in having an explicit analytic inverse. However, unlike the other examples, the inverse Abel transform is not well posed and has a tendency for instability. This can be appreciated by noting the derivative in the definition (1.17). The implications will be pursued later in the text.

One application for satellite-based radio occultation experiments is the estimation of electron density altitude profiles in the ionosphere from radio group delay measurements, which are indicative of the total electron content (TEC) between the transmitter and receiver. Group delay is one aspect of refraction. Another aspect, neglected in the above discussion, is ray bending. The bending angle or radio waves propagating between satellites can also be measured because it influences the Doppler shift of the received signal. Bending angles are the observables mainly used to infer altitude profiles of the index of refraction in the air in the lower atmosphere. The index of refraction is useful to know here, being indicative of water vapor content. Amazingly, the relationship between bending angle and water vapor content in radio occultation experiments is given not by (1.16) but instead by (1.17). This is a well-posed integral transform, not at all prone to instability. Consequently, it is much more straightforward to infer water vapor profiles than electron density profiles from radio occultations.

### 1.3.6 Gravity anomaly

Another prototypical inverse problem concerns the identification of bodies buried beneath the ground on the basis of variations they cause in the local gravity field (gravity anomalies). A simple scenario is illustrated in Figure 1.6. Here, a pipe is buried at a uniform depth, $h$. The mass density of the pipe varies along its length as $\rho(x)$. At the surface, an observer carried a mass suspended by a spring along a path above the pipe and notes its downward displacement $d(s)$. The mass responds to the vertical component of the weight force exerted on it by the buried mass. The formula

relating the displacement to the mass density is:

$$d(s) = G\int_{-\infty}^{\infty} \rho(x)\frac{h/r}{r^2}\,dx \tag{1.24}$$

$$= G\int_{-\infty}^{\infty} \rho(x)\frac{h}{\left[(s-x)^2+h^2\right]^{3/2}}\,dx \tag{1.25}$$

In this formula, $G$ is the gravitational constant, the factor of $h/r$ is the fraction of the weight force that is in the vertical direction, and the Pythagorean theorem has been used in expressing $r$ in terms of $s$ and $x$. The direct problem is linear and in the form of (1.5). We can see that the kernel of the integrand acts as a kind of averaging or smoothing operator, causing the displacement to be gradually varying even if the mass density varies sharply, particularly if $h$ is large.

It is illustrative to consider a minor variation to the problem, making the mass density of the buried pipe a constant but allowing the depth $h$ to vary with position $x$ instead. In that case, the direct problem becomes

$$d(s) = G\rho\int_{-\infty}^{\infty} \frac{h(x)}{\left[(s-x)^2+h(x)^2\right]^{3/2}}\,dx \tag{1.26}$$

Now, the problem is essentially more difficult, as the relationship between the data $d(s)$ and the model $h(x)$ is nonlinear. This problem turns out to be highly unstable, with any number of widely-differing models $h(x)$ being able to reproduce given datasets $d(s)$, To proceed with the solution, it will generally be necessary to reduce the candidate solution space through the imposition of outside information and preferences.

### 1.3.7 Atmospheric optical spectography

One of the easiest ways to explore processes occurring in the upper atmosphere is the observation of airglow – optical emissions from constituent atmospheric gasses. Airglow results from the deactivation of excited states of different atoms and molecules. Many optical emissions come from excited states of atomic oxygen, for example, but other species generate airglow too. The states can be excited through impacts with energetic free electrons at altitudes where ionization is significant. Airglow is observed in spectral lines with wavelengths corresponding to the energies of the various excited states. By studying the intensity of the various lines, clues about the underlying energetic electron population can be gleaned.

For a state to be excited in a constituent, the energy of the impacting electron must be greater than the energy of the state, the threshold energy. The efficiency of the excitation of the state by electrons with energy $E$ is governed by the cross section $\sigma(E)$ for a given state and constituent. The number density of electrons at a given energy $E$ and altitude $z$ is specified by the distribution function $f_e(z,E)$.

The intensity of the emission corresponding to the radiative deactivation of the $i$th state excited by electron impact on the $j$th neutral constituent can then be predicted using the standard formula

$$I_i = 10^{-6}\int_{\text{ionosphere}} C_{ij}(z)n_j(z)\int_{E_{ij}^{th}}^{\infty} v_e(E)\sigma_{ij}(E)f_e(z,E)\,dE\,dz$$

where $I_i$ is the intensity in Rayleighs, $C_{ij}$ is the ratio of the Einstein coefficient to the total deactivation probability of the $i$th state of the $j$th neutral constituent, $n_j$ is the density of the $j$th neutral constituent, $v_e$ is the electron velocity, $E_{ij}^{th}$ is the excitation threshold, $\sigma_{ij}$ is the cross section for the excitation of the $i$th state, and $f_e(z,E)$ is the electron energy distribution. Here, the energy is in electron volts, and the remaining quantities are in cgs units.

We can regard everything in the integrand of (1.27) as being known except the electron energy distribution $f_e(z,E)$. Clearly, different optical emissions represent different moments of this function. Estimating it on the basis of observations of a number of different airglow line intensities $I_i$ is an inverse problem. This is a hybrid problem, since $f_e(z,E)$ is continuous and $I_i$ is discrete. That the electron distribution function is two dimensional makes the problem rather strongly under determined.

The problem becomes simpler if $f_e(z,E)$ can be related to some other one-dimensional quantity of interest. For example, suppose that in equilibrium, the electron distribution can be related linearly to a production function according

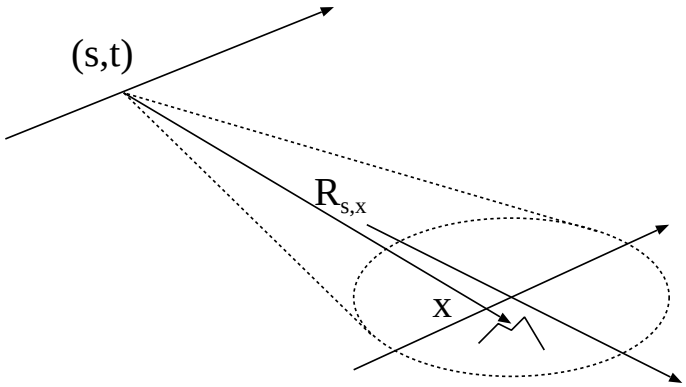


Figure 1.7: Geometry for a synthetic aperture radar experiment.

to:

$$f_e(z,E) \quad = \quad \int G(z,E;z_\circ,E_\circ)Q(z_\circ,E_\circ)dE_\circ \tag{1.27}$$

where $Q$ represents the production of energetic electrons with energy $E_\circ$ due to a source at altitude $z_\circ$ and $G$ is a Green's function, specifying the resulting, steady-state electron distribution versus energy and altitude. This approach would be useful in cases where it is known that energetic electron production is occurring primarily at one altitude for some reason. Substitution of this formula into the direct problem (1.27) reduces it to one of the form:

$$I_i \quad = \quad \int g_i(z_\circ,E_\circ)Q(z_\circ,E_\circ)\,dE_\circ \tag{1.28}$$

where $g_i$ contains all of the attributes of the problem except for the production function, which is the new unknown in the inverse problem. The problem remains under determined but is closer to being tractable.

### 1.3.8 Synthetic aperture radar (SAR)

Synthetic aperture radars or SARs are prime candidates for the application of inverse methods. SARs are deployed on moving vehicles and are used to observe stationary targets. Different targets in the radar field of view will have different Doppler shifts according to the formula $\omega = \mathbf{k}\cdot\mathbf{v}$, where $\omega$ is the Doppler shift, $\mathbf{k}$ is the radar scattering wavevector, and $\mathbf{v}$ is the relative velocity of the target with respect to the radar. While the Doppler shift of the target does not uniquely identify its position, the time history of the Doppler shift does.

Consider a radar moving over a planar surface filled with stationary, reflecting targets, which it illuminates. The radar illumination occupies a range of frequencies $\omega$ as determined by the carrier frequency and the pulse characteristics. The relationship between the received radar signal and the surface reflectivity can be expressed as:

$$d(s,t) \quad = \quad \int e^{-i\omega(t-2R_{s,\mathbf{x}}/c)}A(\omega,s,\mathbf{x})V(\mathbf{x})\,d\omega d^2x \tag{1.29}$$

Here, $d(s,t)$ represents the data, the signal registered by the radar receiver at time $t$ and position $s$ along the vehicle trajectory. The surface reflectivity in the plane is represented by $V(\mathbf{x})$. The function $A(\omega,s,\mathbf{x})$ includes influences on the signal amplitude including the range dependence and the spectral shape of the radar illumination. The term $R_{s,\mathbf{x}}$ is the distance between the radar and the coordinate $\mathbf{x}$. The term in the exponent, $2i\omega R/c$ is the phase of the echo arriving from that coordinate. An inverse Fourier transform transforms the signal from the frequency to the time domain. Finally, the spatial integral sums the returns from all the targets in the plane. This is the direct problem for SAR.

We can propose that an estimate of the surface reflectivity can be formed from a similar linear transformation operating on the data:

$$\hat{V}(\mathbf{x}') \quad = \quad \int e^{i\omega'(t-2R_{s,\mathbf{x}'}/c)}Q(\omega',s,\mathbf{x}')d(s,t)d\omega' dsdt \tag{1.30}$$

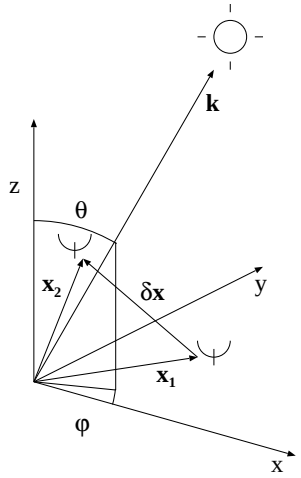


Figure 1.8: Geometry for an aperture-synthesis radar. Two antennas are shown, but multiple antennas with multiple pairings would be used for radar imagery.

where $Q(\omega, s, \mathbf{x}')$ is a function that must be determined. To proceed, substitute (1.29) into (1.30). The time interval can be performed immediately, evolving a factor of $\delta(\omega - \omega')$. After performing the $\omega'$ integral, we are left with:

$$\hat{V}(\mathbf{x}') = \int \underbrace{e^{i2k(R_{s,\mathbf{x}'} - R_{s,\mathbf{x}})} Q A d\omega ds}_{K(\mathbf{x},\mathbf{x}')} V(\mathbf{x}) d^2x$$

where $k = \omega/c$. Now, if the kernel function $K$ can be made to approximate a Dirac delta function, $\delta(\mathbf{x} - \mathbf{x}')$, through the right choice of $Q$, then the approximator for the surface reflectivity will be a good one. In fact, it is nearly in the correct form already. This can be appreciated with the application of the method of stationary phase. The exponential term in the kernel is highly oscillatory, and the greatest contribution to the integral will come from regions in $\omega - s$ space where its argument is small. This will be where the two $R$'s are nearly the same, which is also where $\mathbf{x}$ and $\mathbf{x}'$ are nearly the same. Taylor expanding the argument in this region gives

$$\begin{aligned} 2k(R_{\mathbf{x}',s} - R_{\mathbf{x},s}) &\approx 2k(\mathbf{x}' - \mathbf{x}) \cdot \hat{R} \\ &\equiv (\mathbf{x}' - \mathbf{x}) \cdot \xi \end{aligned}$$

where $\xi$ is a new, two-dimensional auxiliary variable. Finally, transform from $\omega - s$ space to $\xi$ space with the aid of the Jacobian of the transformation:

$$K(\mathbf{x}, \mathbf{x}') = \int e^{i(\mathbf{x}' - \mathbf{x}) \cdot \xi} Q A \left| \frac{\partial(s, \omega)}{\partial \xi} \right| d^2\xi$$

Clearly, making $Q^{-1} = A|\partial(s,\omega)/\partial\xi|$ yields a kernel $K$ which performs like a Dirac delta function. Combined with (1.30), this specifies the transformation for converting raw SAR data into surface imagery.

### 1.3.9 Aperture-synthesis radar imaging

A related problem involves the receipt of radio signals on the ground and the subsequent imaging of the radio sources in the cosmos that created them. This is the problem of very long baseline interferometry or VLBI. The idea is to use multiple, distantly-spaced receivers on the ground to synthesize an aperture for forming the image. This turns out to be a linear inverse problem.

Consider the voltage received by a receiver at location $\mathbf{x}_1$ as being due to a collection of plane waves. Each plane wave has a wavevector $\mathbf{k}$ with a wavenumber $k = 2\pi/\lambda = |\mathbf{k}|$. The radio receiver is tuned to a particular wavelength

$\lambda$ and, hence, to a particular wavenumber. The amplitude of each plane wave is given by $E(\mathbf{k})$, and the phase by the product $\mathbf{k}\cdot\mathbf{x}$. Consequently, the received voltage can be expressed as

$$v(\mathbf{x}) \quad = \quad \int d\Omega\, E(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} \tag{1.31}$$

where the integral is over all differential solid angles in the sky $d\Omega$.

Suppose now that the signals from two spaced receivers are correlated according to:

$$\langle v(\mathbf{x}_1) v^*(\mathbf{x}_2)\rangle \quad = \quad \left\langle \int d\Omega\, E(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}_1} \int d\Omega'\, E^*(\mathbf{k}') e^{-i\mathbf{k}'\cdot\mathbf{x}_2} \right\rangle \tag{1.32}$$

Where the angle brackets denote an ensemble average in principle and a time average in practice. Here, we can regard the amplitudes of the various radio sources random variables. Furthermore, we can regard the amplitudes of signals arriving from different bearings as being statistically uncorrelated. Consequently, one of the integrals in the above equation can be performed trivially, leaving

$$\langle v(\mathbf{x}_1) v^*(\mathbf{x}_2)\rangle \quad = \quad \int d\Omega\, \langle |E(\mathbf{k})|^2\rangle e^{i\mathbf{k}\cdot(\mathbf{x}_1-\mathbf{x}_2)} \tag{1.33}$$

Finally, define the displacement of the two receiving stations as $\delta\mathbf{x} \equiv \mathbf{x}_1 - \mathbf{x}_2$. Further, define the correlation of the two signals as the visibility $V(\delta\mathbf{x}$, a function of the displacement, and the signal power arriving from a given bearing as the brightness distribution $B(\mathbf{k})$. Then,

$$V(\delta\mathbf{x};k) \quad = \quad \int d\Omega\, B(\mathbf{k}) e^{i\mathbf{k}\cdot\delta\mathbf{x}} \tag{1.34}$$

and the functional relationship between the brightness (the desired quantity) and the visibility (the measurement) can be seen to be a linear integral transform. By measuring the visibility using multiple pairs of receivers, information necessary to infer the brightness distribution of the sky is gathered. The brightness distribution is an image of the radio sources at the given radio wavelength.

At first glance, the integral transform in (1.34) might appear to be a Fourier transform. This would be the case if the integral were over all wavevectors, i.e., $d\mathbf{k}$. In fact, the integral is only over wavevector directions, with the wavenumber held constant. To clarify matters, it is helpful to expand (1.34) in Cartesian and spherical coordinates:

$$V(\delta\mathbf{x};k) \quad = \quad \int \sin\theta d\theta d\phi\, B(\theta,\phi) \exp(ik(\sin\theta\cos\phi\delta x + \sin\theta\sin\phi\delta y + \cos\theta\delta z)) \tag{1.35}$$

$$\quad = \quad \int d\eta d\xi \frac{B(\eta,\xi)}{\sqrt{1-\eta^2-\xi^2}} \exp\left(ik\left(\eta\delta x + \xi\delta y + \sqrt{1-\eta^2-\xi^2}\delta z\right)\right) \tag{1.36}$$

Here, we have defined $\eta \equiv \sin\theta\cos\phi$ and $\xi \equiv \sin\theta\sin\phi$ to be the direction cosines of the wavevector with respect to the cardinal $x$ and $y$ directions and used the Jacobian of the transformation ($J = \cos\theta\sin\theta$) in writing (1.36) after noting that $\cos\theta = \sqrt{1-\eta^2-\xi^2}$.

In the case of all the receivers being in a plane such that all the $d\delta z = 0$ and that the brightness is confined to regions where $\eta$ and $\xi$ are small, we have

$$V(\delta x, \delta y; k) \quad \approx \quad \int\int d\eta d\xi B(\eta,\xi) \exp(ik(\eta\delta x + \xi\delta y)) \tag{1.37}$$

which is in the form of a two-dimensional Fourier transform. The formula for an inverse Fourier transform is, of course, well known. However, how should the inverse problem of estimating the brightness be carried out in view of the fact that only a discrete set of visibilities are known, presumably on a sparse, irregular grid of points? This is an important inverse problem not only for radio astronomy but for spectral analysis of discretely-sampled data in general.

### 1.3.10 Factor analysis

Imagine acquiring a sample of material from a river delta. A mass spectrometer can be used to determine the amount of $m$ different major elements in the sample. However, those elements were carried into the delta in the form of a

number of different major compounds or minerals, each with its own breakdown of elements. The number of distinct compounds could be much less than the number of elements they carry, representing hidden, latent structure. Is there a way to identify the compounds from one or more samples? On the surface, the problem seems intractable, but inferences can be made with the help of intuition.

Consider taking $n$ different samples in different parts of the delta. The resulting data would occupy a matrix $D \in \mathbb{R}^{m \times n}$ with components corresponding to the amounts of the $m$ elements found in each of the $n$ samples. Suppose now that there are $p$ possible compounds that can be made of the $m$ elements. Then we can define a matrix $L \in \mathbb{R}^{m \times p}$ which gives the amounts of $m$ elements in the $p$ compounds. An obvious decomposition for $D$ would then be

$$D = LF + E \tag{1.38}$$

where $F \in \mathbb{R}^{p \times n}$ is the unobserved state of the system, the amount of $p$ compounds in each of the $n$ samples, and $E$ is the experimental error possessing the same dimensionality as $D$. The problem of factor analysis is to determine both $L$ and $F$. In that way, factor analysis combines the system identification problem with the inverse problem. In the parlance of factor analysis, $L$ is called the loading matrix, and the compounds in $F$ are called the factors.

The factor analysis problem clearly has no unique solution. Consider any matrix $T$ with an inverse $T^{-1}$. Inserting these into (1.38) gives

$$D = LT^{-1}TF + E \tag{1.39}$$

$$= L'F' + E \tag{1.40}$$

where $LT^{-1} = L'$ and $TF = F'$. Any solution to the problem implies a family of other solutions. The task is to find one that is favored on a priori physical grounds. Another complicating aspect of the factor analysis problem is the nonlinear relationship between the data, loadings, and factors.

The factor analysis problem can also be posed statistically. Consider this time the elements of $D$ to be $m$ random variables observed $n$ times. The elements of $F$ can likewise be considered as $p$ unobserved random variables realized $n$ times. Take the expectation of $D$ and $F$ to be $\mu_D$ and $\mu_F$, respectively, and regard the terms of $E$ as being independently distributed with zero mean. Then the factor analysis problem may be posed as

$$D - \mu_D = L(F - \mu_F) + E \tag{1.41}$$

where the loading matrix is a set of unknown constants understood to remain invariant across observations. The problem can be made more tractable by making further assumptions about the statistical properties of the random variables. In many cases, $F$ and $E$ are taken to be independent, normally-distributed random variables with zero mean, the covariance of $F$ is defined to be the identity, and the covariance of $E$ is the identity times a constant ($\psi$).

Factor analysis can be exploratory, meant to elucidate underlying structure in complex datasets, or confirmatory, meant to test hypothesis about such structure. The most common methods of solution are closely related to principle component analysis which will be examined later in the text.

## 1.4 References

## 1.5 Problems

# Part I: Explicit, discrete methods

# Chapter 2

# Minimum length methods

The text begins with an analysis of linear discrete inverse problems and explicit (versus implicit or iterative) methods of solution. These methods often consider the length of the data and model vectors in the Euclidean sense and so can be referred to as length methods. The chapter opens with the conventional definition of a matrix inverse and then generalizes the concept.

## 2.1 Explicit inverse: eigen decomposition

In the event that $G$ is a square matrix of full rank, an obvious means of solving (1.4) would seem to be to find the inverse of $G$. If $G$ is a square $(n \times n)$ matrix with $n$ linearly independent eigenvectors, then it may be factored as

$$G = Q\Lambda Q^{-1} \tag{2.1}$$

where $Q$ is a square $(n \times n)$ matrix whose columns are the eigenvectors of $G$ and $\Lambda$ is a diagonal matrix whose elements are the associated eigenvalues of $G$. $Q$ can be interpreted as a similarity transformation into a space where $G$ becomes diagonal. Not all matrices can be thus diagonalized; undiagonalizable matrices are said to be "defective."

If $G$ is real symmetric, then $Q$ is an orthogonal matrix with $n$ linearly-independent eigenvectors, and $Q^{-1} = Q^T$. In the case of complex matrices, if $G$ is normal $(GG^H = G^H G)$, $Q$ is orthogonal and $Q^{-1} = Q^H$. If $G$ is unitary $(GG^H = G^H G = I)$, all the eigenvalues will lie on a complex unit circle. If $G$ is Hermitian, the eigenvalues will be real. Also, the determinant of $G$ is the product of the eigenvalues, and the trace of $G$ is the sum of the eigenvalues.

If none of the eigenvalues of $G$ are zero, then $G$ is non-singular and has the inverse:

$$G^{-1} = Q\Lambda^{-1}Q^{-1} \tag{2.2}$$

where the elements of the diagonal matrix $\Lambda^{-1}$ are the reciprocals of the eigenvalues of $G$. This would seem to suggest a practical means for solving for the model $m$ in (1.4) for those cases where $G$ is diagonalizable. However, using this approach will generally be found to be impractical. The problems is the discrepancy between the model prediction $Gm$ and the actual data $d$ which inevitably suffer from distortion. The discrepancy is contained in the vector $e$ in (1.4). Even when $G$ is non-singular, some of its eigenvalues are apt to be much smaller than others, and these will have the effect in (2.2) of amplifying the error terms, causing disproportionate errors in the model estimates. Particularly problematic will be those portions of the error that project onto the eigenvectors with the smallest eigenvalues. The inverse in (2.2) will act as a kind of a filter, allowing some aspects of the experimental error to degrade seriously the model estimate. This is the hallmark of instability.

A possible solution to the problem involves reducing the influence of the small eigenvalues, either by making them larger artificually or by assigning them a minimum value when they fall below some floor. The same strategy can be applied even to the zero eigenvalues. Thus, the filter implied by (2.2) is reshaped. This strategy must be implemented

in such a way that important components of the problem are not discarded. If the expected noise floor for the data and then the model can be established, then the level of filtering could be set so that the fluctuations in the final model values are comparable to the noise floor.

As the case being considered here is a special case of the general discrete linear problem, we will not explore its solution further but will consider the more general case of rectangular systems.

## 2.2 Over determined problems: least squares

Consider now the case where $G$ has more rows than columns and $n > m = p$, where $p$ is the rank of $G$, the number of linearly independent rows and columns. The system matrix is full rank, and the columns are linearly independent. In terms of a system of equations, $G$ has more equations $n$ than unknowns $m$. The system is said to be over determined, and we cannot expect to find exact solutions for the model that accurately predict the data in general in this case as a rule.

Schematically, the situation is like this:

$$\underbrace{\left[\begin{pmatrix}\vdots\end{pmatrix}\quad\begin{pmatrix}\vdots\end{pmatrix}\right]}_{G}\underbrace{\left[\mathbb{R}^m\right]}_{m} \quad\approx\quad \underbrace{\left[\mathbb{R}^n\right]}_{d} \tag{2.3}$$

It is clear that the column space of $G$ does not span $\mathbb{R}^n$ and that the data $d$ cannot in general be reproduced by a linear combination of the columns of $G$, by any model vector $m$. Exact solutions may be possible in the case that $d$ just happens to reside within the subspace of $\mathbb{R}^n$ spanned by the columns of $G$, but those cases are special cases. In any case, no explicit inverse of $G$ exists.

The best that can be hoped for in this case is to find the model vector $m$ that is in some sense best. A natural definition for best might be the model that minimized the discrepancy between the predicted and measured data, that minimizes the model prediction error or "residual" $e = d - Gm$. We can look to various norms of the residual for metrics or cost functions which can be minimized. Cost, penalty, or loss functions are referred to properly as "objective functions."

Vector norms are discussed in the appendix and can serve as positive-definite objective functions. The norm most often applied to the residual is the L2 norm, the length of the vector in the Euclidean sense. This choice has some interesting properties which will be considered elsewhere in the text. Here, we consider the inverse method that results from finding the model vector $m$ that minimizes the square of the L2 norm which is termed "linear least squares" minimization.

$$\begin{aligned} m &= \underset{m}{\operatorname{argmin}} ||Gm-d||_2^2 && (2.4)\\ &= \underset{m}{\operatorname{argmin}} (Gm-d)^T (Gm-d) && (2.5)\\ &= \underset{m}{\operatorname{argmin}} m^T G^T Gm - m^T G^T d - d^T Gm + d^T d && (2.6) \end{aligned}$$

The solution for $m$ is found through ordinary differentiation. Since matrix differentiation will occur frequently throughout the remainder of the text, the following formulas for real matrices are provided as reminders:

$$\begin{aligned} \frac{d}{dx}(Ax) &= A && (2.7)\\ \frac{d}{dx}(x^T A) &= A^T && (2.8)\\ \frac{d}{dx}(x^T Ax) &= x^T(A+A^T) && (2.9) \end{aligned}$$

(where $A$ is a constant). The linear least-squares estimate for the model vector that minimizes the residual in the L2 sense is therefore:

$$m^{\text{est}} = (G^T G)^{-1} G^T d \tag{2.10}$$

where we note that $G^T G$ is a square symmetric matrix. Whereas $G$ was not invertible in this case, $(G^T G)^{-1} G^T$ will exist so long as the conditions stated as the start of the problem are met. This estimator does not solve the direct problem exactly but provides an estimated solution that is guaranteed to be unique. No guarantees regarding the stability of the solution are offered.

## 2.3 Under determined problems: model simplicity

Next, consider the case where $G$ has fewer rows than columns and $m > n = p$ so that the system matrix is still full rank, although the columns are no longer linearly independent. In terms of a system of equations, $G$ has fewer equations $n$ than unknowns $m$. The system is said to be under determined, and we can expect multiple solutions for the model vector $m$ that solve the direct problem and predict the data exactly.

Schematically, the situation is now like this:

$$\underbrace{\left[\begin{pmatrix}\vdots\end{pmatrix} \quad \begin{pmatrix}\vdots\end{pmatrix} \quad \begin{pmatrix}\vdots\end{pmatrix} \quad \begin{pmatrix}\vdots\end{pmatrix}\right]}_{G} \underbrace{\begin{bmatrix}\mathbb{R}^m\end{bmatrix}}_{m} \approx \underbrace{\begin{bmatrix}\mathbb{R}^n\end{bmatrix}}_{d} \tag{2.11}$$

In this case, the columns of $G$ span $\mathbb{R}^n$ so that any data $d$ can be reconstructed from them. Since the columns are not linearly independent, however, reconstruction will not be unique in general, and so multiple solutions for $m$ may exist. Certain data vectors may have unique solutions if they have finite projections into $n$ or fewer columns, but these are special cases.

The problem now is to select one model solution from the many possible. For this, one or more preferences from outside the problem statement have to be imposed. One possibility would be to find the model vector consistent with the data (with zero residual) that is in some sense the simplest. This approach is consistent with Occam's Razor or the "law of parsimony" which states literally that "entities should not be multiplied unnecessarily" and conveys the idea that, among hypotheses which make accurate predictions, the one with the fewest assumptions should be favored. If the length of the model vector can be taken as a measure of simplicity (or the lack thereof), then the prescription for solving the under determined problem might be:

$$m = \underset{m}{\operatorname{argmin}}\ m^t m + 2\lambda^T (Gm - d) \tag{2.12}$$

Here, $m^T m$ is the Euclidean length of the model, which is to be minimized. The minimization involves a constraint, namely that the residual vector $Gm - d$ be zero. The constraint is imposed with the introduction of a Lagrange multiplier $\lambda$ which is here a column vector. For a review of variational mechanics and Lagrange multipliers, consult the appendix.

Applying the rules of differentiation from the last section yields a relationship between the model and the Lagrange multiplier:

$$m^T + \lambda^T G = 0 \tag{2.13}$$

This can be solved for the model $m$ which can then be substituted into the direct problem to yield an expression for $\lambda$. Substituting that expression into the equation above then yields the model simplicity solution:

$$m^{\text{est}} = G^T (G G^T)^{-1} d \tag{2.14}$$

which is guaranteed to find the model vector that solves the direct problem exactly while having the shortest Euclidean length. The term $G^T (G G^T)^{-1}$ is guaranteed to exist this time so long as the aforementioned conditions are met. The model simplicity method resolves the ambiguity inherent in the under determined problem by expressing a preference for the model above and beyond the requirement for consistency with the data. In this case, the preference was positive definite.

Nothing in the preceding analysis addressed the stability of the model simplicity solution. Intuitively, we might expect the approach to be stabilized by the imposition of prior information which might tend to suppress spurious model features lacking explicit support in the data. The problem will have to be considered more deeply to evaluate this premise, however.

## 2.4 Mixed determined problems: damped least squares

Counter-intuitive as it sounds, a problem can behave as if it is under determined and over determined at the same time. This occurs when but $n$ and $m$ are both greater than the rank of the matrix, $p$, i.e., when the system matrix is rank deficient. This means that neither the rows nor the columns of $G$ are linearly independent. Since the column space of $G$ does not span $\mathbb{R}^n$, no exact solution to the direct problem is guaranteed to exist for arbitrary data $d$. Since the columns of $G$ are not linearly independent, there will in general be multiple models $m$ that yield the same model prediction. While it may be possible to reproduce some data uniquely and exactly with the appropriate model, these are special cases.

The mixed determined problem is the worst of all possible worlds – no model is just right, buy many models are about equally good. Many inverse problems of interest are of this variety, and solving mixed determined problems, particularly unstable ones, is the cause celebre of the discipline.

Intuition suggests that hybrid problems such as these might be solved using hybrid solutions. We can propose that solutions can be found by minimizing an objective function that penalizes both the residual and the length of the model estimate:

$$m = \underset{m}{\operatorname{argmin}} (Gm-d)^T (Gm-d) + \varepsilon^2 m^T m \tag{2.15}$$

In this formula, $\varepsilon^2$ is meant to denote a small number, and the model estimator is meant to be suitable for a problem that is only slightly under determined, i.e., with $p$ only a little less than $m$. Applying the same rules for differentiating as before yields the estimate:

$$m^{\text{est}} = (G^T G + \varepsilon^2 I)^{-1} G^T d \tag{2.16}$$

Clearly, the introduction of prior information about the undesirability of long model solutions has contributed some damping to the linear least-squares solution found earlier. Even when $G^T G$ tends toward singularity, the addition of the diagonal term will help ensure the nonsinguarity of the term in the brackets, depending on the size of $\varepsilon^2$. The method in (2.16) is known as "damped least squares" and is something we might have attempted instinctively even absent the foregoing analysis. Adding damping is an intuitive approach to inverse methods and can be seen to follow from the imposition of prior preferences or information. This is the essential approach to stability.

## 2.5 Weighted measures

The aforementioned tools are adequate for some problems but are blunt instruments. The length of the model vector is only one measure of its simplicity, but other choices are also valid. Different problems invite different approaches, and the methods discussed so far can be generalized substantially.

In the event that a prior estimate of $m$ (called $m_\circ$) is somehow available, then it is the length of the vector $m - m_\circ$ that could more gainfully be minimized than the length of $m$. Beyond that, a method could be constructed to minimize the length of the vector $L(m-m_\circ)$ where $L$ is any linear transformation. The square of the L2 norm then becomes $(m-m_\circ)^T L^T L (m-m_\circ) = (m-m_\circ)^T W_m (m-m_\circ)$ where $W_m = L^T L$ is the damping matrix, which is symmetric.

The transformation $L$ could be chosen to minimize the length of some portions of the model to greater degrees than others, for example. For another example, minimizing the length of the vector that is the first derivative of $m$ expresses a preference for model vectors that are uniform if not necessarily small. Using a backward difference operator, $L$ in this case may be written

$$L = \frac{1}{d} \begin{bmatrix} 1 & & & & & -1 \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \end{bmatrix} \tag{2.17}$$

where $d$ is the (uniform) distance between samples and where the boundary condition is taken to be periodic. The transformation could equally well be constructed using a forward or center difference scheme. Another choice might be to minimize the length of the second derivative or curvature of $m$ using the center-difference formula

$$L = \frac{1}{d^2}\begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix} \tag{2.18}$$

where the boundary conditions are again taken to be periodic. This choice for $L$ will have the effect of minimizing roughness in the model. Minimizing the curvature of the model is a common approach with a rationale in problems where the model is expected to vary smoothly. In some circumstances, it could be advantageous to minimize the length of the 0th, 1st, or 2nd derivative of different parts of $m$. Even more possibilities exist. Consider making $L$ a discrete Fourier transform. Different weights could be applied to different Fourier components of $m$ in the objective function which could then be used, for instance, to suppress spurious, high-frequency components of the model.

Just as different weights can be incorporated in the damping function, they can also be incorporated in the residual term $Gm - d$. While we are presently regarding the model $m$ and the predicted data $Gm$ as being deterministic quantities, the error term $e$ is always taken to be stochastic and makes an additive contribution to the experimental data, $d$, which is therefore a random variable. The statistical error associated with different components of the data vector may be different in general, and that information needs to be incorporated in the objective function. Moreover, errors in different parts of the data vector may be statistically correlated. In the case of experimental errors which have zero mean and obey a multivariate Gaussian distribution, the statistical properties of the errors are expressed completely by the data covariance matrix $C_d = \langle ee^T \rangle$. This matrix needs to be estimated either on theoretical grounds or from actual data samples.

The square of the L2 norm of the residual incorporating the data covariance matrix is $(Gm-d)^T C_d^{-1}(Gm-d)$ which is the $\chi$-square value. Motivation for using $\chi$-square in the objective function will be given later in the text. For the present, it seems intuitive to assign more weight to errors in accordance with their covariances. Since the error covariance matrix is positive definite symmetric, its inverse is symmetric and has a square root defined by

$$C_d^{-1} = C_d^{-T/2} C_d^{-1/2} \tag{2.19}$$

where $C_d^{-T/2}$ denotes the transpose of the square-root inverse matrix. Similar remarks hold for the model weight matrix $W_m$ and its inverse.

Incorporating the model and data weights into the estimators formulated thus far, we can write the linear least-squares estimator as:

$$m^{\text{est}} = (G^T C_d^{-1} G)^{-1} G^T C_d^{-1} d \tag{2.20}$$

and the model simplicity solution as:

$$m^{\text{est}} = m_\circ + W_m^{-1} G^T (G W_m^{-1} G^T)^{-1} (d - G m_\circ) \tag{2.21}$$

where the data covariance matrix doesn't enter because the residual is taken to zero. Finally, the weighted-damped least squares estimator for mixed determined problems can be written as:

$$m^{\text{est}} = m_\circ + (G^T C_d^{-1} G + \varepsilon^2 W_m)^{-1} G^T C_d^{-1} (d - G m_\circ) \tag{2.22}$$

The above estimators are perfectly competent and useful and for the basis for solving discrete linear inverse problems. Their main limitation at this point is in knowing when they will be effective, how effective, and why. Another limitation involves all of the matrix-matrix products and matrix inverses involved in the estimators. These operations become computationally expensive quickly when large matrices are involved. The computational cost may be prohibitive for problems involving real-world datasets. Ultimately, we will see that explicit methods are simple but relatively inefficient means of solving linear inverse problems.

## 2.6 Constraints

Finding the model simplicity solution involved the imposition of a constraint. Constraints are conditions that must be met exactly rather than simply preferences that should be taken into account. Imposing constraints is often a useful way to exclude unacceptable solutions in inverse problems. Introducing constraints into linear, discrete problems is straightforward.

Consider the constrained least-squares problem:

$$m^{\text{est}} = \underset{m}{\text{argmin}} \; (Gm-d)^T(Gm-d) + 2\lambda^T(Fm-h) \tag{2.23}$$

Here, the linear constraints have been expressed in a general way in the form $Fm-h=0$ where $F$ is a matrix and $h$ is a vector. As before, the constraints have been introduced to the optimization problem with the help of a vector of Lagrange multipliers $\lambda$. We now apply optimization to a "Lagrangian" function which combines the original objective function with the constraint. In such problems, the solution is found through differentiation with respect to the model vector and also with respect to the Lagrange multiplier vector. Following the usual procedure, we can obtain the equations:

$$G^TGm + F^T\lambda = G^Td \tag{2.24}$$

$$Fm = h \tag{2.25}$$

where the second equation is just the original constraint. In augmented matrix form, the system can combined and written as:

$$\left[\begin{array}{c|c} G^TG & F^T \\ \hline F & 0 \end{array}\right] \left[\begin{array}{c} m \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} G^Td \\ \hline h \end{array}\right] \tag{2.26}$$

The original normal equation before the imposition of constraints was $G^TGm = G^Td$. What has happened with the imposition of constraints is that the normal equation has been augmented. The $G^TG$ term has new rows and columns made up of the elements of $F$ while the $m$ and $G^Td$ vectors have new rows made up of the elements of $\lambda$ and $h$, respectively. Adding constraints is therefore trivial and can be done as through bootstrapping. Note that it is generally unnecessary to solve for the Lagrange multipliers although they will "fall out" of the analysis. Lagrange multipliers represent forces of constraint and so may have some physical significance.

While it is possible to consider inequality constraints of the form $Fm-h \geq 0$ in addition to equality constraints of the form $Fm-h=0$, the former generally necessitate the use of iterative techniques and so fall outside the scope of this chapter. Inequality constraints are discussed in the appendix. A simple method of solution in the present context is as follows. If the solution to the unconstrained inverse problem satisfies the inequality constraint, than no more work need be done. The constraint is said to be inactive or slack in that case. Otherwise, the constraint is said to be active and to bind the solution, which must in the case of linear problems lie at the point on the equality constraint closest to the unconstrained solution (but on the equality constraint for emphasis). Solution then is by the method of equality constraint as before.

A simple example is the problem of constrained linear regression. The direct problem has the form (neglecting errors):

$$\begin{bmatrix} d \end{bmatrix} \approx \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \tag{2.27}$$

where $x_i$ are the fixed coordinates of the data points. Let the problem be constrained by the requirement that the line pass through the data point $(x_\circ, d_\circ)$. The solution to the constrained least squares fit of a line to the data is therefore:

$$\left[\begin{array}{c} m_1^{\text{est}} \\ m_2^{\text{est}} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{cc|c} n & \sum_i x_i & 1 \\ \sum_i x_i & \sum_i x_i^2 & x_\circ \\ \hline 1 & x_\circ & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \sum_i d_i \\ \sum_i x_i d_i \\ \hline d_\circ \end{array}\right] \tag{2.28}$$

where all of the various matrix components correspond to the components in (2.26) above. Complicated systems of constraints can thus be incorporated expediently.

## 2.7 Example: Vertical seismic profiling

A simple example is provided by the vertical seismic profiling problem discussed in Chapter 1. We consider a bore hole 40 m deep. The slowness model is discretized at $m$=80 equally-spaced depths. and travel times are considered as having been measured at $n$=160 equally-spaced depths. The results of a numerical experiment are shown in Figure 2.1. The first row of the figure shows the true slowness profile for this problem along with synthetic data. The synthetic data have added to them normally-distributed independent noise with zero mean and a standard deviation of 0.15 s. The noise is not visible to the eye, and the casual observer has no trouble in identifying two parabolic curves joined by a linear curve in the synthetic travel time data.

The second row of Figure 2.1 shows the results of two essentially equivalent inversion strategies – differentiation in the left panel and linear least squares in the right panel. Differentiation is performed using a midpoint rule on the synthetic data. Both strategies produce "noise" results illustrating the amplification of the noise in the synthetic data. That the curve produced by differentiation appears to be noisier than that produced by linear least squares inversion is a consequence of the fact that 160 model estimates are produced by the former strategy and only 80 by the latter, which therefore benefit from nearest-neighbor averaging. In the $n = m$ case, the two curves are identical.

The third row shows model estimates computed using damped least squares. In the curve on the left, the model weight $L$ is the identity. On the right, $L$ is the 2nd-derivative matrix. Both strategies have achieved a fair degree of smoothing. The curve on the right is smoother than that on the left, but this has come at the cost of additional rounding of the edges of the slowness profile. Edge-preserving strategies for inverse problems will be discussed later in the text.

The bottom row of Figure 2.1 illustrates different strategies for handling the valley in the slowness profile. Suppose there was good (a priori) reason to believe that the valley in the profile is very flat. The curve on the left shows the result of applying a weight matrix to the data de-emphasizing the observations in the valley region. De-emphasizing the data in the weighted damped least squares strategy emphasizes the smoothness constraint, which is again imposed here by a 2nd-derivative matrix for $L$. Consequently, the slowness profile is made to be even smoother in the valley. The curve on the right shows the results of a more direct approach. Here, the constraint that the slowness is equal to 2 s/km has been imposed through a constraint which has been used to augment the linear least squares problem. The unconstrained portion of the model estimate is identical to what was computed using unconstrained linear least squares.

Different results are achieved for different model and data weight matrices and different damping parameters $\varepsilon^2$. A discussion regarding optimum choices for these parameters will be presented later in chapter 4. The efficacy of the various strategy in inverting the synthetic seismic profiling data is clear in this case, however, as well as in cases when $n < m$.

## 2.8 Method of normal equations

While there is nothing especially difficult about the computation of the model estimators above, a substantial shortcut in their calculations can be achieved by considering the form of the objective functions being minimized. Reconsider first the linear least squares problem:

$$m^{\text{est}} = \underset{m}{\text{argmin}} \, ||Gm - d||_2^2 \tag{2.29}$$

$$\tag{2.30}$$

the goal being to find $m$ such that the square of the L2 norm of the residual is a minimum. If the data were in the column space of the matrix $G$, then the residual could be made to be zero with the appropriate weight vector $m$. That it cannot be made to be zero is evidence that part of $d$ lies outside the column space of $G$. The most optimal choice for

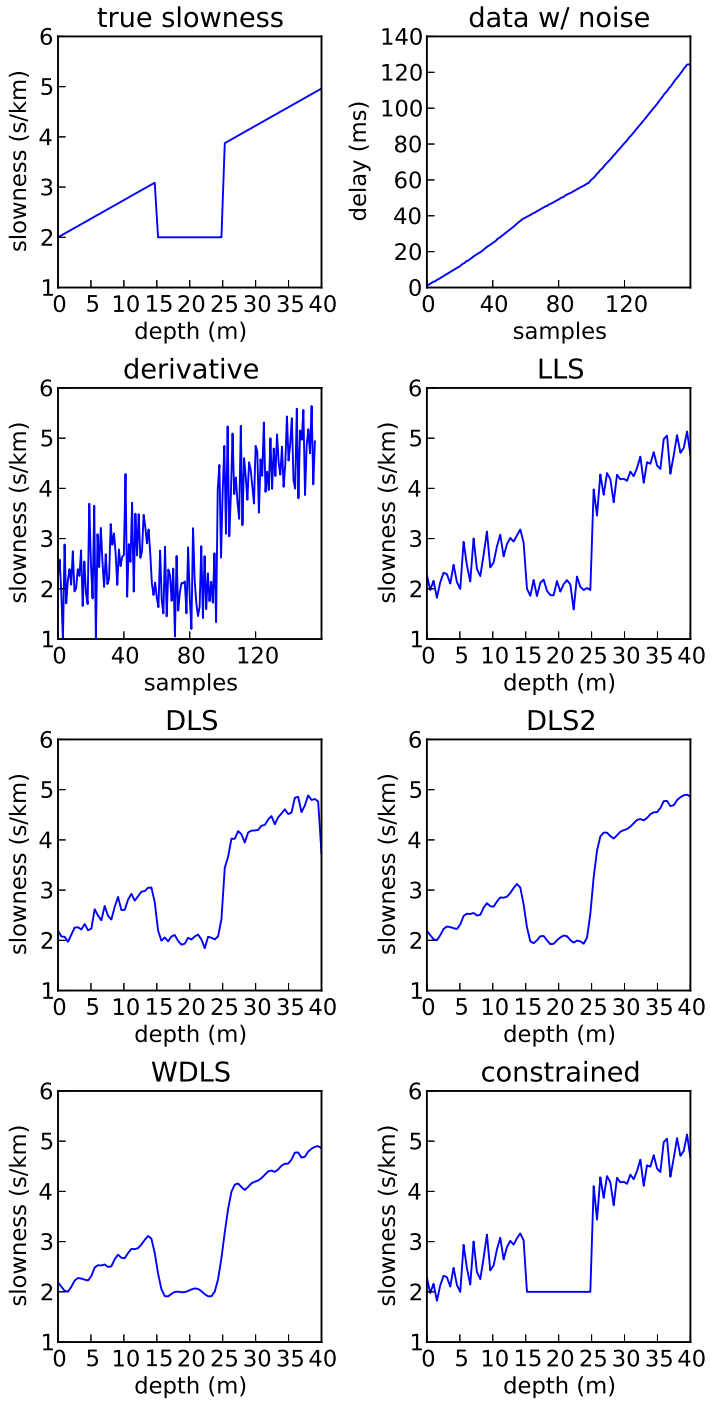


Figure 2.1: Vertical seismic profiling example for a 40-m bore hole.

$m$ will be the one for which the error or residual $e$, the discrepancy between $Gm$ and $d$, is normal to the column space of $G$. This condition can be written in terms of the normal equation:

$$G^T(Gm^{\text{est}} - d) \quad = \quad 0 \tag{2.31}$$

which can be solved for $m$,

$$m^{\text{est}} \quad = \quad \left(G^T G\right)^{-1} G^T d \tag{2.32}$$

reproducing the linear least squares estimator.

This is a powerful approach to solving all the discrete linear inverse problems considered so far, including model and data weights and prior model estimates. For example, the weighted least squares problem statement can be written as:

$$m^{\text{est}} \quad = \quad \underset{m}{\text{argmin}} \left\| C_d^{-1/2}(Gm - d) \right\|_2^2 \tag{2.33}$$

where the optimum solution for $m$ is the one for which the weighted residual is normal to the operator $C_d^{-1/2}G$, i.e.,

$$G^T C_d^{-T/2} C_d^{-1/2}(Gm - d) \quad = \quad 0 \tag{2.34}$$

which has the now-familiar solution

$$m^{\text{est}} \quad = \quad \left(G^T C_d^{-1} G\right)^{-1} G^T C_d^{-1} d \tag{2.35}$$

Evidently, the data weights act like a filter which can be applied to the entire direct problem. Likewise, the weighted damped least squares problem can be stated in terms of the minimization of the L2 norm of a residual vector:

$$m^{\text{est}} \quad = \quad \underset{m}{\text{argmin}} \left\| \begin{pmatrix} C_d^{-1/2} G \\ \varepsilon L \end{pmatrix} (m - m_\circ) - \begin{pmatrix} C_d^{-1/2}(d - Gm_\circ) \\ 0 \end{pmatrix} \right\|_2^2 \tag{2.36}$$

where $L^T L = W_m$ is effectively a filter being applied to the inverse problem. Application of the method of normal equations gives

$$\left(G^T C_d^{-T/2} \quad \varepsilon L^T\right) \left[ \begin{pmatrix} C_d^{-1/2} G \\ \varepsilon L \end{pmatrix} (m - m_\circ) - \begin{pmatrix} C_d^{-1/2}(d - Gm_\circ) \\ 0 \end{pmatrix} \right] \quad = \quad 0 \tag{2.37}$$

with the solution for the model estimate being

$$m^{\text{est}} \quad = \quad m_\circ + (G^T C_d^{-1} G + \varepsilon^2 W_m)^{-1} G^T C_d^{-1}(d - Gm_\circ) \tag{2.38}$$

$$\quad = \quad m_\circ + W_m^{-1} G^T (G W_m^{-1} G^T + \varepsilon^2 C_d)^{-1}(d - Gm_\circ) \tag{2.39}$$

where the latter variant can be shown to be identical to the former through an identity [1]. Many other variations are possible. In order to use the method of normal equations, keep in mind that the weight matrices involved in the residual vectors be positive definite symmetric.

## 2.9 Example: LCMV, radio imaging

The model simplicity problem is an example of a class of problems called linear constrained minimum variance (LCMV) or sometimes minimum variance distortionless response (MVDR). The idea is to solve an under determined

[1] Start with $A^T + A^T B^{-1} A D A^T = A^T B^{-1} A D A^T + A^T$. Write the first term on the left as $A^T B^{-1} B$ and the last term on the right as $D^{-1} D A^T$. Factor out the common term $A^T B^{-1}$ on the left side and $DA^T$ on the right. Finally, multiply both sides by the inverse of the additive term from the other side to arrive at the required identity.

or mixed determined inverse problem by minimizing an objective function subject to constraints. The method is applied widely in science and engineering and deserves further exploration.

Consider the example from radio astronomy where a number of antennas at different locations capture signals from different radio sources in the sky. By summing the various signals with the appropriate weights, it is possible to form a single, synthetic antenna with very high gain. The problem becomes one of selecting the weights which maximize the antenna gain in the direction of a radio source of interest, which is known, while minimizing contamination from one or more interfering sources with unknown locations. This is tantamount to making a phased array antenna with a main lobe in the direction of the source and nulls in the directions of interferers.

Denote $x(t) \in \mathbb{R}^n$ the column vector containing the complex, zero-mean signal voltages coming from $n$ receivers. We regard $x(t)$ as a Gaussian random variable in view of fading of the desired radio signals as well as additive background (cosmic) noise. Take $w \in \mathbb{R}^n$ to be the weight column vector. The total output of the network at any time will then be the sum $y(t) = w^H x$ where $H$ denotes the Hermitian transpose.

The expectation of the output power will be

$$\begin{aligned} \langle |y|^2 \rangle &= \langle w^H x x^H w \rangle && (2.40) \\ &= w^H \langle x x^H \rangle w && (2.41) \\ &= w^H R w && (2.42) \end{aligned}$$

where $R \in \mathbb{R}^{n \times n}$ now is the covariance of the signals from the $n$ different antennas. $R$ is a Hermitian matrix which completely specifies the statistical properties of $x$. Specialized hardware exists to compute $R$ which is the dataset here.

To minimize signal contamination due to interferers, we can find the weights that minimize the expectation of the output power. By itself, this strategy is inadequate since the zero vector solves this problem and since the desire to receive the desired signal has not been accommodated. Doing so requires the addition of a constraint.

Let $d$ demote a vector which contains the spatial positions of the receiving antennas relative to some reference point. Plane waves propagating between a given radio source and the Earth with a given frequency can be described by a wavevector $\mathbf{k}$ with a direction pointing from the source to the Earth. Let $e$ be a column vector representing the outputs of the receiver network corresponding to a signal from one radio source only:

$$e^H = \left[ e^{i\mathbf{k}\cdot\mathbf{d}_1} \; e^{i\mathbf{k}\cdot\mathbf{d}_2} \; \cdots \; e^{i\mathbf{k}\cdot\mathbf{d}_n} \right] \quad (2.43)$$

To ensure reception of such a signal, we can constrain the weights such that signals arriving along the vector $\mathbf{k}$ will produce a fixed output – unity for example. Consequently, the Lagrangian function for the problem may be written

$$L(w, \lambda) = w^H R w + \lambda (w^H e - 1) \quad (2.44)$$

where $\lambda$ is a Lagrange multiplier, as before. The weights are now found by minimization with respect to $w$ and $\lambda$, yielding (using the rules of complex matrix differentiation this time, namely $\partial(w^H R w)/\partial w = w^H R$, $\partial(e^H w)/\partial w = e^H$ in this case):

$$w = R^{-1} e (e^H R^{-1} e)^{-1} \quad (2.45)$$

(after first expressing $w$ in terms of $\lambda$ and then applying the constraint condition). In fact, it is not really the weights that are desired for this problem but rather the output power in the direction of interest. This is found through substitution of the weights found above into the expression for the synthetic antenna output power (2.42).

$$\langle |y|^2 \rangle = \frac{1}{e^H R^{-1} e} \quad (2.46)$$

where $R$ are precomputed data and $e$ is a vector computed for each desired receiving direction (bearing). This is a method of adaptive beamforming.

The result of this inverse problem has several remarkable features. For one, the weights never actually need be calculated, and so the cost of evaluating (2.45) is never realized. For another, the solution suppresses contamination due to interference sources even though their locations are unspecified. In fact, it would be possible a posteriori to find

the interferers by examining the nulls in the synthetic antenna pattern implied by the weights. The algorithm could therefore be used for interference tracking. Finally, the location of the signal of interest need not be known a priori. Instead, every bearing in the sky could be examined one-by-one. The algorithm could be used for discovery and is, in fact, an imaging algorithm suitable for solving the aperture synthesis problem posed in chapter 1.

Note that this method is essentially different from others discussed in this section in that the desired quantity being estimated is not the state vector $w$ itself but rather the power, $\langle |y|^2 \rangle$, which is not itself linearly related to the data contained in $R$. The method of solution is similar, but methods for evaluating the quality of the solution are not.

## 2.10 Model and data resolution matrices

This chapter has mainly dealt with problems of the form

$$G(m - m_\circ) \quad \approx \quad d - Gm_\circ \tag{2.47}$$

where $m$ is the hidden state vector or model, $m_\circ$ is an a priori estimate of $m$ (if available), $d$ are observable data, and $G$ is a linear transformation mapping the model to the data. The imperfect equivalence in (2.47) arises from experimental errors which render straightforward methods of solution ineffective. Length methods were therefore used to develop model estimates, e.g. (2.20), (2.21), and (2.22), which have the form

$$m^{\text{est}} \quad = \quad m_\circ + \tilde{G}(d - Gm_\circ) \tag{2.48}$$

where $\tilde{G}$ is another linear transformation mapping the data to the model estimate. The transformation $\tilde{G}$ behaves something like the inverse of $G$ and is referred to as the pseudoinverse. It is not an actual inverse for sure – $G$ and $\tilde{G}$ need not be square matrices and $G\tilde{G} \neq I$, $\tilde{G}G \neq I$ in general. Equipped with a model estimate, it is then possible to "predict" or estimate data which obey the direct equation identically:

$$G(m^{\text{est}} - m_\circ) \quad = \quad d^{\text{est}} - Gm_\circ \tag{2.49}$$

The resemblance between the estimated and actual data could be one measure of the efficacy of the inverse method. Likewise the resemblance between the estimated and actual model, provided the latter could somehow be known.

Permuting the formulas for the direct and inverse problem above gives two more relationships:

$$d^{\text{est}} - Gm_\circ \quad = \quad G(m^{\text{est}} - m_\circ) \quad = \quad \underbrace{G\tilde{G}}_{R_d}(d - Gm_\circ) \tag{2.50}$$

$$m^{\text{est}} - m_\circ \quad = \quad \tilde{G}(d - Gm_\circ) \quad \approx \quad \underbrace{\tilde{G}G}_{R_m}(m - m_\circ) \tag{2.51}$$

where $R_d \in \mathbb{R}^{n \times n}$ and $R_m \in \mathbb{R}^{m \times m}$ are symmetric matrices known, respectively, as the data and model resolution matrices. As the problems being considered are linear, the model estimate is a linear transformation of the observed data, and the predicted data are a linear transformation of the model estimate and therefore also a linear transformation of the observed data. The model resolution matrix $R_m$ indicates the how the elements of the model estimate depend on the elements of the true model. The data resolution matrix $R_d$ indicates how the elements of the data predicted by the model estimate depend on the elements of the observed data.

Ideally, we would like both $R_d$ and $R_m$ to approach the identity matrix, which would indicate that the model estimate and the data it predicts are strongly and locally determined by the true model and the observed data, respectively. Departures from the identity are indicative of blurring, conflation, and lack of resolution. The two matrices can be interpreted as metrics of the success of the inverse method in that sense.

It is useful to define some measures of the resolution matrices. The Dirichlet spread of $R$ is the square of the entrywise L2 norm of the departure of the matrix from the identity, i.e.

$$\text{spread}(R) \quad \equiv \quad \|R - I\|_F^2 \tag{2.52}$$

The smaller the spread, the close the matrix to the identity. Here, the matrix norm in questions is the Euclidean length of a vector whose elements are all the elements of the matrix. This is usually called the Frobenius norm which may also be evaluated as $||A||_F = \sqrt{\mathrm{Tr}(A^T A)}$.

We can evaluate the weighted least squares inverse in terms of the spread of the two resolution matrices. Consider first the model resolution matrix:

$$\begin{aligned} \tilde{G}G &= (G^T C_d^{-1} G)^{-1} G^T C_d^{-1} G & (2.53) \\ &= I & (2.54) \end{aligned}$$

Evidently, the model resolution matrix has zero spread, and so the elements of the model estimate are completely controlled by the corresponding elements of the true model. This reflect the fact that the model estimator is unique. The data resolution matrix meanwhile is not the identity in this case. We can attempt to formulate an inverse method based on the proposition of minimizing the spread of the data resolution. Following this prescription yields:

$$\tilde{G} = \underset{\bar{G}}{\mathrm{argmin}}\, \mathrm{spread}(R_d) \quad = \quad (G^T G)^{-1} G^T \tag{2.55}$$

which is the least-squares inverse. Note that this result is found readily by first expanding the matrix products in the spread function in terms of explicit summations. Alternatively, one may calculate $(R-I)^T(R-I)$ and differentiate its trace making use of the identities:

$$\begin{aligned} \frac{d}{dX}\mathrm{Tr}(AX) &= A^T & (2.56) \\ \frac{d}{dX}\mathrm{Tr}(X^T A) &= A & (2.57) \\ \frac{d}{dX}\mathrm{Tr}(X^T A X) &= (A + A^T)X & (2.58) \end{aligned}$$

where $X$ and $A$ are matrices. The proof of these identities is then through expansion in summations.

Likewise, it is apparent that the data resolution matrix for the model simplicity estimator has zero spread:

$$\begin{aligned} G\tilde{G} &= G W_M^{-1} G^T (G W_M^{-1} G^T)^{-1} & (2.59) \\ &= I & (2.60) \end{aligned}$$

which is a consequence of fact that the solution is exact. Minimizing the spread of the model resolution matrix now gives

$$\tilde{G} = \underset{\bar{G}}{\mathrm{argmin}}\, \mathrm{spread}(R_m) \quad = \quad G^T (G G^T)^{-1} \tag{2.61}$$

which is the model simplicity inverse. Note that model and data weights can be included in the pseudoinverses developed here by first incorporating them in the original problem statement in (2.47). The prescription is as follows. To the data resolution matrix, apply a filter factor of $C_d^{-1/2}$. To the model resolution matrix, apply a filter factor of $\varepsilon L$. The resulting pseudoinverses found by minimizing the appropriate spread function will agree with what we found earlier for the weighted least squares and model simplicity inverses as the case may be.

So, the least squares inverse has model resolution with zero spread and a data resolution with minimal spread while the model simplicity inverse has data resolution with zero spread and model resolution with minimal spread. What about damped least squares – can this inverse be recovered from the point of view of minimizing the spread of resolution matrices? It turns out that it can, in part, but in the context of a statistical framework. The next chapter develops such a framework and offers an alternative derivation of the damped least squares inverse accordingly.

It should be noted that the spread of a resolution matrix is not the last word in quantifying the performance of the method. Note that the spread penalizes off-diagonal components of the resolution matrix equally no matter how far off the diagonal the component is. The spread is not sensitive to locality in this regard, and other metrics might be more incisive depending on the the importance of locality. Other ways of quantifying the resolution matrices will be considered later in the text.

## 2.11 References

## 2.12 Problems

# Chapter 3

# Statistical perspective

This chapter has dealt with problems of the form

$$G(m - m_\circ) + e \quad = \quad d - Gm_\circ \tag{3.1}$$

where $m$ is viewed as the deterministic, hidden state of the system, $m_\circ$ is some a priori estimate of the state if available (and zero otherwise), $G$ is a deterministic system, $e$ (called observation noise in some contexts) represents experimental error, and where $d$, the observable data, takes its properties from the model and the errors. That $e$ is stochastic makes $d$ stochastic as well.. If the expectation of the error vector is zero, then the expectation of the data is the model prediction, and the data are unbiased. This is the usual circumstance. In practice, any data biases generally need to be removed before analysis can proceed.

The model estimates considered so far have been of the form

$$m^{\text{est}} \quad = \quad m_\circ + \tilde{G}(d - Gm_\circ) \tag{3.2}$$

Since the model estimate is derived from the data, it too is a stochastic quantity. The presence of the errors in the data will tend to cause the model estimate to deviate from the true model. In the event that the expectation of the model estimate is unequal to the true model, the estimator is biased. If the expectation of the model estimate is equal to the true model, it is an unbiased estimator, but errors in the data will nevertheless cause deviations and contribute to model variance.

In view of the fact that the model resolution matrix has zero spread in the case of over determined problems, the least squares estimator would appear to be an unbiased estimator. This conjecture will be proven later in the chapter. The estimators for under determined and mixed determined problems, meanwhile, are presumably biased estimators which are biased in the direction preferred by the model weight matrix.

We are interested in the properties of $e$, $d$, and $m^{\text{est}}$ and require a specification of the associated statistical populations. In a great many practical instances in the natural sciences, the error term $e$ can be regarded as having a multivariate normal distribution. This is a consequence of the central limit theorem which applies to random processes which involve a large number of independent random subprocesses, each with a well-defined expectation and variance. In such circumstances, the aggregate random variable will be normally distributed regardless of the distributions of the subprocesses.

In experimental science, instrumental error arises from a great many independent sources including instrumental and environment noise, sampling error, and round-off error. Experimental data from natural studies are often good candidates for applying the central limit theorem. The remainder of this chapter and this text will therefore concentrate on normally distributed errors. Note, however, that the independence of the myriad subprocesses at work in producing $e$ does not imply that the elements of $e$ themselves will be statistically independent. Errors in different parts of the state vector could be highly correlated depending on the context.

Define the expectation of $e$ as $\mu_e = \langle e \rangle$ and the covariance of $e$ as $C_d = \langle (e - \mu_e)(e - \mu_e)^T \rangle$. Then for $C_d \in \mathbb{R}^{n \times n}$,

the probability density function for normally distributed $e$ (denoted $\mathcal{N}(\mu_e, C_d)$) is given by:

$$P(e) \quad = \quad (2\pi)^{-n/2}|C_d|^{-1/2}e^{-\frac{1}{2}(e-\mu_e)^T C_d^{-1}(e-\mu_e)} \tag{3.3}$$

where $|C_d|$ is the determinant of $C_d$. For unbiased data, $e$ will be $\mathcal{N}(0, C_d)$.

The subscript $d$ has been used for $C_d$ because the data will inherit their covariance from the experimental error and share the same covariance matrix, i.e. $C_d = \langle (d-\mu_d)(d-\mu_d)^T \rangle$ where $\mu_d = \langle d \rangle = Gm$ is nonzero this time. For $d$ $\mathcal{N}(Gm, C_d)$,

$$P(d|m) \quad = \quad (2\pi)^{-n/2}|C_d|^{-1/2}e^{-\frac{1}{2}(d-Gm)^T C_d^{-1}(d-Gm)} \tag{3.4}$$

The error covariance matrix can be estimated theoretically in some circumstances and empirically on the basis of the sample covariance in others. It may or may not be diagonally dominant, as the case may be.

## 3.1 Statistical errors and error propagation

Experimental errors in the data will telegraph through the pseudoinverse operator in (3.2) and affect the model estimate according to

$$\delta m \quad = \quad \tilde{G}\,\delta d \tag{3.5}$$

The relationship is simple due to the linearity of the problem. Consequently, we may write

$$\langle \delta m\, \delta m^T \rangle \quad = \quad \langle \tilde{G}\delta d \delta d^T \tilde{G}^T \rangle \tag{3.6}$$

$$= \quad \tilde{G}\langle \delta d\, \delta d^T \rangle \tilde{G}^T \tag{3.7}$$

Now, the terms inside the angle brackets can be identified as the model and data error covariances, respectively, i.e.

$$C_m \quad = \quad \tilde{G} C_d \tilde{G}^T \tag{3.8}$$

giving the transformation that maps the data error covariance into the model estimate error covariance, $C_m \in \mathbb{R}^{m \times m}$. For linear problems, we also have the implicit assumption that the model errors will be normally distributed if the data errors are. Proven below, this fact and the transformation in (3.8) lie at the heart of error propagation in linear inverse problems. The model error covariance matrix joins the model and data resolution matrices as indicators of the overall performance of a method. Note that the model error covariance matrix need not be diagonally dominant even if the data error covariance matrix is. This fact is too often neglected in practice.

The terms along the main diagonal of $C_m$ are the model error variances, and their square roots are the standard deviations associated with the corresponding model estimate elements. The standard deviations are one measure of confidence in the estimates. When plotting model estimates, it is customary to accompany the plotted points with error bars representing intervals plus-and-minus one or two standard deviations wide. The $\pm\, 1\sigma$ error bars represent 68% confidence intervals whereas the $\pm\, 2\sigma$ error bars represent 95% confidence intervals. This means that in the event that the analysis were repeated under identical conditions but with different realizations of the random variable $e$, the interval would contain the expected value of the model the given fraction of the time (neglecting error biases or other systemic problems in the formulation). Be careful about interpreting confidence intervals any other way.

The model error covariance matrix contains off-diagonal terms as well that may not be negligible. Large off-diagonal terms imply highly correlated errors. While smoothness in model estimates might tend to instill confidence in the analysis, smoothness may also be indicative of errors which, although large, are highly correlated. Smoothness can be misleading and rendering model error covariances is important but can be challenging. Since $C_m$ and its inverse are positive definite symmetric, it can always be diagonalized through similarity transformation, and the model estimates can also be accordingly transformed into and plotted in the space where $C_m$ is diagonal. This is one way to resolve the rendering problem – to plot the model estimate and the corresponding confidence intervals in a space where the model error covariance matrix is diagonal.

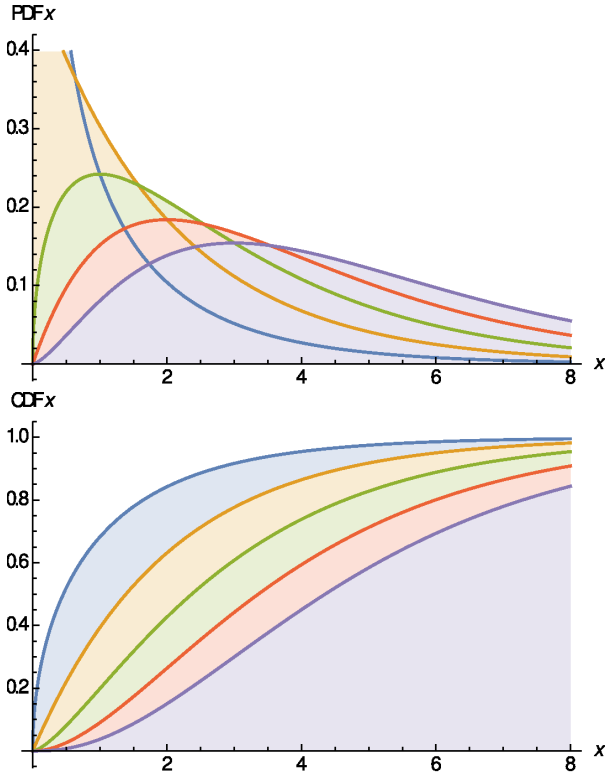


Figure 3.1: (top) PDF for chi-squared distribution. (bottom) CDF for chi-squared distribution. Curves for $k = 1 - 5$ are shown.

## 3.2 Chi-squared and statistical tests

Another metric for evaluating the performance of an inverse method is the chi-squared statistic:

$$\chi^2(k) \quad = \quad (d - Gm)^T C_d^{-1} (d - Gm) \tag{3.9}$$

which involves the model estimates rather than the true model and where $k$ is the number of degrees of freedom. Since the chi-squared statistic derives from the random variable $d$, it too is a random variable that obeys a chi-squared distribution:

$$P(x) \quad = \quad \frac{1}{2^{k/2}\Gamma(\frac{k}{2})} x^{k/2-1} e^{-x/2} \tag{3.10}$$

The expectation of this PDF is $k$, and the variance is $2k$. The corresponding cumulative density function above is

$$C(x) \quad = \quad \int^x P(x)\, dx \tag{3.11}$$

$$\quad = \quad \frac{1}{\Gamma(\frac{k}{2})} \gamma\left(\frac{k}{2}, \frac{x}{2}\right) \tag{3.12}$$

In the above, $\gamma$ is the lower incomplete Gamma function and $\Gamma$ is the regularized Gamma function. See Figure 3.1 for illustrations.

In the context of linear discrete inverse problems, the number of degrees of freedom $k$ is the number of equations or data $n$ minus the number of unknowns or model parameters $m$. For even determined or under determined problems, the chi-squared statistic can be made to be zero. It will be nonzero only for over determined or mixed determined problems with $n > p$.

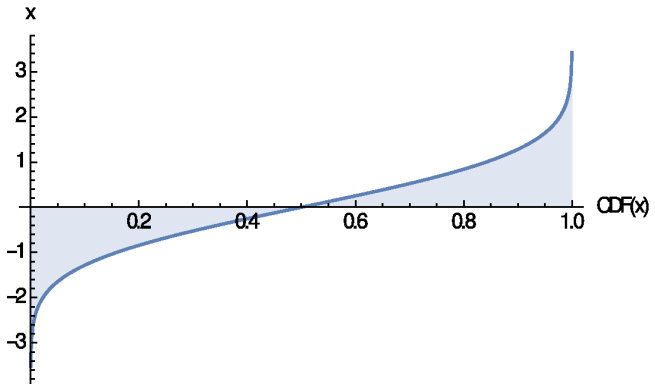


Figure 3.2: Inverse cumulative density function (CDF) corresponding to a Gaussian PDF with zero mean and unity variance.

A number of statistical tests have been developed to help evaluate the efficacy of an inverse method solution based on the chi-squared statistic or something similar. What is being tested is the hypothesis that the inverse model candidate is true. The test is to evaluate the probability of obtaining a statistic less than the one actually implied by the data. If the probability falls outside a predetermined range of thresholds or significance levels, then either the hypothesis is sound but an extreme result occurred or that the hypothesis must be rejected. Multiple applications of the test using independent data can be used to address the occurrences of extreme results.

A test involving chi-squared is Pearson's P-value test. This test involves calculating chi-squared for a given inverse solution with some number of degrees of freedom, substituting the result into (3.11), and observing whether the associated probability lies within some range — 0.05–95 typically. This range is consistent with the rule of two standard deviations but is essentially arbitrary.

It is important to note that the P-value test does not give the probability of the hypothesis being true or some other hypothesis being false. It is furthermore not the probability of a false positive, false negative, true positive, or true negative. The P-value test is rooted in frequency statistics which simply do not address the probabilities of hypotheses. The P-value test can merely indicate that either the hypothesis should be rejected or that an extreme result has been obtained.

The P-value test will give misleading results when the number of data involved in the model estimate is not great. The results of the test depend on the merits of the model together with the size of the dataset in an inseparable way. This is a common criticism of the test. More persuasive methods of hypothesis testing are available but require familiarity with Bayesian statistics which address the probability of hypotheses directly.

## 3.3 Monte-Carlo error propagation

Often times, it is useful to test the efficacy of an inverse method using synthetic data, complete with added random experimental errors. The errors need to be generated so as to exhibit realistic statistical properties. Families of model estimates can be generated from different realizations of experimental errors, and the statistics of the model estimates can thus be evaluated. The model error covariance matrix can be generated this way beginning with a specification of the original error covariance matrix.

The procedure requires the availability of a numerical random number generator. Since number generators capable of producing independent random variables which are uniformly distributed on the interval $[0, 1]$ are commonplace, we can begin with that. In order to generate independent, normally-distributed random numbers, the inverse of the Gaussian cumulative density function is required. While this function has no known analytic form, a number of tools for estimating it numerically are available. An example calculation is shown in Figure 3.2. Passing a stream of random numbers with a uniform PDF through this function will produce random numbers with a Gaussian PDF.

Suppose next that the data error covariance matrix $C_d$ is specified. Using Cholesky decomposition, this symmetric positive definite matrix can be factored into its matrix square root:

$$C_d \quad = \quad LL^T \quad = \quad LIL^T \tag{3.13}$$

where $L$ and $U$ are lower- and upper-triangular matrices, respectively. Finally, applying the $L$ transformation to a vector of independent, normally-distributed random variables will produce a vector of random variables with the covariance matrix $C_d$. These are the error vectors $e$ to be used in the construction of synthetic data for Monte-Carlo tests. That this should work is apparent after comparing (3.13) with (3.8) and noting that the covariance matrix for the original independent RVs was the identity matrix.

Monte-Carlo tests are normally not required for linear inverse problems or for most nonlinear problems for that matter. They may be required for some iterative methods where ordinary error propagation is impossible or impractical. They can furthermore be used for problems in which the errors are not normally distributed, a topic this far neglected in this text. Finally, they are useful as confidence building tools.

## 3.4 Minimum model error norm

The size of the model error covariance matrix is among the metrics that can be used to evaluate its merits of an inverse method. Minimizing the size is in fact another strategy that can be used to formulate the inverse in the first place. There are multiple ways in which to capture the size of the matrix including its L2 norm. If the off-diagonal terms in $C_m$ are relatively small, then the size of the matrix is captured fairly well by its trace.

A general strategy for designing inverse methods then is to minimize some combination of the data and model resolution spreads and the trace of the model error covariance matrix, i.e.

$$\tilde{G} = \underset{\tilde{G}}{\text{argmin}}\ \alpha\,\text{spread}(R_d) + \beta\,\text{spread}(R_m) + \gamma\,\text{Tr}(C_m) \tag{3.14}$$

For example, consider a method for mixed determined problems that are only weakly under determined. In this case, take $\alpha$=1, $\beta$=0, and $\gamma$=$\varepsilon^2$. To simplify things, take $C_d = I$. Applying the usual rules for matrix differentiation to this optimization problem yields the result:

$$\tilde{G} \quad = \quad (G^T G + \varepsilon^2 I)^{-1} G^T \tag{3.15}$$

which implies the damped least squares result given already in (2.16). As before, weights can be incorporated in this result by making use of fully general versions of $R_d$, $R_m$, and $C_m$. In any case, what was achieved before by minimizing the Euclidean length of the model vector in weakly under determined problems has been achieved here by minimizing a certain model error norm.

## 3.5 Bayes' Theorem and maximum likelihood

One of the earliest definitions of probability was the principle of insufficient reason (or the principle of indifference) posed by Bernoulli which associates the probability of a proposition with the state of knowledge about it. To Bernoulli, the probability of a proposition $p$, all else being equal, is the ratio of the total number of cases consistent with the the proposition $m$ to the total number of equally possible cases $n$. A situation where all the possible cases are equally likely seems artificial outside certain games of chance, however, reflecting the historical preoccupation of early statisticians with gambling perhaps!

For the more general problem where the cases are not equally possible or even entirely known, $p$ might still be estimated, according to Bernoulli, by observing the frequencies in a large number of trials. Frequencies are the foundation for one branch of contemporary statistics whereas information is at the root of another. Considering frequencies, Bernoulli developed the binomial distribution

$$P(m|n,p) \quad = \quad \left( \begin{array}{c} n \\ m \end{array} \right) p^m (1-p)^{n-m} \tag{3.16}$$

which predicts the number of successes $m$ in $n$ trials given that each independent trial has a success rate $p$. Bernoulli showed that in the limit that $n$ goes to infinity, the observed frequency $f = m/n$ will be close to $p$. Just how close can only be seen in the continuous limit, where the binomial distribution becomes the normal distribution

$$P(f|n,p) \quad \propto \quad \exp\left(-\frac{n(f-p)^2}{2p(1-p)}\right) \tag{3.17}$$

where the confidence intervals discussed earlier in the chapter apply. Here, the "population numbers" are known, and probabilities are with respect to sample numbers.

The original problem entertained by Bernoulli, however, was about finding the population numbers given that the sample numbers were known. This can be seen as the inverse problem which was addressed first by Thomas Bayes who found the Beta distribution, an inverse to the binomial distribution:

$$P(p|m,n) \quad = \quad \frac{(n+1)!}{m!(n-m)!}p^m(1-p)^{n-m} \tag{3.18}$$

In the continuous limit and for large $n$, it can be shown that this distribution becomes (3.17) only with the variables $f$ and $p$ interchanged. There is therefore symmetry between $f$ given $p$ and $p$ given $f$. Laplace recognized this result and generalized it, stating:

$$P(c_i|E) \quad = \quad \frac{P(E|c_i)}{\sum_{j=1}^{N} P(E|c_j)} \tag{3.19}$$

where $E$ is an observable event with $\{c_i \cdots c_N\}$ equally likely causes and $P(E|c_j)$ is the probability of $E$ for each cause. Laplace went on to generalize this further, allowing for causes with unequal probabilities $P(c_i|I)$, where $I$ denotes "prior" information.

$$P(c_i|E,I) \quad = \quad \frac{P(E|c_i)P(c_i|I)}{\sum_{j=1}^{N} P(E|c_j)P(c_j|I)} \tag{3.20}$$

In this formula, which is usually called "Bayes' Theorem," the $P(c_i|I)$ are known as prior probabilities, $P(E|c_i)$ is the transitional probability, and $P(c_i|E,I)$ is the posterior probability. It gives a recipe for converting probabilities of observing events due to different causes into probabilities of different causes being responsible for observed events. In the event that no prior information is available, the prior probabilities can be taken to be unity (uninformative priors). In fact, the meaning of prior probability is contentious. It will be treated more extensively later in the text.

Note that (3.20) is bootstrapping and that prior probabilities can be incorporated incrementally.

Bayes' theorem has far-reaching implications, but we can begin by exploiting it to better understand discrete inverse problems. The interpretation of (3.4) is that, given that the data are drawn from a jointly-normal distribution with mean $Gm$ and covariance $C_d$, $P(d|m)\,\delta d$ is the probability of observing data $d$ in the interval $\delta d$. Now, since the model estimate will be derived from $d$, it too is a random variable. With prior information about neither the model nor the data being assumed, Bayes' theorem provides the probability distribution function for the model in view of the fact that certain data have already been measured:

$$P(m^{\text{est}}|d) \quad = \quad (2\pi)^{-n/2}|C_d|^{-1/2}e^{-\frac{1}{2}(d-Gm^{\text{est}})^T C_d^{-1}(d-Gm^{\text{est}})} \tag{3.21}$$

Optimizing (3.21) with respect to the model estimate is tantamount to minimizing the term in the exponential, the chi-squared parameter. (Recall the form of (2.33)). This problem is identical to the weighted least squares problem, and the weighted least squares solution can therefore be reinterpreted probabilistically. Namely, the solution maximizes the probability of a model drawn from a population of models with a chi-squared statistic based on the difference between the data that the model predicts and the data that were actually measured.

The least squares solution fails for under determined or mixed determined problems. In the context of (3.21), these conditions imply that $P(m^{\text{est}}|d)$ possesses no unique maximum but instead exhibits a ridge.

Another way to approach the problem is to simply write a prior probability density function for the model estimate directly without any allowance for a transitional probability density:

$$P(m^{\text{est}}) \quad = \quad (2\pi)^{-m/2}|C_m|^{-1/2}e^{-\frac{1}{2}(m^{\text{est}}-m_\circ)^T C_m^{-1}(m^{\text{est}}-m_\circ)} \tag{3.22}$$

where the dimension of the distribution is now $m$. Optimizing the probability is now tantamount to minimizing the exponential in (3.22) which, when combined with the constraint that the residual be zero, is the model simplicity solution with the model weight matrix $W$ being given by the inverse model error covariance matrix $C_m^{-1}$. Once more, the solution will fail for over determined or mixed determined problems in which case (3.22) will once again exhibit a ridge rather than a unique maximum.

Finally, a posterior probability density for the model estimate combining both a transitional and a prior probability density would have the form of the product of (3.21) and (3.22). Such a product could have a unique maximum even in the case where both (3.21) and (3.22) exhibit ridges. The product of two multivariate normal distributions furthermore is another multivariate normal distribution. With the following definitions:

$$\tilde{C}_m \equiv (G^T C_d^{-1} G + C_m^{-1})^{-1} \tag{3.23}$$

$$\hat{m} \equiv m_\circ + (G^T C_d^{-1} G + C_m^{-1})^{-1} G^T C_d^{-1} (d - G m_\circ) \tag{3.24}$$

the probability density function for the model estimate can be expressed (after considerable effort – see problem set for details) as:

$$P(m^{\text{est}}|d) = (2\pi)^{-m/2} |\tilde{C}_m|^{-1/2} e^{-\frac{1}{2}(m^{\text{est}} - \bar{m})^T \bar{C}_m^{-1} (m^{\text{est}} - \bar{m})} \tag{3.25}$$

Minimizing the exponential term in this distribution is equivalent to the weighted damped least squares problem as expressed in (2.36), and the most likely model estimate (the maximum a posteriori or MAP estimate) is the weighted damped least squares solution. Consequently, the method of maximum likelihood can be used to reproduce the estimators found through all of the other methods considered so far, only this time with a statistical foundation. Error propagation can proceed from $\tilde{C}_m$.

## 3.6 Example: cosmic rays

Consider the problem of cosmic rays detected through collisions in vacuum tubes which produce some number of counts over a fixed time interval. The collisions are random events, and we regard each observation made this way as being drawn from a Poisson distribution (see Figure 3.3). For a series of random events, the probability of $k$ events occurring in a time interval is:

$$p(k|\lambda) = \frac{e^{-\lambda}(\lambda)^k}{k!} \tag{3.26}$$

where $\lambda$ is the average number of events per interval. (The formula applies to a broad class of phenomena including phone calls, traffic and parking patterns, and incidences among the Prussian cavalry of being fatally kicked by horses.)

Suppose that in some time interval, $k$ cosmic-ray events are detected. What can be inferred about the average count rate $\lambda$? With the aid of Bayes' theorem, (3.26) can be reinterpreted as:

$$p(\lambda|k) = \frac{e^{-\lambda}(\lambda)^k}{k!} \tag{3.27}$$

Now, optimizing (3.27) with respect to the unknown $\lambda$ yields the result $p(\lambda|k)\left(\frac{k}{\lambda} - 1\right) = 0$. Consequently, the best estimate for the average count rate per unit time $\lambda^{\text{est}}$ is the single sample count $k$. Indeed, it would be difficult to defend any other choice.

In the event that $K$ independent sample counts $k_i$ are taken, what then is the best estimate of the average count rate? Now, the probability of a given joint outcome can be written as:

$$p(\lambda|k_1 \cdots k_K) = \prod_{i=1}^{K} \frac{e^{-\lambda}(\lambda)^{k_i}}{k_i!} \tag{3.28}$$

This time, optimization with respect to $\lambda$ gives the result, with the application of the chain rule, that the sum of the form $p(\lambda|k_1 \cdots k_K) \sum_{i=1}^{K} \left(\frac{k_i}{\lambda} - 1\right) = 0$. Now, the most likely value for $\lambda$ is the sample mean, $\lambda^{\text{est}} = K^{-1} \sum_i k_i$ which is an obvious choice again and one that motivates the idea of estimation through averaging.

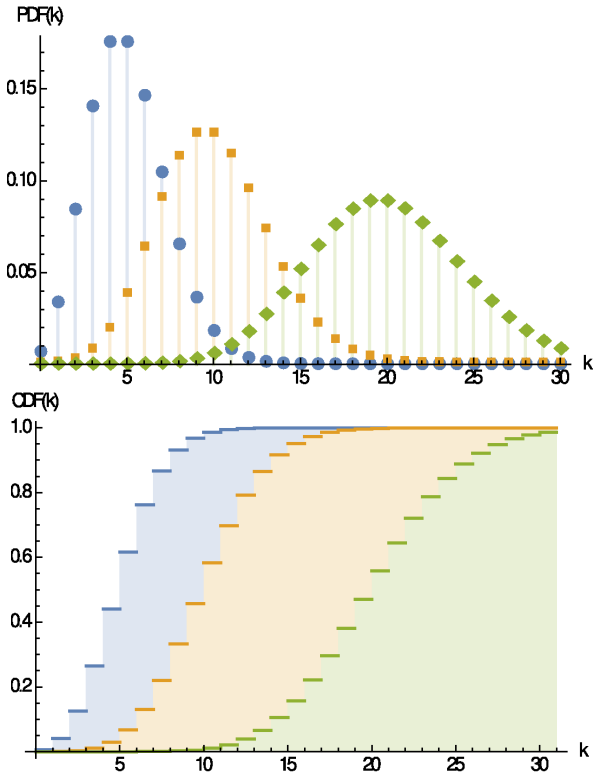


Figure 3.3: Poisson distribution for $\mu = \lambda = 5, 10, 20$. The PDF and CDF are shown above, below.

The question of the accuracy of the $\lambda$ estimates then arises. How reliable are the estimates based on one sample? On many samples? Since expectation of the Poisson distribution is $\lambda$, an estimate of $\lambda$ based on a single sample is an unbiased estimator. The variance of the Poisson distribution is also $\lambda$. The standard deviation is therefore $\lambda^{1/2}$, and the standard deviation as a fraction of the mean, the relative RMS deviation, is $\lambda^{-1/2}$. This can be considered the confidence interval for a single sample estimate. The more cosmic rays you receive in a given time interval, the better you know how many to expect.

Intuitively, the accuracy of the estimate should improve with the number of independent samples that contribute to it. Consider what happens when $K$ single-sample estimates are added together. The expectation of the sum is the sum of the expectation which is just $K\lambda$, and so the sample mean is also an unbiased estimator. What about the sample variance?:

$$\begin{aligned} \mathrm{var}[\lambda^{\mathrm{est}}] &= \left\langle \left[ \sum_{i=1}^{K} (\lambda_i - \lambda) \right]^2 \right\rangle && (3.29) \\ &= K \langle (\lambda_i - \lambda)^2 \rangle && (3.30) \end{aligned}$$

This is the expectation of the sum of the squares of $K$ error terms. Now, if the samples are truly independent, then the cross terms in (3.29) vanish, and only $K$ terms of the form $\langle (\lambda_i - \lambda)^2 \rangle$ survive. In other words, the variance of the sum is the sum of the variance or $K$ times the single-sample variance. The standard deviation of the sum is $\sqrt{K}$ times the single-sample standard deviation, and the relative RMS deviation is $K^{-1/2}$ times the single-sample relative RMS deviation. Ultimately, what matters is the total number of counts collected, meaning the counts per sample times the number of samples. The relative error will be proportional to the reciprocal of the square root of this number.

The cosmic ray problem lies outside the scope of most of the discussion presented in this chapter, there being no system matrix involved in the direct problem. It was examined mainly as a means of demonstrating Bayes' theorem. However, in the event that a number of multi-sample estimate of $\lambda$ were to be made and combined into a data vector $d$ for use in a subsequent inverse problem, we would have a procedure for estimating $C_d$. In this case, the experimental

uncertainty arises from the finiteness of the number of samples being used to construct $d$.

## 3.7 References

## 3.8 Problems

# Chapter 4

# Singular value decomposition (SVD)

In the preceding two chapters, inverse models for over determined, under determined, and mixed determined problems were developed. The approaches used involved first considerations about the Euclidean length of the error and model vectors, then the concept of normal equations, then the concept of resolution and spread, and finally ideas rooted in probability and statistics. In this chapter, the same models are redeveloped, motivated this time by consideration of the properties of vector spaces. The chapter should inspire a deeper perspective into the issues of existence, uniqueness, and stability and to the frequent need for damping, which will henceforth be called regularization. The chapter will also explore some methods for determining the degree of regularization required.

## 4.1 Four vector spaces

A matrix $A \in \mathbb{R}^{n \times m}$ has four fundamental subspaces – the column space and null space of $A$ and $A^T$. The column space of $A$ or $C(A)$ is spanned by the $p$ linearly independent columns of $A$ where $p$ is the rank of the matrix. In the equation $Ax = b$, the left side is linear combination of the columns of $A$, and the right side is a vector that must lie within $C(A)$ for a solution to exist. Likewise in the equation $AX = 0$, the solutions for $X$ span the nullspace of $A$ or $N(A)$. Similar remarks hold for $A^T$. (The column space of $A^T$ is the row space of $A$, and the nullspace of $A^T$ is the left nullspace of $A$). For example, suppose

$$A = \begin{bmatrix} 1 & 0 & a & b \\ 0 & 1 & c & d \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

which is a 2nd-rank matrix written in row reduced echelon form. The matrix has $p = 2$ nonzero rows and two pivot columns. In this case, the bases are very simple:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ span } C(A), \text{ and } \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ spans } N(A^T). \tag{4.2}$$

Likewise, consideration of $A^T$ shows that

$$\begin{bmatrix} 1 \\ 0 \\ a \\ b \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ c \\ d \end{bmatrix} \text{ span } C(A^T), \text{ and } \begin{bmatrix} -a \\ -c \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} -b \\ -d \\ 0 \\ 1 \end{bmatrix} \text{ span } N(A). \tag{4.3}$$

Clearly, the last vector in (4.2) is orthogonal to the other two, and all three vectors are orthogonal and form an orthonormal basis for $\mathbb{R}^3$. While the second two vectors in (4.3) are also orthogonal to the first two, the four vectors

clearly do not form an orthogonal basis for $\mathbb{R}^4$. They are all four linearly independent, however, and so can be transformed into an orthonormal basis through a Gram-Schmidt procedure. The column space and row space both have the same dimension, which is the rank $p$. The nullspace has dimension $m-p$, and the left nullspace has dimension $n-p$.

It is easy to show that the column space and the left nullspace are orthogonal complements and that the row space and the nullspace are also orthogonal complements. Combining $Ax=b$ with $A^TX=0$ gives $X^TAx=X^Tb=0$ so that the vectors in the left nullspace are orthogonal to vectors in the column space. Likewise, combining $A^Tx=b$ with $AX=0$ gives $X^TA^Tx=X^Tb=0$ such that vectors in the row space are orthogonal to vectors in the nullspace.

## 4.2 Fundamental theorem of linear algebra and SVD

The fundamental theorem of linear algebra creates orthonormal bases (through a Gram-Schmidt procedure if necessary) from the four fundamental vector subspaces and expands the matrix in terms of those bases:

$$AV = A\underbrace{\left[(v_1)\cdots(v_p)\cdots(v_m)\right]}_{V} = \underbrace{\left[(u_1)\cdots(u_p)\cdots(u_n)\right]}_{U}\begin{bmatrix}\sigma_1 & & & \\ & \ddots & & \\ & & \sigma_p & \\ & & & 0\end{bmatrix} = U\Sigma \tag{4.4}$$

where the first $p$ of the $v$ basis column vectors span the row space and the remaining vectors span the nullspace. The first $p$ of the $u$ basis column vectors span the column space and the remaining vectors span the left nullspace. Since $U\in\mathbb{R}^{n\times n}$ and $V\in\mathbb{R}^{m\times m}$ are orthogonal, the matrix $A$ can be decomposed as:

$$A = U\Sigma V^T \tag{4.5}$$

which is its singular-value decomposition. The vector $A$ is therefore diagonal with respect to the transformation. The non-negative real-numbered elements of $\Sigma\in\mathbb{R}^{n\times m}$ are called the singular values, and the columns of $U$ and $V$ are known, respectively, as the left- and right-singular vectors. By convention, the singular vectors are normalized, and the singular values appear in non-increasing order.

Geometrically, the transformation $A$ can be thought of as being decomposed into three operations: rotation by $V^T$, scaling by $\Sigma$ and finally rotation by $U$. This is helpful to know, since many matrix properties are invariant under rotation.

Note the resemblance between (4.5) and eigen decomposition. In fact, the left singular vectors are the eigenvectors of $AA^T$ and the right singular vectors are the eigenvectors of $A^TA$. The nonzero singular values are the square roots of the nonsingular eigenvalues of either. This can be seen through the relationships:

$$AA^T = U\Sigma V^TV\Sigma^TU^T = U(\Sigma\Sigma^T)U^T \tag{4.6}$$

$$A^TA = V\Sigma^TU^TU\Sigma V^T = V(\Sigma^T\Sigma)V^T \tag{4.7}$$

where the orthogonality of $U$ and $V$ and the diagonal nature of $\Sigma$ have been exploited. The important difference here is that $A$ need not be square symmetric (since $AA^T$ and $A^TA$ certainly will be).

Lastly, since the elements of $\Sigma$ beyond $p$ are zero, the decomposition in (4.5) can be written in an abbreviated form:

$$A = U_p\Sigma_pV_p^T \tag{4.8}$$

where $U_p\in\mathbb{R}^{n\times p}$, $V_p\in\mathbb{R}^{m\times p}$, and $\Sigma_p\in\mathbb{R}^{p\times p}$. The singular values with zero values remove the nullspace and left nullspace from $U$ and $V$, as is reflected in this formulation. Notice that $U_p$ and $V_p$ are rectangular whereas $\Sigma_p$ is square.

## 4.3 The Moore-Penrose pseudoinverse

Where we read about the matrix $A$ above, we think about the system matrix $G$ and the direct problem $Gm+e=d$. Thinking about the inverse problem motivates the construction of an operator with some of the outward appearances

of the inverse of (4.5):

$$\tilde{G} \equiv V\Sigma^{-T}U^T \tag{4.9}$$
$$\equiv V_p\Sigma_p^{-1}U_p^T \tag{4.10}$$

where $\Sigma^{-T}$ is the pseudoinverse of $\Sigma$ — a matrix with nonzero diagonal elements that are the reciprocals of the singular values. Also $\Sigma_p^{-1}$ is a square diagonal $p \times p$ matrix with rows and columns corresponding to zero singular values removed. As with the singular value decomposition the nullspace and left nullspace do not figure into the pseudoinverse.

Before evaluating the merits of the pseudoinverse, which is certainly not the inverse of the singular value decomposition, it is instructive to explicitly write both the direct problem: both operators:

$$d = \underbrace{\left[\overbrace{(u_1)\cdots(u_p)}^{U_p}\,|\,(u_{p+1})\cdots(u_n)\right]}_{U}\left[\begin{array}{ccc|c}\sigma_1 & & & \\ & \ddots & & \\ & & \sigma_p & \\ \hline & & & 0\end{array}\right]\underbrace{\left[\overbrace{(v_1)\cdots(v_p)}^{V_p}\,|\,(v_{p+1})\cdots(v_m)\right]^T}_{V} m + e \tag{4.11}$$

and its presumptive inverse $m^{\text{est}} = \tilde{G}d$ in terms of their respective operators:

$$m^{\text{est}} = \underbrace{\left[\overbrace{(v_1)\cdots(v_p)}^{V_p}\,|\,(v_{p+1})\cdots(v_m)\right]}_{V}\left[\begin{array}{ccc|c}\sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_p^{-1} & \\ \hline & & & 0\end{array}\right]\underbrace{\left[\overbrace{(u_1)\cdots(u_p)}^{U_p}\,|\,(u_{p+1})\cdots(u_n)\right]^T}_{U} d \tag{4.12}$$

The data are a linear transformation of the model, and the model estimate is a linear transformation of the data. Now, however, the interpretation of the transformations is clear. In predicting the data, the model is projected into the row space and then scaled by the singular values. The resulting weight vector predicts the data as a linear combination of vectors in the column space (with experimental error added).

The existence and uniqueness problems can now be understood clearly. To the extent there exists a nullspace, any component of the model that projects into it will be invisible to the data prediction. It is therefore possible to add any combination of vectors within the nullspace to the model without affecting the prediction. Such a problem is under determined. To the extent there exists a left nullspace, the column space is incomplete, and there will be data vectors $d$ that cannot be reproduced by any weighting of the column space. Such problems are over determined. If $G$ is rank deficient, finite nullspace and left nullspace will exist, and the problem will suffer both limitations and be mixed determined.

The pseudoinverse copes with the aforementioned conditions by constructing model estimates in which the nullspace and left nullspace play no part. Using the pseudoinverse, data vector is projected onto the column space and then scaled by the reciprocal singular values. Components of the data vector that do not project into the column space will be neglected. The resulting weight vector forms the model estimate as a linear combination of vectors in the row space. Once again, components in the null space will not appear in the model estimate. The pseudoinverse is a minimalist model estimate in that regard and is consistent with Occam's Razor.

We can examine the performance of the Moore-Penrose pseudoinverse for all of the classes of problems considered in the preceding chapters of the text. First, it is useful to point out an important property of the singular vectors, namely:

$$U_p^T U_p = V_p^T V_p = I \tag{4.13}$$
$$U_p U_p^T, \quad V_p V_p^T \neq I \quad \text{in general} \tag{4.14}$$

The inequalities in the second line are only satisfied in the event there is no left nullspace ($n = p$) or no nullspace

($m = p$), respectively. Furthermore, we can write

$$\begin{aligned}
\tilde{G}G &= V_p\Sigma_p^{-1}U_p^T U_p\Sigma_p V_p^T && (4.15)\\
&= V_p V_P^T && (4.16)\\
&= I, \quad m = p && (4.17)\\
G\tilde{G} &= U_p\Sigma_p V_p^T V_p\Sigma_p^{-1}U_p^T && (4.18)\\
&= U_p U_P^T && (4.19)\\
&= I, \quad n = p && (4.20)
\end{aligned}$$

which, again, might or might not be unity depending on the existence of a finite nullspace or left nullspace. It is in this sense that the pseudoinverse is not a true inverse. On the other hand, in the event there is neither a nullspace or a left nullspace, the pseudoinverse is a true inverse. The model it estimates exists and is unique. This is an auspicious beginning.

Note that we recognize $R_m = \tilde{G}G$ above as the model resolution matrix and $R_d = G\tilde{G}$ as the data resolution matrix.

### 4.3.1 Over determined problem – least squares

Suppose that there is a finite left nullspace but no nullspace, i.e., $n > p, m = p$. In that case,

$$\begin{aligned}
Gm^{\text{est}} &= U_p\Sigma_p V_p^T V_p\Sigma_p^{-1}U_p^T d && (4.21)\\
&= U_p U_p^T d && (4.22)
\end{aligned}$$

which is unequal to $d$ in general. Evidently, the data estimate will be the projection of the data on the data resolution matrix. If $d$ happens to lie within the column space of $G$, it can be predicted exactly. This will not be the case in general.

As an aside, evaluate the quantity:

$$\begin{aligned}
(G^T G)^{-1} &= (V_p\Sigma_p U_p^T U_p\Sigma_p V_p^T)^{-1} && (4.23)\\
&= (V_p\Sigma_p^2 V_p^T)^{-1} && (4.24)\\
&= V_p\Sigma_p^{-2}V_p^T && (4.25)
\end{aligned}$$

This is a useful quantity in view of the form of the model estimate:

$$\begin{aligned}
m^{\text{est}} &= V_p\Sigma_p^{-1}U_p^T d && (4.26)\\
&= V_p\Sigma_p^{-2}V_p^T V_p\Sigma_p U_p^T d && (4.27)\\
&= (G^T G)^{-1}G^T d && (4.28)
\end{aligned}$$

which makes use of the above equation and is the least squares estimate. The estimate is unique in view of the absence of a nullspace but cannot reproduce the data vector exactly in general due to the existence of a finite left nullspace. This is consistent with our expectations for a model resolution matrix which is unity and a data resolution matrix which is not.

### 4.3.2 Under determined problem – model simplicity

This time, consider the case where a nullspace exists ($m > p$) but no left nullspace ($n = p$). The data estimate will be

$$\begin{aligned}
Gm^{\text{est}} &= U_p\Sigma_P V_p^T V_p\Sigma_p^{-1}U_p^T d && (4.29)\\
&= d && (4.30)
\end{aligned}$$

and so exact reproduction of the data is guaranteed. Insofar as the model estimate goes, however,

$$\begin{aligned} \tilde{G}d &= \tilde{G}Gm && (4.31)\\ &= V_p\Sigma_p^{-1}U_p^T U_p\Sigma_p V_p^T m && (4.32)\\ &= V_p V_p^T m && (4.33) \end{aligned}$$

which will not be equal to the model in general. Evidently, the model estimate will be the projection of the model on the model resolution matrix. If $m$ happens to line within the row space of $G$, it can be estimated exactly. This will not be the case in general.

To analyze the pseudoinverse in this case, we need the results of another auxiliary calculation:

$$\begin{aligned} (GG^T)^{-1} &= (U_p\Sigma_p V_p^T V_p\Sigma_p U_p^T)^{-1} && (4.34)\\ &= (U_p\Sigma_p^2 U_p^T)^{-1} && (4.35)\\ &= U_p\Sigma_p^{-2}U_p^T && (4.36) \end{aligned}$$

The explicit model estimate in this case can then be seen to be:

$$\begin{aligned} m^{\text{est}} &= V_p\Sigma_p^{-1}U_p^T d && (4.37)\\ &= V_p\Sigma_p U_p^T U_p\Sigma_p^{-2}U_p^T d && (4.38)\\ &= G^T(U_p\Sigma_p^{-2}U_p^T)d && (4.39)\\ &= G^T(GG^T)^{-1}d && (4.40) \end{aligned}$$

which is the model simplicity solution. The model estimate is exact in view of the absence of a left nullspace but not unique in view of the finite nullspace. It is consistent with a data resolution matrix which is unity and a model resolution matrix which is not.

### 4.3.3 Mixed determined and ill conditioned problems

At this stage in the analysis, it seems natural to consider the mixed determined case and to seek a model estimate that has the form of damped least squares. In this case, $G$ is rank deficient, $m > p, n > p$ and a finite nullspace and left nullspace coexist. We expect neither exact nor unique solutions from the model estimator based on the pseudoinverse. The analyses leading to (4.22) and (4.31) both hold. Neither the data nor the model resolution matrix will be the identity matrix.

Damping in the damped least squares estimate is required to limit the model solution space for problems that are moderately under determined. Damping also counteracts instability. In the context of the pseudoinverse, the cause of instability is clear. From (4.12), it is evident that small singular values will have the effect of amplifying components of the noise $e$ that project into the corresponding left singular vector. The effect can be severe as the range of singular values can be enormous in practice even in problems that seem superficially benign.

One measure of instability is the model estimate covariance matrix which indicates how errors in the data propagate through to errors in the model. In view of (3.8), the model estimate covariance matrix can be calculated as:

$$C_m = \tilde{G}C_d\tilde{G}^T \quad (4.41)$$

where $\tilde{G}$ is the pseudoinverse or a variant of it, as described below. It has been noted that minimizing the norm of this term is a strategy by itself for constructing inverse methods. That will not be the strategy followed here, however.

#### Condition number

Another measure of the tendency for instability is the condition number. Roughly speaking, this factor expresses the degree of uncertainty amplification inherent in an inverse method. Consider two different realizations of the data, $d$

and $d'$, produces by finite measurement uncertainty $e$. These will give rise to two different model estimates, $m$ and $m'$. The difference between the model estimates is bounded by:

$$\begin{aligned} \|m - m'\| &= \|\tilde{G}(d - d')\| && (4.42) \\ &\leq \|\tilde{G}\|\|d - d'\| && (4.43) \end{aligned}$$

which makes use of the general property of matrix norms that $\|Ax\| \leq \|A\|\|x\|$. Likewise, the size of the data is related to the size of the model by $\|d\| = \|Gm\| \leq \|G\|\|m\|$ so that $\|m\|^{-1} \leq \|G\|\|d\|^{-1}$. Combining this with (4.42) yields an expression relating the relative size of the model and data uncertainties:

$$\frac{\|m - m'\|}{\|m\|} \leq \|G\|\|\tilde{G}\|\frac{\|d - d'\|}{\|d\|} \quad (4.44)$$

The condition number, $\text{cond}(G) \equiv \|G\|\|\tilde{G}\|$, is therefore a measure of the growth of uncertainty which is a symptom of instability.

The preceding analysis is valid for any definition of the matrix norm, but evaluating the condition number requires a choice for the norm to use. Previously in the context of the spread of the resolution matrices, the Frobenius norm, which is an entrywise norm, was considered. A common choice in this context is the spectral norm:

$$\|A\|_2 \equiv \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}(A) \quad (4.45)$$

which is the largest singular value of the matrix. Using this norm, the condition number can be expressed as the ratio of the largest to the smallest of the singular values of $G$, i.e.

$$\text{cond}(G) = \frac{\sigma_1}{\sigma_p} \quad (4.46)$$

Inverse problems with large condition numbers compared to unity are prone to instability and characterized as ill conditioned. In particular, if the base-10 logarithm of the condition number is comparable to or exceeds the precision of the data, inversion cannot proceed.

### TSVD

If unstable inverse model estimates result from the inclusion of singular vectors with small singular values, an obvious method for stabilizing them is simply to exclude those singular vectors in much the same way that the singular vectors with zero singular values were excluded already. This is the method of truncation or, sometimes "brutal truncation." The idea is simply to exclude the singular vectors and corresponding singular values that are resulting is an excessive condition number. In effect, $G$ is decomposed and its pseudoinverse formed as if its rank were less than $p$. Just how much less depends on the degree of stability desired.

Truncated SVD or TSVD is made stable at the loss of row and column space. This means that existence and uniqueness issues are aggravated by TSVD. The method is analogous to limiting the bandwidth of a channel with a low-pass filter to avoid spurious noise and ringing. This is tolerable when the model and data vectors do not project significantly into the nullspace and left nullspace of $G$, respectively.

### Damped least squares

A less abrupt approach to the problem is to apply a filter which is more gradual to a step function, as is generally done in the aforementioned case of low-pass filtering. Filter factors can be applied to the singular values so as to permit them from falling beneath some threshold. This reduces the tendency for instability without completely discarding the parts of the system that make it ill conditioned. While many different choices for the filter factors are available, one choice in particular can be shown to reproduce the method of damped least squares.

We can propose an inverse estimate of the form

$$m^{\text{est}} = VF\Sigma^{-T}U^T d \tag{4.47}$$

where $F \in \mathbb{R}^{m\times m}$ is a diagonal matrix with elements given by

$$F_i \equiv \frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \tag{4.48}$$

for $i < p$ and zero otherwise. Substituting the singular value decomposition of $G$ into the formula for the damped least squares model estimate yields:

$$\left(V\Sigma^T U^T U\Sigma V^T + \alpha^2 I\right) m = V\Sigma^T U^T d \tag{4.49}$$

$$\left(V\Sigma^T \Sigma V^T + \alpha^2 I\right) m = V\Sigma^T U^T d \tag{4.50}$$

We can then substitute (4.47) for $m$ into the left side of the equation and verify that the result is equal to the right side:

$$V\Sigma^T\Sigma V^T VF\Sigma^{-T}U^T d + \alpha^2 VF\Sigma^{-T}U^T d \tag{4.51}$$

$$= V\Sigma^T\left(\Sigma F\Sigma^{-T} + \alpha^2\Sigma^{-T}F\Sigma^{-T}\right)U^T d \tag{4.52}$$

$$= V\Sigma^T U^T d \tag{4.53}$$

which follows from the definition of the filter factors that comprise $F_i$. Consequently, the method of damped least squares can be reproduced using the Moore-Penrose pseudoinverse with the inclusion of the correct filtering. The method is sometimes called Tikhonov or 0th-order Tikhonov regularization when $L = I$, $\alpha$ being the regularization parameter. This is sometimes also called a "water-leveling" strategy. It functions by attenuating the singular vectors with the smallest singular values in a gradual way. Not only is this approach of practical value, it also illustrates more incisively than was possible earlier how damping actually acts to stabilize the inverse model estimate. Note that other choices of filter factors are possible and may be beneficial but will lead to something other than the damped least squares solution.

## 4.4 Application: image compression

Singular value decomposition can be used to produce an optimal approximation of a matrix by another matrix of reduced rank. This makes SVD an ideal tool for data and image compression. The compression procedure is simply to set the smallest of the singular values of the matrix or image to zero, thus reducing the number of singular vectors and left singular vectors that need be stored. This is essentially TSVD.

Take the error between the original image and a compressed copy to be the Frobenius norm of the difference. We have seen that the Frobenius norm is the Euclidean length of the singular values of a matrix. The error norm is therefore the Euclidean length of the singular values that are removed by truncation. Removing the smallest of the singular values therefore compresses the image while making the least contribution to the error norm possible.

Figure 4.1 shows an example of image compression using SVD. The upper-left quadrant of the figure shows a grayscale image with 640×480 pixels in png format. The condition number of the image is of the order of $10^4$, suggesting that quite a few singular vectors and left singular vectors are playing a minor role in the image. To the right is a complete reproduction of the original image using SVD. The error norm for the complete reproduction is of the order of $10^{-8}$ which indicates essentially perfect reproduction.

In the lower left and right quadrants of the figure are images reproduced using just 50 and 25 nonzero singular values, respectively. The error norms in these cases are 0.047 and 0.074, respectively. Very Little of the original image has been lost in either case in this sense. In the sense of human perception, however, the losses are clearer. The text in the image is nearly unreadable in the lower-left panel and is obliterated in the lower right. The gray balance has also shifted considerably throughout the images. Nevertheless, the content is mostly conveyed in the compressed figures. A television program would be completely understandable given even the greater level of compression. As the

Figure 4.1: Illustration of image compression using SVD. (upper left) Original $512 \times 512$ image. (upper right) Complete SVD reconstruction. (lower left) Reconstruction with only 50 nonzero singular values. (lower right) Reconstruction with only 25 nonzero singular values.

compressed image would occupy less bandwidth than the original in a noisy communication channel, the improvement in the signal-to-noise ratio might overshadow the information loss in the overall perceived image quality.

The compression ratio for the figure is given by

$$c = \frac{n \times m}{p \times (n+m+1)}$$

where $m$ and $n$ are the number of rows and columns in the image and $p < m,n$ is the number of singular values retained in the compressed image. The numerator is the total number of image pixels, and the denominator is the total number of elements in the retained left and right singular vectors and singular values. For the $p = 50$ and 25 cases, the compression ratio is 5.11 and 10.23, respectively.

## 4.5 Example: travel-time tomography

The travel time tomography problem introduced in Chapter 1 provides a good demonstration of SVD because it lends itself to easy visualization. We consider the case of sixteen square blocks traversed by ten propagation paths. There are therefore ten equations and sixteen unknowns, and the system appears at first to be purely underdetermined. However, computation of the condition number of the system matrix $G$ produces the number $1.12 \times 10^{16}$. The actual nonzero singular values of the matrix are:

```
   3.47159755e+00, 2.25884939e+00, 2.00000000e+00, 2.00000000e+00, 2.00000000e+00,
2.00000000e+00, 2.00000000e+00, 2.00000000e+00, 9.19570485e-01, 3.09790910e-16
```

The smallest of these is sufficiently small to be considered zero. In fact, the rank of $G$ is effectively nine and not ten, and the matrix is mixed determined. We will regard the smallest singular value as being zero and proceed using TSVD. The condition number then becomes just 3.77.

The top panel of Figure 4.2 shows the model null space of $G$ determined using SVD. Each of these is a column of the matrix $V$ corresponding to a zero singular value. The columns have been rearranged and plotted to resemble the travel time tomography problem physically. Any true model features that project into this space have no bearing on the data and will not be part of the model estimate from the pseudoinverse. The model nullspace is rather extensive for this problem.

The second panel of Figure 4.2 shows the data (left) and model (right) resolution matrices. The data resolution matrix is very nearly the identity. This means that the data predicted by the pseudoinverse model estimate are likely to be very similar to the data upon which the estimate is based. This is a consequence of the fact that $G$ has a small left nullspace. In contrast, the model resolution matrix many nonzero entries off the main diagonal. The model estimate is not precisely mapped to the underlying model and need not resemble it closely. This is a consequence of the existence of a rather significant nullspace.

The third row in Figure 4.2 shows the columns space of $G$. These are the columns of $V$ corresponding to nonzero singular values, rearranged and plotted in 4x4 matrices. To reiterate, there are only nine entries and not ten because we have set the very small nonzero singular value to zero to improve the conditioning of the problem. Any model features which project into this subspace can be reproduced by TSVD.

Finally, the bottom row of Figure 4.2 shows the TSVD estimate of the slowness model for a case where the four inner blocks have true slownesses of unity while the outer blocks have true slownesses of zero. The quality of the reproduction appears to be respectable, as the "thumbtack" quality of the true model is recognizable. This behavior is fortuitous, however, and it would not be difficult to pose slowness models that are reproduced poorly. A "checkerboard" model, for example, is reproduced very poorly by the TSVD pseudoinverse. The fault lies not with the data nor the true model but with the large nullspace of $G$.

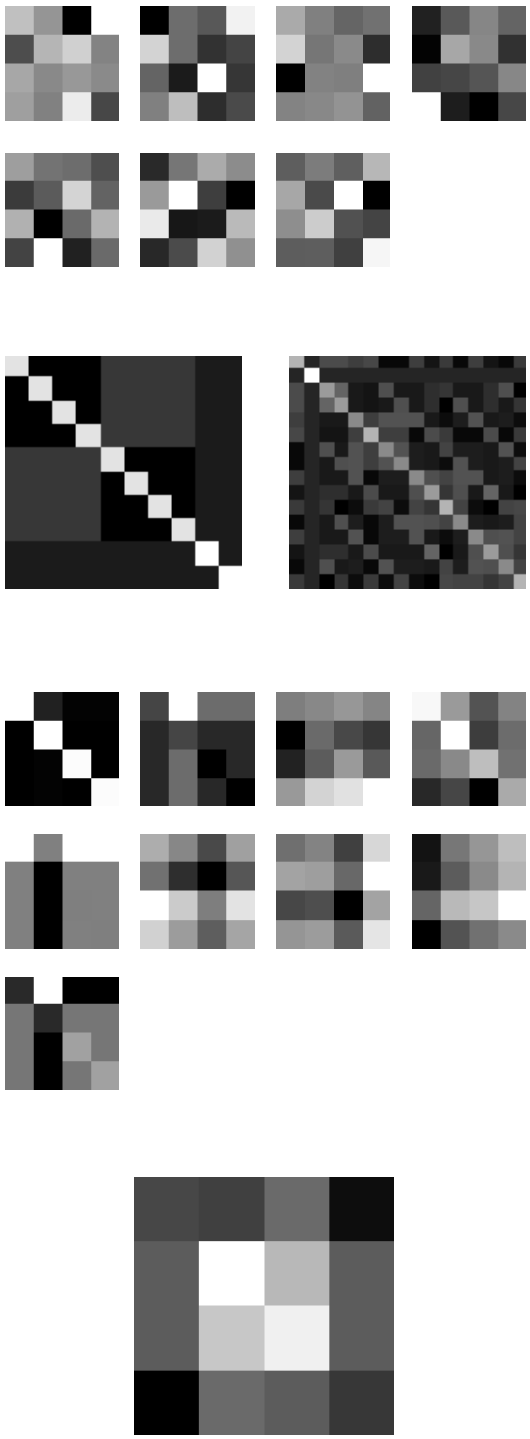

Figure 4.2: Travel time tomography problem example figures. (top row) Model nullspace. (second row) Data (left) and model (right) resolution matrices. (third row) Column space for pseudoinverse solution. (bottom row) Solution for case where the true model slowness is unity in the inner four blocks and zero elsewhere.

## 4.6 Generalized SVD

The Moore-Penrose pseudoinverse can easily be generalized to accommodate arbitrary weighted measures. The prescription for incorporating the data error covariance matrix is given in (2.33), which is to multiply both the system matrix $G$ and the data vector $d$ by $C_d^{-1/2}$ prior to decomposition. Model weights can also be accommodated by replacing the system matrix and data vector with the appropriate augmented versions, as in (2.37). Additional filter factors would be redundant and are unnecessary. In this way, the general weighted damped least squares estimator can be formulated with SVD. First- and second-order regularization can be implemented by using a first- or second-derivative matrix for $L$.

However, this strategy conceals the role of the model weight matrix rather than elucidating it and so defeats the purpose of SVD, which is to promote a deeper understanding of the structure of inverse problems. Furthermore, augmenting $G$ and $d$ as described above is inefficient, as matrix decomposition would have to be performed every time the regularization parameter is changed. This would be computationally expensive. For that deeper insight, we consider a generalization of SVD. Generalized SVD or GSVD will offer a means of modifying the regularization through the simple adjustment of filter factors analogous to (4.48).

The original definition of GSVD is due to Van Loan and considers two matrices $G \in \mathbb{R}^{n\times m}$ and $L \in \mathbb{R}^{p\times m}$ where $n \geq m$. Then there are two orthogonal matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{p\times p}$ and a nonsingular matrix $X \in \mathbb{R}^{m\times m}$ which satisfy

$$U^T G X = \mathrm{diag}(\alpha_1, \cdots, \alpha_m) \tag{4.54}$$

$$V^T L X = \mathrm{diag}(\beta_1, \cdots, \beta_q) \tag{4.55}$$

where $q$ is the smaller of $p$ and $m$ and the $\alpha$s and $\beta$s are non-negative. Here, "diag" refers to the nonzero diagonal elements of a matrix.

A more general formulation that applies to any two matrices with the same number of columns was given by Paige and Saunders. Given matrices $G$, $L$, $U$, and $V$ as above only without the $n \geq m$ restriction and also $W \in \mathbb{R}^{t\times t}$ and $Q \in \mathbb{R}^{m\times m}$,

$$U^T G Q = \Sigma_\alpha \left(W^T R, 0\right) \tag{4.56}$$

$$V^T L Q = \Sigma_\beta \left(W^T R, 0\right) \tag{4.57}$$

where $t$ is the rank of the combined matrix

$$\begin{pmatrix} G \\ L \end{pmatrix} \tag{4.58}$$

$R \in \mathbb{R}^{t\times t}$ is a nonsingular matrix with singular values equal to the singular values of the combined matrix above, and and where $\Sigma_\alpha \in \mathbb{R}^{n\times t}$ and $\Sigma_\beta \in \mathbb{R}^{p\times t}$ are block diagonal with the forms:

$$\Sigma_\alpha = \begin{pmatrix} I_\alpha & & \\ & D_\alpha & \\ & & O_\alpha \end{pmatrix} \tag{4.59}$$

$$\Sigma_\beta = \begin{pmatrix} O_\beta & & \\ & D_\beta & \\ & & I_\beta \end{pmatrix} \tag{4.60}$$

Here, $I_\alpha \in \mathbb{R}^{r\times r}$ and $I_\beta \in \mathbb{R}^{(t-r-s)\times(t-r-s)}$ are identity matrices, $O_\alpha \in \mathbb{R}^{(n-r-s)\times(t-r-s)}$ and $O_\beta \in \mathbb{R}^{(p-t+r)\times r}$ are zero matrices, and the $D$ matrices are diagonal matrices $D_\alpha = \mathrm{diag}(\alpha_{r+1}, \cdots, \alpha_{r+s})$ and $D_\beta = \mathrm{diag}(\beta_{r+1}, \cdots, \beta_{r+s})$. (Here, $r$ is the rank of the combined matrix minus the rank of $L$, and $s$ is the rank of $G$ plus the rank of $L$ minus the rank of the combined matrix). The $\alpha$s and $\beta$s have values between 0 and 1, exclusive, and appear in descending and ascending order, respectively, so as to satisfy $\alpha_i^2 + \beta_i^2 = 1$.

The connection to the formulation by Van Loan can be clarified by writing

$$U^T G X = (\Sigma_\alpha, 0) \tag{4.61}$$
$$V^T L X = (\Sigma_\beta, 0) \tag{4.62}$$
$$X = Q \begin{pmatrix} R^{-1}W & 0 \\ 0 & I \end{pmatrix} \tag{4.63}$$

where $X \in \mathbb{R}^{m \times m}$. This is clearly in the form of (4.54) above. Furthermore, this implies a compact way of writing the GSVD of $G$ and $L$:

$$G = U (\Sigma_\alpha, 0) X^{-1} \tag{4.64}$$
$$L = V (\Sigma_\beta, 0) X^{-1} \tag{4.65}$$

which can further be used to show that

$$G^t G = X^{-T} \begin{pmatrix} {\Sigma_\alpha}^T \Sigma_\alpha & 0 \\ 0 & 0 \end{pmatrix} X^{-1} \tag{4.66}$$
$$L^t L = X^{-T} \begin{pmatrix} {\Sigma_\beta}^T \Sigma_\beta & 0 \\ 0 & 0 \end{pmatrix} X^{-1} \tag{4.67}$$

which is in the form of an eigen expansion of $G^T G$ and $L^T L$, both utilizing the same eigenvectors. The columns of $X$ play the role of the generalized singular vectors of the combined matrix. The generalized singular vales are the ratios $\gamma_i = \alpha_i / \beta_i$, of which $r$ are infinite, $s$ are finite, and $t - r - s$ are zero.

We may now use the GSVD of $G$ and $L$ to solve the damped least squares problem given an arbitrary form of $L$ and coefficient $\alpha$. The generalized pseudoinverse is given by

$$\hat{G} = X (\Sigma_\alpha, 0)^{-1} U^T \tag{4.68}$$

where the nonzero elements of $(\Sigma_\alpha, 0)^{-1}$ are the reciprocals of the nonzero elements of $(\Sigma_\alpha, 0)$. In analogy with (4.47), we can propose a similar form for filter factors to construct a generalized damped least squares estimator from the generalized pseudoinverse. Proceeding as in the derivation immediately following (4.47) and making use of (4.66) and (4.67) reveals that the correct form for the filter factors in this case is:

$$f_i = \frac{\gamma_i^2}{\gamma_i^2 + \alpha^2} \tag{4.69}$$

Using this formula, 1st- and 2nd-order Tikhonov regularization involving the 1st- and 2nd-derivative matrices for $L$, respectively, can be implemented. (These strategies are known collectively as higher-order Tikhonov regularization.) As with SVD, a number of libraries for the efficient computation of the matrices involved in the GSVD pseudoinverse are available. As with SVD, truncation is also an option for GSVD.

## 4.7 Regularization strategies

"Regularization" is not a property of SVD but is a general term in inverse methods applying to any of the methods in which damping appears. Damping is generally introduced in conjunction with a constant factor, an $\alpha$-parameter. In fact, all of the formulations of the damped or weighted damped least squares problems considered thus far have contained such a factor. Generically, the linear discrete inverse problem solves the (now familiar) problem:

$$Gm + e = d \tag{4.70}$$

for $m$ where the measurement error $e$ is unknown, although their statistics should be well characterized. The solution strategy can be stated as:

$$m^{\text{est}} = \underset{m}{\operatorname{argmin}} \|Gm - d\|_2^2 + \alpha^2 \|Lm\|_2^2 \tag{4.71}$$

where the data error covariance matrix $C_d$ has been absorbed into the definitions of $G$ and $d$ and the prior model estimate $m_\circ$ has been omitted for the sake of simplicity. The objective function being minimized here includes the residual and a measure of the length of the model solution vector. The solution is informed by the data but also restricted in such a way to yield unique and stable solutions. The solution to this problem has been found to be (through the method of normal equations for example)

$$\begin{aligned} m^{\text{est}} &= \left(G^T G + \alpha^2 L^T L\right)^{-1} G^T d & (4.72)\\ &= \tilde{G} d & (4.73) \end{aligned}$$

Equivalently, $\tilde{G}$ could be expressed in terms of the SVD or GSVD pseudoinverse (or probabilistically). In any case, the solution represents a balance between accuracy, as measured by the residual, and stability, as measured by the model length.

**Perturbation vs. bias** Nothing in the problem statement above directly informs the choice of $\alpha$, however. We can consider two measures of the model estimate — the model error and the data prediction error:

$$\begin{aligned} m - m^{\text{est}} &= m - \tilde{G} d & (4.74)\\ &= m - \tilde{G}(Gm + e) & (4.75)\\ &= m(I - \underbrace{\tilde{G}G}_{R_m}) - \tilde{G}e & (4.76)\\ d - d^{\text{est}} &= Gm + e - Gm^{\text{est}} & (4.77)\\ &= G(m - m^{\text{est}}) + e & (4.78)\\ &= G[m(I - R_m) - \tilde{G}e] + e & (4.79)\\ &= Gm(I - R_m) - \underbrace{G\tilde{G}}_{R_d} e + e & (4.80)\\ &= Gm(I - R_m) + e(I - R_d) & (4.81) \end{aligned}$$

Both error metrics are composed of two terms: one which involves the measurement error and one that does not. The term involving the measurement error reflects perturbations to the inverse solution arising from those errors. The stability of the solution rests on the size of these perturbation errors. The terms which do not involve measurement errors instead involve either the model or the data resolution matrix and vanish in the event the given resolution matrix is the identity. These terms represent estimate biases arising from finite resolution and the incompleteness in some sense of the inverse operator $\tilde{G}$.

Consider an inverse method based on TSVD where the regularization parameter is the number of singular values/vectors to retain in the inverse. The more terms that are retained, the closer the model and data resolution matrices are to the identity, and the smaller the biases. At the same time, the more terms that are retained, the larger the norm of $\tilde{G}e$, and the larger the perturbation errors. Ideally, the number of terms to retain could be based on balancing the relative sizes of the biases and perturbation errors. Although $e$ is unknown, its covariance should be. This basic reasoning underlies many regularization strategies based on SVD and GSVD, length methods, statistical methods, or (as will be seen) iterative methods.

As an aside, it should be noted that the treatment of error analysis and error propagation at the start of chapter 3 applied to perturbation error only and considered how observational errors telegraph through the inverse problem to uncertainties in the model estimate. Biases were never considered. Increased regularization reduces the effects of perturbations and the size of the model error covariance matrix along with them. Error bars calculated using the prescription in chapter 3 will get smaller as regularization gets larger, always. This does not necessarily mean that our confidence in the model estimates should increase, however. Biases degrade confidence as well and need to be considered explicitly. The important subject will be revisited later in the text.

**Discrete Picard condition** Another metric for assessing stability in discrete linear systems, the condition number, has already been discussed. The condition number is an assessment of the system by itself without consideration of the

particular data driving it. Regularization can be added until the overall system, including filter factors, is reasonably well conditioned. Otherwise, the entire system can be reconstructed using some kind of factorization to improve the conditioning (i.e. preconditioning).

Still another metric rooted in SVD is the discrete Picard condition. This condition considers both the system and the actual dataset. In terms of the Moore-Penrose pseudoinverse (or its GSVD equivalent),

$$m^{\text{est}} = V_p \Sigma_p^{-1} U_p^T d \tag{4.82}$$

the condition is satisfied when the elements of the vector $U_p^T d$, sometimes called the "Fourier coefficients," decrease faster than the elements of $\Sigma_p$, so that the (possibly weighted or truncated) elements of the quotient vector $\Sigma_p^{-1} U_p^T d$ decrease sequentially rather than grow. This means that the components of the data most apt to be amplified are themselves naturally limited. Nature can be conducive to stability when it removes the most troublesome components of the data, for example through diffusive or viscous damping of high-frequency components, but such favorable circumstances do not prevail as a rule. To satisfy the Picard condition under more common circumstances, the regularization can be augmented or otherwise adjusted.

Below, the various metrics for model performance are combined into specific regularization strategies. No one strategy consistently outperforms the others. Performance will be most satisfactory when the strategies can be applied to multiple instances of observations.

### 4.7.1 Morozov's Discrepancy Principle

Under the assumption that the optimum model estimate should exactly predict the observed data in the absence of observation error, it follows that the size of the residual should be governed by the size of the observation error, i.e.,

$$\|Gm^{\text{est}} - d\| = \|e\| \tag{4.83}$$

Morazov's discrepancy principle states simply that the regularization parameter in the weighted damped least squares problem should be set to enforce this equality. Of course, we know neither $e$ nor its norm, $e$ being a random variable. The statistical properties of $e$, the error covariance $C_d$ in particular, are generally characterizable. In particular, if the observation errors are normally distributed, then $(Gm-d)^T C_d^{-1}(Gm-d)$ is Chi-squared distributed with an expected value $E = n$. (In the context of regularization, we use $n$ rather than $n-m$ as the number of degrees of freedom.)

If we consider the function $f(\alpha) \equiv \|Gm_\alpha^{\text{est}} - d\|_2^2 - E$, where $m_\alpha^{\text{est}}$ is the model estimate based on a given choice of $\alpha$, the discrepancy principle becomes the problem of finding the root of $f(\alpha)$. Such problems are discussed later in the text. They are solved iteratively and can be computationally expensive. Fortunately, the SVD need not be computed for each iteration since different instances of the model estimate differ only in their filter factors.

Of course any given vector $\|e\|$ is a realization of a random vector drawn from a statistical distribution. If $e$ is normally distributed, $\|e\|_2^2$ is Chi-squared distributed, and so the equivalence being enforced is only an estimate.

### 4.7.2 UPRE method

The "unbiased predictive risk estimator" is a method for setting the regularization parameter in Tikhonov problems so as to mitigate biases and perturbation errors simultaneously. We define the predictive error as

$$\begin{aligned} p_\alpha &= G(m_\alpha^{\text{est}} - m) && (4.84)\\ &= Gm_\alpha^{\text{est}} - (d-e) && (4.85)\\ &= G\tilde{G}_\alpha d - (d-e) && (4.86)\\ &= (A_\alpha - I)(d-e) + A_\alpha e && (4.87)\\ &= (A_\alpha - I)Gm + A_\alpha e && (4.88) \end{aligned}$$

where the symmetric matrix $A \equiv G\tilde{G}$ is the so-called influence matrix and also the data resolution matrix, $R_d$, and the $\alpha$ subscripts imply inverses computed for the given value of the regularization parameter. Notice that the two components of the predictive error pertain to biases and perturbation errors, respectively.

The goal of the method is to select $\alpha$ to minimize the predictive risk, the expectation of the norm of the predictive error, $\|p_\alpha\|^2/n$.

$$E\left(\frac{1}{n}\|p\|_2^2\right) = \frac{1}{n}\|(A_\alpha - I)Gm\|_2^2 + \frac{\sigma^2}{n}(A_\alpha^T A_\alpha) \tag{4.89}$$

where the elements of the observation noise vector have been taken to be independent and normally distributed with variance $\sigma^2$. However, since the true model $m$ or, equivalently, the error $e$ in the predictive error are not accessible, (4.89) cannot be optimized as stated. What is accessible is the residual vector which involves only the observed data and the model estimate derived from them:

$$\begin{aligned} r &= Gm_\alpha^{\text{est}} - d && (4.90)\\ &= G\tilde{G}_\alpha d - d && (4.91)\\ &= (A_\alpha - I)d && (4.92)\\ &= (A_\alpha - I)Gm + (A_\alpha - I)e && (4.93) \end{aligned}$$

What makes the UPRE method possible is the fact that the expectation of the norm of the residual is closely related to the predictive risk. This becomes clear with the incorporation of the **trace lemma**, which applies to deterministic vectors $f$ and operators $B$ in a Hilbert space and random vectors $\eta$ with independent, normally-distributed elements with variances $\sigma$:

$$E\|f + B\eta\|^2 = \|f\|^2 + \sigma^2 \operatorname{Tr}(B^T B) \tag{4.94}$$

Accordingly, the expectation of the norm of the residual vector can be written as:

$$E\left(\frac{1}{n}\|r\|^2\right) = \frac{1}{n}\|(A_\alpha - I)Gm\|^2 + \frac{\sigma^2}{n}\operatorname{Tr}(A_\alpha^T A_\alpha) - 2\frac{\sigma^2}{n}\operatorname{Tr}(A_\alpha) + \sigma^2 \tag{4.95}$$

Finally, comparison of (4.95) with (4.95) shows how the predictive risk can be constructed from the residual norm. We define the UPRE objective function

$$U(\alpha) = \frac{1}{n}\|r\|^2 + \frac{2\sigma^2}{n}\operatorname{Tr}(A_\alpha) - \sigma^2 \tag{4.96}$$

which can be minimized with respect to the regularization parameter $\alpha$.

Ultimately, the method amounts to minimizing a combination of the residual norm and something like the spread of the data resolution matrix that depends on the size of the noise. The main steps in applying the UPRE method are the computation of $m_\alpha^{\text{est}}$ using SVD or GSVD and also the trace of the influence matrix. The computational cost may be prohibitive for large inverse problems such as are involved in image processing. As with the discrepancy principle, this method also equates the residual norm with its expectation, which is only an approximation.

### 4.7.3 L-curve

Perhaps the most popular method for determining the optimal value for the regularization parameter is the L-curve method which is highly intuitive and easy to implement. This method reflects an attempt to find the an acceptably small value of $\chi$-squared while simultaneously suppressing artifacts in the model estimate. Too large a value of $\chi$-squared implies an inadequate ability of the model estimate to reproduce the data. Too small a value implies that the method may be "fitting the noise," producing spurious features that are not actually supported by the data. An indication of the latter effect is an inordinately large value of the L2 norm of the metric $Lm$.

The L-curve method involves solving the inverse problem repeatedly for different regularization parameters $\alpha$. The results of each solution are plotted parametrically on a graph, where $\alpha$ is the parameter. The axes of the plot are

$||Gm - d||_2$, the residual norm, and $||Lm||_2$, the model norm, respectively. The axes may be linear, logarithmic, or a combination. The squares of the norms might also be plotted. When performed successfully, the method reveals a curve with an elbow or an "L" where the residual norm and the model norm are both near their respective minima. (On a log-log plot, the elbow corresponds to a region of maximum curvature which could be identified computationally.) At the corresponding value of $\alpha$, the model estimate reproduces the data reasonably accurately without exhibiting obvious spurious artifacts.

The L-curve method has two main drawbacks. The first is that it can be computationally expensive since the model estimate must be computed multiple times for multiple values of $\alpha$. This problem is mitigated substantially for inverses based on SVD and GSVD since only the filter factors need be changed for each computation according to (4.48) or (4.69) as the case may be.

Another problem with the L-curve method is that there may be no pronounced elbow in the curve it produces. Even when there is an elbow, the method still only gives general guidance about the optimum choice for $\alpha$. In practice, annalists often find an optimum value of $\alpha$ by some other means and then use the L-curve method afterward as a sanity check. Having a prior estimate of $\alpha$ a priori also helps reduce the computational cost of the method. Examples of the method illustrating its efficacy follow below and later throughout the text.

### 4.7.4 GCV

Another method for setting the regularization parameter with an intuitive and compelling motivation is generalized cross validation or GCV. It is a common procedure in inverse problems to utilize only a fraction of the available data, the "training" data, for computing the model estimate. The resulting $m^{\text{est}}$ can then be used to predict the remaining data which were withheld for the purpose of validation. Setting the regularization parameter becomes part of the validation process. This strategy, called cross validation, has obvious merits but two deficiencies. First, depriving the model estimate of the full dataset surely decreases its capabilities. Second, the procedure for deciding which and how many data to withhold for validation is unclear at best.

Suppose, however, that $n$ model estimates are computed, each with a single datum excluded from the computation and subsequently used to validate the corresponding model estimate. An objective function based on the sum of the prediction errors could be calculated. Suppose further that the entire procedure is repeated for different values of the regularization parameter, $\alpha$, which could be set to minimize the overall residual. Finally, the optimal $\alpha$ would be used for the model estimate together with all the available data. This is generalized cross validation. It would seem to have all the advantages of the aforementioned cross validation scheme with none of the disadvantages. The added disadvantage, potentially, is computational cost, since the model estimates would have to be computed for every value of $\alpha$ and every excluded datum individually. As discussed above, inverse methods based on SVD and GSVD are ideal in such applications since the matrix decomposition only need be performed once. In addition, another short-cut reduces the cost of the GCV procedure even further.

Let us define $m^{\text{est}}_{\alpha,k}$ as the weighted damped least squares model estimate for a given regularization parameter $\alpha$ based on a truncated data vector from which the $k$th element has simply been removed. This model estimate also solves another closely-related optimization problem:

$$m^{\text{est}}_{\alpha,k} \quad = \quad \underset{m}{\text{argmin}} \, ||Gm - d^k||_2^2 + \alpha^2 ||Lm||_2^2 \tag{4.97}$$

where $d^k$ this time is the data vector in which the $k$th element has been replaced by its model prediction, the $k$th element of the vector $Gm$. In either case, the contribution to the residual of the $k$th data element is zero, and so the regression problem is unaffected. The equivalence of the two problems relies on the particular definition of $d^k$ and is known as the "leave-one-out lemma." It permits the modified data vector $k'$ to retain the dimension $n$ of the original data vector which is necessary for the analysis that follows.

As we have seen many times now, the solution to (4.97) can be stated in terms of a pseudo inverse $\tilde{G}$ found using any number of methods including generalized SVD.

$$m^{\text{est}}_{\alpha,k} \quad = \quad \tilde{G}_\alpha d^k \tag{4.98}$$

Note, however, that this is an implicit equation because of the dependence of $d^k$ on the model estiamte. Evaluating $m^{\text{est}}_{\alpha,k}$ would appear to require iteration. Fortunately, evaluation is not actually necessary for GCV.

The generalized cross validation solution corresponds to the minimum of the global leave-one-out residual vector with respect to the regularization parameter:

$$m^{\text{est}} = \underset{\alpha}{\text{argmin}} \frac{1}{n} \sum_{k=1}^{n} \left( (Gm^{\text{est}}_{\alpha,k})_k - d_k \right)^2 \tag{4.99}$$

where the subscripts refer to the $k$th value of the given vector. As written, (4.99) implies looping operations over both the regularization parameter and the index $k$. Such an operation would be computationally expensive and impractical. However, a considerable simplification comes about from the fact that the difference between the $k$th residual computed using and not using the $k$th data element depends only on the $k$th diagonal element of the data resolution matrix, i.e.

$$(G\tilde{G}_\alpha d^k)_k - (G\tilde{G}_\alpha d)_k = (G\tilde{G}_\alpha)_{kk}(d^k_k - d_k) \tag{4.100}$$

or (after some rearranging)

$$\frac{(Gm^{\text{est}}_\alpha)_k - d_k}{1 - (G\tilde{G}_\alpha)_{kk}} = (Gm^{\text{est}}_{\alpha,k})_k - d_k \tag{4.101}$$

where we note that $m^{\text{est}}_{\alpha,k} \equiv \tilde{G}_\alpha d^k$ and $m^{\text{est}}_\alpha \equiv \tilde{G}_\alpha d$. Eq. (4.99) can be rewritten accordingly, this time without the leave-one-out terms, as

$$m^{\text{est}} = \underset{\alpha}{\text{argmin}} \frac{1}{n} \sum_{k=1}^{n} \left( \frac{(Gm^{\text{est}}_\alpha)_k - d_k}{1 - (G\tilde{G}_\alpha)_{kk}} \right)^2 \tag{4.102}$$

Note next that $(G\tilde{G})_{kk}$ could change if the rows of $G$ were permuted, which is to say that it depends on the ordering of the data. By convention, the denominator of (4.102) is therefore replaced with its average, and the summation is expressed in terms of the usual L2 vector norm, yielding the final GCV optimization problem to be solved:

$$m^{\text{est}} = \underset{\alpha}{\text{argmin}} \frac{n\|Gm_\alpha - d\|_2^2}{\text{Tr}(I - G\tilde{G}_\alpha)^2} \tag{4.103}$$

which is explicit and no more difficult to compute than an L-curve.

The GCV method of regularization is closely related to the other methods considered in this text. Notice that the denominator of (4.103) is related to the spread of the data resolution matrix. GCV therefore seeks to optimize the regularization parameter by balancing the data resolution spread and the error norm. For the method to succeed, there must be a well-defined minimum in the curve implied by (4.103). If not, performance may be improved by applying different weights in the filter factor terms in (4.103).

### 4.7.5 NCP method

When the regularization parameter is set correctly, the residual vector should contain only observation noise, and the correct setting of the regularization parameter is the smallest value for which this condition is met. One of our expectations for the observation noise is that it be white, i.e., spectrally flat. The NCP method involves calculating the spectrum of the residual vector using a periodogram, a discrete Fourier transform in practice, and testing whether or not the spectrum is white. To the degree it is dominated by low- or high-frequency components, the residual has a bias, and the regularization parameter is improperly set.

From the discrete Fourier transform of the residual vector, power as a function of frequency can be computed. "NCP" stands for "normalized cumulant periodogram." The cumulant power $c(f_i)$ is the total power in all the spectral bins up through the frequency bin $f_i$ which is obviously an increasing function of $i$. The normalized cumulant is $c(f_i)/c(f_n)$, where $f_n$ is the highest frequency bin. Thus, the normalized NCP($f_i$) takes on values between 0 and 1 in increasing order.

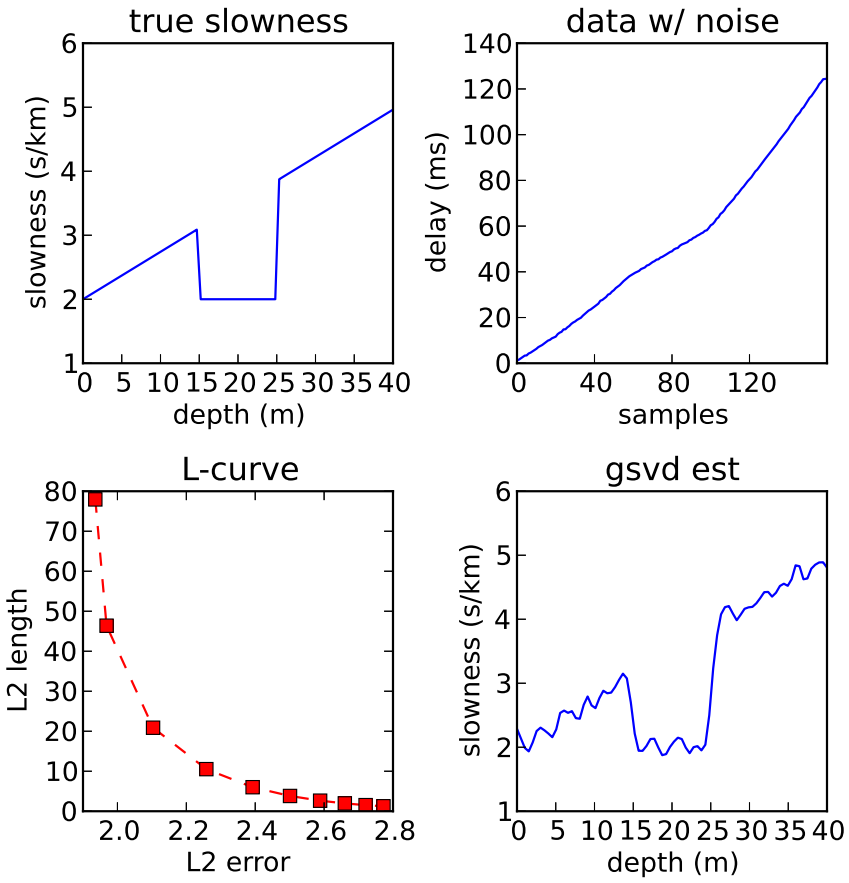


Figure 4.3: Solution of the vertical seismic profiling problem discussed earlier, this time using generalized SVD and second-order Tikhonov regularization. The L-curve represents regularization parameters increasing from left to right from 0.1 – 1.0. The solution shown in the lower-right panel corresponds to a regularization parameter of 0.4.

In the event that the noise in the cumulant is white, the NCP will approximate a straight line increasing with $i$ from 0 to 1. Convex or concave shapes, meanwhile, will indicate dominance of the power spectrum by low- or high-frequency components and signify bias in the residual. The technique then involves finding the regularization parameter $\alpha$ that gives rise to the most linear NCP. In practice, the NCP for a number of realizations of the residual vector are plotted or otherwise considered together. Some numerical strategy for optimizing the linearity of the NCP is also required.

## 4.8 Example: vertical seismic profiling

We can return to the vertical seismic profiling problem considered earlier for an application for GSVD. The results of the analysis for second-order Tikhonov regularization for which $L$ is the 2nd-derivative matrix are shown in Figure 4.3. The true slowness model and synthetic data (with added noise) plotted here are the same as those from Figure 2.1.

Analysis by GSVD reveals that one of the generalized singular values is much larger than the others. This suggests that simple truncation might be an appropriate solution strategy. Pursuing Tikhonov regularization instead, an L-curve was generated for values of $\alpha$ spanning 0.1–1.0 in steps of 0.1, introduced through filter factors in the manner of (4.69). The optimal value was taken to be $\alpha = 0.4$, and the corresponding solution appears in the lower-right corner of

Figure 4.3.

Results are comparable to those resulting from the application of simple length methods. Since different regularization parameters can be tested without the need for complicated matrix operations, the GSVD method is the more efficient choice here where an L-curve has been generated. As will be seen later, even more efficient strategies exist for solving linear regularization problems like this. GSVD is most useful when direct examination of the singular values and vectors for diagnostic purposes is desired.

## 4.9 Example: instrument function deconvolution

The prototypical problem in linear inverse methods is the deconvolution problem. Every scientific instrument has an instrument function that maps the state of the system into the observations, and the mapping is often a form of convolution. Seismometers for example measure the displacement of a weight experiencing mechanical forcing. The finite inertia of the weight limits the response time of the instrument leading to broadening of the signals that are finally observed following a mechanical perturbation. Accurate estimation of the perturbation requires that the instrument function be removed from the observed signal through deconvolution.

Consider an instrument with an impulse response function (or Green's function) given by

$$g(t) = \begin{cases} g_\circ t e^{-t/t_\circ}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{4.104}$$

such that the output is related to the input by the convolution integral

$$s_{\text{out}}(t) = \int_{-\infty}^{t} s_{\text{in}}(\tau) g(t-\tau)\, d\tau \tag{4.105}$$

In terms of a discrete inverse problem, the observational data are the output signals sampled at discrete times $t_i$, the model values are the input signals evaluated at discrete times $t_j$, and the system itself is

$$G_{ij} = \begin{cases} g_\circ (t_i - t_j) e^{-(t_i - t_j)/t_\circ} \Delta t, & t_i \geq t_j \\ 0, & t_i < t_j \end{cases} \tag{4.106}$$

where $\Delta t$ is the time between samples.

Figure 4.4 shows a worked example for the deconvolution problem. Here, $t_\circ$ = 12 s. The problem runs for 100 s, and samples are taken at half-second intervals. The top row of the figure shows the input to the system, the data that would be measured according to the system prescribed by (4.106), and data with added noise ($\mu$ = 0, $\sigma$ = 0.2). The input exhibits two distinct components, but these are conflated by the impulse response of the instrument and rather hard to distinguish in the observable output.

The second row shows the singular values for the system, the pseudoinverse reconstruction of the input signal for noiseless data, and the reconstruction for data with noise. In fact, the smallest of the singular values is much smaller than the others (not shown), and so the the system has been truncated with the removal this smallest singular value and the associated singular vectors. Even so, the condition number of the truncated system is of the order of $10^3$, which is quite large. The result is instability, which is clearly evident in the difference between the two reconstructions despite the relatively small amount of noise that distinguishes them.

The third row of Figure 4.4 shows the noisy signal recovery based on TSVD retaining only the 25 largest singular values and associated singular vectors together with the model resolution matrix for the truncated system and a one-dimensional cut through that matrix. While the recovery is far superior to the previous one, it still suffers from a substantial degree of spurious ringing, and the two features are significantly broader than the originals. The model resolution matrix shows why this is. The main diagonal of $R_m$ is fairly broad, prohibiting the reconstruction of fine detail, and significant lines off the main diagonal underlie the ringing.

The bottom row of Figure 4.4 illustrates improved signal recovery using 0th-order Tikhonov regularization where the L-curve method is used to set the regularization parameter. The L-curve has an elbow in the vicinity of $\alpha$=1. The

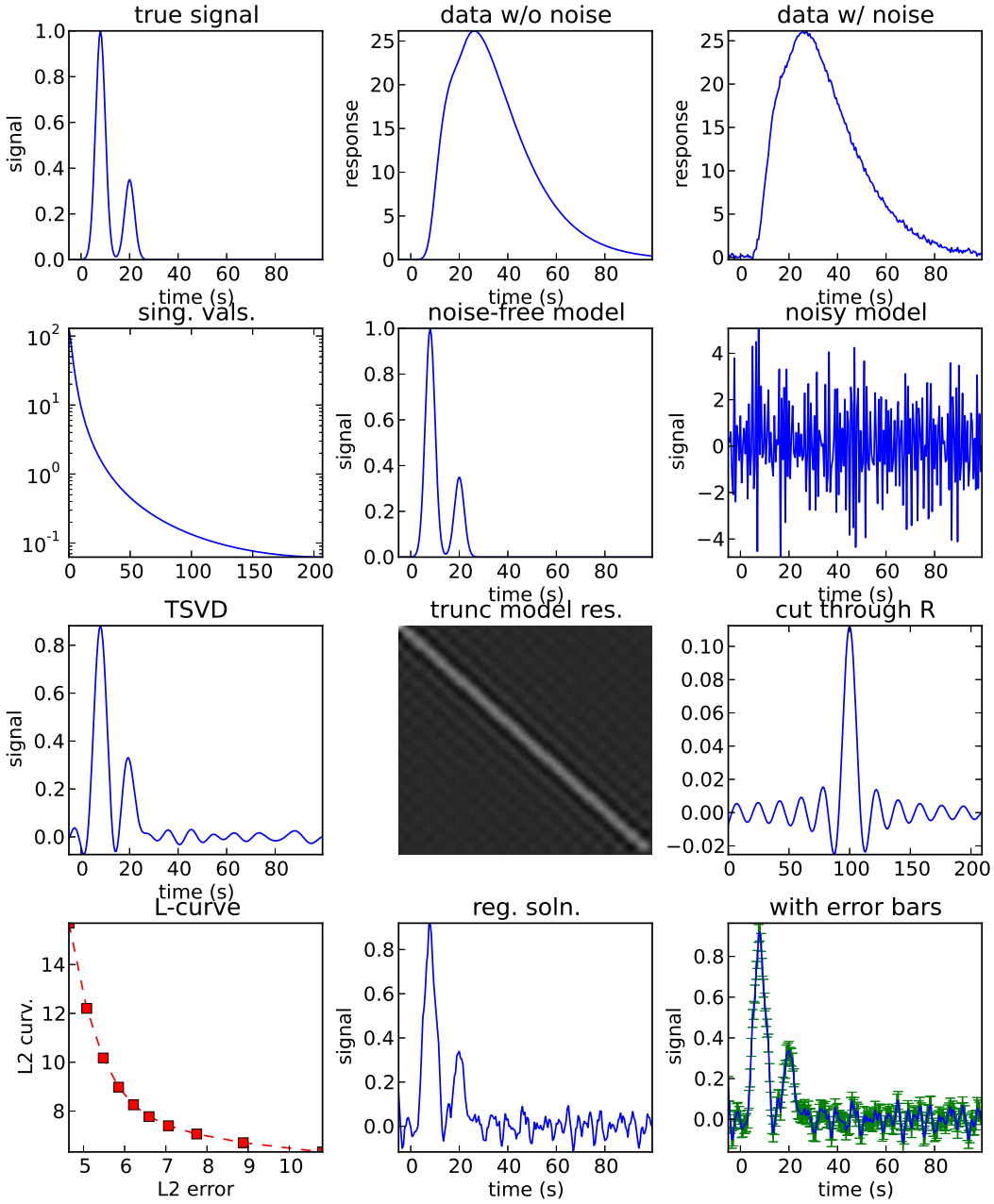


Figure 4.4: Illustration of the numerical deconvolution problem. The top row of the figure shows the true signal, synthetic data without noise, and synthetic data with noise added. The second row shows the singular values for the system (less the smallest singular value), the reconstruction of the noiseless data, and the reconstruction of the noisy data. The third row shows the reconstruction using TSVD (with just 25 singular values/vectors), the model resolution matrix for the truncated system, and a cut through the model resolution matrix. Finally, the bottom row shows the L-curve analysis, the Tikhonov solution for the optimal regularization parameter, and the result with model confidence limits (error bars) superimposed.

model estimate for this value of *alpha* together with the associated model confidence limits (error bars) taken from the diagonal elements of $C_m$ are also plotted. In the Tikhonov solution, the two peaks are are taller and narrower than those in the TSVD solution, and the ringing is greatly suppressed. Small, spurious features in the solution appear to be well characterized by the error analysis.

## 4.10 References

## 4.11 Problems

# Part II: Explicit, continuous and semi-continuous methods

# Chapter 5

# Continuous and semi-continuous problems

The discussion thus far has concerned fully discrete linear inverse problems in which both the model and that data have been vectors. Such problems are amenable to solution by digital computers, and the focus thus far has been on computational methods. Inverse methods are equally applicable to problems where the model, the data, or both are continuous functions, however. If analog computing were still in vogue, continuous inverse problems would probably enjoy more popularity and occupy the introductory chapters of this text! As analog computers exist primarily in museums today, continuous problems are generally transformed into discrete ones prior to being solved. The next chapter in this text discusses means of discretization. In this chapter, continuous and semi-continuous inverse problems are entertained but briefly.

If the foundation of discrete inverse problems is the fundamental theorem of linear algebra, then the foundation of continuous problems is the fundamental theorem of calculus. Methods for solving continuous inverse problems are the methods for solving integral equations. Below, a few important examples pertaining to remote remote sensing are examined.

## 5.1 The matched filter

The matched filter is common in signal processing and in radar applications in particular. (So widespread is matched filtering in radar that all other approaches are relegated to the moniker "unmatched filtering.") The filter is "matched" to the anticipated signal so as to optimize the signal-to-noise ratio at its output. Matched filtering can furthermore be used to deconvolve from a received radar signal the modulation that was applied to the transmitted waveform. The filter in this case is matched to the modulation. Unlike other deconvolution strategies considered in this text, not only the filter but also the modulation is optimized since the radar engineer can choose $G$.

The theory of matched filters is often developed using continuous functions, and practical matched filters can be implemented using analog equipment. It is for this reason that matched filters are being discussed in this chapter. For the sake of expediency, however, the topic will be treated here using discrete mathematics. Similar remarks hold for other sections of the chapter.

In terms of the customary equation $d = Gm + e$, $G$ describes the modulation of the waveform in time used to illuminate a target, $m$ is the scattering cross section as a function of range, $e$ is sample noise, and $d$ is the received signal in time prior to filtering. The matrix-vector product is a convolution operation. The filtered signal is $m^{\text{est}} = \tilde{G}d = \tilde{G}Gm + \tilde{G}e$ where the former component represents signal and the latter component noise.

Suppose we consider just the $j$th value of $m^{\text{est}}$. Then only the $j$th row of the filter $\tilde{G}$ is involved. Denote this row

as the row vector $\tilde{G}_j$. The ratio of the power in the signal (which arises from $Gm$) to the noise (which arises from $e$) is:

$$\text{SNR} = \frac{|\tilde{G}_j Gm|^2}{\langle|\tilde{G}_j e|^2\rangle} \tag{5.1}$$

$$= \frac{|\tilde{G}_j Gm|^2}{\tilde{G}_j C_d \tilde{G}_j^T} \tag{5.2}$$

Here, $|x|^2 \equiv x^T x$ is the power, the square of the amplitude. The expectation operator has been introduced in the denominator since it is the ratio of the signal power to the average noise power that is being maximized. This is the origin of the sample error covariance $C_d$.

With some factoring, the signal-to-noise ratio can be rewritten as

$$\text{SNR} = \frac{|(C_d^{1/2}\tilde{G}_j^T)^T (C_d^{-1/2} Gm)|^2}{(C_d^{1/2}\tilde{G}_j^T)^T (C_d^{1/2}\tilde{G}_j^T)} \tag{5.3}$$

The reason for doing so is that the expression is now in a form to which the Schwarz inequality can be applied:

$$(a^T b)^2 \leq (a^T a)(b^T b) \tag{5.4}$$

Now, the numerator and denominator of (5.3) can be identified with $(a^T b)^2$ and $(a^T a)$, respectively. This implies that signal-to-noise ratio may be no more than $(b^T b)$ or $m^T G^T C_d^{-1} Gm$. The equality is achieved when

$$C_d^{1/2}\tilde{G}_j^T \propto C_d^{-1/2} Gm \tag{5.5}$$

$$\tilde{G}_j^T = \alpha C_d^{-1} Gm \tag{5.6}$$

Which is the matched-filter theorem. It says that the signal-to-noise ratio is maximized by a filter constructed from the anticipated data $Gm$. Adjusting the proportionality constant is a matter of convention.

Of course $m$ is unknown. The approach is to set the rows of the filter so as to maximize the signal-to-noise ratio in each element of $m^{\text{est}}$ given the condition that the corresponding element of $m$ alone is nonzero. The filter that does this is given by

$$\tilde{G}^T = C_d^{-1} G \tag{5.7}$$

which has the function of an inverse operator.

Recalling the discussion in chapter 4, the two metrics for evaluating the efficacy of an inverse method involve perturbation and bias, respectively. The former considers distortions caused by sample errors, and the latter distortions due to the internal construction of the method. Matched filtering is optimal in minimizing perturbation. It may or may not be optimal in minimizing bias. The instruments for gauging bias are the model and data resolution matrices:

$$R_m \equiv \tilde{G}G \tag{5.8}$$

$$R_d \equiv G\tilde{G} \tag{5.9}$$

Ideally, both should be the identity, although this may be impractical in practice due to other demands on $G$. The design of radar experiments often hinges on minimizing the spread of $R_m$ and $R_d$.

The matched filter can be likened to the method of weighted damped least squares which was treated in a Bayesian sense in chapter 3. Comparing (5.7) with (3.24) shows that the two methods are the same when

$$G^T C_d^{-1} G + C_m^{-1} = I \tag{5.10}$$

In other words, the matched filter solution is the weighted damped least squares solution when the transitional probability is given by (3.21) and the prior probability is given by (3.22) with $C_m$ satisfying (5.10). In that case, the posterior probability is given by (3.25) with $\tilde{C}_m$ being the identity. This is an interesting connection with important overtones that is seldom considered. While matched filtering is usually performed outside the context of linear inverse problems, it actually fits neatly within it. While matched filtering is often considered to be optimal, it actually lies on a continuum of methods and may or may not be the best approach for a given problem.

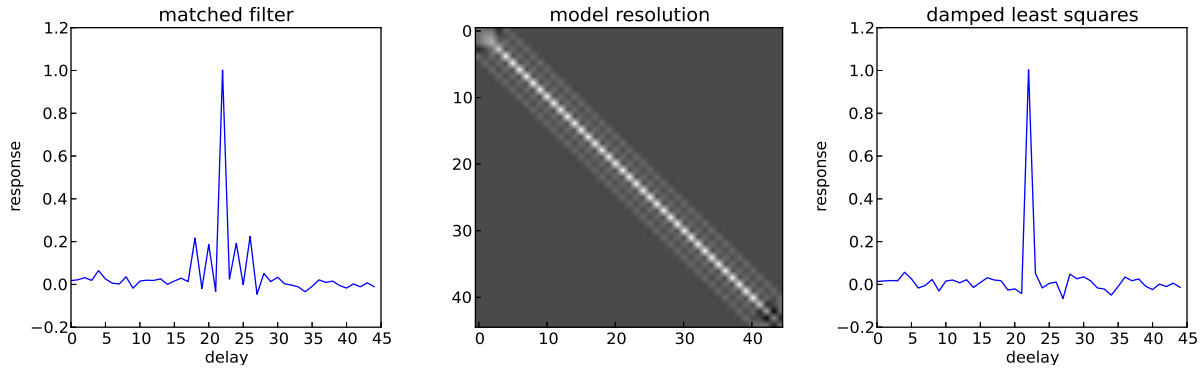


Figure 5.1: Radar pulse processing example. The left panel shows the results of matched filtering, and the right panel the results of damped least squares analysis. The panel in the center is a grayscale representation of $R_m$ which is band-diagonal.

## 5.2 Example: Barker-coded radar pulse

An simple example problem is illustrated in Figure 5.1. We consider a radar waveform which is modulated by a binary phase code such that its amplitude varies in time according to the code (1,1,1,-1,1). The corresponding matrix $G$ is in Toeplitz form with its first row containing the code followed by zeros and its first column containing 1 followed by zeros. Each component in the model, $m_j$, will produce in the data $d$ a copy of the code delayed in time by $j$. The data also contain sample noise. The object of the filter is to deconvolve the code from the data and return a replica of $m$ free of distortion due to perturbation and, hopefully, bias.

The leftmost panel of Figure 5.1 shows the results of matched filtering for a model $m$ with a single target at mid range. The main peak represents the target. Noise is present at closer and farther range, but matched filtering has ensured that the noise is minimal. However, four distinct peaks appear adjacent to the main target. These "range sidelobes" represent bias (or "clutter" in the radar lexicon) and arise from the finite spread of $R_m$, which is plotted in the center panel of Figure 5.1. In fact, the code, an example of a Barker code, was chosen to limit the amplitude of the sidelobes to $1/k$ where $k$ is the length in bauds of the code. Many classes of codes with reasonable resolution matrices have been developed. So-called "perfect codes" have no spread. There are no perfect binary phase codes, however.

The rightmost panel in Figure 5.1 meanwhile shows the results of damped least squares inversion. The matrix $G^T G$ is singular, and so a small amount of damping is required. The method has recovered the central peak indicative of a lone target. This time, there are no sidelobes, no clutter. The noise level of the recovered signal is higher by definition than in the matched filter case. It could be reduced with the introduction of greater damping but only at the expense of a broadening of the main peak which would be another kind of bias not present in the matched filter recovery.

## 5.3 Method of Backus and Gilbert

The method of Backus and Gilbert is a hybrid method in that it relates discrete data to a continuous model function. The method is rooted in basic functional theory. Consider the following direct model:

$$d_i = \int_a^b G_i(x) m(x)\, dx + e_i \tag{5.11}$$

$$d = \int_a^b G(x) m(x)\, dx + e \tag{5.12}$$

where $d_i$ are the elements of a data vector, $m(x)$ is a continuous model function, and $G_i(x)$ are a family of transfer functions, one for each datum. The $i$ subscripts can be replaced with the vector notation. The particular functions $G_i$ could differ dramatically if the $d_i$ represent different types of data entirely or could differ just slightly if, for example,

the $d_i$ are the same type of data merely sampled at different places or times. In any event, each datum is a expressed as a moment of the model function.

In the method of Backus and Gilbert, the model estimate at each point $\hat{x}$ is expressed as a moment or linear combination of the data:

$$m^{\text{est}}(\hat{x}) = c^T d \tag{5.13}$$

where $c(\hat{x})$ is a column vector to be determined for every $\hat{x}$. (The $\hat{x}$ dependence will be understood to be present in what follows if not explicitly written.) Substituting this into the direct problem yields:

$$\begin{aligned} m^{\text{est}}(\hat{x}) &= c^T \int_a^b G(x) m(x)\, dx && (5.14)\\ &= \int_a^b c^T G(x) m(x)\, dx && (5.15)\\ &= \int_a^b K(x,\hat{x}) m(x)\, dx && (5.16) \end{aligned}$$

where $K(x,\hat{x}) \equiv c^T G(x)$ is an averaging kernel and where $G(x)$ is a column vector formed by evaluating the functions $G_i(x)$ at the coordinate $x$.

Ideally, the averaging kernel should be made to approach a Dirac delta function, $K(x,\hat{x}) \approx \delta(x-\hat{x})$ since this would optimize the model resolution and provide a model estimate that is an exact inverse for the problem. In practice, equivalence will not be possible because of the limited degrees of freedom available in the weight vector $c$. The objective then becomes finding the weight vector that optimizes the averaging kernel. A desirable averaging kernel is one that approximates a Dirac delta in being narrow while maintaining unity area. In this regard, the method of Backus and Gilbert is merely an instance of a linear constraint minimum variance (LCMV) method.

The unity area constraint can be expressed in the following way.

$$\begin{aligned} \int_a^b K(x,\hat{x})\, dx &= \int_a^b c^T G(x)\, dx && (5.17)\\ &= c^T \int_a^b G(x)\, dx && (5.18)\\ &= c^T q && (5.19)\\ &= 1 && (5.20) \end{aligned}$$

where the column vector $q$ has as its components the 0th moments of the various $G_i(x)$. While there are a number of different ways to quantify the width $w$ of a function, the most common metric is the 2nd moment:

$$\begin{aligned} w &= \int_a^b K^2(x,\hat{x})(x-\hat{x})^2\, dx && (5.21)\\ &= c^T H c && (5.22)\\ H &= \int_a^b G(x) G^T(x)(x-\hat{x})^2\, dx && (5.23) \end{aligned}$$

where it should be noted that $H$ is symmetric. Finally, the method amounts to finding the coefficients $c$ that minimize the width of the averaging kernel while maintaining the unity constraint:

$$\begin{aligned} c &= \underset{c,\lambda}{\text{argmin}} \quad c^T H c + \lambda(c^T q - 1) && (5.24)\\ &= \frac{H^{-1} q}{q^T H^{-1} q} && (5.25) \end{aligned}$$

which must be evaluated at every coordinate $\hat{x}$ of interest. The model estimate at the given coordinate is then found through the application of (5.13). What the model estimate accomplishes is simply the minimization of the model resolution spread. If required, regularization can be included through additional penalties and constraints. The concepts and operations involved should be well familiar to the reader by now.

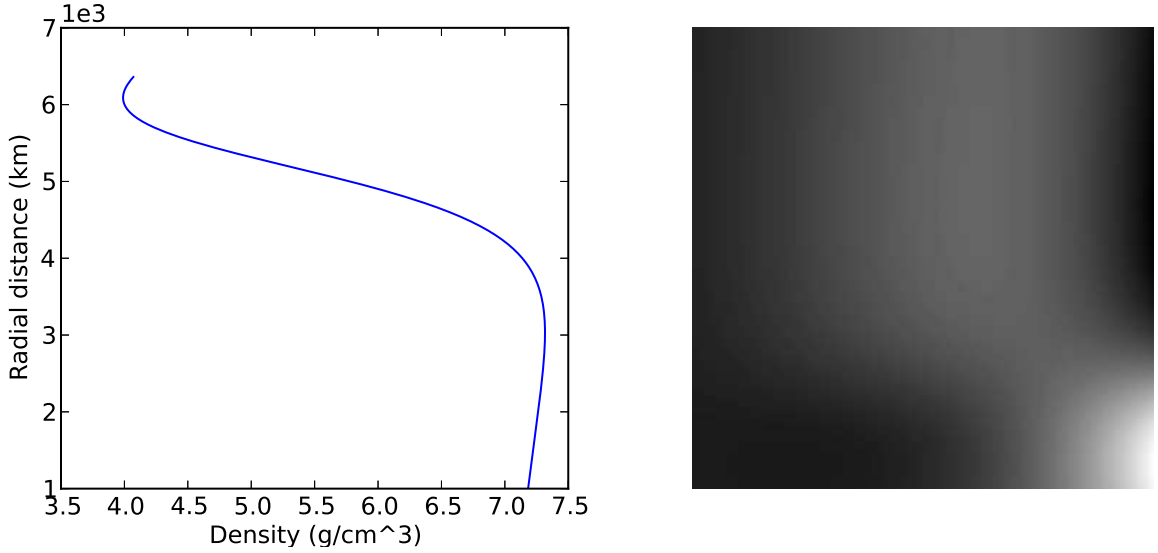


Figure 5.2: Backus-Gilbert estimate of the Earth's density versus radial distance $\hat{r}$. The panel on the right shows the kernel $K(r,\hat{r})$ which ideally approaches a Dirac delta function.

## 5.4 Example: Earth's interior

Backus and Gilbert famously used their method to infer the density profile of the Earth's interior, $\rho(r)$, from just two pieces of information obtained from astronomical observations – the Earth's mass total $M$ and momentum of inertia $I$. The corresponding moment equations are:

$$M = \int_0^{R_e} \underbrace{(4\pi r^2)}_{G_1(r)} \rho(r)\, dr \tag{5.26}$$

$$= 5.972 \times 10^{24}\,\mathrm{kg} \tag{5.27}$$

$$I = \int_0^{R_e} \underbrace{\left(\frac{8}{3}\pi r^4\right)}_{G_2(r)} \rho(r)\, dr \tag{5.28}$$

$$= 8.034 \times 10^{37}\,\mathrm{kg\,m^2} \tag{5.29}$$

Armed with these moment equations and the two corresponding data, it is a straightforward matter to compute the vector $q$, the matrix $H$, the weight vector $c$, and finally the density estimate $c^T d$ for any value of $\hat{r}$, the radial distance from the center of the Earth.

Figure 5.2 shows the mass density profile of the interior of the Earth estimated using the method of Backus and Gilbert. At shallow depths less than about 2000 km, the estimate has the density increasing from about 4 to 6 g/cm$^3$ with increasing depth. This estimate is in reasonable agreement with the known density profile in the Earth's mantle. At greater depths, however, the estimate has the density peaking between 7–7.5 g/cm$^3$ and then decreasing slightly with depth deeper in the interior. This is a poor estimate of the density in the inner and outer core which continues increasing with depth to a value approaching 12 g/cm$^3$ at the Earth's center. Discrete jumps in density at the boundaries between the inner and outer core and between the core and mantle are of course no where to be seen in the estimate which is not informed about stratigraphy.

The right panel of Figure 5.2 shows the kernel function $k(r,\hat{r})$ for the Backus-Gilbert estimate. Ideally, this function, the model resolution function, should approach the Dirac delta function. In fact, it only resembles Dirac's delta in the lower-right corner which is indicative of the model resolution at large radial distances. At decreasing radial

distance (increasing depth), the kernel is quite diffuse, indicating that the estimate is only weakly contingent on the true model curve. It is therefore no wonder that the method performs poorly at depth.

## 5.5 Radon transform and its inverse

CAT scans and the Radon transform were discussed back in Chapter 1 as an early example of a fully continuous inverse problem. To recap (and using slightly different notation), the Radon transform relates the absorption coefficient within a body $F(x,y)$ to the X-ray measurements on the screen:

$$R_\theta(s) = \int\int F(x,y)\delta(x\cos\theta + y\sin\theta - s)dxdy \tag{5.30}$$

where the integration is over the entire area containing the body but where the Dirac delta function picks out only the contributions to the absorption along the line that intersects the screen at the point $s$ for a given orientation $\theta$.

Plots of the quantity $R_\theta(s)$ are called sinograms because a point target in $F(x,y)$ produces a single sine wave in a plot of $R$ versus $\theta$. Suppose that $F(x,y) = \delta(x-x_\circ, y-y_\circ)$. In that case, $R_\theta(s) = A\sin(\alpha+\theta)$, where $x_\circ = A\sin\alpha$ and $y_\circ = A\cos\alpha$. The amplitude and phase of the sine wave therefore reflect the radial and angular positions of the point target, respectively. A filled volume consequently produces a conglomeration of sine waves of different amplitude and phase in the final sinogram. Inferring the shape of the absorbing body from the sinogram 'by eye' would pose quite a challenge, however.

If we knew $F(x,y)$, we could easily calculate $R_\theta(s)$. How do we estimate the former knowing the latter? What is needed is another transformation that converts $R_\theta(s)$ back to $F(x,y)$. To find one, we begin by developing the so-called Fourier slice theorem. Consider the two-dimensional Fourier transform pair:

$$\hat{f}(\mathbf{w}) = \int\int F(\mathbf{x})e^{-i\mathbf{x}\cdot\mathbf{w}}d\mathbf{x} \tag{5.31}$$

$$F(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^2 \int\int \hat{f}(\mathbf{w})e^{i\mathbf{x}\cdot\mathbf{w}}d\mathbf{w} \tag{5.32}$$

Next, consider the one-dimensional Fourier transform of the sinogram:

$$\hat{R}_\theta(w) = \int R_\theta(s)e^{-iws}ds \tag{5.33}$$

$$= \int\int F(x,y)e^{-\overbrace{iw(x\cos\theta + y\sin\theta)}^{w\hat{n}\cdot\mathbf{x}}}dxdy \tag{5.34}$$

$$= \hat{f}(w\hat{n}) \tag{5.35}$$

where $w$ plays the role of a spatial frequency. Equation (5.33) refers to the operation of taking a one-dimensional Fourier transform of the sinogram at some particular angle $\theta$. Performing the $ds$ integral lead directly to (5.34). In that equation, the exponential has the form $-iw\hat{n}\cdot\mathbf{x}$, where $\hat{n}$ is the normal vector in Figure 1.4. According to (5.31), this equation is then the two-dimensional Fourier transform to the space $w\hat{n}$, as shown in (5.35). The consequence of all this is that the Fourier transform of a sinogram measured at some angle $\theta$ is a slice through the two-dimensional Fourier transform of $F(x,y)$ taken along a cut in the normal ($\hat{n}$) direction.

The Fourier slice theorem suggests a means of inverting sinograms. We could measure them along a number of different angles, take the required 1-D Fourier transforms, assemble the appropriate cuts, and then calculate the inverse 2-D Fourier transform of $\hat{f}(\mathbf{w})$ according to (5.32) to arrive at $F(x,y)$. While this is certainly possible, the cylindrical symmetry of the problem prevents the exploitation of a Fast Fourier Transform algorithm, and it is not apparent how to perform the operations involved efficiently.

The radon transform $R$ can be viewed as an operator that maps from $F(x,y)$ to a $RF(x,y) = g_\theta(s)$. There exists an adjoint operator $R^\#$ that satisfies $(RF,g) = (F,R^\#g)$. This is called the back-projection operator. Its definition is

$$R^\# g = \int_0^{2\pi} g(\theta, \hat{n}(\theta)\cdot\mathbf{x})d\theta \tag{5.36}$$

where $g$ now represents the sinogram which is a function of $s = \hat{n}(\theta) \cdot \mathbf{x}$. Formally, (5.36) is $R^{\#}g = R^{\#}RF$. If the back projection operator is constructed such that $R^{\#}RF = F$, then it will transform the sinogram back to the desired absorption function, and the inversion problem is solved. Strict backprojection is essentially analogous to matched filtering.

In fact, (5.36) yields a recognizable but imperfect reproduction of $F(x,y)$ only. In order to improve the inversion, the sinograms must first be filtered. The appropriate filter is one that emphasizes the high-frequency components over low frequencies in $w$-space:

$$\hat{H} = |w| \tag{5.37}$$

The necessary filter is simply a ramp function. Finally, the desired absorption function if estimated as

$$F(\mathbf{x}) = \frac{1}{4\pi} R^{\#} H g \tag{5.38}$$

showing how the sinograms can be transformed using a simple linear transformation that, in this case, differs from the transformation that produced them. The prescription is to filter all the sinograms, one angle at a time, and then integrate over the entire collection using the back projection operator in (5.36). All that is involved numerically is a number of discrete Fourier transforms and other one-dimensional integrals.

That (5.38) actually returns the underlying absorption function $F(x,y)$ is easily shown. We can express (5.38) as:

$$\begin{aligned}
F(\mathbf{x}) &\stackrel{?}{=} \frac{1}{4\pi} \int_0^{2\pi} H g(\theta, s) d\theta \\
&\stackrel{?}{=} \frac{1}{4\pi} \frac{1}{2\pi} \int_0^{2\pi} \int_{-\infty}^{\infty} |w| \hat{R}_\theta(w) e^{iws} dw d\theta \\
&\stackrel{?}{=} \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi} \int_0^{\infty} w \hat{R}_\theta(w) e^{iw(x\cos\theta + y\sin\theta)} dw d\theta \\
&= \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi} \int_0^{\infty} w \hat{f}(w\hat{n}_\theta) e^{i\mathbf{w}\cdot x} dw d\theta
\end{aligned}$$

where the last line is just the definition of the inverse Fourier transform in two dimensions and in cylindrical coordinates. Evidently, the ramp function serves as the differential component $w$ that appears in cylindrical coordinates. Functionally, it serves to emphasize high-frequency components that are otherwise under-emphasized by sampling a two-dimensional function at fixed angular increments. While the effect of the ramp filter function is to preferentially amplify noise, the integral implicit in the sinogram tends to attenuate noise, and the overall CAT scan analysis turns out to be well conditioned.

Equation (5.38) represents a linear integral transformation that reverses the linear transformation in (5.30) with the objective of recovering the most accurate representation of the absorption function $F(x,y)$ possible rather than maximizing the signal-to-noise ratio as in matched filtering. The form of (5.38) essentially compensates for the effects of the cylindrical geometry of the CAT scan. Similar transformations have been calculated for other problems in remote sensing, notably for applications in radio and radar imaging.

## 5.6 Example: CAT scan

An example of a CAT-scan inversion is given in Figure 5.3. Although the subject is continuous inverse theory, solution by digital computer necessitates the discretization of $F$ and its inverse. The original image is a black-and-white rendition of Jolly Roger with 200x200 resolution. From this, a sinogram is computed on a grid with 120 discrete angles $\theta$ and 128 discrete screen positions $s$. The panels on the left of the figure show the corresponding sinograms. The sinogram in the top row represents the noise-free case whereas normally-distributed independent noise has been added to the sinogram in the bottom row.

The panels in the middle row of Figure 5.3 show the results of inversion by unfiltered back projection. The resolution of the recovered image here is 128x128. The main features of the image are visible but murky. Back

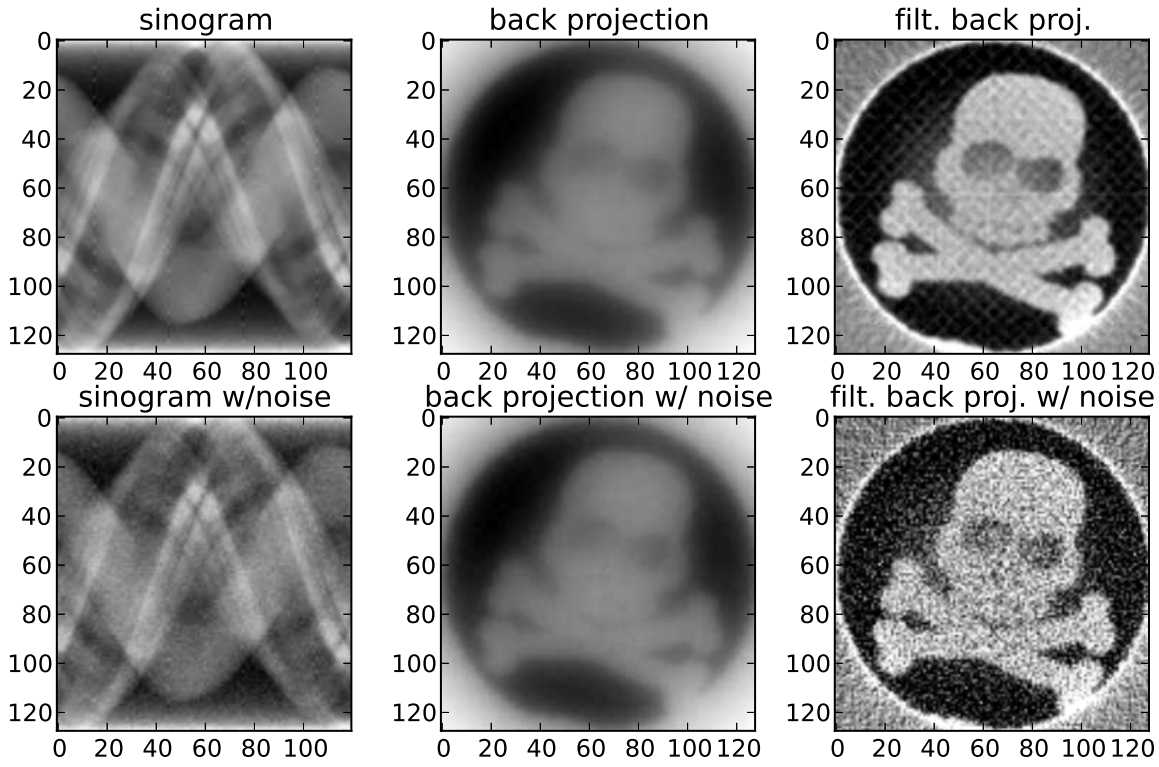


Figure 5.3: Numerical example of CAT-scan reconstruction. The leftmost panels show sinograms of an image. The center panels show the results of reconstruction by back projection. The rightmost panes show the results of reconstruction by filtered back projection. The top row is the noise-free case while the bottom row is the case where normally-distributed independent noise has been added to the sinogram.

projection is not the true inverse of the Radon transform, and the recovered images reflect a bias. Notice, however, that the effects of added noise are strongly suppressed. Back projection involves integration of the sinogram, and perturbation noise is suppressed according to the Riemann-Lebesque Lemma. Back projection is an extremely well-posed operation if not a particularly well resolved one.

In contrast, the panels in the right row of Figure 5.3 show the results of filtered back projection. The high-pass filtering has enhanced the detail in the recovered image compared to the unfiltered recovery. Filtered back projection is in fact the true inverse of the Radon transform and is expected to be free of bias. Any artifacts in the recovered image are a consequence of the discretization, which is rather coarse in this example. At the same time, filtering is akin to numerical differentiation and has the effect of restoring the noise in the sinogram to the recovered image, as can be seen in the bottom panel. Perturbation noise is clearly present here but has not been amplified, and the method is stable without any need for regularization.

## 5.7 Abel transform and its inverse

The introductory chapter of this text introduced the Abel transform and its inverse. The Abel transform is useful for problems involving measurements of path-integrated quantities in media which are either spherically or cylindrically symmetric. Applications include measurements of the Earth's atmosphere and ionosphere collected through radio propagation experiments. When the radio transmitter, receiver, or both are in space, the experiments are referred to as "radio occultations." Sources of radio signals can be natural (e.g. radio stars) or artificial (e.g. satellite beacons).

Several aspects of a radio signal propagating through the atmosphere are indicative of the properties of the intervening material including its amplitude, polarization, and spectrum. The most commonly observed characteristics, however, are the phase of the waves and the time it takes for signals on the waves to travel from the receiver to the transmitter. Both quantities are related to the path-integrated index of refraction of the medium.

The phase of a radio signal at a point in space and time can be expressed as

$$\phi = \omega\left(\int ds\,\frac{n}{c} - t\right) \tag{5.39}$$

where $\omega$=$2\pi f$ is the carrier frequency, $c$ is the speed of light in vacuum, $t$ is time, and $n$ is the index of refraction, which can vary spatially. Note that the ratio $c/n$ is the phase speed $v_p$ of the wave. The integral is over the ray path starting and ending at the transmit and receive points, respectively. Measuring phase unambiguously is complicated by the fact that the phase wraps periodically. The time rate of change of the phase is the Doppler shift which can be measured without ambiguity so long as the constraints of the Nyquist sampling theorem are satisfied.

The time it takes for a signal to propagate from the transmitter to the receiver, meanwhile, is given by

$$\tau = \int \frac{ds}{v_g} \tag{5.40}$$

where $v_g$ is the group speed of the signal. In simple non-dispersive media, the group speed and the phase speed are the same. In more complicated materials, they can be quite different.

In the case of radio waves propagating through the troposphere, the index of refraction increases slightly from unity (by a few hundred parts per million at most) proportionally with the neutral gas pressure, the water vapor pressure, and the content of liquid water. At most radio frequencies, the atmosphere is non-dispersive, and the phase speed and group speed are both given by $c/n$.

In the case of ionized gasses or plasmas, the index of refraction is given by $n = \sqrt{1-\omega_p^2/\omega^2}$ where $\omega_p$ the plasma frequency $\omega_p = 2\pi f_p$, a characteristic frequency of oscillations in a plasma. The plasma frequency is proportional to the square root of the electron number density. In MKS units, $f_p$ = $9N_e$ MHz. Remarkably, the group speed in this case is $v_g = [\partial(\omega n/c)/\partial\omega]^{-1} = cn$ such that the product of the phase speed and the group speed is $c^2$.

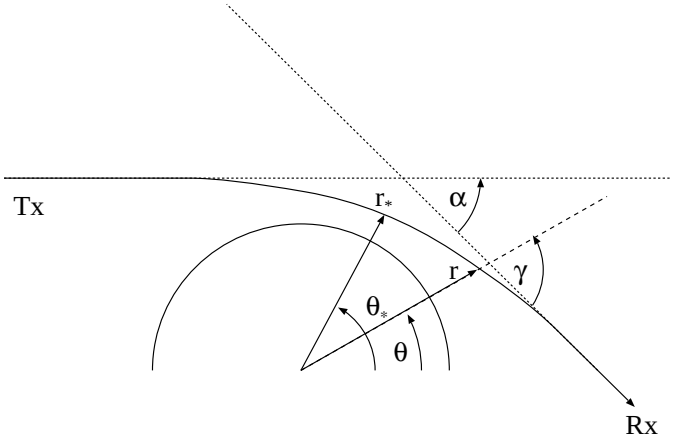


Figure 5.4: Illustration of a radio occultation experiment involving two low-earth-orbit (LEO) satellites.

In the limit that the radio frequency is much greater than the plasma frequency, the limit in which the expression given for $n$ above is strictly accurate, we have (MKS units):

$$\delta\phi = \frac{2\pi f}{c}\int ds\left(1-40.27\frac{N_e}{f^2}\right) \tag{5.41}$$

$$\tau = \int \frac{ds}{c}\left(1+40.27\frac{N_e}{f^2}\right) \tag{5.42}$$

where $\delta\phi$ is the difference in phase at either end of the path length at a common time. Interestingly, electron density content can be seen to decrease the phase offset or delay and increase the group delay. The former effect results from the increase in wavelength that occurs as the plasma frequency increases. The latter is required for propagation at subluminal group speeds.

Deviations from the phase and group delay in vacuum depend on the path integral of the electron number density. A total electron content or "TEC" unit is defined as $10^{16}$ m$^{-2}$ and is a convenient way to express the deviation. Note that the phase and group delays in vacuum can be measured using radio frequencies much greater than the plasma frequency for calibration.

So long as the propagation path between the transmitter and receiver are straight lines, the aforementioned expressions are suitable for inversion using the inverse Abel transform. In the ionosphere and at radio frequencies well above the peak ionospheric plasma frequency (or "critical frequency", which is seldom above about 10 MHz), the propagation paths can be taken to be straight lines. This is not true for radio paths passing through the troposphere, however, where refraction causes significant ray bending for radio waves in the VHF, UHF, and microwave bands. Inverting tropospheric radio occultation data therefore proceeds along somewhat different lines.

Refer to Figure 5.4 for an illustration of a radio occultation experiment involving considerable ray bending. "Tx" and "Rx" denote the transmission and reception points, respectively. The coordinate at some point along the path of the ray is $(r,\theta)$. (Without loss of generality, the problem can be considered in polar coordinates.) The star subscript denotes the path of closest approach such that $r_*$ is the impact parameter for the ray.

According to Fermat's principle, the ray followed by the wave is the one for which the optical path length (the path length measured in wavelengths) is a minimum:

$$\Phi \equiv \int n\,ds \tag{5.43}$$

In polar coordinates, $ds^2 = dr^2 + r^2 d\theta^2$, and so

$$\Phi(r,\theta,\theta') \equiv \int n\sqrt{1+(r\theta')^2}\,dr \tag{5.44}$$

where $\theta' \equiv d\theta(r)/dr$ on the correct path. This functional can be minimized using the Euler-Lagrange equations, yielding:

$$\frac{d}{dr}\frac{\partial}{\partial\theta'}() - \frac{\partial}{\partial\theta}() = 0 \tag{5.45}$$

where the brackets contain the integrand of (5.44). Since this term is not an explicit function of $\theta$, we can write

$$\frac{\partial}{\partial\theta'}\left(n\sqrt{1+(r\theta')^2}\right) = \text{const.} = a \tag{5.46}$$

$$\frac{nr^2\theta'}{\sqrt{1+(r\theta')^2}} = a \tag{5.47}$$

$$nr\sin\gamma = a \tag{5.48}$$

$$= n(r_*)r_* \tag{5.49}$$

such that different values of the constant $a$ imply different ray paths. Now $\gamma$ is the angle between the ray path and the radial direction, and the formula $nr\sin\gamma = \text{const}$ is a conservation equation known as Bouger's law, a generalization of Snell's law for radially-symmetric media. The constant can be set for a given ray with an impact parameter $r_*$ by noting that the angle $\gamma$ is a right angle at the distance of closest approach. Reorganizing the above set of equations produces an ODE that can be used to reconstruct the ray path for given impact parameter and index of refraction profile:

$$\frac{d\theta}{dr} = \pm\frac{a}{r}\frac{1}{\sqrt{n^2r^2-a^2}} \tag{5.50}$$

Here, the plus-minus sign reflects the turning point that occurs at $\theta_*$.

An equation to recover the index of refraction profile from experimental data can be derived from Bouger's law:

$$0 = \frac{d}{dr}(nr\sin\gamma) \tag{5.51}$$

$$= \frac{dn}{dr}r\sin\gamma + n\sin\gamma + nr\cos\gamma\frac{d\gamma}{dr} \tag{5.52}$$

$$= \frac{n'}{n}a + \frac{a}{r} + \sqrt{n^2r^2(1-\sin^2\gamma)}\left[\frac{d\theta}{dr}+\frac{d\alpha}{dr}\right] \tag{5.53}$$

where the last line results from substituting Bouger's law and making use of the fact that $\gamma = \theta + \alpha$, where $\alpha$ is the bending angle of the ray. As above, the prime denotes differentiation with respect to $r$. Some final rearranging results in an equation for the bending angle:

$$\frac{d\alpha}{dr} = -a\frac{n'}{n}\frac{1}{\sqrt{n^2r^2-a^2}} \tag{5.54}$$

$$\alpha(a) = 2\int_{r_*}^{\infty} d\alpha \tag{5.55}$$

$$= -2a\int_{r_*}^{\infty}\frac{1}{\sqrt{n^2r^2-a^2}}\frac{d}{dr}\ln n\,dr \tag{5.56}$$

$$= -2a\int_{a}^{\infty}\frac{1}{\sqrt{n^2r^2-a^2}}\frac{d}{dnr}\ln n\,dnr \tag{5.57}$$

with a change of variables $y = nr$, this transformation becomes:

$$\frac{\alpha(a)}{2\pi a} = -\frac{1}{\pi}\int_a^{\infty}\frac{d\ln n}{dy}\frac{dy}{\sqrt{y^2-a^2}} \tag{5.58}$$

After some effort, we have come to the amazing conclusion that the bending angle of a ray is related to the index of refraction profile through an inverse Abel transform. This means that inversion of the data leading to estimates of $n(r)$ can proceed using the direct Abel transform. The data underlying the measurement are measurements of the bending angle versus $a$. Both the bending angle and the $a$ parameter of the ray connecting the transmitter and the receiver can be inferred from the Doppler shift of the beacon single with the reapplication of Bouger's law.

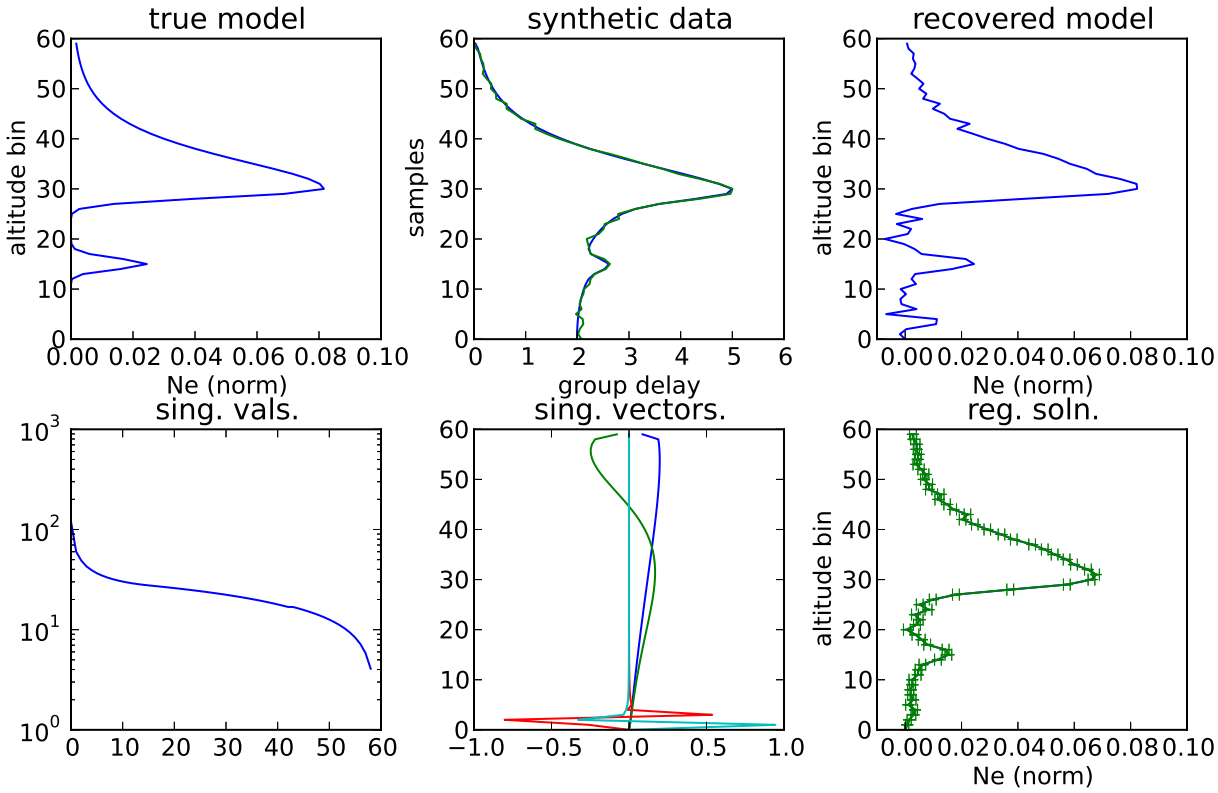


Figure 5.5: Demonstration of the properties of the inverse Abel transform. The profile in this case is meant to be evocative of layers of ionization in the Earth's upper atmosphere. The top row of the figure shows the true model (left panel), synthetic data computed from the true model using the Abel transform (center panel), and the recovered model for the case with observation noise. The bottom row shows the ranked singular values (left panel), a few representative singular vectors (center panel), and the recovered model using Tikhonov regularization (right panel).

## 5.8 Example: ionospheric radio occultation

Intuition suggests that the direct abel transform should be well posed while its inverse should be poorly posed. We investigate this behavior by considering a true model profile which is meant to evocative of layers of ionization in the Earth's upper atmosphere. For this test case, we consider discretized model and data profiles, each taking the form of column vectors with 60 elements. (The matter of discretizing continuous data will be considered in the next chapter.)

The ionospheric layers problem is illustrated in Figure 5.5. Synthetic data are produced from the true model using a discrete form of the Abel transform. Superimposed on the synthetic data profile in Figure 5.4 is another profile with normally-distributed independent observation noise added.

Inversion of the synthetic data could be accomplished with the application of a discrete form of the inverse Abel transform, with some variant of linear least squares, or using singular value decomposition. All of these methods are essentially equivalent for the discrete problem. We proceed using singular value decomposition since it gives the greatest insight into the stability of the problem.

The SVD approach shows that one of the singular values is much smaller than the others. With that singular value and the associated singular vectors discarded, the pseudoinverse yields the recovered model profile shown in the top right panel of Figure 5.5. Clearly, the noise (which is barely visible in the synthetic data) imposes significant distortion. The problem is most acute near the bottom of the profile which oscillates severely and enters negative

territory — something which is not physically permissible.

The lower-left panel of Figure 5.5 shows the ranked singular values for the Abel transform. The condition number for this problem is evidently not very large, and yet the Abel transform possesses a unique pathology that makes it poorly suited for the application pursued here. The center panel of the bottom row of Figure 5.5 shows two singular vectors corresponding to the largest singular values (blue and green) along with two corresponding to the smallest (red and cyan). Whereas the former two curves are gradually varying and have most of their support at large radial distances, the latter two are highly oscillatory and have most of their support at small radial distances. This means that noisy data will have a strong tendency to produce spurious ringing at low altitudes in the inverted profiles, exactly the behavior found above. This is highly problematic in practice, since the ringing could easily be mistaken for actual layers in nature.

The panel in the bottom-right corner of Figure 5.5 shows the results of profile inversion using SVD with Tikhonov regularization. Error bars have been added for reference. Regularization has clearly in this case attenuated the ringing and suppressed the appearance of spurious layers. Most (but not all) of the negative-going excursions of the curve have been removed. Regularization has come at a significant cost, however, as the details in the peaks in the true model have also been lost. Both the major and the minor peaks are broader and shallower in the recovered model than in the original. Tikhonov regularization is not edge preserving. Later in the text, we will discuss inversion strategies that suppress spurious ringing without discarding physical structure.

## 5.9 References

## 5.10 Problems

# Chapter 6

# Discretization and sampling

The previous chapter considered continuous inverse problems and also semi-continuous problems where the model was continuous but the data were discrete. (This was the Backus and Gilbert problem.) The reverse situation could also be true. In every case discussed in the chapter, however, examples were constructed around discretized versions of the given problem. This is necessary in the age of digital computing. Were analog computers still popular, this text might well emphasize continuous inverse problems while giving short shrift to discrete ones!

In fact, fully-continuous inverse problems are an abstraction in modern science where virtually all data are acquired as a set of digitized samples and where models are often expected to be specified on a discrete grid. It is hard to conceive of what a truly continuous dataset would even look like. This chapter therefore considers methods of discretizing inverse problems which have been formulated continuously but which must be solved discretely. We have already made use of some of these methods without realizing it.

## 6.1 Collocation, representers, and expansion bases

Continuing with the theme of linear inverse problems, we recall the Fredholm integral equation of the first kind:

$$d(s) = \int_a^b g(s,x)m(x)\,dx \tag{6.1}$$

In the event that the data are only know at certain discrete points, this becomes

$$\begin{aligned} d_i &= d(s_i) = \int_a^b g(s_i,x)m(x)\,dx, \quad i = 1,n && (6.2)\\ &= \int_a^b g_i(x)m(x)\,dx && (6.3)\\ &= \langle g_i|m\rangle && (6.4) \end{aligned}$$

In this representation, the functions $g_i(x)$ are known as representers. Different representers have different mathematical properties, and there may be ways of combining the representers and the model to exploit them, as will be discussed below. In any case, the problem now becomes one of carrying out the integration numerically by approximating the model and the representers as discrete quantities. This is the problem of numerical quadrature.

The simplest approach to the problem is break the domain of integration into $m$ uniform bins of width $\Delta x = (b-a)/m$ and to evaluate the model and the representers at the midpoints of these domains, i.e.,

$$\begin{aligned} d_i &\approx \sum_{j=1}^{m} g_i(x_j)m(x_j)\Delta x && (6.5)\\ x_j &= a + \frac{\Delta x}{2} + (j-1)\Delta x && (6.6) \end{aligned}$$

If we compute the matrix with the elements $G_{ij} = g_i(x_j)\Delta x$ and the vector with the elements $m_j = m(x_j)$, then we are left with the familiar discrete inverse problem $d = Gm$. This is the method of simple collocation which has been used extensively throughout this text already. Readers may recognize that this is identical to numerical integration by a rectangular rule. Improvement can be realized by implementing a trapezoidal rule, Simpson's rule, etc. A general treatment of numerical quadrature follows in the next section of the text.

An alternative to simple collocation is the (approximate) expansion of the model function in terms of a discrete set of functions. One set which is convenient because it is already on-hand is the representers themselves, i.e.,

$$m(x) = \sum_j \alpha_j g_j(x) \tag{6.7}$$

so that

$$\begin{aligned} d(s_i) &= \sum_j \alpha_j \int_a^b g_j(x) g_i(x)\, dx & (6.8)\\ d_i &= \sum_j \alpha_j \Gamma_{ij} & (6.9)\\ d &= \Gamma\alpha & (6.10) \end{aligned}$$

where the symmetric matrix $\Gamma$ has the elements $\Gamma_{ij} = \langle g_i | g_j \rangle$. The model now is specified by the weight vector $\alpha$. The elements of the new system matrix $\Gamma$ may be calculable exactly, or else numerical integration may be required a priori. It can be shown that $\Gamma$ is nonsingular when the representers are linearly independent. The matrix tends to be ill conditioned when large, however, suggesting a tension between stability and accuracy.

Another possibility is the expansion of the model in terms of convenient set of linearly-independent or orthogonal basis functions $h_j(x)$. In this case,

$$m(x) = \sum_j \alpha_j h_j(x) \tag{6.11}$$

so that

$$\begin{aligned} d(s_i) &= \sum_j \alpha_j \int_a^b h_j(x) g_i(x)\, dx & (6.12)\\ d_i &= \sum_j \alpha_j C_{ij} & (6.13)\\ d &= G\alpha & (6.14) \end{aligned}$$

where the system matrix $G$ now has the elements $G_{ij} = \langle g_i | h_j \rangle$. The different elements of $G$ are different moments of the representers with respect to the basis functions. These may be straightforward to calculate or compute depending on the functional form. In this case, the non-singularity of $G$ is obviously guaranteed although the stability of large systems is still a concern. Furthermore, if the basis functions are orthogonal, minimizing the the L2 norm of the model length is equivalent to minimizing the L2 norm of the weight vector $\alpha$.

## 6.2 Numerical quadrature

Having reduced the discretization of continuous inverse problems to a matter of numerical integration, we turn now to basic methods in numerical quadrature. This topic is broad and predates the invention of practical computers. A few basic ideas are sufficient for proceeding with the remaining topics in the text, however.

Figure 6.1 shows a function $f(x)$ to be integrated numerically over a domain in $x$. We initially consider the domain to be broken into uniform intervals of width $h$. Two elementary formulas for estimating the definite integral of the function are:

$$\begin{aligned} \int_{x_1}^{x_2} f(x)\, dx &= h\left[\frac{1}{2}f_1 + \frac{1}{2}f_2\right] + O(h^3 f'') & (6.15)\\ \int_{x_1}^{x_3} f(x)\, dx &= h\left[\frac{1}{3}f_1 + \frac{4}{3}f_2 + \frac{1}{3}f_3\right] + O(h^5 f'''') & (6.16) \end{aligned}$$

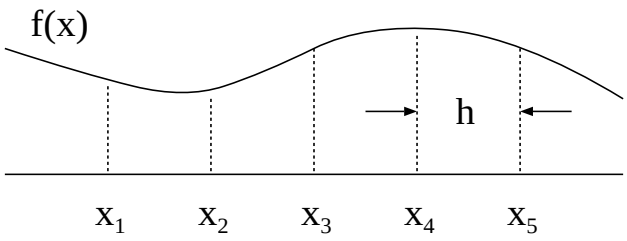


Figure 6.1: Numerical quadrature illustration.

where $f_j$ implies $f(x_j)$. The accuracy limit means that the integral estimate differs from the true value by a knowable constant times the given power of $h$ times the given derivative of the function evaluated somewhere in the given interval.

Eq. (6.15) above is the 2-point trapezoidal rule which comes from approximating the area between $x_1$ and $x_2$ as a trapezoid. The formula is exact for $f$ a polynomial of degree 1. Likewise, (6.16) is Simpson's 3-point rule which can be derived by approximating the curve between the endpoints as a parabola which passes through the endpoints and the midpoint of the curve. This formula turns out fortuitously to be exact for $f$ a polynomial of degree 3. The 4-point formula found in the same way (Simpson's 3/8 rule) is also exact for $f$ a polynomial of degree 3, whereas the 5-point formula (Bode's rule) is fortuitously exact for $f$ a polynomial of degree 5.. In general, an $n$-point rule found in this way will be exact for $f$ a polynomial of degree $n-1$. The weights for the $n$-point formulas $w_i$ can be found by solving the coupled linear system of $n$ equations in $n$ unknowns:

$$\int_a^b f_\circ(x)\,dx = \sum_{i=1}^{n} w_i f_\circ(x_i) \tag{6.17}$$

$$\vdots = \vdots$$

$$\int_a^b f_{n-1}(x)\,dx = \sum_{i=1}^{n} w_i f_{n-1}(x_i) \tag{6.18}$$

where $f_i(x)$ is a monomial of degree $i$. Fortunately this procedure is seldom necessary, as will be discussed below.

The formulas above are closed formulas because the boundaries of integration are coincident with boundary points. It is also possible to derive open and semi-open formulas following the same approach. Below are the 2- and 3-point semi-open formulas corresponding to the closed formulas above:

$$\int_{x_\circ}^{x_2} f(x)\,dx = h\left[\frac{3}{2}f_1 - \frac{1}{2}f_2\right] + O(h^3 f'') \tag{6.19}$$

$$\int_{x_\circ}^{x_3} f(x)\,dx = h\left[\frac{23}{12}f_1 - \frac{16}{12}f_2 + \frac{5}{12}f_3\right] + O(h^4 f''') \tag{6.20}$$

Next, so-called extended or composite formulas combine the elementary formulas above so as to cover wide domains of integration. The extended trapezoidal rule (closed) is found simply by adding the results from adjoining intervals

$$\int_{x_1}^{x_n} f(x)\,dx = h\left[\frac{1}{2}f_1 + f_2 + \cdots + f_{n-1} + \frac{1}{2}f_n\right] + O\left(\frac{(b-a)^3}{n^2}f''\right) \tag{6.21}$$

where the accuracy has been stated in terms of the number of points $n$ instead of the interval width $h$, highlighting the improvement achieved by increasing the former. Adding the results of Simpson's rule for adjoining pairs of intervals gives:

$$\int_{x_1}^{x_n} f(x)\,dx = h\left[\frac{1}{3}f_1 + \frac{4}{3}f_2 + \frac{2}{3}f_3 + \frac{4}{3}f_4 + \cdots + \frac{2}{3}f_{n-2} + \frac{4}{3}f_{n-1} + \frac{1}{3}f_n\right] + O\left(\frac{(b-a)^5}{n^4}f''''\right) \tag{6.22}$$

where the alternating weights arise from the piecewise treatment of the domain of integration. Note that there is an extended formula with errors of order $n^{-3}$ that has been omitted here. Open and semi-open extended formulas can be similarly formulated but are also omitted.

### 6.2.1 Romberg integration

Fortunately, generating functions for the various rules of integration exist which help make the coding of different quadrature schemes expeditious. Using the method of Romberg integration, trapezoidal integration is performed repeatedly, with the interval $h$ halved each time and the number of sample points doubled. The result from the previous iteration $I_n$ is subtracted from the results of the current iteration $I_{2n}$ to reduce the errors.

After two iterations, the first being based on the $n$-point extended trapezoidal rule and the second on the $2n$-point rule, the estimate of the integral is

$$I = \frac{4}{3} I_{2n} - \frac{1}{3} I_n + O(n^{-4}) \tag{6.23}$$

where the particular weighting makes the leading error terms of order $n^{-2}$ cancel identically. Moreover, it turns out that all of the error terms in the extended trapezoidal rule have even negative powers of $n$, so the difference term is actually accurate to order $n^{-4}$. This is a special feature of the trapezoidal rule not generally exhibited by higher-order rules. The behavior of the new composite rule is identical to that of the extended Simpson's rule. In fact, the two rules are one in the same.

Successive iteration yield rules of higher order accuracy (order $n^{-2k}$) and is the preferred way of implementing numerical quadrature with equally-spaced samples. The need to keep track of the particular weights is thus obviated. The method can be made efficient by retaining all of the evaluations of the function $f$ as iteration proceeds so that no redundant evaluations are made.

Numerical quadrature can be made to work with improper integrals when the integral exists but when evaluation of the function at the endpoints is not possible. One way to proceed is to make use of a semi-open formula as the starting point for integration. A change of variables may also eliminate evaluation problems. The same approach may be useful for integration domains that extend to infinity.

It should be emphasized that high-order methods do not necessarily translate to high accuracy since accuracy also involves the size of the derivative of the function somewhere within the domain. High-order methods are only guaranteed to have high accuracy for smooth integrands which are well approximated by polynomials.

### 6.2.2 Gaussian quadrature

The quadrature rules outlined above involved variable weights but regularly-spaced sample points separated by the uniform interval $h$. An $n$th order rule is exact in this case for polynomials of degree $n-1$. Gaussian quadrature generalizes the rules in two ways. First, both the weights and the sample points are variable. This extra degree of freedom allows $n$th order rules to be exact for polynomials of degree $2n-1$. Second, in Gaussian quadrature, the integrand may be factored:

$$\int_a^b f(x) w(x)\, dx = \sum_{i=1}^{n} w_i f(x_i) \tag{6.24}$$

where $w(x)$ is a selectable weight function. Now, only $f(x)$ is required to be smooth like a polynomial for the integration to be accurate while the weight functions can be singular. Several weight functions have been explored, including $w(x) = 1$ (Gauss-Legendre quadrature), $w(x) = e^{-x}$ (Gauss-Laguerre quadrature), $w(x) = e^{-x^2}$ (Gauss-Hermite quadrature), and $w(x) = (1-x^2)^{-1/2}$ (Gauss-Chebychev quadrature) in the intervals where the given function is defined.

In principle, solving for the weights and the sample points amounts to solving the following system of $2n$ coupled equations for $2n$ unknowns:

$$\int_a^b f_\circ(x) w(x)\, dx = \sum_{i=1}^{n} w_i f_\circ(x_i) \tag{6.25}$$

$$\vdots = \vdots$$

$$\int_a^b f_{2n-1}(x) w(x)\, dx = \sum_{i=1}^{n} w_i f_{2n-1}(x_i) \tag{6.26}$$

the unknowns being both the weights $w_i$ and the sample points $x_i$. The equations are nonlinear now because the sample points are arguments of the functions and are consequently difficult to solve. More tractable means of calculating the weights and the sample points have been found including methods involving orthogonal functions and interpolating polynomials. For the purposes of this text, we regard the results as well-known, well-explored, tabulated commodities.

### 6.2.3 Higher dimensionality

## 6.3 Principle component analysis and empirical orthogonal functions

This chapter has been concerned with optimal representations of the candidate model. In this section, different ways of representing data are considered. Multidimensional data can sometime be transformed into new spaces where heretofore unseen order is revealed. If the data are mainly confined to a low-dimensional manifold within a higher-dimensional space, the potential for data compression also exists. Principle component analysis can in that sense be viewed as finding a low-rank approximation to a dataset, much as in image compression (see chapter 7).

Suppose for example that data comprised of three variables are sampled repeatedly in time and that the data are plotted as points in three spatial dimensions. If the data happen to fall along a line, then all three variables would be perfectly correlated, and the data would actually be unidimensional. The line is the direction along which the variance of the data is a maximum. If the data happen to fall in a plane, then the data are correlated and bidimensional. In either case, it would be expedient to transform the data into the appropriate, reduced-dimensional space for storage and analysis. If the data fall nearly on a line or in a plane, that fact alone might offer insight into the best means of analysis. Small deviations from the line or the plane could be neglected in some circumstances.

A tool which assesses the correlation of multidimensional datasets is principle component analysis. Consider a multidimensional data vector that is sampled repeatedly so that the entire dataset occupies a matrix $X$. Let each sample occupy a row of the matrix. For simplicity, each column is taken to have zero sample mean, i.e., the sample mean should be subtracted from each column prior to principle component analysis.

Next define the weight or loading vector $w$ as a unit vector in the space occupied by the data. The principle component score of a data sample is its dot product with the weight vector. The weight vector with the largest component score in the sense of the L2 norm is the one along which the variance of the data is a maximum, i.e.,

$$w = \underset{\|w\|=1}{\operatorname{argmax}} \|Xw\|_2^2 \quad = \quad \underset{\|w\|=1}{\operatorname{argmax}} w^T X^T X w \tag{6.27}$$

The matrix $R = X^T X$ can be recognized as the sample covariance matrix for the data. The problem is then to maximize $w^T R w$ subject to the constraint $w^T w = 1$. The corresponding objective function is then

$$L(w) = w^T R w - \lambda(w^T w - 1) \tag{6.28}$$

Since the covariance matrix is symmetric, the solutions can readily be seen to correspond to $Rw = \lambda w$ with $L_{\max} = \lambda$. Consequently, the critical points of the objective function are the eigenvectors of the sample covariance and the stationary values are the corresponding eigenvalues. The eigenvalues and their eigenvectors may be sorted from largest to smallest by convention.

The full principle component representation of the data $X$ is therefore given by $T = XW$ where $W$ is a matrix whose columns are the eigenvectors of $R = X^T X$. The rows of $T$ are individual samples in the space defined by the principle axes. Transformed thusly, the sample variances of the data are the eigenvalues of $R$, and the sample covariances are zero.

The eigenvectors or $R$ are also known as empirical orthogonal functions since they are a natural basis for a dataset, being defined by the dataset.

Dimensions in the new space in which the sample variances are small may be neglected by simple truncation of the transformation, i.e. $T_l = XW_l$, where the $l$ subscript denotes lower dimensionality by virtue of having neglected some of the columns of $T$ and $W$. Truncation here is similar to truncation in the context of SVD and may reduce instability

associated with over-fitting the data. The technique here is known as principle component regression. Truncation may also be used as a means of visualizing high-dimensional data that would be otherwise difficult to capture in 2D or 3D plots. Note that the compression is lossy just as in the SVD case. Lossless compression is outside the scope of this discussion.

## 6.4 Factor analysis

We return here to the factor analysis problem introduced in chapter 1. The problem is closely related to principle component analysis which places it in this chapter. The basic problem can be stated as

$$D = LF + E \tag{6.29}$$

where $D \in \mathbb{R}^{m \times n}$, $L \in \mathbb{R}^{m \times p}$, and $F \in \mathbb{R}^{p \times n}$. The matrix $D$ represents the known amounts of $m$ elements found in $n$ samples, $L$ the unknown amounts of $m$ elements composing $p$ compounds, and $F$ the unknown amounts of $p$ compounds found in $n$ samples. The sample errors are such that $E$ is a zero-mean random variable with covariance equal to the identity times the scalar $\psi$. It may be necessary to segregate the mean values from the rows of $D$.

The particular compounds in question may not be known a priori. The compounds actually play the role of basis functions for decomposing the samples. Since the problem statement is entirely in terms of the data, the resulting basis functions are defined by the data. In that way, factor analysis can be seen as a special case of principle component analysis, although additional issues come into play.

Principle component analysis and factor analysis share similar vocabulary and are often confused with one another. The former is a means of data decomposition. The latter combines data decomposition with prior information in order to make inferences.

The factor analysis problem is the problem of determining the loadings $L$ which are always presumed to be invariant. Once those are determined, the factors $F$ may be known immediately or may be estimated by minimizing the difference between the measured data and the data predicted by (6.29) is the least squares sense, i.e.

$$F_{\text{est}} = (L^T L)^{-1} L^T D \tag{6.30}$$

$$= L^T D, \quad m \geq p \tag{6.31}$$

(by analogy to the least-squares solution for the standard inverse problem and in the usual case of loadings which are orthogonal matrices.) The decomposition is not unique. If $D$ and $F$ are random variables, the decomposition can be constrained by requiring that $F$ and $E$ are independent and that $F$ have zero mean and identity covariance. In that case, the covariance of $D$ will be the sum of $LL^T$ (called the commonality) and $\psi I$ (called the uniqueness).

Principle value decomposition ($T = XW$) gives an estimate for the loadings when we identify $X^T$ with $D$, $W$ with $L$, and $T^T$ with $F$ (making use of the fact that $W^{-1} = W^T$. In other words, the loadings in the factor analysis context are the same as the loading in PCA within a constant. The decomposition is not unique, as any matrix times its inverse can be inserted between the loadings and the factors without altering the prediction for the data. In order to enforce the constraint that the covariance of the factors is the identity, the loadings computed from PCA should be scaled by the square roots of their corresponding eigenvalues.

As in PCA, truncation is indicated for the columns of $L$ and rows of $F$ associated with small eigenvalues. Truncation is desirable, since fewer factors is consistent with the overarching preference for simpler solutions. How many factors need to be retained? One rule of thumb is that eigenvalues smaller than unity can be neglected. Another is that eigenvalues which appear to be below the elbow in ordered plots (called "Scree" plots) can be neglected. This strategy is similar to the L-curved strategy in regularization problems. The number of important factors is the number of eigenvalues that are retained. A third strategy holds that the number of factors to retain is the number required to account for most of the covariance in the original data.

Since the decomposition is not unique, the loadings can be transformed in accordance with a priori preferences. Pursuing a prejudice for simple solutions, compounds which are mutually distinct and either very light or very heavy in every given element are to be preferred. The metric is therefore the "spikiness" of the loadings or, in mathematical

terms, the variances of the squares of the loadings. Simple rotations can be applied to the loadings to maximize the spikiness. Solving the factor analysis problem by maximizing the variance of the squares is an example of linearly-constrained maximum variance. When the rotations are orthogonal, the procedure is known as "varimax." Oblique rotations are certainly possible but are explored less commonly.

The factor analysis problem can equally well be viewed from the viewpoint of SVD. The problem is to determine the smallest number of compounds $p$ that can account for the data. The solution depends on whether $D$ spans the full space of $m$ elements or rather a smaller subspace $p$? Singular value decomposition is precisely the tool that can answer this question.

Factor analysis can proceed directly from the expansion

$$D \approx U_p \Lambda_p V_p^T = (U_p \Lambda_p) V_P^T = U_p (\Lambda_p V_P^T) = LF \tag{6.32}$$

where $p < m$ is the number of nonzero singular values and, in the present context, the number of essential factors. Where the parentheses go in (6.32) is a matter of convention. The convention for enforcing the identity covariance of the factors can still apply. Since $D$ contains sample noise, it is unlikely that any of the singular values determined through this expansion will be exactly zero. However, brutal truncation may be used to reduce $p$ to the bare minimum, just as in the case of PCA.

Rotation of the loadings can subsequently be used to optimize the solution further, just as with PCA. in PCA, rotations can be taken about the unused (truncated) principal axes. Using SVD, the rotations can be taken about the unused left singular vectors. In either event the factors corresponding to the new loadings can be estimated using (6.30). The unity covariance of the factors can also be enforced through additional scaling of the loadings.

Of course, the two perspectives are equivalent since the left singular vectors $U$ are the eigenvectors of $DD^T = X^T X$ which are just the loadings in PCA. SVD is in fact an efficient means of carrying out PCA. The two methods promote complementary interpretations of the factors – one based on the data covariance and the other on the four fundamental vector spaces. Moreover, the analysis is the same no matter whether the factors are taken to be deterministic or random.

## 6.5 Example: course grades

Applications of factor analysis are not limited to natural science and are widespread in social, behavioral, and medical science. Consider this modest, academic example. The semester grades (on a scale from 0-10) for three small college courses in a department are tabulated and found to be:

| | course 1 | course 2 | course 3 |
|---|---|---|---|
| student 1 | 2 | 6 | 5 |
| student 2 | 6 | 3 | 3 |
| student 3 | 9 | 9 | 8 |
| student 4 | 2 | 9 | 7 |
| student 5 | 9 | 6 | 5 |
| mean | 5.6 | 6.6 | 4.6 |
| st. dev | 3.14 | 2.24 | 1.74 |

Table 6.1: Table 6.1: Student grades in three courses

There appears to be a pattern in the test scores suggesting underlying structure. Reputation holds that course 1 is mathematically rigorous and incorporates long problem sets. Courses 2 and 3 involve more writing, and the grade is based mainly on a long term paper. Is this information reflected in the grades?

For this problem, the number of elements $m$ is three, and the number of samples $n$ is five. Proceeding with a statistical analysis beginning with PCA, the covariance of the data is readily found to be (after subtraction of the

course means and division by the standard deviations)

$$R = \begin{pmatrix} 1. & -0.0511 & 0.0804 \\ -0.0511 & 1. & 0.9810 \\ 0.0804 & 0.9810 & 1. \end{pmatrix}$$

This indicates that the grades in courses 2 and 3 are very highly correlated while being essentially independent of the grades in course 1. Underlying structure in the grades is indicated, and further analysis is warranted.

The eigenvalues of $R$ are:

$$(0.0102, 1.008, 1.9814)$$

and the corresponding loading matrix is:

$$L = W = \begin{pmatrix} -0.0936 & -0.9954 & 0.0212 \\ -0.7033 & 0.0812 & 0.7062 \\ 0.7047 & -0.0512 & 0.7077 \end{pmatrix}$$

whose columns are the eigenvectors of $R$. The eigenvalues represent the amount of variance in the data accounted for by the associated eigenvector. The first of the three eigenvalues is much smaller than the others and can be neglected, along with the corresponding eigenvector (first column of $L$). (A rough rule of thumb is that eigenvalues larger than 1 should be retained. If the system is large, the eigenvalues can be plotted in descending order in a so-called "Scree Plot," and a natural break point can be identified, much as in L-curve analysis.) A $p = 2$ system is what remains here, and the problem has two underlying factors.

Of the two retained principle axes, one is very nearly aligned with the "course 1" axis, and the other is midway between the "course 2" and "course 3" axes. As the loadings represent the amount of the given factor in the given course, it could be reasonable on the basis of prior information to associate the first factor with quantitative weakness and the second factor with writing strength.

By convention, the loadings are multiplied by the square root of the diagonal matrix whose entries are the eigenvalues, and the factors are multiplied by the inverse of the same matrix. This guarantees that the covariance of the factors is the identity. The corresponding loadings become (after truncation):

$$L = W \approx \begin{pmatrix} -0.9995 & 0.0299 \\ 0.0815 & 0.9941 \\ -0.0514 & 0.9961 \end{pmatrix}$$

The loadings are already "spiky," with all of the terms having squares close to zero or one. Varimax rotation about the remaining, neglected principle axes increases the variance of the squares of the loadings only slightly. The spikiness is maximized here by a 1-degree rotation.

Finally, the PCA decomposition of the data has the form $D \approx LF$ or $X^T \approx WT^T$ where the loadings $L = W$ are given above and the factors $F = T^T$ are:

$$F = T^T \approx \begin{pmatrix} 1.1336 & -0.1800 & -1.0581 & 1.1831 & -1.0784 \\ -0.3243 & -1.5522 & 1.2446 & 0.9227 & -0.2907 \end{pmatrix}$$

The covariance of the factors is the identity matrix, as promised. It can readily be verified that the truncation introduces only minor discrepancies in the predicted dataset. These figures can be interpreted as the degree to which the grades of given students exhibit the given factors. For example, the scores for student 4 are indicative both of quantitative weakness and writing strength.

## 6.6 Gaussian process regression: Kriging

Kriging is a popular method for interpolating or predicting data resulting from the measurements of a random process. The process is taken to be Gaussian so that the data are completely described by their mean and covariance. Take the

data to be sampled at $k = 1, \ldots, n$ points, yielding the set of samples $d_k$. What is the best estimate for the unsampled $d_\circ$?

The approach is to consider the ensemble of random processes governed by the specified mean and covariance that pass through the point which have been sampled. The best estimate for the value of an unsampled point is the one that best represents the ensemble at that point in a mean squared error sense. Statistical confidence in the estimate is determined by the variance of the ensemble about it at the sample point.

In Simple Kriging, the mean value of all the samples is taken to be zero. The estimate for $d_\circ$ is composed of a linear sum of the other samples, i.e.

$$\hat{d_\circ} = w^T d \tag{6.33}$$

where $w$ is a column vector of weights. Finding the best estimate for the weights is a matter of minimizing the mean-square error of the estimate:

$$\begin{aligned} w &= \underset{w}{\text{argmin}} \quad \langle \left(d_\circ - w^T d\right)^2 \rangle & (6.34)\\ &= \underset{w}{\text{argmin}} \quad \langle d_\circ^2 \rangle - 2\langle d_\circ w^T d \rangle + \langle w^T d d^T w \rangle & (6.35)\\ &= \underset{w}{\text{argmin}} \quad C_d(0,0) - 2w^T C_d(.,0) + w^T C_d w & (6.36) \end{aligned}$$

where the data covariance matrix has been introduced and where $C_d(.,0)$ refers to the 0th column of $C_d$, for example.

The solution for the weights comes from differentiating with respect to the weights, yielding the result

$$C_d^T w = C_d(.,0) \tag{6.37}$$

which gives the recipe for setting $w$ and estimating $d_\circ$, provided the inverse of $C_d^T$ exists. Of course an estimate of the data covariance matrix including the row and column corresponding to the new sample must be provided. Estimating this will most likely depend on assumptions about the stationarity or homogeneity of the underlying random process, as the case may be. With the Kriging weights thus calculated, the Kriging variance for the estimate is available from (6.36). This should be calculated to evaluate confidence in the estimate.

Ordinary Kriging permits the samples to have a nonzero mean and, moreover, estimates the mean on the fly. Note that the sample mean is not likely to provide a good estimate of the mean itself if the samples are highly correlated and that general least-squares estimation will generally be required to make the estimate properly. One could make that calculation, subtract the mean from $d$, and add it again to $d_\circ$, or one could proceed as follows.

In order to handle nonzero means, a simple constraint need be enforced:

$$w^T u = 1 \tag{6.38}$$

(where $u$ here is a column vector of ones). This ensures that $d_\circ$ is an unbiased representative of the underlying random process. This constraint can be introduced into the optimization problem in the usual way using a Lagrange multiplier $\lambda$, yielding the augmented result:

$$\left(\begin{array}{c|c} C_d^T & u \\ \hline u^T & 0 \end{array}\right) \left(\begin{array}{c} w \\ \lambda \end{array}\right) = \left(\begin{array}{c} C_d(.,0) \\ 1 \end{array}\right) \tag{6.39}$$

It can be shown that the variance of the estimator thus found grows slightly (by $\lambda$) due to the uncertainty in ascertaining the mean.

## 6.7 The Slepian function

## 6.8 References

## 6.9 Problems

# Part III: Iterative methods

# Chapter 7

# Iterative methods for linear problems

The aforementioned methods have all been explicit methods wherein the model estimate was formed by multiplying the data vector by a precomputed matrix which played the role of an approximate inverse operator. The solution is available immediately without the need for iteration. Such methods are only applicable to linear problems, since only then is the model related linearly to the data.

However, explicit methods may not be suitable for all linear problems. The explicit methods investigated to this point involved matrix inverses, matrix-matrix multiplies, and singular value decomposition. For $G \in \mathbb{R}^{n\times n}$, these operations involve $O(n^3)$ operations and so are computationally expensive. Consider for example the problem of image processing. An image with $1000\times1000$ pixels is constituted by $10^6$ pieces of data, and a model (a processed version of the same image) likewise will have $10^6$ elements. The corresponding system will therefore have $10^{12}$ elements!

This chapter provides an introduction to iterative inverse methods for linear inverse problems. While it may sound counter intuitive, iterative methods can be used to reduce computational cost. A successful method should have no operations more complicated than matrix-vector multiplies which involve only $O(n^2)$ operations. What is more, early termination of iterative methods (i.e. termination prior to convergence) is often possible and can be seen as a kind of regularization. When the system matrix is sparse, as is generally the case in large inverse problems, the computational cost can be reduced greatly further.

## 7.1 Method of steepest descent

All of the inverse problems considered thus far have amounted to optimization problems wherein an objective function is minimized. The objective or cost function may be based on the prediction error norm, the model norm, the spread of the model and data resolution matrices, or the predicted model error covariance. When the norms in question are based on L2, the objective function can always be placed in quadratic form:

$$f(x) = \frac{1}{2}(x,Ax) - (h,x) + c \tag{7.1}$$

with $A \in \mathbb{R}^{n\times n}$, $h \in \mathbb{R}^n$ and $c$ a constant. Here, the parentheses denote the vector inner product. We write that $f$ is symmetric, positive, or positive definite if $A$ is. The derivative of $f$ with respect to $x$ is $f' = Ax - h$ if $f$ is symmetric. Finding the critical point (maximum, minimum, or saddle point) of $f$ is therefore related to solving the linear set of equations $Ax - h = 0$.

It can be shown that if $f$ is positive ($(x,Ax) \geq 0\,\forall x$), the solution is a minimum. If $f$ is positive definite ($(x,Ax) > 0\,\forall x \neq 0$), the solution is unique.

The goal is to identify the solution point $z$ such that $Az - h = 0$. This point corresponds to the minimum of the

objective function. Let us define the residual vector $r$ at some point $x$ as $r \equiv -f'(x) = h - Ax = A(z-x)$. Clearly, this vanishes at the solution point.

**Lemma 7.1.1** *For some choice of* $\alpha$, $f(x) = f(x+2\alpha r)$ *and* $f(x+\alpha r) - f(x) \leq 0$.

In other words, there is some distance $2\alpha$ for which you can move in the direction of the residual vector and wind up at an equally high value of the objective function. Then, moving half that direction is guaranteed to reduce $f$. The proof of this lemma is straightforward:

$$\begin{aligned} f(x+2\alpha r) &= \frac{1}{2}\{x+2\alpha r, A(x+2\alpha r)\} - (h, x+2\alpha r) + c & (7.2)\\ &= f(x) + \frac{1}{2}\{(2\alpha r, Ax) + (2\alpha r, A2\alpha r) + (x, A2\alpha r)\} - (h, 2\alpha r) & (7.3)\\ &= f(x) + 2\alpha(r, Ax) + 2\alpha^2(r, Ar) - 2\alpha(h, r) & (7.4)\\ &= f(x) - 2\alpha(r, r) + 2\alpha^2(r, Ar) & (7.5) \end{aligned}$$

in which the symmetry of $A$ has been utilized. The last two terms can be made to vanish (and the first part of the lemma proved) by setting

$$\alpha = \frac{(r, r)}{(r, Ar)} \tag{7.6}$$

For this choice of $\alpha$, following the derivation above gives:

$$f(x+\alpha r) = f(x) - \frac{1}{2}\frac{(r, r)^2}{(r, Ar)} \leq f(x) \tag{7.7}$$

and proves the second part of the lemma.

The discussion above suggests the following algorithm for minimizing $f(x)$: which guarantees the progression

**Algorithm 1** Steepest descent

```
1: procedure ITERATE TO CONVERGENCE
2: guess x_0
3: r_0 = h − A x_0
4:     for k = 1, 2, ... do
5:         α_k = (r_{k−1}, r_{k−1}) / (r_{k−1}, A r_{k−1})
6:         x_k = x_{k−1} + α_k r_{k−1}
7:         r_k = h − A x_k
8:     end for
9: end procedure
```

$f(x_0) \geq f(x_1) \geq \ldots \geq f(x_k)$ and so would serve to be a serviceable algorithm. The algorithm is called the method of steepest descent because iteration is always in the direction of the residual (negative gradient) vector.

A significant problem arises when the condition number of the matrix $A$ is large, however. This implies that the ellipses of constant $f$ are highly elongated, since the condition number is the ratio of the semi-major to the semi-minor axis for quadratic equations. When the ellipses are thin, the residual vector never points toward the minimum except in the special case where it was aligned with a principle axis to begin with.

Consider the following. We have:

$$\begin{aligned} r_k &= h - Ax_k & (7.8)\\ &= h - A(x_{k-1} + \alpha_k r_{k-1}) & (7.9)\\ &= r_{k-1} - \alpha_k A r_{k-1} & (7.10) \end{aligned}$$

Taking the inner product of both sides with the residual vector from the previous timestep gives:

$$(r_{k-1}, r_k) = (r_{k-1}, r_{k-1}) - \frac{r_{k-1}, r_{k-1}}{r_{k-1}, Ar_{k-1}}(r_{k-1} A r_{k-1}) \tag{7.11}$$

$$= 0 \tag{7.12}$$

so that the residual from every iteration is orthogonal to the that from the previous iteration. The tendency will be for residual vectors to alternate back and forth in such a way that progress will occur in many short steps and be slow. Algebraic convergence only occurs in the special case where $r_{k-1} = \alpha_k A r_{k-1}$, i.e., when $r_{k-1}$ is an eigenvector of A. Since the eigenvectors are mutually orthogonal, the first residual must be along an eigenvector, i.e., a principle axis of the residual ellipse.

## 7.2 Method of conjugate gradients

Defficiencies in the method of steepest descent invite a better strategy for solving quadratic optimization problems. Suppose the distance vector $z - x_\circ$ could be deconstructed into into $n$ projections along $n$ orthogonal basis vectors spanning $A$ which is order $n \times n$. If the basis vectors and the distance to move along each could be computed in a single iteration, then convergence on the solution in $n$ iterations could be guaranteed (except for considerations of roundoff error). This is what the method of conjugate gradients accomplishes.

Replace the residual vectors $r_k$ with the basis or "conjugate" vectors $p_k$ in the steepest descent algorithm. In that case, the update equation becomes $x_k = x_{k-1} + \alpha_k p_{k-1}$. Finding the size of $\alpha$ proceeds as before:

$$f(x_k + \alpha p_k) = f(x_k) + \frac{1}{2}\alpha^2 (p_k, Ap_k) - \alpha(p_k, r_k) \tag{7.13}$$

where we note that the residual vector $r_k = h - Ax$ continues to appear in the conjugate-gradient formalism. To find the value of $\alpha$ that minimizes $f$, we set the derivative of the function with respect to $\alpha$ to zero, yielding

$$\alpha_{k+1} = \frac{(p_k, r_k)}{(p_k, Ap_k)} \tag{7.14}$$

$$= \frac{(r_k, r_k)}{(p_k, Ap_k)} \tag{7.15}$$

where the last step is left as an exercise for the reader. Note that the $\alpha_k$ are just the coefficients needed to reconstruct $z - x_\circ$ from the conjugate vectors.

The algorithm is still incomplete as the procedure for finding the conjugate vectors remains to be specified. A crucial property of the conjugate vectors is is their orthogonality with respect to $A$, i.e., $(p_i, Ap_j) = 0$, $i \neq j$. This is referred to as "A-orthogonality" or "conjugacy."

The method of conjugate gradients efficiently determines the conjugate vectors "on the fly," starting from and correcting the residual vectors and incorporating the previous conjugate vector. Suppose $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$. The task then is to find $\beta$ to enforce the orthogonality of the conjugate vectors with respect to $A$. This can be done by taking the appropriate inner product of the recursion relation.

$$(p_k, Ap_{k+1}) = (p_k, Ar_{k+1}) + \beta_{k+1}(p_k, Ap_k) \tag{7.16}$$

The value of beta required to ensure orthogonality with respect to $A$ is:

$$\beta_{k+1} = -\frac{(p_k, Ar_{k+1})}{(p_k, Ap_k)} \tag{7.17}$$

$$= -\frac{(r_{k+1}, Ap_k)}{(p_k, Ap_k)} \tag{7.18}$$

$$= \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)} \tag{7.19}$$

where the second step follows from the symmetry of $A$ and the third step is, once again, left as an exercise for the reader to prove.

In the spirit of computational efficiency, it would also be useful to be able to generate the residual vectors recursively. This can be accomplished as follows:

$$\begin{aligned} r_{k+1} &= h - Ax_{k+1} && (7.20)\\ &= h - A(x_k + \alpha_{k+1} p_k) && (7.21)\\ &= (h - Ax_k) - \alpha_{k+1} A p_k && (7.22)\\ &= r_k - \alpha_{k+1} A p_k && (7.23) \end{aligned}$$

where it can be noted that none of the operations involved are computationally more expensive than matrix-vector multiplies.

We are almost ready for an algorithm, but the efficacy of the conjugate vectors as an expansion basis for $A$ still remains to be demonstrated. It is clear that vectors that are orthogonal with respect to $A$ are automatically linearly independent. It is therefore possible to expand the solution $z$ uniquely as a linear combination of the conjugate vectors

$$z = \gamma_\circ p_\circ + \gamma_1 p_1 + \cdots + \gamma_{n-1} p_{n-1} \tag{7.24}$$

where $A$ is order $n$. The coefficients can be found be using the A-orthogonality of the conjugate vectors

$$(p_i, Az) = \gamma_i (p_i, Ap_i) \quad \rightarrow \quad \gamma_i = \frac{(p_i, Az)}{(p_i, Ap_i)} \tag{7.25}$$

The same expansion can be done for the discrepancy vector $z - x_\circ$, i.e. $z - x_\circ = \sum_{i=0}^{n-1} \xi_i p_i$ where

$$\xi_k = \frac{(p_k, A(z - x_\circ))}{(p_k, Ap_k)} \tag{7.26}$$

Now, we know that the following must be true:

$$\frac{(p_k, A(x_k - x_\circ))}{(p_k, Ap_k)} = 0 \tag{7.27}$$

since $x_k = x_{k-1} + \alpha p_{k-1}$ and so $x_k - x_\circ = \sum_{i=1}^{k} \alpha_k p_{k-1}$, meaning that the difference vector does not have a projection in $p_k$. Subtracting (7.27) from (7.26) gives

$$\begin{aligned} \xi_k &= \frac{(p_k, A(z - x_\circ))}{(p_k, Ap_k)} - \frac{(p_k, A(x_k - x_\circ))}{(p_k, Ap_k)} && (7.28)\\ &= \frac{(p_k, A(z - x_k))}{(p_k, Ap_k)} && (7.29)\\ &= \frac{(p_k, r_k)}{(p_k, Ap_k)} && (7.30) \end{aligned}$$

where the last step makes use of the fact that $Az = h$ and that $x_k$ has no projection in $p_k$. These are the coefficients for $z - x_\circ$ expanded in a basis made from the conjugate vectors. They are also the the $\alpha$ coefficients in the method of conjugate gradients. Thus, the principle underlying the method is demonstrated.

Finally, the method of conjugate gradients: where convergence is guaranteed in at most $n$ steps except for the effects of numerical roundoff error which may necessitate more steps. The algorithm rests upon a number of lemmas and properties which have been left as exercises for the reader:

$$\begin{aligned} (r_i, p_j) &= 0, \quad 0 \le j < i \le n && (7.31)\\ (r_i, p_i) &= (r_i, r_i) && (7.32)\\ (r_i, r_j) &= 0, \quad 0 \le i < j \le n && (7.33)\\ -\frac{(r_{k+1}, Ap_k)}{(p_k, Ap_k)} &= \frac{(r_{k+1}, r_{k+1})}{(p_k, r_k)} && (7.34)\\ \frac{(p_k, r_k)}{(p_k, Ap_k)} &= \frac{(r_k, r_k)}{(p_k, Ap_k)} && (7.35) \end{aligned}$$

**Algorithm 2** Conjugate gradients

**procedure** ITERATE TO CONVERGENCE
guess $x_\circ$
take $p_\circ = r_\circ = h - Ax_\circ$
  **for** $k = 0, 1, \ldots$ **do**
    $\alpha_{k+1} = (r_k, r_k)/(p_k, Ap_k)$
    $x_{k+1} = x_k + \alpha_{k+1} p_k$
    $r_{k+1} = r_k - \alpha_{k+1} A p_k$
    $\beta_{k+1} = (r_{k+1}, r_{k+1})/(r_k, r_k)$
    $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$
  **end for**
**end procedure**

## 7.3 Conjugate gradient least squares

Of interest to this text so far have been inverse problems involving variations on the basic linear least squares problem. The most basic formulation of the problem is the minimization of the model prediction error $\|Gm - d\|_2^2$. By the method of normal equations, the minimum corresponds to the solution of the system of equations $G^T Gm = G^T d$. We therefore recognize $A = G^T G$ and $h = G^T d$ in the current context.

We note that $A$ is symmetric and at least positive semidefinite. For overdetermined problems, $A$ is positive definite and a unique minimum exists. For mixed and under determined problems, $A$ is positive but has a ridge. There is no unique solution, and the solution found iteratively will depend on the initial guess $x_\circ$. If $x_\circ = 0$, conjugate gradient least squares (CGLS) will converge on the minimum model length solution. Other prior information can figure in the selection of $x_\circ$.

In the spirit of numerical expedience, we would prefer never to compute the matrix-matrix product $A = G^T G$. This is possible with the appropriate factorization. For example, in the standard conjugate gradient algorithm, wherever products like $(p, Ap)$ appear, there is the factorization $(p, Ap) = p^T Ap = p^T G^T Gp = q^T q = (q, q)$, where an auxiliary vector $q$ has been introduced.

The standard conjugate gradient algorithm also has a factor $Ap_k$ in the residual recursion relation that needs to be avoided in its present form ($r_{k+1} = r_k - \alpha_{k+1} Ap_k$). In the context of CGLS, the residual vector is $r = h - Ax = G^T(d - Gm)$. We can evaluate the part in parenthesis first, evolve the error vector $d - Gm$, and then multiply by $G^T$ as a final step. For the sake of clarity, we also replace $x$ with $m$. Then, define

$$\begin{aligned} s_{k+1} &= d - Gm_{k+1} && (7.36)\\ &= d - G(m_k + \alpha_{k+1} p_k) && (7.37)\\ &= d - Gm_k - \alpha_{k+1} Gp_k && (7.38)\\ &= s_k - \alpha_{k+1} q_k && (7.39) \end{aligned}$$

making use of the $q_k$ variable once more. Finally, The residual vector is just $r_k = G^T s_k$. Finally, the CGLS algorithm is written out below. While this is admittedly more complicated to code than the explicit methods considered earlier, its performance can be greatly superior, particularly where large systems are involved.

## 7.4 Regularization

Much of the first half of this text concerned augmenting the basic least squares problem to include data and model weights and regularization. All of those features remain available for inclusion in CGLS. The most general statement of the least squares problem was given back in (2.36) which is repeated below.

$$m = \underset{m}{\operatorname{argmin}} \left\| \begin{pmatrix} C_d^{-1/2} G \\ \varepsilon L \end{pmatrix} (m - m_\circ) - \begin{pmatrix} C_d^{-1/2} (d - Gm_\circ) \\ 0 \end{pmatrix} \right\|_2^2 \qquad (7.40)$$

**Algorithm 3** CGLS

**procedure** ITERATE TO CONVERGENCE
guess $m_\circ$
take $s_\circ = d - Gm_\circ$
take $r_\circ = p_\circ = G^T s_\circ$
take $q_\circ = Gp_\circ$
    **for** $k = 0, 1, \ldots$ **do**
        $\alpha_{k+1} = (r_k, r_k)/(q_k, q_k)$
        $m_{k+1} = m_k + \alpha_{k+1} p_k$
        $s_{k+1} = s_k - \alpha_{k+1} q_k$
        $r_{k+1} = G^T s_{k+1}$
        $\beta_{k+1} = (r_{k+1}, r_{k+1})/(r_k, r_k)$
        $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$
        $q_{k+1} = Gp_{k+1}$
    **end for**
**end procedure**

The data weights are contained in the data covariance matrix $C_d$, and the model weights are expressed by $\varepsilon^2 L^T L$. This minimization problem is in the form $\|Gm - d\|_2^2$ only for augmented $G$, $m$, and $d$. The CGLS method is ready to tackle this problem without further generalization. The established methods for setting the regularization parameter apply.

There is, however, another option for regularization that can be exceptionally expedient numerically. As stated above, the method of conjugate gradient moves from the initial guess $m_\circ$ to a solution $z$ progressively as it iterates. As it does so, the chi-squared parameter decreases as the solution is approached. Suppose the initial guess for $m_\circ$ was the zero vector. The length of the model vector would initially be zero but would grow with successive iterations. A record of the norm of the model length versus the norm of the data prediction error during iteration would trace out an L-curve. Terminating iteration at the knee of the curve would be tantamount to $0th$-order Tikhonov regularization!

In the event that a compelling prior model $m_\circ$ exists, it can be specified using (7.40) or some variation on it, and then CGLS can be used as an update. In this case, it would not be the length of the model but rather the length of the deviation of the model from the prior model that would grow with successive iteration. It might very well make more sense to limit the length of the model deviation vector $m - m_\circ$ rather than the length of the model vector $m$ through early termination in such cases. This strategy is termed "creeping" as opposed to regularization in the absence of a prior model which is called "jumping."

## 7.5 Sparse math

Efficient computation using sparse matrices is a field of its own in computational mathematics which will cannot be treated in any real depth here. Optimal methods exist for different operations and different kinds of sparse matrices. However, the utility of sparse matrix computations can be demonstrated by considering the elementary problem of the matrix-vector product. A simple approach to this problem can reduce the computational cost of CGLS dramatically.

Consider the problem $y = Ax$. Explicitly, the product implies (for $A \in \mathbb{R}^{n \times m}$) $m$ multiplies and $m - 1$ additions for each of the $n$ rows of $A$, as summarize by

$$y_i = \sum_{j=1}^{m} A_{ij} x_j \quad \forall\, i \in n \tag{7.41}$$

Suppose that most of the elements of the matrix $A$ are zeros. The computational cost of all the associated multiplies and adds can be eliminated by storing $A$ in compact form. The idea is to store only the nonzero elements of $A$ (in a

floating-point array) together with their indices (in integer arrays), i.e.,

$$A_k = A_{i,j} \tag{7.42}$$

$$i_k = i \tag{7.43}$$

$$j_k = j \tag{7.44}$$

where $k$ is the index counter for all nonzero elements of $A(i, j)$. Then, matrix multiplication is carried out as:

$$y_i = \sum_k A_k x_{j_k} \delta_{i,i_k} \quad \forall k \tag{7.45}$$

The problem with this storage scheme is that it is time-consuming to retrieve any particular value of $A(i, j)$. However, this is not a problem for matrix-vector multiplies when only $A$ is stored in compact form.

## 7.6 Example: image processing

An ideal application for CGLS is image processing. Suppose an image has undergone blurring due to imperfections in the camera optics, target motion, or media degradation. Suppose further that the blurring can be represented as convolution with an averaging kernel and that the kernel is somehow known. Image de-blurring is a linear deconvolution problem like others already considered in this text. As mentioned at the start of the chapter, the difference is one of scope, since the number of elements in the system $G$ is the square of the number of elements in the original image. Mega-pixel cameras being commonplace, the de-blurring problem is intractable by explicit methods but highly suited for CGLS using sparse array storage.

Figure 7.1 is an example of image processing using CGLS. The image in this case is in grayscale format and has of the order of $100 \times 100$ pixels. Blurring of the original image was performed using a Gaussian averaging kernel extending a radial distance of three pixels from the center pixel. In addition to blurring, the original images was also contaminated with the introduction of uniformly distributed independent noise. The noise was purely positive going and therefore darkening since larger values denote darker pixels in this image format.

CGLS inversion was performed without any provision for regularization, and the results were tabulated for 80 iterations. The panel in the lower-left quadrant of Figure 7.1 shows the relationship between the square of the L2 norm of the processes image versus the mean-square error between the processes image and the original. The relationship traces out an L-curve with a knee occurring around the $20th$ iteration. Further iteration seems only to increase the length of the image without reducing the norm of the error significantly.

The image in the lower-right quadrant Figure 7.1 shows the final image recovered. Most of the detail lost through the blurring process was recovered at the cost of some obvious image artifacts. At least the "8" is clearly visible, and the depth in the original image lost after blurring seems to have been mostly recovered. The computation time for 80 iterations on a single i7 processor is about 10 s.

## 7.7 Preconditioning

## 7.8 Biconjugate gradients

## 7.9 References

## 7.10 Problems

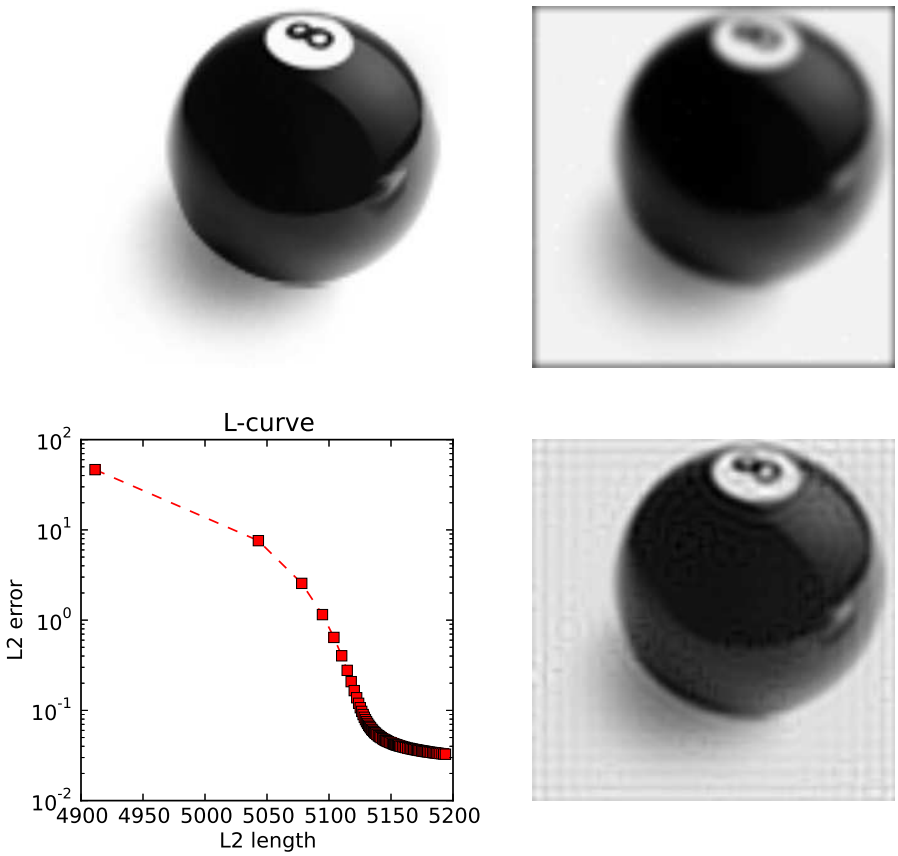


Figure 7.1: Image processing example using conjugate gradient least squares. The original grayscale image, which has $\sim 100 \times 100$ pixels, is shown in the upper-left panel. In the upper-right panel is the image having undergone blurring with a Gaussian filter along with the addition of independent noise. An L-curve is shown in the lower-left panel. The method of regularization is simply early termination, and the L-curve plots results for 1–80 iterations. A reconstructed image using the regularization parameter inferred from the L-curve (20 iterations) is shown in the lower-right panel.

# Chapter 8

# General nonlinear methods

The preceding text was limited to the consideration of linear inverse problem. To summarize, the methods developed to solve them mainly dealt with minimizing an objective function which combined a norm of the data prediction error with other penalties related to the length of the model or one of its derivatives, the model or data resolution, or the model error covariance. These other penalties regularized the problem. The objective in every case was to balance distortion due to fluctuations with those due to biases in the end result. A few tools are available for tuning the regularization and establishing the right balance.

Of course, a great many inverse problems are nonlinear, and solving those problems requires different methods. While the mechanics are quite different, the underlying objectives are the same. Material presented in this chapter, which takes up nonlinear inverse problems, should be familiar.

Nonlinear problems can be converted approximately into linear ones through "linearization" where the dependent variables are expanded in terms of background parameters plus perturbations to them. Only terms that are linear in the perturbations are retained, and so solving for the perturbations is a linear problem by definition. Once calculated, the perturbations can then be added to the background parameters, and a new set of perturbations found using the same method. Analysis proceeds iteratively until criteria for convergence are met. If the initial guess for the background parameters is sufficiently close to the solution, the method should converge provided that the problem was well conditioned. If not, then regularization is required. Distinguishing local minima from the global minimum may be challenging depending on the complexity of the objective function.

The main task for this chapter is to develop the mechanics for posing nonlinear inverse problems computationally, introducing regularization, defining convergence criteria, and carrying out the iterations. The computational framework most often used for the last half century is the one due to Marquardt and Levenberg. To understand their method, some more obvious strategies must first be reviewed.

## 8.1 Newton's method

Newton's method can be used to solve a system of nonlinear equations. Recall from the previous chapter how the solution to optimization problems implies such a system of equations. The system of equations can be stated as $f(x) = 0$ with $x \in \mathbb{R}^m$ and $f \in \mathbb{R}^n$ vectors joined through a nonlinear relationship. The object is to find $x$ that makes $f$ the zero vector.

The function can be expanded in a Taylor series about an initial guess $x_\circ$ as

$$f(x_\circ + dx) = f(x_\circ) + \nabla f(x_\circ) dx + \ldots \tag{8.1}$$

where $\nabla f = J$ is the Jacobian matrix defined as:

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{pmatrix} \tag{8.2}$$

which is to be evaluated at the current guess for $x$. Near the solution, $f(x_\circ + dx)$ vanishes, implying that

$$J(x_\circ)dx \approx -f(x_\circ) \tag{8.3}$$

This gives a formula for updating the displacement vector $dx$ and refining the search for the solution. The algorithm for Newton's method is then simply:

**Algorithm 4** Newton's method

```
1: procedure ITERATE TO CONVERGENCE
2: guess x_o
3:     for k = 0, 1, ... do
4:         dx = −J^{−1}(x_k) f(x_k)
5:         x_{k+1} = x_k + dx
6:     end for
7: end procedure
```

If the derivatives of $f$ are continuous and $J$ is nonsingular, Newton's method (given a suitable initial guess) converges at a quadratic rate such that

$$||x^{k+1} - x||_2^2 < ||x^k - x||_2^2 \tag{8.4}$$

where $x$ is the solution vector and $x^k$ is the $k$th iterate. In cases where convergence is marginal, performance may be improved by updating the solution according to $x_\circ \to x_\circ + \delta dx$ where $\delta$ has a value between zero and one. Another strategy is to vary $\delta$ so as to minimize $f(x_\circ + \delta dx)$. This is called a line search which can also be used to restrict $x_\circ$ from entering a forbidden territory. For example, line searches can be used to exclude state vectors with negative values in an approach known as non-negative least squares (NNLS). In any case, convergence demands an initial guess close to the solution vector.

## 8.2 Newton's optimization method

Newton's method can be applied to optimization functions in which an objective function or cost function is minimized. Consider the Taylor expansion of such a cost function around a guess for the minimum $x_\circ$

$$c(x_\circ + dx) = c(x_\circ) + \nabla^T c(x_\circ)dx + \frac{1}{2}dx^T \nabla^2 c(x_\circ)dx + \ldots \tag{8.5}$$

where the cost function $c$ is a scalar. In this case, $\nabla c$ is the gradient vector

$$\nabla c = \begin{pmatrix} \frac{\partial c}{\partial x_1} \\ \vdots \\ \frac{\partial c}{\partial x_m} \end{pmatrix} \tag{8.6}$$

and $\nabla^2 c = H$ is the Hessian matrix

$$H = \begin{pmatrix} \frac{\partial^2 c}{\partial x_1^2} & \cdots & \frac{\partial^2 c}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 c}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 c}{\partial x_1 \partial^2 x_m} \end{pmatrix} \tag{8.7}$$

which are both to be evaluated at the current guess $x_\circ$. Near the solution, the gradient term vanishes. The gradient can be expanded in a Taylor series as

$$\nabla c(x_\circ + dx) = \nabla c(x_\circ) + \nabla^2 c(x_\circ) dx + \dots \tag{8.8}$$

leading to the update equation

$$H(x_\circ) dx = -\nabla c(x_\circ) \tag{8.9}$$

and the algorithm

**Algorithm 5** Newton's optimization method

```
1: procedure ITERATE TO CONVERGENCE
2: guess x_∘
3:     for k = 0, 1, ... do
4:         dx = −H^{−1}(x_k)∇c(x_k)
5:         x_{k+1} = x_k + dx
6:     end for
7: end procedure
```

This is just Newton's method for $f(x) = \nabla c(x)$ and enjoys the same convergence properties. The method is applied in just the same way, and the requirement for a good initial guess remains the same. The advantage of Newton's method is simplicity and ease of coding. The main disadvantages of Newton's optimization method are finicky convergence when $H$ is near singular and high computational cost, since $H$ involves second derivatives. Because of these disadvantages, the method is generally not used in contemporary practice, much as the method of steepest descent is not used for solving linear optimization problems.

## 8.3 Newton Gauss and Levenberg Marquardt

In the previous chapter, the method of conjugate gradients was developed but eschewed because of the necessity of computing matrix-matrix products ($G^T G$ in particular). The CGLS method was then developed to take advantage of the fact that the objective function was quadratic. A more computationally expedient algorithm resulted. Here, the same strategy will be followed only in the context of nonlinear problems.

Let us consider explicitly a cost function having the form of $\chi^2$, i.e.

$$c(m) = (G(m) - d)^T C_d^{-1} (G(m) - d) \tag{8.10}$$

where $m$ is the vector specifying the model or state vector, $d$ is the data vector, and $G$ is the (now nonlinear) function relating them. The $C_d$ matrix is the data error covariance matrix. Defining the residual function $f(m) = C_d^{-1/2}(G(m) - d)$ means that the cost function can be written in the form $c = f^T f = \|f\|_2^2$. By the rules of vector calculus, the gradient of $c$ is then $\nabla c = 2J^T(f(m))\, f$. In what follows, $J(f(m))$ will be abbreviated $J(m)$ to streamline the notation. The Jacobian is calculated in this case in the manner specified by (8.2).

A similar calculation can be performed for the required Hessian matrix. According to the chain rule, the Hessian matrix should have two components. The first of these is $2J^T J$, and the second involving he gradient of the Jacobian times $f$. Since $f$ will be small near the solution by definition, this term may be neglected, giving

$$H \approx 2J^T J \tag{8.11}$$

which involves only the first derivative matrix rather than the second and should be significantly less expensive to calculate. Consequently, Newton's optimization method for functions of the form in question becomes

$$J^T(m) J(m) dm = -J^T(m) f(m) \tag{8.12}$$

which requires only the calculation of the Jacobian matrix for a given timestep. The iteration scheme based on (8.12) is called the Gauss Newton method.

A serious potential problem with Gauss Newton is singularity or near-singularity in the $J^T J$ term. The Levenberg Marquardt method is based on Gauss Newton but adds a small diagonal term to $J^T J$ to assure invertability. The Levenberg Marquardt algorithm is therefore given by

**Algorithm 6** Levenberg Marquardt

```
1: procedure ITERATE TO CONVERGENCE
2: guess m_o
3: initialize λ
4:     for k = 0, 1, ... do
5:         dm = −(J^T(m_k)J(m_k) + λI)^{−1} J^T(m_k) f(m_k)
6:         m_{k+1} = m_k + dm
7:         update λ
8:     end for
9: end procedure
```

where the $\lambda$ or "damping" parameter is adjusted throughout iteration to assure convergence. Note that when $\lambda$ is large, the method is just steepest descent. When it is small, the method is Newton Gauss. The algorithm can change from the latter to the former when it fails to make progress and from the former to the latter as the solution vector is approached. Various strategies for updating $\lambda$ dynamically exist. Note that $\lambda$ is introduced to improve the stability of the iteration but not of the underlying physical problem. It is not a form of regularization and will ideally be removed before the solution is ultimately rendered.

### 8.3.1 Error propagation

The Levenberg Marquardt algorithm addresses the problem of nonlinear least squares estimation and is analogous to linear least squares. As the solution is approached, Levenberg Marquardt becomes Gauss Newton. When the solution is finally reached, the last iteration of Gauss Newton is equivalent to the linear least squares solution of the linearized problem. This can be seen by comparing (8.12) to the weighted least squares solution from chapter 2:

$$m = m_\circ + \left(G^T C_d^{-1} G\right)^{-1} G^T C_d^{-1} (d - G m_\circ) \tag{8.13}$$

Taking $m$ and $m_\circ$ to be the final and penultimate model vector estimates, respectively, so that $dm = m - m_\circ$ and identifying $J$ in (8.12) with $C_d^{-1/2} G$ reproduces (8.13).

Error propagation with Levenberg Marquardt can therefore be treated in a fashion analogous to linear least squares. Recall that, for linear problems, normally distributed errors in the data produced normally distributed errors in the model estimate governed by the transformation:

$$\begin{aligned} G C_m G^T &= C_d && (8.14)\\ C_d^{1/2} J C_m J^T C_d^{T/2} &= C_d && (8.15)\\ C_m &= \left(J^T J\right)^{-1} && (8.16)\\ &\approx H^{-1} && (8.17) \end{aligned}$$

This is the prescription for error analysis in nonlinear least squares estimation where the data error covariance matrix has already been included as a factor in $f$, $c$, and $J$. Note that no factor of two appears in the relationship between the model error covariance matrix and the Hessian matrix. That relationship will be instructive later when the topic is error propagation in problems involving regularization.

In general, errors in the model estimate will not be normally distributed in nonlinear inverse problems even if the errors in the data are normally distributed. However, they can be regarded as being approximately normally distributed in the event that the data errors are small perturbations and that the solution vector has been found accurately.

### 8.3.2 Implementation

Implementing the Levenberg Marquardt algorithm is straightforward, but at least three subtle issues need to be addressed. These are discussed briefly below.

**Damping parameter** Various strategies have been advanced for setting the $\lambda$ parameter so as to promote local and global convergence. Marquardt suggesting defining an initial value for $\lambda_\circ$ along with some $\nu > 1$. The problem is solved for two values of $\lambda$: $\lambda_\circ$ and $\lambda_\circ/\nu$. The metric for evaluating progress is the length of the residual vector. If $\lambda_\circ/\nu$ results in improvement, it is adopted as the new $\lambda_\circ$. Otherwise if $\lambda_\circ$ results in improvement, the value is retained. If neither results in improvement, the damping factor is multiplied by factors of $\nu$ until there is improvement. A new state vector estimate is set each time improvement occurs.

**Convergence tests** Iteration proceeds until convergence tests are satisfied. Convergence usually means that either the norm of the gradient of the cost function is small and/or that incremental changes to the state vector and cost function are small. Define $\varepsilon$ as the accuracy to which $G(m)$ can be calculated. Then suitable criteria for convergence could be:

$$\|\nabla c(m_k)\|_2 < \sqrt{\varepsilon}(1+|c(m_k)|) \tag{8.18}$$

$$\|m_k - m_{k-1}\|_2 < \sqrt{\varepsilon}(1+\|m_k\|_2) \tag{8.19}$$

$$|c(m_k) - c(m_{k-1})| < \varepsilon(1+|c(m_k)|) \tag{8.20}$$

Convergence requires that the cost function $c$ must be continuous and differentially continuous through its second derivatives. These conditions may be violated if the system is defined in terms of piecewise functions, for example, and so care must be taken in specifying the direct model. Convergence also requires the cost function to have a well-defined minimum as opposed to a ridge or flat depression. The absence of a well-defined minimum can correspond to a Jacobian matrix that is singular or nearly singular, implying poor conditioning and instability. As in linear problems, instability in nonlinear problems can be mitigated by regularization (see below).

**Calculating the Jacobian** Most of the computation time in Levenberg Marquardt is devoted to the computation of the Jacobian 1st-derivative matrix. If at all possible, the computation should be based on analytic forms of the derivatives. If those forms are unavailable due to the complexity of the system or to incomplete knowledge about it, then it may be necessary to compute the Jacobian using finite differences, i.e. (forward difference)

$$\frac{\partial f_i}{\partial m_j} \approx \frac{f_i(m+\Delta m_j) - f_i(m)}{\Delta m_j} \tag{8.21}$$

where $\Delta m_j$ is a small increment to the $j$th component of $m$. The step size $\delta m_j$ is generally set to $\sqrt{\varepsilon}$ so that the finite difference approximation is reasonably accurate while also being reasonably immune to numerical noise.

Finally, nonlinear least squares estimation is prone to fail in the event that the cost function has local minima in the vicinity of the global minimum. Convergence on the global minimum will generally require a good initial guess. When all else fails, running the algorithm multiple times with multiple initial guesses may be an acceptable if computationally expensive strategy. A grid search may be used to explore the space of all initial guesses methodically. Finally, in the case of penalty functions with a large number of closely-spaced local minima, optimization strategies based on stochastic algorithms may be practical. This is the subject of a subsequent chapter in the text.

**Factorization**

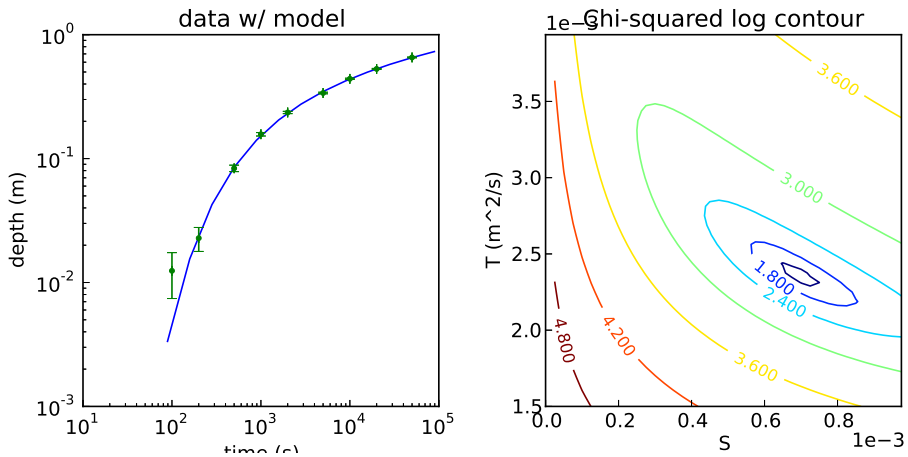


Figure 8.1: LM solution to the Theis hydrology problem.

| $t$ (s) | 100 | 200 | 500 | 1000 | 2000 | 5000 | 10000 | 20000 | 50000 |
|---|---|---|---|---|---|---|---|---|---|
| $h_\circ - h$ (m) | 0.01242 | 0.02281 | 0.08351 | 0.15720 | 0.23568 | 0.33846 | 0.44067 | 0.52940 | 0.65543 |

Table 8.1: Data for $h_\circ - h(t)$ (m).

## 8.4 Example: radial flow from a well

We consider the problem in hydrology of radial flow into a well from which fluid is being pumped. The pumping induces horizontal hydraulic gradients toward the well, causing hydraulic heads (fluid levels) in surrounding wells to drop. The decrease depends on the transmssivity $T$ and storativity $S$ of the aquifer, the constant pumping rate $Q$, and on the radial distance from the well $r$ and time $t$.

The partial differential equation describing the two-dimensional (horizontal) flow in a confined aquifer is a diffusion equation:

$$\nabla^2 h = \frac{S}{T}\frac{\partial h}{\partial t} \tag{8.22}$$

where $h(r,t)$ is the fluid level. The flow is assumed to be radially symmetric here, and $\nabla^2 h = \partial^2 h/\partial r^2 + (1/r)\partial h/\partial r$. The initial condition is a level surface, i.e. $h(r,0) = h_\circ$. The boundary condition at infinity is $h(\infty, r) = h_\circ$, i.e., there is no drawdown at long distances. The well diameter is taken to be infinitesimal, and the boundary condition at $r = 0$ is imposed by the constant pumping rate there:

$$\lim_{r\to 0}\left(r\frac{\partial h}{\partial r}\right) = \frac{Q}{2\pi T} \tag{8.23}$$

The solution to this initial boundary value problem was given first by Charles Theis and can be stated as

$$h_\circ - h(r,t) = \frac{Q}{4\pi T}W(u) \tag{8.24}$$

$$u \equiv \frac{r^2 S}{4Tt} \tag{8.25}$$

where $W(u) \equiv \int_u^\infty (e^{-x}/x)dx$ is the exponential integral, a well-known special function. In the present context, it is also called the well function. Note that the dependence of the well head on the parameters $S$ and $T$ is very clearly nonlinear.

Table 8.1 shows hydraulic head data from a well $r$ = 55 m away from the main well which is being drawn down at the rate of $Q = 4.0\times10^{-3}$ m$^3$/s. The measurements are statistically independent and have equal uncertainties characterized by $\sigma$ = 5 mm.

Figure 8.1 shows the result of performing a nonlinear least-squares fit of the data in Table 8.1 to the formula in (8.24). The panel on the left side of the figure shows the best fit curve which corresponds to $S = 6.96 \times 10^{-4}$ and $T = 2.36 \times 10^{-3}$ m$^2$/s. Experimental uncertainty is conveyed by the error bars which appear to encompass the fit solution.

The best-fit value of $\chi^2$ is 7.46 which has a corresponding p-value of 38% given 7 degrees of freedom, which is within satisfactory limits. The estimated standard deviations for $S$ and $T$ are $1.97 \times 10^{-5}$ and $2.84 \times 10^{-5}$ m$^2$/s, respectively. These are the confidence intervals for the fitting results.

The right panel of Figure 8.1 shows contours of $\chi^2$ on a logarithmic scale for different fit parameters $S$ and $T$. If this had been a linear problem, the contours would have been concentric ellipses. While that is not the case here, the contours are nearly elliptical over a fairly broad range of parameter space surrounding the solution. This is an indication that Levenberg Marquardt should run stably and iterate progressively to solution so long as the initial guess is reasonable.

## 8.5 Regularized nonlinear least squares

In general, nonlinear inverse problems can present with a number of closely-spaced local minima in the vicinity of the global $\chi^2$ minimum. Local minima and plateaued minima can be added to the list of pathologies already present in linear inverse problems which include minima shaped like ridges. In all cases, the solution will not be unique. The problem is also unstable if $J^T J$ is nearly singular. (For linear problems, singularity in $G^T G$ signaled instability) or if the objective function or its first derivatives are not continuous.

As with linear problems, the remedy for nonlinear problems may be regularization. Regularization reduces the space of candidate solutions, simplifies the $\chi^2$ contours, and suppresses fluctuations arising from statistical fluctuations in the data. The price to be paid is bias in the model. As with linear problems, the object is to find the optimum balance between fluctuations and bias.

As has been seen several times now, regularization usually amounts to either minimizing or constraining a combination of penalties, one based on the $\chi^2$ statistic, and the other on some undesirable characteristic of the model. For example,

$$m = \underset{m}{\operatorname{argmin}} \, (G(m)-d)^T C_d^{-1} (G(m)-d) + \alpha^2 \|Lm\|_2^2 \tag{8.26}$$

$$m = \underset{m}{\operatorname{argmin}} \, (G(m)-d)^T C_d^{-1} (G(m)-d) \quad \text{with} \quad \|Lm\|_2^2 = \beta \tag{8.27}$$

$$m = \underset{m}{\operatorname{argmin}} \, \|Lm\|_2^2 \quad \text{with} \quad (G(m)-d)^T C_d^{-1} (G(m)-d) = \gamma \tag{8.28}$$

where $\alpha$, $\beta$, and $\gamma$ are adjustable regularization parameters. In fact, these three formulations are equivalent, and identical results are obtained for suitable choices for $\alpha$, $\beta$, and $\gamma$. The first formulation is weighted damped least squares. Using the method of normal equations, the solution has been shown repeatedly to be:

$$m = \underset{m}{\operatorname{argmin}} \left\| \begin{array}{c} C_d^{-1/2}(G(m)-d) \\ \alpha L m \end{array} \right\|_2^2 \tag{8.29}$$

which suggests a straightforward generalization of Gauss Newton. Eq. (8.29) is the minimization of $n+m$ equations in $m$ unknowns. The first $n$ equations are the same ones as in Gauss Newton, formally called $f$, with the cost function $c = f^T f$. The remaining $m$ equations are new. The new equations can be incorporated in the algorithm simply by augmenting $f$, $c$, and the Jacobian matrix accordingly so as to contain the new equations:

$$K(m) = \begin{pmatrix} J(m) \\ \alpha L \end{pmatrix} \tag{8.30}$$

with $K$ replacing $J$ everywhere in the algorithm. The update equation for the regularized algorithm now reads:

$$K^T(m) K(m)\, dm = -K^T(m) \begin{pmatrix} C_d^{-1/2}(G(m)-d) \\ \alpha L m \end{pmatrix} \tag{8.31}$$

$$= -K^T(m) f(m) \tag{8.32}$$

or, put another a little more transparently,

$$\left(J^T(m)J(m) + \alpha^2 L^T L\right)\, dm = -J^T(m)C_d^{-1/2}\left(G(m) - d\right) - \alpha^2 L^T Lm \tag{8.33}$$

This formula is reminiscent of that for weighted damped linear least squares only with the Jacobian playing the role of the system matrix. It is also reminiscent of the update equation for Levenberg Marquardt, only with the regularization term $\alpha^2 L^T L$ replacing $\lambda I$. The regularization term now performs the damping, rendering the explicit damping term in Levenberg Marquardt unnecessary. (The damping term may be retained if nonetheless if it is computationally expedient to do so, i.e. in order to utilize an existing Levenberg Marquardt code.) The manner in which $\alpha$ is set is the same as in linear problems.

**Algorithm 7** Regularized Levenberg Marquardt

```
1: procedure ITERATE TO CONVERGENCE
2: initialize m_o
3: set α
4:     for k = 0, 1, ... do
5:         dm = −(K^T(m_k)K(m_k))^{−1} K^T(m_k) f(m_k)
6:         m_{k+1} = m_k + dm
7:     end for
8: end procedure
```

## 8.6 Example: gravity anomaly

The gravity anomaly problem was introduced back in chapter 1 and represents a poignant demonstration of the power of regularization. Figure 8.2 presents a solution of a synthetic problem via the algorithm described immediately above. We consider an equal number of data and model values. For the sake of simplicity, both the mass density of the pipe and the gravitational constant are taken to be unity in MKS units. The true model, shown in the upper-left panel of the figure, represents a buried pipe that is higher at the midpoint than at the ends. Synthetic data for the true model are plotted in the upper-right panel. Normally distributed independent noise with a variance of 0.002 has been added to the synthetic data. In the absence of this noise, the Levenberg Marquardt algorithm can reproduce the true model from the data without difficulty. With the noise present, Levenberg Marquardt produces the chaotic model estimate shown in the lower-left panel of the figure.

The performance is obviously unsatisfactory and might discourage an experienced data analyst from proceeding with the problem. What is more, the solution is time consuming, requiring 380 iterations in this case to converge. The unregularized algorithm converged nonetheless, and the model estimate predicts the synthetic data accurately. The problem is that the algorithm has gone to great lengths to reproduce fluctuations in the data that arose from sampling errors rather than from features in the model. The algorithm has "fit the noise" and gone to extraordinary lengths (in the $||Lm||$ sense) to do so. A plot of $\chi^2$ contours for this problem would resemble a cratered landscape (in 40 dimensions).

The lower-right panel in Figure 8.2 shows the model estimate obtained using second-order Tikhonov regularization. This time, a plausible reasonable model estimate has been found, and in only three iterations! Regularization has contracted the solution space in this case, restricting the model estimate to curves that vary smoothly. A contour plot of $c = f^T f$ would look more like ellipses concentric on the solution point.

It should be pointed out that $\chi^2$ for the regularized solution will be larger than for the unregularized solution since the former represents the minimization of something other than $\chi^2$. Regularization has suppressed fluctuations in the model estimate at the expense of bias – bias toward smooth curves in this case. Despite this, error bars for the model estimate in the regularized solution will be smaller than error bars in the unregularized solution. Error bars reflect sensitivity in the model estimate to variations in the data. Regularization suppresses this sensitivity. This does not necessarily mean that the regularized model estimates are more accurate since error bars do not account for biases. What is needed is a way to compute confidence intervals that account for biases and fluctuations. Such a way will be discussed in the following chapter of the text.

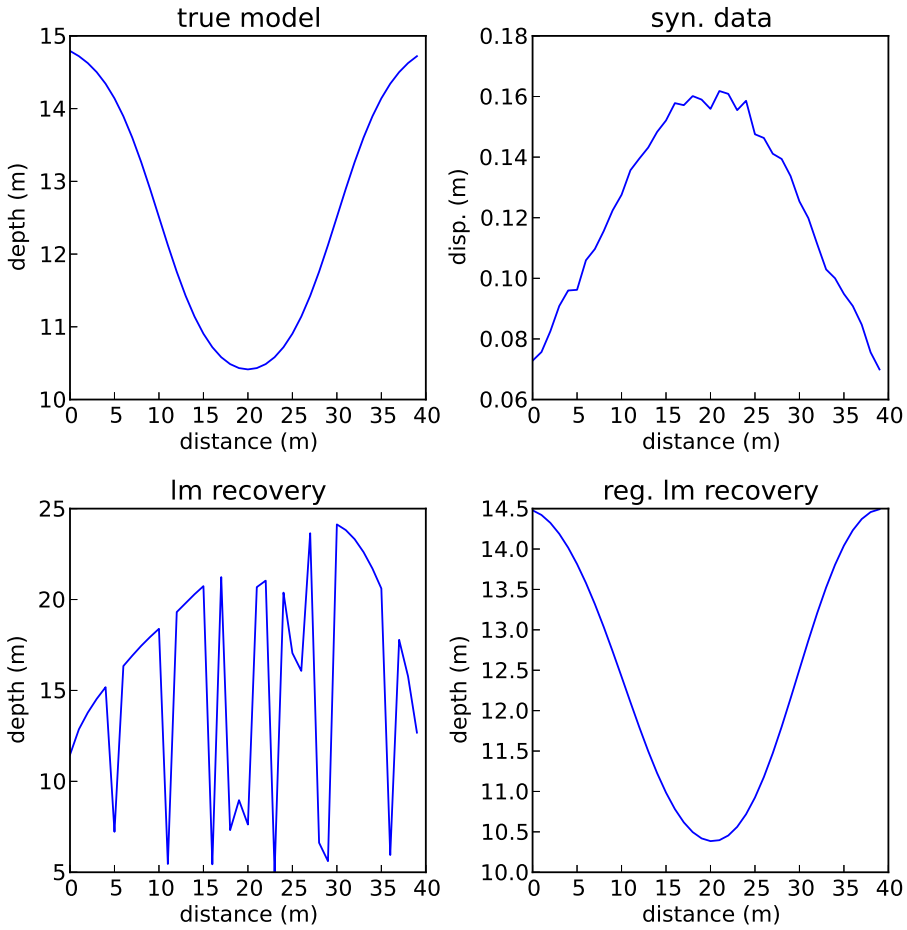


Figure 8.2: LM solution to the gravity anomaly problem.

## 8.7 Nonlinear method of Backus and Gilbert

## 8.8 References

## 8.9 Problems

# Chapter 9

# Bayesian methods and maximum entropy

This chapter revisits the statistical foundations of inverse theory and develops the framework for so-called "Bayesian" inverse methods which rely on Bayesian probability. Bayes' theorem was already derived and discussed back in chapter 3. The theorem is restated here:

$$P(c_i|E,I) = \frac{P(E|c_i)P(c_i|I)}{\sum_{j=1}^{N} P(E|c_j)P(c_j|I)} \tag{9.1}$$

Recall that $P(c_i|I)$ are known as prior probabilities, $P(E|c_i)$ is the transitional probability, and $P(c_i|E,I)$ is the posterior probability. Recall also that bootstrapping is possible with this theorem, meaning that additional prior probabilities can be incorporated as they materialize.

Bayes' theorem transforms probabilities of effects contingent on causes to probabilities of causes evidenced by their effects. This language lies at the root of all inverse problems. It also implies that both the available data and the desired model are random variables. The goal then becomes finding the PDF of the model, its expected value, and/or the model with the greatest probability, the maximum a posterior or MAP solution. This viewpoint may be contrasted with the more widely adopted one where the model is deterministic but the data are stochastic due to additive observing noise. The two viewpoints do not necessarily lead to different results.

At the risk of repeating more material from chapter 3, we can recall the transitional probability for data with additive, normally-distributed sample noise governed by the covariance matrix $C_d$:

$$P(d|m^{\text{est}}) = (2\pi)^{-n/2}|C_d|^{-1/2}e^{-\frac{1}{2}(d-Gm^{\text{est}})^T C_d^{-1}(d-Gm^{\text{est}})} \tag{9.2}$$

Suppose the model is also regarded as being a normally-distributed random variable governed by a covariance matrix $C_m$:

$$P(m^{\text{est}}) = (2\pi)^{-m/2}|C_m|^{-1/2}e^{-\frac{1}{2}(m^{\text{est}}-m_\circ)^T C_m^{-1}(m^{\text{est}}-m_\circ)} \tag{9.3}$$

It can be shown (with some difficulty) that the product of the transitional probability given by (9.2) and the prior probability given by (9.3) is another normal distribution of the form:

$$P(m^{\text{est}}|d) = (2\pi)^{-m/2}|\tilde{C}_m|^{-1/2}e^{-\frac{1}{2}(m^{\text{est}}-\bar{m})^T \tilde{C}_m^{-1}(m^{\text{est}}-\bar{m})} \tag{9.4}$$

with a covariance matrix

$$\tilde{C}_m = (G^T C_d^{-1} G + C_m^{-1})^{-1} \tag{9.5}$$

and with an expected value and a MAP solution (the same in this case) of

$$\hat{m} = m_\circ + (G^T C_d^{-1} G + C_m^{-1})^{-1} G^T C_d^{-1}(d - Gm_\circ) \tag{9.6}$$

We can recognize (9.6) as the weighted damped least squares solution found through a deterministic framework repeatedly throughout this text with (9.5) giving the results of standard error propagation analysis. The weighted damped

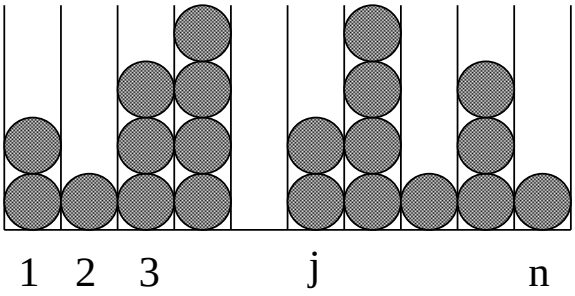


Figure 9.1: Illustration of indistinguishable items sorted into $n$ identical bins.

least squares solution had a Bayesian interpretation all along. Indeed, all inverse methods can be interpreted in a Bayesian framework, and so it is really not correct to regard methods as being Bayesian and non-Bayesian. It is therefore the Bayesian framework that is being investigated here.

The Bayesian interpretation also offers something new – a model PDF. This permits, among other things, the generation of synthetic models for ensemble analysis. The posterior model covariance matrix can be factored using Cholesky factorization as $\tilde{C}_m = R^T R$. Synthetic models can subsequently be generated in a Monte Carlo sense using $m = R^T s + \hat{m}$ where $s$ is a vector of normally-distributed independent random numbers with zero mean and unity standard deviation.

What sets Bayesian statistics truly apart from frequentist statistics is not the use of Bayes' theorem which is applied in any number of contexts without controversy or debate. Rather, Bayesian statistics stand apart from frequentist statistics in defining prior probabilities as something other than frequency distributions. While this approach remains controversial, adoption of the Bayesian framework has become widespread in many fields of study.

## 9.1 Bayesian probability

In the context of the Bayesian framework, prior probability is not bound to the concept of frequency but is something more like plausibility. It derives from prior information and is frequently rooted more in physics than in mathematics. What is needed is a mechanism to convert prior information to prior probability. One of the most powerful tools for accomplishing this is formalized in terms of entropy.

## 9.2 Information theory and Shannon's Entropy

The foundations of contemporary information theory were laid in large part by Claude Shannon working at Bell Laboratories in the 1940s and 50s. Among other things, Shannon was interested in the capacity of a communications channel to carry information. But what is the yardstick for information? The answer would come from work on kinetic theory conducted nearly a century earlier.

The theory for the kinetic theory of gasses was developed by Boltzmann who considered distributions of molecules divided into discrete bins in phase space. The state of a gas can be described in terms of occupation numbers $n_j$, the number of molecules in the $j$th bin. Boltzmann considered the number of ways in which a given distribution or arrangement of molecules could be realized. The situation is illustrated by Figure 9.1. Suppose there are $N$ molecules total. If the ordering of the molecules in the individual bins were significant, then the number would be $N!$. Since the molecules are indistinguishable and and the ordering is not significant, the number is instead

$$p = \frac{N!}{\Pi_{j=1}^{n} n_j!} \tag{9.7}$$

The larger $p$, the greater the number of realizations, and the more "generic" the arrangement. Generic arrangements are probable. Less generic arrangements are less probable, all else being equal.

Eq. (9.7) can be simplified with the help of Stirling's formula for the factorial function which states that $s! \approx \sqrt{2\pi s} s^s e^{-s}$ or that $\ln s! \approx \ln(\sqrt{2\pi s}) + s\ln s - s$. For large values of $s$, the first term can be neglected, giving the familiar formula $\ln s! \approx s\ln s - s$. The number of realizations for a given arrangement (or rather its logarithm) is therefore approximately

$$\begin{aligned}
\ln p &= \ln N! - \ln \Pi_{j=1}^{n} n_j! && (9.8)\\
&\approx N\ln N - N - \sum_{j=1}^{n} (n_j \ln n_j - n_j) && (9.9)\\
&= N\ln N - \sum_{j=1}^{n} n_j \ln n_j && (9.10)\\
&= \sum_{j=1}^{n} n_j (\ln N - \ln n_j) && (9.11)\\
&= -\sum_{j=1}^{n} n_j \ln(n_j/N) && (9.12)\\
&= -N \sum_{j=1}^{n} \frac{n_j}{N} \ln \frac{n_j}{N} && (9.13)
\end{aligned}$$

If the energy associated with each bin $E_k$ is specified, then the energy for a configuration is known and can serve as a constraint. The conservation of the number of molecules is another constraint. The constrained maximization of $p$ is a straightforward optimization problem which yields the most probable value for each $n_j$, which is the Boltzmann distribution law. Averaging the Boltzmann law over the spatial coordinates in phase space gives the Maxwellian velocity distribution function. The partition functions of Gibbs may be constructed using derivative reasoning.

Note that (9.12) and (9.13) are not restricted to problems where $n$ and $N$ are integers. In the event that $n$ is a continuous quantity, $N$ becomes the the integral of $n$ over the domain of interest.

Since distributions with high entropy can come about in a large number of ways, they can be considered to have a greater likelihood of occurrence than low-entropy arrangements, all else being equal. Entropy can therefore form the basis of maximum likelihood inverse methods. Low-entropy solutions are furthermore only to be permitted with justification — with support in the data or in some other prior probability estimate. Such justification must constitute information. Low-entropy arrangements therefore imply the existence of information which needs to be present for such arrangements to be adopted as solutions. Adopting a solution with entropy less than what is demanded by the constraints of the problem is claiming to have information that does not exist. Shannon recognized how entropy is a yardstick for information. Maximum-entropy solutions make the minimum assumptions.

Notice that the logarithm in Shannon's entropy implies that the yardstick can only be applied to positive model quantities. This is a strength for modeling quantities which are necessarily positive such as power or number density since it immediately rules out non-physical solutions. Notice also that entropy favors uniform solutions since the highest entropy solution for any given $N$ is a uniform solution. This is consistent with the regularization schemes discussed in previous chapters. The difference is that entropy is unaware of the order of the $n_j$ and therefore does not explicitly enforce smoothness locally. It is an "edge-preserving" yardstick that does not tend to produce solutions with eroded edges.

With regard to the original information theory problem, Shannon used entropy to show that the channel capacity $C$ in kbps is related to the channel bandwidth $B$ through the formula $C = B\log_2(1 + S/N)$ where $S/N$ is the signal-to-noise ratio on the channel. The surprising result is that the channel capacity can be arbitrarily large so long as the signal-to-noise ratio is sufficiently high. This finding has overtones for imaging methods.

## 9.3 Example - loaded dice

The loaded dice problem illustrates the mechanics of Bayesian inverse problems with prior probability based on Shannon's entropy. The problem is easy to understand, although the solution is not. It is often cited by supporters and critics of Bayesian methods for reasons that will soon be clear.

Consider a normal six-sided die. If fair, every side has an equal frequency of occurrence – namely $f_i = 1/6, i \in 1-6$. The expected numerical value of a roll of a fair die is therefore $\sum_{i=1}^{6} i f_i = 3.5$. If the sample average $x$ based on a large number of rolls is approximately 3.5, then the die may or may not be fair. If the sample average is not approximately 3.5, then the die is not fair.

Suppose only the sample average $x$ is known. What can be said about the individual frequencies $f_i$? It is unclear how to proceed with this problem in a frequentist statistical framework. In a Bayesian framework, the probability distribution with the highest entropy consistent with the constraints can be sought.

The Bayesian MAP solution is found by extremizing the following objective function based on Shannon's entropy:

$$\phi = \sum_{i=1}^{6} f_i \ln(f_i/F) + \lambda \left( \sum_{i=1}^{6} i f_i - x \right) + \Lambda \left( \sum_{i=1}^{6} f_i - 1 \right) \tag{9.14}$$

where $F = \sum_{i=1}^{6} f_i = 1$ is the total of all the frequencies. As usual, the constraints have been introduced with Lagrange multipliers $\lambda$ and $\Lambda$. The first incorporates the sample average $x$, and the second guarantees the normalization condition $F = 1$. Differentiating with respect to the unknowns $f_i$ gives

$$\frac{\partial \phi}{\partial f_i} = \ln f_i + 1 + \lambda i + \Lambda = 0 \tag{9.15}$$

which says that $f_i \propto \exp(-\lambda i)$. The constant of proportionality is set in accordance with the normalization condition, i.e.,

$$f_i = \frac{e^{-\lambda i}}{Z} \tag{9.16}$$

$$Z = \sum_{j=1}^{6} e^{-\lambda j} \tag{9.17}$$

where $Z$ can be seen to playing the role of a partition function. This is no accident, as alluded to in the introduction to the chapter.

At this point, it is customary to rewrite (9.14), replacing the individual frequencies with (9.16) in the entropy term.

$$\phi = \sum_{i=1}^{6} \frac{e^{-\lambda i}}{Z} (-\lambda i - \ln Z) + \lambda \left( \sum_{i=1}^{6} i f_i - x \right) \tag{9.18}$$

$$= -\ln Z - \lambda x \tag{9.19}$$

Notice that the normalization constraint need no longer appear since it is already guaranteed by the form of (9.16). Notice also the simplification that arises in rewriting the objective function. Differentiation with respect to $\lambda$ gives:

$$\frac{\partial \phi}{\partial \lambda} = -\frac{1}{Z} \frac{\partial Z}{\partial \lambda} - x \tag{9.20}$$

$$= -\frac{1}{Z} \sum_{j=1}^{6} (-j) e^{-\lambda j} - x \tag{9.21}$$

$$= 0 \tag{9.22}$$

which gives an equation for the optimum values for the frequencies.

$$x = \frac{\sum_{j=1}^{6} j e^{-\lambda j}}{\sum_{j=1}^{6} e^{-\lambda j}} \tag{9.23}$$

For a given sample average $x$, (9.23) can be solved for $\lambda$ which then specifies $f_i$ through (9.16). The solution will have to be found iteratively, however, as the formula is implicit.

In the case of $x = 3.5$, $\lambda = 0.$ and the frequencies of the different faces of the die coming up are uniform. The most likely conclusion in the Bayesian sense is that the die is fair. In contrast, the case of $x = 4.5$, $\lambda$ = -0.37105, and the most likely frequencies can be calculated to be $f_i = \{0.0543, 0.0788, 0.1142, 0.1654, 0.2398, 0.3475\}$. The die is not fair, and there is a specific frequency distribution which is most likely in the Bayesian sense. This is the MAP estimate.

What has been accomplished? The probabilities thus found clearly satisfy the constraints. They have a certain intuitive appeal, increasing gradually with increasing numbers of dots. How are they better than any other $f_i$ that also satisfy the constraints?

Consider the parable of the dilapidated dice factory that produces all manner of dice with random frequency distributions. Unwilling to invest in the factory, the owner decides instead to keep making faulty dice but to sort the production into different boxes according to their $x$-values which can be measured at the end of the assembly line. Dice from boxes with $x \approx 3.5$ are sold at full value (as premium dice) while dice from other boxed are discounted. Dice from boxes with extreme values of $x$ are melted down and recast. Such are the ways of capitalism!

Every die in every box has its own frequency distribution $f_i$ and therefore its own entropy. The upper limit on the entropy of the dice in any given box is determined by $x$ according to the analysis above. It can be shown that the entropies of the dice in each box will be clustered around that maximum entropy. The dice with entropies closest to the maximum will be most representative of the whole box. They are what we can expect to find in a box labeled by $x$.

## 9.4 Jaynes' concentration theorem

A quantitative treatment of the aforementioned assertion was given by Jaynes who calculated the degree to which possible outcomes are concentrated around the maximum entropy solution. Each die in the scenario described above will have its own frequency distribution $f_i$ which can be regarded as a random variable. Associated with each is a certain entropy $H$, also a random variable. Jaynes showed that entropy is chi-squared distributed, obeying the formula:

$$2N\,\Delta H = \chi^2_k(1-F) \tag{9.24}$$

where $k = n - m - 1$ is the number of degrees of freedom, $n$ is the number of possible outcomes, $m$ is the number of linearly-independent constraints, $N$ is the number of trials (dice in a box), $\Delta H$ is the difference in the entropy from the maximum, and $F$ is the fraction of possible outcomes (dice) with entropy in the range between $H$ and $H - \Delta H$.

Applying the theorem to the loaded dice problem, we take $m = 1$ in view of the fact that the normalization constraint was removed in the formulation of (9.18). The entropy for the $x$=3.5 (4.5) solution is found to be 1.792 (1.614). With $k = 4$, given $N = 1000$, and consulting the chi-squared function at the 95% confidence level ($F = 0.05$), Jaynes that for the $x = 4.5$ case, $\Delta H = 4.74 \times 10^{-3}$. This means that the vast majority of all of the solutions consistent with the constraints (dice with given $x$) have entropies clustered very near the maximum value, i.e.

$$1.609 \leq H \leq 1.614 \tag{9.25}$$

The message is clear: to accept a solution with an entropy significantly less than the maximum entropy is to reject the vast majority of candidate solutions. This course of action in only permissible if there is support for it in the data and/or constraints.

## 9.5 Example - Abel inversion

The Abel transform was introduced in chapter 1 as a means of describing linear convolution-type problems with spherical symmetry. An explicit inverse exists which gives results identical to the least-squares inverse in the discrete case. The problem is poorly conditioned and prone to producing spurious layers. It was tackled again in chapter 5

using SVD and Tikhonov regularization. While the results were satisfactory, regularization had the tendency to reduce the peaks, fill in the gaps, and blend the edges.

The Abel transform relates the spherically-symmetric model quantity of interest $m(r)$ to the corresponding path-integrated measurement $d(y)$ through the integral transform

$$d(y) = 2\int_y^\infty \frac{m(r)r\,dr}{\sqrt{r^2-y^2}} \tag{9.26}$$

which can be discretized using simple collocation to yield the system matrix $G$ relating the model vector $m$ linearly to the data vector $d$ in the usual way. The problem can be solved using maximum entropy. Since entropy is a nonlinear function of the model, this makes the inverse problem a nonlinear one. The added computational burden is great but may be well worth it.

The maximum entropy solution is the extremum of the objective function

$$f_1 = s + \lambda^T(d - Gm) + \Gamma(M - 1^T m) \tag{9.27}$$

where $\lambda$ is a vector of Lagrange multipliers which enforce data congruity, $\Gamma$ is another Lagrange multiplier enforcing the normalization condition, $M = \sum_j m_j$, 1 is a column vector made up of 1s, and $s = \sum_j m_j \ln(m_j/M)$ is the negative entropy or "negentropy". The sums here are over all the elements in the model vector. Differentiating with respect to $m_j$ gives

$$-\ln\frac{m_j}{M} - 1 - \lambda^T G^{[,j]} - \Gamma = 0 \tag{9.28}$$

in which $G^{[,j]}$ stands for the $j$th column of $G$. This can be solved for $m_j$, making use of the normalization condition along the way.

$$m_j = M\frac{e^{-\lambda^T G^{[,j]}}}{Z} \tag{9.29}$$

$$Z = \sum_j e^{-\lambda^T G^{[,j]}} \tag{9.30}$$

with $Z$ once again appearing in the role of a partition function. As with the loaded dice problems (and all maximum entropy problems), the model vector is parametrized by the vector of Lagrange multipliers.

As with the loaded dice problem, it is expedient to substitute this expression for $m_j$ into the original objective function and to reformulate it, adopting a bootstrapping strategy. There is furthermore a fundamental flaw in (9.27) which insists that an exact match between the data and their predictions. In fact, we expect discrepancies arising from sampling errors which can be described by a data covariance matrix $C_d$. The refined objective function is therefore

$$f_2 = s + \lambda^T(d + e - Gm) + \Lambda(e^T C_d^{-1} e - \Sigma) \tag{9.31}$$

$$= \lambda^T(d+e) - M\ln Z + \Lambda(e^T C_d^{-1} e - \Sigma) \tag{9.32}$$

in which (9.29) has been utilized. Here, $\Sigma$ is the prescribed value of the chi-square parameter which is being introduced as another constraint with the help of the Lagrange multiplier $\Lambda$. Different strategies have been advanced for setting $\Sigma$, which will play the role of a regularization parameter in practice. The expected value for $\Sigma$ is the number of data $n$ which seems like a good starting point.

Finally, $f_2$ is differentiated with respect to the Lagrange multipliers and the undetermined error terms.

$$\frac{\partial f_2}{\partial \lambda_j} = d + e - Gm = 0 \tag{9.33}$$

$$\frac{\partial f_2}{\partial e_j} = C_d\lambda + 2\Lambda e = 0 \tag{9.34}$$

$$\frac{\partial f_2}{\partial \Lambda} = e^T C_d^{-1} e - \Sigma = 0 \tag{9.35}$$

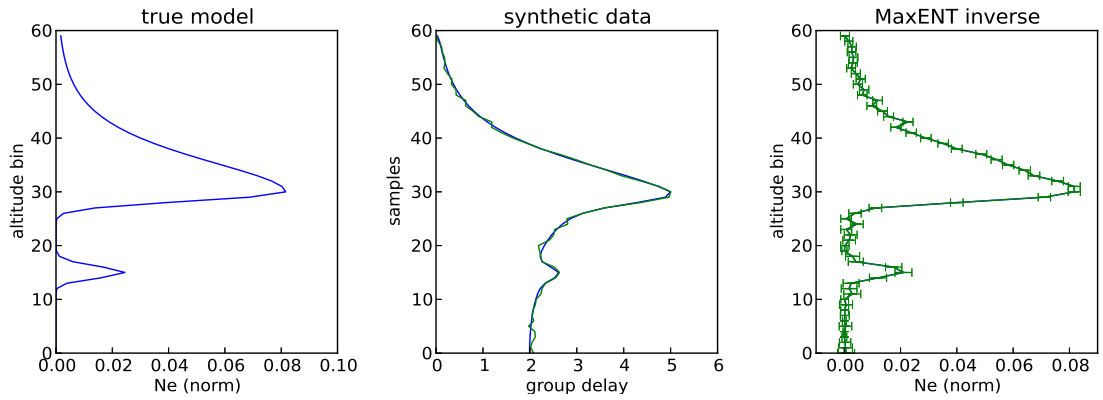


Figure 9.2: Abel transform inversion using maximum entropy method.

which is a coupled system of $2n+1$ nonlinear equations. Eq. (9.33) restates the direct problem. Eq. (9.34) relates the error terms to the Lagrange multipliers in $\lambda$. Eq. (9.35) restates the error constraint. Substituting (9.34) into (9.35) gives a relationship between $\Lambda$ and the other Lagrange multipliers:

$$\Lambda^2 = \frac{\lambda^T C_d \lambda}{4\Sigma} \tag{9.36}$$

All together, there are just $n$ equations in $n$ unknowns, the $\lambda$ vector, with $e$ and $\Lambda$ being derived terms. The system of equations can be solved using Newton's method or a variant given a data vector, a setting for $\Sigma$, and an initial guess for the $\lambda$ vector. The best initial guess is one that is consistent with a uniform model and a small $\Lambda$. Note that not only the equations but also their derivatives are available analytically. Specifically, if the $n$ equations being solved are taken to have the form

$$f = d - \frac{C_d \lambda}{2\Lambda} - Gm = 0 \tag{9.37}$$

with $\Lambda$ given by (9.36) above and $m$ by (9.29), then

$$\frac{\partial f}{\partial \lambda} = -\frac{C_d}{2\Lambda} + \frac{1}{8\Lambda^3 \Sigma} C_d \lambda \lambda^T C_d - G\frac{\partial m}{\partial \lambda} \tag{9.38}$$

where the terms in $\partial m / \partial \lambda$ are most easily expressed in indexed form:

$$\frac{\partial m_j}{\partial \lambda_k} = -G_{kj} m_j + \frac{m_j}{M} (Gm)_k \tag{9.39}$$

where no sum is implied by $G_{kj} m_j$ and where $(Gm)_k$ refers to the $k$th element of the vector $Gm$. While it is not always necessary to specify the Jacobian analytically in solving (9.37), it is usually beneficial.

A potential problem with this method is that $M$ must somehow be specified in order to solve the problem. In the case of synthetic data, $M$ is of course known. In real-world applications, $M$ could be estimated through a search process. It could also be estimated from the solution already found through Tikhonov regularization. In some problems, $M$ is automatically defined by the problem. An example is the Fourier transform, where the area under the curve in the frequency domain is the value at zero argument in the time domain.

Results of the maximum entropy inversion of the Abel transform are shown in Figure 9.2 for the same synthetic data examined earlier in the text. The method was implemented using a simple root-finding scheme and incorporated analytic calculations of the Jacobian matrix. The results are superior to those obtained using Tikhonov regularization in two important respects. First, the model estimate is nowhere negative. This is critical in view of the fact that the physical quantity in question can never be negative anywhere. Making use of this prior information implicitly leads to more accurate data reconstruction and prevents misunderstanding.

Second, the recovered data are sharper than what could be obtained using Tikhonov regularization. The peaks are narrower, and the valleys are deeper. While the entropy prior prefers uniform over nonuniform results, it is not

sensitive to the ordering of the model values and does not suppress steep edges. The edge preserving feature is a hallmark of the method.

The error bars shown in the rightmost panel of Figure 9.2 were calculated according to the prescription given below.

## 9.6 Error propagation

The maximum entropy method amounts to finding the minimum of an objective function that combines the negentropy with the chi-squared parameter, i.e.

$$\phi = \alpha s + \frac{1}{2} e^T C_d^{-1} e \tag{9.40}$$

This is equivalent to maximizing the posterior probability

$$p(m|d) \propto e^{-\alpha s} e^{-\frac{1}{2}(Gm-d)^T C_d^{-1}(Gm-d)} \tag{9.41}$$

where we identify $p(m|d) \propto \exp(-\phi)$. This obviously has the form of the product of prior and transitional probabilities in a Bayesian framework. Here, $\alpha$ is a weight which serves the same role as the $\Sigma$ factor used to constrain the error norm in the inverse Abel transform problem considered previously. Their precise relationship is spelled out by (9.31) which shows that $\alpha$ is equivalent to $\Lambda^{-1}$.

Let us expand the objective function in the neighborhood of its minimum, $m^*$:

$$\phi(m) \quad \approx \quad \phi(m^*) + \delta m^T \nabla\phi(m^*) + \frac{1}{2}\delta m^T H(m^*)\delta m + \cdots$$

where $\nabla m$ is the gradient vector, $H$ is the Hessian matrix,

$$H = \frac{\partial^2 \phi(m)}{\partial m \, \partial m} \tag{9.42}$$

and $\delta m$ is the deviation from the minimum: $\delta m = m - m^*$. Near the minimum, the gradient term vanishes, and the posterior probability is governed by the quadratic term:

$$p(m|d) \propto e^{-\frac{1}{2}\delta m^T H(m^*)\delta m} \tag{9.43}$$

(where the $\phi(m^*)$ term has been absorbed in the constant of proportionality). Now, (9.43) has the form of a multivariate normal probability distribution function with the Hessian playing the role of the inverse model error covariance matrix. Model fluctuations arising from small fluctuations in the data will be normally distributed. Evidently, we may estimate $\tilde{C}_m^{-1} \approx H(m^*)$.

By the rules of vector calculus, the contribution to the Hessian function from the weighted residual terms in (9.40) is simply $G^T C_d^{-1} G$. The contribution from the entropy term in (9.40) can be found by noting:

$$\frac{\partial s}{\partial m_j} \quad = \quad \ln \frac{m_j}{M} - 1 \tag{9.44}$$

$$\frac{\partial^2 s}{\partial m_j \partial m_k} \quad = \quad \frac{1}{m_k}\delta(j,k) \tag{9.45}$$

with $\delta(j,k)$ being the Kronecker delta. Combining this with the contribution from the transitional probability term gives

$$\tilde{C}_m^{-1} \approx G^T C_d^{-1} G + \alpha \, \mathrm{Diag}(m^{-1}) \tag{9.46}$$

This is the prescription for propagating errors through the maximum entropy method (compare (9.5)). Notice that the effect of the entropy term is to reduce the fluctuations in the model output arising from observation errors. The reduction comes at the expense of added bias which cannot be addressed through this analysis. When used for appropriate problems, entropy may introduce less bias that other forms of regularization discussed in this text. Notice also that the errors introduced my maximum entropy are mutually uncorrelated. This is not true of the other techniques considered as a rule.

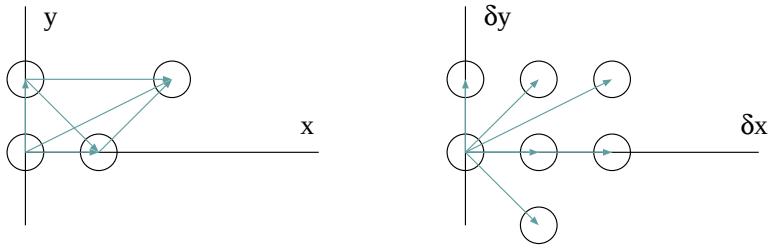


Figure 9.3: Geometry for a simple aperture-synthesis imaging experiment involving just four receivers.

## 9.7 Example - spectral estimation and radio imaging

Chapter 1 introduced the inverse problem associated with aperture-synthesis radio and radar imaging. This is the method by which images of distant radio sources are formed from signals received with antennas located at different locations on the ground. Central to that problem is the integral transform

$$V(\delta x, \delta y; k) \quad \approx \quad \iint d\eta d\xi \, B(\eta, \xi) \exp\left(ik(\eta \delta x + \xi \delta y)\right) \tag{9.47}$$

which relates the visibility function $V$ to the brightness distribution $B$. Samples of the visibility are formed by correlating the signals from antennas on a plane with cardinal displacements $\delta x$ and $\delta y$. The brightness distribution is the distribution of radiated power versus bearing in the sky, where $\eta$ and $\xi$ are the direction cosines with respect to the $\delta x$ and $\delta y$ axes, respectively. The brightness distribution is the desired image.

Note that the integral transform in (9.47) is just a Fourier transform. Statistical inverse methods are usually required to invert (9.47) in radio imaging problems because the data are incomplete and sparsely sampled in $(\delta x, \delta y)$ as opposed to being available on a regular grid. The aperture synthesis imaging problem is therefore the problem of spectral analysis of sparsely and irregularly sampled data. It is actually a subset of the more general Fourier transform problem in that $B$ is a real function, implying that $V$ is a Hermitian function.

Fortunately, little about the algorithm described above for inverting the Abel transform needs to be modified for application to the imaging problem. Each complex-valued visibility measurement can be regarded as two real-valued measurements – one for the real part, and one for the imaginary part. Separating (9.47) into its real and imaginary parts gives the direct model for either part

$$\Re(V(\delta x, \delta y; k)) \quad = \quad \iint d\eta d\xi \, B(\eta, \xi) \cos\left(k(\eta \delta x + \xi \delta y)\right) \tag{9.48}$$

$$\Im(V(\delta x, \delta y; k)) \quad = \quad \iint d\eta d\xi \, B(\eta, \xi) \sin\left(k(\eta \delta x + \xi \delta y)\right) \tag{9.49}$$

Consequently, the formulation for the real-valued Abel transform problem still applies. In particular, equations (9.33) – (9.36) apply, as do the auxiliary equations pertaining to their derivatives. The system matrix $G$ is composed of terms that are either the cosine or the sine of $ik(\eta \delta x + \xi \delta y)$. The visibility data are discrete by nature, and the brightness distribution can be discretized in a regular grid using simple collocation.

A simple aperture-synthesis imaging setup is illustrated in Figure 9.3. The left panel shows the spatial distribution of four antennas in the $(x, y)$ plane. Given $n$ receivers, visibilities can be measured with at most $n(n-1)/2$ non-redundant, nonzero spacings (or baselines). The right panel of Figure 9.3 shows the six finite baselines for this configuration (along with the zero baseline measurement). The zero-baseline measurement is important since it gives a way to normalize the visibility measurements. When the visibilities are so normalized, the value for $M$ in the maximum-entropy formulation is identically unity. All together, the experiment shown produces 14 real-valued data (the two associated with the zero baseline being trivial).

For the sake of simplicity, we take the baselines to have lengths that are integer multiples of unity and also take $k = 2\pi$. Further take the brightness distribution to be confined to the region $-1/4 < \eta < 1/4$ and $-1/2 < \xi < 1/2$ which can be represented by the available baselines and measurements.

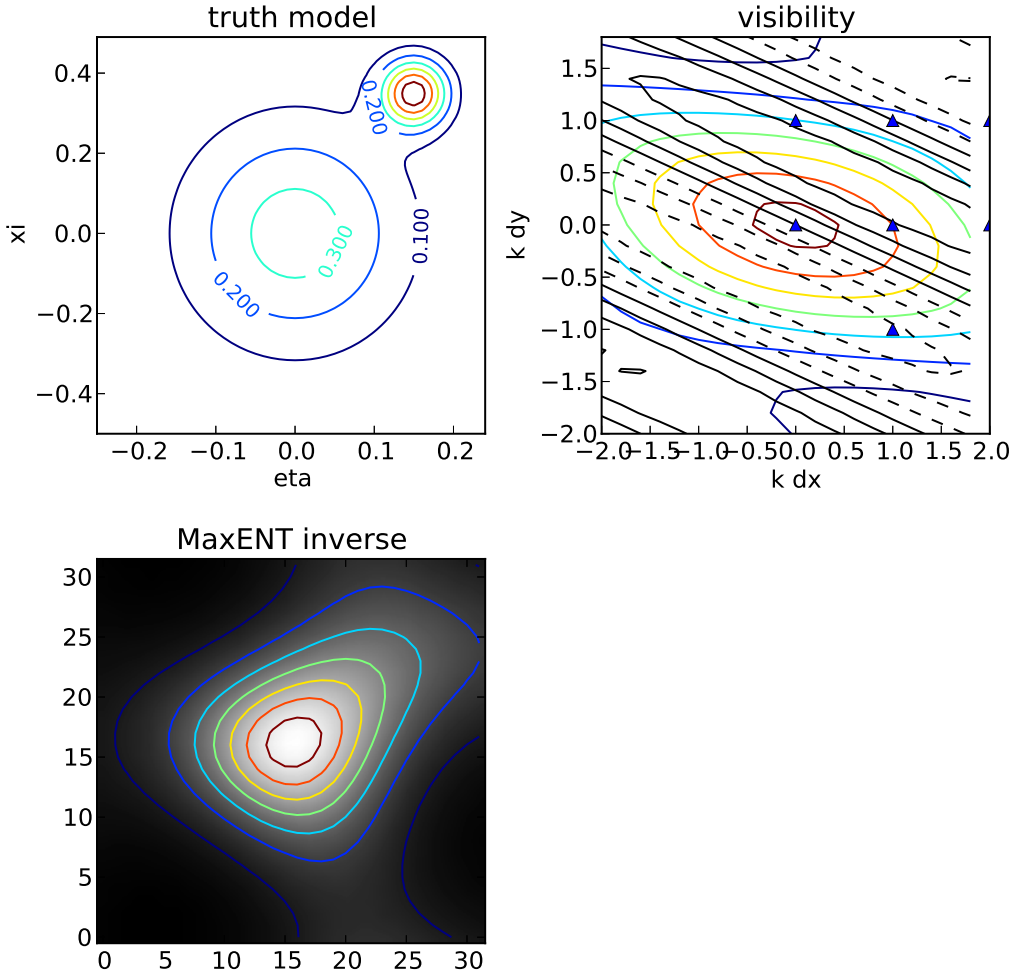


Figure 9.4: MaxENT recovery of sparsely-sampled visibility data. The top-left panel shows the idealized model image to be recovered. It contains a wide feature (Gaussian ellipsoid) in the center with a narrower but stronger feature in the first quadrant. The top-right panel shows the associated visibility function. The real part is represented by colored contours and the imaginary part by black contours. Dashed contours indicate negative values. The baselines for sampling the visibility are indicated by plotter symbols. Note that the visibility function extends well beyond the region of visibility space being sampled. Finally, the lower-left panel shows the MaxENT solution. The image spans the region bounded by $-1/4 < \eta < 1/4$ (horizontal axis) and $-1/2 < \xi < 1/2$ (vertical axis).

Figure 9.4 shows an example inversion based on the visibility samples indicated in Figure 9.3. The truth model for the brightness distribution is composed of a pair of Gaussian ellipsoids. The visibility distribution is the Fourier transform of the brightness distribution and can be computed exactly using numerical integration. Plotter symbols indicate where the visibility function is sampled. Independent normally-distributed noise has been added to the visibility samples at the level of $\sigma = 0.01$. The initial guesses for the Lagrange multipliers are set according to the prescription given for the Abel transform problem above.

The bottom-left panel of Figure 9.4 shows the MaxENT inverse solution. The image has been generated on a $32 \times 32$ uniform grid in this example. This griding is arbitrary; any granularity is possible, although the resolution of the features in the image will be limited by the lengths of the baselines and the data collected on them. The resolution is also limited by the signal-to-noise ratio in accordance with the Shannon-Hartley theorem in information theory.

On the basis of very few samples, the algorithm is able to recover the general shape of the truth model (although much of the detail is absent). Distortion in the recovered image results from the paucity of samples and the fact that the samples are nonuniform in visibility space. With just a few more samples, the accuracy of the image improves significantly. Nonetheless, the performance demonstrated here is impressive. The MaxENT algorithm limited the vast space of candidate images consistent with the visibility samples and the error norm to the space of images that are positive everywhere. The maximum-entropy solution is furthermore the one most representative of that class of solutions.

## 9.8 Super-resolution

## 9.9 References

## 9.10 Problems

# Part IV: Specialized applications

# Chapter 10

# Geometric optics

## 10.1 References

## 10.2 Problems

# Chapter 11

# Inverse scattering

## 11.1 References

## 11.2 Problems

# Chapter 12

# Total variation regularization and compressive sensing

## 12.1 References

## 12.2 Problems

# Chapter 13

# Stochastic optimization

The iterative methods discussed so far are apt to have difficulty in situations where the objective function is very complicated, presenting a torturous landscape, replete with local minima. While a grid search might help identify the neighborhood around the global minimum, the computational cost could be prohibitive, particularly in a high-dimensional parameter spaces. Contrary to intuition, such circumstances can often be handled expeditiously with the insertion of random elements in the optimization scheme. Randomized iterative schemes can be useful in exploring vast parameter spaces and can also help overcome convergence problems associated with deterministic deficiencies in the data or the system.

The key feature of random iterations is that they permit occasional movements "uphill" in the objective function, thus allowing a more complete search of parameter space. (Uphill moves are not considered by any of the algorithms considered so far.) Even when the global minimum cannot be found, random iteration schemes can often find nearly optimal solutions that are more satisfactory than what could be uncovered through deterministic means.

The task becomes finding an appropriate means of introducing randomness. For clues for doing this, analysts turn to nature (including human nature) for examples. Musical improvisation, bacteriologic adaptation, cultural evolution, super-organism migration, speech acquisition, and signal processing have provided inspiration for optimization methods.

For example, ant colonies solve the problem of foraging for food simply and efficiently and without centralized planning. When a foraging ant leaves the colony, it either wanders randomly in search of food or follows a pheromone trail when it encounters one. The stronger the pheromones, the more likely the ant will follow the trail. When an ant locates food, it secures what it can carry and continues its journey. As before, it either wanders randomly or follows a pheromone trail if it encounters one. The difference is that, wherever it goes, the food-laden ant now leaves behind a trail of its own. The simple strategy ultimately leads ants to and from food along nearly straight lines. When a food source is depleted, the pheromone trails dissipate, and the optimization problem is renewed.

Flocks of birds and schools of fish likewise emerge to solve optimization problems without centralized organization. The nature of the collaboration is that each individual is aware both of its own current best solution of the problem (of where to be for some purpose) and the current best solution of the group. By changing velocity so as to continually alternate between the two positions, individuals (and hence the group) explore a broad range of solutions. Random overshoot seems to be an important component of the optimization process.

An important aspect of random searches is expressed by the "no free lunch theorem" which states that no particular search stratagey outperforms a purely random search on all problems. Different problems are best solved by different methods. Two widely used methods based on annealing in metals and evolution in biology are discussed in detail below.

## 13.1 Monte Carlo Markov chains

## 13.2 Simulated annealing: Metropolis Hastings algorithm

Annealing is the process of crystal formation undergone by metals as they cool from liquids to solids. As the metal cools, crystals form, and the free energy decreases. It has long been known that cooling at different rates produces different terminal free energies. The crystal lattice in the metal is somehow able to approach global minimum more effectively when the metal is cooled slowly. Different heating and cooling schemes appear to be the secrets underlying occasional appearances of surprisingly high-quality low-carbon steel in ancient societies.

When the metal is warm, crystals can form and reform quasi-adiabatically. The configuration changes, with the free energy rising and falling, moving in fits and starts with only a general tendency to move downhill. As the metal cools, the atoms become less mobile, and crystal reformation becomes more restricted to free energy reductions. Eventually, a near global minimum can be reached this way. In contrast, rapid cooling "freezes" the metal configuration in some arrangement which is not close to the global free energy minimum.

In a stochastic optimization problem, the objective function plays the role of the free energy in the metal. The "temperature" becomes an abstraction which represents the ease of uphill movement in terms of the objective function. The algorithm begins with a high temperature that gradually decreases according to some schedule. Candidate changes in the state parameters of the system are generated randomly and considered for adoption. Candidates are accepted and rejected randomly according to criteria that depend on the temperature. Ultimately, the algorithm terminates when the temperature reaches a floor. Provisions for restarting the algorithm based on its performance can be made.

The Metropolis algorithm is a simulated annealing algorithm with the following features:

**Candidate generation** Candidates for the state vector $m$ can be generated by almost any means. Usually, a new candidate is generated by moving some distance $\delta m$ away from the current state estimate. The displacement can be determined entirely randomly or through an algorithm with a random element. If the state vector represents an ordered list, new candidates can be generated through small permutations of the current list. Small, progressive movements allow for the exploration of the neighborhood around a state. A persistent random walk ultimately permits exploration of large regions of state space.

**Candidate acceptance** Whether or not a candidate state replaces the current state estimate is also determined randomly and depends on the relative costs or "energies" of both, as measured by the objective function, along with the current temperature. The barrier to acceptance should be low at the start of the run, becoming high at the end. Past states are usually not stored, although very promising states can be retained and reevaluated later. History is preserved through the temperature.

Denote the probability of accepting a candidate as $P(e, e', T)$ where $e$ and $e'$ are the objective function evaluations or energies of the current and candidate state and $T$ is the temperature. The acceptance probability $P$ should be small (large) if $\Delta e = e' - e > 0$ ($< 0$). How small or large depends on the temperature. For example,

$$P(e, e', T) = e^{-\Delta e/T} \tag{13.1}$$

could serve as the acceptance probability and could be compared with a random number in the interval (0,1). Here, the temperature $T$ could be selected initially to afford finite probability even to significant energy increases. A value of $P$ larger than the random number would trigger acceptance.

**Cooling schedule** The temperature determines the degree to which energy increases are considered for adoption. Over time, the temperature $T$ should be made to diminish until the point comes when energy increases are ruled out. The temperature can be made to decrease gradually and fractionally with every $k$th iteration, for example. In addition, the temperature can also be made to decrease further with every $l$th candidate adoption, since successful adoptions suggest concrete progress.

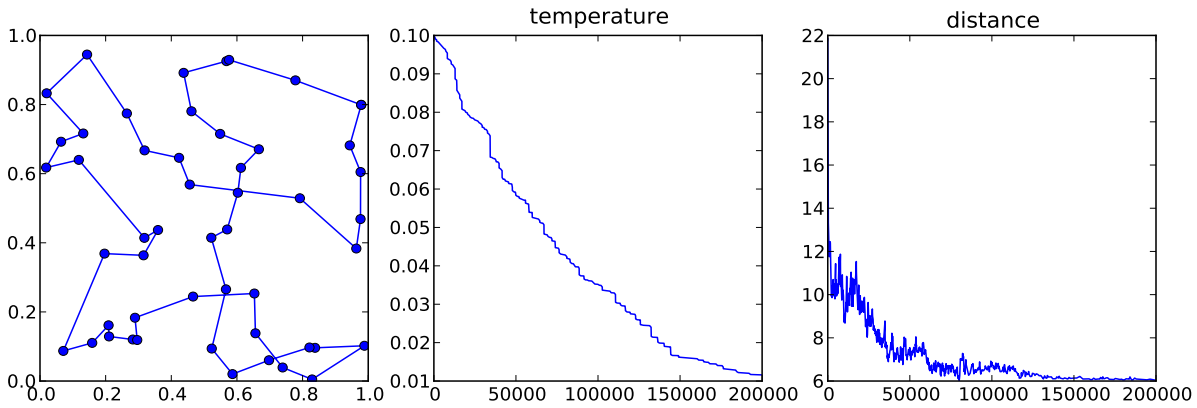


Figure 13.1: Simulated annealing solution to the traveling salesperson problem.

**Stopping and restarting** Termination of the algorithm can be triggered by three factors – the completion of a preset number of iterations, the attainment of a preset terminal energy level, and/or cooling to a preset terminal temperature. If progress is slow, the current state could be set to the best state achieved previously.

**Algorithm 8** Metropolis algorithm

```
1: procedure ITERATE TO CONVERGENCE
2: initialize m
3:     for k = 0 through k_max do
4:         T ← temperature (k, l)
5:         Consider random move m' ← m + δm
6:         If P(e(m), e'(m'), T) ≥ random(0,1) m ← m'
7:     end for
8: output m
9: end procedure
```

## 13.3 Example: traveling salesperson

Few demonstration problems have been so widely explored as that of the fabled traveling salesperson. In this problem, a salesperson must visit each of a number of different cities exactly once, the journey concluding at the same city in which it began. The problem is to minimize the total distance traveled, the sum of the Euclidean distances between the cities on the route. The cities and their locations can be considered an ordered list, and the optimization problem is one of selecting the right order. There are $N!$ ways to arrange $N$ entries on a list. Since the starting point is irrelevant, there are actually $(N-1)!$ distinct routes. Since the routes going forward and in reverse are equivalent, there are actually $(N-1)!/2$ distinct possible solutions to this problem. If $N$=50, that means $\sim 3\times 10^{62}$.

A near-optimum solution can be obtained readily using the Metropolis algorithm. Results are shown for a 50-city example run in Figure 13.1. The city positions were generated randomly on a square grid. Here, the objective function is the total length of the closed path that joins them. For this algorithm, candidate lists were generated simply by swapping two cities in the current list. The acceptance penalty used is the exponential function discussed above. A total of 200,000 iterations were performed. The temperature, initially set to 0.1, was decreased by 0.1% after 100 candidate acceptances or 1000 rejections, whichever came first.

The figure shows a gradual decrease in temperature with occasional plummets. Attendant with decreasing temperature is the decreasing total path length. Increases appear from time to time but occur less frequently as the temperature decreases. The algorithm "bottomed out" at a total distance of about 6 units after about 150,000 iterations when uphill

climbs became all but impossible. Further progress appears unlikely.

The route found by the algorithm is clearly suboptimal, as evidenced by the closed loops seen in the left panel of Figure 13.1. The overall distance could be reduced by unlooping the three loops in the solution. While this appears to be a straightforward alteration, it is actually unlikely to occur spontaneously in view of the fact that the direction of travel would have to reverse everywhere on the loop. Unlooping the loops requires drastic and coordinated rearrangements of the list, something simulated annealing is simply incapable of doing. Adding a procedure to the basic algorithm to detect and unloop loops would be a straightforward augmentation of the basic algorithm, however.

## 13.4 Genetic algorithms

Another powerful natural model for stochastic optimization is biological evolution. Genetic algorithms based on evolutionary models utilize randomness in their iterations much as in simulated annealing. An important difference is that in genetic algorithms, not one candidate solution but rather a whole population of candidates is maintained and cultivated. There is also no concept of temperature in genetic algorithms. History is preserved instead through the population itself which retains and passes along characteristics inherited from past generations.

A candidate individual in a genetic algorithm can be just about anything: a floating point or integer number (composed of bits), a vector of numbers, a string of characters, or a list of abstract objects. Individuals are like chromosomes, and their elements are their genes. The only requirement is that swapping genes in an individual should produce another individual, i.e., the groups should be closed.

Genetic algorithms also require an objective function for determining the fitness of an individual. The fitness influences the likelihood of an individual progressing to the next generation and producing offspring. Selections are random, so high fitness no more guarantees survival than low fitness rules it out.

A population is made up of a fixed number of individuals. It is through the diversity of the individuals that a broad range of state space can be explored. At every iteration, the population is reduced through a selection process which "culls the herd" It is then restored with the production of offspring with genes copied from the survivors and blended or rearranged. This process may be likened to reproduction but is called "crossover." Finally, the population is modified through the introduction of random errors or mutations. Each of these elements (plus criteria for terminating the algorithm) is developed in more detail below.

**Selection** Individuals are selected for survival or termination in a random process that involves their individual fitness (think natural selection). One way of selecting individuals is through a roulette wheel approach. All of the members of the population are afforded sectors in a roulette wheel, and selections are made by spinning the wheel until enough survivors are identified. The size of the sector occupied by an individual can be made to be proportional to its fitness, i.e. $p_i = f_i / \sum f_i$, where $f_i$ and $p_i$ are the fitness and probability of the $i$th individual, respectively. Alternatively, the sizes can be set according to a formula based on the rank of the individual fitnesses. In this scheme, every individual competes with every other individual on each spin of the wheel.

Another means of selection pits pairs or small groups of individuals against one another. The pairings or groups are chosen randomly, and the probability of victory is controlled by the relative fitnesses of the candidates. Tournaments are run until the desired number of survivors is reached. Each candidate may or may not be guaranteed the opportunity to compete in a tournament.

A potential problem with selection is that the best individuals may not survive to the next generation. A strategy for guaranteeing that one or more of the most fit individuals necessarily survives is called "elitism." The idea is simply to exempt the elite individuals from the selection process, conveying them automatically to the next generation. The pitfall with elitism is that it may perpetuate a good solution at the expense of allowing a better one, i.e., restrict the solution to a local minimum.

**Crossover** The population is next restored to its original strength through reproduction or crossover. Here, new individuals (offspring) are created by copying genes from the existing ones (parents). One method for doing this is to randomly identify pairs of individuals and to generate new pairs by splicing their chromosomes together at a random point. Symbolically, an example could be $(abc, a'b'c') \to (ab'c', a'bc)$ where the splice has been made between the first and second gene in this case. The concept can be generalized to numbers of parents and offspring greater than two. Surviving individuals can be guaranteed to produce offspring or not.

In this way, individuals which are lists of items can readily undergo crossover In the event that the individuals are integers, digits can be treated like genes. In the event that the individuals are continuous quantities (e.g. floating point numbers), recombination can be fractional, i.e. $(x, y) \to (fx + (1-f)y, (1-f)x + fy)$.

**Mutation** Finally, mutations are introduced randomly in an attempt to further broaden the range of configuration space being explored. Mutations are errors which can be introduced to the offspring or the offspring and the parents (but maybe not the elites). Mutations can be random changes to random genes, random gene swapping, or changes at the bit level. The probability of a mutation can be assigned based on normal or uniform PDFs.

**Termination** Termination normally occurs either when a suitably fit individual has been produced (to serve as the solution), when the best fitness in the population has reached a plateau, or when a fixed number of generations have come and gone. Restarting may be considered if the solution is unsatisfactory. As genetic algorithms are usually reserved for the most complex problems with the most diabolical objective functions, establishing strict, rigorous termination criteria can be challenging.

Genetic algorithms tend to be computationally expensive, particularly when the objective function is complicated, and slow to converge. They are not purposeful and not internally constructed to reach a goal. They have a tendency to become trapped in local minima and to return near-optimal rather than optimal solutions. This problem is particularly acute in problems with simple right-and-wrong answers in which the fitness presents no hill to be climbed. Performance varies with variations in the selection, crossover, and mutation rules and rates, parameters which must be set in the absence of rigorous rules or guidelines. Finally, setting confidence limits on the solution is impractical with genetic algorithms. Nevertheless, genetic algorithms have found favor in a class of problems in which grid searches and exhaustive searches are simply intractable.

**Algorithm 9** Genetic algorithm

```
1:  procedure ITERATE TO CONVERGENCE
2:  initialize population
3:      for k = 0 through k_max do
4:          perform selection
5:          perform crossover
6:          introduce mutations
7:          exit loop if fitness threshold reached
8:      end for
9:  output most fit individual
10: end procedure
```

## 13.5 Example: Sodoku puzzle

An illustrative problem for genetic algorithms and their limitations is a Sodoku puzzle. The object in Sodoku is to fill a 9-by-9 grid of digits numbering 1–9 in such a way that no row, column, or 3-by-3 subgrid contains no repeated digit. Some of the digits are pre-supplied as hints and make the solution unique. An simple example problem with its solution is given in Table 10.1.

An individual in this case is a list of 81 digits. The fitness of an individual is the total number of unique digits in

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | **6** | 1 | **3** | 7 | 5 | **8** | 9 | **4** |
| **5** | **3** | **7** | 8 | **9** | 4 | 1 | 6 | 2 |
| 9 | **4** | 8 | 2 | 1 | **6** | **3** | 5 | **7** |
| 6 | **9** | 4 | 7 | **5** | **1** | **2** | **3** | **8** |
| 8 | 2 | 5 | 9 | 4 | 3 | 6 | 7 | 1 |
| **7** | **1** | **3** | **6** | **2** | 8 | 9 | **4** | 5 |
| **3** | 5 | **6** | **4** | 8 | 2 | 7 | **1** | 9 |
| 4 | 8 | 9 | 1 | **6** | 7 | **5** | **2** | **3** |
| **1** | 7 | **2** | 5 | 3 | **9** | 4 | **8** | 6 |

Table 13.1: Solved Sodoku puzzle example. Boldface numbers are clues.

each row, column, and 3-by-3 subgrid which could be as large as $9^4$. In this implementation, the fitness is this total minus the fitness of the initial guess which is constructed by filling the rows of the table with random arrangements of all the digits from 1 to 9. The hints are included by perpetually overwriting the individual in those entries where hints are given.

We consider a population of 20 individuals. Selection of 10 survivors is by simple roulette wheel with wedge sizes proportional to fitness. The two most fit individuals are exempted from selection. Crossover is used to generate 10 additional individuals by pairing off the survivors and splicing their chromosomes at a random point. (Every survivor is therefore guaranteed to reproduce.) Finally, mutations occur rarely and involve swapping two digits somewhere within an individual. In this implementation, the elites are not exempted from mutation.

For this puzzle, the algorithm found the correct solution in fewer than 75,000 generations. Some tuning of the mutation rate was required to achieve this performance, however, and the algorithm is not robust. Given other puzzles, the algorithm frequently finds a solution with two errors. Solutions with a few errors can in fact be very different from correct solutions, but the objective function does not provide a gently sloping contour leading to improvement. In fact, genetic algorithms are not well suited to the Sodoku problem and the demand for the correct answer among many almost correct ones. Fast, direct methods of solution exist for Sodoku, and even a mere human player can find a solution more quickly as a rule. Nevertheless, the algorithm was simple to code and found the answer to the problem given.

## 13.6 References

## 13.7 Problems

# Chapter 14

# Linear estimation theory and the Kalman filter

Inverse methods represent a corner of the broader field if estimation theory. Estimation theory considers not only hidden states in a system but also the dynamic evolution of those states, possibly in real time. One of the main tools in estimation theory is the Kalman filter. While a complete discussion of Kalman filtering is beyond the scope of this text, it turns out that many of the principles involved have already been covered. The most important connections are highlighted here.

The Kalman filter estimates the hidden state of a system by combining multiple observations with a dynamical model of the state over time. Allowances are made for observation noise and other inaccuracies in the problem specification. The algorithm is a two-step, predictor-corrector process. The dynamical model produces an estimate of the state in the prediction step. In the correction step, the estimate is modified on the basis of data which are incorporated through weighted averages. In this sense, Kalman filtering is an example of model/data fusion. The filter has minimal storage requirements since the current state estimate is based entirely on the previous state and the most recent data (i.e. the filter is recursive). Kalman filters are often implemented in real-time and used for time-critical applications like navigation and control. While the basic filter is linear, the extended Kalman filter can be used for nonlinear problems.

## 14.1 Linear estimation theory

The Kalman filter assumes that the state of the system $x$ at timestep $k$ evolves from the state at the previous timestep according to the rule

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k u_k + w_k \tag{14.1}$$

$$z_k = H_k x_k + v_k \tag{14.2}$$

where $F$ is the state transition model, $B$ is a control input model, $u$ is the control vector, and $w$ is process noise which is assumed to conform to a multivariate normal distribution with covariance $Q_k$. The control vector provides an avenue for influencing the evolution of the state, e.g. for vehicle trajectory control.

While the state of the system is hidden, observations of a derived parameter $z$ are available. The observation model $H$ maps the state into the observation which is contaminated by observation noise $v$. That noise is assumed to be normally distributed with covariance $R_k$.

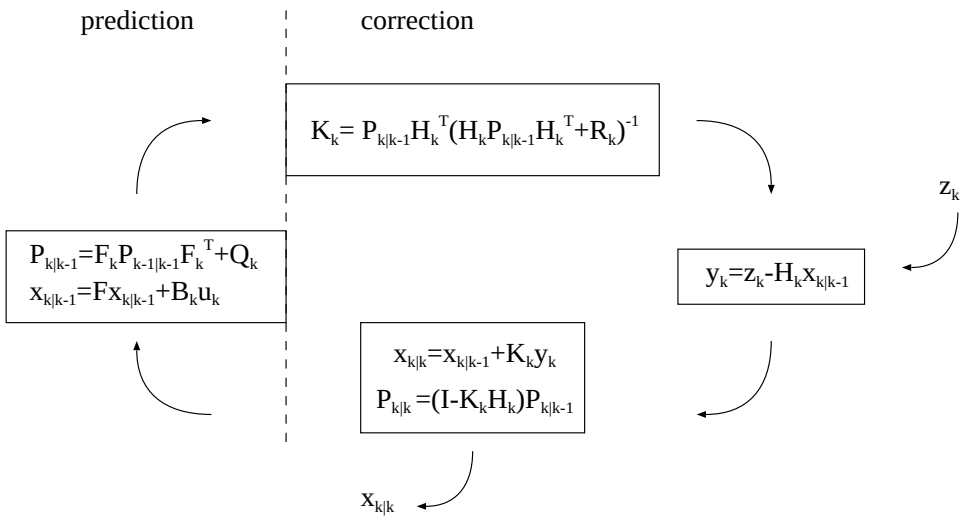


Figure 14.1: Diagram of the Kalman filter algorithm (see text).

The prediction step is carried out according to

$$\hat{x}_{k|k-1} = F_k\hat{x}_{k-1|k-1} + B_ku_k \tag{14.3}$$
$$P_{k|k-1} = F_kP_{k-1|k-1}F_k^T + Q_k \tag{14.4}$$

where the hats denote an estimate. Also predicted is the a priori covariance of the state estimate $P$.

The correction or update step is carried out with

$$\hat{y}_k = z_k - H_k\hat{x}_{k|k-1} \tag{14.5}$$
$$S_k = H_kP_{k|k-1}H_k^T + R_k \tag{14.6}$$
$$K_k = P_{k|k-1}H_k^TS_k^{-1} \tag{14.7}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\hat{y}_k \tag{14.8}$$
$$P_{k|k} = (I - K_kH_k)P_{k|k-1} \tag{14.9}$$

Here, $\hat{y}$ is called the innovation or measurement residual and $S_k$ is called the innovation or residual covariance. The $K$ term is the Kalman gain and gives the prescription for weighting the observations in the correction process. The corrected state estimate is $\hat{x}_{k|k}$, and the posterior estimate covariance is given by $P_{k|k}$.

It is not necessary to derive the components of the Kalman filter here because they should already be familiar by now and can be found elsewhere in the pages of this text. For example, combining (14.8), (14.7), (14.6), and (14.5) gives:

$$\hat{x}_{k|k} - \hat{x}_{k|k-1} = P_{k|k-1}H_k^T\left(H_kP_{k|k-1}H_k^T + R_k\right)^{-1}(z_k - H_k\hat{x}_{k|k-1}) \tag{14.10}$$

which is identical to the weighted-damped least squares estimator as described by (2.39):

$$m^{\text{est}} - m_\circ = C_mG^T\left(GC_mG^T + C_d\right)^{-1}(d - Gm_\circ) \tag{14.11}$$

We need only make the identifications $x \to m$, $P \to C_m$, $H \to G$, $R \to C_d$, and $z \to d$. The Kalman gain term itself is just $\tilde{G}$ in the parlance of inverse problems. Data fusion in the Kalman filter is accomplished by means of weighted damped least squares.

Another component of the filter to be accounted for is (14.4) which is just the standard formula for error propagation that has appeared in this text many times. Finally, (14.9) has the form

$$\tilde{C}_m = (I - \tilde{G}G)C_m \tag{14.12}$$

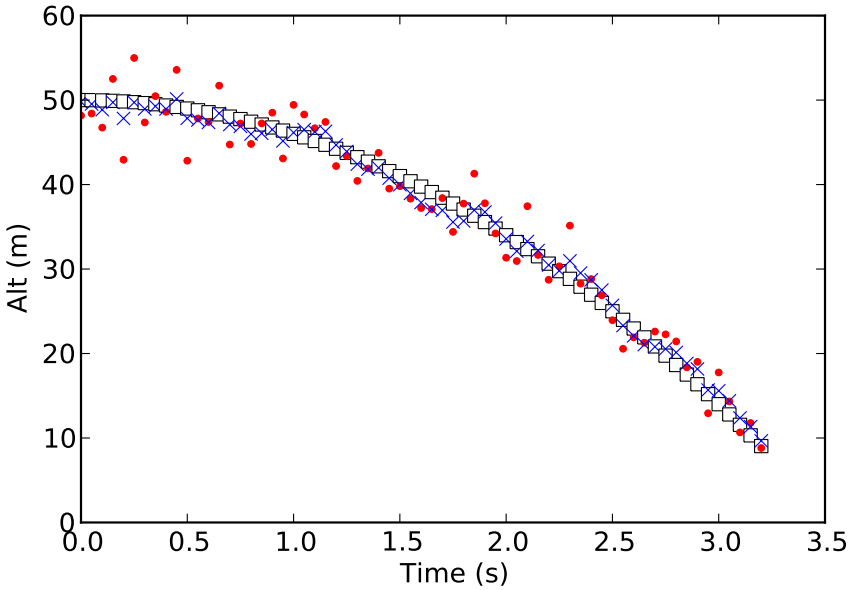


Figure 14.2: Falling-body dynamics modeled with a Kalman filter. The open squares are the true trajectory, the red circles are synthetic data, and the blue crosses are the Kalman-filter estimate. Note that the state-transition matrix assumes a higher rate of acceleration than the data support. The Kalman-filter estimate is much closer to the truth model than either the state-transition model or the data alone.

where we recognize $R_m \equiv \tilde{G}G$. The model covariances in the Kalman filter are evidently updated by a factor based on the spread of the model resolution matrix. It governs how data assimilation reduces estimation errors, capturing a recurring theme in the text.

### 14.1.1 Example: ballistic trajectory

A simple example of the basic Kalman filter is illustrated in Figure 14.2 which presents simulated dynamics of a falling body. The body begins at rest at an elevation of 50 m and accelerates downward thereafter under the influence of gravity at an anticipated rate of -9.8 m/s$^2$. The state vector for this problem has three components — the vertical position (altitude), velocity, and acceleration.

The state transition matrix for this case is

$$F = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{14.13}$$

where $\Delta t$ is the interval between adjacent timesteps. For this example, we take $\Delta t$ = 0.05 s and consider 65 timesteps, enough time nominally for the object to fall to the ground.

The only observable in this example is the altitude of the body, and those observations will be regarded as being relatively imprecise. The observation noise is such that the scalar $R$ has an autocovariance of 9.0 in MKS units. The process noise meanwhile is such that $Q$ is diagonal with autocovariances of 0.1 in MKS units.

For this problem, we impose the condition that gravity is offset by an updraft and that the actual acceleration of the body is just 8.0 m/s$^2$. This makes the state transition model inaccurate. No input control is considered.

The results in Figure 14.2 illustrate the performance of the Kalman filter. The filter output (blue crosses) is quite close to the truth model (open squares). The data are spread far and wide around either and would, by themselves,

provide a poor estimate of the true altitude at any given timestep. The state transition model, meanwhile, would have predicted impact with the ground by the end of the run in the absence of data, noise notwithstanding. The utility of the Kalman filter in real-world systems dynamics problems lies in its ability to assimilate data into physics-based iterative models effectively.

The results in Figure 14.2 are similar to many other examples in this text where a truth model was used to generate synthetic data that were subsequently incorporated into a model estimate. Every other example in the text was conducted in "batch mode" such that all the data were available simultaneously for the calculation of the model estimate. With a Kalman filter, only past data are available when the current model estimate is calculated. What is more, historical data need not be stored for long, as the current model estimate is based solely on the most current data and the previous model estimate. Storage requirements are minimal, and complex metric calculations are avoided. This makes Kalman filters suitable for applications involving huge datasets, e.g. in weather forecasting.

Since the observables could be anything linearly related to the state vector, Kalman filters also lend themselves naturally to problems involving data fusion. Someone with one watch knows what time it is. Someone with two watches does not – without a Kalman filter.

## 14.2 Extended Kalman filter

Finally, the basic Kalman filter can readily be extended to include nonlinear state transition and observation models $f$ and $h$. The derivation can be brief since the generalization required resides on ground already covered by this text. Consider the following:

$$\hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}) + w_{k-1} \tag{14.14}$$

$$z_k = h(x_k) + v_k \tag{14.15}$$

where $f$ and $h$ are differentiable functions and $w$ and $v$ represent process and observing noise as before with covariance $Q$ and $R$, respectively, and $u$ is the control vector. The state transition and measurement prediction can be calculated using (14.14) and (14.15) in a straightforward way. Additionally, wherever the matrices $F$ and $H$ appear in triple products in the basic Kalman filter formalism, the Jacobian matrices $J_f$ and $J_h$ should now be substituted:

$$J_f = \frac{\partial f}{\partial x}\Big|_{\hat{x}_{k-1|k-1}u_{k-1}} \tag{14.16}$$

$$J_h = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k|k-1}} \tag{14.17}$$

giving the prescription for a first-order extended Kalman filter. Higher-order filters can also be defined, as can filters where the process and observation noise appear as arguments in the nonlinear functions.

## 14.3 References

## 14.4 Problems

# Chapter 15

# Appendix

## 15.1 Vector and matrix norms

A crucial aspect of inverse methods is the necessity of a metric or metrics for evaluating the quality of a candidate model solution. Among the most useful metrics are vector and matrix norms. Both topics are briefly developed below. While the following discussion is written in terms of discrete quantities, it can be generalized to include continuous function spaces.

A vector space is a set of objects (vectors) accompanied by two operators, vector addition and scalar multiplication, each satisfying certain conditions. Vector addition combines two vectors to form a third. Scalar multiplication combines a vector with a scalar to form another vector. The required conditions on the operators are that:

1. vector addition be communative
2. vector addition be associative
3. bector addition have an identity (the zero vector)
4. scalar multiplication has an indenty (unity)
5. vector addition have an inverse (the vector minus)
6. scalar multiplication over vector addition be distributive
7. vector addition over scalar multiplicatin be distributive
8. successive scalar multiplications are compatible

an example of a vector space is Euclidean space, $\mathbb{R}^3$. Vector spaces with dimensionality greater than 3 reside in a Hilbert space.

If a set of vectors in a vector space are linearly independent and span the vector space, they form a basis for the space. Linearly independence requires that no vector can be expressed as a linear combination of the others. The set of vectors spans the space if all other vectors can be written as a linear combination of the set.

Inverse methods require a metric corresponding in some sense to the size of a vector. A natural definition is the generalization of the Euclidean lengths in $\mathbb{R}^3$. Given a vector $x$, the norm of the vector is a real number with the following attributes:

1. $||x|| > 0 \; \forall x \neq 0$

2. $||cx|| = |c|||x||,\ c \in \mathbb{R}$

3. $||x_1 + x_2|| \leq ||x_1|| + ||x_2||$

The first of these requires that sizes be positive and the second that size scales with lengths. The third is the triangle inequality from Euclidean geometry. Any function that has these attributes is a vector norm.

Returning to the example of $\mathbb{R}^3$, note that the Euclidean length of a vector can be expressed in terms of the dot product, viz. $||x|| = \sqrt{x \cdot x}$. Inner product spaces are vector spaces in which an inner product is defined. More generally then, the length of a vector in an inner product space can be expressed as $||x|| = \sqrt{\{(x,x)\}}$, $(x,x)$ being the inner product. A frequently-used result in inner-product spaces is the Schwarz inequality which can be stated as

$$(x_1, x_2)^2 \leq (x_1, x_1)(x_2, x_2) \tag{15.1}$$

Note that the Schwarz inequality can be used to prove the triangle inequality in inner-product vector spaces.

One of the most commonly-used norms in $\mathbb{R}^n$ is the L$-p$ norm which is defined by

$$||x||_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \quad p \geq 1 \tag{15.2}$$

of which the most common choices are L-1 and L-2. This norm can also be evaluated at $p = \infty$ or L-$\infty$ in which case the norm evaluates to the largest single component of the vector.

We are also interested in defining the norm of a matrix in a manner which generalizes the more intuitive definition of the vector norm. The subordinate or induced matrix norm is defined in terms of the vector norm by:

$$||A|| = \max_{||x|| \neq 0} \left( \frac{||Ax||}{||x||} \right) \tag{15.3}$$

in which $x$, $Ax \in \mathbb{R}^n$. By this definition, the norm of $A$ is the largest of all possible vectors $Ax$, normalized to $x$.

It can be shown that the subordinate matrix norm has all of the attributes of the vector norm with two additional ones, making five:

1. $||A|| > 0,\ A \neq 0$

2. $||cA|| = |c|||A||,\ c \in \mathbb{R}$

3. $||A_1 + A_2|| \leq ||A_1|| + ||A_2||$

4. $||Ax|| \leq ||A|||x||$

5. $||A_1 A_2|| \leq ||A_1||||A_2||$

How the matrix norm (properly denoted $||A||_p$) evaluates depends on the vector norm $||x||_p$ to which it is subordinate. Evaluating any but the L-1 and L-$\infty$ norms is generally nontrivial. Note also that different vector norms can be used for the numerator and denominator of (15.3), giving rise to expressions of the form $||A||_{\alpha,\beta}$.

Another kind of matrix norm in regular use is the so-called spectral norm which is identical to the subordinate norm in the special case of $p = 2$. This is the largest singular value of $A$, i.e., the square root of the largest eigenvalue of the matrix $A^T A$:

$$||A||_2 = \sqrt{\lambda_{\max}(A^T A)} \tag{15.4}$$

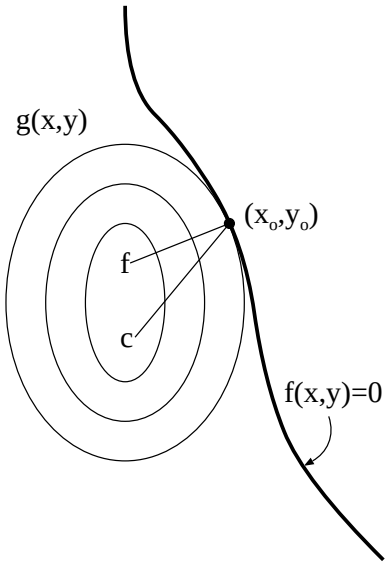


Figure 15.1: Illustration of constrained optimization.

This formulation is often more amenable to analysis than the subordinate formulation.

Finally, so-called entrywise norms are norms that treat matrices of size $n \times m$ as vectors of length $nm$, i.e.

$$\|A\|_p = \left( \sum_{i=1}^{n} \sum_{j=1}^{m} |A_{ij}|^2 \right)^{1/p} \tag{15.5}$$

This is a different defintion from the subordinate norm and one that is generally easier to evaluate. The special case of $p = 2$ is called the Frobenius norm and is suitable for some applications described in the text.

## 15.2 Constrained optimization, inequality constraints

Constrained optimization can easily be understood with the help of a classic fable (see Figure 15.1). A farmhand (denoted 'f') needs to milk a cow (denoted 'c') with dispatch. Before the cow can be milked, the farmhand's pail must be cleaned in a nearby river. The problem is to find the shortest route from the farmhand to the river to the cow. The objective function is the total distance traveled which is to be minimized. The constraint is that the intervening point must lie on the river.

Denote the point on the river $(x_\circ, y_\circ)$. The course of the river satisfies a constraint equation $f(x,y)$=0. The objective function, the total distance traveled, is $g(x,y)$. In this case, contours of constant $g$ are ellipses with foci at 'f' and 'c'. The solution to the problem is on the smallest ellipse that intercepts the river. At the intercept point, the contour $g(x,y)$ =min. is by definition tangent to the curve $f(x,y) = 0$. This implies that the gradients of the two functions are parallel at the solution point or that $\nabla g(x_\circ, y_\circ) \propto \nabla f(x_\circ, y_\circ)$.

If the constant of proportionality is called $\lambda$, then the solution point is the minimum of a new objective function $L$:

$$L(x,y) \quad = \quad g(x,y) - \lambda f(x,y) \tag{15.6}$$

with the solution $(x_\circ, y_\circ)$ corresponding to the zero of the gradient of $L$ with respect to the coordinates and also $\lambda$, the Lagrange multiplier. Note that $\partial L / \partial \lambda = 0$ just returns the constraint equation to be satisfied. The objective function $L$ is sometimes called the Lagrangian function because of the central role it plays in variational mechanics. Multiple constraint equations can be incorporated in the optimization problem using multiple Lagrange multipliers (so long as they are not mutually inconsistent).

If the fable did not contain the constraint of the river, any point on the line joining the points 'f' and 'c' would constitute a minimum of the function $g(x,y)$, making the problem under determined in this case. Here, the addition of the constraint equation removed the ambiguity. A brute-force solution to the problem might have involved using the constraint to eliminate one independent variable from the optimization problem. This approach, which could be impractical, was avoided by the introduction of the Lagrange multiplier.

Just how all of this works can better be appreciated by returning to the original equations. At the solution point, the original objective function satisfies

$$dg = 0 = \frac{\partial g}{\partial x}dx + \frac{\partial g}{\partial y}dy \tag{15.7}$$

Since $dx$ and $dy$ are independent, the solution may be found by solving $\partial g/\partial x = 0$ and $\partial g/\partial y$ separately. With the imposition of the constraint $f(x,y) = 0$, $dx$ and $dy$ are no longer independent but are coupled by

$$df = 0 = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy \tag{15.8}$$

As an alternative to eliminating one of the independent variables, the two equations are added fractionally through the incorporation of the Lagrange multiplier:

$$\left(\frac{\partial g}{\partial x} - \lambda\frac{\partial f}{\partial x}\right)dx + \left(\frac{\partial g}{\partial y} - \lambda\frac{\partial f}{\partial y}\right)dy = 0 \tag{15.9}$$

Now it is always possible to set $\lambda$ so as to make the second term in parentheses in (15.9) zero identically for any $dy$. This decouples the first and second terms which can be solved independently, along with the constraint equation.

Suffice it to write that the aforementioned reasoning applies not only to the fable but to objective functions and optimization problems generally. It applies to linear and nonlinear, continuous and discrete problems.

Sometimes, constraints are expressed as inequalities rather than equalities, i.e. $f(x,y) > 0$. While such constraints may appear at first to be fundamentally more difficult to incorporate in optimization problems than equality constraints, the approach is essentially similar.

Referring again to the fable, in the event that $f(x,y) > 0$ in the region to the left of the river, the constraint would not affect the outcome of the minimization problem and is said to be inactive or slack. If the $f(x,y) > 0$ region is to the right of the river, however, the constraint is said to be active or binding and the solution shifts back to $(x_\circ, y_\circ)$, the solution for the equality constraint. Both situations can be treated through the minimization of (15.6). If the constraint is inactive, then the Lagrange multiplier $\lambda$ will be zero. If the constraint is active, then the solution will fall on the line of constraint, and $f(x,y)$ will be zero. Both conditions are covered by adding to the problem statement the additional condition

$$\lambda f(x,y) \quad = \quad 0 \tag{15.10}$$

which gives the prescription for handling inequality conditions. Note that in the case of active constraints, the sign of $\lambda$ indicates whether $\nabla f$ and $\nabla g$ are parallel or antiparallel and therefore whether $(x_\circ, y_\circ)$ corresponds to a maximum or a minimum.

The following simple example is instructive. Consider the energy of a particle of mass $m$ confined to a box with a volume $V = abc$:

$$E = \frac{h^2}{8m}\left(\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2}\right) \tag{15.11}$$

where $h$ is Planck's constant. The dimensions of the box of constant volume that minimizes the energy can be found by minimizing the objective function

$$L = E - \lambda(abc - V) \tag{15.12}$$

Differentiating with respect to $a$, $b$, and $c$ in turn and comparing the results reveals the condition

$$\frac{1}{a^2} = \frac{1}{b^2} = \frac{1}{c^2} \tag{15.13}$$

In view of the constraint, the energy is minimized when the box is a cube with $V = a^3 = b^3 = c^3$.