

Chapter 2

Naïve reconstructions and inverse crimes

This book is about developing computational solution methods for real-life inverse problems. The design of reconstruction algorithms is best done by first testing the code extensively with simulated data because every new aspect of the code can be systematically tested. Working directly with measured data may lead to very hard debugging problems, as the source of difficulties can be hard to track.

What happens if proper simulation of errors is neglected? For example, using the same computational grid for the data simulation and reconstruction sometimes results in perfect reconstructions from noise-free data. Such a situation is not realistic and is referred to as an *inverse crime*. Excellent inversion results may be obtained, but these are not representative of any realistic inverse problem, since noise is present in any experimental setting. Such studies are inconclusive at best since robustness against modeling and measurement errors is not tested.

In this chapter, we will introduce these concepts in the context of the three guiding examples in Part I: deconvolution, the backward heat equation, and X-ray tomography.

2.1 Convolution

Linear convolution is a useful process for modeling a variety of practical measurements. *Deconvolution*, the corresponding inverse problem, is related to many engineering problems such as removing unwanted echoes from sound recordings or sharpening a misfocused photograph.

One-dimensional deconvolution will serve as a basic example throughout Part I of the book. Two-dimensional deconvolution is a project topic in Section 10.

2.1.1 Continuum model for one-dimensional convolution

We build a computational model for one-dimensional convolution with periodic boundary conditions. We consider 1-periodic functions $f : \mathbb{R} \rightarrow \mathbb{R}$ satisfying $f(x) = f(x + n)$ with any integer $n \in \mathbb{Z}$. Essentially the function f is defined on an interval of length 1 such as $[0, 1]$ or $[-\frac{1}{2}, \frac{1}{2}]$ with the endpoints identified; another way of thinking about this is to consider $f(x)$ defined on a circle with radius $(2\pi)^{-1}$ and x being the arc length variable.

The reason for considering periodic functions is that we can avoid some technicalities related to boundary conditions that would obscure the main message about ill-posedness. Also, the Fourier transform and the wavelet transform are easily defined and implemented in the periodic setting.

The continuum measurement model concerns a 1-periodic signal $f : \mathbb{R} \rightarrow \mathbb{R}$ blurred by a 1-periodic *point spread function* ψ . Other common names for the point spread function include *device function*, *impulse response*, *blurring kernel*, *convolution kernel*, and *transfer function*.

Let us first construct the point spread function using a building block ψ_0 defined in the interval $[-a, a] \subset \mathbb{R}$ with some constant $0 < a < 1/2$:

$$\psi_0(x) = C_a(x+a)^2(x-a)^2 \quad \text{for } -a \leq x \leq a, \quad (2.1)$$

where the constant $C_a := (\int_{-a}^a (x+a)^2(x-a)^2 dx)^{-1}$ is chosen to enforce the following normalization:

$$\int_{-a}^a \psi_0(x) dx = 1. \quad (2.2)$$

The periodic point spread function is defined by copying $\psi_0(x)$ to every interval $[n-a, n+a]$ with $n \in \mathbb{Z}$ and setting $\psi(x)$ to zero outside those intervals. The resulting ψ is a non-negative and even function:

$$\psi(x) \geq 0 \quad \text{and} \quad \psi(x) = \psi(-x) \quad \text{for all } x \in \mathbb{R}. \quad (2.3)$$

See Figure 2.1 for a plot of the point spread function with $a = 0.04$.

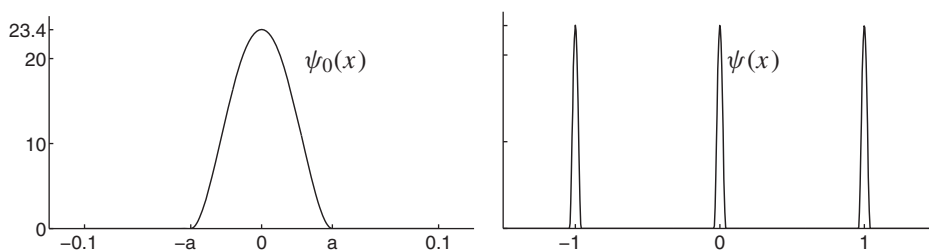


Figure 2.1. Point spread function according to (2.4) with $a = 0.04$ for one-dimensional convolution. Left: the continuously differentiable building block $\psi_0(x)$ used for constructing the periodic point spread function. Right: the periodic point spread function $\psi(x)$.

We remark that instead of (2.2) one often requires $\int_{-a}^a \psi_0(x)^2 dx = 1$. However, we prefer (2.2) since then constant functions remain unchanged in convolution with ψ ; this will be convenient below when we compare plots of reconstructions to the plot of the true signal by showing them in the same figure.

Definition 2.1.1. *The continuum model of convolution, or blurring, is given by the following integral:*

$$(\psi * f)(x) = \int_{-a}^a \psi(x') f(x - x') dx'. \quad (2.4)$$

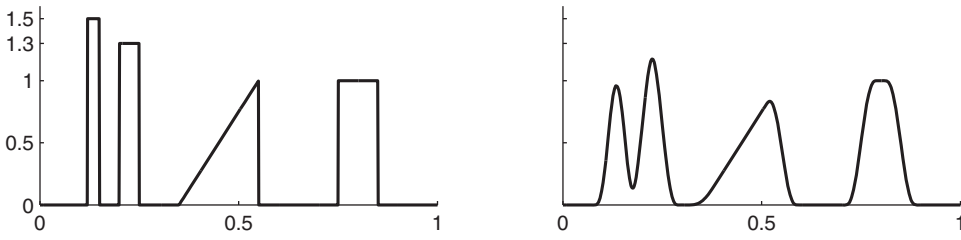


Figure 2.2. Effect of convolution on a piecewise continuous function. Left: target function $f(x)$. Right: the function $(\psi * f)(x)$.

An example of the effect of convolution with the point spread function is found in Figure 2.2. The smoothing effect of the convolution is evident, and motivates the terminology blurring kernel for the point spread function.

Note that formula (2.4) is not of the form (1.1) since the left-hand side is not a k -dimensional vector. However, suppose the function f is defined on an interval $[b, b+1]$, and assume that we have a device that measures the values of the convolution function $(\psi * f)(x)$ at a collection of k equally spaced points $\tilde{x}_1 = b, \tilde{x}_2 = b + \frac{1}{k}, \tilde{x}_3 = b + \frac{2}{k}, \dots, \tilde{x}_k = b + \frac{k-1}{k}$ and define

$$\mathbf{m} := [(\psi * f)(\tilde{x}_1), (\psi * f)(\tilde{x}_2), \dots, (\psi * f)(\tilde{x}_k)]^T \in \mathbb{R}^k. \quad (2.5)$$

Then $\mathcal{A}f = \mathbf{m}$ is of the form (1.1).

2.1.2 Discrete convolution model

Next we need to discretize the continuum model to arrive at a finite-dimensional measurement model of the form (1.3). Define

$$x_j = b + \frac{j-1}{n} \quad \text{for } j = 1, 2, \dots, n; \quad (2.6)$$

then the 1-periodic real-valued function $f(x)$ is represented by a vector $\mathbf{f}$ containing values at the grid points:

$$\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n]^T = [f(x_1), f(x_2), \dots, f(x_n)]^T \in \mathbb{R}^n. \quad (2.7)$$

Furthermore, denote $\Delta x := x_2 - x_1 = 1/n$.

We can approximate the integral appearing in (2.4) by numerical quadrature. For any reasonably well-behaved function $g : [b, b+1] \rightarrow \mathbb{R}$ we have

$$\int_b^{b+1} g(x) dx \approx \Delta x \sum_{j=1}^n g(x_j), \quad (2.8)$$

the approximation becoming better as n increases.

For convenience, let us take $k = n$ and measure the convolution at the same points (2.6) as where the unknown function f is sampled. This is not necessary in general, but

it will lead to a square-shaped matrix A , making it easy to illustrate naïve reconstructions and inverse crimes.

Let us construct an $n \times n$ matrix A so that $A\mathbf{f} \in \mathbb{R}^k$ approximates $\mathcal{A}f$ defined by (2.4). We define a discrete point spread function denoted by

$$\mathbf{p} = [\mathbf{p}_{-v}, \mathbf{p}_{-v+1}, \dots, \mathbf{p}_{-1}, \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{v-1}, \mathbf{p}_v]^T$$

as follows. Recall that $\psi_0(x) \equiv 0$ for $|x| > a > 0$. Take $v > 0$ to be the smallest integer satisfying the inequality $(v+1)\Delta x > a$ and set

$$\tilde{\mathbf{p}}_j = \psi_0(j\Delta x) \quad \text{for } j = -v, \dots, v.$$

For example, with $a = 0.04$ as in Figure 2.1 and $n = 64$, we get $v = 2$. By (2.8) the normalization condition (2.2) almost holds: $\Delta x \sum_{j=-v}^v \tilde{\mathbf{p}}_j \approx 1$. However, in practice it is a good idea to normalize the discrete point spread function explicitly by the formula

$$\mathbf{p} = \left(\Delta x \sum_{j=-v}^v \tilde{\mathbf{p}}_j \right)^{-1} \tilde{\mathbf{p}}; \quad (2.9)$$

then it follows that

$$\Delta x \sum_{j=-v}^v \mathbf{p}_j = 1. \quad (2.10)$$

Now

$$\begin{aligned} \int_{-a}^a \psi(x') f(x_j - x') dx' &\approx \Delta x \sum_{\ell=-v}^v \psi(x_\ell) f(x_j - x_\ell) \\ &\approx \Delta x \sum_{\ell=-v}^v \mathbf{p}_\ell \mathbf{f}_{j-\ell}. \end{aligned}$$

Hence discrete convolution is defined by the formula

$$(\mathbf{p} * \mathbf{f})_j = \sum_{\ell=-v}^v \mathbf{p}_\ell \mathbf{f}_{j-\ell}, \quad (2.11)$$

where $\mathbf{f}_{j-\ell}$ is defined using periodic boundary conditions for the cases $j-\ell < 1$ and $j-\ell > n$. Then

$$\Delta x (\mathbf{p} * \mathbf{f}) \approx \mathcal{A}f, \quad (2.12)$$

and we define the measurement vector $\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_k]^T$ by

$$\mathbf{m}_j = \Delta x (\mathbf{p} * \mathbf{f})_j + \varepsilon_j. \quad (2.13)$$

We would like to write formula (2.13) using a matrix A so that we would arrive at the desired model (1.3). To this end, set

$$\begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_k \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kn} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{bmatrix}.$$

The answer is to build a circulant matrix having the elements of $\mathbf{p}$ appearing systematically on every row of A .

Let us illustrate the structure of the convolution matrix A by an example in the case $n = 64$. As observed above, if $a = 0.04$, then $\nu = 2$, and the point spread function takes the form $p = [p_{-2} \ p_{-1} \ p_0 \ p_1 \ p_2]^T$. According to (2.11) we have

$$\begin{aligned} (\mathbf{p} * \mathbf{f})_1 &= \mathbf{p}_0 \mathbf{f}_1 + \mathbf{p}_{-1} \mathbf{f}_2 + \mathbf{p}_{-2} \mathbf{f}_3 + \mathbf{p}_2 \mathbf{f}_{n-1} + \mathbf{p}_1 \mathbf{f}_n, \\ (\mathbf{p} * \mathbf{f})_2 &= \mathbf{p}_1 \mathbf{f}_1 + \mathbf{p}_0 \mathbf{f}_2 + \mathbf{p}_{-1} \mathbf{f}_3 + \mathbf{p}_{-2} \mathbf{f}_4 + \mathbf{p}_2 \mathbf{f}_n, \\ (\mathbf{p} * \mathbf{f})_3 &= \mathbf{p}_2 \mathbf{f}_1 + \mathbf{p}_1 \mathbf{f}_2 + \mathbf{p}_0 \mathbf{f}_3 + \mathbf{p}_{-1} \mathbf{f}_4 + \mathbf{p}_{-2} \mathbf{f}_5, \\ &\vdots \\ (\mathbf{p} * \mathbf{f})_n &= \mathbf{p}_{-1} \mathbf{f}_1 + \mathbf{p}_{-2} \mathbf{f}_2 + \mathbf{p}_2 \mathbf{f}_{n-2} + \mathbf{p}_1 \mathbf{f}_{n-1} + \mathbf{p}_0 \mathbf{f}_n. \end{aligned}$$

Consequently the matrix A looks like this:

$$A = \Delta x \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_{-1} & \mathbf{p}_{-2} & 0 & 0 & 0 & \cdots & \mathbf{p}_2 & \mathbf{p}_1 \\ \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & \mathbf{p}_{-2} & 0 & 0 & \cdots & 0 & \mathbf{p}_2 \\ \mathbf{p}_2 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & \mathbf{p}_{-2} & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{p}_2 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & \mathbf{p}_{-2} & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & & & \\ \vdots & & & & & \ddots & & & \\ 0 & 0 & \cdots & \mathbf{p}_2 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & \mathbf{p}_{-2} \\ \mathbf{p}_{-2} & 0 & \cdots & 0 & \mathbf{p}_2 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} \\ \mathbf{p}_{-1} & \mathbf{p}_{-2} & \cdots & 0 & 0 & \mathbf{p}_2 & \mathbf{p}_1 & \mathbf{p}_0 \end{bmatrix}; \quad (2.14)$$

note the systematic band-diagonal structure, which characterizes A as a circulant matrix. Linear systems involving circulant matrices can be quickly solved using fast Fourier transforms, a topic we will return to later.

Returning to the general case of $\mathbf{p}$ defined by (2.9), the approximation formula (2.12) can be written in the form

$$A\mathbf{f} \approx \mathcal{A}f. \quad (2.15)$$

Figure 2.3 shows data computed by the discrete model $A\mathbf{f}$ and compares the result to the continuous data $(\psi * f)(x)$ defined by (2.4).

Now let's add a little noise to the data. For example, we might take $k = 64 = n$ and construct the measurement noise in a probabilistic manner by taking a realization of a random vector with 64 independently distributed Gaussian elements having standard deviation $\sigma = 0.01 \cdot \max |f(x)|$. This corresponds to a relative noise level of 1%.

2.1.3 Naïve deconvolution and inverse crimes

We illustrate numerically the failure of the following naïve reconstruction attempt:

$$\mathbf{f} \approx A^{-1}\mathbf{m} \approx A^{-1}(A\mathbf{f} + \varepsilon) = \mathbf{f} + A^{-1}(\varepsilon). \quad (2.16)$$

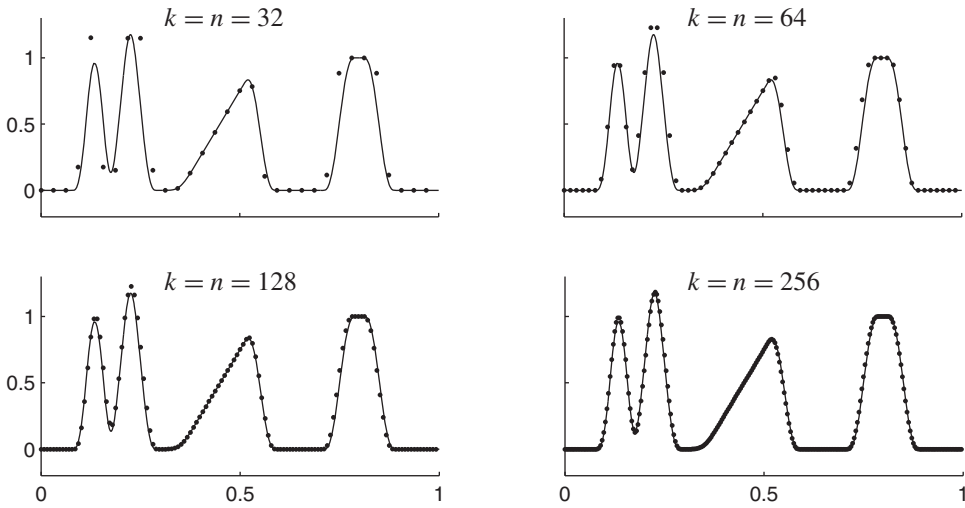


Figure 2.3. Illustration of the approximation $A\mathbf{f} \approx \mathcal{A}f$ of formula (2.15) for different choices of $k = n$. The actual function $(\psi * f)(x)$ defined by (2.4) is shown with a thin solid line, and the data points are indicated as dots. Note how the discrete approximation becomes better as the discretization is refined.

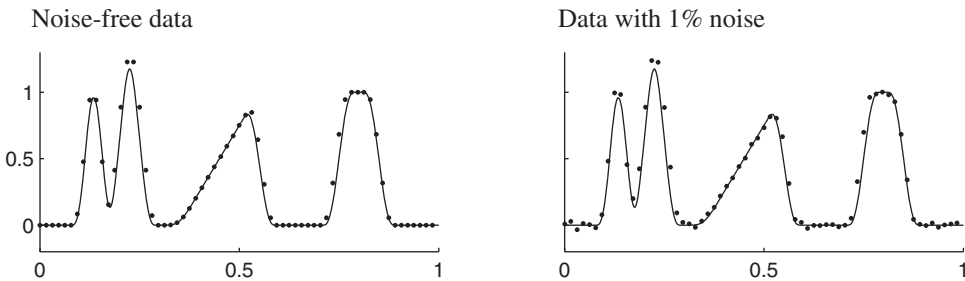


Figure 2.4. Illustration of simulated measurement noise. The actual function $(\psi * f)(x)$ defined by (2.4) is shown with a thin solid line, and the data points are indicated as dots. Left: noise-free discrete data $A\mathbf{f}$ with $n = 64 = k$. Right: the same data corrupted with 1% white noise.

In the case of no added noise ($\varepsilon = 0$) we use the data shown in the left plot of Figure 2.4 and get the left plot in Figure 2.5. The naïve reconstruction seems perfect! However, there is a catch. This apparently accurate reconstruction is not to be trusted; it is an example of an *inverse crime*. We will show how to avoid inverse crimes in Section 2.1.4.

If we apply naïve reconstruction (2.16) to the slightly noisy data shown in the right plot of Figure 2.4, we get the result shown in the right plot in Figure 2.5. It is completely useless. This example shows how sensitive inverse problems are to the smallest errors in measurement. We need to introduce *regularization* to overcome extreme sensitivity to measurement errors.

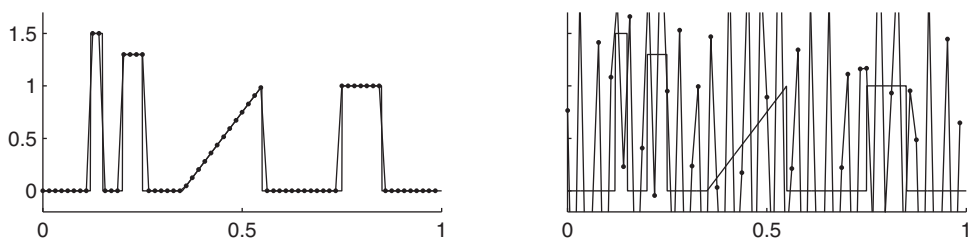


Figure 2.5. Two naïve deconvolutions by applying the inverse matrix A^{-1} to data. The original target function $f(x)$ is shown with a thin solid line, and the reconstruction is shown as dots. Left: naïve reconstruction (involving inverse crime) from the noise-free discrete data Af with $n = k = 64$ shown in the left plot in Figure 2.4. Right: naïve reconstruction from the noisy data shown in the right plot of Figure 2.4.

2.1.4 Naïve reconstruction without inverse crime

In the case of the deconvolution problem, we first simulate the measurements by convolving our known function f with a known discretized point spread function. In reality, when a blurred signal or image is encountered, the point spread function that “caused” the blurring is both unknown and can unlikely be expressed in simple terms. Thus, using the same point spread function for simulating a blurred signal and deconvolving the signal constitutes a serious inverse crime. Using the same point spread function *and* the same discretization mesh is an inverse felony!

We show one simple way to avoid inverse crime. We use a modified point spread function by taking $a = 0.041$ in (2.1) when simulating data. We compute the function $(\psi * f)(x)$ defined in (2.4) approximately at 1000 uniformly spaced points in the interval $[0, 1]$ using trapezoidal rule with 400 quadrature points for the evaluation of the integral. Finally, we interpolate the values of $\psi * f$ at the 64 grid points using splines.

Now the data has been simulated completely differently than using the 64×64 model matrix A as was (criminally) done in Section 2.1.3.

We apply naïve inversion (2.16) to the crime-free data and show the results in Figure 2.6. Compare the left plots in Figures 2.5 and 2.6. Proper simulation of crime-free data

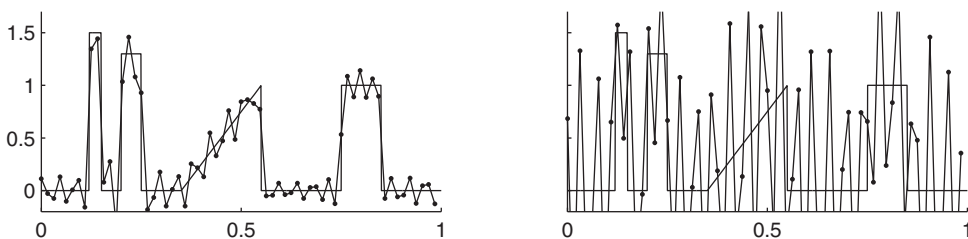


Figure 2.6. Two naïve deconvolutions by applying the inverse matrix A^{-1} to data generated avoiding inverse crime. The original target function $f(x)$ is shown with a thin solid line, and the reconstruction is shown as dots. Left: naïve reconstruction from noise-free discrete data with $n = k = 64$. Right: naïve reconstruction from noisy data. Compare to Figure 2.5.

reveals the ill-posedness of the deconvolution problem: The slightest perturbations in the data are amplified in naïve reconstruction using (2.16).

Exercise 2.1.1. Determine whether the point spread function ψ is a $C^\infty(\mathbb{R})$ function.

Exercise 2.1.2. What is the effect of increasing the support of ψ_0 on v ? Use the MATLAB programs `DCcontdatacomp.m` and `DCcontdataplot.m` to study the effect of increasing a on the convolved function. What do you observe?

Exercise 2.1.3. Plot a constant function of height 2 on $[0, 1]$ before and after convolution with ψ . Use the MATLAB program `DC2discretedatacomp.m` to add noise to the convolved function and `DC2naiveplot.m` to compute a naïve reconstruction. Plot your results.

2.2 Heat propagation

A classic ill-posed problem is that of determining the temperature distribution in a region from knowledge of the temperature distribution at the present time. This problem is known as the backward heat equation. We will begin with a discussion of the governing PDEs and their origins and then move to a simple discrete model.

2.2.1 Diffusion processes

The heat equation is the prototypical equation for modeling processes governed by pure diffusion. Following a probabilistic description as in, for example, [181], it can be derived by modeling the Brownian motion of the individual molecules in what we will assume to be a homogeneous material.

Suppose we have a material, such as depicted in Figure 2.7, containing n molecules, each of mass m , and suppose each molecule in this small volume is continually in motion. We will derive a model for one-dimensional spatial motion for simplicity, and so assume each molecule can only move to the left or to the right a distance Δx , representing an average displacement in time period Δt . To extend to higher dimensions, discrete motion in each of the three Cartesian coordinates would be permissible. Let p be the probability that the molecule moves to the right, and let q be the probability that the molecule moves to the left. Note that $p + q = 1$. Let $u(x, t)$ be the probability per unit length that a molecule

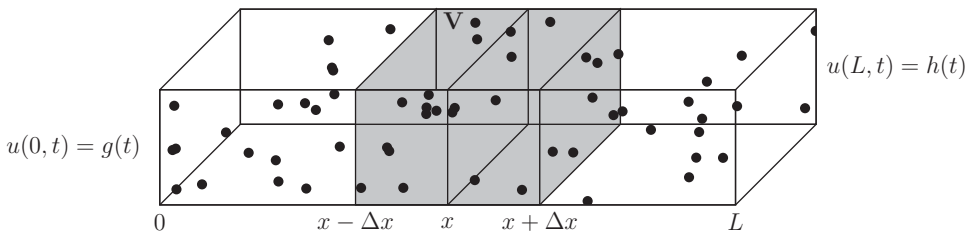


Figure 2.7. An illustration of the molecules in Brownian motion.

is located in the interval $[x - \frac{1}{2}\Delta x, x + \frac{1}{2}\Delta x]$ at time t . The integral $\int_a^b u(x, t) dx$ is the probability that a molecule is located in $[a, b]$ at time t and $u(x, t)$ is a *probability density*.

The mass of molecules between $x - \Delta x$ and x is approximately

$$mnu\left(x - \frac{1}{2}\Delta x, t\right)\Delta x,$$

so the mass of the molecules crossing the plane at x from left to right at time t is approximately $pmnu(x - \frac{1}{2}\Delta x, t)\Delta x$. Similarly, the mass of the molecules crossing the plane at x from right to left at time t is approximately $qmn u(x + \frac{1}{2}\Delta x, t)\Delta x$. Thus, the net mass flux ϕ across the plane at x in the positive x -direction over a time interval Δt is approximately

$$\phi(x, t) \approx mn \frac{\Delta x}{\Delta t} \left(pu\left(x - \frac{1}{2}\Delta x, t\right) - qu\left(x + \frac{1}{2}\Delta x, t\right) \right).$$

Taking a Taylor series expansion for $u(x \pm \frac{1}{2}\Delta x, t)$ about (x, t) ,

$$u\left(x \pm \frac{1}{2}\Delta x, t\right) = u(x, t) + u_x(x, t)\left(\pm \frac{1}{2}\Delta x\right) + \frac{1}{2!}u_{xx}(x, t)\left(\pm \frac{1}{2}\Delta x\right)^2 + \dots,$$

gives the linear approximations

$$\begin{aligned} u\left(x + \frac{1}{2}\Delta x, t\right) &\approx u(x, t) + \frac{\Delta x}{2}u_x(x, t) \\ u\left(x - \frac{1}{2}\Delta x, t\right) &\approx u(x, t) - \frac{\Delta x}{2}u_x(x, t). \end{aligned}$$

Now the net mass flux across the plane at x is approximately

$$\begin{aligned} \phi(x, t) &\approx pmn \frac{\Delta x}{\Delta t} \left(u(x, t) + \frac{\Delta x}{2}u_x(x, t) \right) - qmn \frac{\Delta x}{\Delta t} \left(u(x, t) - \frac{\Delta x}{2}u_x(x, t) \right) \\ &= (p - q)mn \frac{\Delta x}{\Delta t} u(x, t) - \frac{mn(\Delta x)^2}{2\Delta t} u_x(x, t)(p + q). \end{aligned}$$

The concentration $c(x, t)$ of molecules between $x - \frac{1}{2}\Delta x$ and $x + \frac{1}{2}\Delta x$ is approximately

$$c(x, t) \approx \frac{1}{\Delta x}(mnu(x, t)\Delta x).$$

So in terms of concentration, our expression for net flux becomes

$$\phi(x, t) \approx (p - q) \frac{\Delta x}{\Delta t} c(x, t) - \frac{1}{2} \frac{(\Delta x)^2}{\Delta t} c_x(x, t).$$

As $\Delta t \rightarrow 0$ assume that

$$\lim_{\Delta t \rightarrow 0} (p - q) \frac{\Delta x}{\Delta t} = \lambda,$$

where λ is a constant known as the drift constant, and

$$\lim_{\Delta t \rightarrow 0} \frac{1}{2} \frac{(\Delta x)^2}{\Delta t} = \frac{1}{2} D > 0,$$

where D is a constant known as the diffusion coefficient. Then in the limit as $\Delta t \rightarrow 0$ the net flux becomes

$$\phi(x, t) = \lambda c(x, t) - \frac{1}{2} D c_x(x, t). \quad (2.17)$$

For a volume unit V , the quantity $\int_V c(x, t) dx$ represents the total mass in V , and conservation of mass implies that the time rate of change of the total mass in the volume V equals the flux across the boundary plus any mass created by sources f inside the volume V :

$$\frac{d}{dt} \int_V c(x, t) dx = - \int_{\partial V} \phi \cdot nds + \int_V f dx. \quad (2.18)$$

By the divergence theorem

$$\int_{\partial V} \phi \cdot nds = \int_V \nabla \cdot \phi dx,$$

and here in our one-dimensional model, equation (2.18) takes the form

$$\frac{d}{dt} \int_{x-\Delta x}^{x+\Delta x} c(x, t) dx = - \int_{x-\Delta x}^{x+\Delta x} \frac{\partial}{\partial x} \phi(x, t) dx + \int_{x-\Delta x}^{x+\Delta x} f dx. \quad (2.19)$$

Since the interval $[x - \Delta x, x + \Delta x]$ is arbitrary, we have

$$c_t(x, t) + \phi_x(x, t) = f(x, t)$$

or, from (2.17),

$$c_t(x, t) + \lambda c_x(x, t) - \frac{1}{2} D c_{xx}(x, t) = f(x, t).$$

Since $c(x, t) = mnu(x, t)$, under the assumption of no sources or sinks ($f = 0$) and the assumption that $\lambda = 0$ (which can also be thought of as $p = q$), we have the familiar heat equation with initial and boundary conditions:

$$\begin{aligned} u_t - Du_{xx} &= 0, & 0 < x < L, & \quad t > 0, \\ u(0, t) &= g(t), & t > 0, \\ u(L, t) &= h(t), & t > 0, \\ u(x, 0) &= f(x), & 0 < x < L. \end{aligned} \quad (2.20)$$

The forward problem is to determine the temperature distribution $u(x, t)$ throughout the domain at time t from knowledge of $u(x, 0)$. If the endpoints of the bar are kept at zero temperature, we have $g(t) = h(t) = 0$, which we will henceforth take for simplicity. Problems with nonzero boundary conditions can be transformed to zero boundary conditions through a change of variables.

The backward problem is to determine the temperature distribution $u(x, t)$ at some prior time $t < T$ from knowledge of $u(x, T)$,

$$\begin{aligned} u_t - Du_{xx} &= 0, & 0 < x < L, & \quad t > 0, \\ u(0, t) &= 0, & t > 0, \\ u(L, t) &= 0, & t > 0, \end{aligned} \quad (2.21)$$

$$u(x, T) = m(x), \quad 0 < x < L. \quad (2.22)$$

2.2.2 A finite difference discrete model

In the following section, we will look at the fundamental solution for the heat equation and discretize the integral equation for the solution of the backward problem. Here, we consider another elementary approach: a finite difference discretization with explicit time stepping.

Define a mesh on the spatial domain by

$$x_0 = 0, \quad x_1 = \frac{L}{M+1}, \quad x_2 = \frac{2L}{M+1}, \dots, x_{M+1} = L.$$

Then $\Delta x = L/(M+1)$. Define a sequence of uniform time steps up to time T by

$$t_0 = 0, \quad t_1 = \frac{T}{N+1}, \quad t_2 = \frac{2T}{N+1}, \dots, t_{N+1} = T.$$

Then $\Delta t = T/(N+1)$. Denote $u(x_i, t_j)$ by u_{ij} . The Taylor series expansion for $u(x + \Delta x, t)$ about (x, t) is

$$u(x + \Delta x, t) = u(x, t) + (\Delta x)u_x(x, t) + \frac{(\Delta x)^2}{2!}u_{xx}(x, t) + O((\Delta x)^3). \quad (2.23)$$

For a linear approximation to $u(x + \Delta x, t)$, the terms of order $(\Delta x)^2$ and higher are dropped. Solving for $u_x(x, t)$ in the linear approximation results in the *forward difference formula*

$$u_x(x, t) = \frac{1}{\Delta x}(u(x + \Delta x, t) - u(x, t)) + O(\Delta x),$$

or in subscript notation

$$(u_x)_{ij} \approx \frac{1}{\Delta x}(u_{i+1,j} - u_{ij}).$$

Similarly, in the time variable

$$(u_t)_{ij} \approx \frac{1}{\Delta t}(u_{i,j+1} - u_{ij}). \quad (2.24)$$

Alternatively, by evaluating the Taylor expansion with respect to t about (x, t) at $(x, t - \Delta t)$ one obtains the *backward difference formula*

$$(u_t)_{ij} \approx \frac{1}{\Delta t}(u_{ij} - u_{i,j-1}). \quad (2.25)$$

An approximation to the second spatial derivative of u can be obtained by adding (2.23) and the Taylor expansion in (2.23) evaluated at $(x - \Delta x, t)$ to obtain

$$(u_{xx})_{ij} \approx \frac{1}{(\Delta x)^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}). \quad (2.26)$$

Combining the approximations (2.24) and (2.26), the finite difference discretization of the PDE (2.20) is

$$\frac{u_{i,j+1} - u_{ij}}{\Delta t} = \frac{D}{\Delta x^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}). \quad (2.27)$$

In the solution of the forward model, we step forward in time solving for $u_{i,j}$ for $i = 1, \dots, M$ and $j = 1, 2, \dots$ until we reach the desired time at which we want to compute the solution. An analysis of the error in making these approximations to the derivatives shows that the finite difference method will converge to the solution of the forward problem, provided

$$0 < \frac{\Delta t}{(\Delta x)^2} < \frac{1}{2}. \quad (2.28)$$

Most texts on the numerical solution of PDEs contain a proof of this result. See, for example, [9]. The boundary conditions are assumed to be known, and they are included in the solution as follows:

$$u_{0,j} = g(j \Delta t), \quad (2.29)$$

$$u_{M+1,j} = h(j \Delta t). \quad (2.30)$$

The initial condition is included by setting

$$u_{i,0} = f(i \Delta x).$$

Now we can write the solution at time step t_{i+1} as a linear system as follows. For simplicity, let $d = \frac{D \Delta t}{(\Delta x)^2}$ and denote by A the banded matrix

$$A = \begin{bmatrix} 1-2d & d & 0 & \dots & \dots & 0 \\ d & 1-2d & d & 0 & \dots & 0 \\ 0 & d & 1-2d & d & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & d & 1-2d & d \\ 0 & \dots & \dots & 0 & d & 1-2d \end{bmatrix}.$$

Let $\mathbf{u}^j$ denote the vector at the j th time step $\mathbf{u}^j = [u_{1,j}, \dots, u_{M,j}]^T$ and $\mathbf{v}_j$ the vector at the j th time step $\mathbf{v}^j = [du_{0,j}, 0, \dots, 0, du_{M+1,j}]^T$. Provided the stability criterion (2.28) holds, the solution at the $(j+1)$ st time step is approximated by computing

$$\mathbf{u}^{j+1} = A\mathbf{u}^j + \mathbf{v}^j. \quad (2.31)$$

Example. Consider the forward problem

$$u_t - u_{xx} = 0, \quad 0 < x < \pi, \quad t > 0, \quad (2.32)$$

$$u(0, t) = 0, \quad t > 0, \quad (2.33)$$

$$u(L, t) = 0, \quad t > 0, \quad (2.34)$$

$$u(x, 0) = 10 \sin 2x, \quad 0 < x < \pi. \quad (2.35)$$

One can show that the actual solution to the forward problem is $u(x, t) = 10e^{-4t} \sin 2x$. The relative errors for the solution computed out to times $T = 0.1, 0.2, 0.3, 0.4$ are given in Table 2.1. A plot of the evolution of the solution in time is found in Figure 2.8. Notice that the solution has decayed to nearly zero by time $T = 1$.

Data at time $T = 0.4$ was simulated using (2.31) with $\Delta x = 0.1366$ and $\Delta t = 0.0037$, which corresponds to $M+1 = 24$ in our spatial discretization and $K = 108$ time steps. The noise-free data and data with 2% noise is found in Figure 2.8.

Table 2.1. Accuracy of the finite difference forward solver on the problem (2.32)–(2.35).

Time T	Δx	Δt	$\ u(x, T)\ _\infty$	Relative sup-norm error
0.1	0.0668	4.65e-4	6.67	2.24e-4
0.2	0.0806	0.002	4.49	0.0015
0.3	0.0668	0.0014	3.01	0.0016
0.4	0.1366	0.0037	2.01	0.0020

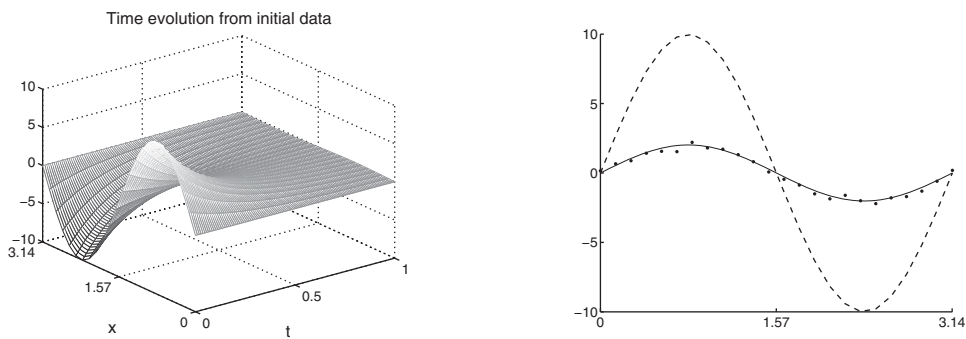


Figure 2.8. Left: illustration of the time evolution of the forward solution of the heat equation with initial condition $u_0(x) = 10\sin 2x$ computed to time $T = 1$. Right: illustration of simulated measurement noise for the heat equation. The dashed line is the initial profile. The noise-free data at time $T = 0.4$ is the solid line. The computed solution corrupted with 2% white noise is plotted with large dots.

2.2.3 Naïve reconstruction of the initial temperature

A naïve approach to the inverse problem of determining u at the previous time step would be to solve

$$\mathbf{u}^j = A^{-1}(\mathbf{u}^{j+1} - \mathbf{v}^j). \tag{2.36}$$

This casts the problem in a discrete form. Since the stepsize in time is limited by the stability criterion, it will take numerous time steps to reach the initial condition, but this can be achieved by iterating the method (2.36). The results of applying (2.36) and iterating backward in time with and without noisy data at $T = 0.4$ are shown in Figure 2.9. The results are displayed at the iterate at which the solution to the backward problem begins to become unstable. In the case of noise-free data, this occurs at approximately 22 backward steps, or at time $t = 0.3215$. However, the method is very sensitive to noise in the data, and with just 0.01% noise, it is only stable for approximately four backward steps, or $t = 0.3888$. In either case, a serious inverse crime is being committed here. The same method on the same mesh is being used to both generate and reconstruct the data, and the results are therefore better than they should be!

The approach (2.36) is not equivalent to using backward differences and time-stepping backward in that manner since A^{-1} is not equal to the matrix that arises from that approach. Let us next investigate that approach. Stepping backward in time from $j = K - 1$, we solve

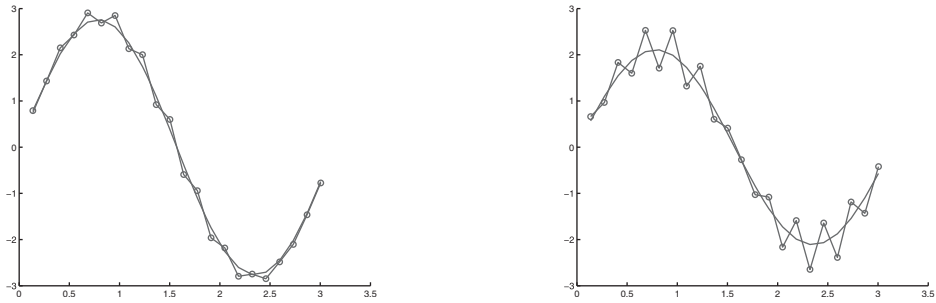


Figure 2.9. Left: reconstruction of the heat profile at time $t = 0.3215$ from noise-free data measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.36) 22 steps. Right: reconstruction of the heat profile at time $t = 0.3888$ from data with 0.01% random noise measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.36) 4 steps.

for each $u_{i,j}$ from

$$u_{i,j} = u_{i,j+1} - \frac{D\Delta t}{\Delta x^2}(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}). \quad (2.37)$$

Plots of the results from four time steps backward with and without noisy data are found in Figures 2.10 and 2.11. We see that this method is somewhat more stable than (2.36), but it is still not useful for long times. In the computations resulting in Figure 2.10 an inverse crime is still being committed since the same mesh is used for the solution of the inverse

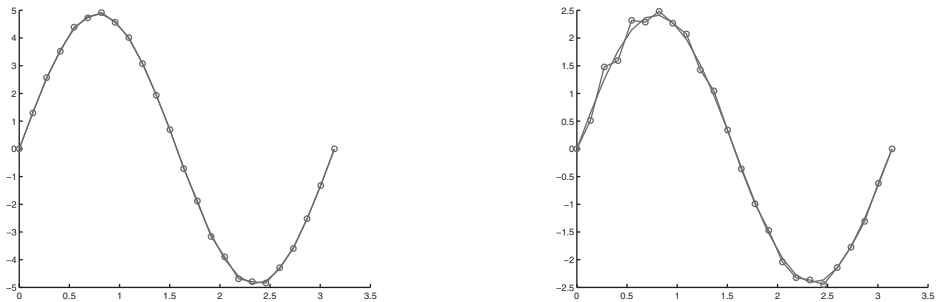


Figure 2.10. Left: reconstruction of the heat profile at time $t = 0.1794$ from noise-free data measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.37) 60 steps. Here, an inverse crime was still committed since the same mesh and time steps were used to construct the data and compute the solution. The results are undeservingly good. Right: reconstruction of the heat profile at time $t = 0.3551$ from data with 0.01% random noise measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.37) 13 steps. The same inverse crime is committed here.

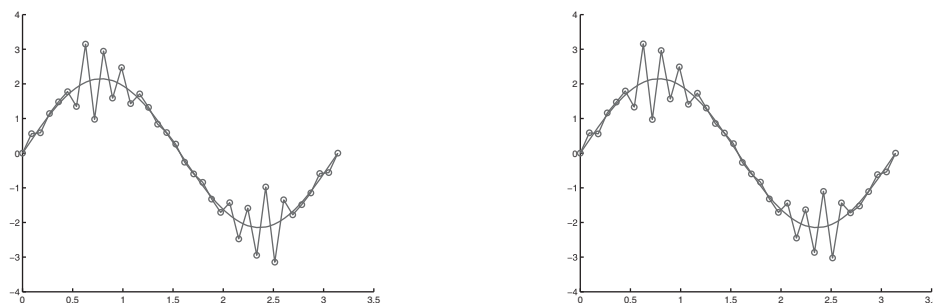


Figure 2.11. Left: reconstruction of the heat profile at time $t = 0.3844$ from noise-free data measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.37) 8 steps. Equation (2.31) was used to compute the data on a different mesh from (2.37). Right: reconstruction of the heat profile at time $t = 0.3844$ from data with 0.01% random noise measured at $T = 0.4$. The solid line is the actual solution and the line with dots is the reconstruction computed by iterating (2.37) 8 steps.

problem as was used for the construction of the data by (2.31). By using a different mesh and time steps for the solution of (2.31) and (2.37) and interpolating the data to the mesh of (2.37), an inverse crime is avoided. This was the approach used in computing the results in Figure 2.11.

Exercise 2.2.1. Complete Table 2.1 for later times $T = 0.5, 0.6, \dots, 1.0$. Observe what happens to the solution when the stability criterion is violated.

Exercise 2.2.2. Compute reconstructions by the methods in this section for the same example but use data at time $T = 0.1$. Do not commit any inverse crimes. Is the method more stable than from the final time data at $T = 0.4$?

Exercise 2.2.3. Study the numerical forward solution of the heat equation by adding noise to the initial data $u_0(x) = 10 \sin 2x$ in the MATLAB program `InverseHeatCondData-Simulator.m` and computing the solution at time $T = 0.4$. Plot the difference between the computed $u(x, T)$ from a noisy and noise-free initial condition. How does this differ from what we see in Figures 2.9 and 2.11?

Exercise 2.2.4. Use the MATLAB program `InvHeatCondNaiveSolver.m` and modify it to use the initial temperature distribution $f(x) = 10 \chi_{[\pi/4, 3\pi/4]}(x)$, where χ is the characteristic function. Compute noise-free and noisy simulated data, and compute naive reconstructions at four prior time steps using methods (2.36) and (2.37). Plot your results. Give the reason that the discontinuity in the initial condition can never be reconstructed.

2.3 Tomographic X-ray projection data

In tomographic X-ray imaging one takes X-ray projection images of an object from several different directions and attempts to recover the inner structure of the object from the data.

We show how such a measurement can be written in the form $\mathbf{m} = \mathbf{A}\mathbf{f} + \varepsilon$ and illustrate numerically how the naïve reconstruction approach (1.4) fails.

2.3.1 A simple example: Probing two aluminum slabs

Let us first demonstrate the exponential attenuation law of X-rays using a very simple example, where two aluminum slabs are probed as shown in Figure 2.12. Typically, X-rays emanate from a roughly point-like location inside an X-ray tube. That point is called the *X-ray source* and shown as a black dot in Figures 2.12 and 2.13. Three X-rays are sent traveling towards a detector, each consisting initially of 1000 photons. The detector is capable of counting how many photons arrive at each point. One of the rays arrives at the detector through empty space, delivering all 1000 photons. Another ray travels through an aluminum slab whose width is chosen to be the *half-thickness* of the X-radiation used here. This means that half of the photons entering the slab will be absorbed inside the slab. The third ray encounters two such aluminum slabs. We call these three X-rays the empty-space ray, the one-slab ray, and the two-slab ray, respectively.

The photon count data can now be transformed into line integral data via two simple steps. First, take the logarithm of each photon count. Then, realizing that the integral of the empty-space ray must be zero, subtract each logarithm from the logarithm corresponding to the empty-space ray. As seen from the actual numbers shown in Figure 2.12, the resulting attenuation data is zero for the empty-space ray, a positive number (0.693) for the one-slab ray, and twice that number (1.386) for the two-slab ray.

We have described the basic calibration process for ideal photon count data based on the exponential attenuation law. However, we ignored at least a couple of properties of real-world measurements. First, practical detectors (for instance, charge-coupled devices or CCDs) do not provide the actual photon count but rather an integer that is proportional to the photon count. However, this is not a serious problem, as you can find out in Exercise 2.3.2 below. Second, the photon count is not a deterministic number; it is better modeled as a random variable with Poisson distribution. This results in random measurement noise in the data; we will discuss this below.

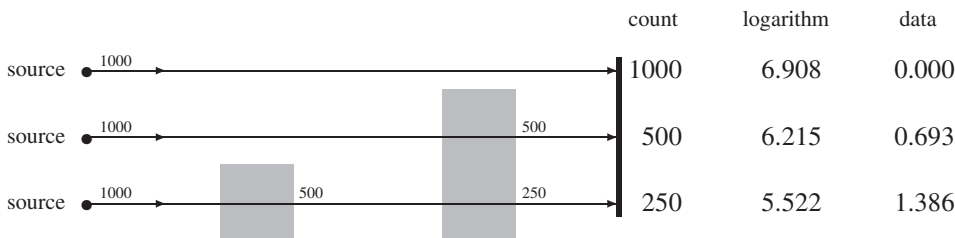


Figure 2.12. Simple experiment illustrating the attenuation of X-rays and interpretation of measurements. The three black dots show the positions of the X-ray source, and the horizontal lines depict X-rays. The gray boxes are slabs of attenuating material, and their width has been chosen to be the half-thickness of the X-radiation. The vertical thick line is the detector counting how many photons arrive at each point.

2.3.2 From photon count data to line integral data

The two-slab example in Section 2.3.1 is quite simple as it concerns only homogeneous material. Consider now an X-ray traveling through a phantom¹ representing a two-dimensional cross-section of a patient's head along a straight line, as shown in the left panel of Figure 2.13. We place the target slice inside the unit square defined by $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$. For the sake of argument, assume that the X-ray travels along the horizontal path defined by $0 \leq x_1 \leq 1$ and $x_2 = \frac{1}{2}$.

Interaction between radiation and matter lowers the intensity of the ray. We think of the X-ray having initial intensity $I_0 := I(0)$ when entering the patient's head and smaller intensity $I_1 := I(1)$ when exiting. Also, we denote by $I(x_1)$ the intensity of the X-ray at the point $(x_1, \frac{1}{2})$ while traveling from the source to the detector.

In contrast to the simple homogeneous slab example above, the cross-section of a head contains various tissues with different X-ray attenuation properties. We model this situation using a nonnegative attenuation coefficient function $f(x_1, x_2)$, whose value gives the relative intensity loss of the X-ray within a small distance dx :

$$\frac{dI(x_1)}{I(x_1)} = -f\left(x_1, \frac{1}{2}\right) dx_1.$$

For example, bone has higher attenuation coefficient than brain tissue, and cerebrospinal fluid (white ovals in the left panel of Figure 2.13) provides practically zero attenuation. See the right panel in Figure 2.13 for a plot of the profile $f(x_1, \frac{1}{2})$.

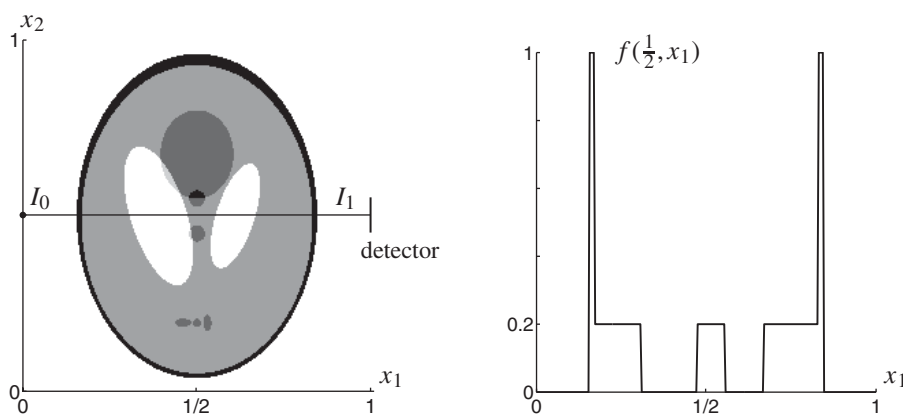


Figure 2.13. X-ray measurement. Left: an X-ray traveling through a simulated cross-section of a human head (a low-contrast version of the infamous Shepp–Logan phantom). Note that high attenuation is shown here as darker shades of gray and low attenuation as lighter shades. Right: plot of the attenuation coefficient along the path of the X-ray.

¹A *phantom* can be either a physical calibration device or a mathematical model. Here it is a mathematical model to simulate an idealized cross-section of a human head. The use of “phantom” will be clear from the context.

Integration along the X-ray from source to detector gives

$$\int_0^1 f\left(x_1, \frac{1}{2}\right) dx_1 = -\int_0^1 \frac{I'(x_1)}{I(x_1)} dx_1 = \log I_0 - \log I_1. \quad (2.38)$$

Now the right-hand side of (2.38) is known: I_0 by calibration and I_1 from the measurement. The left-hand side of (2.38) consists of an integral of the unknown function f over a straight line, as wished.

Regarding noise, the quantity I_1 is a constant multiple of a Poisson-distributed random variable. It is typically sampled in practice using an analog-to-digital converter that produces integer output containing truncation errors and additional electronic noise. Taking logarithm of I_1 leads to a random variable with remarkably complicated statistics. However, it is usually quite plausible to model the measurement as

$$\log I_0 - \log I_1 = \int_0^1 f\left(x_1, \frac{1}{2}\right) dx_1 + \varepsilon, \quad (2.39)$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is a normally distributed random variable. The standard deviation σ of the noise can be estimated, for example, by measuring the same target repeatedly and calculating the standard deviation of the samples. This procedure is a reasonably accurate model when the photon count is large enough; see [414, Appendix].

We remark that in the above model we neglect the energy dependence of the attenuation function. Namely, most X-ray sources produce a multispectral beam, and an energy-dependent f may result in different measured line integrals depending on the propagation direction of the X-ray along the line. This is called *beam hardening*.

2.3.3 Continuous tomographic data: The Radon transform

In the previous section we described how to turn attenuation data from one single X-ray into line integral data concerning a nonnegative, compactly supported attenuation coefficient $f: \mathbb{R}^2 \rightarrow \mathbb{R}$. The aim of tomographic imaging is to collect information about f using different angles of view.

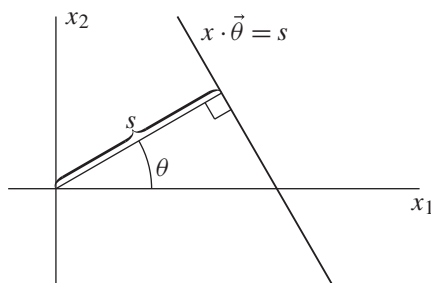
Let us define the *Radon transform*, denoted by $\mathfrak{R}$, as follows. We interpret $\theta \in \mathbb{R}$ as an angle measured in radians, and denote by

$$\vec{\theta} := \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \in \mathbb{R}^2$$

the unit vector with angle θ with respect to the x_1 -axis. The Radon transform of the function f depends on the angular parameter θ and on a linear parameter $s \in \mathbb{R}$ in the following way:

$$\mathfrak{R}f(s, \theta) = \int_{x \cdot \vec{\theta} = s} f(x) dx^\perp, \quad (2.40)$$

where $dx^\perp$ denotes the one-dimensional Lebesgue measure along the line defined by $\{x \in \mathbb{R}^2 : x \cdot \vec{\theta} = s\}$. We remark that the parametrization of tomographic data provided by formula (2.40) is related to the so-called parallel-beam geometry used in the first-generation computed tomography (CT) scanners in the 1970s.



$$\mathfrak{R}f(s, \theta) = \int_{x \cdot \vec{\theta} = s} f(x) dx^\perp$$

Figure 2.14. Illustration of the definition (2.40) of the Radon transform. The line $\{x \in \mathbb{R}^2 : x \cdot \vec{\theta} = s\}$ is drawn with a thick line.

Many variations in the data geometry are possible, such as limited angle data, local tomography data, exterior tomography, and combinations thereof. We refer the reader to the classical texts by Natterer [353] and Kak and Slaney [249]. See [353] for analytic inversion formulas, a thorough analysis of the mapping properties of $\mathfrak{R}$, and its generalizations to higher dimensions. See [354] for another perspective on image reconstruction.

The Fourier transform and Radon transform are connected in a simple way. This result is known as the central slice theorem. First, define the Fourier transform in one dimension as follows. See Figure 2.14 for an illustration.

Definition 2.3.1. The Fourier transform of a function defined on $\mathbb{R}$ is given by

$$\mathcal{F}(x)(\xi) = \hat{f}(\xi) = \frac{1}{(2\pi)^n} \int_{\mathbb{R}} f(x) e^{-ix\xi} dx.$$

Theorem 2.1. Let f be an absolutely integrable function defined on the whole real line. For any real number r and unit vector $\vec{\theta}$, we have the identity

$$\int_{-\infty}^{\infty} \mathfrak{R}f(s, \vec{\theta}) e^{-isr} ds = \hat{f}(r\vec{\theta}). \quad (2.41)$$

Proof. By the definition of the Radon transform

$$\begin{aligned} \int_{-\infty}^{\infty} \mathfrak{R}f(s, \vec{\theta}) e^{-isr} ds &= \int_{-\infty}^{\infty} \int_{x \cdot \vec{\theta} = s} f(x) e^{-isr} dx^\perp ds \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x) e^{-ix \cdot (r\vec{\theta})} dx_1 dx_2 \\ &= \hat{f}(r\vec{\theta}). \quad \square \end{aligned}$$

It will prove convenient to have a notation for the one-dimensional Fourier transform of a function in the scalar parameter as appears in the central slice theorem. Let $\tilde{h}(s, \vec{\theta})$ denote such a Fourier transform:

$$\tilde{h}(s, \vec{\theta}) = \int_{-\infty}^{\infty} h(t, \vec{\theta}) e^{-its} dt. \quad (2.42)$$

Then the central slice theorem says

$$\widetilde{\Re} f(r, \vec{\theta}) = \hat{f}(r\vec{\theta}).$$

The Radon inversion formula provides a way to obtain f from its Radon transform in the ideal case.

Theorem 2.2. *If f is an absolutely integrable function defined on the real line and $\hat{f}$ is absolutely integrable, then*

$$f(x) = \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^\infty e^{isx \cdot \vec{\theta}} \widetilde{\Re} f(s, \vec{\theta}) |s| ds d\theta. \quad (2.43)$$

Proof. First note that since the Radon transform satisfies $\Re f(-s, -\vec{\theta}) = \Re f(s, \vec{\theta})$,

$$\begin{aligned} \widetilde{\Re} f(-s, -\vec{\theta}) &= \int_{-\infty}^\infty \Re f(t, -\vec{\theta}) e^{-it(-s)} dt \\ &= \int_{-\infty}^\infty \Re f(t, -\vec{\theta}) e^{-i(-t)s} dt \\ &= \int_{-\infty}^\infty \Re f(-t, -\vec{\theta}) e^{-its} dt \\ &= \int_{-\infty}^\infty \Re f(t, \vec{\theta}) e^{-its} dt \\ &= \widetilde{\Re} f(s, \vec{\theta}). \end{aligned}$$

Now by the Fourier inversion formula, with $\xi = (r \cos \theta, r \sin \theta)$,

$$\begin{aligned} f(x) &= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \hat{f}(\xi) e^{ix \cdot \xi} d\xi \\ &= \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^\infty \hat{f}(r\vec{\theta}) e^{irx \cdot \vec{\theta}} r dr d\theta \\ &= \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^\infty \widetilde{\Re} f(r, \vec{\theta}) e^{irx \cdot \vec{\theta}} r dr d\theta \\ &= \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^\infty \widetilde{\Re} f(r, \vec{\theta}) e^{irx \cdot \vec{\theta}} |r| dr d\theta, \end{aligned}$$

where the last equality follows from the fact that $\widetilde{\Re} f(-s, -\vec{\theta}) = \widetilde{\Re} f(s, \vec{\theta})$. $\square$

To summarize, this results in the following idealized reconstruction algorithm for X-ray CT imaging:

- Let f be the attenuation coefficient of a two-dimensional slice of a three-dimensional object. Then the intensity $I_{(s, \vec{\theta})}$ of the beam satisfies the differential equation

$$\frac{dI_{(s, \vec{\theta})}}{I_{(s, \vec{\theta})}} = -f(s, \vec{\theta}) ds.$$

- We measure the Radon transform of f ,

$$\Re f(s, \vec{\theta}) = \log \left(\frac{I_0}{I_d} \right),$$

where I_0 is the intensity of the beam at the source, and I_d is the intensity of the beam at the detector.

- Reconstruct f from the Radon inversion formula (2.43).

For the filtered back-projection algorithm, we regard the radial integral in the Radon Inversion Formula as a filter. We denote the output of the filter by $\mathcal{G}\Re f(t, \vec{\theta})$, where

$$\mathcal{G}\Re f(t, \vec{\theta}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widetilde{\Re f}(r, \vec{\theta}) e^{irt} |r| dr.$$

Then, with $t = x \cdot \vec{\theta}$,

$$f(x) = \frac{1}{2\pi} \int_0^\pi \mathcal{G}\Re f(x \cdot \vec{\theta}, \vec{\theta}) d\theta.$$

Note that one sees from this formula that low-frequency components are suppressed by $|r|$ and high-frequency components are amplified. Let's look at the filter a little more carefully. Recall that the Fourier transform of $g'(t)$ is

$$\mathcal{F}(\partial_t g)(\xi) = i\xi \hat{g}(\xi).$$

Thus if we had r instead of $|r|$ in the Radon inversion formula, we would have had the

$$\text{“inversion formula”} = \frac{1}{2\pi i} \int_0^\pi \partial_r \Re f(r, \theta) d\theta.$$

If f is real-valued, this quantity is purely imaginary! Thus, the $|r|$ is very important!

The MATLAB function `iradon.m` in the Image Processing Toolbox implements filtered back-projection. In the subsequent sections, we will be comparing the results of filtered back-projection implemented with `iradon.m` to other inversion techniques.

2.3.4 Discrete tomographic data

We model practical tomographic X-ray data by a bounded set $\Omega \subset \mathbb{R}^2$, a nonnegative attenuation coefficient f supported in $\overline{\Omega}$, and some finite collection $\{L_j\}_{j=1}^k$ of lines $L_j \subset \mathbb{R}^2$ intersecting Ω .

As a first example, we will use the following data set. It is an example of *parallel-beam geometry* illustrated in Figure 2.15. The angular variable is sampled with equidistant steps over the half circle:

$$\theta_j = \theta_1 + \left(\frac{j-1}{J} \right) \pi, \quad 1 \leq j \leq J, \quad (2.44)$$

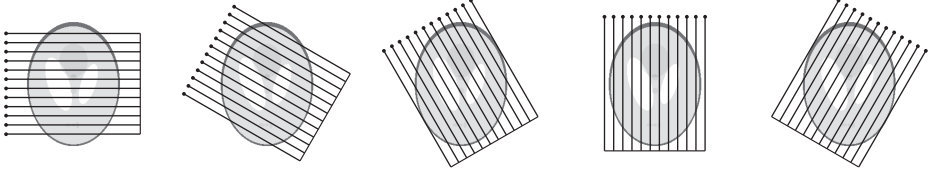


Figure 2.15. Parallel beam X-ray measurement geometry. Here $J = 5$ and $N = 11$. Black dots show the locations of the X-ray source at different times of measurement. The thick line represents the detector measuring the intensity of the X-rays after passing through the target. High attenuation is shown here as darker shades of gray and low attenuation as lighter shades.

where $\theta_1 \in \mathbb{R}$ is an appropriate constant, a reference angle. The linear parameter s is also sampled uniformly over a suitable interval:

$$s_v = -S + 2 \left(\frac{v-1}{N} \right) S, \quad 1 \leq v \leq N, \quad (2.45)$$

where $S > 0$.

Defining $k = JN$, the measurement (1.1) then takes the form

$$\mathbf{m} = \mathcal{A}f + \varepsilon = \begin{bmatrix} \int_{L_1} f(x_1, x_2) ds_1 \\ \vdots \\ \int_{L_k} f(x_1, x_2) ds_k \end{bmatrix} + \varepsilon, \quad (2.46)$$

where ds_j denotes the one-dimensional Lebesgue measure along the line L_j . Each integral in (2.46) can be understood as a suitable rotation and scaling of formula (2.39).

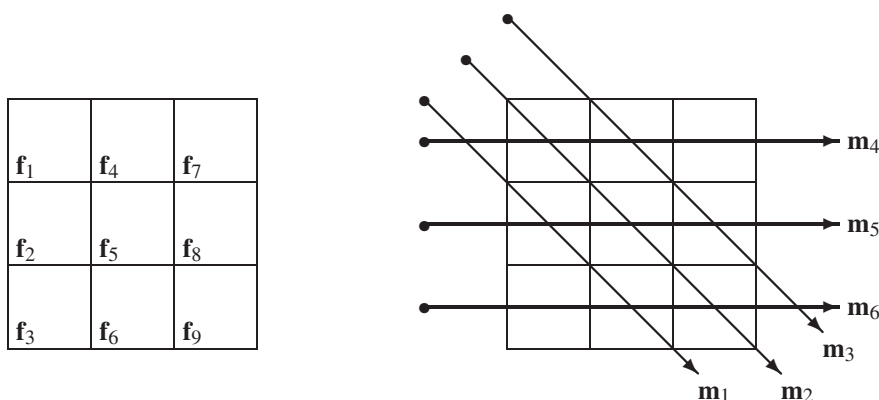
For the computational solution we need to build a finite-dimensional measurement model of the form (1.3). We discretize the tomographic problem by dividing the unknown area into n pixels and assume that attenuation values are constant within each pixel. We number the pixels from 1 to n and call the corresponding attenuation values $\mathbf{f}_j \geq 0$ for $j = 1, \dots, n$.

The measurement $\mathbf{m}_i$ of the line integral of f over line L_i is approximated by

$$\mathbf{m}_i = \int_{L_i} f(x_1, x_2) ds \approx \sum_{j=1}^n a_{ij} \mathbf{f}_j, \quad (2.47)$$

where a_{ij} is the distance that L_i travels in the j th pixel. Note that only pixels that intersect the beam L_i are included in this sum. Further, if we have k measurements in the vector $\mathbf{m} \in \mathbb{R}^k$, then (2.47) yields a matrix equation $\mathbf{m} = \mathbf{A}\mathbf{f}$, where the matrix is defined by $\mathbf{A} = (a_{ij})$.

Consider the following discretization and measurements, where $J = 2$, $k = 6$, $N = 3$ and the total number of pixels is $N^2 = 9$:



Here we have divided the square-shaped domain $\Omega \subset \mathbb{R}^2$ into 9 pixels, denoted by thin lines. The length of the side of each pixel is 1. Inside the pixels there is a constant value f_j of attenuation. The six arrows are X-rays used for probing the inner structure of Ω . Measurement data is the vector $\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_6]^T$ modeled by (2.47). The resulting measurement model is

$$\begin{bmatrix} 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{bmatrix}. \quad (2.48)$$

The model (2.48) is low-dimensional and simple. However, it already demonstrates one feature typical for inverse problems: nonuniqueness of solution. Namely, as can be seen in Exercise 2.3.5, there are several targets that produce exactly the same data. Thus the inverse problem cannot be uniquely solved using the measurement information alone.

Let us build a more realistic (higher-dimensional) data simulation model. We work with the so-called Shepp–Logan phantom, which is a piecewise constant model of a cross-section of a human head. The phantom is defined using ellipses and can be realized at any desired discrete resolution. See Figure 2.16 for pictures of the Shepp–Logan phantom at discretizations with $16 \times 16 = 256$ pixels, $50 \times 50 = 2500$ pixels, and $512 \times 512 = 262144$ pixels.

Let us construct the measurement matrix A corresponding to the low-resolution case with 16×16 pixels and projection directions specified by taking $J = 16$ in (2.44). In this case the size of A is not too large and we can show a picture of the nonzero elements of A for observing its structure.

We use MATLAB's command `radon.m` to simulate parallel-beam X-ray projection data from 16×16 pixel images with zero entries except one pixel with value 1. The pixel value 1 is first located in pixel 1 in the numeration shown in Figure 2.17(b), then in pixel 2, and so on. This way, column by column, we construct a measurement matrix A for a computational tomography model of the form (1.3).

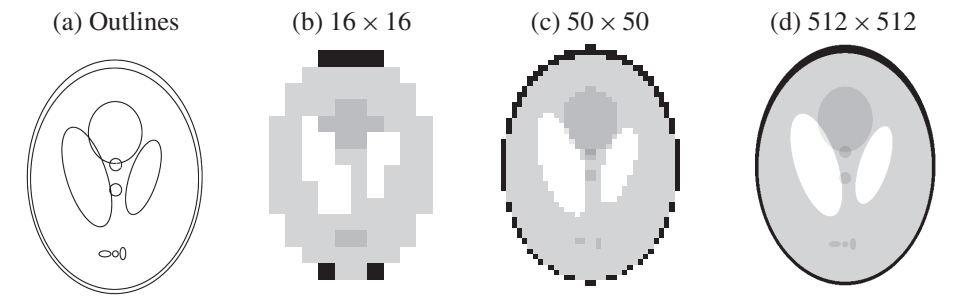


Figure 2.16. (a) Schematic illustration of the Shepp–Logan phantom. Areas of different attenuation values are bounded by various ellipses. (b)–(d) Plots of grayscale images of the Shepp–Logan phantom at different resolutions. High attenuation is shown here as darker shades of gray and low attenuation as lighter shades.

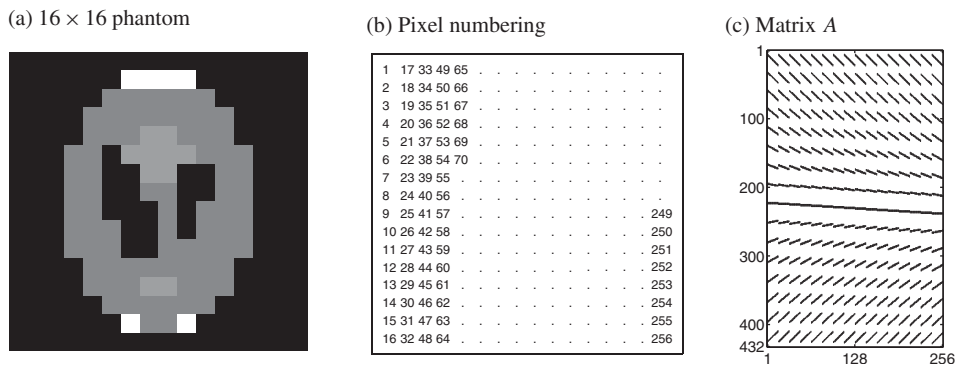


Figure 2.17. (a) Low-resolution Shepp–Logan phantom (16 × 16). Here black denotes zero attenuation, and white denotes maximum attenuation. (b) Numbering of the pixels in a 16 × 16 image when interpreted as a vector in $\mathbb{R}^{256}$. (c) Nonzero elements of the 432 × 256 tomographic measurement matrix.

What is the size of A ? Obviously, the number of columns must be 256, the number of pixels in a 16 × 16 image. The `radon.m` algorithm picked automatically the value $N = 27$ in formula (2.45), so A has $JN = 16 \cdot 27 = 432$ rows. We observe that there are 11086 nonzero elements out of the total 110592; this means that roughly 90% of the elements in A are zero.

Three-dimensional X-ray tomography problems can be approached similarly to the above explanation using voxelization instead of pixelization and by tracing the paths of X-rays through the voxels in a three-dimensional manner. However, for illustration and simplicity purposes we stick to the two-dimensional case in this book. In principle there is no essential difference between the two- and three-dimensional cases, only the computations will be significantly more demanding in three dimensions.

2.3.5 Naïve reconstruction

The 16×16 Shepp–Logan phantom used in Section 2.3.4 has too low a resolution to really show the intended anatomic features properly. In the rest of the book we work with the 50×50 Shepp–Logan phantom when we need to construct the matrix A explicitly, and with the 512×512 phantom when we illustrate matrix-free large-scale methods in Section 9. See Figure 2.16.

Next we wish to experiment with naïve reconstructions of the 50×50 Shepp–Logan phantom shown in Figure 2.16(c). We choose the number of projection directions to be $J = 50$ in formula (2.44). We construct the measurement matrix A column by column as explained in Section 2.3.4. MATLAB's `radon.m` algorithm picked automatically the value $N = 75$ in formula (2.45), so A has $JN = 50 \cdot 75 = 3750$ rows. We arrive at the following measurement model:

$$A \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{2500} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{3750} \end{bmatrix}, \quad (2.49)$$

where the elements of the 50×50 pixel image $\mathbf{f}$ and the elements of the 75×50 sinogram $\mathbf{m}$ are numbered similarly to Figure 2.17(b). The phantom and sinogram are shown in Figure 2.18.

(a) 50×50 phantom



(b) Sinogram with 50 angles

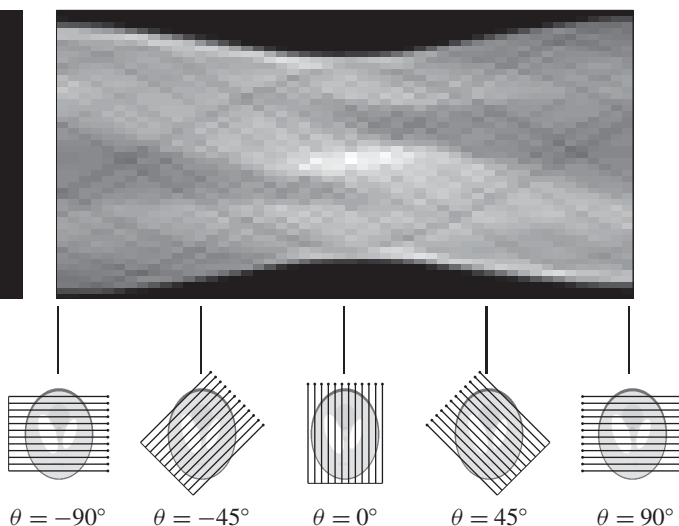


Figure 2.18. (a) Shepp–Logan phantom at resolution 50×50 . Here black denotes zero attenuation, and white denotes maximum attenuation. (b) Measured data (involving inverse crime) in sinogram form, where the horizontal axis is the angle θ and the vertical axis the variable s in (2.40). We have removed some purely zero rows from the top and bottom of the sinogram for clarity. Underneath the sinogram we show some of the projection directions to illustrate the structure of the sinogram.

Once A is in place, we can try out naïve inversion, but not in the sense of (1.4) since A is not a square matrix. Instead we use least-squares naïve inversion defined as follows:

$$\mathbf{f} \approx (A^T A)^{-1} A^T \mathbf{m}. \quad (2.50)$$

Derivation and interpretation of formula (2.50) is postponed to Section 5.2. The result of applying (2.50) to ideal (noise-free) tomographic data is shown in Figure 2.19(b), and it looks very good indeed. The relative error of this reconstruction is very small. Perhaps we can conclude that we succeeded in reconstructing the 50×50 phantom from indirect tomographic measurements?

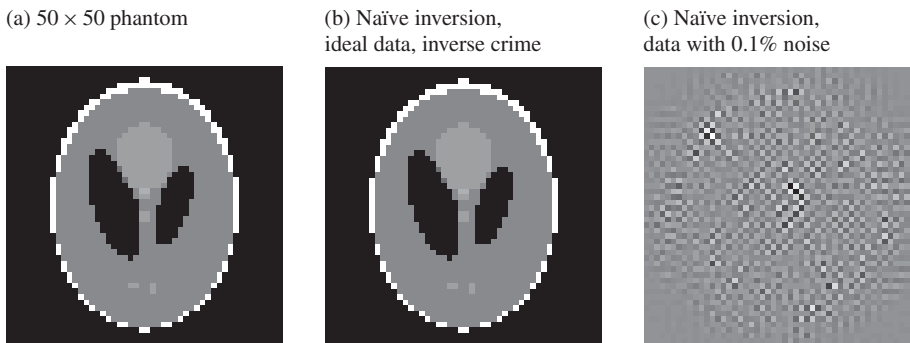


Figure 2.19. (a) Shepp–Logan phantom at resolution 50×50 . (b) Result of naïve inversion (2.50) from noise-free data. The seemingly successful result is not to be trusted because an inverse crime was committed. (c) Result of naïve inversion (2.50) from data contaminated by 0.1% noise. The much worse performance of (c) compared to (b) indicates that naïve inversion is not stable with respect to noise.

Before jumping to such a (wrong) conclusion, let us see what a small amount of measurement noise does to the naïve reconstruction. We add white noise of relative amplitude 0.1% to the sinogram and try formula (2.50) again. The result is shown in Figure 2.19(c), and it consists merely of numerical garbage. This shows that formula (2.50) is not practically useful since real measurements always contain noise.

2.3.6 Naïve reconstruction without inverse crime

We wish to avoid the inverse crime evident in Figure 2.19. To this end, we interpolate our data from tomographic data simulated using the Shepp–Logan phantom on a twice finer grid (100×100) than the grid used in the naïve reconstruction. This can be done conveniently as the phantom is defined analytically using ellipses, so it can be evaluated with arbitrary resolution. The measurement angles are the same. See Figure 2.20 for plots of the data.

Figure 2.21 shows the result of applying naïve reconstruction to the crime-free data. Now the result has unacceptable quality even when there is no added noise.

Exercise 2.3.1. Let $\mathbf{f} \in \mathbb{R}^8$ be a signal and $\mathbf{p} = [\mathbf{p}_{-1} \quad \mathbf{p}_0 \quad \mathbf{p}_1]^T$ a point spread function. Write down the 8×8 matrix A modeling the one-dimensional convolution (2.11) with the assumption that $\mathbf{f}_{j-\ell} = 0$ for the cases $j - \ell < 1$ and $j - \ell > 8$.

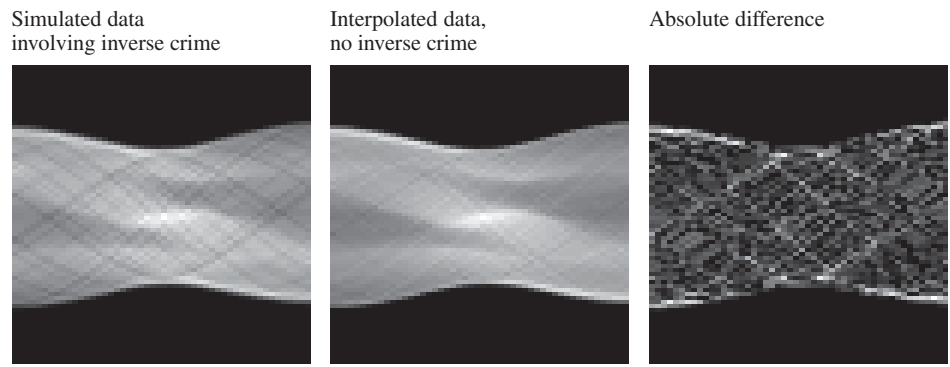


Figure 2.20. Tomographic data with and without inverse crime. Left: ideal data obtained by applying the measurement model matrix to the Shepp–Logan phantom at the final reconstruction resolution of 50×50 pixels. Middle: tomographic data computed from 100×100 Shepp–Logan phantom (at same measurement angles but finer arrangement of X-rays) and interpolated to lower resolution. Right: absolute difference between the two data sets.

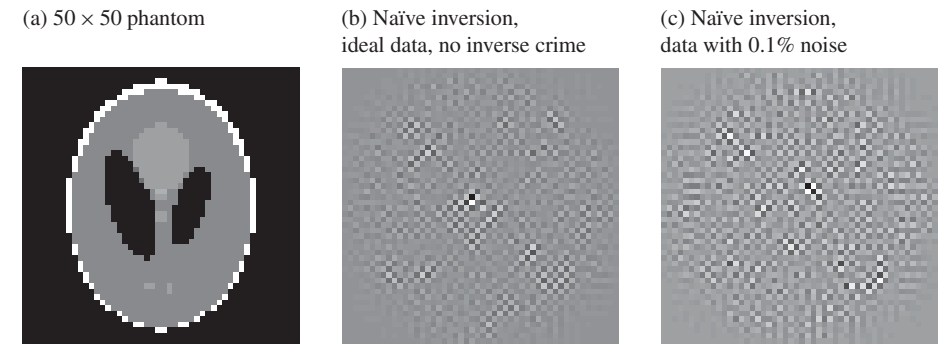


Figure 2.21. (a) Shepp–Logan phantom at resolution 50×50 . (b) Result of naïve inversion (2.50) from noise-free data with no inverse crime. (c) Result of naïve inversion (2.50) from data contaminated by 0.1% noise. Compare to Figure 2.19.

Exercise 2.3.2. Assume that an X-ray detector provides the proportional number Mc instead of the actual photon count c . Here $M > 0$ is a positive constant. Show that the calibration procedure described in Figure 2.12 works fine even if M is unknown.

Exercise 2.3.3. Define f and I_0 and I_1 appropriately in the context of the simple example shown in Figure 2.12. Furthermore, describe the measurement data in Figure 2.12 in terms of formula (2.38).

Exercise 2.3.4. In Figure 2.22, thin lines depict pixels and thick lines X-rays. Give a numbering to the nine pixels ($\mathbf{f} \in \mathbb{R}^9$) and to the six X-rays ($\mathbf{m} \in \mathbb{R}^6$), and construct the matrix A for the measurement model $\mathbf{m} = A\mathbf{f}$. The length of the side of a pixel is one.

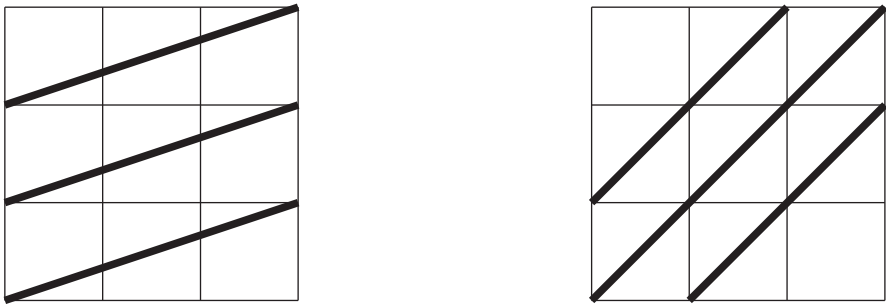


Figure 2.22. Tomographic X-ray measurement configuration related to Exercise 2.3.4. Thick lines depict X-rays, and the length of the side of a pixel is one.

Exercise 2.3.5. Show that the following targets produce exactly the same data in the measurement model (2.48):

4	4	5
1	3	4
1	0	2

5	6	2
1	5	2
4	0	−1

(a) What’s wrong with the negative value -1 above? (b) Can you find more examples that produce the same data but have only nonnegative entries?