Chapter 1 Introduction to Inverse Problems

1.1 Examples of Inverse Problems

The aim of collecting data is to gain meaningful information about a physical system or phenomenon of interest. However, in many situations the quantities that we wish to determine are different from the ones which we are able to measure, or have measured. If the measured data depends, in some way, on the quantities we want, then the data at least contains some information about those quantities. Starting with the data that we have measured, the problem of trying to reconstruct the quantities that we really want is called an *inverse problem.* Loosely speaking, we often say an inverse problem is where we measure an *effect* and want to determine the *cause*.

Here are some typical inverse problems:

- **Computer axial tomography.** Given a patient, we wish to obtain transverse slices through the body *in vivo,* and display pictures of these slices. It is known that the X-rays are partially transmitted through the body, and that the opacity of various internal structures to X-rays varies, so that a picture of the variation of the absorption coefficient in the body would give a good picture. However, the only measurements that one can make non-invasively is to shine X-rays through the patient and to measure the *total* absorption along lines through the body. Given a collection of such line integrals (the "data"), how do we reconstruct the absorption as a function of position in the body (the "image")?

- **Model fitting.** According to some theoretical model, the value of a quantity y depends on another quantity x via an equation such as

$$y = a + bx + cx^2 + dx^3. \tag{1.1}$$

  Given a set of measured points (x_i, y_i) (which are the "data" in this problem), how do we determine the values of a, b, c and d, and how confident are we of the result? In this case the "image" which we wish to determine is the set of numbers a through d. More generally, of course, the model can be more complicated and may depend on the image in a non-linear way. Determining the half-life of a radioactive substance from measurements of the times at which decay products are detected is an example of model fitting.

- **Deconvolution.** Given a blurred photograph, or the result of passing a signal through a medium which acts as a filter, how can we reconstruct an unblurred version of the photograph, or the original signal before the filtering occurred? This type of problem is very important in designing computer modems, for example, because telephone lines will distort signals passing through them, and it is necessary to compensate for these distortions to recover the original signal. The problem of characterizing a linear, shift-invariant system by determining its impulse response is usually a problem in deconvolution.

- **Gridding or regridding.** Suppose that we wish to make a contour map of the height above sea-level of a region. This would be relatively easy if we had measurements of height on a regular grid of points so that the height between these points can be inferred by interpolation. In practice, we usually collect height data at irregularly spaced points with variable density of points in different locations. How do we use such data to reconstruct estimates of height on a regular grid? The problem of drawing isobars on a weather map from isolated barometer readings is essentially the same.

- **Radio-astronomical imaging.** When using a multi-element interferometer as a radio telescope, it turns out that the measured data is not the distribution of radio sources in the sky (called the "sky brightness" function) but is rather the Fourier transform of the sky brightness. It is not possible to measure the entire Fourier transform, but only to sample this transform on a collection of irregular curves in Fourier space. From such data, how is it possible to reconstruct the desired distribution of sky brightness?

- **Navigation.** When travelling in a boat or plane, it is useful to have an idea of the current location in close to real time. This is often done by making a variety of measurements, for example by using bearings to landmarks, stellar or satellite positions, and also by considering one's previous position and using information such as records of speed and heading. How should all of these separate pieces of information be combined together to give a coherent description of the vessel's motion?

- **Image analysis.** How does one automatically count the number of stars in a photograph of the sky, or the number of red blood cells in a microphotograph of a slide of a blood sample? The objects will generally overlap or may be of a variety of shapes. In these cases, the "data" is, typically, a picture of a scene containing the objects to be counted and the inverse problem is to find the number of objects. Closely related is the problem of image segmentation; A typical example is the problem of classifying regions of a satellite image of the earth's surface into regions of ocean, forest, agricultural land, etc.
- **Numerical analysis.** Solution of integral equations such as the Fredholm equation of the first kind

$$\int_a^b k(x,s)\, z(s)\, \mathrm{d}s = y(x), \qquad c \le x \le d, \tag{1.2}$$

  where the kernel k and the function y are given, may be treated as an inverse problem for the unknown function z. The special case where $k(x,s) = u(x-s)$ (u is the unit step) is the problem of numerical differentiation of the function u.
- **Geophysics.** Inverse problems have always played an important role in geophysics as the interior of the Earth is not directly observable yet the surface manifestation of waves that propagate through its interior are measurable. Using the measurements of seismic waves to determine the location of an earthquake's epicentre, or the density of the rock through which the waves propagate, are typical of inverse problems in which wave propagation is used to probe an object. Like many classes of inverse problems, "inverse eigenvalue problems" were first investigated in geophysics when, in 1959, the normal modes of vibration of the Earth were first recorded and the modal frequencies and shapes were used to learn about the structure of the Earth in the large.

From this very short and incomplete list, it is apparent that the scope of inverse problem theory is extensive and its applications can be found in many diverse fields. In this course, we shall be discussing various *general* methods for approaching such problems.

In this introductory chapter, we shall consider one of these problems – deblurring of a photograph – and highlight the ways in which it is representative of other inverse problems.

1.2 Image Space, Data Space, and Noise

In accordance with convention, the collection of values that we want to reconstruct is referred to as the *image*, even if those values do not represent a picture but are simply parameters that define a model. The set of all images is called *image space*. We usually denote the image by f .

The *forward problem* is the mapping from the image to the quantities that we are able to measure. In most of the examples that we consider, the details of the forward problem is given by some physical theory. For example, given the half-life of a radioactive substance, nuclear physicists can tell us how to calculate the time at which we will detect the decay products – at least in a statistical sense. The forward mapping may be linear or nonlinear and is denoted by A.

In practice we are never able to make exact measurements and the *data* that we measure are a corrupted version of the measurement quantities. *Data space* is the set of all possible data. The corruption could be as small as the roundoff error produced by a computer representation of the measurements, it could be intrinsic in the measurement process such as the twinkle of star brightness produced by a turbulent atmosphere, or, more usually, the corruption is due to the inherent errors in the measurement process. So, strictly, the forward process is a mapping from the image to error-free data, $\bar{d}$, and the data we actually measure, d, is the corrupted form. The difference $\bar{d} - d$ is called the *noise* which we denote by n.

Thus the mapping from the image to actual data is given by the relation

$$d = A(f) + n.$$

The *inverse problem* is then the problem of finding the original image given the data and knowledge of the forward problem.

For example, in the case of deblurring photographs, the "image" is the sharp photograph, the "data" is the blurred photograph, and the forward problem is the blurring process. The inverse problem is to find the sharp photograph (image) from the blurred photograph (data) and knowledge of the blurring process.

1.3 Ill-Posed Problems and Ill-Conditioning

There are several basic classifications of the forward problem depending on whether the image and data are functions of a continuous or discrete variable, i.e., are infinite-dimensional or finite-dimensional. These could be classified as:

	Image	-	Data
a)	continuous	-	continuous
b)	continuous	-	discrete
c)	discrete	-	continuous
d)	discrete	-	discrete

The Fredholm equation 1.2 is an example of a continuous-continuous forward problem since both the functions $y(x)$ and $z(s)$ are defined on an interval (and require an infinite number of values to define them). The model-fitting problem in equation 1.1 is a case where the image is discrete (defined by the 4 values a, b, c, and d) and if there are a finite number of data values it gives a discrete-discrete problem

In practice, we can only ever measure a finite number of data values and so case a) is always an idealization of an actual problem, though it is one that is commonly used in Physics as it can lead to a simplified analysis. Ultimately, computer implementation is necessarily purely discrete, and for these purposes, each of cases a) through c) is approximated by a problem falling in class d).

Whichever class the mapping A belongs to, the inverse problem of solving

$$A(f) = d \tag{1.3}$$

for f given d is called *well-posed* (by Hadamard in 1923) if:

1. a solution *exists* for any data d in data space,
2. the solution is *unique* in image space, and
3. the inverse mapping $d \mapsto f$ is *continuous.*

Conditions 1 and 2 are equivalent to saying that the operator A has a well defined inverse A^{-1} and that the domain of A^{-1} is all of data space.

The requirement in condition 3 of continuous dependence of the solution image on the data is a necessary but not sufficient condition for the stability of the solution. In the case of a well-posed problem, relative error propagation from the data to the solution is controlled by the *condition number*: if Δd is a variation of d and Δf the corresponding variation of f, then

$$\frac{||\Delta f||}{||f||} \leq \text{cond}(A) \frac{||\Delta d||}{||d||}$$

where (for linear forward problems)

$$\text{cond}\,(A) = ||A|| \, \left|\left|A^{-1}\right|\right|.$$

If you have not met the concept of the norm of an linear operator (or transformation) before, the formal definition is

$$||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}.$$

The quantity on the right hand side is in terms of (usual) vector norms and measures by how much the transformation "stretches" a vector x (in general, of course, A will also change the direction of the vector, so that Ax and x do not point in the same "direction"). The norm of the operator is found by considering the stretching factor for all non-zero vectors x, and finding the largest such factor. (*Technical note:* Mathematically, we use the supremum rather than the maximum operation in the definition in case the maximum is not achieved by any non-zero vector x. In such cases there is a sequence of nonzero x_n with the property that $||Ax_n|| \, / \, ||x_n||$ increases asymptotically to $||A||$.)

Since the fractional error in f equals the condition number multiplied by the fractional error in d, smaller values of cond(A) are desirable. In the linear case, cond(A) ≥ 1, and the case cond(A) $= 1$ occurs when A is similar to a multiple of the identity. If cond(A) is not too large, the problem 1.3 is said to be *well-conditioned* and the solution is stable with respect to small variations of the data. Otherwise the problem is said to be *ill-conditioned*. It is clear that the separation between well-conditioned and ill-conditioned problems is not very sharp and that the concept of well-conditioned problem is more vague than the concept of well-posed problem.

Hadamard went on to define a problem to be *ill-posed* if it does not satisfy all three conditions. So an ill-posed problem is one where an inverse does not exist because the data is outside the range of A, or the inverse is not unique because more than one image is mapped to the same data, or because an arbitrarily small change in the data can cause an arbitrarily large change in the image.

Hadamard believed (many pure mathematicians still do) that ill-posed problems were actually incorrectly-posed and "artificial" in that they would not describe physical systems. He was wrong. Most correctly stated inverse problems turn out to be ill-posed; In fact all of the examples in section 1.1 are examples of actual problems in which the inverse problem is ill-posed or at least ill-conditioned. The facts that CAT scans are performed successfully every day, or that oil reservoirs have been found by seismic investigation, is evidence that meaningful information can be gained from ill-posed inverse problems even though they cannot be strictly inverted.

The classical example of an ill-posed problem is a Fredholm integral equation of the first kind with a square integrable (Hilbert-Schmidt) kernel

$$\int_a^b k\,(x,s)\, f\,(s)\ \mathrm{d}s = d\,(x)\,, \qquad a \leq x \leq b. \tag{1.4}$$

If the solution f is perturbed by $\Delta f\,(s) = \epsilon \sin(2\pi p s)$, $p = 1, 2, \ldots$, $\epsilon =$ constant, then the corresponding perturbation of the right-hand side $d\,(x)$ is given by

$$\Delta d\,(x) = \epsilon \int_a^b k\,(x,s) \sin(2\pi p s)\, \mathrm{d}s, \qquad p = 1, 2, \ldots\,,$$

and due to the Riemann-Lebesgue lemma it follows that $\Delta d \to 0$ as $p \to \infty$. Hence, the ratio $\frac{||\Delta f||}{||\Delta d||}$ can become arbitrarily large by choosing the integer p large enough[1], thus showing that 1.4 is an ill-posed problem because it fails condition 3. In particular, this example illustrates that Fredholm integral equations of the first kind with square integrable kernels are extremely sensitive to high-frequency perturbations.

Strictly speaking, a problem that is ill-posed because it fails condition 3 must be infinite dimensional – otherwise the ratio $\frac{||\Delta f||}{||\Delta d||}$ stays bounded, although it may become very large. However, certain finite-dimensional discrete problems have properties very similar to these ill-posed problems, such as being highly sensitive to high-frequency perturbations and so we refer to them as (discrete) ill-posed problems.

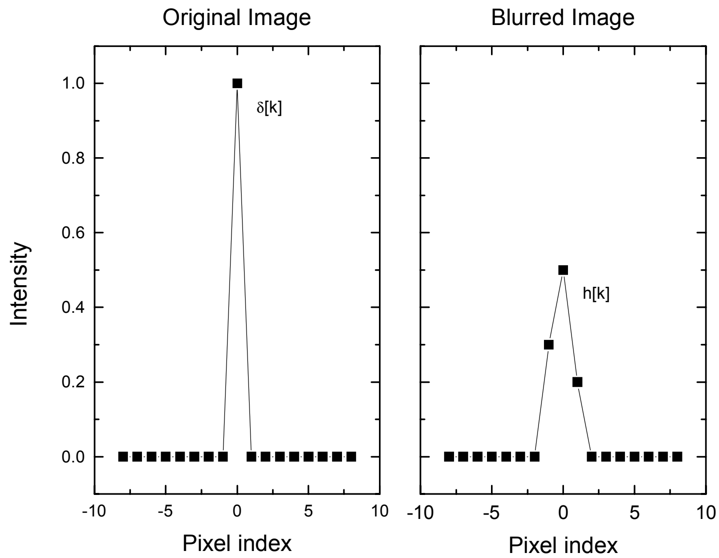
[1] Students of 445.763 will have recognised that the forward operator in this case is a Hilbert-Schmidt integral operator and hence is "compact". It follows that its inverse cannot be bounded.

1.4 The Deconvolution Problem

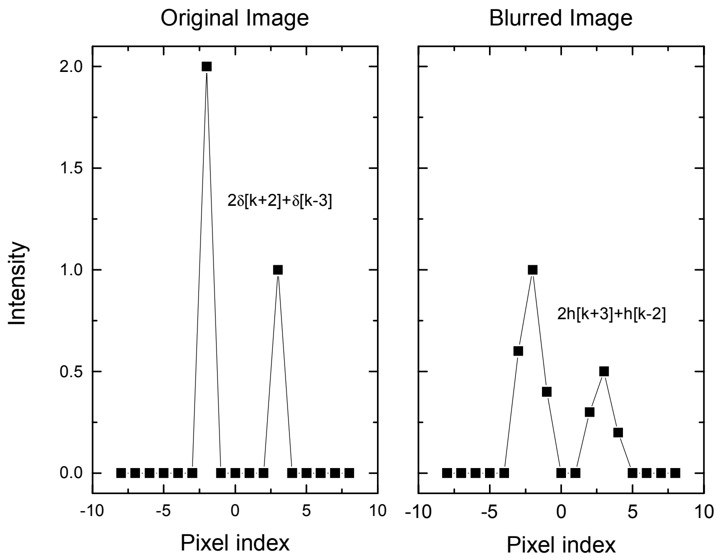
Let us consider the deconvolution or deblurring problem. The desired quantity in this case is a sampled signal $x[k]$ evaluated on a grid of regularly-spaced times or an image, $x[k,l]$ represented its intensity values on a regular array of pixels. The quantity we can measure is a filtered version of this signal or a blurred version of the image. Given the measured data, how do we reconstruct the image?

1.4.1 Modelling the forward problem

In this step we model how the image is turned into the measurements by the measurement process. For simplicity, we initially consider a discrete one-dimensional "image" which is simply a sequence of numbers representing intensity as a function of position. Suppose first that the image consists of a single bright point in the centre of a dark background. Mathematically, this is represented by a sequence $\delta[k]$ which is all zeros except at $k = 0$ where it takes on the value one. After the image is blurred, this single bright point becomes spread out into a region called the *point-spread function.* The graph below shows an example of a point spread function which is represented by the sequence $h[k]$.



If the blurring is *linear* and *spatially invariant,* an image which consists of two unequally bright points, one at $k = -2$ and the other at $k = 3$ which may be represented by the sequence $2\delta[k+2] + \delta[k-3]$ will be blurred into $2h[k+2] + h[k-3]$ as shown in the diagram below



We may readily generalize this blurring process to see that if the original image $x[k]$ is a linear combination of shifted δ sequences

$$x[k] = \sum_m c_m \delta[k-m], \tag{1.5}$$

then the blurred image $y[k]$ will be the *same* linear combination of shifted point-spread functions, i.e.,

$$y[k] = \sum_m c_m h[k-m]. \tag{1.6}$$

We now consider the problem of determining the coefficients c_m. This is easy once we realize that the δ sequence in (1.5) collapses the summation so that $c_k = x[k]$. Thus we may write the blurred image (1.6) as

$$y[k] = \sum_m x[m]\, h[k-m], \tag{1.7}$$

which we recognize simply as the *convolution* of the sequences x and h denoted $x * h$. Thus the process of convolution corresponds to the simple idea of spatially-invariant blurring. In Matlab, the function `conv` calculates the convolution of two sequences. For sequences of finite length, the convolution of a sequence of length M and a sequence of length N yields a sequence of length $M + N - 1$.

In two dimensions (e.g., for a photograph), images and point spread functions are represented by sequences with two indices. For example $x[m,n]$ denotes the intensity of the pixel in row m and column n of an image. The convolutional relationship readily generalizes in this situation to

$$y[k,l] = \sum_m \sum_n x[m,n]\, h[k-m, l-n]. \tag{1.8}$$

and again we write $y = x * h$.

1.5 Transformation into Fourier space

The process of convolution of two sequences is a moderately complicated process involving many additions and multiplications. Recall from the theory of Fourier transforms that we defined the convolution of two functions of t as

$$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, \mathrm{d}\tau \tag{1.9}$$

and we discovered that if we write the Fourier transforms of $x(t)$ and $h(t)$ as $X(\nu)$ and $H(\nu)$ respectively, where for example

$$X(\nu) = \int_{-\infty}^{\infty} x(t) \exp(-\mathrm{j}2\pi\nu t)\, \mathrm{d}t, \tag{1.10}$$

then the Fourier transform of $(x * h)$ is simply the product $X(\nu)\, H(\nu)$. Thus the operation of convolution in the t domain corresponds to multiplication of functions in the Fourier domain. The mapping from the function x to the data is then $d = Ax+n$, where A is the convolution $Ax \equiv x*h$, and n is an unknown function representing noise. The Fourier transform of this relation gives the relation between Fourier transforms:

$$D(\nu) = X(\nu)\, H(\nu) + N(\nu). \tag{1.11}$$

If the $H(\nu)$ is never zero, i.e., the forward problem is invertible, a straightforward inversion scheme is to calculate an estimate of X by

$$\hat{X}(\nu) = \frac{D(\nu)}{H(\nu)} \tag{1.12}$$

and an estimate of the function x is $\hat{x}$, the inverse Fourier transform of $\hat{X}$.

We wish to see if we can carry out a similar process using sequences. Since we will want to represent these sequences on a computer, we shall consider sequences of finite length N. Given a sequence $x[k]$ of length N, we shall assume that the index k ranges from 0 to $N-1$.

The *finite Fourier transform* of a sequence of length N is defined to be another sequence of length N. We shall use the convention of denoting the Fourier transform by the upper-case letter corresponding to the lower-case letter of the original sequence. For example, the finite Fourier transform $X[r]$ of $x[k]$ is defined by

$$X[r] = \sum_{k=0}^{N-1} x[k] \exp\left(-\frac{\mathrm{j}2\pi rk}{N}\right) \qquad \text{for } r = 0, 1, ..., N-1. \tag{1.13}$$

With this definition, we can recover $x[k]$ from $X[r]$ by the *inverse finite Fourier transform*

$$x[k] = \frac{1}{N} \sum_{r=0}^{N-1} X[r] \exp\left(\frac{\mathrm{j}2\pi rk}{N}\right) \qquad \text{for } k = 0, 1, ..., N-1. \tag{1.14}$$

You should be able to show that these relationships are indeed inverses of each other by using the fact that

$$\sum_{k=0}^{N-1} \exp\left(\frac{\mathrm{j}2\pi rk}{N}\right) = \begin{cases} N & \text{if } r \bmod N = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{1.15}$$

which may simply be demonstrated by summing the geometric series.

Note that the normalizations in front of the forward and inverse transforms are different. The above conforms to the convention adopted in Matlab. An alternative convention which has the advantage that $X[0]$ is the average of the sequence $x[k]$ is to place the factor $1/N$ in front of the forward transform and have a factor of unity in the inverse transform. Yet another convention which leads to a more symmetrical forward and inverse transform is to place a factor of $1/\sqrt{N}$ in front of both transforms.

Consider now the product of two finite Fourier transforms:

$$\begin{aligned}X[r]H[r] &= \sum_{k=0}^{N-1} x[k]\exp\left(-\frac{j2\pi rk}{N}\right)\sum_{l=0}^{N-1} h[l]\exp\left(-\frac{j2\pi rl}{N}\right) \\ &= \sum_{k=0}^{N-1}\left\{\begin{array}{c}\sum_{l=0}^{N-1-k} x[k]h[l]\exp\left(-\frac{j2\pi r(k+l)}{N}\right) \\ +\sum_{l=N-k}^{N-1} x[k]h[l]\exp\left(-\frac{j2\pi r(k+l)}{N}\right)\end{array}\right\} \\ &= \sum_{k=0}^{N-1}\left\{\begin{array}{c}\sum_{m=k}^{N-1} x[k]h[m-k]\exp\left(-\frac{j2\pi rm}{N}\right) \\ +\sum_{m=0}^{k-1} x[k]h[m-k+N]\exp\left(-\frac{j2\pi rm}{N}\right)\end{array}\right\} \\ &= \sum_{m=0}^{N-1}\left(\sum_{k=0}^{N-1} x[k]h[(m-k)\bmod N]\right)\exp\left(-\frac{j2\pi rm}{N}\right)\end{aligned}$$

where the substitutions $l = m - k$ and $l = m - k + N$ were used to go from the second to third line. The term

$$(x \circledast h)[m] = \sum_{k=0}^{N-1} x[k]h[(m-k)\bmod N]$$

is the *circular convolution*[2] of the sequences x and h. Hence the product of finite Fourier transforms is the finite Fourier transform of the circular convolution of the original sequences.

Note that the wrap-around nature of the sum in the circular convolution can be understood in terms of the continuous Fourier transform. Since the finite Fourier transform corresponds to the continuous Fourier transform when both the original function and the transform are sampled and periodically repeated, the wrap-around comes from *spatial aliasing* resulting from the sampling in (spatial) frequency. The circular convolution can be used to calculate the usual convolution by first zero-padding the sequences x and h to twice their original length to avoid the aliasing. Setting

$$x_p[k] = \begin{cases} x[k], & k = 1, 2, \ldots, N-1 \\ 0, & k = N, N+1, \ldots, 2N-1 \end{cases}$$

and similarly for h_p we can see that

$$X_p[r]H_p[r] = \sum_{m=0}^{2N-1} (x * h)[m]\exp\left(-\frac{j2\pi rm}{2N}\right)$$

is the finite Fourier transform (of length $2N$) of the desired convolution. The convolution may then be found by taking the inverse transform.

In summary, the mapping from the sequence x to the data is then $d = Ax + n$, where A is the convolution with the point-spread function $Ax \equiv x * h$, and n is the unknown noise sequence. Equivalently, we have the the relation between finite Fourier transforms of the zero-padded sequences:

$$D_p[r] = X_p[r]H_p[r] + N_p[r].$$

If the sequence $H_p[r]$ is never zero then the forward problem is invertible and an inversion scheme, analagous to the continuous case, is to calculate an estimate of X_p by

$$\hat{X}_p[r] = \frac{D_p[r]}{H_p[r]}$$

and an estimate of the sequence x is the first half of the sequence $\hat{x}_p$ calculated as the inverse finite Fourier transform of $\hat{X}_p$. So, as in the continuous case, the forward mapping may be inverted by division in the Fourier domain. This process is often called *deconvolution* as it undoes the operation of the convolution with the point-spread function.

[2] The circular reference comes from the observation that the sum can be calculated as the sum of the products of the two sequences *wrapped* onto a circle with h shifted round by the amount m with respect to x.

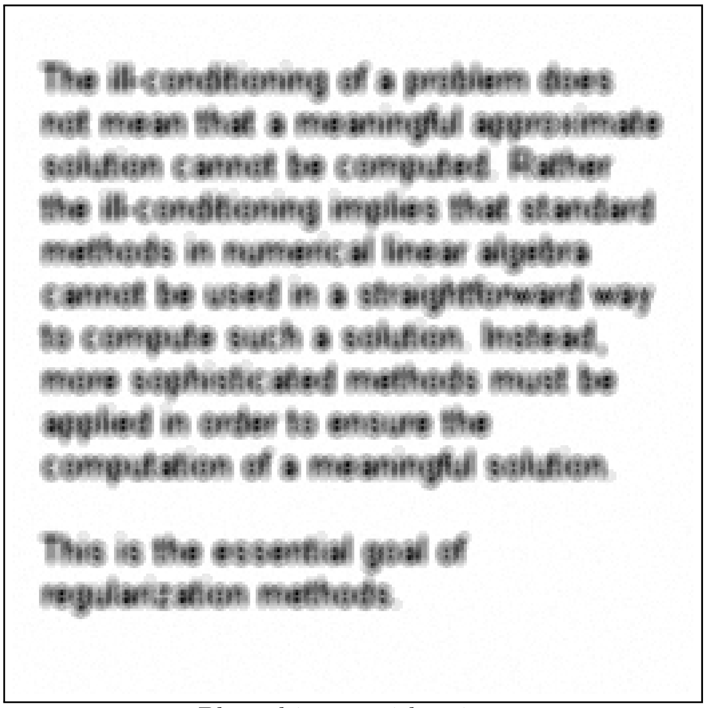
1.6 A Pictorial Example

In the following example, an image of size 256×256 pixels is blurred with a point spread function which is a circle of radius $\sqrt{5}$ pixels. Random noise of r.m.s. amplitude about 1% of the maximum value is added to the blurred image and deconvolution is carried out by Fourier space division.

The original image has pixel values between 0 (background) and 255 (lettering). Usually the background in images is dark and the features in the image are brighter. However, in the following sequence of pictures, the value 0 is displayed as white so that the images are black on a white background; This has been done for ease of viewing and photocopying only.

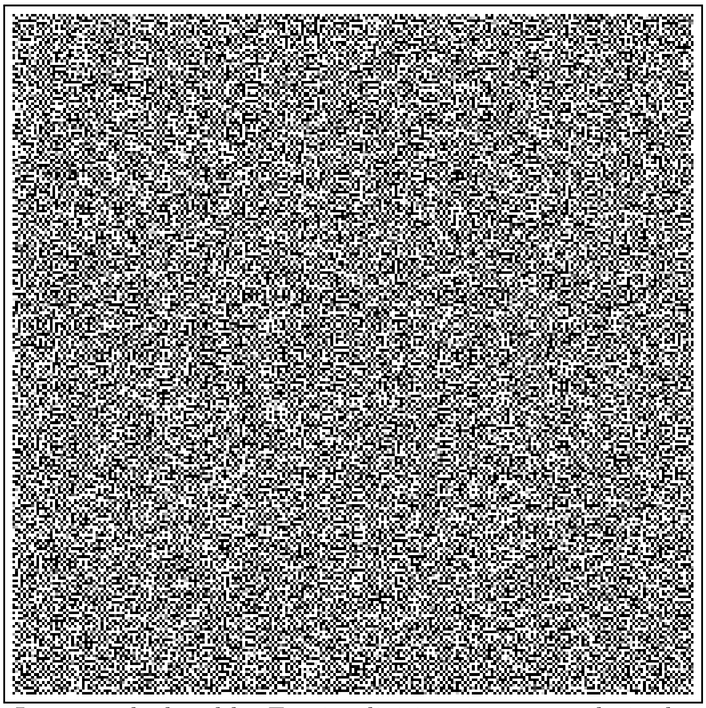
Further, for the particular image and point-spread function chosen, no spatial aliasing takes place when the circular convolution is used to calculate the convolution representing the blurring in the forward process. This is because the original image has the value 0 for a distance around the edge of the image which is wider than $\sqrt{5}$ pixels, i.e., the half-width of the point-spread function. Hence, again for ease of manipulation, the circular convolution is used to model the forward problem – without initial zero-padding.

The following picture shows the blurred image with noise added.



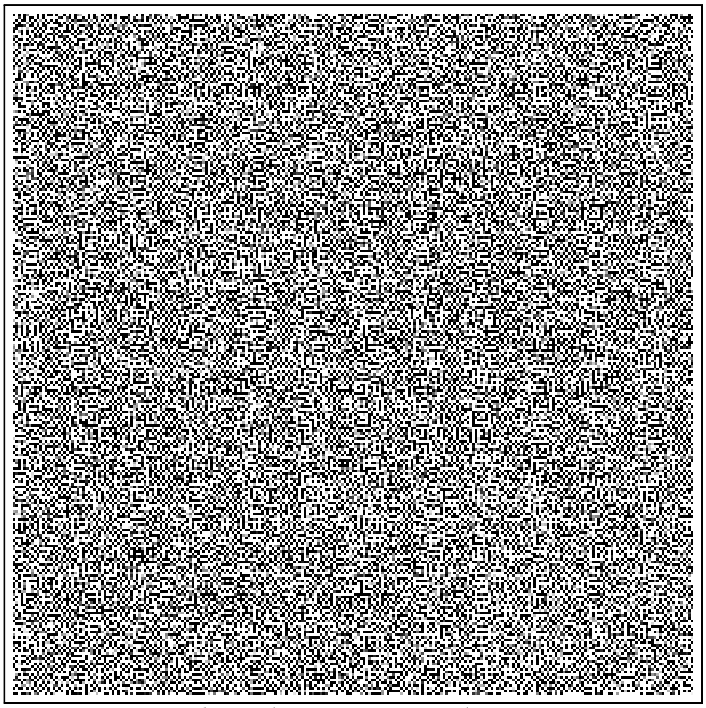
Blurred image with noise

It happens in this case that the forward mapping is invertible, so we can calculate the inverse image of the blurred image by division of the Fourier transforms. The result is shown in the following picture.



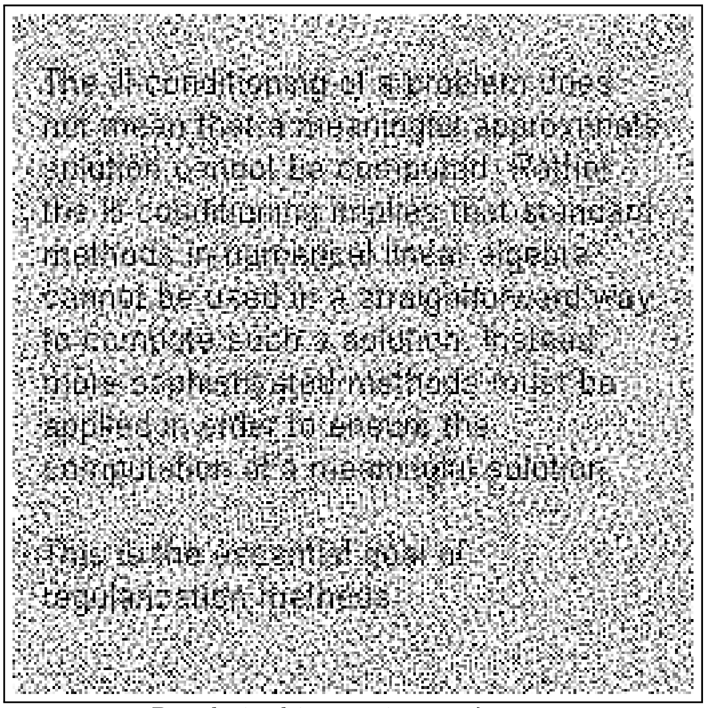
Inverse calculated by Fourier division, i.e., unregularized solution.

Note that the direct inverse does not give a useful reconstruction. One of the methods that we will investigate for reconstructing the image from noisy (and incomplete) data is 'regularization' in which we trade-off the accuracy of inverting the forward problem against a requirement that the reconstruction be 'smooth'. The details of that method are given later, but for the moment it is sufficient to know that the trade-off is controlled by the parameter λ; increasing λ makes the reconstruction smoother and the inversion more approximate. A small amount of regularization is achieved by setting $\lambda = 0.1$ and the result is the following picture.



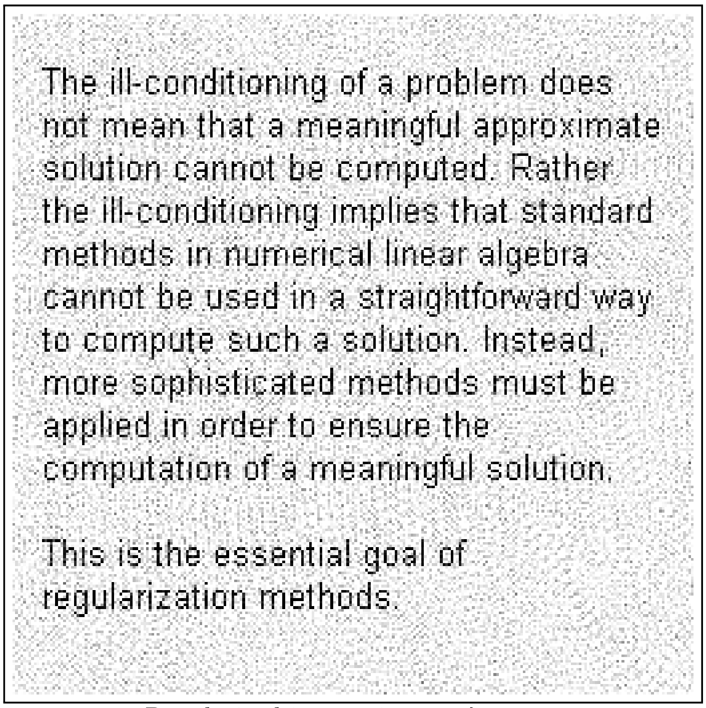
Regularized inverse image: $\lambda = 0.1$

That reconstruction is effectively the same as the first given by exact inversion, which is the case $\lambda = 0$ so there is no regularization. Greater regularizing is achieved by setting $\lambda = 1$,



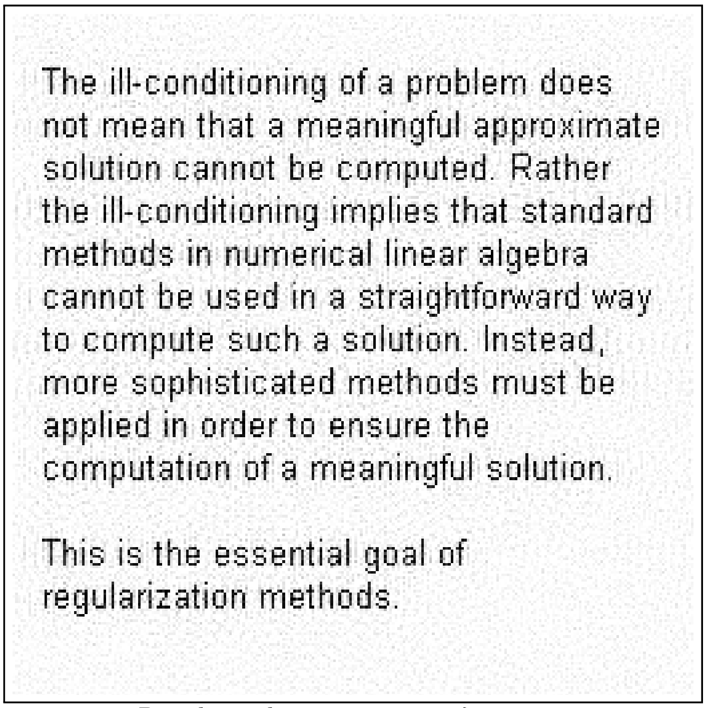
Regularized inverse image: $\lambda = 1$

and more still with $\lambda = 10$.



Regularized inverse image: $\lambda = 10$

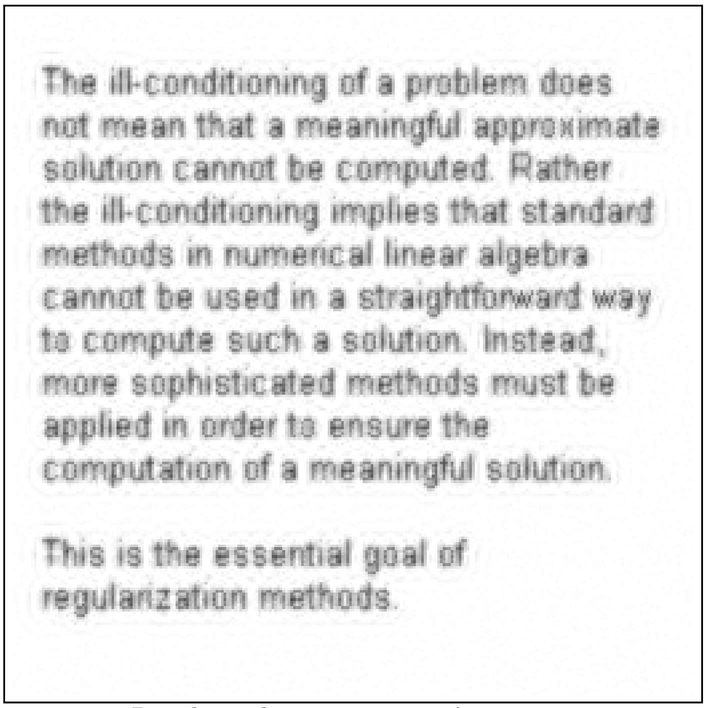
In terms of the L-curve, also covered later, $\lambda = 10$ is close to the optimum value. However, a bit clearer reconstruction is achieved using $\lambda = 100$.



Regularized inverse image: $\lambda = 100$

As you can see, the resulting reconstruction is perfectly readable. Note that this image has resulted from not inverting the forward mapping in the strict sense, but rather an approximation to the forward mapping. In doing so we have produced a result which is useful though not correct.

Regularizing further by setting $\lambda = 1000$ smooths more than is necessary and results in the following overly smooth reconstruction.



Regularized inverse image: $\lambda = 1000$

1.7 What Went Wrong with the Inverse?

By expressing the deblurring example of the previous section in the Fourier domain, the forward and inverse mapping were shown to have the simple structure of componentwise multiplication and division, respectively.

In the continuous image – continuous data case, the mapping from image to data was given in equation 1.11 as $D(\nu) = X(\nu) H(\nu) + N(\nu)$, so each Fourier component in the image is multiplied by a (complex) scalar and is corrupted by (complex) scalar valued noise to give the coefficient of a single Fourier component in the data. Similarly, the straightforward inverse of equation 1.12, $\hat{X}(\nu) = \frac{D(\nu)}{H(\nu)}$, is a scalar equation for each component. Combining these two equations we find that

$$\hat{X}(\nu) = \frac{X(\nu) H(\nu) + N(\nu)}{H(\nu)} = X(\nu) + \frac{N(\nu)}{H(\nu)},$$

i.e., the reconstruction is equal to the original image plus the term $\frac{N(\nu)}{H(\nu)}$. If $|H(\nu)|$ is small enough (or worse still zero) that term will be large, in fact larger than $X(\nu)$, and that component of the reconstruction will be principally determined by the unknown noise value $N(\nu)$. For the case of blurring, $H(\nu)$ necessarily has arbitrarily small (or zero) values whenever the point-spread function $h(t)$ is square integrable since then, by Parseval's identity, so is $H(\nu)$ and hence $H(\nu) \to 0$ as $\nu \to \infty$. Thus, high frequency components in the reconstruction will necessarily be determined by the details of the unknown noise. Note that the forward mapping for blurring of a finite-sized picture is a Fredholm integral equation and the observation of high-frequency difficulties in the inverse is exactly the same as reasoned in section 1.3.

Thus, the continuous image – continuous data deblurring problem is ill-posed because if the inverse exists it cannot be continuous. Of course, there will not be an inverse at all if $H(\nu)$ is zero for any values of ν

(the division by $H(\nu)$ is undefined) in which case the beblurring problem is ill-posed because it fails the first condition for being well-posed. However, if $H(\nu)$ does take the value zero for some ν, and we are able to make *perfect noise-free measurements*, then the corresponding value of the data $D(\nu)$ is also zero and so we can find an inverse image of the data by chosing any value we please for $X(\nu)$. But then the inverse image is certainly not unique and the problem fails the second condition for being well-posed.

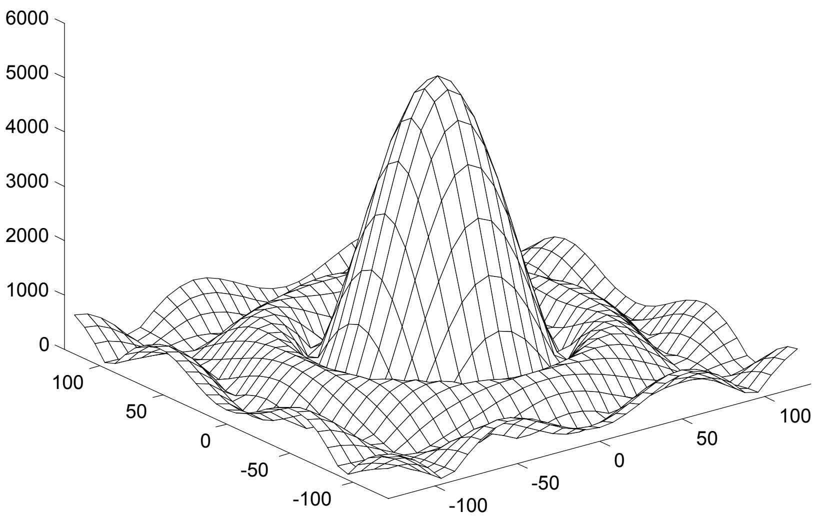
The discrete-discrete deblurring example has many of the same properties. Once again the inverse of the forward problem leads to a solution by Fourier division: $\hat{X}_p[r] = \frac{D_p[r]}{H_p[r]}$. In this case the Fourier transform $H_p[r]$ has a finite number of values and so, assuming the forward problem is invertible, $H_p[r]$ has a smallest non-zero value. However, if that value is small enough, the corresponding component in the reconstruction, given by

$$\hat{X}_p[r] = X_p[r] + \frac{N_p[r]}{H_p[r]},$$

is dominated by the noise, and the data effectively give no information about the corresponding component in the image.

In the presence of noise, it makes no difference whether a value of $H_p[r]$, or $H(\nu)$, is very small or actually zero. In one case the forward mapping is invertible and in the other it is not, yet the data are effectively the same. We must conclude that the issue of invertibility of the forward problem, *per se*, is irrelevant when considering the practical inverse problem.

The blurring function used in the pictorial example was a circle of radius $\sqrt{5}$ pixels in a 256×256 pixels image. The magnitude of the finite Fourier transform is shown in the following figure.



Magnitude of finite Fourier transform of the point-spread function.

From the picture it can be seen the the Fourier transform does indeed have values that are close to zero and so we expect that inverse will show the symptoms of being ill-posed. The minimum and maximum magnitudes of the transform turn out to be 0.071 and 5355, respectively. So the forward problem is invertible, and strictly well-posed, but it is not well-conditioned with a condition number of 7.5×10^4 compared to the noise level of about 1%.