

ENGSCI 721

INVERSE PROBLEMS

Oliver Maclaren
oliver.maclaren@auckland.ac.nz

MODULE OVERVIEW

3. Preview of the statistical view of inverse problems

[1 lectures]

Bayesians, Frequentists and all that. Basic frequentist analysis.

MODULE OVERVIEW

Inverse Problems (Oliver Maclaren) [~8 lectures/2 tutorials]

1. Basic concepts [4 lectures]

Forward vs inverse problems. Well-posed vs ill-posed problems. Algebra and calculus of inverse problems (matrix calculus, generalised inverses etc). Regularisation and trade-offs.

2. More regularisation [3 lectures]

Higher-order Tikhonov regularisation, truncated singular value decompositions, iterative regularisation.

LECTURE 1: OVERVIEW

Topics:

- Forward vs inverse problems
- Examples
- Well-posed vs ill-posed problems
- Illustrations of ill-posedness

Eng Sci 721 : Inverse Problems & UQ (A brief intro !)

Lecture 1: Overview

- Inverse problems & UQ.
- Forward vs inverse problems
- Examples of inverse problems
- Well-posed & ill-posed problems
- Illustrations of ill-posedness

Prelude:
Inverse Problems, UQ, Data-centric engineering, scientific machine learning--

The imperative of physics-based modeling and inverse theory in computational science

To best learn from data about large-scale complex systems, physics-based models representing the laws of nature must be integrated into the learning process. Inverse theory provides a crucial perspective for addressing the challenges of ill-posedness, uncertainty, nonlinearity and under-sampling.

Karen E. Willcox, Omar Ghattas and Patrick Heimbach

The notions of 'artificial intelligence (AI) for science' and 'scientific machine learning' (SciML) are gaining widespread attention in the scientific community. These initiatives target development and adoption of AI approaches in scientific and engineering fields with the goal of accelerating research and development breakthroughs in energy, basic science, engineering, medicine and national security. For the past six decades, these fields have been advanced through the synergistic and principled use of theory, experiments and physics-based simulations. Our increased ability to sense and acquire data is clearly a game-changer in these endeavors. Yet, in our excitement to define a new generation of data-centric approaches, we must be careful not to chart our course based entirely on the successes of data science and machine learning in the vastly different domains of social media, online entertainment, online retail, image recognition, machine translation and natural language processing — domains for which data are plentiful and physics-based models do not exist. In contrast, many of today's scientific grand challenges suffer from the lack of adequate sampling of the processes underlying the complex, large-scale systems. Yet, for many of these systems, a great deal is known regarding the underlying physical principles or governing equations; we must continue to appeal to computational science to unleash this information. As Coveney et al. argue elegantly, big data need big theory — and big physics-based simulation models — too.

The unreasonable effectiveness of physics-based models
But what are physics-based models and why are they indispensable? A physics-based model is a representation of the governing laws of nature that innately embeds the concepts of time, space, causality and generalizability. These laws of nature define how physical, chemical, biological and

geological processes evolve. Physics-based models typically encode knowledge in the form of conservation and constitutive laws, often based on decades if not centuries of theoretical development and experimental validation. These laws often manifest as systems of differential equations that are solved numerically with high-performance computing (HPC).

In his famous 1960 article, Eugene Wigner wrote about 'The unreasonable effectiveness of mathematics in the natural sciences', pointing to 'the "laws of nature" being of almost fantastic accuracy but of strictly limited scope.' As Wigner discusses, physics-based modeling is powerful and effective because it gives us a predictive window into the future based on understanding. It achieves this because any particular model limits its scope to a particular class of physical systems or processes, building a universal representation within that class. Armed with that universal representation, physics-based modeling is a way to simulate 'what if' scenarios and to issue predictions that have explanatory power or projections with quantified uncertainties that go beyond the current state and available data. For example, in our modern world, physics-based models are used to issue predictions about the future evolution of a cancer patient's tumor, or about the loads that a yet to be built aircraft may find itself experiencing under different operating conditions. They enable predictions of weather over the next five to ten days, or scenario-based projections about the future state of the Earth's climate in the decades to come.

The role of inverse theory in learning from data
As attention turns from simulation to learning from data (that is, from the forward problem to the inverse problem), we must bring these learned lessons — the big theory and the big physics-based simulation models — with us. Without physical

constraints, purely data-driven approaches are unlikely to be predictive, no matter how expressive the underlying representation.

Even when physical models are not well-established (such as for many biological processes, in constitutive laws for complex materials, or in subgrid scale models for unresolved physics), we know that certain universal properties and relationships must hold, such as conservation properties, material frame indifference, objectivity, symmetries, or other invariants. The learning-from-data problem is fundamentally an inverse problem that merges the partial knowledge reservoir of data with that of physics-based models in a systematic and rigorous way, and in a way that exploits the complementary and mutually reinforcing aspects of both data and models.

Data and models invariably come with uncertainty. Data are often noisy, sparsely and heterogeneously sampled, and representative of disparate observables. Experiments and data gathering are costly, time-consuming, and sometimes dangerous or impossible. Often data are hardest to acquire and are thus sparsest in the most decision-critical regions (for example, failure, instability, extreme environments). Even if it is possible to generate more data (for example, via simulation) a fundamental challenge remains: due to information loss in the forward problem and resulting ill-posedness of the inverse problem, data often contain only low-dimensional information about the physics, even when the data are large-scale.

In turn, physics-based models are typically characterized by uncertain parameters, which may include initial and boundary conditions, sources, material properties, geometry and model structure, all of which can be heterogeneous in space or time. In this setting, rather than ignore known physics, we must employ them to determine the maps from parameters to observables, and invert them to project

Statistics? (Frequentist? Bayesian? ...)

Inverse problems as statistics

Steven N Evans¹ and Philip B Stark²

¹ Department of Statistics and Department of Mathematics, University of California, Berkeley, CA 94720-3860, USA

² Department of Statistics, Space Sciences Laboratory, and Center for Theoretical Astrophysics, University of California, Berkeley, CA 94720-3860, USA

E-mail: evans@stat.berkeley.edu and stark@stat.berkeley.edu

Received 27 September 2001, in final form 26 February 2002

Published 20 May 2002

Online at stacks.iop.org/IP/18/R55

Abstract

What mathematicians, scientists, engineers and statisticians mean by ‘inverse problem’ differs. For a statistician, an inverse problem is an inference or estimation problem. The data are finite in number and contain errors, as they do in classical estimation or inference problems, and the unknown typically is infinite dimensional, as it is in nonparametric regression. The additional complication in an inverse problem is that the data are only indirectly related to the unknown. Canonical abstract formulations of statistical estimation problems subsume this complication by allowing probability distributions to be indexed in more-or-less arbitrary ways by parameters, which can be infinite dimensional. Standard statistical concepts, questions and considerations such as bias, variance, mean-squared error, identifiability, consistency, efficiency and various forms of optimality apply to inverse problems. This paper discusses inverse problems as statistical estimation and inference problems, and points to the literature for a variety of techniques and results. It shows how statistical measures of performance apply to techniques used in practical inverse problems, such as regularization, maximum penalized likelihood, Bayes estimation and the Backus–Gilbert method. The paper generalizes results of Backus and Gilbert characterizing parameters in inverse problems that can be estimated with finite bias. It also establishes general conditions under which parameters in inverse problems can be estimated consistently.

Motivation/Working principle:

→ The mathematics underlying ‘classical’ inverse problems sheds light on statistics, machine learning etc as well as vice-versa

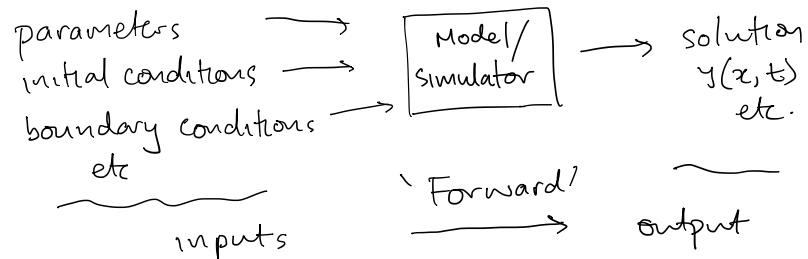
we'll look at both!

statistics : mappings from data to parameter } eg
ML : mappings from data to data

| inverse problems: mappings between general spaces |

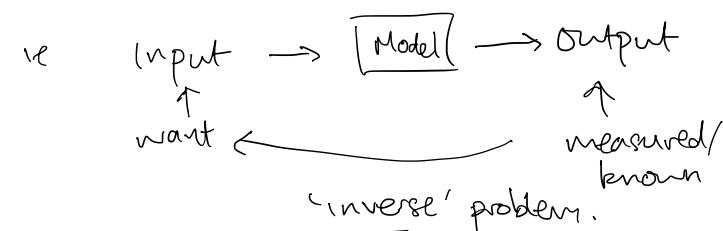
Forward vs Inverse Problems

Typical modelling / simulation setup:



& Typical ‘forward’ models } ODE
PDE
ABM (Agent-based model)
etc
(many things!)

Problem: usually measure (noisy) outputs & want inputs



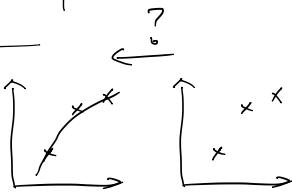
Examples

Many important problems can be considered 'inverse' problems

Obvious example:

parameter estimation, fitting
parameterisation

e.g. 'fit' an ODE, PDE, polynomial etc to given data.



But the general concept is very broad & essentially about (math) 'inverting' given functions, or (physics) determining 'causes' from 'effects'

e.g. $y = f(x)$, say $y = x^2$

$$x \mapsto x^2$$

observe $y=4$

what was x ?

Note: no unique soln!

↳ solution set: $\{2, -2\}$

(note: $f^{-1}\{4\}$, not $f^{-1}(4)$!)

Examples (see readings for details)

Some common 'physical' examples include

- Tomography ↘ CT (computerised)
 ↘ EIT (electrical impedance)

↳ 'imaging by sections'

→ given slices, reconstruct object

- Deconvolution (deblurring)

↳ given a blurred photograph, reconstruct unblurred version

- Geophysics → determine the origin of an earthquake, given seismographs measuring arrival times

- Geothermal engineering → determine the density or permeability etc of underground rocks given surface or well measurements of temperature

Examples (see Aster et al L1 reading for details)

More physical/mathematical examples:

- Determine the origin of a groundwater contaminant after it has been transported (by advection/diffusion) downstream to measurement locations
- Given a function, determine its derivative!
L HuH? See later →
- Given crime scene clues, determine the perpetrator...
→ many more!

What's the difference between 'forward' & 'inverse' though?

Keller (1976): two problems are inverse to each other if the formulation of one is naturally 'opposite' to the other

→ e.g. differentiation & integration
→ intuitive rather than fully-formal

However, often one problem is more naturally considered the direct or forward problem & the other more naturally the inverse problem

Example pairs :

Forward

multiplication to
find product

given present/past
predict future

integration



features

Inverse

division to find
factors

given present/future
predict past

differentiation

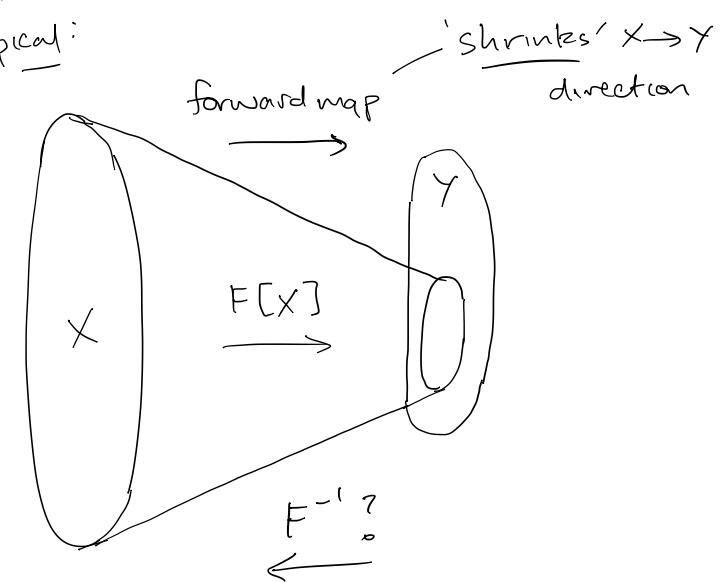


- 'easier'
- well-defined
- tend to 'smooth',
'combine' or
'reduce' inputs
to simpler
output
- respect causality/
flow of time
- stable

- 'harder'
- can be ill-defined
eg $\frac{1}{0}$
- often require
recovering 'lost'
info or 'uncombining'
to get inputs from
combined output
- amplify noise
- acausal/backwards
in time?
- unstable ?

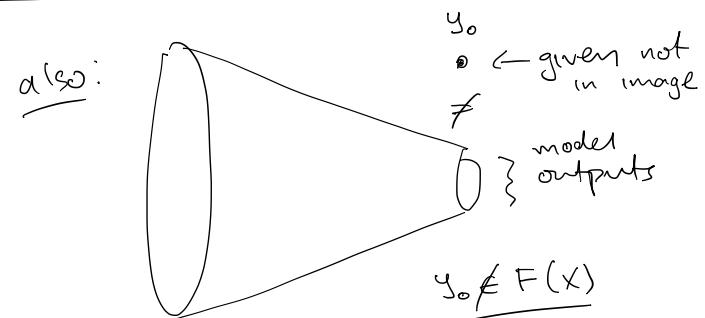
Mathematical Picture

Typical:



Think:

size { inputs / domain } > size { outputs / range / image space }



Well-posed vs ill-posed Problems

Hadamard (1902) call a problem well-posed if a solution:

- Exists (possible to satisfy)
- Is unique
- Is stable wrt small' changes in the givens

Problems that are not well-posed

are said to be

ill-posed

Key : forward problems \rightarrow well-posed
inverse problems \rightarrow ill-posed
(generally speaking)

Well-posed vs ill-posed Problems

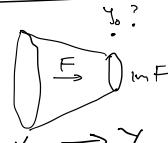
Math: given model $F: X \rightarrow Y$ & data $y_0 \in Y$

1. Existence: $y_0 \in \text{im } F$

$\text{im } F = \text{image of } F'$

$= F[X]$

$= \{F(x) \mid x \in X\}$



(F is onto)

note: image of a set is a set.

2. Uniqueness: F is 1-1

$$F(x_1) = F(x_2) \Rightarrow x_1 = x_2$$

i.e.

$$x_1 \neq x_2 \Rightarrow F(x_1) \neq F(x_2)$$

$x_1 \xrightarrow{\circ} \bullet$ No! want:
 $x_2 \xrightarrow{\circ} \bullet$ distinct \rightarrow distinct.

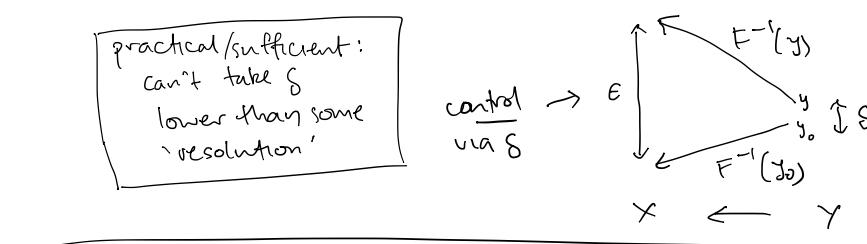
3. Stability: F^{-1} (which exists if 1&2 sat.)

is (sufficiently) continuous:

$$(\forall \epsilon)(\exists \delta(\epsilon)) \left[\|y - y_0\| < \delta \Rightarrow \|F^{-1}(y) - F^{-1}(y_0)\| < \epsilon \right]$$

[Given a tolerance ϵ , find a δ for that, s.t. if 'inputs' of F^{-1} differ by $< \delta$ then 'outputs' of F^{-1} differ less than ϵ .]

practical/sufficient:
can't take δ
lower than some
'resolution'

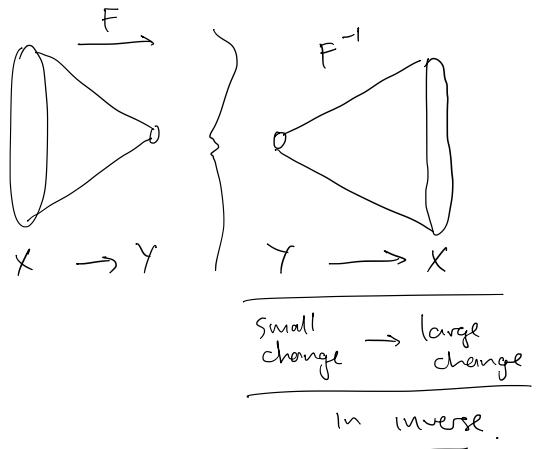


Well-posed vs ill-posed Problems

Intuition:

- Forward 'shrinks'
- Inverse needs to 'expand'

→ risks instability



Key Message:

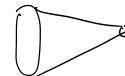
Even when inverse exists in principle
it may be unstable

Illustrations: instability even when inverse exists

Integration: 'sums up' → reduces

$$\begin{aligned} \text{eg } 1+2+3 &= 6 \\ 3+3 &= 6 \\ 2+a &= 6 \end{aligned}$$

&



Differentiation: 'breaks down' → expands?

↳ Numerical Example.

◦ Given a signal vector $\mathbf{x} \in \mathbb{R}^n$,

◦ We can sum the first k elements
by taking dot product with

$$w_k = (\underbrace{1, 1, 1, \dots, 1}_k, 0, 0, \dots, 0)^T = (I_k^T, 0, 0, \dots, 0)^T$$

$$I_k^T = (\underbrace{1, 1, \dots, 1}_k)$$

(Note we will often think of 'samples' / 'signals'
as finite dimensional vectors (x_1, x_2, \dots)
approximating infinite-dimensional
random vars $X(\omega), \omega \in \Omega$, k operations
like summation etc as finite versions
of integration).

Illustrations

$$\text{Eq } x = (1 \ 2 \ 3 \ 0 \ 0)^T$$

$$w_1 = (1 \ 0 \ 0 \ 0 \ 0)^T$$

$$w_2 = (1 \ 1 \ 0 \ 0 \ 0)^T$$

:

$$\text{cumsum}(k, x) = w_k^T x$$

$$\text{eg } CS(2, x) = (1 \ 1 \ 0 \ 0 \ 0) \begin{pmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \end{pmatrix}$$

$$= 1 + 2 = 3$$

Return vector of all cumulative sums

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \leftarrow \begin{array}{l} \text{forward} \\ \text{map} \\ X \rightarrow Y \end{array}$$

$$y = Ax$$

Note:
(assume all vectors
are column by
default)

Illustrations

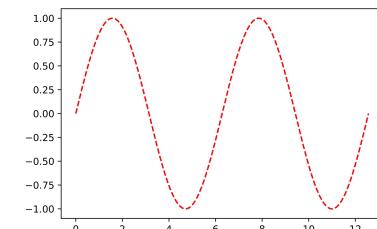
Python (see Canvas)

```
t = np.linspace(0, 4*np.pi, 1000)
x = np.sin(t)
plt.plot(t, x, 'r--')
plt.show()
```



(import numpy as np)

Input signal. $x(t)$

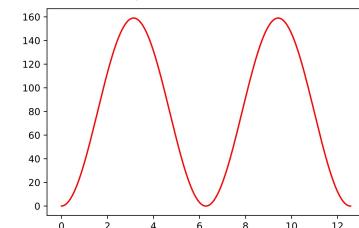


numpy's 'cumsum'

```
#built-in 'integration'
plt.plot(t, np.cumsum(x), 'r')
plt.show()
```



'Integral', $y = \int_0^t x(\xi) d\xi$



Roll our own as matrix mapping

#forward mapping for integration (cumulative sum)

```
def create_fmap_int(d=1):
    A = np.zeros([d, d])
    for i in range(0, d):
        A[i, :i+1] = 1
    return A
```

```
[[ 1.  0.  0. ... , 0.  0.  0.]
 [ 1.  1.  0. ... , 0.  0.  0.]
 [ 1.  1.  1. ... , 0.  0.  0.]
 ... ,
 [ 1.  1.  1. ... , 1.  0.  0.]
 [ 1.  1.  1. ... , 1.  1.  0.]
 [ 1.  1.  1. ... , 1.  1.  1.]]
```

#create forward mapping

```
A = create_fmap_int(len(x))
print(A)
```

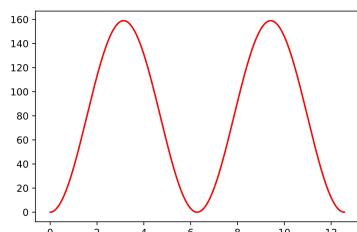


check A correct

```
#calculate output of forward mapping  
y = np.dot(A,x)  
  
#compare our forward mapping to built-in  
plt.plot(t,np.cumsum(x), 'r--')  
plt.plot(t,y, 'r')  
plt.show()
```



indistinguishable output

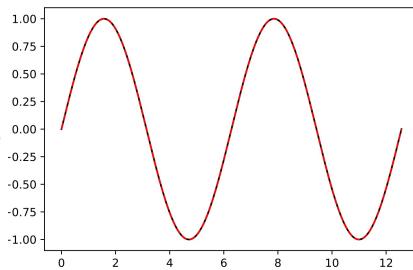


Invert using A^{-1} & compare to original output

```
#invert noise-free case  
plt.plot(t,np.dot(np.linalg.inv(A),y), 'k')  
plt.plot(t,x, 'r--')  
plt.show()
```



indistinguishable input



Note: A^{-1} exists (1-1 & onto)

$\rightarrow A$ is square

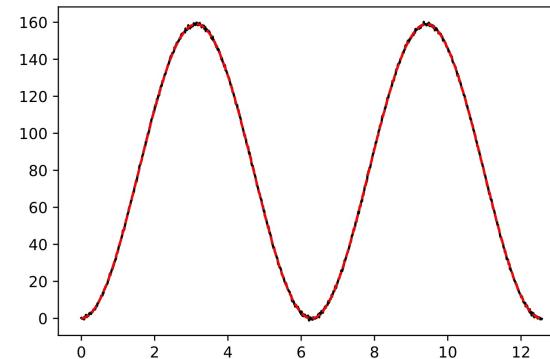
& all rows/cols are linearly independent

stability?

Add small amount of noise to output

$$\text{ie } Y_{\text{observed}} = Ax + \epsilon = y + \epsilon$$

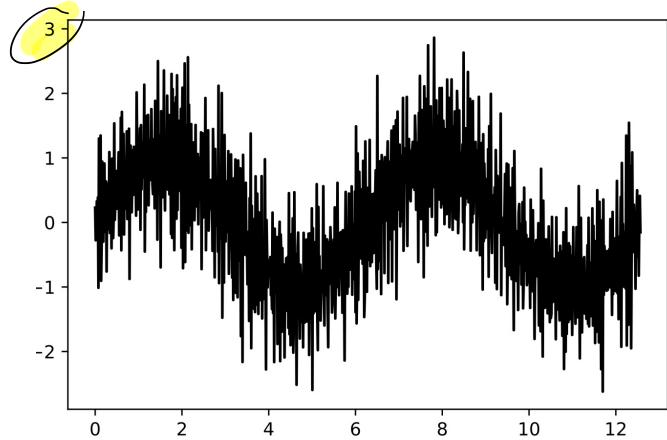
```
#add almost undetectable noise  
y_noisy = y+np.random.normal(0,0.5,size=len(y))  
plt.plot(t,y_noisy, 'k')  
plt.plot(t,y, 'r--')  
plt.show()
```



Question: Is A^{-1} still 'good'?

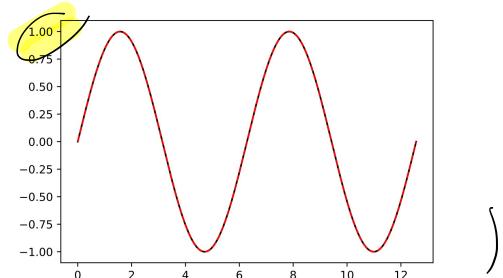
Instability of differentiation

```
#invert noisy  
plt.plot(t,x,'r--')  
plt.plot(t,np.dot(np.linalg.inv(A),y_noisy),'k')  
plt.show()
```



Very 'irregular'! Large deviations |

(Recall noise-free:



Can we just make grid finer?

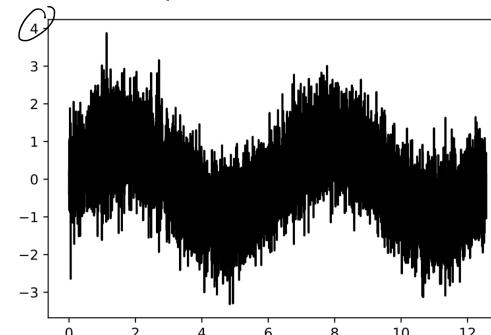
→ Makes worse!

Finer grid:

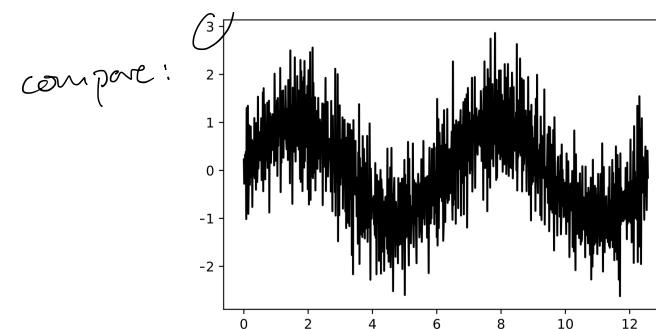
```
t = np.linspace(0, 4*np.pi, 10000)  
x = np.sin(t)  
plt.plot(t,x,'r--')  
plt.show()
```

$\times 10$ finer } like
finer discretis.

Same process gives:



noise ↑
even more!



Instability

- Integration 'smooths'
- Differentiation 'coarsens'

Mathematically, differentiation
is an unbounded (discontinuous)
operator [see functional analysis]

Numerically, differentiation is

ill-conditioned

↳ not quite unbounded
since discrete / finite,
but practically
unbounded.

discrete ↓ continuous: ill-posed
discrete : ill-conditioned

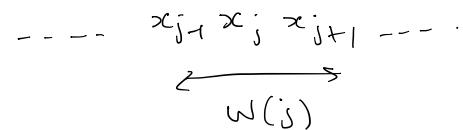
see [Engel et al L1 reading]

Illustration : Local averaging (smoothing)

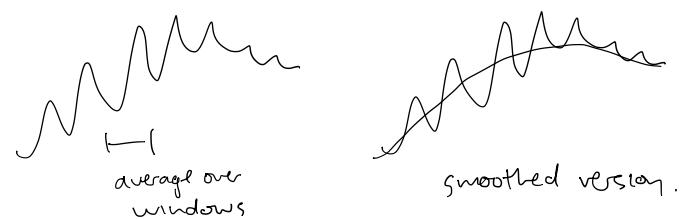
Given ['signal'] or ['input'] [$x \in \mathbb{R}^n$]

Local average over neighbourhood $w(i)$
of i is:

$$\frac{1}{|w|} \sum_{i \in w(i)} x(i)$$



Smoothing: moving average



Inverse Problem:

- given a smoothed (or 'blurred') image, determine original
→ eg photography

same idea as integration/differentiation

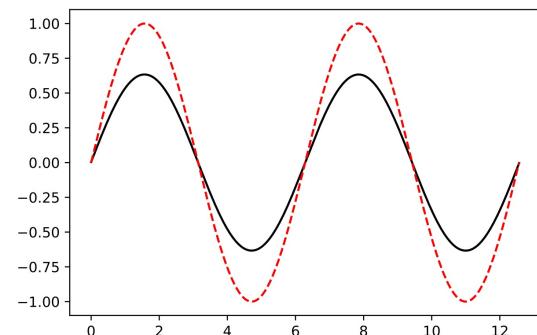
- smoothing forward map
- inverse amplifies any noise in output when trying to recover

$$y_0 = \tilde{y} + \epsilon = \tilde{A}x + \epsilon$$

smoothing noise

A^{-1} unstable!

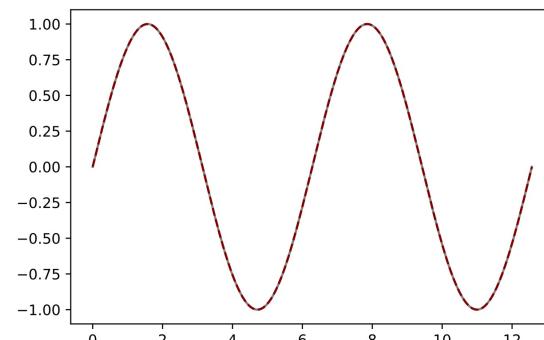
Smoothing map:



red: original
black: local av.

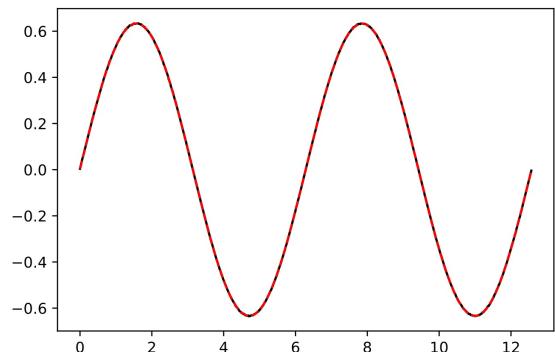
can implement as linear forward map, i.e. matrix A (exercise?)

Invert noise free gives:



Instability of 'deblurring':

Add small noise:



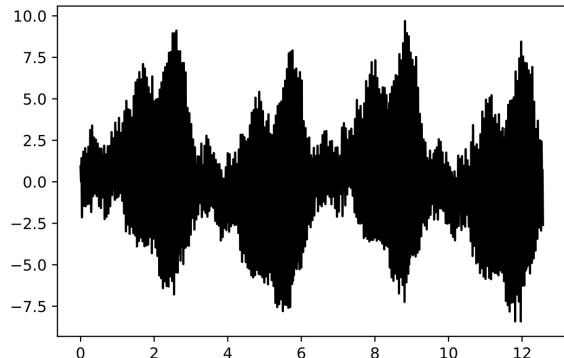
practically
indetectable!

Should we give up?

No! (Well...)

We can stabilise the
inversion procedure

Invert noisy using A^{-1}



Terrible!

"Regularisation"

→ tomorrow.

1. Introduction: Examples of Inverse Problems

In this introductory section, we give, after a general discussion of the term “inverse problems”, examples of various classes of inverse problems arising in various application fields. As we will see, *linear inverse problems* frequently lead to *integral equations of the first kind*, which is why such equations play an important role in the study of inverse problems, as we will see throughout a large part of this book. On the other hand, many basic *inverse problems* are inherently nonlinear even if the corresponding *direct problem* is linear. This fact is our motivation for devoting a significant part of this book to the mathematics of nonlinear inverse problems.

When using the term *inverse problem*, one immediately is tempted to ask “inverse to what?”. Following J.B. Keller [150], one calls *two* problems *inverse to each other* if the formulation of one problem involves the other one. For mostly historic reasons, one might call one of these problems (usually the simpler one or the one which was studied earlier) the *direct problem*, the other one the *inverse problem*. However, if there is a *real-world* problem behind the mathematical problem studied, there is, in most cases, a quite natural distinction between the direct and the inverse problem. E.g., if one wants to predict the future behaviour of a physical system from knowledge of its present state and the physical laws (including concrete values of all relevant physical parameters), one will call this the *direct problem*. Possible *inverse problems* are the determination of the present state of the system from future observations (i.e., the calculation of the evolution of the system backwards in time) or the identification of physical parameters from observations of the evolution of the system (*parameter identification*).

There are, from the applications point of view, two different motivations for studying such inverse problems: first, one wants to *know* past states or parameters of a physical system. Second, one wants to find out how to influence a system via its present state or via parameters in order to *steer* it to a desired state in the future.

Thus, one might say the *inverse problems are concerned with determining causes for a desired or an observed effect*.

As we will see, such inverse problems most often do not fulfill Hadamard’s postulates of *well-posedness* (see Chapter 2). They might not have a solution in the strict sense, solutions might not be unique and/or might not depend continuously on the data. Mathematical problems having these undesirable properties are called *ill-posed problems (improperly posed problems)* and pose (mostly because of the discontinuous dependence of solutions on the data) severe numerical difficulties. While the study of concrete inverse problems frequently involves the question how to enforce uniqueness by additional information or assumptions, not much can be said about this in a general context. The aspect of lack of stability and its restoration by appropriate methods (*regularization methods*), however, can be treated in sufficient generality. The theory of regularization methods is well-developed for linear inverse problems and at least emerging for nonlinear problems and forms the core of this

4 1. Introduction: Examples of Inverse Problems

book.

There is a vast literature on inverse and ill-posed problems. In addition to the books already quoted in the Preface, we mention, as general references,

- the following monographs: [13, 32, 45, 95, 136, 225, 230, 282, 288],
- the following conference proceedings: [14, 36, 43, 72, 78], [81] (which emphasizes inverse problems arising in industry), [83, 138, 223, 241, 242, 265, 289],
- the journals *Inverse Problems* (Institute of Physics Publ.), *Inverse Problems in Engineering* (Gordon & Breach), and *Journal of Inverse and Ill-Posed Problems* (VSP).

Many more references will be given in the appropriate sections.

1.1. Differentiation as an Inverse Problem

Two mathematical problems inverse to each other are differentiation and integration. A-priori, it is not clear which of these problems should be the direct problem and which one the inverse problem. However, as we will now see, differentiation has (as opposed to integration) the properties of an ill-posed problem. Moreover, since, as we will see later, many inverse problems involve at some step differentiation of the data, differentiation might be viewed as the “inverse problem”, although in most calculus courses, it is treated first.

Let $f \in C^1[0, 1]$ be any function, $\delta \in (0, 1)$, $n \in \mathbb{N}$ ($n \geq 2$) be arbitrary, and define

$$f_n^\delta(x) := f(x) + \delta \sin \frac{nx}{\delta}, \quad x \in [0, 1]. \quad (1.1)$$

Then

$$(f_n^\delta)'(x) = f'(x) + n \cos \frac{nx}{\delta}, \quad x \in [0, 1]. \quad (1.2)$$

Now, in the uniform norm,

$$\|f - f_n^\delta\|_\infty = \delta,$$

but

$$\|f' - (f_n^\delta)'\|_\infty = n.$$

Hence, if we consider f and f_n^δ as the exact and perturbed data, respectively, then for an arbitrarily small data error δ , the error in the result, namely the derivative, can be arbitrarily large, namely n . Hence, the derivative does not depend continuously on the data with respect to the uniform norm. Of course, we could enforce continuous dependence by measuring the data error in the C^1 -norm. However, this would be a sort of cheating, since then, we would call a data error small if the error in the function values and in the values of the derivative, which is exactly what we want to compute, would be small.

For later reference, note that f' solves the simple integral equation of the first kind

$$(Kx)(s) := \int_0^s x(t) dt = f(s) - f(0), \quad (1.3)$$

which is solvable in $C[0, 1]$ only if $f \in C^1[0, 1]$. The corresponding direct problem would be to compute f from x , i.e., integration, which is a stable process on $C[0, 1]$. Note that integration is a smoothing process, i.e., highly oscillatory errors in x (e.g., of the form $n \cos(nx/\delta)$ as they appeared in (1.2)) are damped out (to $\delta \sin(nx/\delta)$) and have a very small effect on the data for the inverse problem. This smoothing is responsible for the fact that errors of small amplitude, but high frequency, create large oscillations in the solution of the inverse problem. These considerations are not restricted to this concrete problem: whenever a direct problem has smoothing properties one has to expect the appearance of oscillations coming from small data perturbations (of high frequency) in the solution of the inverse problem. This effect is the more pronounced the stronger smoothing the direct problem is.

Why (or, under what circumstances) can we differentiate a function in spite of these problems? We have to be able to exclude the presence of data errors of arbitrarily high frequency, e.g., of the form as in (1.1); this can be done, e.g., if we know a bound for f'' . In the example above, such a bound would give a bound for n in terms of δ , thus coupling the amplitude and the frequency of the possible data errors. A functional analytic argument is the following:

If we consider the operator K as defined in (1.3) on $C[0, 1]$, then it is a continuous linear injective operator, whose inverse (defined on $C^1[0, 1]$, considered as a subspace of $C[0, 1]$) is unbounded. However, if we restrict K to the set $\{x \in C[0, 1] \mid \|x\|_\infty + \|x'\|_\infty \leq \gamma\}$, which is compact in $C[0, 1]$ due to the Arzelà-Ascoli Theorem, then the inverse of this restricted operator is continuous on its range, as the inverse of a continuous bijective (not necessarily linear) map defined on a compact set is again continuous. Thus, we can “restore stability” by assuming an a-priori bound for f' and f'' .

The stability problems addressed must appear somehow when computing the derivative via difference quotients: let f be the function we want to differentiate, f^δ its noisy version with

$$\|f - f^\delta\|_\infty \leq \delta.$$

We want to use the central difference quotient with step size h . If $f \in C^2[0, 1]$, Taylor expansion yields

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + O(h),$$

while for $f \in C^3[0, 1]$,

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + O(h^2).$$

Thus, the accuracy of the central difference quotient depends on the smoothness of

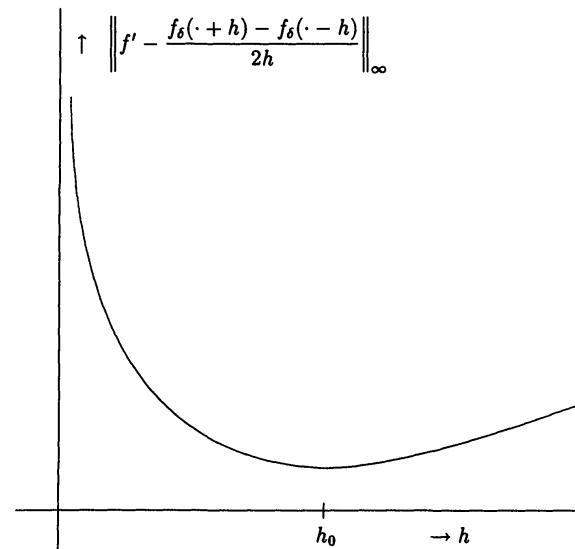


Figure 1.1: Total error depending on h

the exact data. Instead of f' , we are actually computing

$$\frac{f^\delta(x+h) - f^\delta(x-h)}{2h} \sim \frac{f(x+h) - f(x-h)}{2h} + \frac{\delta}{h}.$$

Thus, the total error behaves like

$$O(h^\nu) + \frac{\delta}{h}, \quad (1.4)$$

where $\nu = 1$ or 2 if $f \in C^2[0, 1]$ or $f \in C^3[0, 1]$, respectively. For a fixed error level δ , it looks as in Figure 1.1.

If h becomes too small, the total error increases due to the error term δ/h , the propagated data error. Of course, if h is too large, then the approximation error becomes too large. There is an *optimal* discretization parameter h_0 , which can, however, not be computed explicitly, since it depends on unavailable information about the exact data, e.g., their smoothness. However, one can at least estimate the asymptotic behaviour of h_0 if h is chosen as a power of δ , i.e.,

$$h \sim \delta^\mu,$$

then one can minimize (1.4) by taking $\mu = 1/2$ or $\mu = 1/3$, which results in a behaviour of the total error as $O(\sqrt{\delta})$ or $O(\delta^{2/3})$ for $f \in C^2[0, 1]$ or $f \in C^3[0, 1]$,