ENGSCI 741

INVERSE PROBLEMS

Oliver Maclaren
oliver.maclaren@auckland.ac.nz

MODULE OVERVIEW

Inverse Problems (*Oliver Maclaren*) [~9 lectures/tutorials]

1. *Basic concepts* [3 lectures]
   Forward vs inverse problems. Well-posed vs ill-posed problems. Algebra of inverse problems (generalised inverses etc). Regularisation and trade-offs.

2. *More regularisation* [3-4 lectures/tutorials]
   Higher-order Tikhonov regularisation, truncated singular value decompositions, iterative regularisation. Statistical view of inverse problems

MODULE OVERVIEW

3. *Applications* [2-3 lectures/tutorials]
   Deblurring, numerical differentiation, parameter estimation, tomography, remote sensing...(we'll see!)

LECTURE 1: OVERVIEW

Topics:

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

EngSci 741 : Inverse Problems
(A brief intro!)

Lecture 1: Overview.

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

Forward vs Inverse Problems

Typical modelling/simulation setup:

parameters →
initial conditions →
boundary conditions →
etc
$\underbrace{\hphantom{boundary conditions etc}}_{\text{inputs}}$

$\xrightarrow{} \boxed{\text{Model}} \xrightarrow{}$ solution $y(x,t)$ etc.

'Forward' → output

& Typical 'forward' models:
- ODE
- PDE
- ABM (Agent-based model)
- etc
- (Many things!)

Problem: usually measure (noisy) |outputs| & want |inputs|

---

ie input → $\boxed{\text{Model}}$ → output

↑ want ←—— ↑ measured/known

'inverse' problem.

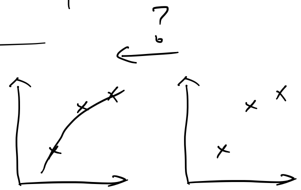
Examples

Many important problems can be considered 'inverse' problems

Obvious example:

$\boxed{\text{parameter estimation, fitting, parameterisation}}$

eg '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'

eg $y = f(x)$, say $y = x^2$

$$x \longmapsto x^2$$

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)

  } etc

  ↳ '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!
  - └ 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

→ eg 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	Inverse
multiplication to find product	division to find factors
given present/past predict future	given present/future predict past
integration	differentiation

↓ **features** ↓

Forward

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

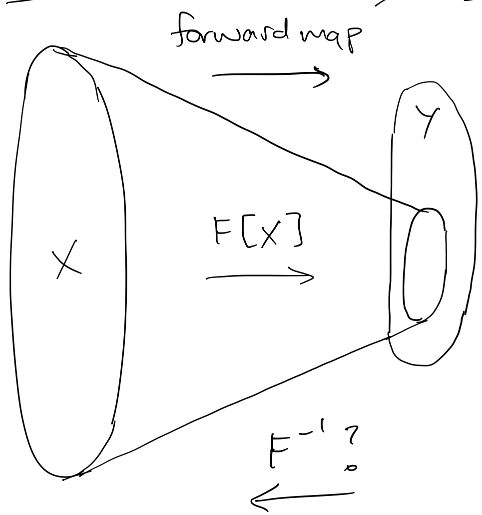
Inverse

- '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:

'shrinks' $X \to Y$ direction

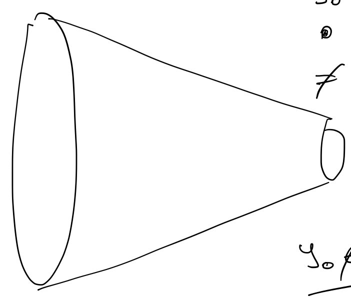


Think:

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

---

also:



y_0 ← given not in image

≠

} model outputs

$y_0 \notin F(X)$

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

$\boxed{\text{ill-posed}}$

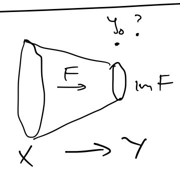
Key: forward problems → well-posed
inverse problems → ill-posed
(generally speaking)

Well-posed vs ill-posed Problems

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

---

1. $\boxed{\text{Existence}}$: $y_0 \in \text{Im}\,F$ (F is onto)



$\text{Im}\,F$ = 'image of F'
$= F[X]$
$= \{F(x) \mid x \in X\}$

note: image of a set is a set.

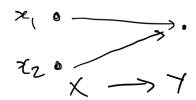
---

2. $\boxed{\text{Uniqueness}}$: F is 1–1

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

ie

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



No! Want: distinct → distinct.

---

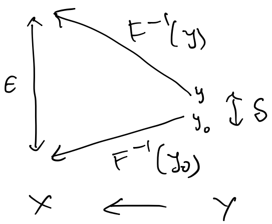
3. $\boxed{\text{Stability}}$: F^{-1} (which exists if 1&2 sat.) is (sufficiently) continuous:

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

[Given a tolerance ϵ, find a δ tolerance 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'

control ϵ via δ

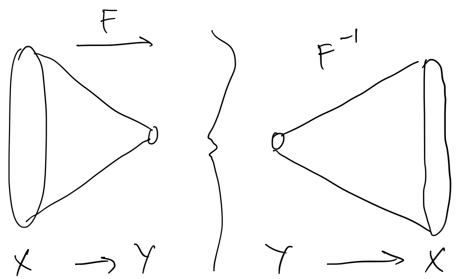


Well-posed vs ill-posed Problems

Intuition:

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

→ risks instability



small change → large change

in inverse.

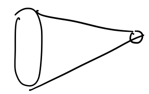
> **Key Message:**
>
> Even when inverse exists in principle it may be unstable

Illustrations: instability even when inverse exists

Integration: 'sums up' → reduces

eg $1+2+3 = 6$
$3+3 = 6$
$2+4 = 6$

&



differentiation: 'breaks down' → expands?

↳ **Numerical Example.**

- Given a signal 'vector' $x \in \mathbb{R}^n$,
- We can sum the first k elements by taking dot product with

$$w_k = (\underbrace{1\ 1\ 1 \dots 1}_{k}\ 0\ 0 \dots 0)^T$$

→

Illustrations

Note: (assume all vectors are column by default)

Eg $x = (1\ \ 2\ \ 3\ \ 0\ \ 0)^T$

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

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

$\vdots$

cumsum $(k, x) = w_k^T x$

(sigh...)

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

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

$y = Ax$

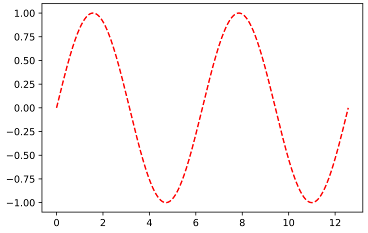
Illustrations

(import numpy as np)

Python (see Canvas)

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

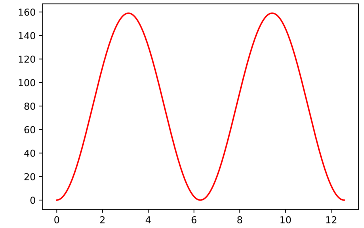
Input signal. $x(t)$



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

#create forward mapping
A = create_fmap_int(len(x))
print(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.]]
```

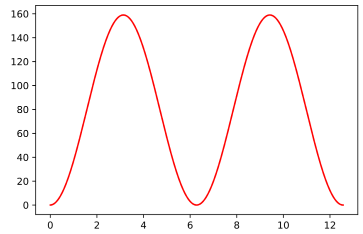
Check A correct

```python
#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()
```

$\longrightarrow$

indistinguishable output

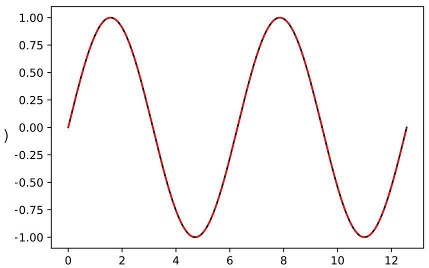


Invert using A^{-1} & compare to original output

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

$\longrightarrow$

indistinguishable input



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

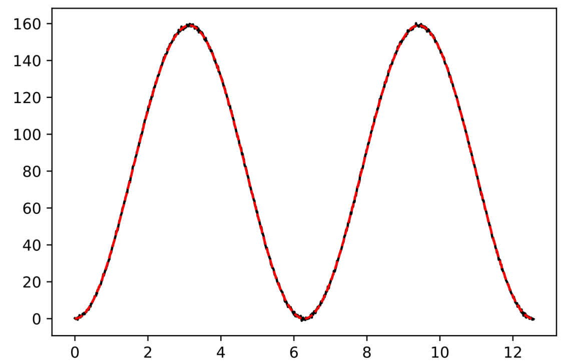
$\rightarrow$ A is square
& all rows/cols are linearly independent

stability?

Add small amount of noise to output

ie $y_{observed} = Ax + \epsilon = y + \epsilon$

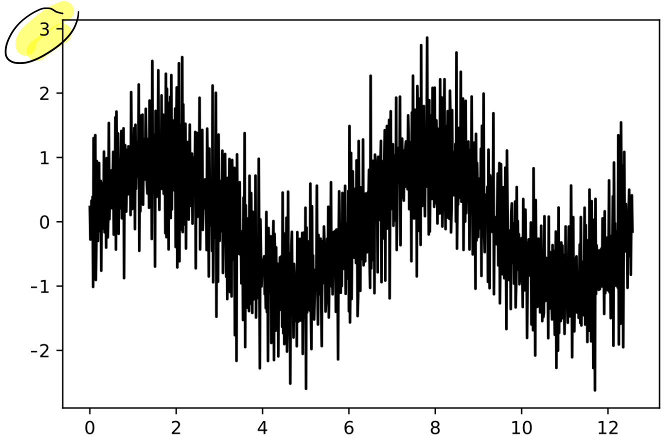
```python
#add almost undetectable noise
y_noisy = y+np.random.normal(0,0.5,size=len(y))
plt.plot(t,y_noisy,'k')
plt.plot(t,y,'r--')
plt.show()
```



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

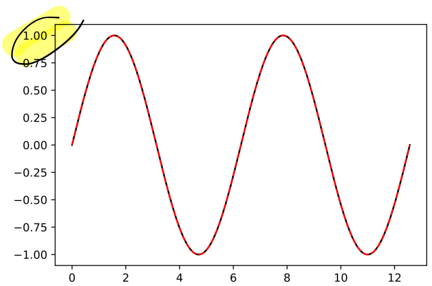
Instability of differentiation

```python
#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:



)

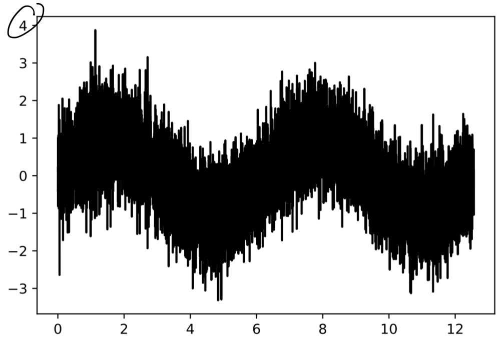
Can we just make grid finer?

→ Makes worse!

Finer grid: ×10 finer } like finer discretis.

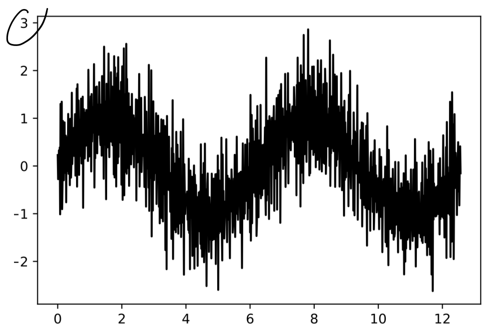
```python
t = np.linspace(0,4*np.pi,10000)
x = np.sin(t)
plt.plot(t,x,'r--')
plt.show()
```

Same process gives:



noise ↑ even more!

compare:



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.

---

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

---

see [Engel et al L1 reading]

Illustration : Local averaging (smoothing)

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

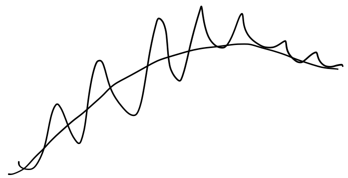
Local average over neighbourhood $W(i)$ of i is

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

$$- - - - \quad \underbrace{x_{j-1} \;\; x_j \;\; x_{j+1}}_{W(j)} \quad - - - .$$

Smoothing : moving average

average over windows



smoothed version.

Inverse Problem:

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

same idea as integration / differentiation

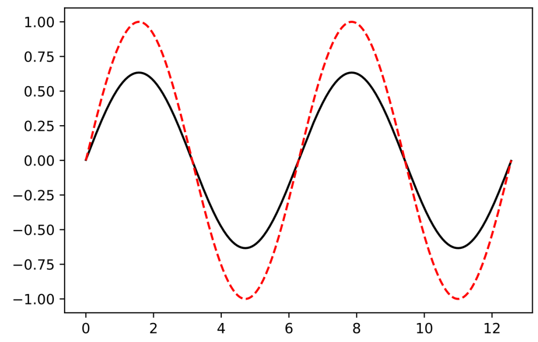
→ smoothing forward map

→ inverse amplifies any noise in output when trying to recover

$$y_o = y + \epsilon = \underbrace{Ax}_{\text{smoothing}} + \underbrace{\epsilon}_{\text{noise}}$$

A^{-1} unstable!

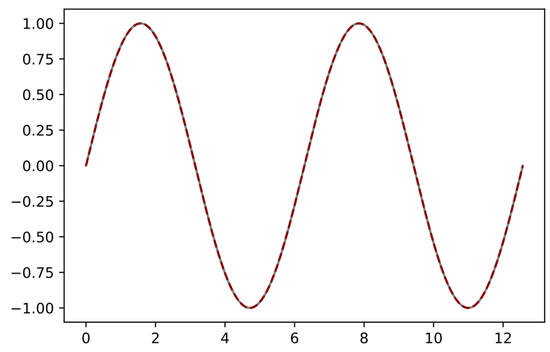
Smoothing map:



red: original
black: local av.

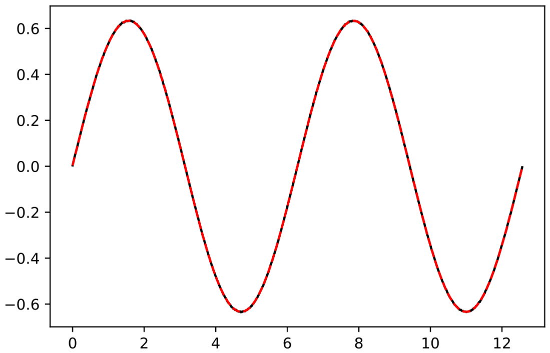
Can implement as linear forward map, ie matrix A (assignment?)

Invert noise free gives:



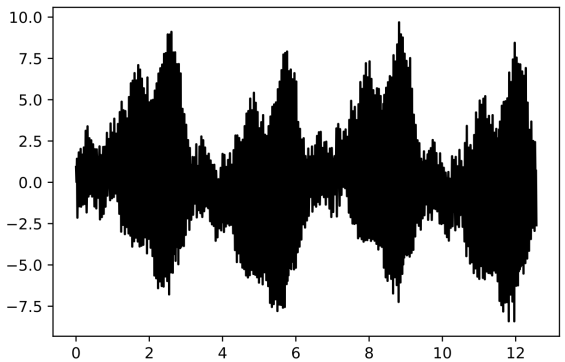
Instability of 'deblurring':

Add small noise:



practically indectable!

Invert noisy using A^{-1}



Terrible!

Should we give up?

No! (Well....)

We can stabilise the inversion procedure

"Regularisation"

→ tomorrow.

Engl, Hanke, Neubauer et al 'Regularization of Inverse Problems'

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

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

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

Then

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

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.

Engl, Hanke, Neubauer et al 'Regularization of Inverse Problems'

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

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

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 $\{x \in C^1[0,1] \mid \|x\|_\infty + \|x'\|_\infty \le \gamma\}$, which is compact in $C[0,1]$ due to the Arzela-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

$$\|f - f^\delta\|_\infty \le \delta\,.$$

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

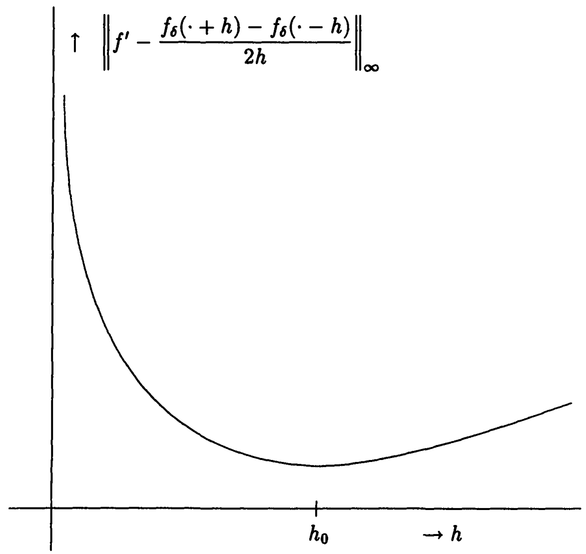


Figure 1.1: Total error depending on h

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

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

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 $O(\delta^{\frac{2}{3}})$ for $f \in C^2[0,1]$ or $f \in C^3[0,1]$, Engl, Hanke, Neubauer et al 'Regularization of Inverse Problems'

respectively. Thus, even in the best possible case ($\nu > 2$ is obviously not possible in (1.4)) and for an optimal choice of h, we obtain only a convergence rate of $O(\delta^{\frac{2}{3}})$, where δ denotes the data error, i.e., there is an intrinsic loss of information. This rate cannot be improved unless f is a quadratic polynomial [106]. If μ is not chosen optimally, i.e., if the discretization parameter h and the noise level are not linked appropriately, this loss of information becomes more severe.

Early references to the treatment of numerical differentiation by regularization are [11, 12].

In this example, we saw some effects that are typical for ill-posed problems:

- amplification of *high frequency* errors
- restoration of stability by using *a-priori information*
- two error terms of different nature, one for the approximation error, the other one for the propagation of the data error, adding up to a total error as in Figure 1.1
- the appearance of an optimal discretization parameter, whose choice depends on a-priori information
- loss of information even under optimal circumstances.

1.2. Radon Inversion (X-Ray Tomography)

An inverse problem that has been widely studied because of its importance in, e.g., medical applications arises in *Computerized Tomography* (CT). We consider the two-dimensional situation:

Let $D \subseteq \mathbb{R}^2$ be a compact domain with a spatially varying density f. In medical applications, D symbolizes a cross-section of the human body; in nondestructive testing, D is a cross-section of the material to be tested. The aim is to recover the density f from X-ray measurements in the plane where D lies. These X-rays travel along lines, which are parameterized by their normal vector $w \in \mathbb{R}^2$ ($\|w\| = 1$) and their distance $s > 0$ from the origin (see Figure 1.2).

If one assumes that the decay $-\Delta I$ of an X-ray beam along a distance Δt is proportional to the intensity I, the density f, and to Δt, one obtains

$$\Delta I(sw + tw^{\perp}) = -I(sw + tw^{\perp})f(sw + tw^{\perp})\Delta t, \tag{1.5}$$

where $w^{\perp}$ is a unit vector orthogonal to w. By letting Δt tend to 0 in (1.5), one obtains

$$\frac{d}{dt}I(sw + tw^{\perp}) = -I(sw + tw^{\perp})f(sw + tw^{\perp}). \tag{1.6}$$

We denote by $I_L(s, w)$ and $I_0(s, w)$ the intensity of the X-ray beam measured at the detector and at the emitter, respectively, where detector and emitter are connected