Inverse Problems: Can We Approximate an Unsolvable Equation?

Oscar Armando Hidalgo Arevalo, School of Mathematics, University of Leeds
Supervised by Professor Daniel Lesnic (Department of Applied Mathematics)

What is an Inverse Problem?

J.B. Keller said: “We call two problems inverses of one another if the formulation of each involves all or part of the solution of the other”. In other words, an Inverse Problem is considered the process of calculating from a set of observations the factors that produce them. For instance, in mechanics one direct problem is to calculate the trajectory of an object from knowing the initial forces. The inverse problem would then be to calculate the initial forces from knowing the object’s trajectory. This is a situation often found in real life.

Well-Posed and Ill-Posed Problems

A problem is consider well-posed when it satisfies three conditions:

- Existence: The problem has a solution,
- Uniqueness: There is not more than one solution, and
- Stability: The solution depends continuously on the data.

On the other hand, a problem that is not well-posed, it is ill-posed resulting in several solutions or solution which lose reliability with the data as it does not depend on it.

For many years, it was believed that only well-posed problems were motivated by physical reality due to their characteristic properties. This led to a period in which ill-posed problems were not studied and even forgotten.

However, the discovery of the ill-posedness of inverse problems modified that belief and led to the following statement: A direct problem is well-posed while the corresponding inverse problem is ill-posed. This opened a whole field of study as now ill-posed problems could be found in real life. Nevertheless, the next question is how to model these problems.

Fredholm Equations

Fredholm integral equations of the first kind model ill-posed problems. These equations have the following general form:

$$g(x) = \int_a^b k(x,t) f(t)\,dt, \qquad c \le x \le d$$

Where *f(t)* is the unknown function.

One example of these equations is the Abel’s problem to find the curve of descent as a function of the vertical distance of fall

$$g(z) = \int_0^z \frac{f(y)}{\sqrt{2a(z-y)}}\,dy.$$

Where a is the gravitational acceleration and *f(y)* is the wanted function (the solution).

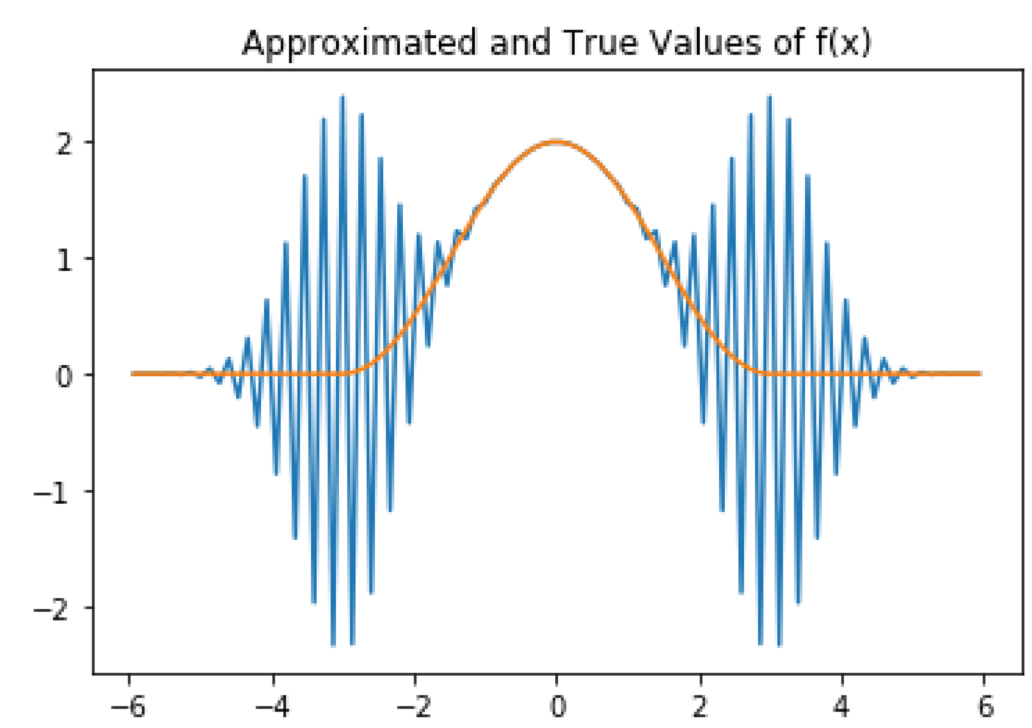
Computer Power and Noise

As expected, much computer power is necessary to find numerical solutions of inverse problems. It was not until after WWII that technology allowed finding these solutions to be practical. Now, solutions are obtained with less-than-a-page piece of code like the one presented here.

```python
import math
import matplotlib.pyplot as plt
import numpy as np
def k(z):
    if -3<=z<=3:
        return 1+math.cos((1/3)*math.pi*z)
    else:
        return 0.0
def sgn(x):
    if x>=0:
        return 1
    else:
        return -1
def g(x):
    if -6<=x<=6:
        g=(6-math.fabs(x))*(1+0.5*math.cos((1/3)*math.pi*x))
        g=g+(9/(2*math.pi))*sgn(x)*math.sin(math.pi*x/3)
        return g
    else:
        return str("value of x out of range")
def f(y):
    return k(y)
def aproxf(n):
    b=6
    a=-6
    N=n
    yi=[]
    for i in range(N+1):
        yi.append(a+i*(b-a)/N)
    midyi=[]
    for j in range(N):
        midyi.append(0.5*(yi[j]+yi[j+1]))
    xi=[]
    for i in range(n):
        xi.append(a+i*(b-a)/(n-1))
    K=[]
    gi=[]
    for i in range(n):
        for j in range(n):
            K.append(k(xi[i]-midyi[j])*(b-a)/n)
        gi.append(g(xi[i]))
    Ki=[]
    for  i in range(n):
        Ki.append(K[i*n:(i+1)*n])
    A=np.array(Ki)
    B=np.array(gi)
    C=np.linalg.solve(A,B)
    plt.plot(midyi,C)
    awe=[]
    for i in midyi:
        awe.append(k(i))
        plt.title("Approximated and True Values of f(x)")
    plt.plot(midyi,awe)
    return midyi,list(C),awe
aproxf(90)
```

Nevertheless, computational tools brought not just solutions but also awareness of the instability of these solutions due to ill-posedness. This is a result of using discrete data to solve continuous equations.

In many cases it occurs that the solution is very sensitive to the data causing that small oscillations, or error, in the data produce large oscillations in the solutions, or noise, as it appears in the graph below. Real data is not exact and it always comes with error thus the solutions will present it as well which makes the solutions unreliable or even absurd. Currently, there are many methods to find solutions to ill-posed problems but not many to cure ill-posedness.



Conclusion: Inverse Problems are important

Its main utility falls in the fact that when encountering a phenomenon that cannot be analysed directly, it is possible to construct mathematical objects, usually equations or functions, from observational data to emulate such data. These objects are easier to understand and manage and through analysing them the phenomenon can be understood.

Inverse Problems analysis helps to solve and deal with real life problems that otherwise would not be approachable because of ill-posed behaviour.

Current and future applications are limitless mainly because every problem has its corresponding inverse problem and when dealing with a phenomenon most times only data of the effect is available and not the causes. On those situations, Inverse Problems is the perfect tool to obtain answers.

References

C W Groetsch 2007 J. Phys.: Conf. Ser. 73 012001

Baker, C. T. H. Fox, L. Mayers, D. F. Wright, K. Numerical solution of Fredholm integral equations of first kind, *The Computer Journal*, Volume 7, Issue 2, 1 January 1964, Pages 141–148, https://doi.org/10.1093/comjnl/7.2.141

Bertero, M. and Boccacci, P. 1998. Introduction to Inverse Problems in Imaging. 1st Edition. Boca Raton: CRC Press

Phillips, D.L. A Technique for the Numerical Solution of Certain Integral Equations of the First Kind, Journal of the ACM (JACM), Volume 9 Issue 1, Jan. 1962, Pages 84-97, https://dl.acm.org/citation.cfm?doid=321105.321114

Supported by donations to the
FOOTSTEPS FUND
by alumni and friends of the University of Leeds

UNIVERSITY OF LEEDS