Using inverse problems as projects in an ordinary differential equations class

Jeff Graham
Western Carolina University

1 Introduction

Inverse problems are important and widespread in industry, but they are conspicuously absent in the mathematics curriculum. To make our courses more relevant, we should make an effort to include coverage of inverse problems. This paper will present an example of an inverse problem which is suitable for a differential equations class.

2 What is an inverse problem?

A very loose definition of an inverse problem is that an inverse problem is a problem which is posed opposite of the direct problem. An example which should be familiar to most people is the Jeopardy game show. In Jeopardy, you are given the answer and must supply the question. This is the opposite of the way most quiz shows operate. In a more mathematical vein, a classic inverse problem is the brachistocrone problem (John Bernoulli 1696) :

> Of all curves joining two points in a vertical plane, find the one along which a particle sliding from rest without friction takes the least time to descend.

This inverse problem led to a new branch of mathematics (the calculus of variations) to achieve its solution. In this short paper, it will not be possible to even scratch the surface of the variety of inverse problems that exist. The interested reader should consult the excellent annotated bibliography in [2].

2.1 Mathematical Features

The distinguishing mathematical aspect of inverse problems is that they are usually ill-posed. Ill-posed means that an inverse problem will generally violate one or more of the properties of a well-posed problem as defined by Hadamard [4]. He stated that for a problem to be well-posed it must have a solution, that solution must be unique and stable with respect to perturbations in the data. Inverse problems often have no unique answer and are usually unstable.

At this point you may be asking yourself, if there is no unique answer to the problem or the answer is unstable, how does one achieve any useful information? To resolve the ill-posedness one usually has to supply some prior knowledge of the desired solution. For example, when one does a least squares fit to data one computes a solution to the normal equations that is as close to a solution to the original problem as possible. In other words, one has to specify what qualifies as a solution in advance of the computation.

2.2 Are inverse problems important?

Inverse problems are very important in industry. For example, medical imaging techniques such as NMR, MRI, CATSCAN, and impedence imaging are all based on solving some type of inverse problem. Other important areas of application include geophysical prospecting and nondestructive testing. This list of applications is by no means exhaustive.

Inverse problems may also be used as an important teaching tool in the classroom. Solving these problems involves more creativity than solving the standard direct problems, making them more intellectually challenging. The solution to an inverse problem cannot generally be accomplished without employing a computer, which allows us to make use of appropriate technology in a meaningful way. Solving an inverse problem may also enable students to better understand the direct problem on which the inverse problem is based. Finally, inverse problems provide an excellent opportunity to bring interdisciplinary materials into the math classroom.

3 An example problem

In this section we present an inverse problem based on the familiar second order ordinary differential equation that is used to model harmonic motion. The problem posed will no be solved in full generality since the point of this paper is to introduce the idea of what an inverse problem is. Instead the assumption will be made that the problem is overdamped or critically damped. The underdamped case may require modifications to the approaches presented. This problem could easily be incorporated into most differential equations classes.

> Devise a method of recovering the coefficients in the 2nd order ODE
>
> $$mx''(t) + bx'(t) + kx(t) = f(t)$$
>
> from measurements of $x(t)$. You are free to select initial conditions and function(s) $f(t)$ in any manner you see fit. You may assume that the solution to the homogeneous equation approaches zero over time.

4 Two approaches to the solution

4.1 A data fitting method

Keeping in mind that this enterprise will go easier if some use of a priori knowledge is made, it is decided that $f(t) = t^2$ may be a good choice for the right hand side since this choice leads to a particular solution

$$x_p(t) = \frac{1}{k}t^2 - \frac{2b}{k^2}t - \frac{2m}{k^2} + \frac{2b^2}{k^3}.$$

If we let the system run until $x_p(t)$ is dominant, then perform a least squares fit of the form $a_2t^2 + a_1t + a_0$ to the data and solve for m, b, k from the computed values a_2, a_1, a_0. Clearly, using this approach the initial conditions are not particularly important.

4.2 Results for method one

An experiment using $m = 1$, $b = 10$, and $k = 49$ was run. Data was generated using ODE45 (in MATLAB, see the m-files in the appendix) to numerically determine the solution to the example problem with a tolerance of 10^{-6}. A least squares fit was performed on the data to determine values of a_2, a_1, and a_0 from which m, b, k were determined to be $m = .9875$, $b = 10.0065$, and $k = 48.9983$.

4.3 A nonlinear equation method

In this method, three different forcing functions of the form $F_0 \cos(\omega t)$ are used and the amplitude of the resulting steady state is measured. The analytic solution should have an amplitude

$$A(m, k, b) = \frac{F_0}{\sqrt{(k - m\omega^2)^2 + b^2\omega^2}}.$$

From three different ω values, a 3×3 system of nonlinear equations is obtained. This system of equations can be solved for m, b, and k using Newton's method. This approach has the advantage over the other method in that it could be easily be realized as an RLC circuit. In this case, real data could be gathered. Once again, the initial conditions are not important.

4.4 Results for method two

The forcing functions $f_1(t) = \cos(\omega t)$ for $\omega = 1, 2, 3$ were used on the same problem as above. The amplitudes corresponding to these forcing functions were computed and some noise was added to the data to simulate measurement error. The nonlinear system of equations was solved using Newton's method with the results $m = .9856, k = 9.9457$, and $k = 49.9930$.

5 Summary

The nature of inverse problems and their importance in industry and in the classroom was discussed. A sample problem and two possible avenues to a solution were demonstrated. The latter method could be used as a laboratory problem. In addition to reinforcing knowledge of the direct problems, these projects use common numerical techniques which is a valuable experience. We ought to do our best to make good use of technology in our classes. Towards this end, we should introduce projects which require it and which reflect to some degree the way it is used outside of an academic setting. Projects based on inverse problems meet these criteria.

6 References

[**1**] Anger, Gottfried, *Inverse Problems in Differential Equations*, Plenum Press (1990).

[**2**] Groetsch, Charles W., *Inverse Problems in the Mathematical Sciences*, Vieweg (1993).

[**3**] Gal-Ezar J., Zwas G., *Real-world models in the teaching of calculus*, The UMAP Journal 13 (1992) p93-100.

[**4**] Hadamard, J., *Four Lectures on Mathematics*, Columbia University Press, New York, 1915.

[**5**] *MATLAB User's Guide*, The MathWorks, Inc., 1993

A MATLAB m-files

A.1 M-files for the first approach

The first m-file is a script file for generating data via the numerical solution of the sample problem and performing the fit using the MATLAB command polyfit. The second m-file is a function necessary for use with the MATLAB routine ode45.

```
odeinv1.m

tol=10e-6; t0=0; tf=5;
f='yp'; y0=[1;0];
[T,U]=ode45(f,t0,tf,y0,tol,1)
q=length(T);
x=T(q-20:q); y=U(q-20:q,1);
p=polyfit(x,y,2);
```

```
yp.m

function yprime = yp(t,y)
% This function is necessary for solving the sample problem
% using ode45.  See MATLAB documentation for the use of ode45.

  yprime(1)=y(2);
  yprime(2)=t*t-10*y(2)-49*y(1);
```

A.2 M-files for the second approach

The first m-file is a function used to evaluate the Jacobian of the function in the second m-file. The third m-file computes the amplitudes of the analytic solution. The final m-file is a script which computes the solution via a simplified version of Newton's method.

jac.m

```
function y = jac(m,b,k,w)
% Evaluate the Jacobian for the example problem.
% m,b,k are scalars. w is a vector of 3 elements
% containing the frequencies of the data.
% jac(m,b,k,w)

y=zeros(3);
for j=1:3
 denom = ((k-m*w(j)*w(j))^2+b*b*w(j)*w(j))^(3/2);
 y(j,1) = (k-m*w(j)^2)*w(j)^2/denom; % derivative wrt m
 y(j,2) =  -b*w(j)*w(j)/denom; % derivative wrt b
 y(j,3) = (m*w(j)^2-k)/denom;
end
```

F.m

```
function y=F(m,b,k,w,amp)
% F(m,b,k,w,amp)  where m,b,k are scalars
% representing the unknown quantities,
% w is a vector with 3 elements containing the
% frequencies of the forcing functions used, and
% amp is a constant vector with 3 elements.

y=zeros(3,1);
for j=1:3
  y(j) = 1/((k-m*w(j)^2)^2+b^2*w(j)^2)^(1/2) - amp(j);
end
```

camp.m

```
function y=camp(m,b,k,w)
% camp(m,b,k,w) where m,b,k,w are scalars.
% returns the amplitude of the solution to
% a 2nd order ode mx"+bx'+kx=cos(wt).

y=1/((k-m*w^2)^2 + b^2*w^2)^(1/2);
```

odeinv2.m

```
for j=1:3
  amp(j) = camp(10,3,49,j);
end
w=1:3;
x=ones(3,1);
noise=2*rand(3,1)-1;
amp2=amp'+.001*noise*max(amp);
for i=1:10
  x=x-inv(jac(x(1),x(2),x(3),w))*F(x(1),x(2),x(3),w,amp2)
end
```