2012

Nonlinear curve fitting (linear plus exponential) for magnetic cooling data

Shannon Bourke
Eastern Michigan University

Nonlinear curve fitting (linear plus exponential) for magnetic cooling data

Abstract

Instruments sent into space must be cooled to temperatures lower than 50 milliKelvin to decrease noise and increase sensitivity. One way we cool the instruments is through the use of an adiabatic demagnetization refrigerator. To determine the best magnetic coolant we must measure the magnetic coolant's thermal resistance and heat capacitance. We do this by fitting curves to data taken during cooling experiments. We explore a model that uses an exponential plus linear fit. We use our model to predict data points and compare them to the actual data taken during the experiments, as well as explore the fitting properties with artificial data.

Degree Type

Open Access Senior Honors Thesis

Department

Mathematics

First Advisor

Dr. Andrew Ross

Nonlinear Curve Fitting (Linear Plus Exponential) for Magnetic Cooling Data

Shannon Bourke
Faculty Advisor: Dr. Andrew Ross

ABSTRACT

Instruments sent into space must be cooled to temperatures lower than 50 milliKelvin to decrease noise and increase sensitivity. One way we cool the instruments is through the use of an adiabatic demagnetization refrigerator. To determine the best magnetic coolant we must measure the magnetic coolant's thermal resistance and heat capacitance. We do this by fitting curves to data taken during cooling experiments. We explore a model that uses an exponential plus linear fit. We use our model to predict data points and compare them to the actual data taken during the experiments, as well as explore the fitting properties with artificial data.

A senior thesis submitted to the Eastern Michigan University honors college in partial fulfillment of the requirements for graduation with honors in mathematics

Table of Contents

1. Introduction

1.1 Background

Infrared spectroscopy is a powerful tool that enables astrophysicists to determine the distance to other galaxies in the universe, as well as their components. Space borne infrared spectrometers for astrophysics require detectors that are cooled to temperatures lower than 1K to decrease the noise and increase sensitivity. A common way to achieve sub-Kelvin temperatures is an adiabatic demagnetization refrigerator (ADR). ADRs use a magnetic coolant, or salt pill, suspended in the bore of a superconducting magnet. Salt pills are characterized by measuring heat capacity and the thermal resistance between the point where the instrument is attached and the salt pills. Heat capacity is a measure of the amount of heat required to change a substance's temperature by a certain amount, and thermal resistance is a measure of the temperature difference across a structure when energy flows through it during a certain amount of time.

1.2 Adiabatic Demagnetization Refrigerators

ADRs use heat and magnetic properties of materials to cool down instruments. There are four basic parts to an ADR:

1. The salt pill
2. The magnet
3. The thermal sink
4. The heat switch

The salt pill is a chemical mixture that is grown on a series of many gold wires. The pills take a few weeks to grow and there are a few different common

chemical formulas for them. The most important thing about the salt pill is that it is a paramagnetic material, meaning that each molecule in the substance has a small magnetic moment that can be aligned with an external magnetic field. In a non-paramagnetic material the magnetic moments cancel each other out so that the molecules do not line up with external magnetic fields. The magnet provides the external magnetic field that will align the magnetic moments of the molecules in the salt pill. The thermal sink will allow the heat to be dumped from the salt pill; this is normally a cryogen such as liquid nitrogen or liquid helium. Finally the heat switch will allow for scientists to control thermal contact between the salt pill and the thermal sink.

To cool the salt pill it is first placed in a strong external magnetic field, where the molecules will align themselves with this external field. At this point the molecules have a low magnetic energy. As the external field is decreased the molecules will twist out of alignment. For this to happen energy is required; this energy comes from the thermal energy of the molecules in the salt pill which causes the temperature of the salt pill and any thermal mass to decrease. Once the external magnetic field is zero, no more heat can be extracted from the pill and the pill slowly warms up. At this point the heat from the pill must be recycled. A heat switch is then turned on and an external magnetic field is increased. The heat of magnetization of the salt pill flows through the heat switch to the thermal sink. Once all (or enough) of the heat energy has moved to the thermal sink, the heat switch is turned off, breaking thermal contact. The external magnetic field is

now decreased again causing cooling. This process can be repeated many times.

1.3 Previous Work

This project began during the summer of 2011 at California Institute of Technology's Jet Propulsion Laboratory. The magnetic coolant, Chromic Cesium Alum, CCA, was cooled to various temperature ranges including 500mK, 150mK, 100mK, and 30mK. Figure 1 shows a plot of the curves during the cooling cycle.

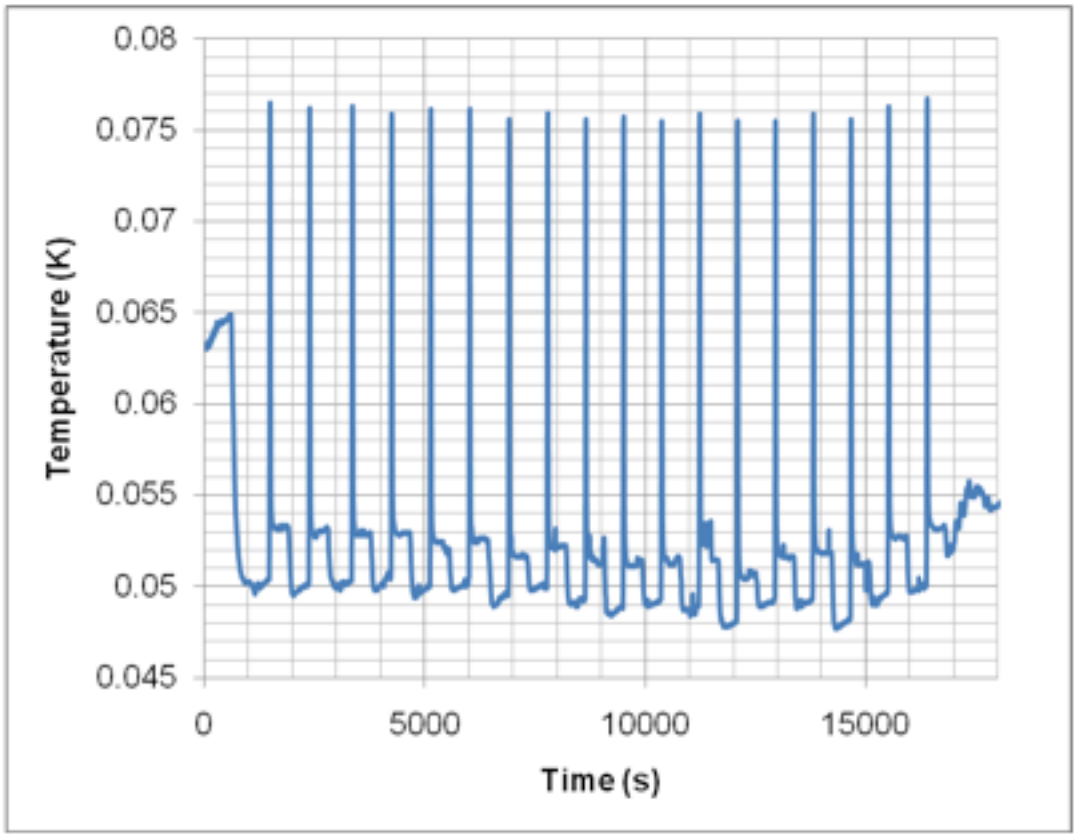

FIGURE 1

Each curve in Figure 1 has three segments. We can get different information from each segment. Figure 2 zooms in on one curve and shows each of the three segments. For this project, we focus in on Segment 3.

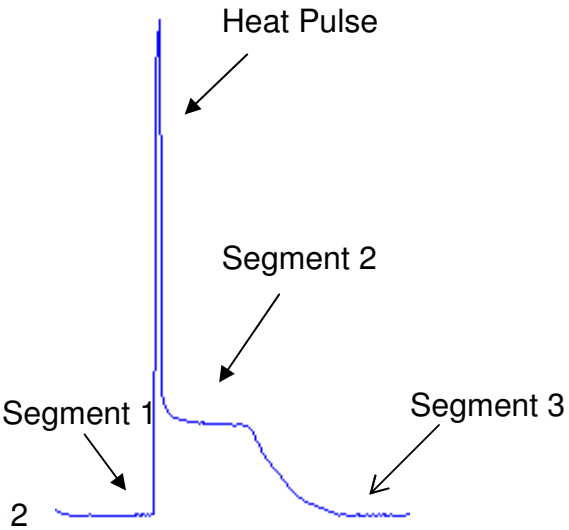

Figure 2

2. Model

2.1 Linear Plus Exponential

One possible model for the cooling data in phase 3 is a linear plus exponential model, following the equation:

$$\mathrm{f(t)} = \mathrm{m} * \mathrm{t} + \mathrm{b} + \exp(c * t + d) \qquad (1)$$

We chose to explore this model because the curves in the graph of the cooling data appear to decrease exponentially and then increase linearly.

To begin exploring this model we created an artificial set of data that closely modeled the various magnetic cooling data sets. We began with isolating each data trace and plotting it on its own graph. Figure 3 shows a plot of one of the data traces with its fitted curve.

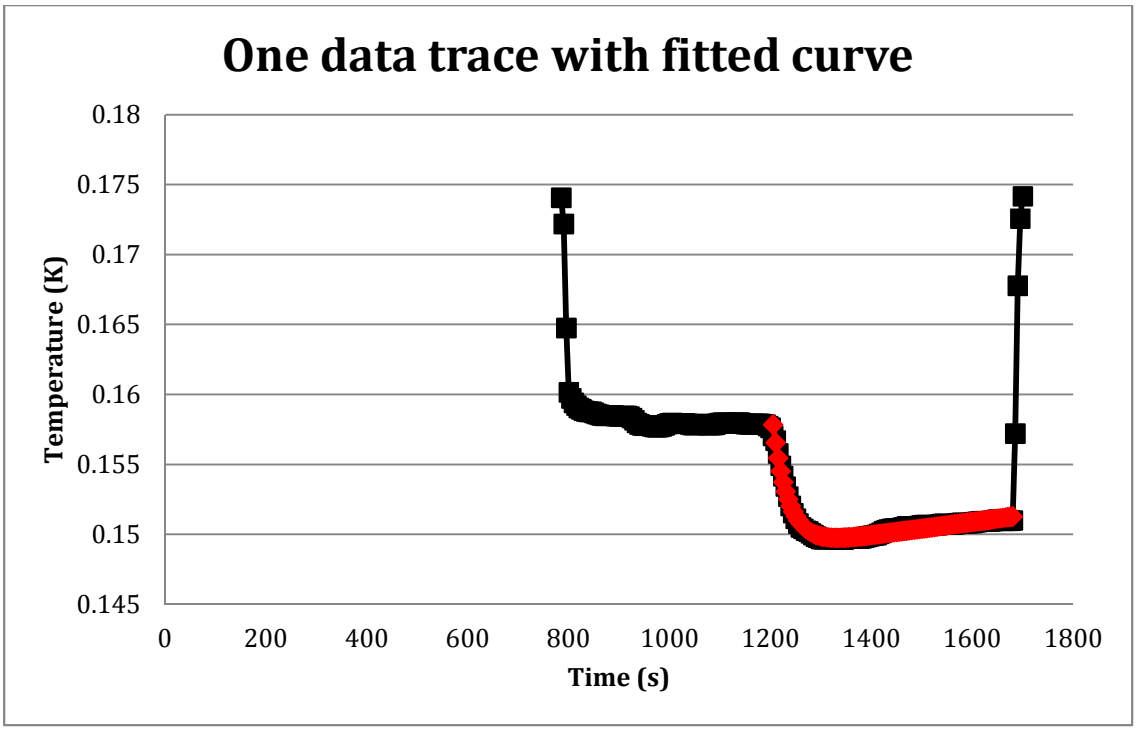

Figure 3

We used Microsoft Office Excel's solver to fit a curve with the same equation as Eq. 1 to one data trace. Solver was set up to change the variables m, b, c, and d, and to minimize the sum of the squares of the difference between the actual temperature and the predicted values. We then subtracted out the fitted curve so that we could get the residuals. The standard deviation of the residuals gave us a measurement of the noise in the data set. While there were clear correlations from one residual to the next, as a simple model we will suppose that they are independent. We then created a set of data that used the same values for m, b, c, and d as determined by solver. Finally we gave our artificial data set a similar amount of noise as an actual data set. We repeated this to obtain a total of 100 sets of artificial data. Figure 4 shows all 100 fitted curves on the same plot.

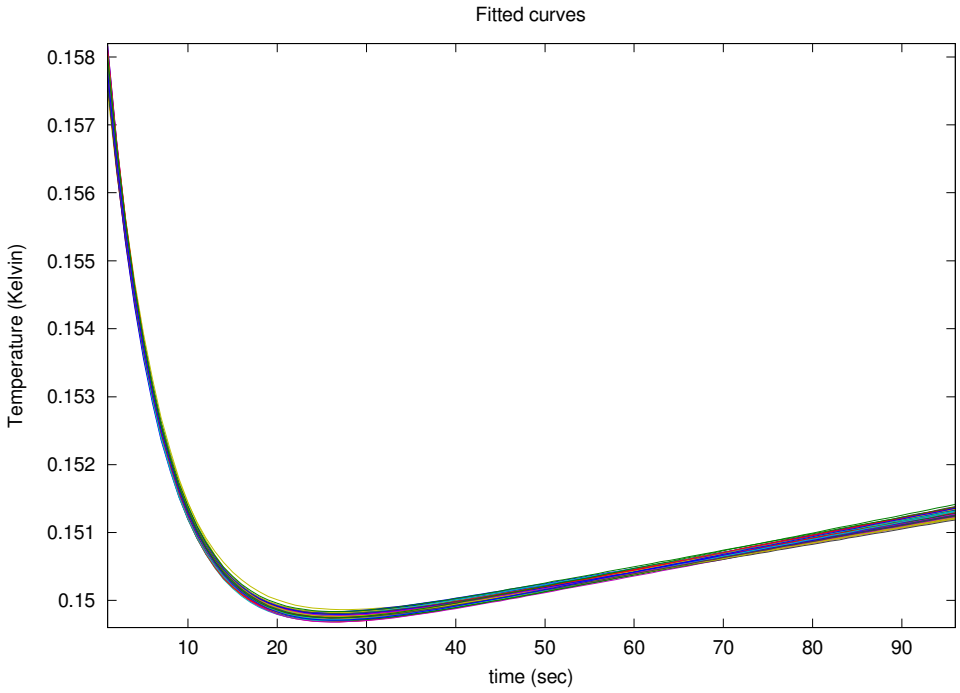

Figure 4

2.2 Differential Equation Model

A system that is heating up or cooling down can be described with Newton's Law of Heating or Cooling, Eq. 2.

$$k\frac{dT}{dt} = (T - T_0) \qquad (2)$$

This is the classic equation for a system that is heating up or cooling down, however in our system we can either add heat by turning on the heater, or we can subtract heat by turning on the ADR. This gives us an extra power term, so Eq. 2 will be as shown in Eq. 3.

$$k\frac{dT}{dt} = (T - T_0) + P(t) \qquad (3)$$

The curves in Figure 1 should follow Eq. 4.

$$P(t) = G(T - T_0) + C\frac{dT}{dt} \qquad (4)$$

Where P(t) is the power of heating or cooling at a given time, G is related to the thermal resistance, $T\text{-}T_0$ is the temperature difference during the cooling process, C is the heat capacity and $\frac{dT}{dt}$ is the temperature derivative with respect to time.

We are not entirely satisfied with these equations because if the power were constant during segment 3, then the curves should approach an equilibrium temperature, instead of a linear rise. However, from Figure 1, we see that this does not happen. This leads us to believe that constant power in this model would not be a good choice. Unfortunately the LabView data file did not contain power data, so we had to abandon this inquiry.

3. Discussion

3.1 Parallel Coordinate Plots

To determine how accurate our model was we created parallel coordinate plots. Parallel coordinate plots allow you to compare multiple variables to each other; here each simulation result (set of four parameter values) is represented by a set of four points connected by line segments. We created two different parallel axis plots in Octave (the code for each of these plots can be found in Appendix A). For the first plot, Figure 5, we compared the parameters from the 100 generated sets of data to the original parameters, using percent difference from true value.

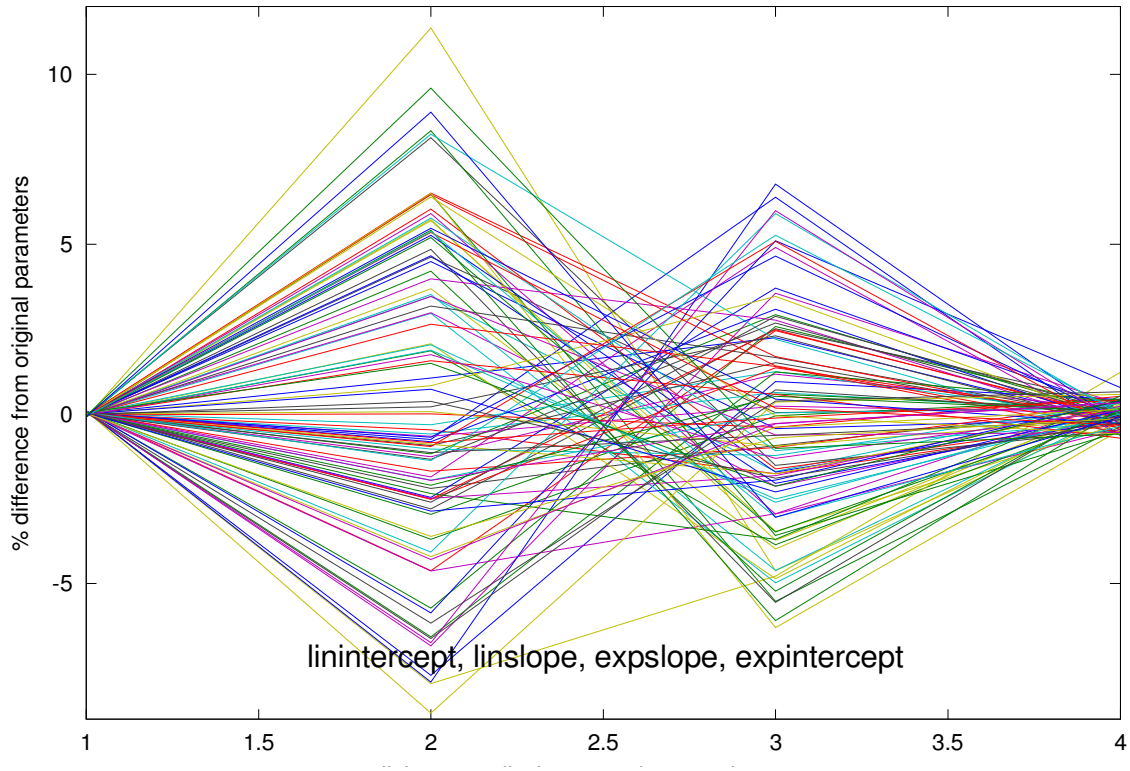

Figure 5

This plot essentially tells us how accurate our parameter estimates were. We can see from the plot that most of the parameters from the generated data sets are within +/- 5% of the original parameters, and all were within +/- 10%. We can also see from the plot that there is no high or low bias detected.

The second plot, Figure 6, shows the standardized estimates of the parameters. The Octave code used to create this plot can be found in Appendix A. This plot was created by subtracting the average value of the parameter estimates from each individual parameter estimate, and then dividing by the standard deviation.

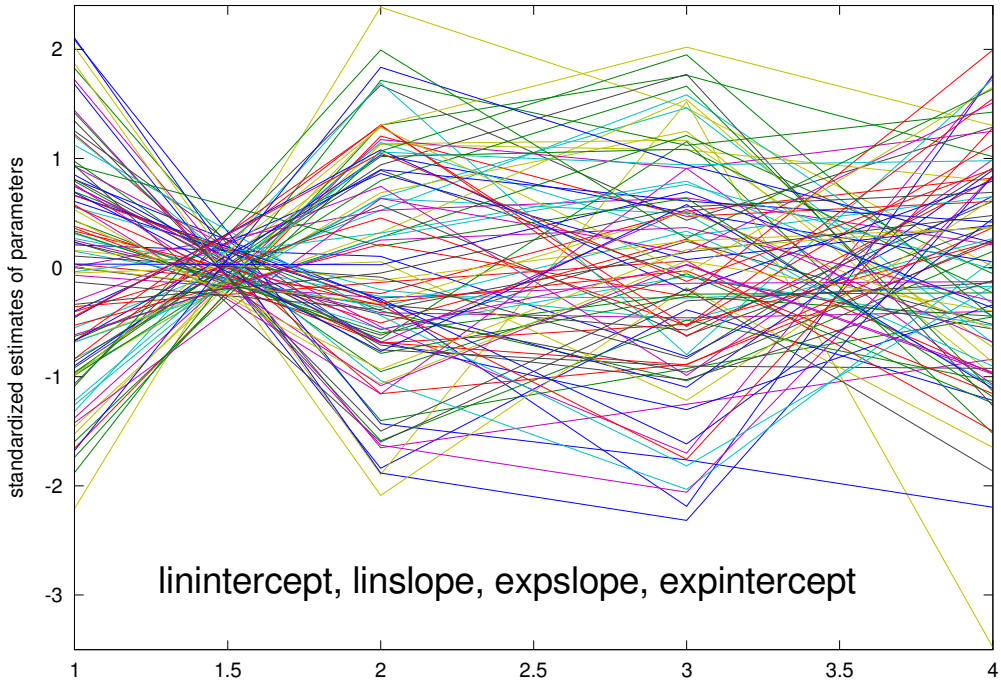

Figure 6

Unlike the previous plot, this plot allows us to see a relatively strong negative correlation between the linear slope and the linear intercept. We can also see a slight negative correlation between the linear slope and the exponential slope, as well as the exponential slope and the exponential intercept. These negative correlations are not as obvious due the there being less intersection of the lines, but when we look at it closely we can see that some of the lines do in fact intersect in between those variables.

3.2 Conclusions and Future Work

From Figures 5 and 6 we can tell that this linear plus exponential model is fairly reliable. We know this because of the percent differences between the parameters of the generated data sets and the original parameters are reasonably small. Additionally that fact that there is no high or low bias being

detected makes us even more confident in the reliability of our model. In the future, it would be good to do more work with the original differential equation.

Resources

Brando M. Development of a relaxation calorimeter for temperatures between 0.05K and 4K. *Review of Scientific Instruments* (2009) 80

Feng B, Ma W, Li Z, Zhang, X. Simultaneous measurements of the specific heat and thermal conductivity of suspended thin samples by transient electrothermal method. *Review of Scientific Instruments* (2009) 80

Hagmann C, Benford DJ, and Richards PL. Paramagnetic salt pill design for magnetic refrigerators used in space applications *Cryogenics* (1994) 34 213-219

Hagmann C, Richards PL. Adiabatic Demagnetization Refrigerators for Small Laboratory Experiments and Space Astronomy. *Cryogenics* (1995) 35 303-309

Hatta I. Heat Capacity measurements by means of thermal relaxation method in medium temperature range. *Review of Scientific Instruments* (1979) 50

Holmes W, Bock JJ, Bradford C Matt, Chui TCP, Koch TC, Lamborn AU, Moore D, Paine CG, Thelen MP, Yazzie A. Sub-Kelvin Cooler Configuration Study for the Background Limited Infrared Submillimeter Spectrometer BLISS on SPICA *Cryogenics* (2010) 50 516-521

Klaasse JC, Bruck EH. Heat capacity measurements on small samples: the hybrid method. *Review of Scientific Instruments* (2008) 79

Miyoshi Y, Morrison K, Moore JD, Caplin AD, Cohen LF. Heat capacity and latent heat measurements of CoMnSi using a microcalorimeter. *Review of Scientific Instruments* (2008) 79

The Adiabatic Demagnetization Refrigerator. *NASA* http://imagine.gsfc.nasa.gov/docs/teachers/lessons/xray_spectra/background-adr.html

Wilson Grant W, Timbie Peter T. Construction techniques for adiabatic demagnetization refrigerators using ferric ammonium alum *Cryogenics* (1999) 39 319-322

Appendix A: Octave Code

A.1: Octave code to create plots

```
% basic optimization

% original sample fitted values:
% from CCA1_150mK
% phase 3
%tstep  5
%tmin  0
%tmax  475

%lin intercept  0.149004126
%lin slope  4.787056E-06
%exp intercept  -4.728742274
%exp slope  -0.032241254
%sumsq  3.32789E-06
%std(resid)  0.000187164
linslope=  4.787056E-06;
linintercept=  0.149004126;
expslope=  -0.032241254;
expintercept=  -4.728742274;
sigma=  0.000187164
tvec=0:5:475;
ytrue = linslope * tvec +linintercept + exp(expslope*tvec + expintercept);

% sometimes we might want to use the same random values from one run to the next, so
we'll use:
seedval = 48197; % why not use a postal zip code?
rand("state",seedval);

noise = randn(size(ytrue)) * sigma;
y = ytrue + noise;

mydata = [tvec(:), y(:)];

pin = [linslope, linintercept, expslope, expintercept]; % the true values we started the
simulation with

objfunc=@(params) objfunc4(params,mydata);
%[f,xopt,gradopt,xhist,fhist,ahist] = steepest_descent(objfunc,x0,mydata)


y=y';
```

```
tvec=tvec';

% try a bunch of different sets of noise
nseeds = 100;
for ns = 1:nseeds
seedval = 48197+ns; % why not use a postal zip code?
rand("state",seedval);

noise = randn(size(ytrue)) * sigma;
y = ytrue + noise;

[f,p,kvg,iter,corp,covp,covr,stdresid,Z,r2]= ...
                leasqr(tvec,y,pin,'objfunc4leasqr');
optimal_params = p;
optimal_sumsq = sum( (f(:)-y(:)).^2 );

param_hist(ns,:) = optimal_params(:)';

% attempt to verify that leasqr found a good solution
% by trying random points nearby.
nsamp = 100;
sumsq = NaN*ones(1,nsamp);
for nsa=1:nsamp
pguess = optimal_params .* ( ones(size(optimal_params)) +
0.001*randn(size(optimal_params)) );
yguess = objfunc4leasqr(tvec,pguess);
sumsq(nsa) = sum( (yguess(:) - y(:)).^2 );
end % for nsa
if( min(sumsq) < optimal_sumsq ) % did our random searching happen to beat leasqr?
[ns, min(sumsq), optimal_sumsq ]
end
end % for nseeds

for np=1:min(size(param_hist))
tmp = param_hist(:,np);
% this is how we would standardize the various columns if we didn't know the true
parameter values
stdized(:,np) = (tmp - mean(tmp)) / std(tmp) ;
% but since we do know the true parameter values, we use them:
pctdiff(:,np) = 100* (tmp/pin(np) - 1); % computes percent difference from true original
parameter value
end

figure
plot(stdized'); % make a parallel-axes plot
[linslope, linintercept, expslope, expintercept]
xlabel('linslope, linintercept, expslope, expintercept')
```

```
ylabel('standardized estimates of parameters')
figure
plot(pctdiff');
xlabel('linslope, linintercept, expslope, expintercept')
ylabel('% difference from original parameters')

% try plotting all of the fitted curves on one plot
figure
fitted_curves = [];
fitted_minus_true = [];
for nn=1:nseeds
f = objfunc4leasqr(tvec, param_hist(nn,:) );
fitted_curves = [fitted_curves, f(:)];
fitted_minus_true = [fitted_minus_true, (f(:) - ytrue(:)) ];
end
plot(fitted_curves)
xlabel('time (sec)')
ylabel('Temperature (Kelvin)')
title('Fitted curves')
figure
plot(fitted_minus_true)
xlabel('time (sec)')
ylabel('Temperature Difference (Kelvin)')
title('Fitted curves minus Original curve')
```

A.2: Linear plus exponential function

```
function [ predictions] = objfunc4leasqr(xvals,params)


linslope = params(1);
linintercept = params(2);
expslope = params(3);
expintercept = params(4);

predictions = linslope * xvals + linintercept + exp(expslope* xvals + expintercept);
```