Lab 4 notes

In this lab, I created fake data with errors that should approximately be fit by $y = 6.3t^{.51}$ and then used nonlinear least squares to determine the parameters of the model. I then transformed the data, by taking the natural logarithm, into a problem that can be solved using linear least squares and again found parameters for the model. The two results are then compared by summing the square of the errors.

Here are the subroutines I used for the nonlinear least squares.

This calculates the error vector.

```
function x=r(t,y,c,k)

m=length(y);
x=zeros(m,1);

for i=1:m
x(i)=(y(i)-c*t(i)^k);
end
```

This function returns the Jacobian of r defined above.

```
function J=Dr(t,y,c,k)

m=length(y);
J=zeros(m,2);

n=2;

  for i=1:m
    J(i,1)=-t(i)^k;
    J(i,2)=-c*log(t(i))*t(i)^k;
  end
```

Here is a copy of the Matlab session.

```
octave:2> t=linspace(1,3)';
octave:3> y=6.3.*t.^(.51)+(rand(100,1)-1).*.2;
octave:4> plot(t,y,'x')
octave:5> c=1
c = 1
octave:6> a=1
a = 1
octave:7> for i=1:50
> del=-inv(Dr(t,y,c,a)'*Dr(t,y,c,a))*Dr(t,y,c,a)'*r(t,y,c,a);
> c=c+del(1);
> a=a+del(2);
> end
octave:8> c
c = 6.2015
octave:9> a
a = 0.51478
```

We will now try to use linear least squares to fit a straight line through transformed data. If the data is of the form $y = ct^{at}$ and we take the natural logarithm of both sides we get $\ln(y) = \ln(c) + a\ln(t)$. Or $\ln(y)$ is linear in $\ln(t)$ with slope a and intercept $\ln(c)$. So we just need to find the line of best fit for the transformed data $(\ln(t), \ln(y))$. The slope of this line will be our a and the intercept of this line will be $\ln(c)$. I carry out the calculations below:

```
octave:10> t1=log(t);
octave:11> y1=log(y);
octave:12> A=zeros(2)
A =
```

```
  0  0
  0  0

octave:13> A(1,1)=length(t1)
A =

  100    0
    0    0

octave:14> for i=1:100
> A(1,2)=A(1,2)+t1(i);
> A(2,2)=A(2,2)+t1(i)^2;
> end
octave:15> A(2,1)=A(1,2)
A =

  100.000   64.692
   64.692   51.549

octave:16> b=zeros(2,1)
b =

  0
  0

octave:17> for i=1:100
> b(1)=b(1)+y1(i);
> b(2)=b(2)+y1(i)*t1(i);
> end
octave:18> newcs=inv(A)*b
newcs =

  1.82428
  0.51550

octave:19> c1new=exp(newcs(1))
c1new = 6.1983
octave:20> c2new=newcs(2)
c2new = 0.51550
octave:21> sum1=0
sum1 = 0
octave:22> sum2=0
sum2 = 0
octave:23> for i=1:100
> sum1=sum1+(y(i)-c*t(i)^a)^2;
> sum2=sum2+(y(i)-c1new*t(i)^c2new)^2;
> end
octave:24> sum1
sum1 = 0.29663
octave:25> sum2
sum2 = 0.29695
octave:26> diary off
```