

Contents

- [Project 6](#)
- [Part 1: Importing the data and linear regression](#)
- [Part 2: Calculation of thermodynamic parameters](#)
- [Part 3: Trouton's Rule](#)
- [Part 4: Error Calculations...](#)
- [Part 5: Plotting the residuals...](#)
- [Conclusion:](#)

Project 6

In this project we are examining the vapor liquid equilibria of fluorobenzene and benzenethiol, where the temperature at which the liquids boil at a range of pressures is measured. We will do both linear regression to an ideal thermodynamic model (the August Equation, a simplification of the Antoine Equation assuming constant heat of vaporization), and then examine both error calculations and the thermodynamic behavior including an extra term.

```
clear
```

Part 1: Importing the data and linear regression

We first import the data from the provided excel sheet, break it apart into the appropriate columns, and then plot it up. We follow it with fitting to a line (the August Equation). As can be seen, the data –looks– pretty linear, but as we shall find it really isn't.

```
data=xlsread('LVEquilibriumData.xls',1,'','basic'); % We import the data

% We break it up and convert to 1/Kelvin and log(p/p0):
itf=1.0./(data(:,2)+273.16);
lpf=log(data(:,3)/760);
itb=1.0./(data(:,5)+273.16);
lpb=log(data(:,6)/760);

% We plot it up:
figure(1)
plot(itf,lpf,'ob',itb,lpb,'*r')
xlabel('1/T (Kelvin)')
ylabel('log(p/p0)')
legend('Fluorobenzene','Benzenethiol')

% We do the regression:
af=[ones(size(itf)),itf];
kf=inv(af'*af)*af';
xf=kf*lpf;

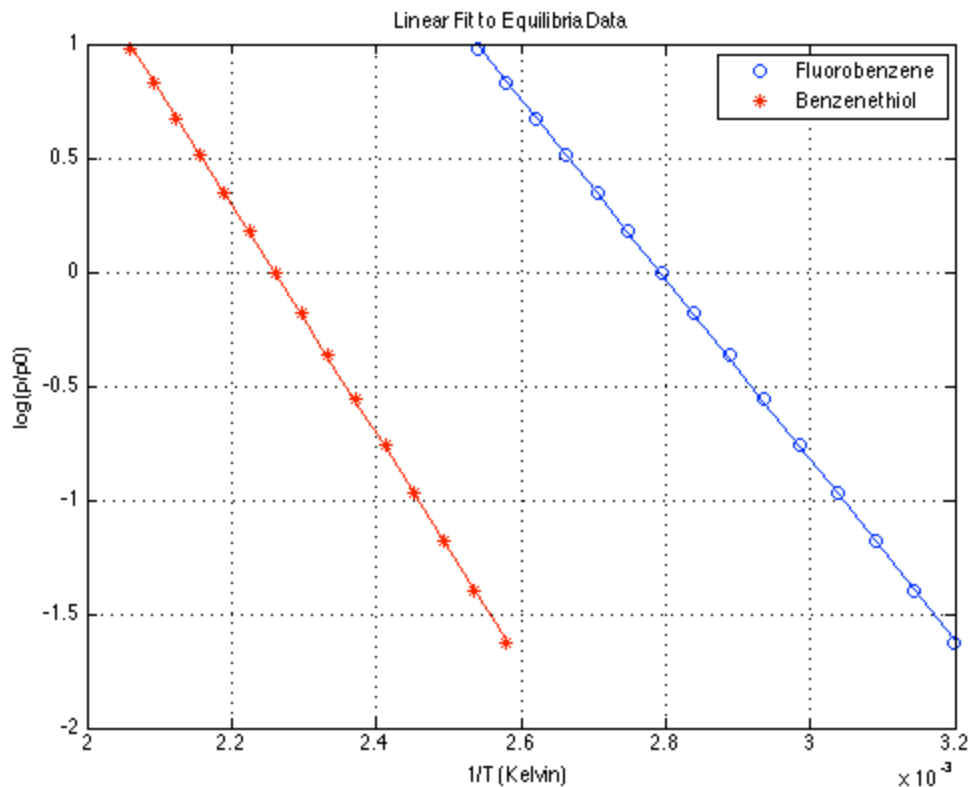
ab=[ones(size(itb)),itb];
kb=inv(ab'*ab)*ab';
xb=kb*lpb;

%We could have done these using the \ command, of course, but we will need
%kf and kb later...

%We add the regression lines to the plot:
```

```
figure(1)
hold on
plot(itf,af*xf,'b',itb,ab*xb,'r')
hold off
grid on
title('Linear Fit to Equilibria Data')
```

Warning: XLSREAD has limited import functionality on in basic mode. Refer to HELP XLSREAD for more information.



Part 2: Calculation of thermodynamic parameters

The heat of vaporization and entropy of vaporization can be calculated from the slope and intercept of each line, and compared to the values obtained from the NIST website. We will look at entropy in part 3, but now we focus on the heat of vaporization. As you can see, the values are pretty close.

```
gasr=8.3144621; %Gas constant in units of J/mol*deg K

hvapf=-xf(2)*gasr*1e-3;
hvapfnist=34.68; %This is the value reported at standard conditions

hvapb=-xb(2)*gasr*1e-3;
hvapbnist=47.56; %Again at standard conditions for benzenethiol

%We can present this in a table form:
hvaptable=num2str([hvapf,hvapb;hvapfnist,hvapbnist]);
```

```

left=str2mat('linear fit      ','NIST data      ');
hvaptable=[left,hvaptable];
top='Hvap (kJ/mol)  Fluorobenzene Benzenethiol';
hvaptable=str2mat(top,hvaptable);
disp(hvaptable)

```

```

Hvap (kJ/mol)  Fluorobenzene Benzenethiol
linear fit      32.856      41.739
NIST data       34.68      47.56

```

Part 3: Trouton's Rule

We use the calculated intercept to evaluate the entropy of vaporization and compare that to Trouton's Rule. Again, the values are close to Trouton's Rule, but a bit higher. That would indicate that the liquid state is lower entropy due to non-ideal association between the molecules.

```

dsf=xf(1)*gasr;
dsb=xb(1)*gasr;
trouton=88;
dstable=num2str([dsf,dsb,trouton])
dstable=str2mat('deltaS (J/mol)', 'Fluorobenzene Benzenethiol Trouton',dstable);
disp(dstable)

```

```

dstable =

91.7358      94.2902      88

deltaS (J/mol)
Fluorobenzene Benzenethiol Trouton
91.7358      94.2902      88

```

Part 4: Error Calculations...

We calculate the error in the heat of vaporization and the entropy, and determine the 95% confidence interval, using the regression approximations that the model is perfect (e.g., that deviation is due to random experimental error) and that the error in the data points is uniform. As we shall see in part 5, of course, this assumption is quite incorrect! A good lesson in why you should always plot your residuals, and to be very cautious in evaluating reported errorbars!

```

rf=af*xf-lpf; %The residual for fluorobenzene
varpf=norm(rf)^2/(length(lpf)-2); %The variance
varxf=kf*varpf*kf'; %The matrix of covariance in the fluorobenzene fit
sigxf=diag(varxf).^5; %The standard deviations of the two parameters

rb=ab*xb-lpb; %The same for benzenethiol
varpb=norm(rb)^2/(length(lpb)-2);
varxb=kb*varpb*kb';
sigxb=diag(varxb).^5;

% Now we report the 95% confidence intervals. We get the t statistic:
t=tinv([.025 .975],length(lpf)-2);

```

```
finterval=[hvapf+t*sigxf(2)*gasr*1e-3;dsf+t*sigxf(1)*gasr];
binterval=[hvapb+t*sigxb(2)*gasr*1e-3;dsb+t*sigxb(1)*gasr];

conftable=str2mat('Fluorobenze',num2str(finterval),' ','Benzenethiol',num2str(finter
left=str2mat(' ','Hvap (kJ/mol) ','Svap (J/mol) ',' ',' ','Hvap (kJ/mol) ','Svap (J/
conftable=str2mat(' ','95% Confidence Intervals',' ',[left,conftable]);
disp(conftable);
```

95% Confidence Intervals

| | Fluorobenze | |
|---------------|-------------|---------|
| Hvap (kJ/mol) | 32.6379 | 33.0742 |
| Svap (J/mol) | 91.1121 | 92.3595 |

| | Benzenethiol | |
|---------------|--------------|---------|
| Hvap (kJ/mol) | 32.6379 | 33.0742 |
| Svap (J/mol) | 91.1121 | 92.3595 |

Part 5: Plotting the residuals...

It is always important to plot the residuals to check the assumptions going into error calculations. Here we find that the residuals are actually quite systematic, meaning that the linear fit didn't completely capture the data (in other words, that the heat of vaporization wasn't constant!) and thus we have both overestimated the random errors in the data and have underestimated the uncertainty in the calculated values of delta H. To get a better estimate of the range, we shall add a quadratic term to the model and recalculate the fitting parameters. The heat of vaporization is assumed to be proportional to the local slope of the fitted curve.

```
figure(2)
plot(itf,rf,'bo',itb,rb,'*r')
xlabel('1/T (Kelvin)')
ylabel('log(p/p0)')
legend('Fluorobenzene','Benzenethiol')
title('Residuals from Linear Fit')

% Now we fix this by adding another term to the models:
af2=[af,itf.^2];
ab2=[ab,itb.^2];

xf2=af2\lpf;
xb2=ab2\lpb;

% and we get the range of heats of vaporization:

hvapf2=-gasr*1e-3*(xf2(2)+2*xf2(3)*itf);
hvapb2=-gasr*1e-3*(xb2(2)+2*xb2(3)*itb);

% The range of values is thus the minimum and maximum of these values:
rangehvap=[min(hvapf2),max(hvapf2);min(hvapb2),max(hvapb2)];
hvaptable=num2str(rangehvap);
left=str2mat('Fluorobenzene ','Benzenethiol ');
hvaptable=str2mat('Range of Heats of Vaporization (kJ/mol)',[left,hvaptable]);
disp(hvaptable);
```

```

% We can also compare this range graphically with the formula on the NIST
% website as a function of temperature. This is given in terms of a three
% parameter model in terms of the reduced temperature. They also give a
% domain of validity for their formula:

trangef=[318:382];
nistf=49.8*exp(-0.283*trangef/560.1).*(1-trangef/560.1).^283;

trangeb=[298:417];
nistb=61.79*exp(-.262*trangeb/689.5).*(1-trangeb/689.5).^262;

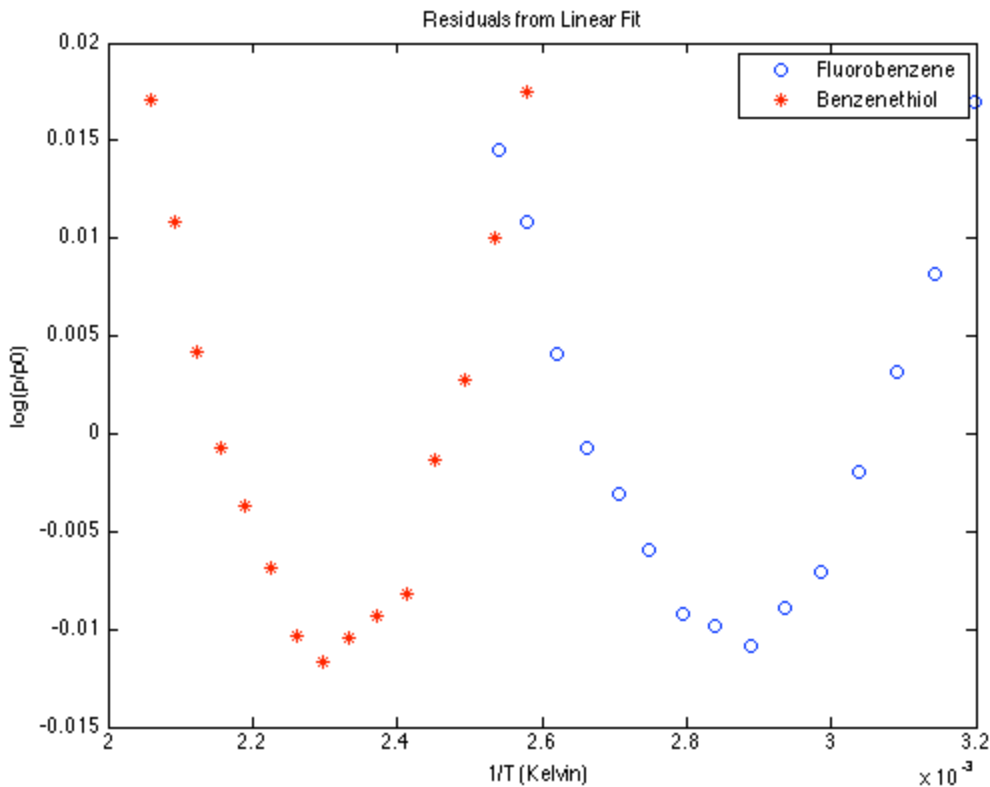
figure(3)
plot(1.0./itf,hvapf2,'b',1.0./itb,hvapb2,'r',trangef,nistf,'b--',trangeb,nistb,'r--')

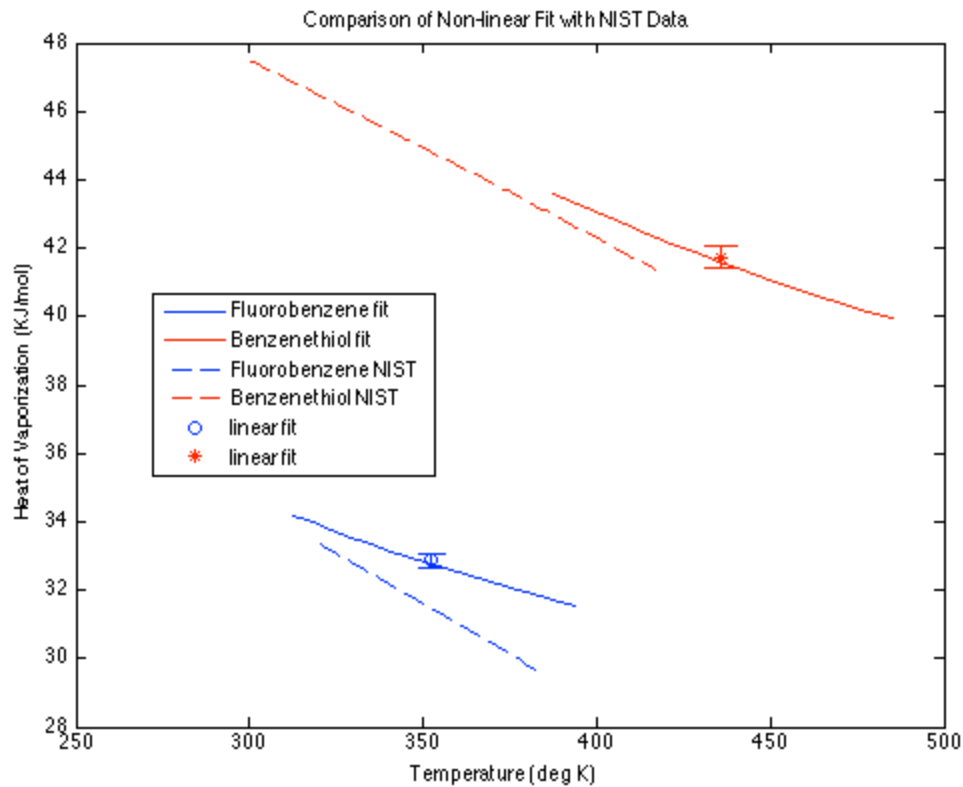
% We add in the linear approximations with errorbars:
hold on
errorbar(mean(1.0./itf),hvapf,tinv(.975,length(itf)-2)*gasr*1e-3*sigxf(2),'bo')
errorbar(mean(1.0./itb),hvapb,tinv(.975,length(itb)-2)*gasr*1e-3*sigxb(2),'r*')
hold off
xlabel('Temperature (deg K)')
ylabel('Heat of Vaporization (KJ/mol)')
title('Comparison of Non-linear Fit with NIST Data')
legend('Fluorobenzene fit','Benzenethiol fit','Fluorobenzene NIST','Benzenethiol NIST')

```

Range of Heats of Vaporization (kJ/mol)

| | | |
|---------------|---------|---------|
| Fluorobenzene | 31.554 | 34.1984 |
| Benzenethiol | 39.9349 | 43.5975 |





Conclusion:

From the plots, it is pretty clear that using the linear fit makes some pretty serious errors in calculating the range of heats of vaporization, which isn't constant (as assumed by the model). This yields a systematic dependence of the residual, and thus dramatically underestimates the error in the calculated values. The moral of the story is to -always- plot your residuals!