ABE 557 | Prof. Okos

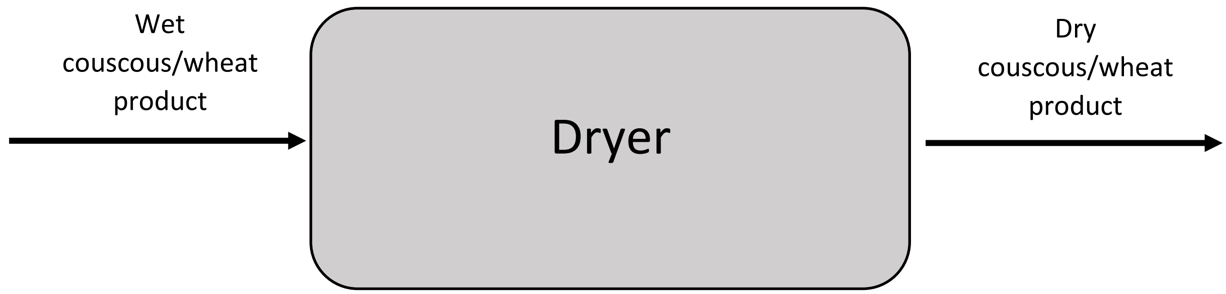
Dryer Design Part 1

Nathan LeRoy

Oct. 25th, 2018

The System and Background

In this unit operation, we are drying two wheat products continuously in a dryer. The wheat products are known as semolina and farina:



Before we design the drying conditions, it is important to understand how water activity relates to the moisture content of our products. For this, we will employ the GAB model. The GAB model will calculate a moisture content of a material given a specific water activity and three constants, M0, C, and K.

$$M = \frac{M0CKa_w}{(1 - Ka_w)(1 + (C - 1)a_w)}$$

The GAB model, however, is only valid for a constant temperature system as the three constants M0, C, and K are all dependent on temperature via an Arrhenius-type equation:

$$X(T) = X_0 e^{\frac{E_a}{RT}}$$

We do not know what Ea, or Xo is for any of the constants, but we are given data on the constants as they change with temperature. We can rearrange the previous equation to obtain:

$$ln[X(T)] = ln[X_0] + \frac{E_a}{R}\frac{1}{T}$$

When we plot the natural logarithm of the constants versus 1 over the temperature, we obtain a linear relationship which allows us to calculate each constants Xo value and the activation energy. After this, we can substitute the Arrhenius-type equations into the GAB model to now achieve an equation for moisture content as a function of both activation energy and temperature. A plotted GAB equation at a single temperature is known as an isotherm. Multiple isotherms for each product were plotted.

In addition to the isotherms, we wanted to investigate the binding energy of moisture inside our food product. Binding energies are calculated by examining differences between the water activities as they change when temperature changes. To do so, we can use the following equation:

$$\ln \frac{a_{w2}}{a_{w1}} = \frac{E_b}{R}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)_M$$

Finally, we can also characterize the food product via diffusion coefficient. This characterizes how quickly water moves through the product. The following equation can be used to calculate that:

$$D_{eff} = D_o e^{-E_a/RT}\left(\frac{Ke^{-E_a/RT}}{1 + Ke^{-E_a/RT}}\right)$$

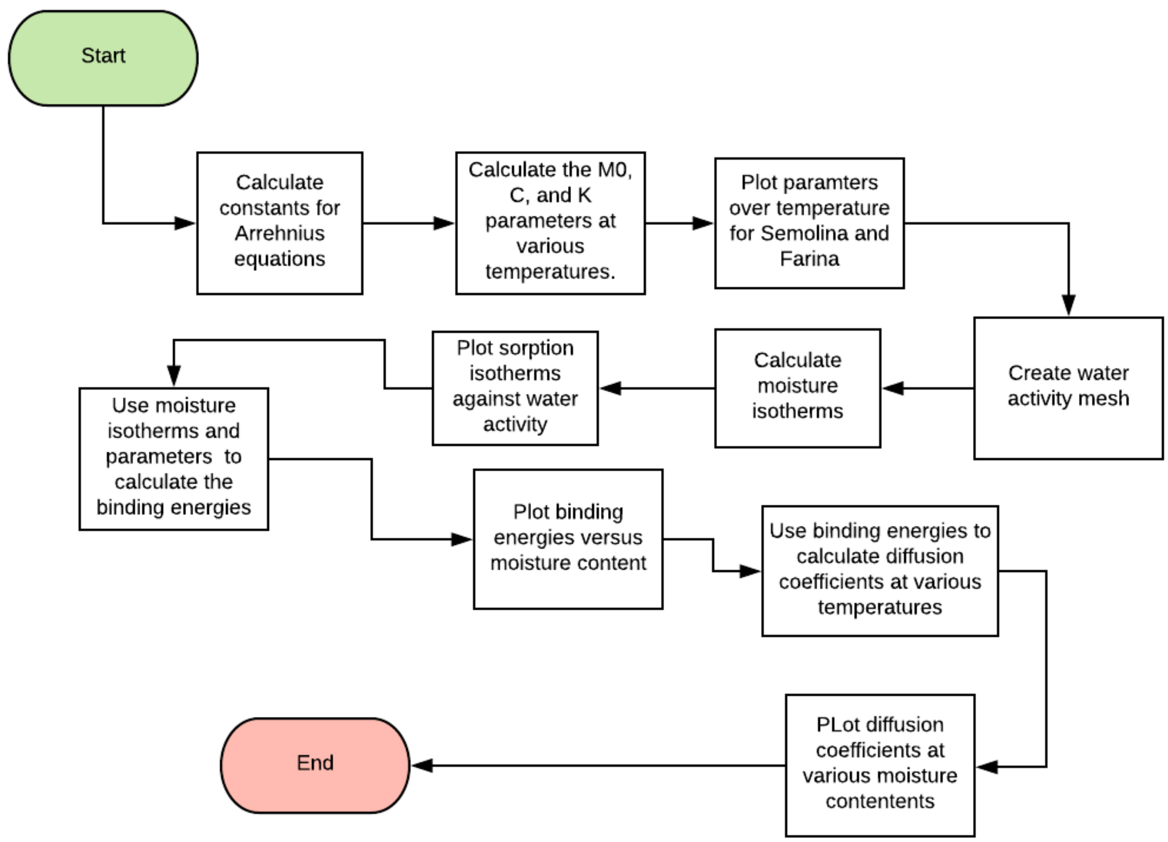
When we have this information, we can then begin to design our dryer. Before, however, we need to investigate the glass transition temperature of our material. The glass transition temperature is the temperature at which our food product begins to turn from a viscous rubbery state, to a hard-amorphous state. For this, we can employ the Fox equation:

$$\frac{1}{T_g} = \sum \frac{X_i}{T_{g_i}}$$

Where, X_i is the mass fraction of a component in the product and $T_{g,I}$ is the glass transition temperature of the specific component in our product.

Algorithm Steps

The code is very procedural and flows easily. It does not jump around to solve parameters, rather uses analytical mathematics to calculate the necessary values and data.



Part B uses a separate script to calculate the glass transition temperatures and the dryer operation temperatures. We wanted to keep the operating temperature at least 10 degrees C above the glass transition temperature but not over 50 degrees above the glass transition temperature. The following pseudo code represents the algorithm used to find these operating temperatures:

```
op_temp = moisture_content.60.Tg() + 10
for each moisture_content {
     min_temp = moisture content.Tg()
     }
     if op_temp < min_temp {
     }
          op_temp = moisture_content.Tg() + 50 {
          }
```

See the code section for the exact code used to calculate these operating temperatures.

The Results

GAB Model Constants – The following results were found for each product and their respective Arrhenius Equation's constants:

	M0		**C**		**K**	
	ΔH	$M0_o$	*ΔH*	C_o	*ΔH*	K_o
Semolina	2.195e4	0.0014	-6.539e4	1.299e12	-2.899e3	2.156
Farina	1.295e4	0.0424	-4.030e4	2.011e8	-2.280e3	1.665

These constants were used to calculate how the GAB constants change with temperature using the Arrhenius-type equation. In general, it was found that both the K and C constant increased with time, while the monolayer constant, M0 actually decreased with time for both Semolina and Farina (Fig. 1).

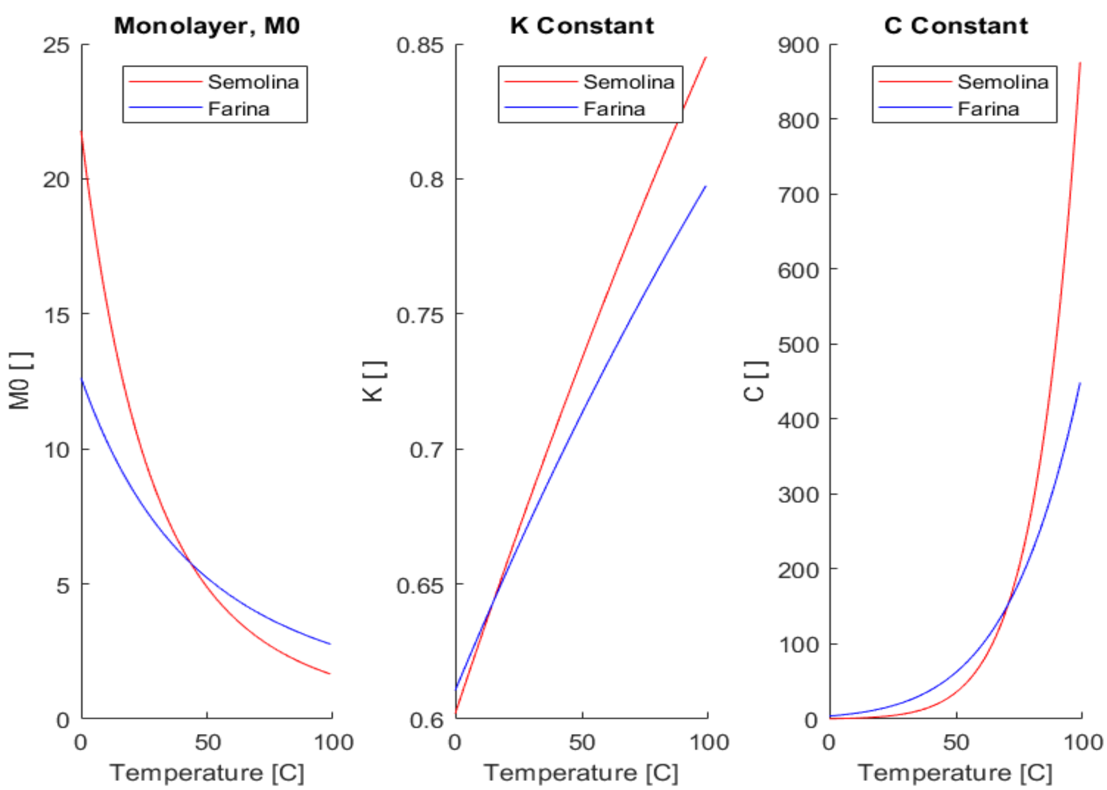


Figure 1 The GAB equations plotted against temperature.

Moisture Isotherms – In general, for all isotherms and food products, we see that as the water activity increases, the moisture content inside the product increases with a plateau around a water activity around 0.5. Please see Fig. 2 below for the graphs of the isotherms. It should be noted that these isotherms were calculated with the fitted data. We see that some of the isotherms cross, which indicates that one water activity corresponds to one moisture content at two different temperatures – which is impossible. One reason for this issue could be due to poorly fitted data or erroneous data collection.

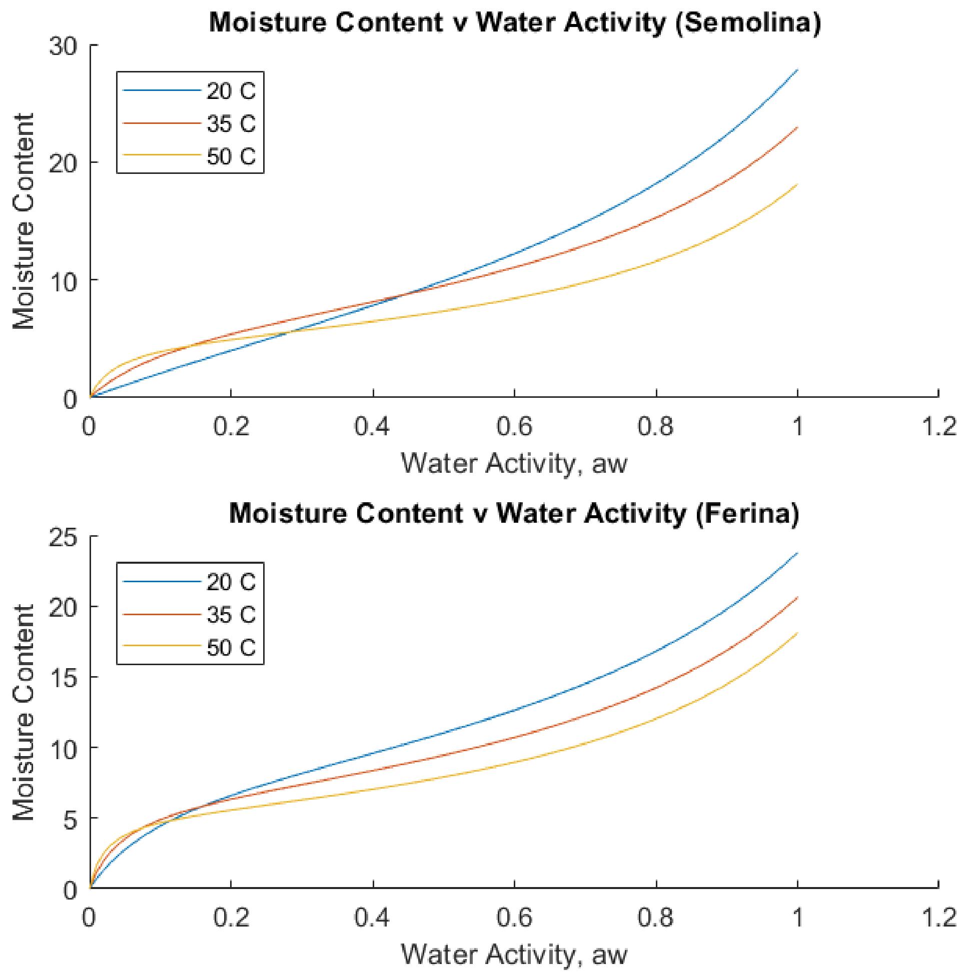


Figure 2 Moisture isotherms for both semolina and farina

Binding Energies and Diffusion Coefficients – The binding energies of moisture to our food product is essential to calculating the diffusion coefficients of water through our food. As the binding energy of the food product increases, the diffusion coefficient decrease. This can be seen mathematically in the effective diffusion coefficient equation or be thought of in the sense that an increase in binding energy is associated with a more difficult time for water to migrate through the food product. The binding energy increases as the moisture content increases and it peaks around 7.5 g H_2O/g DS for Semolina and around 9.5 g H_2O/g; it then decreases as we continuously increase the moisture content DS (Fig 3.).

These phenomena are reflected with the diffusion coefficients. For each temperature, they decrease as the moisture content increases, reach a peak low, and then continuously increase as the binding energy begins to decrease again (Fig. 4). All of this is true for both the Semolina and the Farina food products.

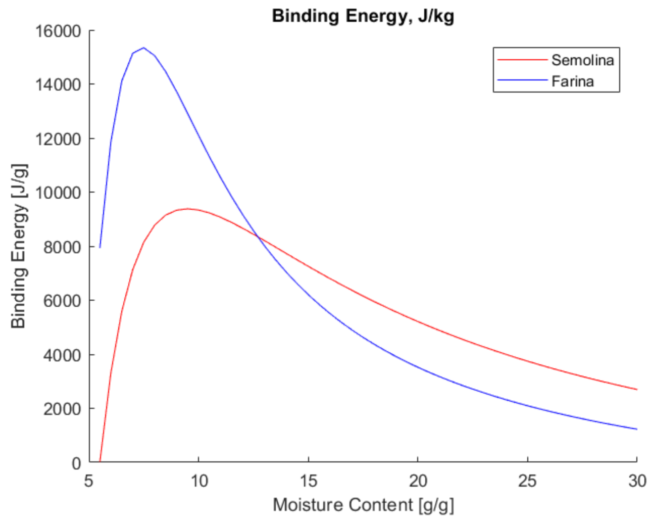


Figure 3 Binding energies for both Semolina and Farina as function of moisture content

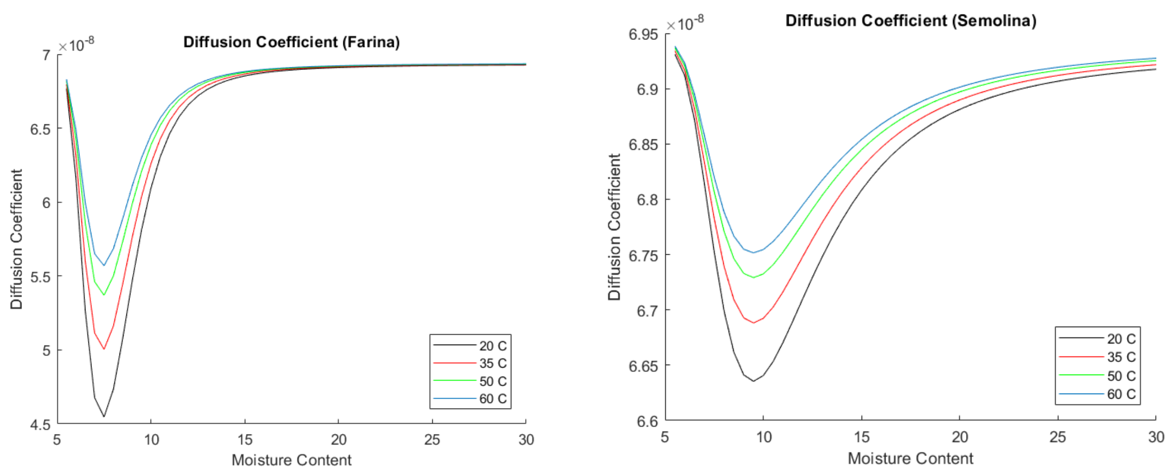


Figure 4 Diffusion coefficients for both Semolina and Farina as a function of moisture content.

Glass Transition and Operating Conditions – Part B of the algorithm asked us to calculate the glass transition temperature as a function of moisture content, and then subsequently design the dryer around these data. In general, the glass transition temperature decreases as the moisture content increases in our food product. This makes sense as the amount of water in out food product is increasing. The curve is logarithmically shaped, which makes sense given the equation for glass transition temperature (Fig. 5). On the same graph, we can see the drying stages which are set based on the glass transition temperature and the defined operating conditions.

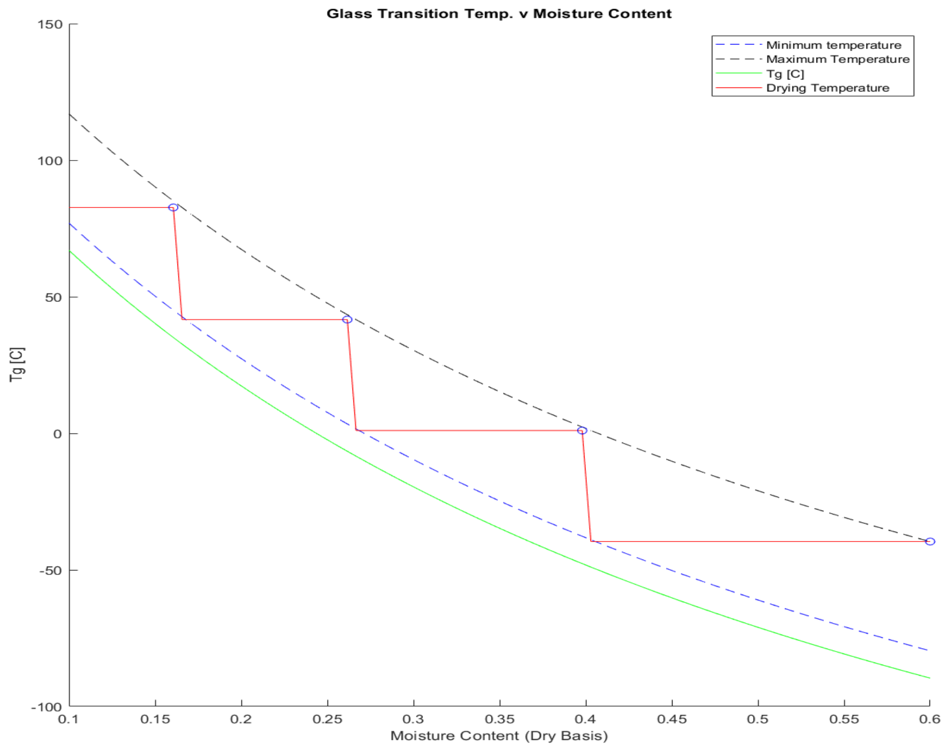


Figure 5 Glass transition temperature as a function of moisture content. The drying temperatures are also plotted on this graph.

The MATLAB script calculated and outputted the following drying conditions based on the found stages:

A 4 stage dryer using the following conditions:

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Stage 1: Air at -39.62 C and 97.55% RH
Stage 2: Air at 1.03 C and 6.35% RH
Stage 3: Air at 41.64 C and 0.33% RH
Stage 4: Air at 82.70 C and 0.03% RH
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

If we separate a dryer into four stages, we should be able to dry our product to the necessary moisture content without having to sacrifice any product or nutrient quality.

Conclusions

- As temperature increases, both he C and K constant increase in the GAB model for both Semolina and Farina as temperature increases. The M0 constant decreases as temperature increases for both Semolina and Farina.
- The sorption isotherms follow normal isotherms for similar products, however they cross indicating one moisture content corresponds to one water activity at two temperatures. This is incorrect thermodynamically and most likely is an artifact of poor data.
- Binding energy increases as moisture content increases. It peaks between a 0.75 and 1.0 moisture content, and then it proceeds to decrease with increasing moisture content.
- Diffusion coefficient reflects the binding energy curve
- Our dryer should employ 4 stages:
    - Stage 1: Air at -39.62 C and 97.55% RH
    - Stage 2: Air at 1.03 C and 6.35% RH
    - Stage 3: Air at 41.64 C and 0.33% RH
    - Stage 4: Air at 82.70 C and 0.03% RH

Code Part A

```
%Clear command window, clear variable workspace, and close all figures
clc
clear all
close all

%Semolina Constants%
M0s = [11.8, 6.45, 5.92, 3.53]; % Data on monolayer constant
Ks = [0.65, 0.71, 0.72, 0.76]; % Data on K constant
Cs = [4.21, 8.77, 10.1, 200.1]; % Data on C constant


%Farina Constants
M0f = [9.16, 6.01, 5.23, 4.76];
Kf = [0.65, 0.69, 0.71, 0.73];
Cf = [14.06, 31.95, 37.04, 137.0];

%Create temperature Vector
T_C = [20, 35, 50, 60]; % Temps in C
T_K = T_C + 273; % Temps in Kelvin

%Define generic constants
R = 8.314; %J/mol-K

%Apply a linear regression analysis to each data set against the inverse of
%the temperature
coeffs_s_M0 = polyfit(1./T_K,log(M0s),1);
coeffs_s_K = polyfit(1./T_K,log(Ks),1);
coeffs_s_C = polyfit(1./T_K,log(Cs),1);
coeffs_f_M0 = polyfit(1./T_K,log(M0f),1);
coeffs_f_K = polyfit(1./T_K,log(Kf),1);
coeffs_f_C = polyfit(1./T_K,log(Cf),1);

%Calculate Parameters for semolina%
%See the backround section of the report
M0o_s = exp(coeffs_s_M0(2));
Ko_s = exp(coeffs_s_K(2));
Co_s = exp(coeffs_s_C(2));
H_M0_s = R*coeffs_s_M0(1);
H_K_s = R*coeffs_s_K(1);
H_C_s = R*coeffs_s_C(1);

%Calculate Patameters for ferina
M0o_f = exp(coeffs_f_M0(2));
Ko_f = exp(coeffs_f_K(2));
Co_f = exp(coeffs_f_C(2));
H_M0_f = R*coeffs_f_M0(1);
H_K_f = R*coeffs_f_K(1);
H_C_f = R*coeffs_f_C(1);

%Define the temperature vector to plot against
T0 = 0;
Tf = 100;
temp_profile = zeros(1,Tf);

%populate the vector with the necessary temperatures in celsius
for i = 1:1:length(temp_profile)-1
    temp_profile(i+1) = temp_profile(i)+1; %Celcius
end

%Initialize the vectors for the GAB model constants as a function
%of temperature in our system.
M0_s_vect = zeros(1,length(temp_profile));
K_s_vect = zeros(1,length(temp_profile));
C_s_vect = zeros(1,length(temp_profile));
M0_f_vect = zeros(1,length(temp_profile));
K_f_vect = zeros(1,length(temp_profile));
```

```
C_f_vect = zeros(1,length(temp_profile));

%Iterate through the temperatures and calculate the GAB model equation
%constnats that are associated with each temp.

%Semoline m0
for i = 1:1:length(M0_s_vect);
    T = temp_profile(i);
    M0_s_vect(i) = para_calc(M0o_s,H_M0_s,T+273);
end

%Semolina K
for i = 1:1:length(K_s_vect);
    T = temp_profile(i);
    K_s_vect(i) = para_calc(Ko_s,H_K_s,T+273);
end

%Semolina C
for i = 1:1:length(C_s_vect);
    T = temp_profile(i);
    C_s_vect(i) = para_calc(Co_s,H_C_s,T+273);
end

%Farina M0
for i = 1:1:length(M0_f_vect);
    T = temp_profile(i);
    M0_f_vect(i) = para_calc(M0o_f,H_M0_f,T+273);
end

%Farina K
for i = 1:1:length(K_f_vect);
    T = temp_profile(i);
    K_f_vect(i) = para_calc(Ko_f,H_K_f,T+273);
end

%Farina C
for i = 1:1:length(C_f_vect);
    T = temp_profile(i);
    C_f_vect(i) = para_calc(Co_f,H_C_f,T+273);
end



%PLOT PARAMETERS V TEMP%
figure('NumberTitle', 'off', 'Name', 'GAB Model Constants v Temperature')
subplot(1,3,1)
hold on
plot(temp_profile,M0_s_vect,'-r');
plot(temp_profile,M0_f_vect,'-b');
%format
title('Monolayer, M0')
xlabel('Temperature [C]');
ylabel('M0 [ ]');
legend('Semolina','Farina');

subplot(1,3,2)
hold on
plot(temp_profile,K_s_vect,'-r');
plot(temp_profile,K_f_vect,'-b');
%format
title('K Constant')
xlabel('Temperature [C]');
ylabel('K [ ]');
legend('Semolina','Farina');

subplot(1,3,3)
hold on
```

```
plot(temp_profile,C_s_vect,'r-');
plot(temp_profile,C_f_vect,'b-');
%format
title('C Constant')
xlabel('Temperature [C]');
ylabel('C [ ]');
legend('Semolina','Farina');

%initialize the water activity mesh
num_steps = 100;
aw_mesh = zeros(1,num_steps+1);
step_size = 1/num_steps;

%populate the mesh
for i = 1:1:length(aw_mesh)-1
   aw_mesh(i+1) = aw_mesh(i) + step_size;
end

%initialize the temperatures to be used
temp_mesh = [20 35 50];

X_s = zeros(length(temp_mesh),length(aw_mesh));

%CALCULATE THE MOISTURE CONTENT AS A FUNCTION OF EATER ACTIVITY USING THE
%GAB MODEL

%Semoline, 20 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 1;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_s,H_M0_s,temp);
    K = para_calc(Ko_s,H_K_s,temp);
    C = para_calc(Co_s,H_C_s,temp);
    aw = aw_mesh(i);
    X_s(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%Semolina, 35 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 2;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_s,H_M0_s,temp);
    K = para_calc(Ko_s,H_K_s,temp);
    C = para_calc(Co_s,H_C_s,temp);
    aw = aw_mesh(i);
    X_s(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%Semoline, 50 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 3;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_s,H_M0_s,temp);
    K = para_calc(Ko_s,H_K_s,temp);
    C = para_calc(Co_s,H_C_s,temp);
    aw = aw_mesh(i);
    X_s(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%intialize the farina moisture content vector
X_f = zeros(length(temp_mesh),length(aw_mesh));
```

```
%FARINA, 20 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 1;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_f,H_M0_f,temp);
    K = para_calc(Ko_f,H_K_f,temp);
    C = para_calc(Co_f,H_C_f,temp);
    aw = aw_mesh(i);
    X_f(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%FARINA, 35 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 2;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_f,H_M0_f,temp);
    K = para_calc(Ko_f,H_K_f,temp);
    C = para_calc(Co_f,H_C_f,temp);
    aw = aw_mesh(i);
    X_f(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%FARINA, 50 C GAB MODEL
for i = 1:1:length(aw_mesh)

    temp_ind = 3;
    temp = temp_mesh(temp_ind) + 273;
    M0 = para_calc(M0o_f,H_M0_f,temp);
    K = para_calc(Ko_f,H_K_f,temp);
    C = para_calc(Co_f,H_C_f,temp);
    aw = aw_mesh(i);
    X_f(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

% PLOT THE DATA THAT WAS JUST CALCUALTED. MOISTURE CONTENT AS A FUNCTION OF
% WATER ACTIVITY
figure('NumberTitle', 'off', 'Name', 'Mositure Isotherms for Product (FITTED DATA)')
subplot(2,1,1)
hold on
plot(aw_mesh,X_s(1,:));
plot(aw_mesh,X_s(2,:));
plot(aw_mesh,X_s(3,:));
%plot(aw_mesh,X_s(4,:));
legend('20 C', '35 C', '50 C','location','northwest');
title('Moisture Content v Water Activity (Semolina)');
xlabel('Water Activity, aw')
ylabel('Moisture Content');

subplot(2,1,2)
hold on
plot(aw_mesh,X_f(1,:));
plot(aw_mesh,X_f(2,:));
plot(aw_mesh,X_f(3,:));
%plot(aw_mesh,X_f(4,:));
legend('20 C', '35 C', '50 C','location','northwest');
title('Moisture Content v Water Activity (Ferina)');
xlabel('Water Activity, aw')
ylabel('Moisture Content');

%Define the temperature mesh.
temp_mesh = [20 35 50 60];
```

```
X_s_emp = zeros(length(temp_mesh),length(aw_mesh));

%CALCUALTE THE GAB MODEL EQUATION AGAIN BUT USING THE EMPIRCALLY DEFINED
%CONSTANTS
for i = 1:1:length(aw_mesh)

    temp_ind = 1;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 11.8;
    K = 0.65;
    C = 4.21;
    aw = aw_mesh(i);
    X_s_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

for i = 1:1:length(aw_mesh)

    temp_ind = 2;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 6.45;
    K = 0.71;
    C = 8.77;
    aw = aw_mesh(i);
    X_s_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end
for i = 1:1:length(aw_mesh)

    temp_ind = 3;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 5.92;
    K = 0.72;
    C = 10.01;
    aw = aw_mesh(i);
    X_s_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end


X_f_emp = zeros(length(temp_mesh),length(aw_mesh));

for i = 1:1:length(aw_mesh)

    temp_ind = 1;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 9.16;
    K = 0.65;
    C = 14.06;
    aw = aw_mesh(i);
    X_f_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

for i = 1:1:length(aw_mesh)

    temp_ind = 2;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 6.01;
    K = 0.69;
    C = 31.95;
    aw = aw_mesh(i);
    X_f_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end
```

```
for i = 1:1:length(aw_mesh)

    temp_ind = 3;
    temp = temp_mesh(temp_ind) + 273;
    M0 = 5.23;
    K = 0.71;
    C = 37.04;
    aw = aw_mesh(i);
    X_f_emp(temp_ind,i) = (M0*K*C*aw)/((1-K*aw)*(1+(C-1)*K*aw));

end

%PLOT THE GAB MODEL EQUATION DATA AS CALCULATED USING THE EXPERIMETNALLY
%DETERMINED DATA
figure('NumberTitle', 'off', 'Name', 'Mositure Isotherms for Product (ACTUAL DATA)')
subplot(2,1,1)
hold on
plot(aw_mesh,X_s_emp(1,:));
plot(aw_mesh,X_s_emp(2,:));
plot(aw_mesh,X_s_emp(3,:));
%plot(aw_mesh,X_s_emp(4,:));
legend('20 C', '35 C', '50 C','location','northwest');
title('Moisture Content v Water Activity (Semolina)');
xlabel('Water Activity, aw')
ylabel('Moisture Content');

subplot(2,1,2)
hold on
plot(aw_mesh,X_f_emp(1,:));
plot(aw_mesh,X_f_emp(2,:));
plot(aw_mesh,X_f_emp(3,:));
%plot(aw_mesh,X_f_emp(4,:));
legend('20 C', '35 C', '50 C','location','northwest');
title('Moisture Content v Water Activity (Ferina)');
xlabel('Water Activity, aw')
ylabel('Moisture Content');

%Eb calculation for Semolina
moist_vect= linspace(5.5,30,50);
Eb_vect_S = zeros(1,length(moist_vect));
Eb_vect_F = zeros(1,length(moist_vect));

%Define 3 temperatures
T1 = 20 + 273;
T2 = 35 + 273;
T3 = 50 + 273;

%CALCULATE THE BINDING ENERGY FOR THE SEMOLINA
for i = 1:1:length(moist_vect)
    M = moist_vect(i);
    aw1 = aw_calc(M,para_calc(M0o_s,H_M0_s,T1),para_calc(Co_s,H_C_s,T1),para_calc(Ko_s,H_K_s,T1));
    aw2 = aw_calc(M,para_calc(M0o_s,H_M0_s,T2),para_calc(Co_s,H_C_s,T2),para_calc(Ko_s,H_K_s,T2));
    aw3 = aw_calc(M,para_calc(M0o_s,H_M0_s,T3),para_calc(Co_s,H_C_s,T3),para_calc(Ko_s,H_K_s,T3));
    Eb1 = (log(aw2/aw1)*R)/(1/T1 - 1/T2);
    Eb2 = (log(aw3/aw1)*R)/(1/T1 - 1/T3);
    Eb3 = (log(aw3/aw2)*R)/(1/T2 - 1/T3);
    Eb_avg = (Eb1 + Eb2 + Eb3)/3;
    Eb_vect_S(i) = Eb_avg;
end

%CALCULATE THE BINDING ENERGY FOR THE FARINA
for i = 1:1:length(moist_vect)
    M = moist_vect(i);
    aw1 = aw_calc(M,para_calc(M0o_f,H_M0_f,T1),para_calc(Co_f,H_C_f,T1),para_calc(Ko_f,H_K_f,T1));
    aw2 = aw_calc(M,para_calc(M0o_f,H_M0_f,T2),para_calc(Co_f,H_C_f,T2),para_calc(Ko_f,H_K_f,T2));
    aw3 = aw_calc(M,para_calc(M0o_f,H_M0_f,T3),para_calc(Co_f,H_C_f,T3),para_calc(Ko_f,H_K_f,T3));
    Eb1 = (log(aw2/aw1)*R)/(1/T1 - 1/T2);
    Eb2 = (log(aw3/aw1)*R)/(1/T1 - 1/T3);
```

```
    Eb3 = (log(aw3/aw2)*R)/(1/T2 - 1/T3);
    Eb_avg = (Eb1 + Eb2 + Eb3)/3;
    Eb_vect_f(i) = Eb_avg;
end


%PLOT THE BINDING ENERGY DATA AS A FUNCTION OF MOISTURE CONTENT
figure('NumberTitle', 'off', 'Name', 'Binding Energy v Moisture Content')
hold on
plot(moist_vect,Eb_vect_S,'r-');
plot(moist_vect,Eb_vect_f,'b-');
title('Binding Energy, J/kg');
xlabel('Moisture Content [g/g]')
ylabel('Binding Energy [J/g]')
legend('Semolina','Farina');

%CALCULATE THE EFFECTIVE DIFFUSIVITY USING THE FOLLOWING EQUATION FOR SEMOLINA.:
%Deff = Do*exp(-Ea/RT)*(K*exp(-Eb/RT)/(1 + K*exp(-Eb/RT))
%Define constants
K = 1032.6;
Ea = 5.2*4.184;
Do = 7e-8;
Deff_vect_S = zeros(length(T_C),length(Eb_vect_S));

%loop through each moiture content and temperature to calcualte the
%diffusivity,.
for j = 1:1:length(T_K)
    T = T_K(j);
    for i = 1:1:length(Deff_vect_S(1,:))
       Eb = Eb_vect_S(i);
       Deff_vect_S(j,i) = Do*exp(-1*Ea/(R*T))*(K*exp(-1*Eb/(R*T))/(1 + K*exp(-1*Eb/(R*T))));

    end
end

%CALCULATE THE EFFECTIVE DIFFUSIVITY USING THE FOLLOWING EQUATION FOR FARINA.:
%Deff = Do*exp(-Ea/RT)*(K*exp(-Eb/RT)/(1 + K*exp(-Eb/RT))
%Define constants
K = 1032.6;
Ea = 5.2*4.184;
Do = 7e-8;
Deff_vect_f = zeros(length(T_C),length(Eb_vect_S));

for j = 1:1:length(T_K)
    T = T_K(j);
    for i = 1:1:length(Deff_vect_f(1,:))
       Eb = Eb_vect_f(i);
       Deff_vect_f(j,i) = Do*exp(-1*Ea/(R*T))*(K*exp(-1*Eb/(R*T))/(1 + K*exp(-1*Eb/(R*T))));

    end
end

%PLOT THE DATA
figure('NumberTitle', 'off', 'Name', 'Diffusion Coefficient (Semolina)')
hold on
plot(moist_vect,Deff_vect_S(1,:),'-k');
plot(moist_vect,Deff_vect_S(2,:),'-r');
plot(moist_vect,Deff_vect_S(3,:),'-g');
plot(moist_vect,Deff_vect_S(4,:));
title('Diffusion Coefficient (Semolina)');
xlabel('Moisture Content')
ylabel('Diffusion Coefficient')
legend('20 C', '35 C', '50 C', '60 C', 'location', 'southeast');

figure('NumberTitle', 'off', 'Name', 'Diffusion Coefficient (Farina)')
hold on
plot(moist_vect,Deff_vect_f(1,:),'-k');
plot(moist_vect,Deff_vect_f(2,:),'-r');
```

```
plot(moist_vect,Deff_vect_f(3,:),'-g');
plot(moist_vect,Deff_vect_f(4,:));
title('Diffusion Coefficient (Farina)');
xlabel('Moisture Content')
ylabel('Diffusion Coefficient')
legend('20 C', '35 C', '50 C', '60 C', 'location', 'southeast');

function const = para_calc(const0,H,T);
    R = 8.324; %J/mol-K
    const = const0*exp(H/(R*T));
end

function aw = aw_calc(M,M0,C,K);

    A = (M0/M) - 1;
    aw = (2 + A*C - ((2 + A*C)^2 - 4*(1-C))^0.5)/(2*K*(1-C));

end
```

Code Part B

```
clc
clear all
close all

Tg_soy = 410; %Kelvin
Tg_w = 134; %Kelvin

moist_vect_wet = linspace(0.1,0.6,100);
moist_vect_dry = zeros(1,length(moist_vect_wet));

for i = 1:1:length(moist_vect_wet)

    moist_vect_dry(i) = moist_vect_wet(i)/(1 - moist_vect_wet(i));

end

Tg_vect = zeros(1,length(moist_vect_wet));

for i = 1:1:length(moist_vect_wet)
    M = moist_vect_wet(i);
    Tg_vect(i) = 1/(M/Tg_w + (1-M)/Tg_soy);
end

Tg_vect = Tg_vect - 273; % Convert to celcius

T_dry_vect = ones(1,length(moist_vect_wet));
T_dry_vect = T_dry_vect.*Tg_vect(length(Tg_vect)) + 50;
drying_temps_s = [];
drying_temps_s(1,1) = Tg_vect(length(Tg_vect)) + 50;
drying_temps_s(2,1) = moist_vect_wet(length(moist_vect_wet));

for i = length(moist_vect_wet):-1:2;

    T_min = Tg_vect(i) + 10;
    T_dry = T_dry_vect(i);

    if (T_dry <= T_min)
        T_dry = Tg_vect(i) + 50;
        drying_temps_s(1,length(drying_temps_s(1,:))+1) = T_dry;
        drying_temps_s(2,length(drying_temps_s(2,:))) = moist_vect_wet(i-1);
    end

    T_dry_vect(i-1) = T_dry;
end

%disp(T_dry_vect)

figure(1)
hold on
plot(moist_vect_wet,Tg_vect + 10,'b--');
plot(moist_vect_wet,Tg_vect + 50,'k--');
plot(moist_vect_wet,Tg_vect,'g-');
plot(moist_vect_wet,T_dry_vect,'r');
plot(drying_temps_s(2,:),drying_temps_s(1,:),'ob');
xlabel('Moisture Content (Dry Basis)');
ylabel('Tg [C]');
title('Glass Transition Temp. v Moisture Content');
legend('Minimum temperature','Maximum Temperature','Tg [C]','Drying Temperature');

%Define constants
H_M0_s = 2.195e4;
H_C_s =-6.539e4;
H_K_s =-2.899e3;
H_M0_f = 1.295e4;
```

```
H_C_f =-4.030e4;
H_K_f = -2.280e3;
M0o_s = 0.0014;
Co_s = 1.299e12;
Ko_s = 2.156;
M0o_f = 0.0424;
Co_f = 2.011e8;
Ko_f = 1.665;

fprintf('A %d stage dryer using the following conditions: \n',length(drying_temps_s(1,:)));
fprintf('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-\n');

for i = 1:1:length(drying_temps_s(1,:))
    M = drying_temps_s(2,i);
    temp = drying_temps_s(1,i) + 273;

    M0 = para_calc(M0o_s,H_M0_s,temp);
    C = para_calc(Co_s,H_C_s,temp);
    K = para_calc(Ko_s,H_K_s,temp);

    RH = aw_calc(M,M0,C,K)*100;

    fprintf('Stage %d: Air at %0.2f C and %0.2f%% RH \n',i,temp-273,RH);

end

fprintf('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-\n');

function const = para_calc(const0,H,T);
    R = 8.324; %J/mol-K
    const = const0*exp(H/(R*T));
end

function aw = aw_calc(M,M0,C,K);

    A = (M0/M) - 1;
    aw = (2 + A*C - ((2 + A*C)^2 - 4*(1-C))^0.5)/(2*K*(1-C));

end
```