Prediction of Couscous Moisture Diffusivity and Dryer Operating Conditions

Kathryn Atherton

ABE 557

October 25, 2018

```
clear;
clc;
close all;
```

Problem:

Develop a working algorithm (detailed calculation process) using Math Lab to determine the diffusivity of moisture of a couscous product as a function of temperature, moisture, and porosity based on equilibrium moisture.

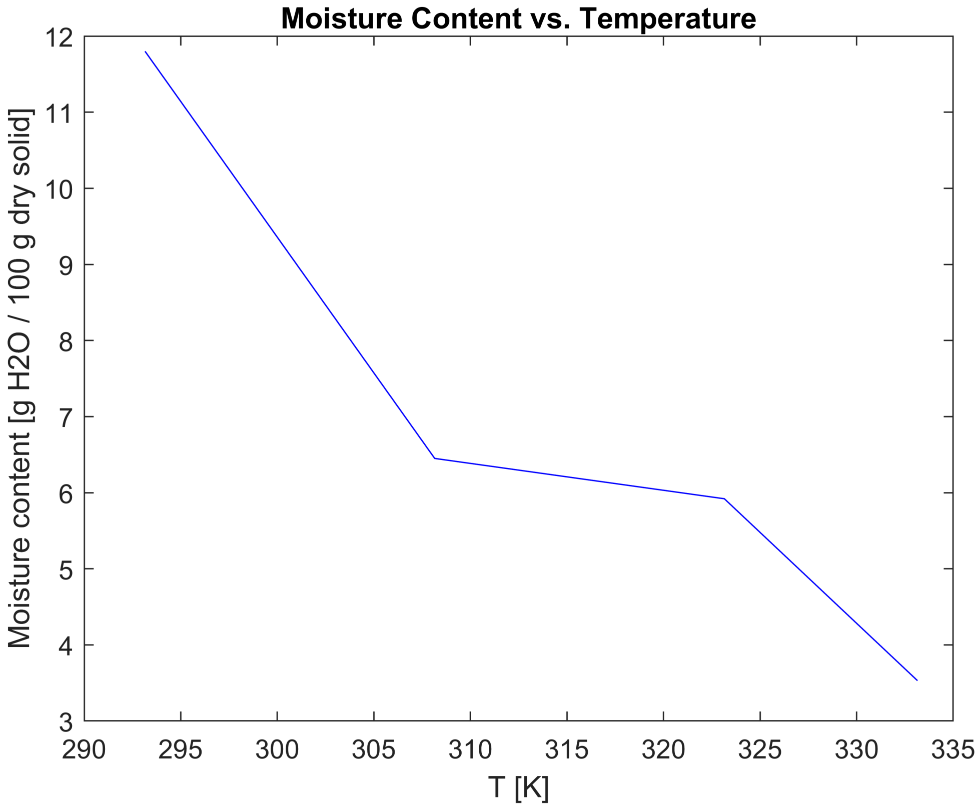
GAB model coefficients for couscous flour at various temperatures:

```
Temp = [20, 35, 50, 60]+273.15;
Mo = [11.8, 6.45, 5.92, 3.53];
K = [0.65, 0.71, 0.72, 0.76];
C = [4.21, 8.77, 10.01];
```

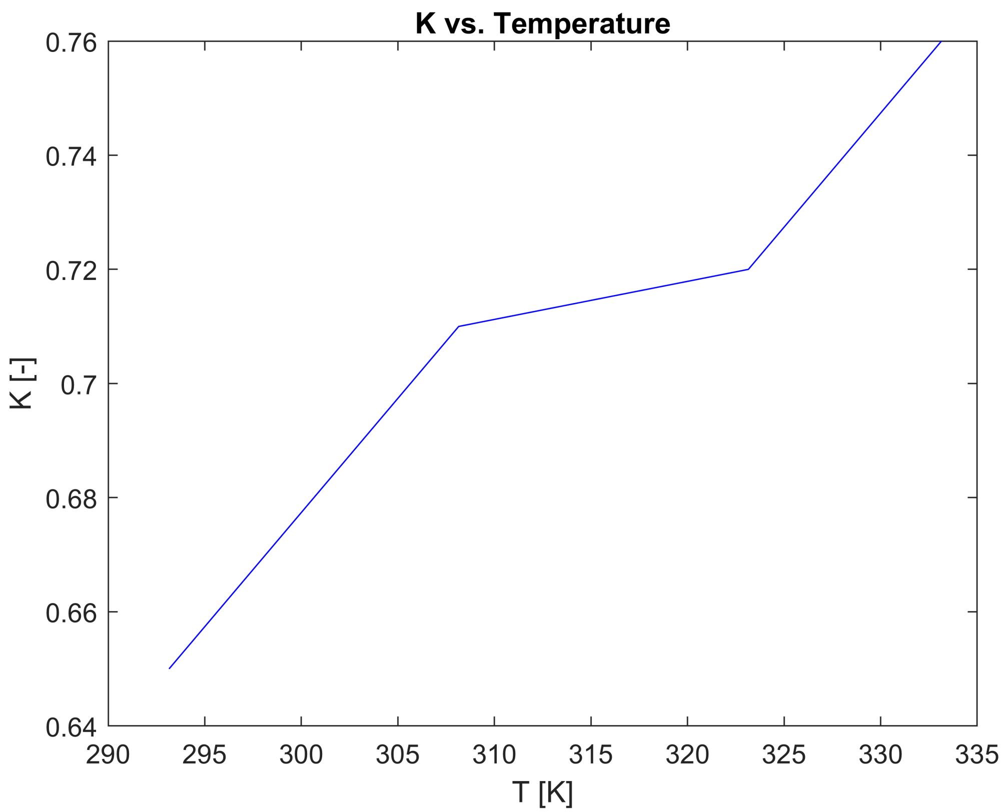
Part A: Determine the relation for moisture diffusion as a function of temperature and moisture.

Develop a GAB equation with Xm, Cg, and K as a function of temperature for your product. Plot Xm, Cg and K vs temperature. Provide a plot of X vs aw at various temperatures for your product and compare with data given above.

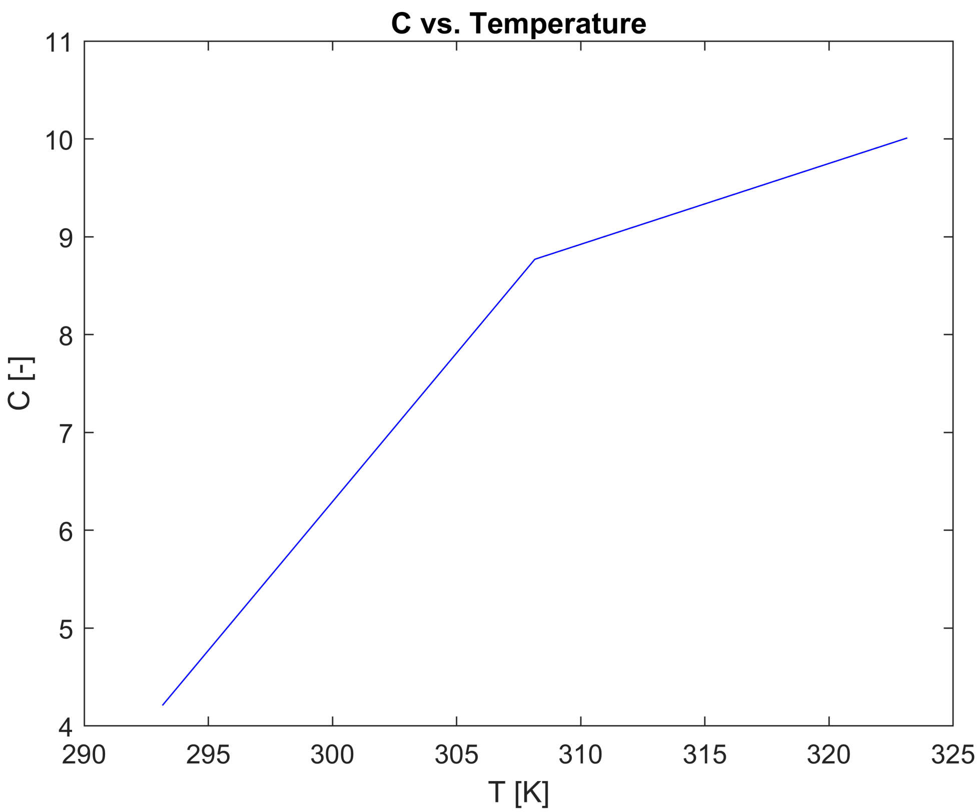
```
figure
plot(Temp, Mo);
title('Moisture Content vs. Temperature')
xlabel('T [K]')
ylabel('Moisture content [g H2O / 100 g dry solid]')
```




```
figure
plot(Temp, K);
title('K vs. Temperature')
xlabel('T [K]')
ylabel('K [-]')
```



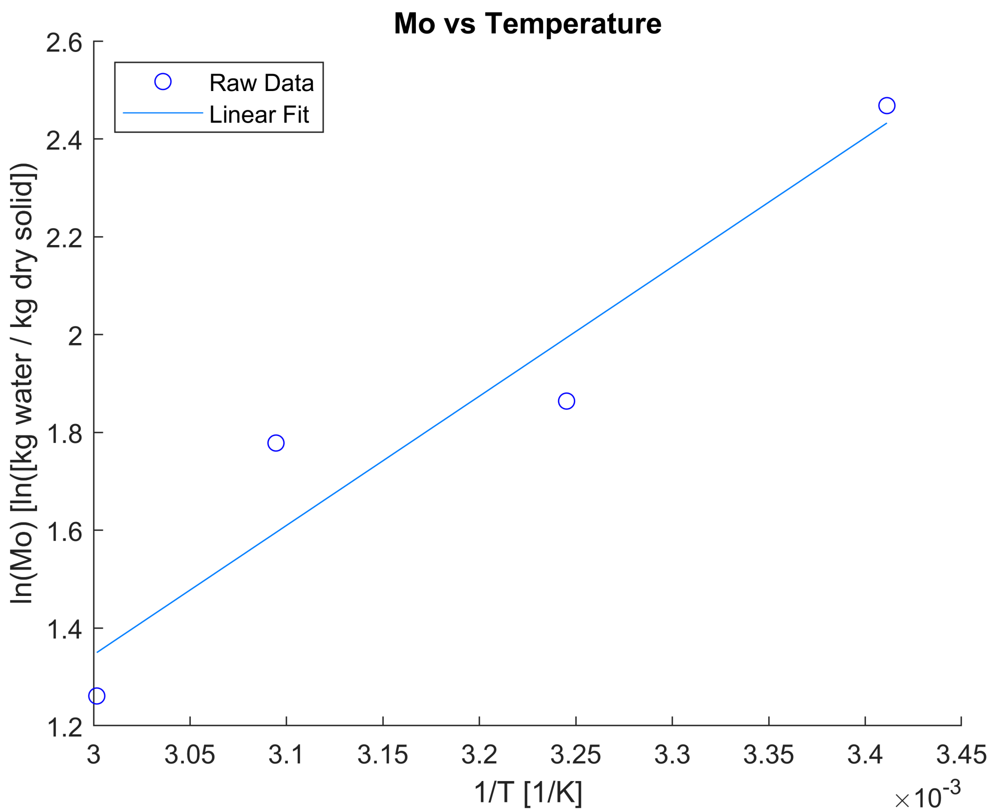

```
figure
plot(Temp(:,1:3), C);
title('C vs. Temperature')
xlabel('T [K]')
ylabel('C [-]')
```



Equation 1: $m = m_0 e^{\frac{-\Delta H_m}{\mathrm{RT}}}$

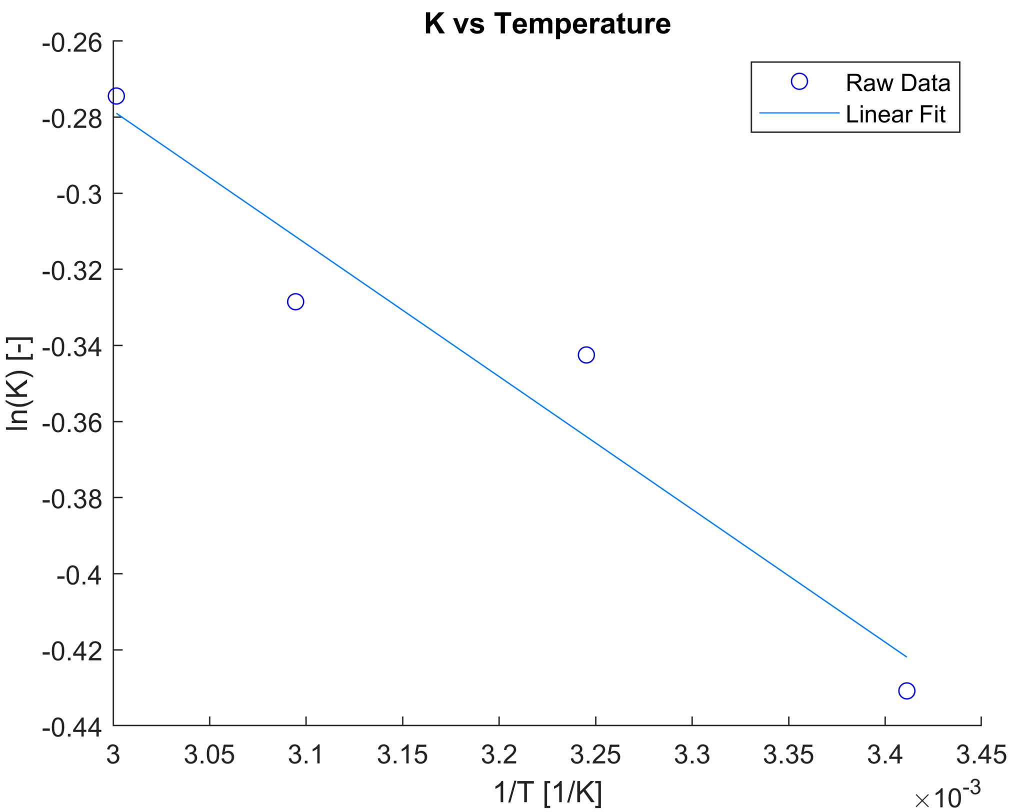
```
R = 8.314; %universal gas constant J/mol*K

figure()
hold on
scatter(1./Temp, log(Mo),'DisplayName','Raw Data');
X = [1./Temp;ones(1,length(1./Temp))];
linear_coef_mo = polyfit((1./Temp), (log(Mo)), 1);
plot(1./Temp, linear_coef_mo(2) + linear_coef_mo(1).*(1./Temp),'DisplayName','Linear Fit')
legend('show', 'Location', 'northwest');
title('Mo vs Temperature')
xlabel('1/T [1/K]');
ylabel('ln(Mo) [ln([kg water / kg dry solid])');
```



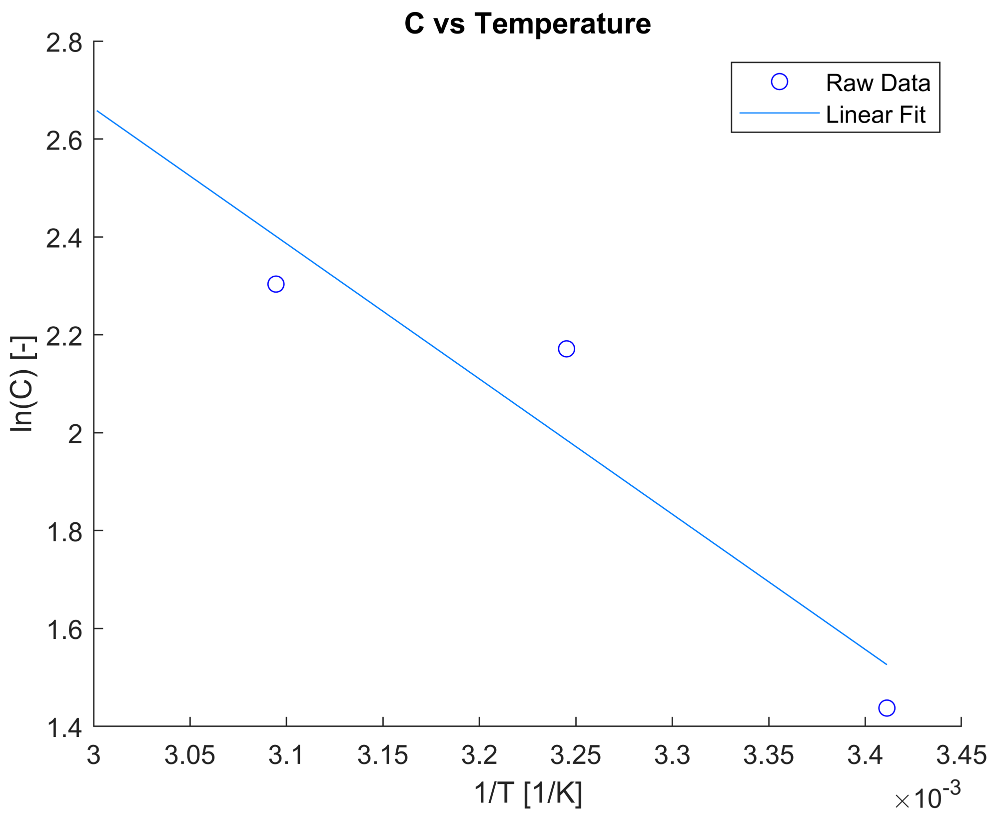
Equation 2: $k = k_0 e^{\frac{-\Delta H_k}{\mathrm{RT}}}$

```
figure()
hold on
scatter(1./Temp, log(K),'DisplayName','Raw Data');
X = [1./Temp;ones(1,length(1./Temp))];
linear_coef_K = polyfit((1./Temp), (log(K)),1);
plot(1./Temp, linear_coef_K(2) + linear_coef_K(1).*(1./Temp),'DisplayName','Linear Fit')
legend('show');
title('K vs Temperature')
xlabel('1/T [1/K]');
ylabel('ln(K) [-]');
```



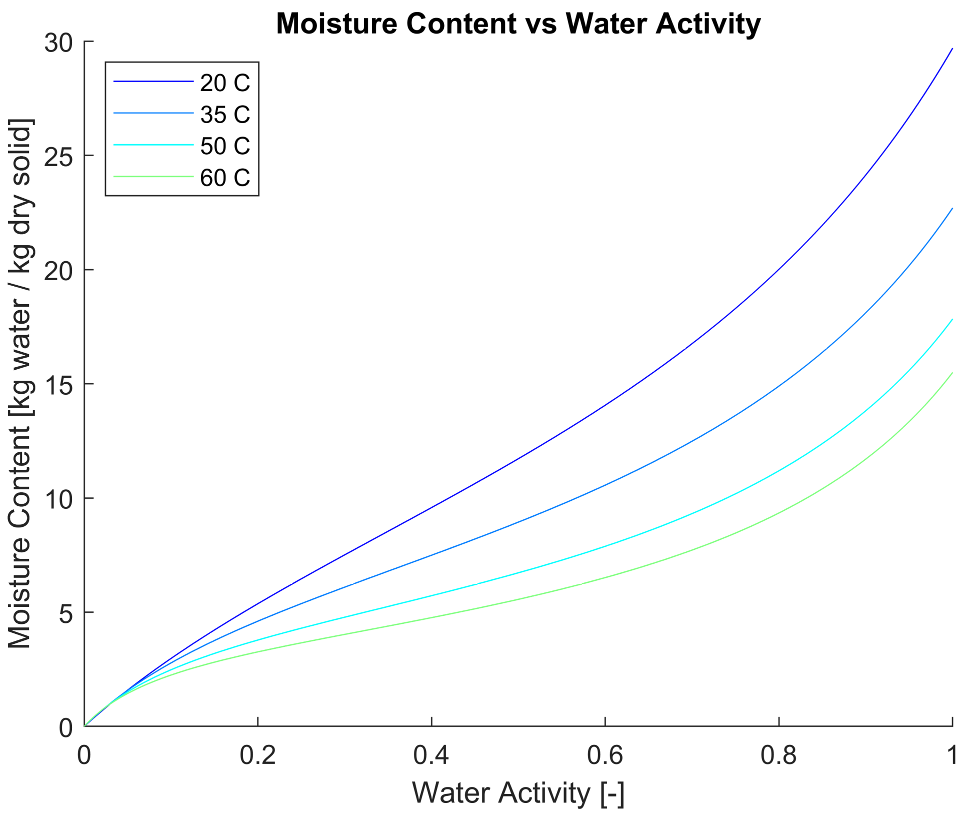
Equation 3: $c = c_0 e^{\frac{-\Delta H_c}{\mathrm{RT}}}$

```
figure()
hold on
scatter(1./Temp(1:3), log(C),'DisplayName','Raw Data');
X = [1./Temp;ones(1,length(1./Temp))];
linear_coef_C = polyfit((1./Temp(1:3)), (log(C)),1);
plot(1./Temp, linear_coef_C(2) + linear_coef_C(1).*(1./Temp),'DisplayName','Linear Fit')
legend('show');
title('C vs Temperature')
xlabel('1/T [1/K]');
ylabel('ln(C) [-]');
```



Equation 4: $\frac{m}{m_0} = \frac{c \cdot k \cdot \alpha_w}{(1 - k \cdot \alpha_w)(1 + (c-1)k \cdot \alpha_w)}$

```
step = 0.00001;
aw = 0:step:1;
M_major = [];
figure()
hold on
for i = 1:length(Temp)
    Mo_new = exp(linear_coef_mo(2))*exp(linear_coef_mo(1)/Temp(i));
    C_new = exp(linear_coef_C(2))*exp(linear_coef_C(1)/Temp(i));
    K_new = exp(linear_coef_K(2))*exp(linear_coef_K(1)/Temp(i));
    for j = 1:length(aw)
        M(j) = (Mo_new*C_new*K_new*aw(j))/((1-K_new*aw(j))*(1+(C_new-1)*K_new*aw(j)));
        M_major(i,j) = M(j);
    end
    plot(aw,M)
end
title('Moisture Content vs Water Activity')
xlabel('Water Activity [-]');
ylabel('Moisture Content [kg water / kg dry solid]');
legend('20 C', '35 C', '50 C', '60 C', 'Location', 'northwest')
```

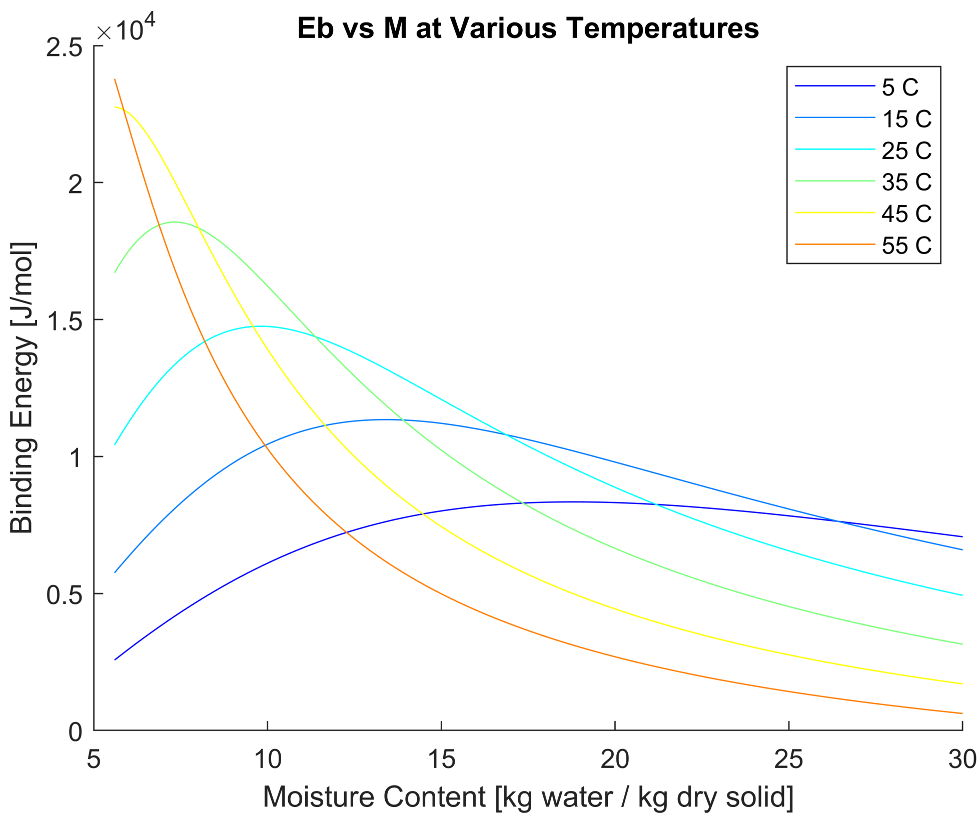


Determine Eb as a function of moisture content and temperature. Plot Eb vs moisture at various temperatures.

Equation 5: $E_b = \ln\left(\frac{\alpha_{w_2}}{\alpha_{w_1}}\right)\frac{R}{\left(\frac{1}{T_1}-\frac{1}{T_2}\right)_M}$

```
Temperature = (5+273.15):(56+273.15);
M_test = 5.6:0.1:30;
aw_isotherm = [];
Eb_major = [];
for i = 1:length(M_test)
    for j = 1:length(Temperature)
        Mo_temp = exp(linear_coef_mo(2))*exp(linear_coef_mo(1)/Temperature(j));
        C_temp = exp(linear_coef_C(2))*exp(linear_coef_C(1)/Temperature(j));
        K_temp = exp(linear_coef_K(2))*exp(linear_coef_K(1)/Temperature(j));
        aw_isotherm(i,j) = (2 + ((Mo_temp/M_test(i))-1)*C_temp - ((2+(Mo_temp/M_test(i)-1)*C_te
    end
end
for i = 1:length(M_test)
    for j = 1:length(Temperature)-1
        Eb(j) = log(aw_isotherm(i,j+1)/aw_isotherm(i,j))*R/((1/Temperature(j)) - (1/Temperature
        Eb_major(i,j) = Eb(j);
    end
end
```

```
figure()
hold on
plot(M_test,Eb_major(:,1))
plot(M_test,Eb_major(:,11))
plot(M_test,Eb_major(:,21))
plot(M_test,Eb_major(:,31))
plot(M_test,Eb_major(:,41))
plot(M_test,Eb_major(:,51))
title('Eb vs M at Various Temperatures')
xlabel('Moisture Content [kg water / kg dry solid]');
ylabel('Binding Energy [J/mol]')
legend('5 C','15 C','25 C','35 C','45 C', '55 C')
```



Determine Deff from Eq 48 of the Food Dehydration chapter (10). Calculate Ea, Do (liquid water), Do (vapor water), from Diffusivity data at different temperatures given Geankoplis or other literature. Use K as given in Chp 10 p 661. Compare calculated Deff with Deff values found in literature. Plot Deff vs. moisture at various temperatures and plot Deff vs. moisture at various porosities. Compare with data given in Chp 10 Table 10.7.

Equation 6: $D_{\mathrm{AVeff}} = \frac{\varepsilon}{\tau} D_{\mathrm{AV}} = \frac{\varepsilon}{\tau}\left(D_{\mathrm{AVD}} e^{\frac{-E_c}{RT}}\right) = \frac{\varepsilon}{\tau}\left(D_{\mathrm{AVO}} e^{\frac{-E_A}{RT}}\right) \frac{K e^{\frac{-E_b}{RT}}}{1 + K e^{\frac{-E_b}{RT}}}$

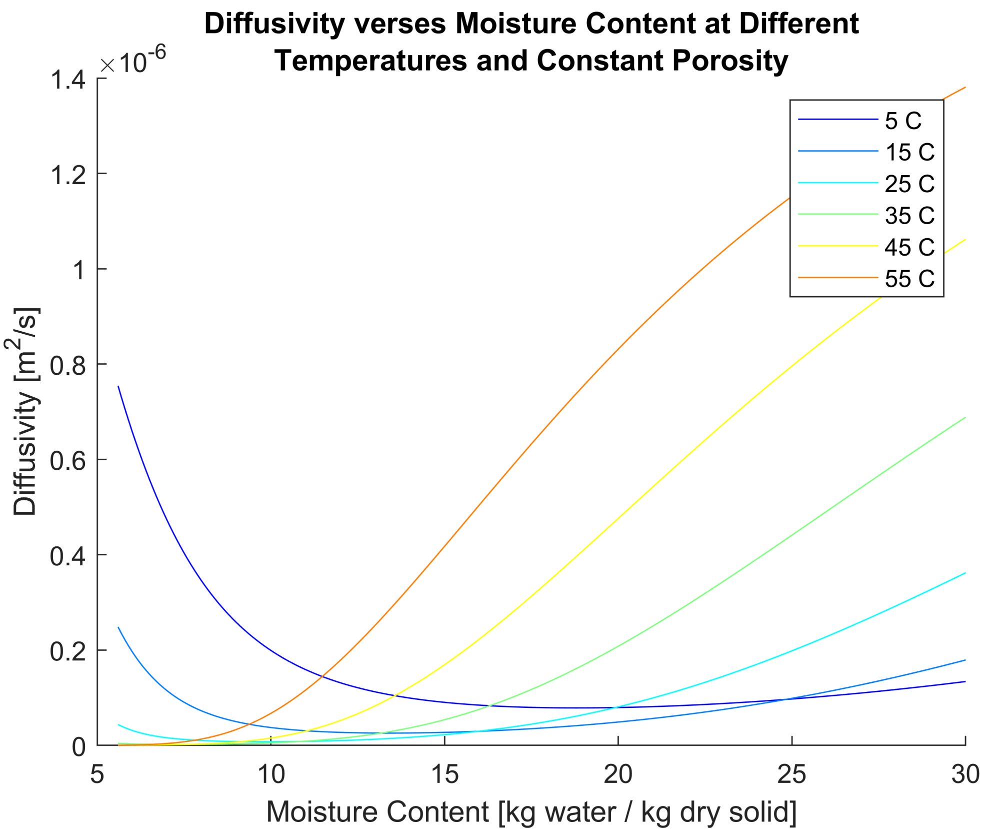
Equation 7: $D_{\text{ALeff}} = \frac{1-\varepsilon}{\tau} D_{\text{AL}} = \frac{1-\varepsilon}{\tau}\left(D_{\text{ALD}} e^{\frac{-E_A}{RT}}\right) = \frac{1-\varepsilon}{\tau}\left(D_{\text{ALO}} e^{\frac{-E_A}{RT}}\right) \frac{K e^{\frac{-E_b}{RT}}}{1 + K e^{\frac{-E_b}{RT}}}$

Equation 8: $= D_{\text{eff,series}} = \frac{\varepsilon}{\tau} D_V + \frac{1-\varepsilon}{\tau} D_L$

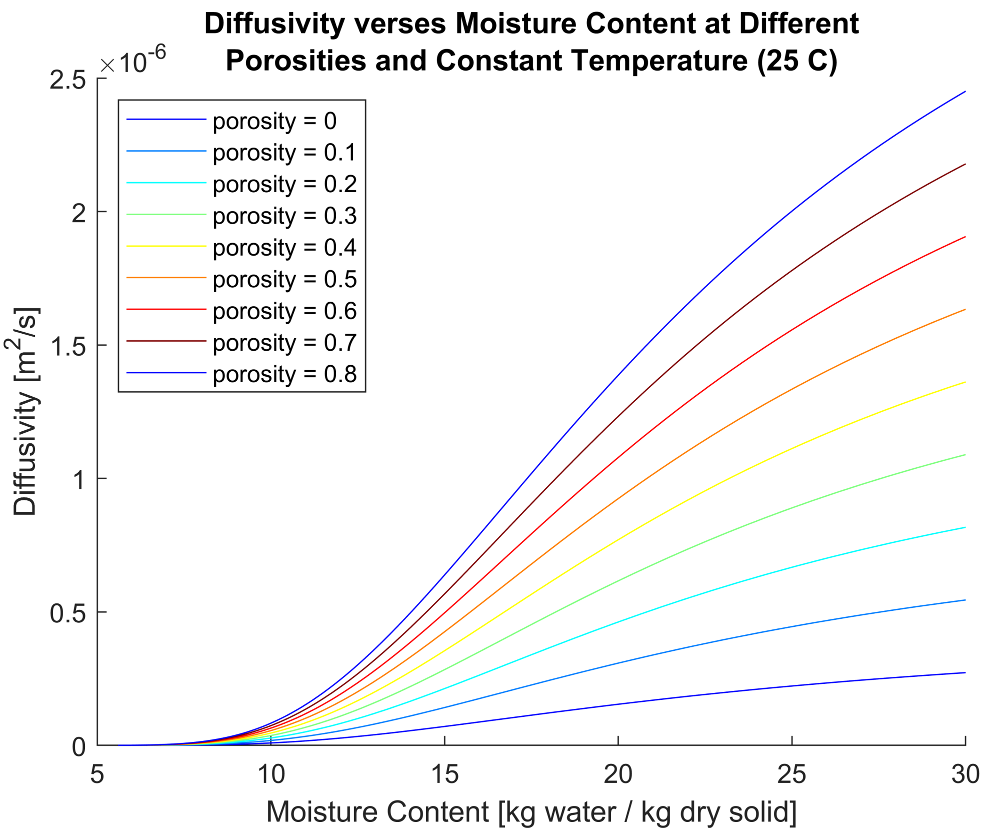
Equation 9: $\frac{1}{D_{\text{eff,perpendicular}}} = \frac{1}{\frac{\varepsilon}{\tau} D_V} + \frac{1}{\frac{1-\varepsilon}{\tau} D_L}$

Equation 10: $D_{\text{eff}} = \varphi D_{\text{eff,series}} + (1-\varphi) D_{\text{eff,perpendicular}}$

```
porosity = 0.5;
tau = 1.5;
phi = 0.5;
Davo = 2*10^-5;
Dalo = 1*10^-9;
Ea = 5.2*4.184;
K = 0.9;
for i = 1:length(M_test)
    for j = 1:length(Temperature)-1
        DAVeff(i,j) = Davo*(exp(-Ea/(R*Temperature(j))))*(K*(exp(-Eb_major(i,j)/(R*Temperature(
        DALeff(i,j) = (Dalo*exp(-Ea/(R*Temperature(j))))*(K*exp(-Eb_major(i,j)/(R*Temperature(j
        Dserieseff(i,j) = (porosity/tau)*DAVeff(i,j) + ((1-porosity)/tau)*DALeff(i,j);
        Dperpeff(i,j) = ((1/((porosity/tau)*DAVeff(i,j))) + (1/(((1-porosity)/tau)*DALeff(i,j))
        Deff(i,j) = phi*Dserieseff(i,j) + (1-phi)*Dperpeff(i,j);
    end
end
figure()
hold on
plot(M_test,Deff(:,1),M_test,Deff(:,11),M_test,Deff(:,21),M_test,Deff(:,31),M_test,Deff(:,41),M
title({'Diffusivity verses Moisture Content at Different', 'Temperatures and Constant Porosity
xlabel('Moisture Content [kg water / kg dry solid]')
ylabel('Diffusivity [m^2/s]')
legend('5 C','15 C','25 C','35 C','45 C', '55 C')
```




```
Temp_constant = 25+273.15;
Eb_at_temp_const = Eb_major(:,51).';
porosities = 0.1:0.1:0.9;
for i = 1:length(M_test)
    for j = 1:length(porosities)
        p_DAVeff(i,j) = Davo*exp(-Ea/(R*Temp_constant))*((K*exp(-Eb_at_temp_const(i)/(R*Temp_co
        p_DALeff(i,j) = (Dalo*exp(-Ea/(R*Temp_constant)))*((K*exp(-Eb_at_temp_const(i)/(R*Temp_
        p_Dserieseff(i,j) = (porosities(j)/tau)*p_DAVeff(i,j) + ((1-porosities(j))/tau)*p_DALe
        p_Dperpeff(i,j) = ((1/((porosities(j)/tau)*p_DAVeff(i,j))) + (1/(((1-porosities(j))/tau
        p_Deff(i,j) = phi*p_Dserieseff(i,j) + (1-phi)*p_Dperpeff(i,j);
    end
end
figure()
hold on
plot(M_test,p_Deff(:,1),M_test,p_Deff(:,2),M_test,p_Deff(:,3),M_test,p_Deff(:,4),M_test,p_Deff(
%axis([5 15 0.5E-6 5E-6])
title({'Diffusivity verses Moisture Content at Different', 'Porosities and Constant Temperature
xlabel('Moisture Content [kg water / kg dry solid]')
ylabel('Diffusivity [m^2/s]')
legend('porosity = 0','porosity = 0.1','porosity = 0.2','porosity = 0.3','porosity = 0.4', 'por
```



Part B: Determine the dryer operating conditions (temperature and humidity) maximize shrinkage to dehydrate the soy product form 60% H2O wb to 10% H2O wb.

Plot Tg as a funciton of moisture ocntent using the Fox equation and determine the temperature and humidity for one stage of the dryer. Tg of the soy solid is 410K and Tg of water is 134 K.

Single-Stage Dryer:

```
Tg_solid = 410; %Kelvin
Tg_water = 134; %Kelvin
M_range = 0:0.001:0.9;
Tman = 150:500;
for i = 1:length(M_range)
    Tg_original(i) = ((M_range(i)/Tg_water) + ((1-M_range(i))/Tg_solid))^-1;
end
Tg_30 = Tg_original + 30;
Tg_10 = Tg_original + 10;
Mc_initial = 0.6;
Mc_desired = 0.1;
Mc_final = 0.6;
Tg_initial = Tg_original;
stages = [1 0.6;0 0];
i = 2;
```

```
temp_change = Tg_10(Mc_desired*1000+1) - Tg_10(Mc_initial*1000+1);
Tg_op = Tg_original + temp_change;
Operating_Temperature = Tg_op(Mc_desired*1000+1);
Mo_temp = exp(linear_coef_mo(2))*exp(linear_coef_mo(1)/Operating_Temperature);
C_temp = exp(linear_coef_C(2))*exp(linear_coef_C(1)/Operating_Temperature);
K_temp = exp(linear_coef_K(2))*exp(linear_coef_K(1)/Operating_Temperature);
RH = (2 + ((Mo_temp/10)-1)*C_temp - ((2+(Mo_temp/10-1)*C_temp)^2 - 4*(1-C_temp))^0.5)/(2*K_temp
RH = RH*100;
fprintf('For a one stage dryer, the operating temperature must be %.2f K and the RH must be %.2
```

```
For a one stage dryer, the operating temperature must be 496.57 K and the RH must be 90.98%.
```

Multi-Stage Dryer

```
while Mc_final > 0.1
    Op_Temp = Tg_30(Mc_final*1000+1);

    Mo_temp = exp(linear_coef_mo(2))*exp(linear_coef_mo(1)/Op_Temp);
    C_temp = exp(linear_coef_C(2))*exp(linear_coef_C(1)/Op_Temp);
    K_temp = exp(linear_coef_K(2))*exp(linear_coef_K(1)/Op_Temp);

    Mc_final = round(Tg_water*(Tg_solid - Tg_30(Mc_final*1000+1))/(Tg_30(Mc_final*1000+1)*(Tg_s
    stages(i,1) = i;
    stages(i,2) = Mc_final;

    RH_stage = (2 + ((Mo_temp/(Mc_final*100))-1)*C_temp - ((2+(Mo_temp/(Mc_final*100)-1)*C_temp
    RH_stage = RH_stage*100;

    stages(i-1,3) = RH_stage;
    i = i+1;
end
```

The drying process is to be designed to produce a dense product by maintaining the surface conditions at least 10C above the Tg.

```
op_temp_range = (round(Tg_original(Mc_initial*1000+1),0)+10)+1:0.1:(round(Tg_original(Mc_initia
mc_in = [];
mc_out = [];
final_mc_values = [0 0;0 0];
RH_values = [];
stage_op_temps = [];
for i = 1:length(op_temp_range)
    final_mc_values(i,1) = op_temp_range(i);
    Tg_op_temp = Tg_original + (op_temp_range(i)-Tg_original(Mc_initial*1000+1));
    Mc_final = Mc_initial;
    stages_temp = [1 0.6;0 0];
    j = 2;
    while Mc_final > 0.1
        mc_in(i,j) = Mc_final;
        Op_temp_index = Mc_final*1000+1;
```

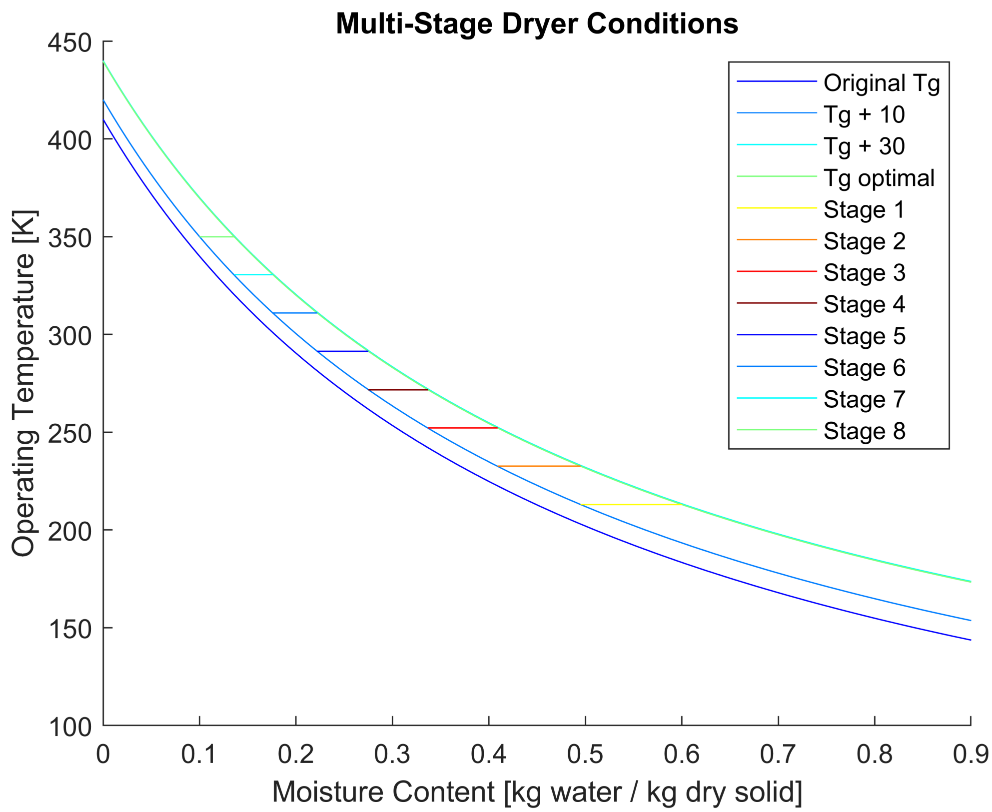
```
        Op_Temp = Tg_op_temp(Op_temp_index);
        stage_op_temps(i,j-1) = Op_Temp;
        Mo_temp = exp(linear_coef_mo(2))*exp(linear_coef_mo(1)/Op_Temp);
        C_temp = exp(linear_coef_C(2))*exp(linear_coef_C(1)/Op_Temp);
        K_temp = exp(linear_coef_K(2))*exp(linear_coef_K(1)/Op_Temp);
        Mc_final = round(Tg_water*(Tg_solid - Op_Temp + 10)/((Op_Temp-10)*(Tg_solid - Tg_water)
        stages_temp(j,1) = j;
        stages_temp(j,2) = Mc_final;
        mc_out(i,j) = Mc_final;
        RH_stage = (2 + ((Mo_temp/(Mc_final*100))-1)*C_temp - ((2+(Mo_temp/(Mc_final*100)-1)*C_
        RH_stage = RH_stage*100;
        RH_values(i,j) = round(RH_stage);
        stages_temp(j-1,3) = RH_stage;
        j = j+1;
    end
    final_mc_values(i,2) = stages_temp(length(stages_temp),2);
    final_mc_values(i,3) = j-2;
end
optimal_temp_values(1,:) = op_temp_range(find(final_mc_values(:,2)==0.1));
optimal_temp_values(2,:) = final_mc_values(find(final_mc_values(:,2) == 0.1),3);
optimal_temperature_index = find(op_temp_range == optimal_temp_values(1,length(optimal_temp_va
mc_in_final = mc_in(optimal_temperature_index,:);
mc_out_final = mc_out(optimal_temperature_index,:);
optimal_op_temp_values = stage_op_temps(optimal_temperature_index,:);
optimal_RH_values = RH_values(optimal_temperature_index,:);
fprintf('Multi-Stage Dryer Operating Conditions:\n');
```

```
Multi-Stage Dryer Operating Conditions:
```

```
for i = 1:optimal_temp_values(2,length(optimal_temp_values))
    fprintf('Stage %d Operating Temperature: %.2f K RH: %d percent MC_in = %.2f MC_out = %.2f\
end
```

```
Stage 1 Operating Temperature: 213.00 K RH: 99 percent MC_in = 0.60 MC_out = 0.49
Stage 2 Operating Temperature: 232.64 K RH: 83 percent MC_in = 0.49 MC_out = 0.41
Stage 3 Operating Temperature: 252.16 K RH: 75 percent MC_in = 0.41 MC_out = 0.34
Stage 4 Operating Temperature: 271.64 K RH: 76 percent MC_in = 0.34 MC_out = 0.28
Stage 5 Operating Temperature: 291.37 K RH: 84 percent MC_in = 0.28 MC_out = 0.22
Stage 6 Operating Temperature: 310.97 K RH: 91 percent MC_in = 0.22 MC_out = 0.18
Stage 7 Operating Temperature: 330.54 K RH: 94 percent MC_in = 0.18 MC_out = 0.14
Stage 8 Operating Temperature: 349.90 K RH: 93 percent MC_in = 0.14 MC_out = 0.10
```

```
figure()
hold on
plot(M_range,Tg_original,M_range,Tg_original+10,M_range, Tg_original+30,M_range, Tg_original+(
for i = 1:optimal_temp_values(2,length(optimal_temp_values))
    plot([mc_out_final(i+1),mc_in_final(i+1)],[optimal_op_temp_values(i),optimal_op_temp_value
        ])
end
title('Multi-Stage Dryer Conditions')
xlabel('Moisture Content [kg water / kg dry solid]')
ylabel('Operating Temperature [K]')
legend('Original Tg', 'Tg + 10', 'Tg + 30', 'Tg optimal', 'Stage 1', 'Stage 2', 'Stage 3', 'Sta
```


Multi-Stage Dryer Conditions
Original Tg
Tg + 10
Tg + 30
Tg optimal
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6
Stage 7
Stage 8
Operating Temperature [K]
Moisture Content [kg water / kg dry solid]
450
400
350
300
250
200
150
100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9