

Iran University of Science and Technology

The School of Chemical, Petroleum and Gas Engineering

Computer Project

Name:

Date

**I. Introduction**

In Vapor/Liquid Equilibrium (VLE), the temperatures, pressures, and compositions of the mixtures in different phases in equilibrium are calculated for by using models for the behaviour of such systems. You are supposed to develop a program to solve for VLE problems. For this particular program which should be written in MATLAB, the gamma/phi formulation of VLE will be used. The gamma/phi formulation of VLE (Eqn. 1) is an extension of the modified Raoult’s law wherein a fugacity coefficient is added to account for the nonidialities in the vapor phase. Using the modified Raoult’s law in moderate pressures would produce unreliable results.

yi φi P = xi γi Pisat (1)

Generally, dewpoint and bubblepoint calculations are used to solve for VLE problems. VLE problems usually involve solving for the bubble (vapor) or dew (liquid) composition, T, or P, given that any two of those parameters are provided, as required by the phase rule for VLE.

In your program, there should be five functions/subroutines namely:

1. *bubble\_P\_gammaphi*
2. *dew\_P\_gammaphi*
3. *bubble\_T\_gammaphi*
4. *dew\_T\_gammaphi*
5. *flashcalc\_gammaphi*

These functions require the calculation for the values of Bij, φi and γi. To make the program shorter, solutions for Bij, φi and γi were separated into different subroutines. The user must provide the following at the beginning of the program:

Tc : row vector containing critical temperature (K) of each species

w : row vector containing ω of each species

Zc : row vector containing compressibility factor of each species

Vc : row vector of the critical volume (cm3/mol) of each species

Pc : row vector of the critical pressure (bar) of each species

A,b,C : row vectors containing the Antoine constants of each species

Par : matrix containing the parameters of the functional groups in the mixture. It is important to note that the format is as follows:

[main\_grp\_no. subgroup\_no. Rk Qk Vk(1) Vk(2)...Vk(i)]

a\_mn : matrix containing all interaction parameters for all possible pairs of the

 functional groups in the mixture

The function *bubble\_P\_gammaphi* calculates for the composition of the mixture in the vapor phase and pressure in bar given that the molar composition of the mixture at the liquid phase is provided. Along with the necessary inputs above, the user should include:

 x : row vector containing liquid phase composition of the mixture

T : temperature in K

The output of this function is a row vector (y) containing the molar composition of the
mixture at vapor phase and the bubble P (b\_P) in bar.

Another function *dew\_P\_gammaphi* calculates for the composition of the mixture at the liquid phase and pressure at the dewpoint wherein the vapor phase composition and the temperature is given. The user should input the following:

y : row vector containing vapor phase composition of the mixture
T : temperature in K

The output of this function is the dewpoint pressure (d\_P) in bar and a row vector (x) of the composition in the liquid phase.

The function *bubble\_T\_gammaphi* calculates for the bubblepoint temperature and the composition of the mixture in the vapor phase, given the liquid phase composition of the mixture and the pressure in bar. The user should also input these:

x : row vector containing liquid phase composition of the mixture

 P : pressure in bar

This function displays a row vector (y) containing the molar composition of the mixture in the vapor phase and the temperature at bubblepoint (b\_T) in K.

The function *dew\_T\_gammaphi* calculates for the composition of the mixture in the liquid phase and the dew point temperature in K, given the vapor phase composition and the pressure in bar. The following should be provided also:

y : row vector containing vapor phase composition of the mixture
P : pressure in bar

This function displays the liquid phase composition of the mixture in a row vector (x) and the pressure at the dewpoint (d\_T) in bar.

The last function *flashcalc\_gammaphi* calculates the vapor phase and liquid phase compositions of a mixture at a particular pressure and temperature. This function requires the initial calculations for dew P and bubble P, thus the functions *bubble\_P\_gammaphi* and *dew\_P\_gammaphi* were included in the program. The user should input the following:

T : temperature in K

P: pressure in bar

The function displays two row vectors, one containing the vapor phase composition (y) and the other containing the liquid phase composition (x) of the mixture.

**II. The Code**

*A. Main Program*

*1. Bubble P Calculation*

function [b\_P,y] = bubble\_P\_gammaphi(x, T, Tc, w, Zc, Vc, Pc, a, b, c, par, a\_mn)

%bubble\_P\_gammaphi calculates the bubble pressure and vapor phase

%composition of a liquid mixture

%INPUT:

% x - row vector of the mole fraction of the components

% T - Temperature in Kelvin

% Tc - row vector of the critical temperature (K) of each species

% w - row vector

% Zc - row vector

% Vc - row vector of the critical volume (cm3/mol) of each species

% Pc - row vector of the critical pressure (bar) of each species

% a,b,c - row vectors containing the Antoine constants of each species

% par, a\_mn - parameters for UNIFAC

%OUTPUT

% y - vapor phase composition of the liquid mixture

% b\_P - bubble pressure in bar

%matrix holders for the values of PHI, Psat, z, and y

PHI = ones (size(x,2),1);

Psat = zeros (size(x,2),1);

z = zeros (1,size(x,2));

y = zeros (1,size(x,2));

%solves the gamma of the species using UNIFAC

gamma = gamma\_UNIFAC(x, T, par, a\_mn);

%calculates the saturated pressure (kPa) of each species for a given temperature

for i = 1:size(x,2)

 Psat(i) = exp(a(i)-(b(i)/((T-273.15)+c(i))));

end

tol = 10^-6; %tolerance for the iterations

diff = 1;

%iterations for the calculation of bubble P and vapor composition

fprintf('\n\nIteration values of P (bubble P) in bar :\n\n');

while abs(diff)>tol

 %for loop that calculates the value of P

 for i = 1:size(x,2)

 z(i) = (x(i)\*gamma(i)\*Psat(i))/PHI(i);

 end

 P = sum(z);

 %calculates the value of Bij

 B = B\_ij(T, Tc, w, Zc, Vc, Pc);

 %for loop that calculates the vapor phase composition of each species

 for i = 1:size(x,2)

 y(i) = (x(i)\*gamma(i)\*Psat(i))/(P\*PHI(i));

 end

 %calculates phihat and phisat for each species

 phi = phi\_i(y, B, T, P, Psat);

 %for loop that calculates the value of PHI and the bubble pressure

 for i = 1:size(x,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 z(i) = (x(i)\*gamma(i)\*Psat(i))/PHI(i);

 end

 Pnew = sum(z); %bubble pressure in kPa

 diff = P-Pnew;

 disp(P/100);

end

b\_P = Pnew/100; %bubble pressure in bar

end

*2. Dew P Calculation*

function [d\_P,x] = dew\_P\_gammaphi(y,T,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

%dew\_P\_gammaphi calculates the dew pressure and liquid phase

%composition of a liquid mixture

%INPUT:

%y - row vector of the mole fraction of the components

%T - Temperature in Kelvin

%Tc - row vector of the critical temperature (K) of each species

%w - row vector

%Zc - row vector

%Vc - row vector of the critical volume (cm3/mol) of each species

%Pc - row vector of the critical pressure (bar) of each species

%a,b,c - row vectors containing the Antoine constants of each species

%par, a\_mn - parameters for UNIFAC

%OUTPUT

%x - liquid phase composition of the mixture

%d\_P - dew pressure in bar

%matrix holders for the values of PHI, gamma, Psat, Z, X, and Xnew

PHI = ones(size(y,2),1); %PHI = 1

gamma = ones(size(y,2),1); %gamma = 1

Psat = zeros(size(y,2),1);

z = zeros(1,size(y,2));

x = zeros(1,size(y,2));

xnew = zeros(1,size(y,2));

tol = 10^-6; %tolerance

%solves for the saturated pressure (kPa) of each species and the Pressure

%where PHI = 1 and gamma = 1

for i = 1:size(y,2)

 Psat(i) = exp(a(i)-(b(i)/((T-273.15)+c(i))));

 z(i) = (y(i)\*PHI(i))/(gamma(i)\*Psat(i));

end

P = 1/sum(z);

%calculates the new liquid phase composition using the pressure obtained

for i = 1:size(y,2)

 x(i) = (y(i)\*P\*PHI(i))/(gamma(i)\*Psat(i));

end

%solves for the value of gamma for each species

bubble\_P\_gammaphi

%for loop that calculates the new pressure using the new values of gamma

%and PHI = 1;

for i = 1:size(y,2)

 z(i) = (x(i)\*gamma(i)\*Psat(i))/PHI(i);

end

P = sum(z);

error = 1;

fprintf('\n\nIteration values of P (dew P) in kPa :\n\n');

while abs(error) > tol

 B = B\_ij(T, Tc, w, Zc, Vc, Pc); %solves for the values of Bij

 phi = phi\_i(y, B, T, P, Psat); %solves for phihat and phisat

 diff = 1;

 while abs(diff) > tol

 %calculates the new value of PHI and x

 for i = 1:size(y,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 x(i) = (y(i)\*PHI(i)\*P)/(gamma(i)\*Psat(i));

 end

 %normalizes the value of x

 while sum(x) ~= 1

 for i = 1:size(y,2)

 xnew(i) = x(i)/sum(x);

 end

 x = xnew;

 end

 %calculates the new value of gamma using the new value of x

 gammanew = gamma\_UNIFAC(x, T, par, a\_mn);

 diff = gammanew-gamma;

 gamma = gammanew;

 end

 %calculates the new Pressure using the new values of PHI, gamma and x

 for i = 1:size(y,2)

 z(i) = (x(i)\*gamma(i)\*Psat(i))/PHI(i);

 end

 Pnew = sum(z);

 error = Pnew-P;

 P = Pnew;

 disp(P);

end

d\_P = P/100;

end

*3. Bubble T Calculation*

function [b\_T,y] = bubble\_T\_gammaphi(x,Tc,w,Zc,Vc,P,Pc,a,b,c,par,a\_mn)

%bubble\_T\_gammaphi calculates the bubble temperature and vapor phase

%composition of a liquid mixture

%INPUT:

%x - row vector of the mole fraction of the components

%P - Pressure in bar

%Tc - row vector of the critical temperature (K) of each species

%w - row vector

%Zc - row vector

%Vc - row vector of the critical volume (cm3/mol) of each species

%Pc - row vector of the critical pressure (bar) of each species

%a,b,c - row vectors containing the Antoine constants of each species

%par, a\_mn - parameters for UNIFAC

%OUTPUT

%y - vapor phase composition of the liquid mixture

%b\_T - bubble pressure in bar

%convert the pressure to kPa

P=P\*100;

%matrix holders for the values of PHI, Psat, z, and y

PHI = ones(size(x,2),1);

Tsat = zeros(size(x,2),1);

Psat = zeros(size(x,2),1);

z = zeros(1,size(x,2));

y = zeros(1,size(x,2));

%find the value of Tsat using the given Pressure

for i=1:size(x,2)

 Tsat(i)=((b(i)/(a(i)-log(P)))-c(i));

end

%find the initial value of T

for i=1:size(x,2)

 z(i)=((x(i)\*Tsat(i)));

end

T=sum(z);

%find the value of Psat(i) using Reverse Antoine and

%initial value of T

for i=1:size(x,2)

 Psat(i)=exp(a(i)-(b(i)/(T+c(i))));

end

%convert the initial temperature to Kelvin for

%calculation of activity coefficient.

T=T+273.15;

%calculate activity coeffucients using UNIFAC and the T initial

gamma = gamma\_UNIFAC(x, T, par, a\_mn);

%calculating Pjsat where j=1

Pjsat=Psat(1);

for i=1:size(x,2)

 z(i)=((x(i)\*gamma(i)/PHI(i))\*(Psat(i)/Pjsat));

end

j=sum(z);

Pjsat=P/j;

%find the new value of T from Pjsat

T=((b(1)/(a(1)-log(Pjsat)))-c(1));

%iteration to find the bubble temperature and

%composition of the mixture

fprintf('\n\nIteration values of T (bubble T) in Kelvin :\n\n');

tol=10^-6; %tolerance for the iteration

diff=1;

counter=1;

while abs(diff)>tol

 %Evaluate Psat and y from the new value of T

 for i=1:size(x,2)

 Psat(i)=exp(a(i)-(b(i)/(T+c(i))));

 end

 %solve for the vapor phase composition of the mixture

 for i=1:size(x,2)

 y(i)=((x(i)\*gamma(i)\*Psat(i))/(PHI(i)\*P));

 end

 T=T+273.15; %convert the temperate to Kelvin

 %calculate the value of B and phi(hat) and phi(sat)

 %from subroutine functions

 B = B\_ij(T, Tc, w, Zc, Vc, Pc);

 phi = phi\_i(y, B, T, P, Psat);

 %calculate activity coefficients from UNIFAC

 for i = 1:size(x,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 end

 gamma = gamma\_UNIFAC(x, T, par, a\_mn);

 %calculating Pjsat from the previous activity coefficient

 Pjsat=Psat(1);

 for i=1:size(x,2)

 z(i)=((x(i)\*gamma(i)/PHI(i))\*(Psat(i)/Pjsat));

 end

 j=sum(z);

 Pjsat=P/j;

 %find the new value of T from Pjsat

 Tnew=((b(1)/(a(1)-log(Pjsat)))-c(1));

 T=T-273.15;

 disp(T+273.15);

 counter=counter+1;

 %check for the tolerance criterion

 diff=(T-Tnew);

 T=Tnew;

end

disp('No of iterations: ');

disp(counter);

b\_T = T+273.15;

end

*4. Dew T Calculation*

function [d\_T,x] = dew\_T\_gammaphi(y,P,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

%dew\_T\_gammaphi calculates for the composition of the mixture in the liquid

%phase and the dew point temperature in K

%INPUT:

%y - vapor phase composition of the mixture

%P - Pressure in bar

%Tc - row vector of the critical temperature (K) of each species

%w - row vector

%Zc - row vector

%Vc - row vector of the critical volume (cm3/mol) of each species

%Pc - row vector of the critical pressure (bar) of each species

%A,b,C - row vectors containing the Antoine constants of each species

%par, a\_mn - parameters for UNIFAC

%OUTPUT

%x - liquid phase composition of the mixture

%T - dew point temperature in K

PHI = ones(size(y,2),1);

gamma = ones(size(y,2),1);

Tsat = zeros(size(y,2),1);

Psat = zeros(size(y,2),1);

z = zeros(1,size(y,2));

x = zeros(1,size(y,2));

xnew = zeros(1,size(y,2));

%convert input P from bar to kPa

P = P\*100; %P(kPa)

%find value of Tsat using the given P

for i=1:size(y,2)

 Tsat(i)=((b(i)/(a(i)-log(P)))-c(i)); %T(oC)

end

%find the initial value of T

for i=1:size(y,2)

 z(i)=((y(i)\*Tsat(i)));

end

T=sum(z); %T(oC)

%find the value of Psat(i) using Reverse Antoine

for i=1:size(y,2)

 Psat(i)=exp(a(i)-(b(i)/(T+c(i)))); %P(kPa)

end

%calculating Pjsat

Pjsat=Psat(1);

for i=1:size(y,2) %1x2

 z(i)=((y(i)\*PHI(i))/gamma(i))\*(Pjsat/Psat(i));

end

j=sum(z);

Pjsat=P\*j; %(kPa)

%find the new value of T from Pjsat

T=((b(1)/(a(1)-log(Pjsat)))-c(1)); % T(oC)

%solve for Psat(i)

for i=1:size(y,2)

 Psat(i)=exp(a(i)-(b(i)/(T+c(i)))); %kPa

end

%solve for phi(i) and x

T=T+273.15; %K

B = B\_ij(T, Tc, w, Zc, Vc, Pc);

phi = phi\_i(y, B, T, P, Psat);

for i = 1:size(y,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 x(i) = (y(i)\*PHI(i)\*P)/(gamma(i)\*Psat(i));

end

%solve for gamma

gamma = gamma\_UNIFAC(x, T, par, a\_mn);

%solve for Pjsat

Pjsat=Psat(1);

for i=1:size(y,2) %1x2

 z(i)=((y(i)\*PHI(i))/gamma(i))\*(Pjsat/Psat(i));

end

j=sum(z);

Pjsat=P\*j; %(kPa)

%solve for T

T=((b(1)/(a(1)-log(Pjsat)))-c(1)); % T(oC)

Tdiff = 1;

tol = 10^-6;

int = 1;

fprintf('\n\nIteration values of T (dew T) in K :\n\n');

while abs(Tdiff)>tol

 %find Psat

 for i=1:size(y,2)

 Psat(i)=exp(a(i)-(b(i)/(T+c(i)))); %kPa

 end

 %solve for phi(i) and x

 T=T+273.15; %K

 B = B\_ij(T, Tc, w, Zc, Vc, Pc);

 phi = phi\_i(y, B, T, P, Psat);

 for i = 1:size(y,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 x(i) = (y(i)\*PHI(i)\*P)/(gamma(i)\*Psat(i));

 end

 %normalize x

 while sum(x) ~= 1

 for i = 1:size(y,2)

 xnew(i) = x(i)/sum(x);

 end

 x = xnew;

 end

 %solve for gamma

 gamma = gamma\_UNIFAC(x, T, par, a\_mn);

 gammadiff = 1;

 while abs(gammadiff) > tol

 %solve for x

 for i = 1:size(x,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 x(i) = (y(i)\*PHI(i)\*P)/(gamma(i)\*Psat(i));

 end

 while sum(x) ~= 1

 for i = 1:size(y,2)

 xnew(i) = x(i)/sum(x);

 end

 x = xnew;

 end

 gammanew = gamma\_UNIFAC(x, T, par, a\_mn);

 gammadiff = gammanew-gamma;

 gamma = gammanew;

 end

 %calculating Pjsat

 Pjsat=Psat(1);

 for i=1:size(y,2) %1x2

 z(i)=(y(i)\*PHI(i)\*Pjsat)/(Psat(i)\*gamma(i));

 end

 j=sum(z);

 Pjsat=P\*j; %kPa

 %find the new value of T from Pjsat

 Tnew=((b(1)/(a(1)-log(Pjsat)))-c(1)); %oC

 T=T-273.15;

Tdiff = Tnew-T;

T=Tnew;

disp(T+273.15) %K

int=int+1;

end

disp('No. of iterations =')

disp(int)

d\_T = T+273.15;

end

*5. Flash Calculation*

function [x,y] = flashcalc\_gammaphi(T,P,z,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

%flashcalc\_gammaphi calculates the vapor phase and liquid phase

%compositions of a liquid mixture

%INPUT:

%P - Pressure in kPa

%T - Temperature in Kelvin

%Tc - row vector of the critical temperature (K) of each species

%w - row vector

%Zc - row vector

%Vc - row vector of the critical volume (cm3/mol) of each species

%Pc - row vector of the critical pressure (bar) of each species

%a,b,c - row vectors containing the Antoine constants of each species

%par, a\_mn - parameters for UNIFAC

%OUTPUT

%x - liquid phase composition of the mixture

%y - vapor phase composition of the mixture

PG = P;

%matrix holders for the values of xold, yold, Psat, PHI, K, u, q and gamma

xold = zeros(1,size(z,2));

yold = zeros(1,size(z,2));

Psat = zeros(size(z,2),1);

PHI = zeros(size(z,2),1);

K = zeros(size(z,2),1);

u = zeros(1,size(z,2));

q = zeros(1,size(z,2));

gamma = zeros(size(z,2),1);

%calculates the saturated pressure (kPa) of each species

for i = 1:size(z,2)

 Psat(i) = exp(a(i)-(b(i)/((T-273.15)+c(i))));

end

%for the dew point values

y = z; %vapor phase composition

V\_d = 1; %value of V for dew point

%calculates the dew pressure (bar) and liquid phase composition

[d\_P,x] = dew\_P\_gammaphi(y,T,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn);

gamma\_d = gamma\_UNIFAC(x, T, par, a\_mn); %calculates the gamma of each species

B = B\_ij(T, Tc, w, Zc, Vc, Pc); %calculates the Bij

P = d\_P\*100; %converts the dew pressure to kPa

phi = phi\_i(y, B, T, P, Psat); %give the values of phihat and phisat

%calculates the values of PHI

for i = 1:size(z,2)

 PHI(i) = (phi(i,1)/phi(i,2));

end

PHI\_d = PHI;

%for bubble point values

x = z; %liquid phase composition

V\_b = 0; %value of V for bubble point

%calculates the bubble pressure (bar) and vapor phase composition

[b\_P,y] = bubble\_P\_gammaphi(x,T,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn);

gamma\_b = gamma\_UNIFAC(x, T, par, a\_mn); %calculates the gamma of each species

B = B\_ij(T, Tc, w, Zc, Vc, Pc); %calculates the Bij

P = b\_P\*100; %converts the bubble pressure to kPa

phi = phi\_i(y, B, T, P, Psat); %give the values of phihat and phisat

%calculates the values of PHI

for i = 1:size(z,2)

 PHI(i) = (phi(i,1)/phi(i,2));

end

PHI\_b = PHI;

%for flash calculation values

PG = PG/100; %converts the given pressure to bar

if PG < b\_P && PG > d\_P

 %interpolates the values of V, gamma and PHI from the calculates values

 %of V, gamma, and PHI from the dew and bubble point values

 V = V\_d - ((d\_P-P)\*(V\_d-V\_b))/(d\_P-b\_P);

 for i = 1:size(z,2)

 gamma(i) = gamma\_d(i) - ((d\_P-P)\*(gamma\_d(i)-gamma\_b(i)))/(d\_P-b\_P);

 PHI(i) = PHI\_d(i) - ((d\_P)\*(PHI\_d(i)-PHI\_b(i))/(d\_P-b\_P));

 end

 diff1 = 1;

 diff2 = 1;

 diff3 = 1;

 tol = 10^-6;

 PG = PG\*100; %converts the given pressure to kPa

 while abs(diff1) > tol && abs(diff2) > tol && abs(diff3) > tol

 %calculates the values of K, F (sum of u) and dF/dV (sum of q)

 for i = 1:size(z,2)

 K(i) = (gamma(i)\*Psat(i))/(PHI(i)\*PG);

 u(i) = (z(i)\*(K(i)-1))/(1+V\*(K(i)-1));

 q(i) = (z(i)\*(K(i)-1)^2)/((1+V\*(K(i)-1))^2);

 end

 F = sum(u);

 dFdV = -sum(q);

 Vnew = V - (F/dFdV); %new value of V using Newton's method

 %solves for the vapor and liquid phase compositions

 for i = 1:size(z,2)

 x(i) = z(i)/(1+Vnew\*(K(i)-1));

 y(i) = K(i)\*x(i);

 end

 gamma = gamma\_UNIFAC(x, T, par, a\_mn); %solves for the gamma of each species

 B = B\_ij(T, Tc, w, Zc, Vc, Pc); %solves for Bij

 P = PG;

 phi = phi\_i(y, B, T, P, Psat); %solves for phihat and phisat

 %solves for PHI

 for i = 1:size(z,2)

 PHI(i) = (phi(i,1)/phi(i,2));

 end

 PG = P;

 diff1 = V - Vnew;

 V = Vnew;

 diff2 = xold - x;

 xold = x;

 diff3 = yold - y;

 yold = y;

 end

 x = xold;

 y = yold;

end

end

*B. Subroutines*

*1. Bij computation*

function B = B\_ij(T, Tc, w, Zc, Vc, Pc)

%B\_ij calculates for the values of B which will be used to calculate the

%value of phihat

%INPUT:

%T - Temperature in Kelvin

%Tc - row vector of the critical temperature (K) of each species

%w - row vector

%Zc - row vector

%Vc - row vector of the critical volume (cm3/mol) of each species

%Pc - row vector of the critical pressure (bar) of each species

%OUTPUT

%B - square matrix

%matrix holders for the values of w\_ij, Tc\_ij, Zc\_ij, Vc\_ij, Pc\_ij, Tr\_ij,

%and B\_ij

w\_ij = zeros(size(w,2));

Tc\_ij = zeros(size(Tc,2));

Zc\_ij = zeros(size(Zc,2));

Vc\_ij = zeros(size(Vc,2));

Pc\_ij = zeros(size(Pc,2));

Tr\_ij = zeros(size(Tc,2));

B\_ij = zeros(size(Tc,2));

R = 83.14; %unit: (bar cm3/mol K)

%calculates the values of w\_ij, Tc\_ij, Zc\_ij, Vc\_ij, Pc\_ij, Tr\_ij,and B\_ij

for i = 1:size(w,2)

 for j = 1:size(w,2)

 w\_ij(i,j) = (w(i)+w(j))/2;

 Tc\_ij(i,j) = (Tc(i)\*Tc(j))^(1/2);

 Zc\_ij(i,j) = (Zc(i)+Zc(j))/2;

 Vc\_ij(i,j) = (((Vc(i)^(1/3))+(Vc(j)^(1/3)))/2)^3;

 Pc\_ij(i,j) = (Zc\_ij(i,j)\*R\*Tc\_ij(i,j))/Vc\_ij(i,j);

 Tr\_ij(i,j) = T/Tc\_ij(i,j);

 B0 = 0.083-(0.422/(Tr\_ij(i,j)^1.6));

 B1 = 0.139-(0.172/(Tr\_ij(i,j)^4.2));

 B\_ij(i,j) = ((R\*Tc\_ij(i,j))/Pc\_ij(i,j))\*(B0+w\_ij(i,j)\*B1);

 end

end

B = B\_ij;

end

*2. Computation of φ^ values*

function phi = phi\_i(y, B, T, P, Psat)

%input P and Psat is in kPa

%input T is in K

%B\_ij calculates for the values of B which will be used to calculate the

%value of phihat

%INPUT:

%y - row vector of the vapor phase composition

%T - Temperature in Kelvin

%P - Pressure in kPa

%Psat - saturated pressure in kPa

%B - square matrix of the values of B

%OUTPUT

%phi - row matrix [phihat phisat]

%matrix holders for the values of del\_ij, z, phihat, and phisat

del\_ij = zeros(size(y,2));

z = zeros(size(y,2));

phihat = zeros(size(y,2),1);

phisat = zeros(size(y,2),1);

R = 8314; %unit: (kPa cm3/mol K)

%calculates the values of del\_ij, phihat and phisat

for k = 1:size(y,2)

 for i = 1:size(y,2)

 for j = 1:size(y,2)

 del\_ij(i,j) = 2\*B(i,j)-B(i,i)-B(j,j);

 z(i,j) = y(i)\*y(j)\*(2\*del\_ij(i,k)-del\_ij(i,j));

 end

 end

 a = sum(z);

 phihat(k) = exp((P/(R\*T))\*(B(k,k)+(1/2)\*sum(a)));

 phisat(k) = exp((B(k,k)\*Psat(k))/(R\*T));

end

phi = [phihat phisat];

end

 *2. Computation of* γi *values*

function gamma = gamma\_UNIFAC(x, T, par, a\_mn);

%gamma\_UNIFAC calculates the activity coefficient

%INPUT:

%x - row vector of the mole fraction of the components

%T - Temperature in Kelvin

%par, a\_mn - parameters for UNIFAC

%OUTPUT

%gamma - activity coefficient

 gamma =[1,1];

end

**III. Results of Test Data**

System: methyl acetate(1)/methanol(2)

Table 1. Critical constants of species 1 and 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| species | T­c (K) | ωc | Zc | Vc (cm3/mol) | Pc (bar) |
| 1 | 506.6 | 0.331 | 0.257 | 228 | 47.5 |
| 2 | 512.6 | 0.564 | 0.224 | 118 | 80.97 |

Table 2. Antoine Constants.

|  |  |
| --- | --- |
| Antoine Constants | Range of Validity |
| Species | A | B | C | T (oC) |
| 1 | 14.2456 | 2662.78 | 219.69 | -23 to 78 |
| 2 | 16.5785 | 3638.27 | 239.50 | -11 to 83 |

**Table 3.** Parameters of different functional groups in the mixture.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Main group No.** | **Subgroup No., k** | **Rk** | **Qk** | **vk(1)** | **v­k(2)** |
| **CH3** | 1 | 1 | 0.9011 | 0.848 | 1 | 0 |
| **CH3COO** | 11 | 21 | 1.9031 | 1.728 | 1 | 0 |
| **CH3OH** | 6 | 15 | 1.4311 | 1.432 | 0 | 1 |

**Table 4.** Interaction Parameters for all possible pairs of functional groups.

|  |  |  |  |
| --- | --- | --- | --- |
| am,n | **1** | **11** | **4** |
| **1** | 0 | 232.1 | 697.2 |
| **11** | 114.8 | 0 | 249.6 |
| **6** | 16.51 | -10.72 | 0 |

A. Bubble P Calculation

*Input to the program:*

>>x = [0.3 0.7];

>>T = 348.15;

>>Tc = [506.6 512.6];

>>w = [0.331 0.564];

>>Zc = [0.257 0.224];

>>Vc = [228 118];

>>Pc = [47.5 80.97];

>>a = [14.2456 16.5785];

>>b = [2662.78 3638.27];

>>c = [219.69 239.5];

>>par = [1 1 0.9011 0.848 1 0;11 21 1.9031 1.728 1 0;6 15 1.4311 1.432 0 1];

>>a\_mn = [0 232.1 697.2;114.8 0 249.6;16.51 -10.72 0];

*Invoke the program*

bubble\_P\_gammaphi(x,T,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

*Output of the program*

B. Dew P Calculation

*Input to the program:*

>>y = [0.4309 0.591];

>>T = 348.15;

>>Tc = [506.6 512.6];

>>w = [0.331 0.564];

>>Zc = [0.257 0.224];

>>Vc = [228 118];

>>Pc = [47.5 80.97];

>>a = [14.2456 16.5785];

>>b = [2662.78 3638.27];

>>c = [219.69 239.5];

>>par = [1 1 0.9011 0.848 1 0;11 21 1.9031 1.728 1 0;6 15 1.4311 1.432 0 1];

>>a\_mn = [0 232.1 697.2;114.8 0 249.6;16.51 -10.72 0];

*Invoke the program*

dew\_P\_gammaphi(y,T,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

*Output of the program*

C. Bubble T Calculation

*Input to the program:*

>>x = [0.3 0.7];

>>P = 2.0144;

>>Tc = [506.6 512.6];

>>w = [0.331 0.564];

>>Zc = [0.257 0.224];

>>Vc = [228 118];

>>Pc = [47.5 80.97];

>>a = [14.2456 16.5785];

>>b = [2662.78 3638.27];

>>c = [219.69 239.5];

>>par = [1 1 0.9011 0.848 1 0;11 21 1.9031 1.728 1 0;6 15 1.4311 1.432 0 1];

>>a\_mn = [0 232.1 697.2;114.8 0 249.6;16.51 -10.72 0];

*Invoke the program*

bubble\_T\_gammaphi(x,Tc,w,Zc,Vc,P,Pc,a,b,c,par,a\_mn)

*Output of the program*

D. Dew T Calculation

*Input to the program:*

>>y = [0.4309 0.5691];

>>P = 2.0144;

>>Tc = [506.6 512.6];

>>w = [0.331 0.564];

>>Zc = [0.257 0.224];

>>Vc = [228 118];

>>Pc = [47.5 80.97];

>>a = [14.2456 16.5785];

>>b = [2662.78 3638.27];

>>c = [219.69 239.5];

>>par = [1 1 0.9011 0.848 1 0;11 21 1.9031 1.728 1 0;6 15 1.4311 1.432 0 1];

>>a\_mn = [0 232.1 697.2;114.8 0 249.6;16.51 -10.72 0];

*Invoke the program*

dew\_T\_gammaphi(x,Tc,w,Zc,Vc,P,Pc,a,b,c,par,a\_mn)

*Output of the program*

E. Flash Calculation

*Input to the program:*

>>z = [0.2999 0.7001];

>>T = 348.15;

>>P = 2.0144;

>>Tc = [506.6 512.6];

>>w = [0.331 0.564];

>>Zc = [0.257 0.224];

>>Vc = [228 118];

>>Pc = [47.5 80.97];

>>a = [14.2456 16.5785];

>>b = [2662.78 3638.27];

>>c = [219.69 239.5];

>>par = [1 1 0.9011 0.848 1 0;11 21 1.9031 1.728 1 0;6 15 1.4311 1.432 0 1];

>>a\_mn = [0 232.1 697.2;114.8 0 249.6;16.51 -10.72 0];

*Invoke the program:*

[x,y] = flashcalc\_gammaphi(T,P,z,Tc,w,Zc,Vc,Pc,a,b,c,par,a\_mn)

*Output of the program:*