**Dew Temperature & Pressure**

function bubble\_temp\_and\_pres()

% Constants for Antoine's equations

A1 = 14.5463;

B1 = 2940.46;

C1 = 237.22;

A2 = 14.2724;

B2 = 2945.467;

C2 = 224.0;

A3 = 14.2043;

B3 = 2972.64;

C3 = 209.00;

% Part (i) - P and xi given t=70°C, y1=0.5, y2=0.3, y3=0.2

t1 = 70; % Temperature in °C

y1 = 0.5;

y2 = 0.3;

y3 = 0.2;

% Calculate P using Raoult's law

P1 = exp(A1 - B1 / (t1 + C1));

P2 = exp(A2 - B2 / (t1 + C2));

P3 = exp(A3 - B3 / (t1 + C3));

P = y1 \* P1 + y2 \* P2 + y3 \* P3;

% Calculate xi using Raoult's law

x1 = y1 \* P1 / P;

x2 = y2 \* P2 / P;

x3 = y3 \* P3 / P;

% Display results for Part (i)

disp('Part (i): P and xi given t=70°C, y1=0.5, y2=0.3, y3=0.2')

disp(['P: ', num2str(P), ' kPa'])

disp(['x1: ', num2str(x1)])

disp(['x2: ', num2str(x2)])

disp(['x3: ', num2str(x3)])

disp(' ')

% Part (ii) - t and xi given P=90kPa, y1=0.6, y2=0.2, y3=0.2

P\_target = 90; % Target pressure in kPa

y1 = 0.6;

y2 = 0.2;

y3 = 0.2;

% Solve for t using Antoine's equation and Raoult's law

fun = @(t) y1 \* exp(A1 - B1 / (t + C1)) + y2 \* exp(A2 - B2 / (t + C2)) + y3 \* exp(A3 - B3 / (t + C3)) - P\_target;

t2 = fsolve(fun, 70);

% Calculate xi using Raoult's law

P1 = exp(A1 - B1 / (t2 + C1));

P2 = exp(A2 - B2 / (t2 + C2));

P3 = exp(A3 - B3 / (t2 + C3));

x1 = y1 \* P1 / P\_target;

x2 = y2 \* P2 / P\_target;

x3 = y3 \* P3 / P\_target;

% Display results for Part (ii)

disp('Part (ii): t and xi given P=90kPa, y1=0.6, y2=0.2, y3=0.2')

disp(['t: ', num2str(t2), ' °C'])

disp(['x1: ', num2str(x1)])

disp(['x2: ', num2str(x2)])

disp(['x3: ', num2str(x3)])

end

**Bubble Pressure**

t\_i = 80; % °C

x\_i = [0.25, 0.35, 0.4];

vaporPhase\_i = calculateVaporPhase(t\_i, x\_i);

disp('(i) Given t = 80°C, x1 = 0.25, x2 = 0.35, x3 = 0.4:');

disp([' P = ', num2str(vaporPhase\_i.P), ' kPa']);

disp([' yi = ', num2str(vaporPhase\_i.yi)]);

function vaporPhase = calculateVaporPhase(t, x)

% Antoine's equations

lnP1 = 14.5463 - 2940.46 / (t + 237.22);

lnP2 = 14.2724 - 2945.467 / (t + 224.0);

lnP3 = 14.2043 - 2972.64 / (t + 209.00);

% Vapor pressures

P1 = exp(lnP1);

P2 = exp(lnP2);

P3 = exp(lnP3);

% Raoult's law for vapor pressure of the mixture

P = x(1) \* P1 + x(2) \* P2 + x(3) \* P3;

% Vapor phase mole fractions

yi = [x(1) \* P1 / P, x(2) \* P2 / P, x(3) \* P3 / P];

% Output

vaporPhase.P = P;

vaporPhase.yi = yi;

end

**Bubble Temperature & Pressure**

t\_i = 80; % °C

x\_i = [0.25, 0.35, 0.4];

vaporPhase\_i = calculateVaporPhase(t\_i, x\_i);

disp('(i) Given t = 80°C, x1 = 0.25, x2 = 0.35, x3 = 0.4:');

disp([' P = ', num2str(vaporPhase\_i.P), ' kPa']);

disp([' yi = ', num2str(vaporPhase\_i.yi)]);

P\_ii = 80; % kPa

x\_ii = [0.3, 0.45, 0.25];

% Solve for t and yi

options = optimset('Display', 'off');

[t\_ii, ~, exitflag] = fsolve(@(t) calculateVaporPhase(t, x\_ii).P - P\_ii, 80, options);

if exitflag > 0

vaporPhase\_ii = calculateVaporPhase(t\_ii, x\_ii);

disp('(ii) Given P = 80 kPa, x1 = 0.3, x2 = 0.45, x3 = 0.25:');

disp([' t = ', num2str(t\_ii), ' °C']);

disp([' yi = ', num2str(vaporPhase\_ii.yi)]);

else

disp('No solution found.');

end

function vaporPhase = calculateVaporPhase(t, x)

% Antoine's equations

lnP1 = 14.5463 - 2940.46 / (t + 237.22);

lnP2 = 14.2724 - 2945.467 / (t + 224.0);

lnP3 = 14.2043 - 2972.64 / (t + 209.00);

% Vapor pressures

P1 = exp(lnP1);

P2 = exp(lnP2);

P3 = exp(lnP3);

% Raoult's law for vapor pressure of the mixture

P = x(1) \* P1 + x(2) \* P2 + x(3) \* P3;

% Vapor phase mole fractions

yi = [x(1) \* P1 / P, x(2) \* P2 / P, x(3) \* P3 / P];

% Output

vaporPhase.P = P;

vaporPhase.yi = yi;

end

**Dew Temperature**

function bubble\_temp\_and\_pres()

% Constants for Antoine's equations

A1 = 14.5463;

B1 = 2940.46;

C1 = 237.22;

A2 = 14.2724;

B2 = 2945.467;

C2 = 224.0;

A3 = 14.2043;

B3 = 2972.64;

C3 = 209.00;

% Part (ii) - t and xi given P=90kPa, y1=0.6, y2=0.2, y3=0.2

P\_target = 90; % Target pressure in kPa

y1 = 0.6;

y2 = 0.2;

y3 = 0.2;

% Solve for t using Antoine's equation and Raoult's law

fun = @(t) y1 \* exp(A1 - B1 / (t + C1)) + y2 \* exp(A2 - B2 / (t + C2)) + y3 \* exp(A3 - B3 / (t + C3)) - P\_target;

t2 = fsolve(fun, 70);

% Calculate xi using Raoult's law

P1 = exp(A1 - B1 / (t2 + C1));

P2 = exp(A2 - B2 / (t2 + C2));

P3 = exp(A3 - B3 / (t2 + C3));

x1 = y1 \* P1 / P\_target;

x2 = y2 \* P2 / P\_target;

x3 = y3 \* P3 / P\_target;

% Display results for Part (ii)

disp('Part (ii): t and xi given P=90kPa, y1=0.6, y2=0.2, y3=0.2')

disp(['t: ', num2str(t2), ' °C'])

disp(['x1: ', num2str(x1)])

disp(['x2: ', num2str(x2)])

disp(['x3: ', num2str(x3)])

end

**Bubble Temperature**

P\_ii = 80; % kPa

x\_ii = [0.3, 0.45, 0.25];

% Solve for t and yi

options = optimset('Display', 'off');

[t\_ii, ~, exitflag] = fsolve(@(t) calculateVaporPhase(t, x\_ii).P - P\_ii, 80, options);

if exitflag > 0

vaporPhase\_ii = calculateVaporPhase(t\_ii, x\_ii);

disp('( Given P = 80 kPa, x1 = 0.3, x2 = 0.45, x3 = 0.25:');

disp([' t = ', num2str(t\_ii), ' °C']);

disp([' yi = ', num2str(vaporPhase\_ii.yi)]);

else

disp('No solution found.');

end

function vaporPhase = calculateVaporPhase(t, x)

% Antoine's equations

lnP1 = 14.5463 - 2940.46 / (t + 237.22);

lnP2 = 14.2724 - 2945.467 / (t + 224.0);

lnP3 = 14.2043 - 2972.64 / (t + 209.00);

% Vapor pressures

P1 = exp(lnP1);

P2 = exp(lnP2);

P3 = exp(lnP3);

% Raoult's law for vapor pressure of the mixture

P = x(1) \* P1 + x(2) \* P2 + x(3) \* P3;

% Vapor phase mole fractions

yi = [x(1) \* P1 / P, x(2) \* P2 / P, x(3) \* P3 / P];

% Output

vaporPhase.P = P;

vaporPhase.yi = yi;

end

**Dew Pressure**

t=input('given temperature T= ');

n=input('no. of components n= ');

for i=1:n

y(i)=input('mole composition in gaseous phase y= ');

a(i)=input('A = ');

b(i)=input('B = ');

c(i)=input('C = ');

end

fprintf('saturation pressures\n: ')

for i=1:n

ps(i)=exp(a(i)-b(i)/(t+c(i)));

fprintf('Ps(%d) is:%f\n',i,ps(i));

end

for i=1:n

P=1/sum(y./ps);

end

fprintf('the total pressure Pt= :%f\n',P);

fprintf('mole composition in liquid phase: \n')

for i=1:n

x(i)=(y(i)\*P)/ps(i);

fprintf('x(%d)= %f\n',i,x(i));

end

**Laminar Flow**

% Given parameters

Diameter = 0.05; % Pipe internal diameter in meters (50mm to meters)

umax = 1; % Maximum velocity in m/s

r0 = Diameter / 2; % Pipe radius

% Generate radial positions

r = linspace(0, r0, 100); % Adjust the number of points as needed

% Calculate velocity profile

u = umax \* (1 - (r.^2 / r0^2));

% Plot the velocity profile

figure;

plot(r, u, 'LineWidth', 2);

title('Laminar Flow Velocity Profile');

xlabel('Radius (m)');

ylabel('Velocity (m/s)');

grid on;

% Add labels to the plot

legend('Velocity Profile');

**Velocity Profile**

r=[-0.025 -0.02 -0.015 -0.01 -0.005 0 0.005 0.01 0.015 0.02 0.025 ];

umax=1

r0=0.025

for i=1:11

u(i)=umax\*(1-((r(i)\*r(i))/(r0\*r0)));

disp(u(i));

end

**Nusselt’s No.**

Re=[2000:500:8000];

Pr=6;

Nnu=(0.023\*(power(Re,0.8))\*(power(Pr,0.3))\*1);

plot(Re,Nnu)

xlabel('reynolds no')

ylabel('nusselt no')

**Venturimeter Discharge**

Rm=[0.002:0.002:0.02];

d1=0.0254;

d2=0.0127;

rhom=13600;

rho=1000;

cd=0.98;

g=9.81;

h=((rhom-rho)/rho).\*Rm;

a1=((d1\*d1)\*(3.14/4));

a2=((d2\*d2)\*(3.14/4));

L=(a1\*a2);

F=((2\*g).\*h)

N=(a1\*a1)-(a2\*a2);

K=(L/N).\*F;

Q=cd\*power(K,0.5);

plot(Rm,Q)

xlabel('radius')

ylabel('discharge')

**Flash Vapourisation**

P = 110; % Pressure in kPa

T = 80; % Temperature in °C

z = [0.45, 0.35, 0.2]; % Overall composition

[L, V, xi, yi] = flash(P, T, z);

function [L, V, xi, yi] = flash(P, T, z)

% Antoine equation parameters

A = [14.5463, 14.2724, 14.2043];

B = [2940.46, 2945.467, 2972.64];

C = [237.22, 224, 209];

% Calculate saturation pressures (ps) for each component

for i = 1:3

ps(i) = exp(A(i) - (B(i) / (C(i) + T)));

end

% Calculate saturation pressure of the mixture (pb)

pb = sum(z .\* ps);

% Calculate inverse of ps (pd)

pd = sum(z ./ ps);

pd = 1 / pd;

% Check if flash vaporization is possible

if (pd < P) && (P < pb)

% Calculate ki

k = ps / P;

% Initial guess for vapor fraction (vo)

Vo = (P - pd) / (pb - pd);

% Flash calculation using Newton-Raphson method

for i = 1:10

fv = sum((z .\* k) ./ (1 + Vo .\* (k - 1))) - 1;

f1v = sum((-z .\* k) .\* (k - 1) ./ ((1 + Vo .\* (k - 1)).^2));

% Update vapor fraction

Vn = Vo - (fv / f1v);

% Check for convergence

esp = abs((Vn - Vo) / Vo);

if (esp <= 0.001)

V = Vn;

L = 1 - V;

y = (z .\* k) ./ (1 + Vn .\* (k - 1));

x = y .\* k;

% Normalize compositions

xi = x / sum(x);

yi = y / sum(y);

% Display results

disp('Flash Vaporization Results:');

disp(['Liquid Fraction (L): ', num2str(L)]);

disp(['Vapor Fraction (V): ', num2str(V)]);

disp(['Liquid Phase Composition (xi): ', num2str(xi)]);

disp(['Vapor Phase Composition (yi): ', num2str(yi)]);

return;

else

Vo = Vn;

end

end

else

disp('Flash vaporization not possible for the given conditions.');

end

end