BENEDICT-WEBB-RUBIN EQUATION OF STATE FOR NITROGEN

CHE221A COMPUTATIONAL ASSIGNMENT-1 **MANEESH P M, 180404**

INTRODUCTION

The Benedict-Webb-Rubin(BWR) equation of state was developed by Manson Benedict, G. B. Webb and L. C. Rubin while working at the M. W. Kellogg laboratory. The three researchers rearranged the Beattie-Bridgeman equation of state and increased the number of experimentally determined constants to eight.

THE ORIGINAL BWR EQUATION OF STATE [1]

$$P = \left(R_u \frac{T}{v}\right) + \frac{\left(B_o R_u T - A_o - \frac{C_o}{T^2}\right)}{v^2} + \frac{bR_u T - a}{v^3} + a\frac{\alpha}{v^6} + \frac{c}{T^2 v^3}\left(1 + \frac{\gamma}{v^2}\right)\left(e^{-\frac{\gamma}{v^2}}\right)$$

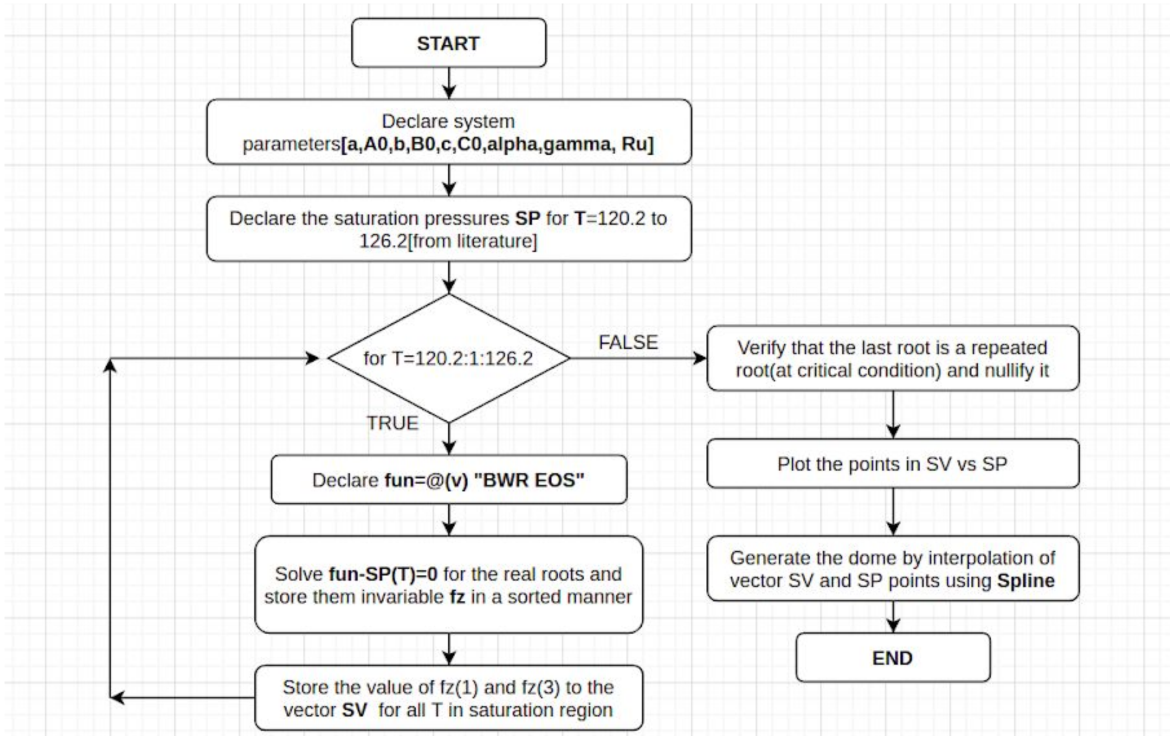
Further modifications progressed BWR equation of state to a 32 term version. This Modified BWR(mBWR) equation of state was developed by Jacobson and Stewart by fitting the equation of state to empirical data for reference fluid.

VALUES OF CONSTANTS APPEARING IN THE EQUATION FOR NITROGEN [2]

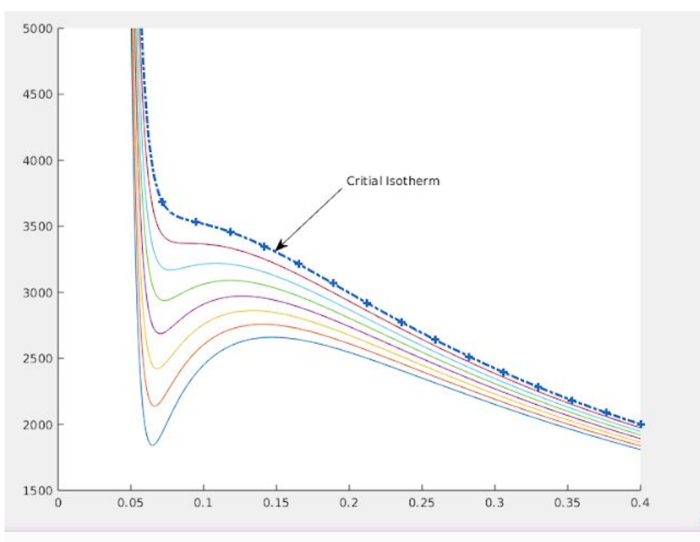
Benedict-Webb-Rubin Equation Of State, Nitrogen	
Parameter	Value
a	2.54
A_0	106.73
b	0.002328
B_0	0.04074
c	7.379×10^4
C_0	8.164×10^5
α	1.272×10^4
γ	0.0053

[1] & [2] INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN CHEMICAL SCIENCES(IJARCS) V. 4 ISS. 5, 2017 PP 21-29

LOGIC OF THE CODE

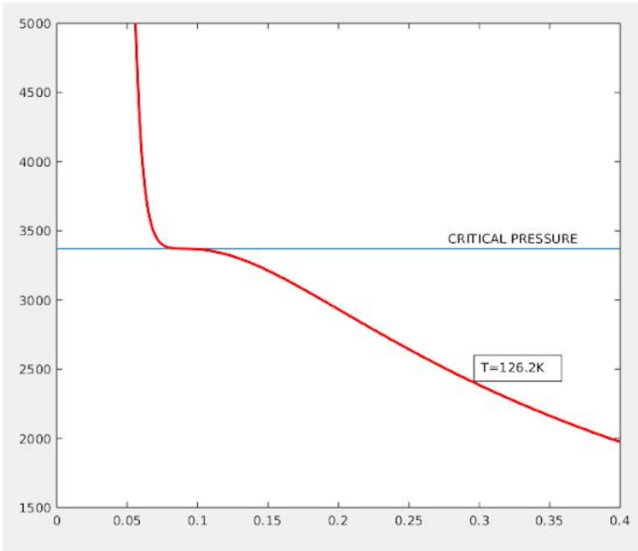


BEHAVIOUR OF EOS AT VARIOUS TEMPERATURE



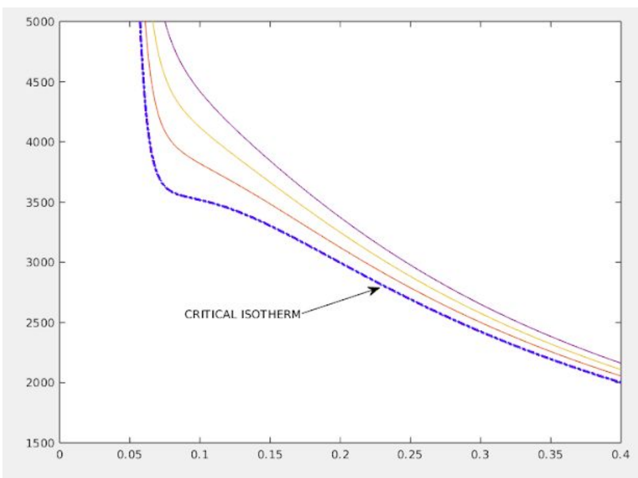
i) T < Tc

Below the critical temperature, the isotherms in the region follow a cubic trend. In this region, the isotherms have two roots at a particular pressure(saturation pressure). This is the liquid-vapour saturation region.



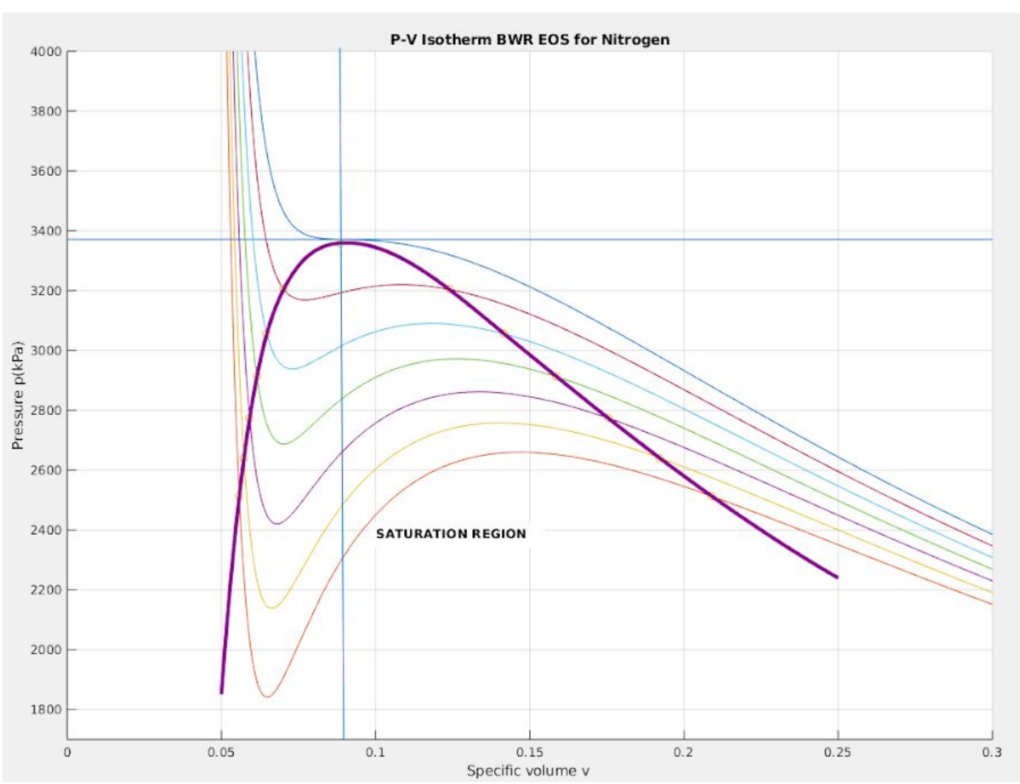
ii) T =Tc

At the critical temperature, the distinct roots merges into a single root. This point of inflection is the critical point. At this point $(\frac{\delta P}{\delta V})_T$ and $(\frac{\delta^2 P}{\delta V^2})_T$ vanishes.



iii) T > Tc

Above the critical temperature the curve seems to have no real root. Physically above the critical temperature, there is no distinction between the solid-liquid-gas phases.



iv) Two phase region

MATLAB CODE

```
% CHE221A computational assignment-1
% Maneesh P M 180404
clc;

% Declaring constants for Nitrogen
a=2.54;
A0=106.73;
b=0.002328;
B0=0.04074;
c=73790;
C0=816400;
al=0.0001272;
gam=0.0053;
Ru=8.31434;

SP=3370.9237;                 % Critical pressure from literature
fplot(SP);

% Saturation Pressure for various pressures from literature
sp(1)=2512.86;
sp(2)=2641.03613;
sp(3)=2774.07585;
sp(4)=2912.18183;
sp(5)=3055.5567;
sp(6)=3204.40312;
sp(7)=3370.9237;


for T=120.2:1:126.2          % To iterate over temperature
        i=T-119.2;           % Loop counter
        hold on

        syms v;              % Plotting bwr for the temp
        fplot(Ru*T/v+(B0*Ru*T-A0-C0/(T^2))/(v^2)+(b*Ru*T-a)/(v^3)+a*al/(v^6)+c*(1+gam/(v^2))*exp(-gam/(v^2))/((
        v^3)*(T^2)), [0.001 1]);
        xlim([0 0.4])
        ylim([1500 5000])

        v = linspace(0.05,0.25);          % Interval To Evaluate Over, by inspection

        %converting the equation to a function and using a guessed value approach to find the roots
        f = @(v) (Ru .* T./v + (B0.*Ru.*T - A0 - C0./(T.^2))./(v.^2) + (b.*Ru.*T - a)./(v.^3) + a.*al./(v.^6) +
        c.*(1+gam./(v.^2)).*exp(-gam./(v.^2))./((v.^3).*(T.^2))-sp(i)) ;          % Function
        fx = f(v);                               % Function Evaluated Over ‘x’
        cs = fx.*circshift(fx,-1,2);             % Product Negative At Zero-Crossings
        xc = v(cs <= 0);                         % Values Of ‘x’ Near Zero Crossings
        for k1 = 1:length(xc)
                fz(k1) = fzero(f, xc(k1));       % Use ‘xc’ As Initial Zero Estimate
        end
```

```
        SV(i*2-1)=fz(1);                    %saving the first and third root in the saturation region
        SV(i*2)=fz(3);

end

% X axis vector for plotting
SP=[2512.86   2512.86   2641.03613   2641.03613   2774.07585   2774.07585   2912.18183   2912.18183   3055.5567
3055.5567 3204.40312 3204.40312 3358.9237 3358.9237];

SV(14) = [];    % nullifying the last root since repeated root at Tc
SP(14) = [];

% plotting the saved values using a cubic spline
y=SP;
x=SV;
plot(SV,SP,'o');
xx = 0.05:.001:0.25;
yy = spline(x,y,xx);
SP=plot(x,y,'o',xx,yy);
SP(2).LineWidth=3;
hold on
```