Homework Assignment Number Two Solutions

Problem 1.
Consider the Peng-Robinson Equation of state as given below. The critical temperature and pressure of oxygen are also given below. Each root of this equation is a molar volume. Find all of the roots of the Peng-Robinson equation for oxygen at the temperatures given below and for a pressure of 1.0 atmosphere.

(a) T = 98.0 K
(b) T = 298.0 K

(For those of you who have not had thermodynamics, there can only be one phase above the critical temperature.)

$$P = \frac{RT}{V-b} - \frac{a(T)}{V(V+b)+b(V-b)}$$

where, $R = 8.314$ J/mol/K, and $a(T) = 0.45724\frac{R^2 T_c^{\,2}}{P_c}\alpha(T)$, $b = 0.07780\frac{RT_c}{P_c}$,

$\alpha(T) = \left[1+\kappa\left(1-\sqrt{\frac{T}{T_c}}\right)\right]^2$ where, for the oxygen molecule $\kappa = 0.4069$, $T_c = 154.6$ K, and $P_c = 5.046 \cdot 10^6$ Pa.

Solution:

Using the Newton-Raphson method with numerical derivatives as implemented in the code, nrnd1.m, we input the function,

```
function f = funkeval(x)
V = x;
T = 98; % K
p = 101325; % Pa
R = 8.314; % J/mol/K
Tc = 154.6; % K
pc = 5.046e+6; % Pa
kappa = 0.4069;
alpha = (1 + kappa*(1 - sqrt(T/Tc)))^2;
b = 0.07780*R*Tc/pc;
a = 0.45724*R^2*Tc^2/pc*alpha;
f = R*T/(V-b) - a/(V*(V+b) + b*(V-b)) - p;
```

(a) T = 98.0 K

This temperature is below the critical temperature. We should find three roots.

The ideal gas should give a good estimate of the vapor root

```
T = 98; % K
p = 101325; % Pa
R = 8.314; % J/mol/K
V = R*T/p

V =   0.008041174438687
```

We now run the Newton Raphson with Numerical Derivatives code for 1 unknown (nrnd1) with that initial guess.

```
>>  [x0,err] = nrnd1(0.008)
```

```
icount = 1 xold = 8.000000e-03 f = -1.955300e+03 df = -1.211452e+07 xnew = 7.838599e-03  err = 1.000000e+02
icount = 2 xold = 7.838599e-03 f = 3.903320e+01 df = -1.260542e+07 xnew = 7.841695e-03  err = 3.948818e-04
icount = 3 xold = 7.841695e-03 f = 1.871688e-02 df = -1.259572e+07 xnew = 7.841697e-03  err = 1.894962e-07

x0 =   0.007841696626165

err =     1.894961977606645e-07
```

Therefore, the vapor root is 0.00784 moles/m^3.

Next, we find the liquid root, which is a little larger, say 10% larger, than the b parameter. First we calculate b.

```
>> R = 8.314; % J/mol/K
Tc = 154.6; % K
pc = 5.046e+6; % Pa
b = 0.07780*R*Tc/pc

b =   1.981763660721363e-05
```

Then we run nrnd1 with a guess 10% larger than b.

```
>>  [x0,err] = nrnd1(1.1*1.98e-05)

icount = 1 xold = 2.178000e-05 f = 2.292908e+08 df = -1.978558e+14 xnew = 2.293888e-05  err = 1.000000e+02
icount = 2 xold = 2.293888e-05 f = 9.255337e+07 df = -7.027407e+13 xnew = 2.425591e-05  err = 5.429746e-02
icount = 3 xold = 2.425591e-05 f = 3.173420e+07 df = -2.992303e+13 xnew = 2.531644e-05  err = 4.189086e-02
icount = 4 xold = 2.531644e-05 f = 7.798334e+06 df = -1.688450e+13 xnew = 2.577830e-05  err = 1.791675e-02
icount = 5 xold = 2.577830e-05 f = 8.614611e+05 df = -1.340424e+13 xnew = 2.584257e-05  err = 2.486897e-03
icount = 6 xold = 2.584257e-05 f = 1.599850e+04 df = -1.298885e+13 xnew = 2.584380e-05  err = 4.765980e-05
icount = 7 xold = 2.584380e-05 f = 5.309695e+01 df = -1.298103e+13 xnew = 2.584381e-05  err = 1.582719e-07

x0 =    2.584380714366121e-05

err =    1.582718742543172e-07
```

Therefore, the liquid root is 2.58x10^{-5} moles/m^3.

Next we find the intermediate root. This root must be between the liquid and vapor root. We can try 10*b.

```
>>  [x0,err] = nrnd1(10*1.98e-05)

icount = 1 xold = 1.980000e-04 f = 7.086039e+05 df = 9.471210e+09 xnew = 1.231834e-04  err = 1.000000e+02
icount = 2 xold = 1.231834e-04 f = -1.147320e+06 df = 5.361335e+10 xnew = 1.445833e-04  err = 1.480109e-01
icount = 3 xold = 1.445833e-04 f = -2.611509e+05 df = 3.149117e+10 xnew = 1.528761e-04  err = 5.424543e-02
icount = 4 xold = 1.528761e-04 f = -2.408423e+04 df = 2.590168e+10 xnew = 1.538059e-04  err = 6.045494e-03
icount = 5 xold = 1.538059e-04 f = -2.641220e+02 df = 2.534796e+10 xnew = 1.538164e-04  err = 6.774216e-05
icount = 6 xold = 1.538164e-04 f = -9.473805e-02 df = 2.534183e+10 xnew = 1.538164e-04  err = 2.430434e-08

x0 =   1.538163686174564e-04

err =     2.430434345736832e-08
```

Therefore, the intermediate root is 1.54x10^{-4} moles/m^3.

(b) T = 298.0 K

This temperature is above the critical temperature. We should find one root.

First, I change the input function in nrnd1.m. The only change is a change in temperature.

```
function f = funkeval(x)
V = x;
T = 298; % K
p = 101325; % Pa
R = 8.314; % J/mol/K
Tc = 154.6; % K
pc = 5.046e+6; % Pa
kappa = 0.4069;
alpha = (1 + kappa*(1 - sqrt(T/Tc)))^2;
b = 0.07780*R*Tc/pc;
a = 0.45724*R^2*Tc^2/pc*alpha;
f = R*T/(V-b) - a/(V*(V+b) + b*(V-b)) - p;
```

The ideal gas should give a good estimate of the root.

```
>> T = 298; % K
p = 101325; % Pa
R = 8.314; % J/mol/K
V = R*T/p

V =   0.024451734517641

>> [x0,err] = nrnd1(0.024)
icount = 1 xold = 2.400000e-02 f = 1.808531e+03 df = -4.293565e+06 xnew = 2.442122e-02  err = 1.000000e+02
icount = 2 xold = 2.442122e-02 f = 3.134197e+01 df = -4.146867e+06 xnew = 2.442878e-02  err = 3.093887e-04
icount = 3 xold = 2.442878e-02 f = 1.281518e-02 df = -4.144303e+06 xnew = 2.442878e-02  err = 1.265819e-07

x0 =   0.024428780172505

err =     1.265818784914305e-07
```

Therefore, the root is 0.0244 moles/m^3.

Problem 2.
Consider the van der Waals equation of state.

$$p = \frac{RT}{V-b} - \frac{a}{V^2}$$

Vapor-liquid equilibrium (VLE) at a given temperature, T, occurs at a specific pressure, the vapor pressure, p_{vap}. Therefore, we can consider the vapor pressure our unknown. However, you can't calculate the vapor pressure without knowing the molar volumes of the vapor, V_{vap}, and liquid, V_{liq}, phases. Therefore, in a standard VLE problem, we have three unknowns: p_{vap}, V_{vap} and V_{liq}. These three variables require three equations. The first equation is the relationship between p_{vap}, V_{vap} or V_{liq}., given by the equation of state,

$$f_1\left(p_{vap}, V_{vap}, V_{liq}\right) = \frac{RT}{V_{vap} - b} - \frac{a}{V_{vap}^2} - p_{vap} = 0$$

The second equation is the thermodynamic constraint establishing mechanical equilibrium,

$$f_2\left(p_{vap}, V_{vap}, V_{liq}\right) = p_{vap} - p_{liq} = \left(\frac{RT}{V_{vap} - b} - \frac{a}{V_{vap}^2}\right) - \left(\frac{RT}{V_{liq} - b} - \frac{a}{V_{liq}^2}\right) = 0$$

The third equation is the thermodynamic constraint establishing chemical equilibrium,

$$f_3\left(p_{vap}, V_{vap}, V_{liq}\right) = \mu_{vap} - \mu_{liq} = 0$$

where the chemical potential for a van der Waals gas is given by

$$\mu = -RT\left[\ln\left(\frac{V-b}{\Lambda^3}\right) - \frac{b}{V-b} + \frac{2a}{VRT}\right]$$

This expression for the chemical potential of a van der Waals gas introduces a new constant, the thermal de Broglie wavelength, Λ, but we don't need it because it drops out when we equate chemical potentials.

$$
\begin{aligned}
\mu_{liq} - \mu_{vap} &= -RT\left[\ln\left(\frac{V_{liq}-b}{\Lambda^3}\right) - \frac{b}{V_{liq}-b} + \frac{2a}{V_{liq}RT}\right] + RT\left[\ln\left(\frac{V_{vap}-b}{\Lambda^3}\right) - \frac{b}{V_{vap}-b} + \frac{2a}{V_{vap}RT}\right] \\
&= -RT\left[\ln\left(V_{liq}-b\right) - \frac{b}{V_{liq}-b} + \frac{2a}{V_{liq}RT}\right] + RT\left[\ln\left(V_{vap}-b\right) - \frac{b}{V_{vap}-b} + \frac{2a}{V_{vap}RT}\right]
\end{aligned}
$$

If we so choose, we could solve the VLE problem by finding the roots to these three equations. This requires good guesses for all three unknown variables.

Here is a better alternative solution approach. Solving a system of three equations and three unknowns is much harder than solving a system of equations with one equation and one unknown. Therefore, a useful trick in this problem is to recognize that the van der Waal's equation of state is a cubic equation of state and the roots of a cubic equation of state are easily obtained using the Matlab "roots" function.

$$
pV^3 - \left(pb + RT\right)V^2 + aV - ab = 0
$$

Therefore, we can pose the solution of the VLE problem of the van der Waals fluid as a single equation with a single equation, in which we guess the vapor pressure, use the roots command to solve for the V_{vap} and V_{liq}. and substitute them into the equation stating that the chemical potentials are equal.

So, in order to solve this problem as a single nonlinear algebraic equation with a single variable, p_{vap}, we have an equation,

$$
f\left(p_{vap}\right) = \mu_{liq} - \mu_{vap} = 0
$$

where we solve for the molar volumes each iteration and then substitute their values into the function above.

Find the vapor pressure of Argon at T=77 K. The van der Waals constants for argon are a=0.1381 m^6/mol^2 and b=3.184x10^{-5} m^3/mol. The gas constant is R=8.314 J/mol/K. Also report the liquid and vapor molar volumes.

Solution:

First we modify the input function in nrnd1.m

```
function f = funkeval(x)
p = x;
T = 77; % K
R = 8.314; % J/mol/K
a = 0.1381;
b = 3.184e-5;
Vvec = roots([p,-(p*b+R*T),a,-a*b]);
Vvap = max(Vvec);
Vliq = min(Vvec);
muliq = -R*T*(log(Vliq-b) - b/(Vliq-b) + 2*a/(Vliq*R*T));
muvap = -R*T*(log(Vvap-b) - b/(Vvap-b) + 2*a/(Vvap*R*T));
f = muliq - muvap;
```

Next we guess that the vapor pressure is 10 atm (1013250 Pa).

```
>> [x0,err] = nrnd1(1013250)
icount = 1 xold = 1.013250e+06 f = 1.031053e+03 df = 2.737563e-04 xnew = -2.158245e+06  err = 1.000000e+02
...
icount = 100 xold = 1.808793e+07 f = 3.215774e+03 df = -6.053046e-05 xnew = 1.808841e+07  err = 1.650221e+00
icount = 101 xold = 1.808841e+07 f = 3.215771e+03 df = -6.052910e-05 xnew = 1.808742e+07  err = 1.650262e+00
Sorry.  You did not converge in 100 iterations.
The final value of x was 1.808742e+07
```

```
x0 =      1.808741850893457e+07 - 2.641805411122832e+07i

err =   1.650261686195342
```

Well, 10 atm was a bad guess and did not converge. Let's try a new guess of 1 atm (101325 Pa).

```
>> [x0,err] = nrnd1(101325)
icount = 1 xold = 1.013250e+05 f = -1.836244e+02 df = 6.089856e-03 xnew = 1.314775e+05  err = 1.000000e+02
icount = 2 xold = 1.314775e+05 f = -2.376403e+01 df = 4.639125e-03 xnew = 1.366000e+05  err = 3.750018e-02
icount = 3 xold = 1.366000e+05 f = -4.744139e-01 df = 4.456224e-03 xnew = 1.367065e+05  err = 7.787557e-04
icount = 4 xold = 1.367065e+05 f = -1.946510e-04 df = 4.452568e-03 xnew = 1.367065e+05  err = 3.197840e-07

x0 =     1.367065379805641e+05

err = 3.197840381851619e-07
```

Well, 1 atm was a good guess and converged. The vapor pressure of vdW Argon at 77 K is 1.367×10^5 Pa.

The corresponding liquid and vapor molar volumes are

```
>> p = x0

p =     1.367065379805641e+05

>> T = 77; % K
R = 8.314; % J/mol/K
a = 0.1381;
b = 3.184e-5;
Vvec = roots([p,-(p*b+R*T),a,-a*b]);
Vvap = max(Vvec)
Vliq = min(Vvec)

Vvap =   0.004491379329435

Vliq =     3.881286647807421e-05
```

Therefore, the vapor root is 0.00449 moles/m^3 and the liquid root is 3.88×10^{-5} moles/m^3.

Problem 3.
Find the solution to the following system of nonlinear algebraic equations near (1,1,1).

$$f_1 = x_1 + 2x_2 + 3x_3 - 4$$
$$f_2 = x_1^3 - 4x_2^3$$
$$f_3 = x_3 - \sin x_3$$

Solution:

Since this is a system of three nonlinear algebraic equations with three unknowns, we can use the multivariate Newton-Raphson with numerical approximations to the derivatives. We will set a tolerance of 10^{-6}.
The input function looks like:

```
function f = funkeval(x)
%
%  these two lines force a column vector of length n
%
n = max(size(x));
f = zeros(n,1);
%
%  enter the functions here
%
f(1) = x(1) + 2*x(2) + 3*x(3) - 4;
f(2) = x(1)^3 - 4*x(2)^3;
f(3) = x(3) - sin(x(3));
```

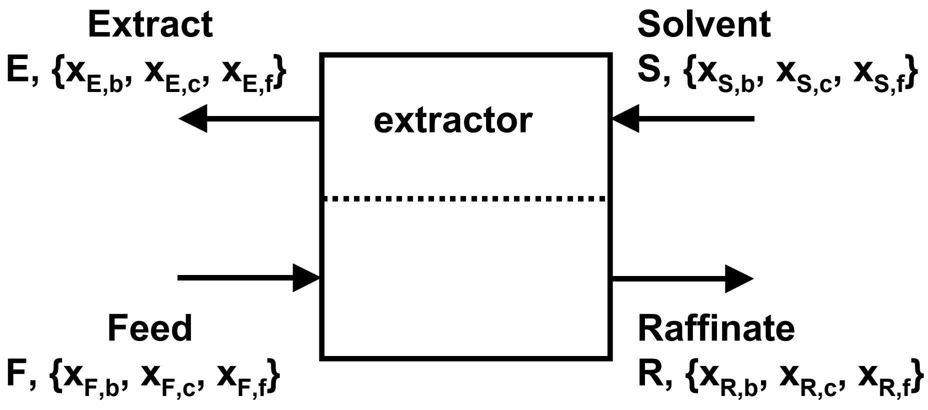
The command line prompt and output yields

```
>> [x,err,f] = nrndn([1,1,1],1.0e-6,1)
iter =    1, err =  3.28e-01 f =  2.08e+00
 iter =    2, err =  3.97e-01 f =  5.13e-01
 iter =    3, err =  1.50e-01 f =  6.63e-01
...
 iter =   31, err =  1.81e-06 f =  2.95e-10
 iter =   32, err =  1.21e-06 f =  1.97e-10
 iter =   33, err =  8.03e-07 f =  1.31e-10

x =   1.769971344010968   1.115012077004957   0.000001500659706

err =     8.031984760117370e-07

f =     1.310558041829265e-10
```

Therefore, the solution is $x_1 = 1.76997, x_2 = 1.11501 \text{and} \, x_3 = 0$.

Problem 4.
This problem requires steady state mass balances on a single-stage liquid-liquid extractor.

Consider an extractor:



In this system, a solvent stream, S, removes an impurity, b=benzene, from a feed stream, F. Exiting the extractor are two streams, the raffinate stream, R, which is the cleaned-up feed stream, and the extract stream, E, which now contains the impurity. This unit uses a recycled furfural stream as the solvent to extract benzene from a cyclohexane product stream. F, E, S and R are feed rates with units of moles per hour. The x variables are mole fractions. The mole fractions have two indices. The first uppercase letter indicates the stream and the second lowercase letter indicates the component. The data you are given is as follows.

$F = 100\, mol/hr$	$S = 150\, mol/hr$	$R = ?\, mol/hr$	$E = ?\, mol/hr$
$x_{F,b} = 0.1$	$x_{S,b} = 0.0010$	$x_{R,b} = ?$	$x_{E,b} = ?$
$x_{F,c} = 0.9$	$x_{S,c} = 0.0001$	$x_{R,c} = ?$	$x_{E,c} = ?$
$x_{F,f} = 0.0$	$x_{S,f} = 0.9989$	$x_{R,f} = ?$	$x_{E,f} = ?$

The equilibrium constants are: $K_b = \frac{x_{E,b}}{x_{R,b}} = 25.0$, $K_c = \frac{x_{E,c}}{x_{R,c}} = 0.04$ and $K_f = \frac{x_{E,f}}{x_{R,f}} = 50.0$.

You have eight unknowns, the flowrates of the raffinate and extract and the compositions of the raffinate stream and the composition of the extract stream.

(a) Write a set of eight independent algebraic equations describing this system.
(b) Is the set of equations linear or non-linear?
(c) Solve for the 8 unknowns. Clearly identify each variable.

For those unfamiliar with liquid-liquid extractors, the relative steady state mass balance has the form

traditional mole balance: $$accumulation = in - out + generation - consumption$$

In this system, since it is at steady state, there is no accumulation term. There is no chemical reaction so the generation and consumption terms are zero. Each component has two inputs and two outputs. Furthermore, the sum of the mole fractions in each stream must be zero.

Solution:

(a) Write a set of eight independent algebraic equations describing this system.

There are three independent mole balances. Any three of these four can be used.

benzene mole balance: $0 = Fx_{F,b} + Sx_{S,b} - Rx_{R,b} - Ex_{E,b}$

cyclohexane mole balance: $0 = Fx_{F,c} + Sx_{S,c} - Rx_{R,c} - Ex_{E,c}$

furfural mole balance: $0 = Fx_{F,f} + Sx_{S,f} - Rx_{R,f} - Ex_{E,f}$

total mole balance: $0 = F + S - R - E$

Furthermore, the sum of the mole fractions must be zero.

raffinate mole fraction constraint: $1 = x_{R,b} + x_{R,c} + x_{R,f}$

extract mole fraction constraint: $1 = x_{E,b} + x_{E,c} + x_{E,f}$

Finally, the equilibrium ratios can be rewritten as

equilibrium constants for benzene: $x_{R,b}K_b - x_{E,b} = 0$

equilibrium constants for cyclohexane: $x_{R,c}K_c - x_{E,c} = 0$

equilibrium constants for furfural: $x_{R,f}K_f - x_{E,f} = 0$

(b) Is the set of equations linear or non-linear?

The first three equations are nonlinear. Therefore the entire set of equations must be treated as nonlinear.

(c) Solve for the 8 unknowns. Clearly identify each variable.

I used the Newton-Raphson with Numerical Derivatives code, nrndn.m. I altered the input function as follows.

```
function f = funkeval(x)
%
%  these two lines force a column vector of length n
%
n = max(size(x));
f = zeros(n,1);
%
% identify variables
%
R = x(1);
E = x(2);
xRb = x(3);
xRc = x(4);
xRf = x(5);
xEb = x(6);
xEc = x(7);
xEf = x(8);
%
%  input parameters
%
%  input flow rates
F = 100.0; % mol/hr
```

```
S = 150.0; % mol/hr
% input compositions
xFb = 0.10;
xFc = 0.90;
xFf = 0.00;
xSb = 0.0010;
xSc = 0.0001;
xSf = 0.9989;
% equilibrium constants
Kb = 25.0;
Kc = 0.04;
Kf = 50.0;
%
% write 8 equations
%
f(1) = F*xFb + S*xSb - R*xRb - E*xEb;
f(2) = F*xFc + S*xSc - R*xRc - E*xEc;
f(3) = F*xFf + S*xSf - R*xRf - E*xEf;
f(4) = xRb + xRc + xRf - 1.0;
f(5) = xEb + xEc + xEf - 1.0;
f(6) = xRb*Kb - xEb;
f(7) = xRc*Kc - xEc;
f(8) = xRf*Kf - xEf;
```

The next challenge is to come up with good initial guesses. I used physically reasonable values. R = F. E = S. The compositions of R are close to those of F (but none are zero). The compositions of E are like S but enriched in b and c.

At the command line prompt, I typed

```
>> [x,err,f] = nrndn([100, 150, 0.01, 0.98, 0.01, 0.1, 0.1, 0.8],1.0e-6,1)
```

This generated the following output:

```
iter =    1, err =  7.86e+00 f =  1.32e+01
 iter =    2, err =  5.09e-01 f =  6.61e-01
 iter =    3, err =  5.93e-05 f =  1.79e-03
 iter =    4, err =  3.97e-11 f =  1.20e-09

x =   85.3039  164.6961   0.0024   0.9796   0.0180   0.0604   0.0392   0.9004

err =   3.9656e-11

f =   1.1967e-09
```

Therefore the program converged in four iterations. The solution is as follows

$$\begin{array}{llll}
F = 100\, mol/hr & S = 150\, mol/hr & R = 85.30\, mol/hr & E = 164.70\, mol/hr \\
\quad x_{F,b} = 0.1 & \quad x_{S,b} = 0.0010 & \quad x_{R,b} = 0.0024 & \quad x_{E,b} = 0.0604 \\
\quad x_{F,c} = 0.9 & \quad x_{S,c} = 0.0001 & \quad x_{R,c} = 0.9796 & \quad x_{E,c} = 0.0392 \\
\quad x_{F,f} = 0.0 & \quad x_{S,f} = 0.9989 & \quad x_{R,f} = 0.0180 & \quad x_{E,f} = 0.9004
\end{array}$$

Problem 5.
Consider a system with two anions, A and B, and two cations, C and H. The following reactions are possible

$$AH \overset{K_1}{\leftrightarrow} A^- + H^+ \qquad \text{reaction 1.}$$

$$AC \overset{K_2}{\leftrightarrow} A^- + C^+ \qquad \text{reaction 2.}$$

$$BH \overset{K_3}{\leftrightarrow} B^- + H^+ \qquad \text{reaction 3.}$$

$$BC \overset{K_4}{\leftrightarrow} B^- + C^+ \qquad \text{reaction 4.}$$

There are a total of eight species: *AH*, *AC*, *BH*, *BC*, A^-, B^-, H^+ and C^+. Potentially the composition of each of these eight species is unknown. In general, if we have eight unknowns, then we need eight equations to solve for a unique solution. Four of the equations come from thermodynamic equilibrium coefficients. Equilibrium coefficients govern the distribution of each of these reactions. Presumably the equilibrium coefficients are given.

$$K_1 = \frac{[A^-][H^+]}{[AH]} \quad \text{or} \quad [AH] = \frac{[A^-][H^+]}{K_1} \qquad \text{equation 1.}$$

$$K_2 = \frac{[A^-][C^+]}{[AC]} \quad \text{or} \quad [AC] = \frac{[A^-][C^+]}{K_2} \qquad \text{equation 2.}$$

$$K_3 = \frac{[B^-][H^+]}{[BH]} \quad \text{or} \quad [BH] = \frac{[B^-][H^+]}{K_3} \qquad \text{equation 3.}$$

$$K_4 = \frac{[B^-][C^+]}{[BC]} \quad \text{or} \quad [BC] = \frac{[B^-][C^+]}{K_4} \qquad \text{equation 4.}$$

Four more reactions come from molar balances on *A*, *B*, *C* and *H*.

$$[A_{Tot}] = [AH] + [AC] + [A^-] \qquad \text{equation 5.}$$

$$[B_{Tot}] = [BH] + [BC] + [B^-] \qquad \text{equation 6.}$$

$$[C_{Tot}] = [AC] + [BC] + [C^+] \qquad \text{equation 7.}$$

$$[H_{Tot}] = [AH] + [BH] + [H^+] \qquad \text{equation 8.}$$

Presumably the total amounts of each component added to the solution are given. Often the pH of the solution is given, which effectively provides the value of $[H^+]$, in which case we have one less unknown and equation 8 is redundant.

You could choose to solve this as 8 (or 7) equations with 8 (or 7) unknowns. Since the relationships in the equilibrium coefficients are so simple, one can alternatively eliminate some of the ions (*AH*, *AC*, *BH* and *BC*) from the molar balances. This leaves just four equations (the molar balances) (three if the pH is given) and four unknowns (A^-, B^-, C^+ and H^+.), (three if the pH is given).

Assume the pH is given. Solve for the concentration of the other seven ions as a function of pH from 1 to 7 for the following parameter values. Both the equilibrium coefficients and the concentrations have units of $\frac{mol}{\ell}$.

$K_1 = 10^{-1}$, $K_2 = 10^{-2}$, $K_3 = 10^{-3}$ and $K_4 = 10^{-4}$, $\left[A_{Tot}\right] = 1.0 \cdot 10^{-3}$, $\left[B_{Tot}\right] = 2.0 \cdot 10^{-5}$, and $\left[C_{Tot}\right] = 0.5 \cdot 10^{-3}$.

Solution:

To solve this problem I used the following script (based on the script in the notes).

script: paramstep_SolEqlb.m

```
%
%  parameter stepping example 3:  Solution Equilibria
%
clear all;
close all;
format longe;
% make current value of lambda available globally
global pHi
%
% set up pH vector
%
phlo =1.0;
phhi = 7.0;
dph = 0.001;
nph = (phhi - phlo)/dph + 1;
phvec = zeros(nph,1);
phvec=[phlo:dph:phhi]';
%
%  reserve memory for solution matrix
%
nvar = 3;
solmat = zeros(nph,nvar);
%
%  initial guess for low pH
%
Aguess = 1.0e-3;
Bguess = 1.0e-3;
Cguess = 1.0e-3;
x0=[Aguess; Bguess; Cguess];
% set the tolerance and printing switch
tol=1.0e-6;
iprint=0; % 0 is off, 1 is on
%
%  parameter step
%
for i =1:1:nph
% call Newton-Raphson
    pHi = phvec(i);
    [x,err,f] = nrndn_SolEqlb(x0,tol,iprint);
% store result
  solmat(i,1:nvar) = x(1:nvar);
  fmat(i) = f;
% update initial guess
  x0(1:nvar) = x(1:nvar);
end
```

```
%
%  plot result
%
figure(1)
semilogy(phvec(1:nph),solmat(1:nph,1),'k-');
hold on;
semilogy(phvec(1:nph),solmat(1:nph,2),'r-');
hold on;
semilogy(phvec(1:nph),solmat(1:nph,3),'b-');
hold off;
xlabel('pH');
ylabel('molarity');
legend('A','B','C')
%
%  plot error
%
figure(2)
semilogy(phvec(1:nph),fmat(1:nph),'k-');
xlabel('pH');
ylabel('average absolute error on f');

%
%  calculate other variables to Reproduce Figure 4 of reference
%
global K1 K2 K3 K4

solmat2 = zeros(nph,4);
for i = 1:1:nph
    H = 10.0^(-phvec(i));
%    OH = 10.0^(-(14.0-phvec(i)));
    solmat2(i,1) =solmat(i,1)*H/K1; %AH
    solmat2(i,2) =solmat(i,1)*solmat(i,3)/K2; %AC
    solmat2(i,3) =solmat(i,2)*H/K3; %BH
    solmat2(i,4) =solmat(i,2)*solmat(i,3)/K4; %BC
end

figure(3)
semilogy(phvec(1:nph),solmat2(1:nph,1),'k-');
hold on;
semilogy(phvec(1:nph),solmat2(1:nph,2),'r-');
hold on;
semilogy(phvec(1:nph),solmat2(1:nph,3),'b-');
hold on;
semilogy(phvec(1:nph),solmat2(1:nph,4),'g-');
xlabel('pH')
ylabel('Molarity')
legend('AH','AC','BH','BC')
hold off;
```

I also modified the `nrndn.m` input function as follows:

```
function f = funkeval(x)
%
% these two lines force a column vector of length n
%
n = max(size(x));
f = zeros(n,1);
%
% EDIT BELOW:
```

```
%
%global variables for post processing
%
global pHi
global K1 K2 K3 K4
%Setting input variables for solver
An = x(1);
Bn = x(2);
Cp = x(3);
%Known Molarities & Citrate ratio control
A_tot = 1.0e-3;
B_tot = 2.0e-5;
C_tot = 0.5e-3;
%H and OH calculator
pH = pHi;
%pOH= 14.0-pH;
Hp = 10.0^(-pH);
%OH = 10.0^(-pOH);
%equillibrium constants
K1 = 1.0e-1;
K2 = 1.0e-2;
K3 = 1.0e-3;
K4 = 1.0e-4;
% total amount of material
Atot = 1.0e-3;
Btot = 1.0e-5;
Ctot = 0.5e-3;
%  equilibrium relations
AH = An*Hp/K1;
AC = An*Cp/K2;
BH = Bn*Hp/K3;
BC = Bn*Cp/K4;
%Molar Balances
% left hand side
A_balance_lhs = Atot;
B_balance_lhs = Btot;
C_balance_lhs = Ctot;
% right hand side
A_balance_rhs = AH + AC + An;
B_balance_rhs = BH + BC + Bn;
C_balance_rhs = AC + BC  + Cp;
% Functions = RHS - LHS = 0
f(1) = A_balance_lhs - A_balance_rhs;
f(2) = B_balance_lhs - B_balance_rhs;
f(3) = C_balance_lhs - C_balance_rhs;
```

At the command line prompt, I executed the script:

```
>> paramstep_SolEqlb
```

which generated the following three plots.

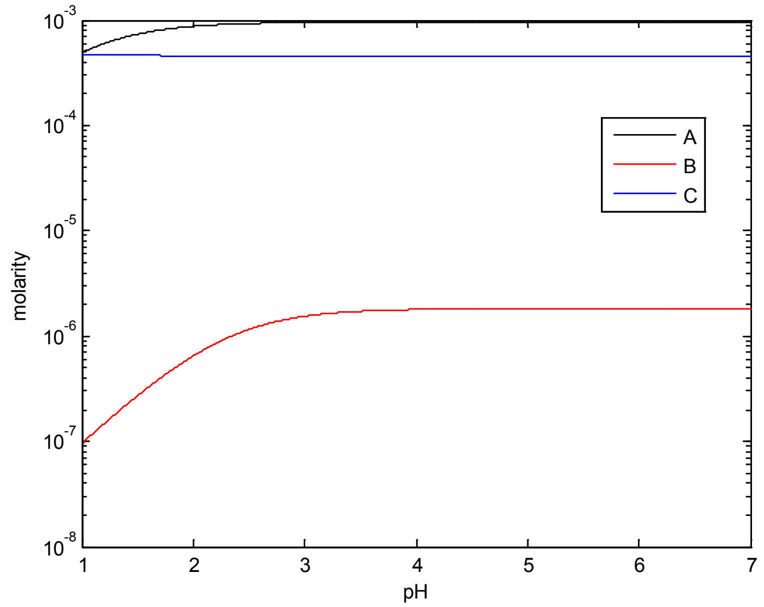


Figure 1. Concentrations of A^-, B^- and C^+ as a function of pH.

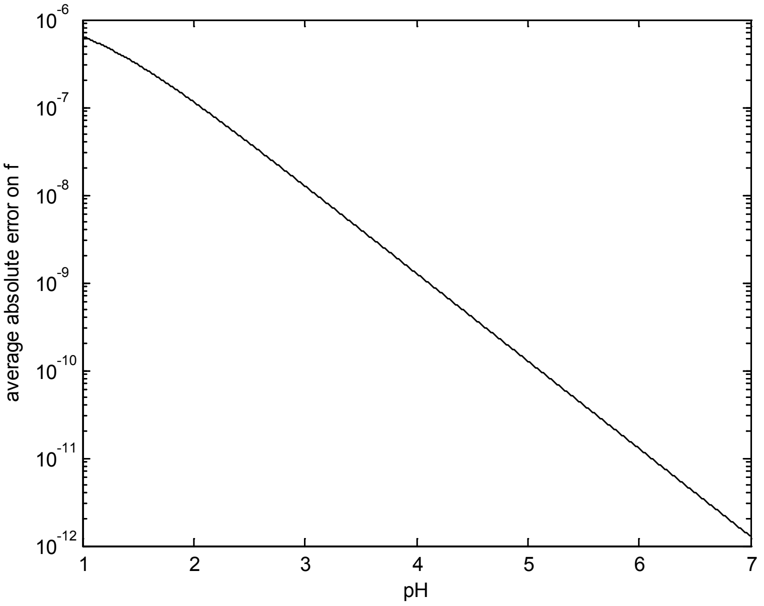


Figure 2. Error as a function of pH. This function shows that the tolerance was always satisfied.

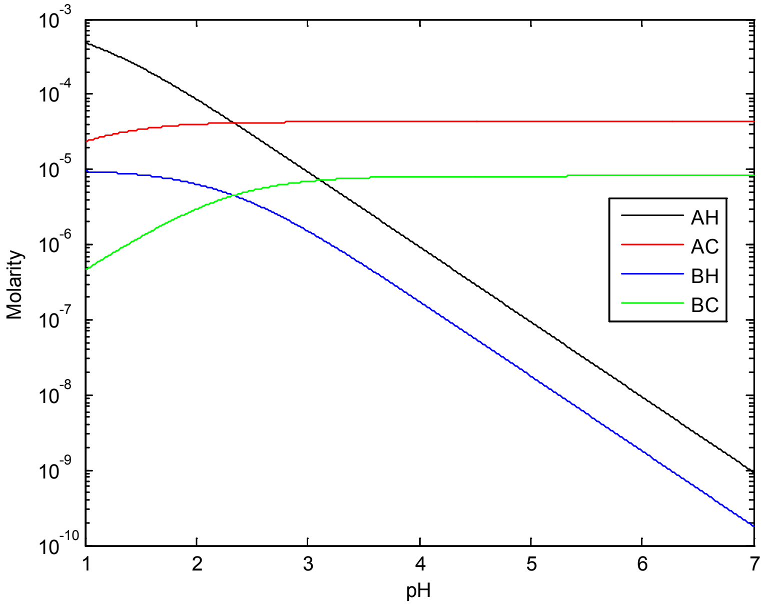


Figure 3. Concentration of *AH*, *AC*, *BH* and *BC* as a function of pH.