Exam III Solutions
Administered: Wednesday, November 10, 2017
16 points

For each problem part: 0 points if not attempted or no work shown,
1 point for partial credit, if work is shown,
2 points for correct numerical value of solution, if work is shown

Problem 1. (8 points)
For an ideal mixture, the volume of the mixture, V_{mix}, is given by the sum of the pure component molar volumes, V_i, weighted by the mole fraction, x_i. Similarly, the enthalpy of the mixture, H_{mix}, is given by the sum of the pure component enthalpies, H_i, weighted by the mole fraction. The cost of the mixture, C_{mix}, is also given by the sum of the pure component costs, C_i, weighted by the mole fraction. As a reminder, the sum of the mole fractions is unity.

$$V_{mix} = \sum_{i=1}^{n_c} x_i V_i \qquad H_{mix} = \sum_{i=1}^{n_c} x_i H_i \qquad C_{mix} = \sum_{i=1}^{n_c} x_i C_i \qquad 1 = \sum_{i=1}^{n_c} x_i$$

where n_c is the number of components. Now consider a four component ideal mixture ($n_c = 4$) with the following pure component properties and mixture properties.

component	A	B	C	D	mixture
molar volume (liter/mol)	11	14	17	8	12.8
enthalpy (kJ/mol)	49	74	45	63	57.1
cost ($/mol)	40.18	89.67	38.22	55.99	56.216

(a) Is this system of algebraic equations linear or nonlinear? (2 pts)
(b) Determine the composition of this mixture. Show reasoning and method. (6 pts)

Solution:

This is a set of linear algebraic equations, which can be written as follows.

$$x_A V_A + x_B V_B + x_C V_C + x_D V_D = V_{mix}$$
$$x_A H_A + x_B H_B + x_C H_C + x_D H_D = H_{mix}$$
$$x_A C_A + x_B C_B + x_C C_C + x_D C_D = C_{mix}$$
$$x_A + x_B + x_C + x_D = 1$$

This set of linear algebraic equations can be written in matrix form as

$$\underline{\underline{A}}\,\underline{x} = \underline{b}$$

where $\underline{\underline{A}} = \begin{bmatrix} V_A & V_B & V_C & V_D \\ H_A & H_B & H_C & H_D \\ C_A & C_B & C_C & C_D \\ 1 & 1 & 1 & 1 \end{bmatrix}$, $\underline{b} = \begin{bmatrix} V_{mix} \\ H_{mix} \\ C_{mix} \\ 1 \end{bmatrix}$ and $\underline{x} = \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix}$.

To solve this problem, I wrote the following Matlab script, `xm03p01_f17.m`.

```
clear all;
A = [11     14      17      8
     49     74      45      63
     40.18  89.67   38.22   55.99
     1      1       1       1];
b = [12.8
   57.1
   56.216
    1.0];
detA = det(A)
invA = inv(A)
x = inv(A)*b
```

At the command line prompt, I executed the script with the command

```
>> xm03p01_f17
```

which generated the following output

```
detA =  -1.2314e+03

invA =
   -0.3336   -0.3327    0.1681   14.2197
   -0.0291   -0.0723    0.0585    1.5112
    0.2417    0.1591   -0.0950   -6.6363
    0.1210    0.2459   -0.1316   -8.0946

x =
    0.4000
    0.3000
    0.2000
    0.1000
```

Therefore the composition of the mixture is given by

$$\underline{x} = \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.2 \\ 0.1 \end{bmatrix}$$

Problem 2. (8 points)
Consider a non-ideal mixture in which the enthalpy of the mixture is given by the expression

$$H_{mix} = \sum_{i=1}^{n_c} \sum_{j \geq i}^{n_c} \left(2 - \delta_{ij}\right) \Omega_{ij} x_i x_j$$

where δ_{ij} is the Kronecker delta function and is 1 for $i = j$ and 0 for $i \neq j$. For a four-component mixture, this equation becomes

$$\begin{aligned} H_{mix} = \Omega_{AA} x_A x_A + 2\Omega_{AB} x_A x_B + 2\Omega_{AC} x_A x_C + 2\Omega_{AD} x_A x_D \\ + \ \Omega_{BB} x_B x_B + 2\Omega_{BC} x_B x_C + 2\Omega_{BD} x_B x_D \\ + \ \Omega_{CC} x_C x_C + 2\Omega_{CD} x_C x_D \\ + \ \Omega_{DD} x_D x_D \end{aligned}$$

The expressions for the molar volume, cost and sum of the mole fractions remain unchanged from problem 1. The mixing parameters, $\Omega_{ij} = \sqrt{H_i H_j}$, where the pure component enthalpies are given in problem 1. The same numerical values of the mixture properties are observed as those in the problem 1.

(a) Is this system of algebraic equations linear or nonlinear? (2 pts)
(b) Determine the composition of this mixture. Show reasoning and method. (6 pts)

Solution:

This is a set of non-linear algebraic equations, which can be written as follows.

$$\begin{aligned} f_1(x_A, x_B, x_C, x_D) &= x_A V_A + x_B V_B + x_C V_C + x_D V_D - V_{mix} \\ f_2(x_A, x_B, x_C, x_D) &= x_A C_A + x_B C_B + x_C C_C + x_D C_D - C_{mix} \\ f_3(x_A, x_B, x_C, x_D) &= x_A + x_B + x_C + x_D - 1 \\ f_4(x_A, x_B, x_C, x_D) &= \Omega_{AA} x_A x_A + \Omega_{AB} x_A x_B + \Omega_{AC} x_A x_C + \Omega_{AD} x_A x_D \\ &\quad + \Omega_{BB} x_B x_B + \Omega_{BC} x_B x_C + \Omega_{BD} x_B x_D \\ &\quad + \Omega_{CC} x_C x_C + \Omega_{CD} x_C x_D + \Omega_{DD} x_D x_D - H_{mix} \end{aligned}$$

I will solve this using the Newton Raphson Method with Numerical Approximations to the Derivatives, as implemented in the code `nrndn.m`.

This code requires that I input my system of nonlinear algebraic equations in the function, `funkeval.m`.

```
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
%
```

```
VA = 11;
VB = 14;
VC = 17;
VD = 8;
Vmix = 12.8;
CA = 40.18;
CB = 89.67;
CC = 38.22;
CD = 55.99;
Cmix = 56.216;
HA = 49;
HB = 74;
HC = 45;
HD = 63;
omega_AA = sqrt(HA*HA);
omega_AB = sqrt(HA*HB);
omega_AC = sqrt(HA*HC);
omega_AD = sqrt(HA*HD);
omega_BB = sqrt(HB*HB);
omega_BC = sqrt(HB*HC);
omega_BD = sqrt(HB*HD);
omega_CC = sqrt(HC*HC);
omega_CD = sqrt(HC*HD);
omega_DD = sqrt(HD*HD);
Hmix = 57.1;
f(1) = VA*x(1) + VB*x(2) + VC*x(3) + VD*x(4) - Vmix;
f(2) = CA*x(1) + CB*x(2) + CC*x(3) + CD*x(4) - Cmix;
f(3) = x(1) + x(2) + x(3) + x(4) - 1.0;
f(4) = omega_AA*x(1)*x(1) + 2*omega_AB*x(1)*x(2) + 2*omega_AC*x(1)*x(3) + 2*omega_AD*x(1)*x(4) ...
                          +   omega_BB*x(2)*x(2) + 2*omega_BC*x(2)*x(3) + 2*omega_BD*x(2)*x(4) ...
                                                 +   omega_CC*x(3)*x(3) + 2*omega_CD*x(3)*x(4) ...
                                                                        +   omega_DD*x(4)*x(4) - Hmix;
```

The Newton Raphson method requires an initial guess. I will use the solution from problem 1 as my initial guess. I want the tolerance to be 1.0^{-6}. I set the print flag to 1. At the command line prompt, I executed the following commands:

```
>> x0 = [0.4; 0.3; 0.2; 0.1];
>> tol = 1.0e-6;
>> iprint = 1;
>> [x,err,f] = nrndn(x0,tol,iprint)
```

This command provided the following output.

```
 iter =    1, err =  1.33e-01 f =  3.04e-01
 iter =    2, err =  3.57e-04 f =  8.20e-04
 iter =    3, err =  2.56e-09 f =  5.89e-09

x =
    0.2032
    0.2573
    0.2941
    0.2455

err =   2.5631e-09

f =   5.8889e-09
```

Because the error is less than the specified tolerance, the Newton Raphson method has converged. Therefore the composition of the mixture is given by

$$\underline{x} = \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix} = \begin{bmatrix} 0.2032 \\ 0.2573 \\ 0.2941 \\ 0.2455 \end{bmatrix}$$

This solution is not very sensitive to the initial guess. All of the following initial guesses converged to this solution.

```
x0 = [0.4; 0.3; 0.2; 0.1];
x0 = [0.25; 0.25; 0.25; 0.25];
x0 = [0.7; 0.1; 0.1; 0.1];
x0 = [0.1; 0.7; 0.1; 0.1];
x0 = [0.1; 0.1; 0.7; 0.1];
x0 = [0.1; 0.1; 0.1; 0.7];
```