```
% Computation of methanol synthesis equilibrium composition
%
% Richard K. Herz, rherz@ucsd.edu, www.ReactorLab.net
%
% The set of Independent Stoichiometric Equations (ISE) used here
% ISE 1: CO + 2H2 = CH3OH
% ISE 2: CO2 + H2 = CO + H2O
%
% Equilibrium constant K's from Chang et al.
% Chang, T., Rousseau, R. W., and Kilpatrick, P. K.,
% Ind. Eng. Chem. Process Dev. 1986, 25, 477-481
%
% for T = 300 C and P = 5 Mpa = 50 bar

clear
clc
fprintf('------------ begin run ----------- \n')

% use of globals is lazy way to get values to functions
% better practice is to supply as function arguments...
global logK logKphi Y No P

% Y = complete stoichiometric matrix for these ISE's
% components in order: CO, CO2, H2, H2O, CH3OH, inert
Y = [-1 0 -2 0 1 0; 1 -1 -1 1 0 0]'; % note transpose (')

% No = initial moles
% components in order: CO, CO2, H2, H2O, CH3OH, inert
No = [15 8 74 0 0 3]';

P = 50; % bar, assume std P = 1 bar, 50 bar = 5 MPa
% standard state pressure not specified in Chang et al.
% but assume standard pressure = 1 bar
T = 300 + 273.15; % K

% get results that do NOT agree with Table 1 of Chang et al.
% for this logK(1) as written in Eqn 7 of Change et al.
% logK(1) = log(9.74e-5) + 21.225 + 9143.6/T - 7.492*log(T)...
%    + 4.076e-3*T - 7.161e-8*T^2;

% get results that DO agree with Table 1 of Chang et al. when
% delete 1st term in logK(1) since logK(2) in Chang doesn't have it
logK(1) = 21.225 + 9143.6/T - 7.492*log(T)...
    + 4.076e-3*T - 7.161e-8*T^2;

logK(2) = 13.148 - 5639.5/T - 1.077*log(T)...
    - 5.44e-4*T + 1.125e-7*T^2 + 49170/T^2;

% fugacity coefficient ratios from Figs. 3 & 4 of Chang et al. (1986)
logKphi(1) = log(0.90);
logKphi(2) = log(0.94);
```

```
% initial guesses of equilbrium stoichiometric extents
% note: zero value for e(1) gave bad results
e = [0.01 0]';

% use Matlab standard function fminsearch, which uses
% user-written function MeOHequilFun, to find
% equilbrium extents that minimize sum squared errors
% in equilibrium equations
e = fminsearch('MeOHequilFun',e);

N = No + Y*e; % final moles at equilibrium

% compare these results to those for 5 MPa, 300 C in
% Table 1 of Chang et al.
COconversion = 100 * (No(1) - N(1))/No(1)
CO_2_conversion = 100 * (No(2) - N(2))/No(2)
MethanolPercent = 100 * N(5)/sum(N)

fprintf('---------------- end run -------------- \n\n')

fprintf('-- listing of function MeOHequilfun --- \n\n')

type 'MeOHequilFun'

------------ begin run -----------

COconversion =

   24.4504


CO_2_conversion =

   13.3294


MethanolPercent =

    5.2290

---------------- end run --------------

-- listing of function MeOHequilfun ---


function sse = MeOHequilFun(e)
% input extent vector
% return sum of squared errors of equil functions

global logK logKphi Y No P

% components in order: CO, CO2, H2, H2O, CH3OH, inert
N = No + Y*e; % current number moles
```

```
if min(N) < 0
    % have negative moles at these extents
    % so assign large error
    sse = 1e6
else
    X = N/sum(N); % mole fractions

    f(1) = X(5) / ( X(1) * X(3)^2 * P^2 );
    f(2) = X(1) * X(4) / ( X(2) * X(3) );

    z = logK - log(f) - logKphi;

    sse = sum(z.^2);
end
```

*Published with MATLAB® R2016a*