The object of this column is to enhance our readers' collections of interesting and novel problems in chemical engineering. Problems of the type that can be used to motivate the student by presenting a particular principle in class, or in a new light, or that can be assigned as a novel home problem, are requested, as well as those that are more traditional in nature and that elucidate difficult concepts. Manuscripts should not exceed ten double-spaced pages if possible and should be accompanied by the originals of any figures or photographs. Please submit them to Professor James O. Wilkes (e-mail: wilkes@umich.edu), Chemical Engineering Department, University of Michigan, Ann Arbor, MI 48109-2136.

# AN EXERCISE FOR PRACTICING PROGRAMMING IN THE ChE CURRICULUM

## *Calculation of Thermodynamic Properties Using the Redlich-Kwong Equation of State*

MORDECHAI SHACHAM, NEIMA BRAUNER,[1] MICHAEL B. CUTLIP[2]
*Ben-Gurion University of the Negev • Beer-Sheva 84105, Israel*

Many students find it difficult to learn programming. One source of difficulty has to do with the complexity and relevance of the examples and exercises being used. Exercises that are simple enough for a student to write a working program in a reasonable length of time, without too much frustration, are often irrelevant to their chemical engineering studies. Consequently, they often do not see the benefit in learning programming and lose interest. More complex and realistic exercises, however, may require a long and frustrating debugging period, causing them to lose faith in their ability to make the program run and discouraging them from further programming attempts.

A good exercise to help students learn programming would be one of practical importance that can be constructed gradually in several steps. At each step, new types and more complex commands would be added to the program, but only after debugging of the previous step had been completed.

This paper presents such an exercise—one that involves analytical solution of the Redlich-Kwong equation for the compressibility factor and consequent calculation of molar volume, fugacity coefficient, isothermal enthalpy, and entropy departures. The solution is demonstrated using MATLAB,[1] but other programming languages (such as C or C++) can also be used.

***Mordechai Shacham*** *received his BSc (1969) and his DSc (1973) from Technion, Israel Institute of Technology. He is a professor of chemical engineering at the Ben-Gurion University of the Negev. His research interests include analysis, modeling, regression of data, applied numerical methods, computer-aided instruction, and process simulation, design, and optimization.*

***Neima Brauner*** *is professor and head of mechanical engineering undergraduate studies at Tel-Aviv University. She received her BSc and MSc in chemical engineering from the Technion Israel Institute of Technology, and her PhD in mechanical engineering from Tel-Aviv University. Her research has focused on the field of hydrodynamics and transport phenomena in two-phase flow systems.*

***Michael B. Cutlip*** *is a BS and MS graduate of The Ohio State University (1964) and a PhD graduate of the University of Colorado (1968), all in chemical engineering. He is coauthor with Mordechai Shacham of the POLYMATH software package and a recent textbook on numerical problem solving.*

[1] *Tel-Aviv University, Tel-Aviv 69978, Israel*
[2] *University of Connecticut, Storrs, CT 06269*

## Calculation of the Compressibility Factor and Derived Thermodynamic Properties Using the Redlich-Kwong Equation of State

The two-parameter Redlich-Kwong (R-K) equation of state has an accuracy that compares well with more complicated equations that incorporate many more constants (when applied to non-polar compounds[2]). The R-K equation is a cubic equation in the volume (or in the compressibility factor) for which analytical solutions can be found.[3] After solving for the molar volume (or compressibility factor), several important thermodynamic functions (such as fugacity coefficient, isothermal enthalpy, and entropy departures) can be calculated.

In this exercise, the molar volume, the compressibility factor, the isothermal enthalpy departure, the isothermal entropy departure, and the fugacity coefficients are calculated and plotted for water vapor in the supercritical region. The values of reduced pressure and reduced temperature used are shown in Table 1.

***Equations and Numerical Data***

The R-K equation is usually written[4]

$$P = \frac{RT}{V-b} - \frac{a}{V(V+b)\sqrt{T}} \qquad (1)$$

where

$$a = 0.42747\left(\frac{R^2T_c^{5/2}}{P_c}\right) \qquad (2)$$

$$b = 0.08664\left(\frac{RT_c}{P_c}\right) \qquad (3)$$

and

P pressure (atm)

**TABLE 1**
**Reduced Pressure and Reduced Temperature Values for Example 1**

| *Pr* | *Pr* | *Pr* | *Pr* | *Pr* | *Tr* |
|---|---|---|---|---|---|
| 0.1 | 2 | 4 | 6 | 8 | 1 |
| 0.2 | 2.2 | 4.2 | 6.2 | 8.2 | 1.05 |
| 0.4 | 2.4 | 4.4 | 6.4 | 8.4 | 1.1 |
| 0.6 | 2.6 | 4.6 | 6.6 | 8.6 | 1.15 |
| 0.8 | 2.8 | 4.8 | 6.8 | 8.8 | 1.2 |
| 1 | 3 | 5 | 7 | 9 | 1.3 |
| 1.2 | 3.2 | 5.2 | 7.2 | 9.2 | 1.5 |
| 1.4 | 3.4 | 5.4 | 7.4 | 9.4 | 1.7 |
| 1.6 | 3.6 | 5.6 | 7.6 | 9.6 | 2 |
| 1.8 | 3.8 | 5.8 | 7.8 | 9.8 | 3 |

***The exercise presented here enables students to start a programming assignment at a fairly simple level and to build it up gradually to a more complex assignment of practical importance...***

V molar volume (liters/g-mol)
T temperature (K)
R gas constant [R=0.08206 (atm·liter/g-mol·K)]
$T_c$ critical temperature (K)
$P_c$ critical pressure (atm)

Eliminating V from Eq. (1) and writing it as a cubic equation of the compressibility factor, z, yields

$$f(z) = z^3 - z^2 - qz - r = 0 \qquad (4)$$

where

$$r = A^2B \qquad (5)$$

$$q = B^2 + B - A^2 \qquad (6)$$

$$A^2 = 0.42747\left(\frac{P_R}{T_R^{5/2}}\right) \qquad (7)$$

$$B = 0.08664\left(\frac{P_R}{T_R}\right) \qquad (8)$$

in which $P_R$ is the reduced pressure ($P/P_c$) and $T_R$ is the reduced temperature ($T/T_c$).

Equation (4) can be solved analytically for three roots, some of which may be complex. Considering only the real roots, the sequence of calculations involves the steps

$$C = \left(\frac{f}{3}\right)^3 + \left(\frac{g}{2}\right)^2 \qquad (9)$$

where

$$f = \frac{-3q-1}{3} \qquad (10)$$

$$g = \frac{-27r-9q-2}{27} \qquad (11)$$

If C > 0, there is one real solution for z:

$$z = D + E + 1/3 \qquad (12)$$

where

$$D = \left(-\frac{g}{2} + \sqrt{C}\right)^{1/3} \qquad (13)$$

$$E = \left(-\frac{g}{2} - \sqrt{C}\right)^{1/3} \qquad (14)$$

If C < 0, there are three real solutions for z:

$$z_k = 2\sqrt{\frac{-f}{3}}\cos\left[\left(\frac{\phi}{3}\right) + \frac{2\pi(k-1)}{3}\right] + \frac{1}{3} \qquad k = 1,2,3 \quad (15)$$

where

$$\phi = a\cos\sqrt{\frac{g^2/4}{-f^3/27}} \qquad (16)$$

In the supercritical region, two of these solutions are negative, so the maximal $z_k$ is selected as the true compressibility factor.

After calculating the compressibility factor, the molar volume (V), the isothermal enthalpy departure ($\Delta H^*$), the isothermal entropy departure ($\Delta S^*$), and the fugacity coefficient ($\psi$) are calculated from[4]

$$V = \frac{zRT}{P} \qquad (17)$$

$$\frac{\Delta H^*}{RT} = \frac{3a}{2bRT^{3/2}}\ell n\left(1+\frac{b}{V}\right) - (z-1) \qquad (18)$$

$$\frac{\Delta S^*}{R} = \frac{a}{2bRT^{3/2}}\ell n\left(1+\frac{b}{V}\right) - \ell n\left(z-\frac{Pb}{RT}\right) \qquad (19)$$

$$\psi = \exp\left\{z-1-\ell n\left[z\left(1-\frac{b}{V}\right)\right]-\frac{a}{bRT^{3/2}}\ell n\left(1+\frac{b}{V}\right)\right\} \qquad (20)$$

The numerical data needed for solving this problem include R = 0.08206 liter·atm/g-mol·K, critical temperature for water $T_c$ = 647.4 K, and critical pressure of water $P_c$ = 218.3 atm.

***Recommended Steps for Solution***

1. Prepare a MATLAB m-file for solving the set of equations for Tr = 1.2 and Pr = 5 (C, in Eq. 9, is positive) and Tr = 10 and Pr = 5 (C, in Eq. 9, is negative). Compare the results obtained with values from generalized charts of thermodynamic properties.
2. Convert the program developed in part 1 to a function and write a main program to carry out the calculations for Pr = 5 and the set of Tr values shown in Table 1.
3. Extend the main program to carry out the calculations for all Pr and Tr values shown in Table 1. Store all the results of z, V, enthalphy and entropy departures, and fugacity coefficients in column vectors. Display the various variables versus Pr and Tr in tabular and graphic forms.

***Solution***

The MATLAB program (m-file) for solving the set of equations for one value of Tr and Pr and displaying the values of selected variables is shown in Figure 1. Preparation of the program requires that students rewrite the equations using the MATLAB syntax. This stage includes changing some variable names to valid MATLAB names, changing some algebraic operators, and changing some intrinsic function names (such as converting ln to log). The use of the "max" intrinsic function to select the maximal compressibility factor from the values obtained in Eq. (15) requires storing these values in a vector.

The equations must also be reordered according to a proper computational order (thus a variable is not used before a value is assigned to it). This can be most easily achieved by first entering

```
R = 0.08206; % Gas constant (L-atm/g-mol-K)
Tc = 647.4; % Critical temperature (K)
Pc = 218.3; % Critical pressure (atm)
a = 0.42747*R^2*Tc^(5/2)/Pc; % Eq. (2), RK equation constant
b = 0.08664*R*Tc/Pc; % Eq. (3),RK equation constant
Pr = 6; % Reduced pressure (dimensionless)
Tr = 1.2; % Reduced temperature (dimensionless)
Asqr = 0.42747*Pr/(Tr^2.5); % Eq. (7)
B = 0.08664*Pr/Tr; % Eq. (8)
r=Asqr*B; % Eq. (5)
q=B^2+B-Asqr; % Eq. (6)
f = (-3*q-1)/3; % Eq. (10)
g=(-27*r-9*q-2)/27; % Eq. (11)
C=(f/3)^3+(g/2)^2; % Eq. (9)
if (C>0)
   D=((-g/2+sqrt(C))^(1/3)); % Eq. (13)
   E1=(-g/2-sqrt(C)); % Eq. (14)
   E = ((sign(E1)*(abs(E1))^(1/3))); % Eq. (14)
   z = (D+E+1/3)    % Compressibility factor (dimensionless)Eq. (12)
else
   psii = (acos(sqrt((g^2/4)/(-f^3/27)))); % Eq. (16)
   zv(1) = (2*sqrt(-f/3)*cos((psii/3))+1/3) ; % Eq. (15)
   zv(2) = (2*sqrt(-f/3)*cos((psii/3)+2*3.1416*1/3)+1/3) ; % Eq. (15)
   zv(3) = (2*sqrt(-f/3)*cos((psii/3)+2*3.1416*2/3)+1/3) ; % Eq. (15)
   z = max(zv)  % Compressibility factor (dimensionless)
end
P = Pr*Pc   % Pressure (atm)
T = Tr*Tc   % Temperature (K)
V = z*R*T/P   % Eq. (17), Molar volume (L/g-mol)
Hdep = (3*a/(2*b*R*T^1.5))*log(1+b/V)-(z-1)
% Eq. (18), Enthalpy departure (dimensionless)
Sdep = (a/(2*b*R*T^1.5))*log((1+b/V))-log(z-P*b/(R*T))
%Eq. (19), Entropy departure (dimensionless)
f_coeff = exp(z-1-log(z*(1-b/V))-a/(b*R*T^1.5)*log(1+b/V))
 % Eq. (20), Fugacity cefficient (dimensionless)
```

***Figure 1.*** *MATLAB program for calculating compressibility factor and thermodyamic properties for one value of Re and Pr.*

**TABLE 2**
**Comparison of Calculated and Generalized Chart[5] Values for $P_r$ = 5**

| | *Tr = 1.2* | | *Tr = 10* | |
|---|---|---|---|---|
| | *Calc.* | *Chart* | *Calc.* | *Chart* |
| Compressibility factor | 0.7326 | 0.7 | 1.0373 | 1.0 |
| Enthalpy departure $\Delta H^*/T_c$(cal/g•mol K) | 6.0167 | 6.5 | -0.5515 | - |
| Entropy departure $\Delta S^*$ (cal/g•mol K) | 3.4616 | 4 | 0.0183 | - |
| Fugacity coefficient $\psi$ | 0.4579 | 0.47 | 1.0376 | 1.05 |

```
% A script file for calculating compressibility factor and derived
% thermodynamic  properties using the Redlich Kwong equation of state.
clear, clc, format compact, format short g
Tc = 647.4 ; % Critical temperature (K)
Pc = 218.3; % Critical pressure (atm)
Pr = 5; %  Reduced pressure (dimensionless)
Tr_list=[1 1.05 1.1 1.15 1.2 1.3 1.5 1.7 2 3];
for j=1:10
   Tr = Tr_list(j); %  Reduced temperature (dimensionless)
   [P,T,V,z,Hdep,Sdep,f_coeff]=RKfunc(Tc,Pc,Tr,Pr)
end
```

***Figure 2.*** *Main program for carrying out the calculations for one Pr and ten Tr values.*

```
Tr_list=[1 1.05 1.1 1.15 1.2 1.3 1.5 1.7 2 3];
Pr_list=[0.1 0.2];
i=2;
while (Pr_list(i)<9.9)
   i=i+1;
   Pr_list(i)=Pr_list(i-1)+0.2;
end
n_Tr=10;
n_Pr=size(Pr_list,2);
for i=1:n_Tr
   for j=1:n_Pr
      [P(j,i),T(j,i),V(j,i),z(j,i),Hdep(j,i),Sdep(j,i),f_coeff(j,i)]=RKfunc(Tc,Pc,Tr_list(i),Pr_list(j));
   end
end
%
% Print tabular results
%
for i=1:n_Tr
   disp(' ');
   disp(['Tr =' num2str(Tr_list(i)) '    T(K)=' num2str(Tr_list(i)*Tc)]);
   disp(' ');
   disp('      Pr      P(atm)    V(L/g-mol)     z       Hdep       Sdep       f_coeff');
   Res=[Pr_list(:) P(:,i) V(:,i) z(:,i) Hdep(:,i) Sdep(:,i) f_coeff(:,i)];
   disp(Res);
   pause
end
%
% Plot results
%
plot(Pr_list,z(:,1),'.',Pr_list,z(:,2),'-',Pr_list,z(:,3),':',Pr_list,z(:,4),'-.',Pr_list,z(:,5),'--',...
Pr_list,z(:,6),'*',Pr_list,z(:,7),'o',Pr_list,z(:,8),'+',Pr_list,z(:,9),'v',Pr_list,z(:,10),'^');
legend('Tr=1','Tr=1.05','Tr=1.1','Tr=1.15','Tr=1.2','Tr=1.3','Tr=1.5','Tr=1.7','Tr=2','Tr=3');
title(' Compressibility Factor Versus Tr and Pr')
xlabel('Reduced Pressure Pr');
ylabel('Compressibility Factor(z)');
pause
plot(Pr_list,f_coeff(:,1),'.',Pr_list,f_coeff(:,2),'-',Pr_list,f_coeff(:,3),':',Pr_list,f_coeff(:,4),...
'-.',Pr_list,f_coeff(:,5),'--',Pr_list,f_coeff(:,6),'*',Pr_list,f_coeff(:,7),'o',Pr_list,f_coeff(:,8),...
'+',Pr_list,f_coeff(:,9),'v',Pr_list,f_coeff(:,10),'^');
legend('Tr=1','Tr=1.05','Tr=1.1','Tr=1.15','Tr=1.2','Tr=1.3','Tr=1.5','Tr=1.7','Tr=2','Tr=3');
title(' Fugacity Coefficient Versus Tr and Pr')
xlabel('Reduced Pressure Pr');
ylabel('Fugacity Coefficient(f/P)');
pause
```

***Figure 3.*** *Part of the main program in its final form.*

the equations to a program that automatically reorders them (POLYMATH, for example). The ordered set of equations can then be pasted into the MATLAB editor. In addition to the "assignment" statements, this simple program requires only the "if" statement. No commands for printing the results are used, but selected variables are shown during the program execution by selective addition or removal of the semicolon from the ends of the commands. Good programming practice requires clear descriptions of the variables and the equations by adding comments.

The results obtained for compressibility factor, enthalpy and entropy departures, and fugacity coefficient by the MATLAB program are compared to values of generalized charts (Kyle[5]) in Table 2. The differences between the calculated values (presumed to be more accurate) and the generalized chart values are small enough to validate the correctness of the MATLAB program. For Tr = 10, no generalized chart values are available for enthalpy and entropy departure, but the calculated values match the trend observed in the generalized chart.

The principal change that has to be introduced in the program, when proceeding to the second step of the development, includes the addition of the function definition statement

```
function[P,T,V,z,Hdep,Sdep,f_coeff]=RKfunc(Tc,Pc,Tr,Pr)
```

and removal of the definition of the variables Tc, Pc, Tr, and Pr. The Tr and Pr are the parameters that are changed in the main program. Putting the definition of Tc and Pc in the main program enables easy modification of the program for different substances. All the variables that should be displayed in tabular or graphic form are included in the list of returned variables. The main program that calls this function in order to perform the calculations for Pr = 5 and the ten Tr values (shown in Table 1) is displayed in Figure 2.

The program starts with commands that are not specific to the problem at hand and fall into the category of "good programming practice." The workspace and the command window are cleared and the preferred format for printing is defined. The ten specified Tr values are stored in a row vector Tr_list and a "for"

statement is used to call the function while changing the parameter values. The results are displayed in a very rudimentary form, just by omitting the semicolon after the call to the function.

After verifying that this function works properly, the assignment can be finished by adding to the main program the set of Pr values shown in Table 1, storing the results, and displaying them in tabular and graphic forms. Part of the main program in its final form is shown in Figure 3.

In this program, a "while" statement is used to input the required Pr values into the row vector Pr_list. The intrinsic function "size" is used to determine the number of elements in Pr_list. The values returned from the function are stored in two-dimensional matrices, one column for each Tr and one row for each Pr value. Tables of results are printed for a constant Tr value, where the respective columns of the results matrices are united into a single matrix, "Res" which is displayed.

Only the code for plotting the compressibility factor and the fugacity coefficient is shown, and the additional variables can be plotted similarly. The plots of the compressibility factor versus Tr and Pr and the fugacity coefficient versus Tr and Pr are shown in Figures 4 and 5, respectively. These plots are almost identical to the generalized charts that can be found in the thermodynamics textbooks.

## CONCLUSION

The exercise presented here enables students to start a programming assignment at a fairly simple level and to build it up gradually to a more complex assignment of practical importance in chemical engineering. It demonstrates several aspects of good programming practice:

- *The use of comments to clearly describe equations and variables*
- *Clearing the workspace and command window before starting execution*
- *Proper ordering of the equations*
- *Modular construction of the program, where each module is tested separately before its integration with the other components*

A variety of the variable types (*i.e.*, scalar and matrix), intrinsic functions, and simple and complex commands are used. Thus, the exercise can cover a considerable portion of a programming course.

Because of the gradual increase of difficulty in building this program, most students can successfully complete it and thus gain confidence in their ability to write a "real" program. The outcome of the exercise, the set of diagrams that for many decades has been a very important component in all thermodynamic textbooks, provides an excellent demonstration of the importance of programming in chemical engineering.

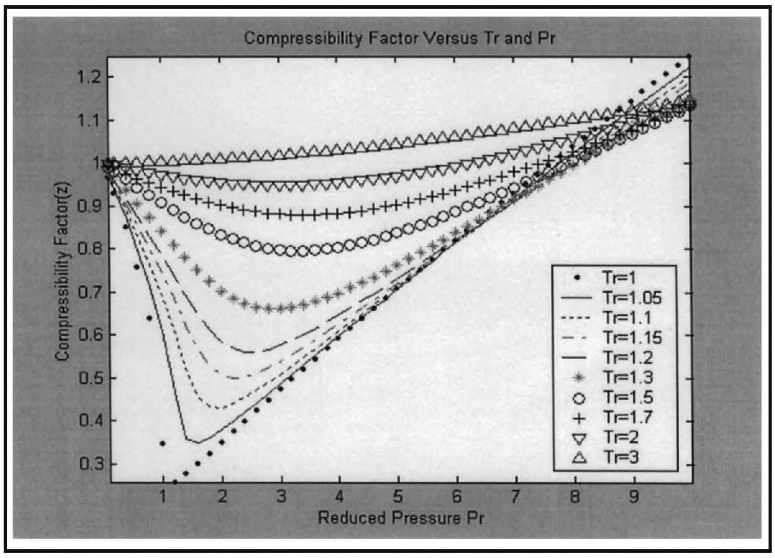


***Figure 4.*** *Plot of the compressibility factor versus reduced temperature and pressure.*

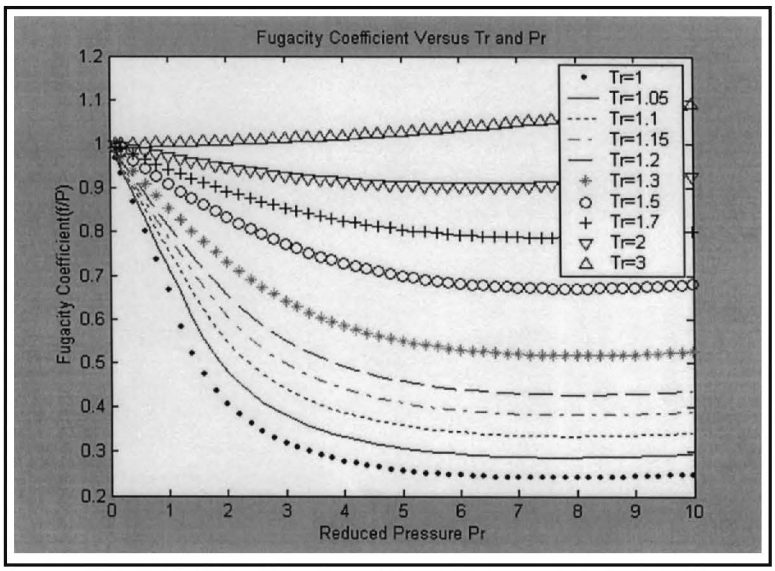


***Figure 5.*** *Plot of the fugacity coefficient versus reduced temperature and pressure.*

## REFERENCES


1. MATLAB is a trademark of The Math Works, Inc. <http://www.mathworks.com>
2. Seader, J.D., and E.J. Henley, *Separation Process Principles,* John Wiley & Sons, New York, NY, page 55 (1999)
3. Perry, R.H., C.H. Chilton, and S.D. Kirkpatrick, eds, *Perry's Chemical Engineers Handbook,* 4th ed., McGraw-Hill, New York, NY, pages 2-10 (1963)
4. Cutlip, M.B., and M. Shacham, *Problem Solving in Chemical Engineering with Numerical Methods,* Prentice-Hall, Upper Saddle River, NJ (1999)
5. Kyle, B.G., *Chemical and Process Thermodynamics,* 3rd ed., Prentice-Hall, Upper Saddle River, NJ (1999) ❐