Manchester University Press

It's not as easy as it looks: revisiting Peng–Robinson equation of state convergence issues for dew point, bubble point and flash calculations

Vamshi Krishna Kandula,[a] John C. Telotte[b] and F. Carl Knopf (corresponding author)[a]
[a] *Chemical Engineering Department, Louisiana State University, USA*
E-mail: knopf@lsu.edu
[b] *Chemical Engineering Department, Florida A&M University – Florida State University, USA*

Abstract The use of equations of state (EOS) is a 'modern' method of physical property prediction and here we present a detailed exposition of the use of the Peng–Robinson EOS for mixtures. This EOS can be used to determine whether a stream is in the vapor phase or liquid phase. However, for two-phase systems the approaches presented in standard textbooks for the calculation of the state frequently will not converge. We provide modifications to standard dew point, bubble point and flash algorithms to help improve the convergence of any EOS. Commercial software can show different values for stream energetic properties (entropy values) and here we utilize species formation from its elements as well as species ideal gas capacities and mixture departure functions to explain observed differences.

Keywords Peng–Robinson equation of state; cubic equations of state; cubic equations of state convergence issues; enthalpy and entropy determination from equations of state

Introduction

Because of the relative simplicity and reasonable accuracy of modern cubic equations of state, they are commonly used as both educational tools and for the simulation of chemical processes. Students in mechanical and chemical engineering use equations of state, especially the Peng–Robinson equation of state (PR-EOS), in their thermodynamics, separations and design courses. While it is possible to write closed-form expressions for many thermodynamic functions using cubic equations, problems often arise that are associated with numerical solutions. Several of the most commonly used thermodynamic texts supply equation-of-state software [e.g. 1, 2]. Also, most commercial process simulators have the option to use cubic equations of state and detailed sets of optimized parameters for application of these equations to multi-component systems.

In our classes we use both these 'canned algorithms' as well as assigned projects to perform calculations using cubic equations of state. For gas or liquid phase calculations, few challenges arise in the calculation of volumetric properties but energetic properties (entropy) can be difficult for students, as different commercial software packages give different values for identical systems. Even more challenging is that for two-phase systems the obvious approaches for calculation of the state frequently will not converge. Commercial codes such as Aspen Plus have addressed

International Journal of Mechanical Engineering Education, Volume 41, Number 3 (July 2013), © Manchester University Press
http://dx.doi.org/10.7227/IJMEE.41.3.2

this problem and have very robust algorithms. This is crucial, as every stream in a process simulator must have its state determined or defined for the calculation of stream enthalpy and entropy.

In this article we first revisit the PR-EOS to discuss issues surrounding the determination of the dew point and bubble point for mixtures. We will show a simpler and more robust convergence algorithm for EOS than those found in standard textbooks. We extend the use of this algorithm to flash and enthalpy and entropy calculations. All of the code and methods discussed here are freely available for download from our website, www.cogened.lsu.edu; all code is written in VBA with Excel used as the pre- and post-processor to promote student use.

This work began as a straightforward class project to find the dew point and bubble point temperature of a binary mixture using the PR-EOS. The project quickly expanded as the standard dew point and bubble point algorithm in texts often failed on simple systems or converged to incorrect solutions. Some software that is available for free often has difficulty converging, whereas commercial codes seem to be more robust. With the discussion and algorithms presented here, the PR-EOS can be used more effectively in class assignments. The applicability of the developed code to problems from the literature is also shown.

The PR-EOS for mixtures

The PR-EOS for mixtures [3] can be written as follows:

$$P = \frac{RT}{\upsilon - b_{\text{mix}}} - \frac{a_{\text{mix}}}{\upsilon(\upsilon + b_{\text{mix}}) + b_{\text{mix}}(\upsilon - b_{\text{mix}})} \tag{1}$$

where R is the universal gas constant, T is the absolute temperature and υ is the molar volume.

The mixture parameters can be determined in several ways. In this work we employ the standard van der Waals one-fluid mixing rule:

$$a_{\text{mix}} = \sum_{i=1}^{N\text{ species}} \sum_{j=1}^{N\text{ species}} y_i y_j \sqrt{a_i a_j}\,(1 - k_{ij})$$

$$b_{\text{mix}} = \sum_{i=1}^{N\text{ species}} y_i b_i$$

The pure component parameters are determined by imposing the critical conditions and fitting vapor pressure data:

$$a_i = 0.45724\left(\frac{R^2 T_{ci}^2}{P_{ci}}\right)\alpha_i$$

$$\alpha_i = \left(1 + \kappa_i\left(1 - \sqrt{\frac{T}{T_{ci}}}\right)\right)^2$$

k_{ij} is the binary interaction parameter; $k_{ij} = k_{ji}$ and $k_{ii} = 0$. To estimate binary interaction parameters for use with the PR-EOS see [4, 5].

$$\kappa_i = 0.37464 + 1.54226\omega_i - 0.26992\omega_i^2$$

where ω_i is the Pitzer acentric factor for species i (tabulated values are available).

$$b_i = 0.0778\frac{RT_{ci}}{P_{ci}}$$

T_{ci} is the critical temperature for species i and P_{ci} is the critical pressure for species i.

The PR-EOS is often written in a cubic polynomial form utilizing the compressibility factor $Z = \frac{P\upsilon}{RT}$:

$$Z^3 + (B_{\text{mix}} - 1)Z^2 + (A_{\text{mix}} - 3B_{\text{mix}}^2 - 2B_{\text{mix}})Z + (B_{\text{mix}}^3 + B_{\text{mix}}^2 - A_{\text{mix}}B_{\text{mix}}) = 0 \qquad (2)$$

where

$$A_{\text{mix}} = \frac{a_{\text{mix}}P}{R^2T^2}$$

$$B_{\text{mix}} = \frac{b_{\text{mix}}P}{RT}$$

This cubic equation (2) can be solved analytically [6], to give either one or three real roots. Care must be taken in analysis of these roots. At any temperature less than the critical value and for any pressure between the upper and lower spinodal pressures, there will be three real roots to the cubic equation. If there are three real roots, the largest will represent a vapor state, Z_V, and the smallest a liquid state, Z_L; the middle root is that of an unstable fluid and should always be neglected. However, it is only at a specified vapor pressure that Z_V and Z_L are both physical solutions. It is necessary to calculate the vapor pressure to be sure which root to use. If three roots exist and the pressure is greater than the saturation pressure then the liquid root is the proper physical solution. If the pressure is less than the vapor pressure then the vapor root is appropriate.

For mixture phase equilibrium calculations one always desires only one root at any time and software must have the appropriate logic to choose the proper form. If the cubic equation above has only one real root, then this is either a vapor or liquid value and which it is can be determined by comparison of the calculated Z with the value at the critical point.

Dew point and bubble point calculations using the PR-EOS

For students, as well as in commercial flowsheeting packages, an important stream calculation is determination of the dew point and bubble point temperatures. Here

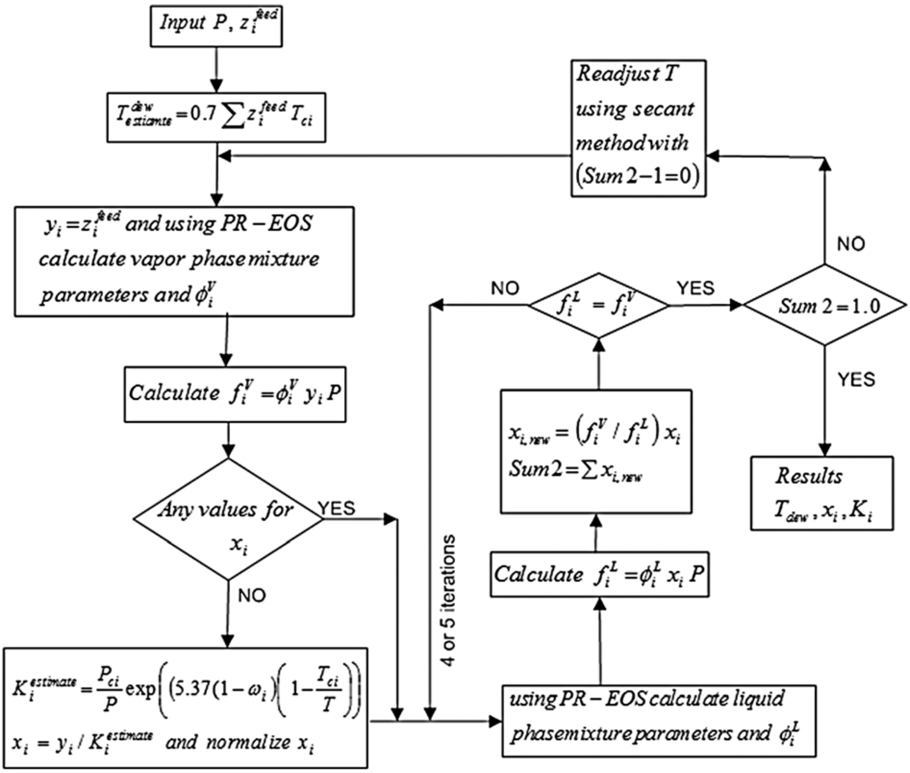
Input P, z_i^{feed}
$T_{estimate}^{dew} = 0.7\sum z_i^{feed} T_{ci}$
$y_i = z_i^{feed}$ and using PR – EOS calculate vapor phase mixture parameters and ϕ_i^V
Calculate $f_i^V = \phi_i^V y_i P$
Any values for x_i
YES
NO
$K_i^{estimate} = \frac{P_{ci}}{P}\exp\left((5.37(1-\omega_i)\left(1-\frac{T_{ci}}{T}\right)\right)$
$x_i = y_i / K_i^{estimate}$ and normalize x_i
4 or 5 iterations
using PR – EOS calculate liquid phase mixture parameters and ϕ_i^L
Calculate $f_i^L = \phi_i^L x_i P$
$x_{i,new} = (f_i^V / f_i^L) x_i$
$Sum2 = \sum x_{i,new}$
$f_i^L = f_i^V$
NO
YES
$Sum\,2 = 1.0$
NO
YES
Readjust T using secant method with ($Sum\,2 - 1 = 0$)
Results
T_{dew}, x_i, K_i

Fig. 1 *Standard algorithm for calculation of the dew point temperature.*

we assume the stream pressure is known – which is generally true in flowsheeting problems. The dew point temperature occurs when the first drop of liquid appears as a vapor mixture is cooled (at constant pressure). The bubble point temperature occurs when the first vapor bubble appears as a liquid mixture is heated (at constant pressure). If a stream's temperature is between the dew point and bubble point temperatures, an isothermal flash calculation must be performed in order to determine stream quality, enthalpy and entropy values.

Both bubble point and dew point calculations using equations of state are discussed in many texts. Flow charts for these calculations can be found in Sandler [2] and Van Ness and Abbott [7]. The dew point temperature algorithm presented in Fig. 1 is a variant of this type of approach. For the dew point calculation, the composition of the vapor mixture is known and fixed but the composition of the liquid droplet is unknown. At the dew point, the fugacity of each component in each phase are equal, $f^{\mathrm{V}}_{\mathrm{i}}= f^{\mathrm{L}}_{\mathrm{i}}$, and the sum of the mole fractions (x_i) in the liquid phase is 1 (Σx_i = 1). Fig. 1 provides the logic for the dew point code.

The vapor phase fugacity coefficient of a species in a mixture, ϕ^{V}_{i}, when using the PR-EOS, can be found [8] as:

$$\ln\phi_i^{\mathrm{V}} = \frac{b_i}{b_{\mathrm{mix}}}(Z_V - 1) - \ln\left(Z_{\mathrm{V}} - \frac{b_{\mathrm{mix}}P}{RT}\right) - \left(\frac{a_{\mathrm{mix}}}{2\sqrt{2}b_{\mathrm{mix}}RT}\right)\left(\frac{2\sum_{j=1}^{N\text{ species}} y_j\sqrt{a_i a_j}\,(1-k_{ij})}{a_{\mathrm{mix}}} - \frac{b_i}{b_{\mathrm{mix}}}\right)\ln\left(\frac{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1+\sqrt{2}\right)}{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1-\sqrt{2}\right)}\right) \tag{3}$$

For the liquid phase fugacity coefficient, y_i values would be replaced by x_i values and Z_{V} would be replaced by Z_{L}. For the dew point calculation, the composition of the vapor phase is known. As the algorithm shows, an initial guess for $K_i = \frac{y_i}{x_i}$ (distribution coefficient values) can be made and values for ϕ_i^{V} and ϕ_i^{L} determined. The temperature is changed to ensure $\sum x_i = 1$.

A caveat discussed in texts and papers is that often this method (Fig. 1) may fail to converge or may converge to the wrong solution (i.e. a trivial solution – where the vapor and liquid compositions and compressibility factors have the same values). Here, then, different, 'better' starting T_{dew} and K_i values should be used. However, the algorithm provided in Fig. 1 somewhat addresses this caveat by employing initial T_{dew} and K_i values as determined by Parekh and Mathias [9], with initial $T_{\mathrm{dew}} = 0.7\sum_i y_i T_{ci}$ and initial $K_i = \frac{P_{ci}}{P}\exp\left((5.37(1-\omega_i)\left(1-\frac{T_{ci}}{T}\right)\right)$; these starting values are commonly used in commercial flowsheeting packages.

Example 1 highlights the potential pitfalls of using the dew point algorithm (Fig. 1) to solve a simple binary hydrocarbon system at moderate pressure.

Example 1

Use the algorithm presented in Fig. 1 to determine the dew point and bubble point temperatures for ethane/n-heptane mixtures at 400 psia. For comparison, Table 1

TABLE 1 *Ethane (C2)/n-heptane system, P = 400 psia, all temperatures in ºF*

% C2	Experimental T_{bubble}	Experimental T_{dew}	Fig. 1 T_{bubble}	Fig. 1 T_{dew}	LSU-PR T_{bubble}	LSU-PR T_{dew}	Hysys T_{bubble}	Hysys T_{dew}	Sandler T_{bubble}	Sandler T_{dew}
10	435.5	501.5	524.5	189.9	442.5	487.3	442.5	487.3	N/C	N/C
20	322	468	322.7	159.7	322.7	465.3	322.7	465.3	322.5	N/C
30	232.5	445	222.8	−12.3	222.8	444.1	222.8	444.0	222.7	N/C
40	171	423	163.2	824.8	163.2	422.3	163.2	422.3	163.1	N/C
50	131.5	400	125.5	N/C	125.5	399.0	125.5	399.0	125.5	N/C
60	103	377.5	99.6	N/C	99.6	373.2	99.6	373.2	99.6	N/C
70	82	348	80.7	887.4	80.7	342.8	80.7	342.8	80.7	N/C
80	67	307.5	66.2	N/C	66.2	304.4	66.2	304.4	66.2	N/C
90	54	249	−43.2	N/C	54.3	247.6	54.3	247.6	54.4	N/C
100	43		−73.5	−77.2	42.5	42.5	42.5	42.5		

N/C = no convergence.

also provides experimental dew point and bubble point temperatures from the literature [10], results from the commercial PR code found in Hysys, results from our PR solution algorithm (LSU-PR), which will be discussed below (Figs 2–5), and results from the PR code provided in Sandler [2] (available from the student companion website for this text).

There are no results (T_{bubble} and T_{dew}) for pure n-heptane as at $P = 400$ psia we are above its critical pressure ($P_{c,\text{n-heptane}} = 396.93$ psia). The solution based on Fig. 1 gave unreasonable results for several bubble point and all dew point calculations, compared with the experimentally determined values. The LSU-PR algorithm (which will be described below) converged for all cases and gave reasonable results compared to experimental values for T_{bubble} and T_{dew}. The LSU-PR algorithm gave identical values for T_{bubble} and T_{dew} to those determined from the commercial PR code in Hysys. The Sandler code (PR with $k_{12} = 0.0158$) was similar to Hysys for T_{bubble} except it could not converge (N/C) for the 10% ethane mixture. The Sandler code would not converge for any of the T_{dew} calculations using the initial temperature guess built into the software. The Sandler code allows for a user-supplied initial temperature guess and with some effort most of the dew point states could be converged. Frequently, the initial guess had to be within 10°F of the dew point temperature and in some cases a T_{dew} guess within 5°F of the Hysys or experimental result was required for the code to converge. These results emphasize the need for an extremely good initial guess for either T_{bubble} or T_{dew} and the need to improve the convergence strategies shown in Fig. 1.

Modifications to the 'traditional' dew point and bubble point algorithms

There are several methods available to address convergence issues; however, these are often difficult for students to implement (see for example [11]). The NIST code Supertrapp [12] provides straightforward modifications to the standard dew point temperature algorithm (Fig. 1) to improve T_{bubble}, T_{dew} and flash calculations. An overview of the improved PR algorithm is provided in Fig. 2. The NIST code served as the basis for LSU-PR code provided here.

As indicated in Fig. 2 there are actually two major tasks (convergence strategies) which must be implemented to obtain accurate results from T_{bubble}, T_{dew} and flash calculations. First, an algorithm is used to develop very good initial guesses for both T_{bubble} and T_{dew}. These values will then be used as starting guesses in the PR algorithm to determine the final values for T_{bubble} or T_{dew}. Here (and in Figs 3 and 4) z_i^{feed} may be taken as either vapor at its dew point condition or a liquid at its bubble point condition.

Fig. 3 provides the actual convergence algorithm to find $T_{\text{initial}}^{\text{bub}}$ and $T_{\text{initial}}^{\text{dew}}$ (as indicated in the upper part of Fig. 2). Here, initial K_i values are determined independently of the PR-EOS [8]:

$$K_i = \frac{(P_{ci})^{\left[\left(\frac{1}{T_{\text{estimate}}^{\text{bub}}} - \frac{1}{T_i^{\text{NBP}}}\right) \Big/ \left(\frac{1}{T_{ci}} - \frac{1}{T_i^{\text{NBP}}}\right)\right]}}{P} \tag{4}$$

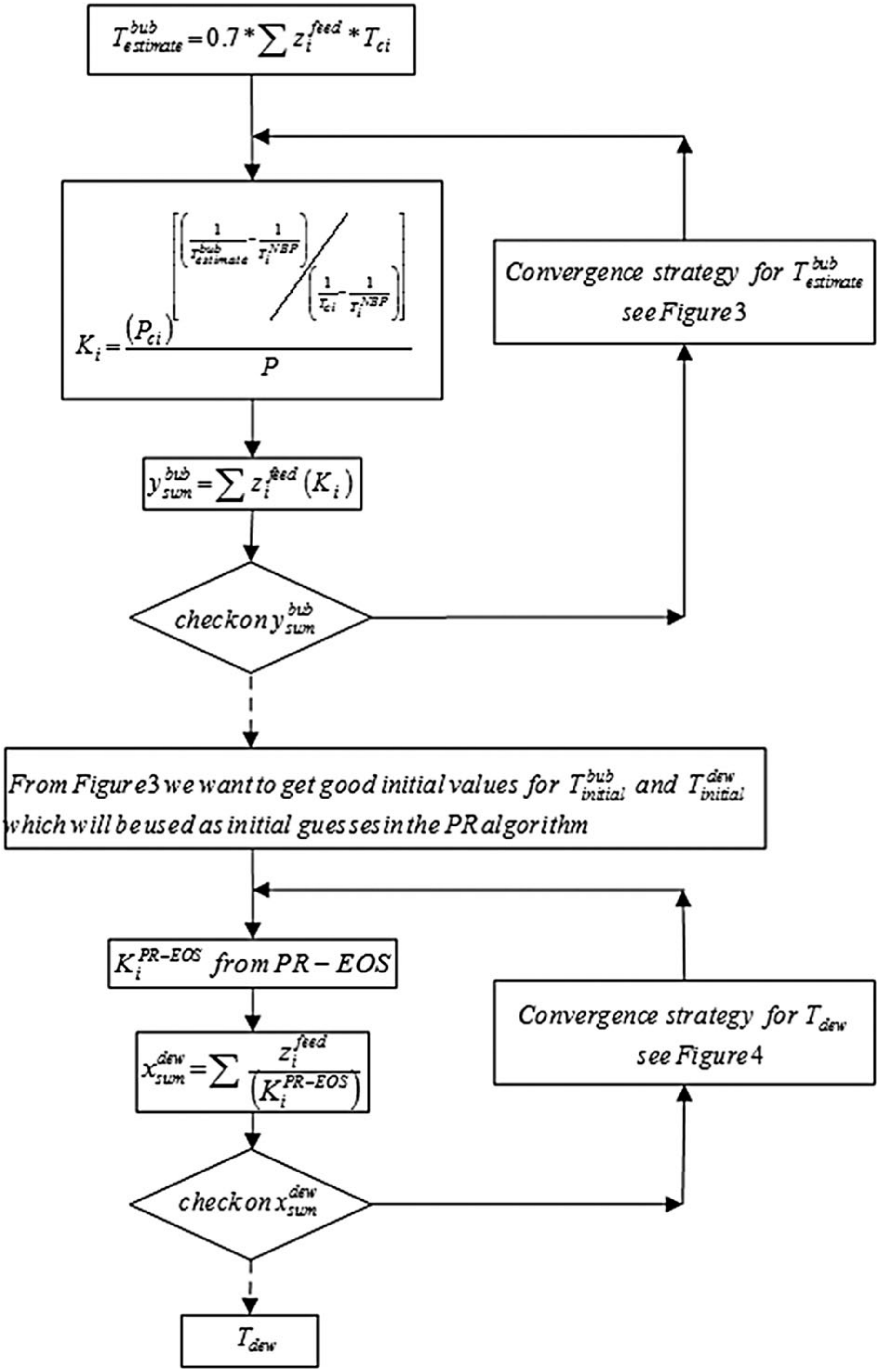
$T_{estimate}^{bub} = 0.7 * \sum z_i^{feed} * T_{ci}$
$K_i = \frac{(P_{ci})^{\left[\left(\frac{1}{T_{estimate}^{bub}} - \frac{1}{T_i^{NBP}}\right) / \left(\frac{1}{T_{ci}} - \frac{1}{T_i^{NBP}}\right)\right]}}{P}$
Convergence strategy for $T_{estimate}^{bub}$ see Figure 3
$y_{sum}^{bub} = \sum z_i^{feed}(K_i)$
check on y_{sum}^{bub}
From Figure 3 we want to get good initial values for $T_{initial}^{bub}$ and $T_{initial}^{dew}$ which will be used as initial guesses in the PR algorithm
K_i^{PR-EOS} from PR – EOS
Convergence strategy for T_{dew} see Figure 4
$x_{sum}^{dew} = \sum \frac{z_i^{feed}}{(K_i^{PR-EOS})}$
check on x_{sum}^{dew}
T_{dew}

Fig. 2 *Overview of an improved algorithm to determine T_{dew}.*

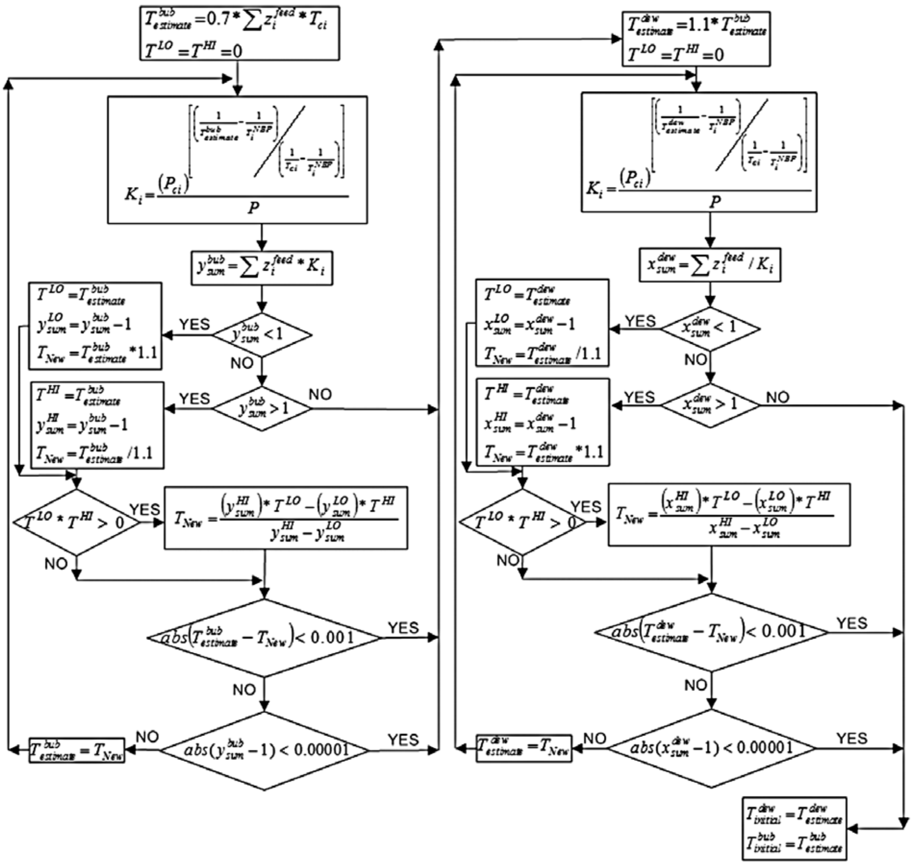
$T^{bub}_{estimate}=0.7*\sum z_i^{feed}*T_{ci}$
$T^{LO}=T^{HI}=0$
$K_i=\frac{(P_{ci})^{\left[\left(\frac{1}{T^{bub}_{estimate}}-\frac{1}{T_i^{NBP}}\right)/\left(\frac{1}{T_{ci}}-\frac{1}{T_i^{NBP}}\right)\right]}}{P}$
$y^{bub}_{sum}=\sum z_i^{feed}*K_i$
$y^{bub}_{sum}<1$
YES
NO
$T^{LO}=T^{bub}_{estimate}$
$y^{LO}_{sum}=y^{bub}_{sum}-1$
$T_{New}=T^{bub}_{estimate}*1.1$
$y^{bub}_{sum}>1$
YES
NO
$T^{HI}=T^{bub}_{estimate}$
$y^{HI}_{sum}=y^{bub}_{sum}-1$
$T_{New}=T^{bub}_{estimate}/1.1$
$T^{LO}*T^{HI}>0$
YES
NO
$T_{New}=\frac{(y^{HI}_{sum})*T^{LO}-(y^{LO}_{sum})*T^{HI}}{y^{HI}_{sum}-y^{LO}_{sum}}$
$abs(T^{bub}_{estimate}-T_{New})<0.001$
YES
NO
$abs(y^{bub}_{sum}-1)<0.00001$
YES
NO
$T^{bub}_{estimate}=T_{New}$
$T^{dew}_{estimate}=1.1*T^{bub}_{estimate}$
$T^{LO}=T^{HI}=0$
$K_i=\frac{(P_{ci})^{\left[\left(\frac{1}{T^{dew}_{estimate}}-\frac{1}{T_i^{NBP}}\right)/\left(\frac{1}{T_{ci}}-\frac{1}{T_i^{NBP}}\right)\right]}}{P}$
$x^{dew}_{sum}=\sum z_i^{feed}/K_i$
$x^{dew}_{sum}<1$
YES
NO
$T^{LO}=T^{dew}_{estimate}$
$x^{LO}_{sum}=x^{dew}_{sum}-1$
$T_{New}=T^{dew}_{estimate}/1.1$
$x^{dew}_{sum}>1$
YES
NO
$T^{HI}=T^{dew}_{estimate}$
$x^{HI}_{sum}=x^{dew}_{sum}-1$
$T_{New}=T^{dew}_{estimate}*1.1$
$T^{LO}*T^{HI}>0$
YES
NO
$T_{New}=\frac{(x^{HI}_{sum})*T^{LO}-(x^{LO}_{sum})*T^{HI}}{x^{HI}_{sum}-x^{LO}_{sum}}$
$abs(T^{dew}_{estimate}-T_{New})<0.001$
YES
NO
$abs(x^{dew}_{sum}-1)<0.00001$
YES
NO
$T^{dew}_{estimate}=T_{New}$
$T^{dew}_{initial}=T^{dew}_{estimate}$
$T^{bub}_{initial}=T^{bub}_{estimate}$

Fig. 3 *Algorithm to determine $T^{bub}_{initial}$ and $T^{dew}_{initial}$ for use in the PR algorithm.*

The algorithm in Fig. 3 may appear complex (when compared with that in Fig. 1) but it is straightforward to implement. In Fig. 3, the working algorithm is actually presented twice, for greater clarity – once for $T^{\mathrm{bub}}_{\mathrm{initial}}$ and once for $T^{\mathrm{dew}}_{\mathrm{initial}}$. The $T^{\mathrm{bub}}_{\mathrm{initial}}$ algorithm (the left-hand side of Fig. 3) consists of two loops: one to obtain a value for $T^{\mathrm{bub}}_{\mathrm{estimate}}$, where the $\Sigma y_i < 1$ which is $y^{\mathrm{LO}}_{\mathrm{sum}}$; and a loop where $\Sigma y_i > 1$, which is $y^{\mathrm{HI}}_{\mathrm{sum}}$. The temperature interval is then refined using a weighted difference technique and $T^{\mathrm{bub}}_{\mathrm{initial}}$ is determined. Values for $T^{\mathrm{bub}}_{\mathrm{initial}}$ and $T^{\mathrm{dew}}_{\mathrm{initial}}$ are then sent to the PR algorithm (as indicated in the lower part of Fig. 2). In Fig. 4 we provide the PR dew point algorithm.

In Fig. 4, a key addition is that K_i values are first determined using equation 4 and these values are retained as K_i^{old} throughout the algorithm. This allows the estimated x_i^{dew} values, termed x_i^{old}, to be unchanged at the start of each temperature iteration loop. The bounding algorithm for T^{HI} (corresponding to $x^{\mathrm{HI}}_{\mathrm{sum}}$) and T^{LO} (corresponding to $x^{\mathrm{LO}}_{\mathrm{sum}}$) is similar to our previous discussion (Fig. 3) but here smaller

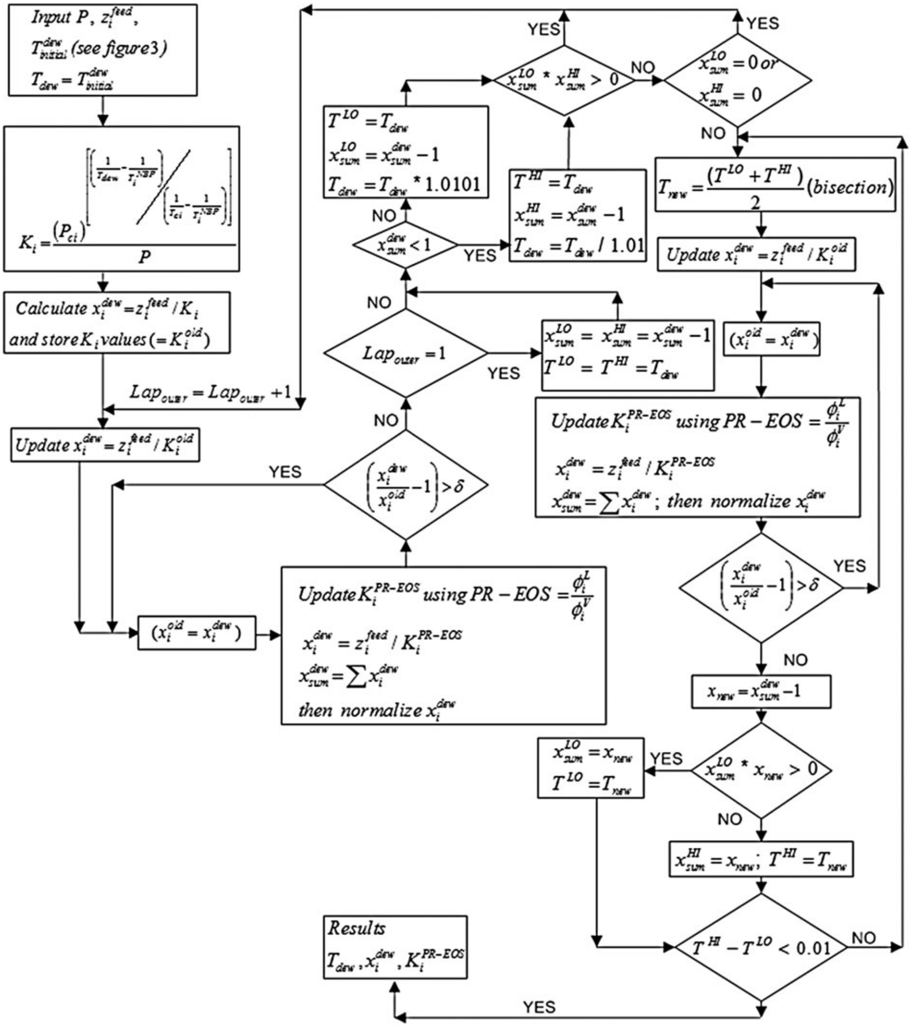
Input P, z_i^{feed}, $T_{initial}^{dew}$ (see figure3) $T_{dew} = T_{initial}^{dew}$
$K_i = \frac{(P_{ci})^{\left[\left(\frac{1}{T_{dew}} - \frac{1}{T_i^{NBP}}\right) / \left(\frac{1}{T_{ci}} - \frac{1}{T_i^{NBP}}\right)\right]}}{P}$
Calculate $x_i^{dew} = z_i^{feed} / K_i$ and store K_i values $(= K_i^{old})$
$Lap_{OUTER} = Lap_{OUTER} + 1$
Update $x_i^{dew} = z_i^{feed} / K_i^{old}$
YES
$(x_i^{old} = x_i^{dew})$
Update K_i^{PR-EOS} using $PR - EOS = \frac{\phi_i^L}{\phi_i^V}$ $x_i^{dew} = z_i^{feed} / K_i^{PR-EOS}$ $x_{sum}^{dew} = \sum x_i^{dew}$ then normalize x_i^{dew}
$\left(\frac{x_i^{dew}}{x_i^{old}} - 1\right) > \delta$
NO
$Lap_{outer} = 1$
YES
$x_{sum}^{LO} = x_{sum}^{HI} = x_{sum}^{dew} - 1$ $T^{LO} = T^{HI} = T_{dew}$
NO
$x_{sum}^{dew} < 1$
YES
NO
$T^{LO} = T_{dew}$ $x_{sum}^{LO} = x_{sum}^{dew} - 1$ $T_{dew} = T_{dew} * 1.0101$
$T^{HI} = T_{dew}$ $x_{sum}^{HI} = x_{sum}^{dew} - 1$ $T_{dew} = T_{dew} / 1.01$
$x_{sum}^{LO} * x_{sum}^{HI} > 0$
YES
NO
$x_{sum}^{LO} = 0$ or $x_{sum}^{HI} = 0$
YES
NO
$T_{new} = \frac{(T^{LO} + T^{HI})}{2}$ (bisection)
Update $x_i^{dew} = z_i^{feed} / K_i^{old}$
$(x_i^{old} = x_i^{dew})$
Update K_i^{PR-EOS} using $PR - EOS = \frac{\phi_i^L}{\phi_i^V}$ $x_i^{dew} = z_i^{feed} / K_i^{PR-EOS}$ $x_{sum}^{dew} = \sum x_i^{dew}$; then normalize x_i^{dew}
$\left(\frac{x_i^{dew}}{x_i^{old}} - 1\right) > \delta$
YES
NO
$x_{new} = x_{sum}^{dew} - 1$
$x_{sum}^{LO} * x_{new} > 0$
YES
$x_{sum}^{LO} = x_{new}$ $T^{LO} = T_{new}$
NO
$x_{sum}^{HI} = x_{new}$; $T^{HI} = T_{new}$
$T^{HI} - T^{LO} < 0.01$
NO
YES
Results T_{dew}, x_i^{dew}, K_i^{PR-EOS}

Fig. 4 *Improved bounding algorithm for PR T_{dew}.*

and different values for the temperature multipliers are used. Once bounded, a simple bisection method can be used to refine the temperature interval and determine T_{dew}.

Isothermal flash calculations using the PR-EOS

A stream at a given pressure, with temperature between its dew point temperature and bubble point temperature, will require a flash calculation in order to determine the stream quality. From the quality, the stream enthalpy and entropy and other state properties can be determined. This calculation is performed for every stream in a commercial process simulator.

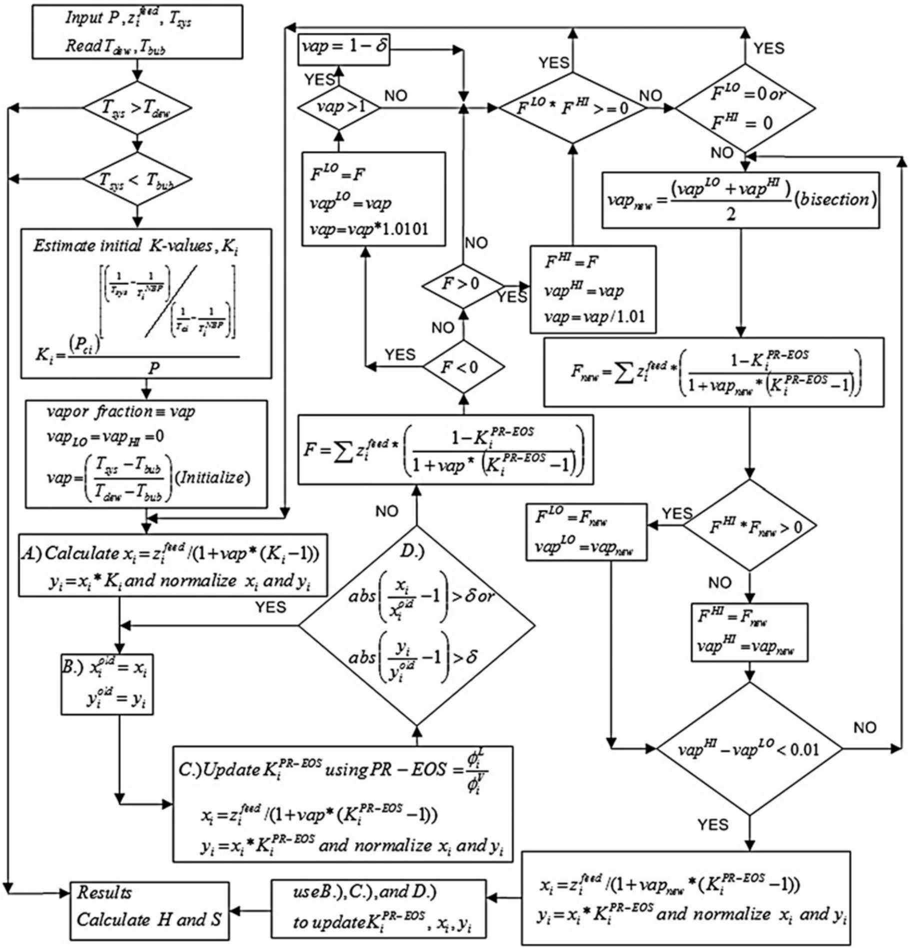
Input P, z_i^{feed}, T_{sys}
Read T_{dew}, T_{bub}
$T_{sys} > T_{dew}$
$T_{sys} < T_{bub}$
Estimate initial K-values, K_i
$K_i = \frac{(P_{ci})^{\left[\left(\frac{1}{T_{sys}} - \frac{1}{T_i^{NBP}}\right) / \left(\frac{1}{T_{ci}} - \frac{1}{T_i^{NBP}}\right)\right]}}{P}$
vapor fraction ≡ vap
$vap_{LO} = vap_{HI} = 0$
$vap = \left(\frac{T_{sys} - T_{bub}}{T_{dew} - T_{bub}}\right)$ (Initialize)
A.) Calculate $x_i = z_i^{feed}/(1 + vap*(K_i - 1))$
$y_i = x_i * K_i$ and normalize x_i and y_i
YES
B.) $x_i^{old} = x_i$
$y_i^{old} = y_i$
C.) Update K_i^{PR-EOS} using PR − EOS $= \frac{\phi_i^L}{\phi_i^V}$
$x_i = z_i^{feed}/(1 + vap*(K_i^{PR-EOS} - 1))$
$y_i = x_i * K_i^{PR-EOS}$ and normalize x_i and y_i
D.)
$abs\left(\frac{x_i}{x_i^{old}} - 1\right) > \delta$ or
$abs\left(\frac{y_i}{y_i^{old}} - 1\right) > \delta$
NO
$F = \sum z_i^{feed} * \left(\frac{1 - K_i^{PR-EOS}}{1 + vap*(K_i^{PR-EOS} - 1)}\right)$
$F < 0$
YES
NO
$F > 0$
YES
NO
$F^{LO} = F$
$vap^{LO} = vap$
$vap = vap*1.0101$
$vap > 1$
YES
NO
$vap = 1 - \delta$
$F^{HI} = F$
$vap^{HI} = vap$
$vap = vap/1.01$
$F^{LO} * F^{HI} >= 0$
YES
NO
$F^{LO} = 0$ or
$F^{HI} = 0$
YES
NO
$vap_{new} = \frac{(vap^{LO} + vap^{HI})}{2}$ (bisection)
$F_{new} = \sum z_i^{feed} * \left(\frac{1 - K_i^{PR-EOS}}{1 + vap_{new} * (K_i^{PR-EOS} - 1)}\right)$
$F^{HI} * F_{new} > 0$
YES
$F^{LO} = F_{new}$
$vap^{LO} = vap_{new}$
NO
$F^{HI} = F_{new}$
$vap^{HI} = vap_{new}$
$vap^{HI} - vap^{LO} < 0.01$
NO
YES
$x_i = z_i^{feed}/(1 + vap_{new} * (K_i^{PR-EOS} - 1))$
$y_i = x_i * K_i^{PR-EOS}$ and normalize x_i and y_i
use B.), C.), and D.)
to update K_i^{PR-EOS}, x_i, y_i
Results
Calculate H and S

Fig. 5 *Isothermal flash (inner loop A, B, C, D – K_i values are updating; outer loop – vapor fraction updating).*

The standard isothermal flash algorithm solves equation 5:

$$\sum \frac{z_i^{\text{feed}}(1 - K_i)}{1 + v(K_i - 1)} = 0 \tag{5}$$

where v is the unknown vapor fraction. Often in our undergraduate classes, to simplify flash calculations, we assume K_i values are constant. EOS allow for K_i values to change with changing composition. Therefore we will need a flash algorithm with an inner loop to update K_i values with changing composition and an outer loop to update the vapor fraction to solve equation 5. In Fig. 5 we provide a robust flash algorithm. In Fig. 5, as in Figs 2–4, a key is again the bounding and here the unknown vapor fraction is bounded.

Determination of enthalpy and entropy values for mixtures

Finally, we want to discuss enthalpy and entropy values as determined in commercial flowsheeting programs. This is can be a point of confusion for students, as ‘absolute’ values from commercial programs often do not match values found in look-up tables. Even more confusing is that entropy values from commercial programs differ.

The thermodynamic properties of a real gas mixture can be expressed in terms of the ideal gas contributions of individual species, a departure function (or residual property), which is the difference between properties of the real gas species and its ideal gas contribution, and a term accounting for the formation of each species from its elements.

The ideal gas enthalpy (h) contribution (J/mol) and entropy (s) contribution (J/mol-K) for each pure species can we can express as:

$$h_i^{\mathrm{IG}} = \int_{T_{\mathrm{ref}}}^{T} C_{P,i}^{\mathrm{IG}}\,\mathrm{d}T \tag{6}$$

$$s_i^{\mathrm{IG}} = \int_{T_{ref}}^{T} \frac{C_{P,i}^{\mathrm{IG}}}{T}\,\mathrm{d}T - R\ln\left(\frac{P}{P_{\mathrm{ref}}}\right) \tag{7}$$

The ideal gas mixture contributions are:

$$h_{\mathrm{mix}}^{\mathrm{IG}} = \sum y_i h_i^{\mathrm{IG}} \tag{8}$$

$$s_{\mathrm{mix}}^{\mathrm{IG}} = \sum y_i s_i^{\mathrm{IG}} - R\sum y_i \ln y_i \tag{9}$$

The general departure function or residual can be determined from an equation of state as:

$$h_{\mathrm{mix}}^{\mathrm{Depar}} = h_{\mathrm{mix}}^{\mathrm{EOS}} - h_{\mathrm{mix}}^{\mathrm{IG}} = RT\,(Z_{\mathrm{V}} - 1) + \int_{\infty}^{\upsilon}\left[T\left(\frac{\partial P}{\partial T}\right)_{\upsilon} - P\right]\mathrm{d}\upsilon \tag{10}$$

$$s_{\mathrm{mix}}^{\mathrm{Depar}} = s_{\mathrm{mix}}^{\mathrm{EOS}} - s_{\mathrm{mix}}^{\mathrm{IG}} = \int_{\infty}^{\upsilon}\left[\left(\frac{\partial P}{\partial T}\right)_{\upsilon} - \frac{R}{V}\right]\mathrm{d}\upsilon \tag{11}$$

and here specifically for the PR-EOS as:

$$h_{\mathrm{mix}}^{\mathrm{Depar}} = RT\,(Z_{\mathrm{V}} - 1) + \frac{T\left(\dfrac{\mathrm{d}a_{\mathrm{mix}}}{\mathrm{d}T}\right) - a_{\mathrm{mix}}}{2\sqrt{2}b_{\mathrm{mix}}}\ln\left(\frac{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1+\sqrt{2}\right)}{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1-\sqrt{2}\right)}\right) \tag{12}$$

$$s_{\mathrm{mix}}^{\mathrm{Depar}} = R\ln(Z_{\mathrm{V}} - B_{\mathrm{mix}}) + \frac{\left(\dfrac{\mathrm{d}a_{\mathrm{mix}}}{\mathrm{d}T}\right)}{2\sqrt{2}b_{\mathrm{mix}}}\ln\left(\frac{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1+\sqrt{2}\right)}{Z_{\mathrm{V}} + B_{\mathrm{mix}}\left(1-\sqrt{2}\right)}\right) \tag{13}$$

For the mixture, $\left(\dfrac{\mathrm{d}a_{\mathrm{mix}}}{\mathrm{d}T}\right)$ can be found [13]:

$$\left(\frac{\mathrm{d}a_{\mathrm{mix}}}{\mathrm{d}T}\right) = \frac{1}{2}\sum_{i=1}^{N\text{ species}}\ \sum_{j=1}^{N\text{ species}} y_i y_j (1-k_{ij})\left(\sqrt{\frac{a_j}{a_i}}\left(\frac{\mathrm{d}a_i}{\mathrm{d}T}\right) + \sqrt{\frac{a_i}{a_j}}\left(\frac{\mathrm{d}a_j}{\mathrm{d}T}\right)\right) \tag{14}$$

and

$$\frac{\mathrm{d}a_i}{\mathrm{d}T} = \frac{-\kappa_i a_i}{\sqrt{\alpha_i T T_{ci}}} \tag{15}$$

For a vapor system we can write the total molar enthalpy and entropy of the system as:

$$h_{\mathrm{mix}}^{\mathrm{real}} = h_{\mathrm{mix}}^{\mathrm{IG}} + h_{\mathrm{mix}}^{\mathrm{Depar}} + \sum y_i h_i^{\mathrm{ref}} \tag{16}$$

$$s_{\mathrm{mix}}^{\mathrm{real}} = s_{\mathrm{mix}}^{\mathrm{IG}} + s_{\mathrm{mix}}^{\mathrm{Depar}} + \sum y_i s_i^{\mathrm{ref}} \tag{17}$$

In the total enthalpy (equation 16), h_i^{ref} is the enthalpy of formation of each species in its reference phase from the elemental species at P_{ref} and T_{ref}, which are generally taken as 1 bar and 298.15 K. The reference phase used in these calculations is the ideal gas. The total entropy also includes an entropy of formation term, s_i^{ref}. For a liquid system, throughout all the equations provided here, y_i values would be replaced by x_i values and Z_{V} would be replaced by Z_{L}. It is important to appreciate that the departure functions ($h_{\mathrm{mix}}^{\mathrm{Depar}}$ and $s_{\mathrm{mix}}^{\mathrm{Depar}}$) will account for system non-ideality in either the vapor or the liquid state.

For a two-phase system, vapor and liquid properties are determined and the system volume, enthalpy and entropy are found based on quality q (which is the vapor fraction). For example, $s_{\mathrm{mix}}^{\mathrm{real}} = q\ (s_{\mathrm{mix}}^{\mathrm{real}})_{\mathrm{V}} + (1 - q)\ (s_{\mathrm{mix}}^{\mathrm{real}})_{\mathrm{L}}$; here, the additional subscripts V and L have been added to emphasize use of vapor phase and liquid phase values.

Example 2

Using the algorithm presented in Fig. 5, determine the phase compositions, enthalpy and entropy for the ethane/n-heptane feed mixtures shown in Table 2 at $P = 400$ psia and $T = 300$°F. Tables 2 and 3 provide dew point and bubble point temperatures from example 1 as well as stream vapor fractions, compositions and enthalpy and entropy

TABLE 2 *Ethane/n-heptane, P = 400 psia and T = 300°F, vapor fraction and phase composition*

		LSU-PR	LSU-PR	LSU-PR	LSU-PR	Hysys	Hysys	Hysys
z_{C2}	LSU-PR	T_{dew}	v_{frac}	x_{C2}	y_{C2}	v_{frac}	x_{C2}	y_{C2}
0.1	442.5	487.3	0.00	0.1		0.00	0.1	
0.2	322.7	465.3	0.00	0.2		0.00	0.2	
0.3	222.8	444.1	0.14	0.2191	0.8097	0.14	0.2191	0.8097
0.4	163.2	422.3	0.31	0.2191	0.8098	0.31	0.2191	0.8097
0.5	125.5	399.0	0.48	0.2192	0.8099	0.48	0.2191	0.8097
0.6	99.6	373.2	0.64	0.2192	0.8098	0.64	0.2191	0.8097
0.7	80.7	342.8	0.81	0.2192	0.8099	0.81	0.2191	0.8097
0.8	66.2	304.4	0.98	0.2185	0.8089	0.98	0.2191	0.8097
0.9	54.3	247.6	1.0		0.9	1.0		0.9
1.0	42.5	42.5	1.0		1.0	1.0		1.0

TABLE 3 *Ethane/n-heptane (400 psia, 300°F), h_{mix} = BTU/lb-mol, s_{mix} = BTU/lb-mol-F*

z_{C2}	LSU-PR h_{mix}	Hysys h_{mix}	Aspen h_{mix}	LSU-PRs_{mix}	Hysys s_{mix}	Aspen s_{mix}
0.0	−83 024	−83 090	−82 990	−160.30	43.84	−160.12
0.1	−78 235	−78 340	−78 218	−148.52	44.03	−148.36
0.2	−73 413	−73 550	−73 412	−136.98	43.99	−136.84
0.3	−68 145	−68 320	−68 172	−124.93	44.44	−124.84
0.4	−62 778	−62 990	−62 804	−112.76	45.02	−112.68
0.5	−57 411	−57 660	−57 437	−100.60	45.61	−100.52
0.6	−52 044	−52 330	−52 070	−88.42	46.19	−88.36
0.7	−46 677	−47 000	−46 703	−76.26	46.77	−76.19
0.8	−41 298	−41 670	−41 333	−64.07	47.35	−64.02
0.9	−37 162	−37 560	−37 191	−53.57	46.28	−53.53
1.0	−33 206	−33 640	−33 239	−43.59	44.67	−43.56

values from the LSU-PR algorithm, and results from the commercial PR codes found in Hysys and Aspen.

The values for the vapor fraction, vapor and liquid compositions and enthalpy values for the system, h_{mix}, show agreement. There are substantial differences in the calculated entropy for the system, s_{mix}, from these codes. These differences can be traced to the choice of reference states among the different programs but the LSU-PR and Aspen codes do provide the same entropy values. The Hysys routines use a different reference state for the entropies and the enthalpies. The entropies are set equal to zero for the pure species as ideal gases at 0 K and 1 atm pressure. Both this work and Aspen use the elemental species as references. This is very helpful for performing energy balances for reacting systems, as one does not need to enter the enthalpy-of-reaction data separately. For the LSU-PR code, the species entropy of formation is found from $G \equiv H - TS$, with H and G values taken from thermodynamic tables [14].

The code for the methods discussed here is freely available to download from our website, www.cogened.lsu.edu. The provided code contains some 20 species, which are oriented toward hydrocarbon and energy systems. The user can replace or expand the existing species set.

We do caution that the existing code does not check for multiple liquid phases. The formation of multiple liquid phases is not a common occurrence for mixtures of hydrocarbons and light gases, but can occur at elevated pressures. If one suspects that multiple liquid phases exist, then the liquid phase solution from the flash calculation must be checked for stability. If this phase is unstable then a three-phase flash calculation should be performed. That calculation can be accomplished using an approach similar to that shown here but is significantly more complicated.

Another problem that has been approached using the PR-EOS is the analysis of fuel gas mixtures containing small amounts of water [8]. An analysis of the dew point temperature of such mixtures must be approached with care, as these systems can have two dew points. If the first dew point is that associated with a nearly pure water liquid phase then the PR-EOS approach will not yield reliable results. The Hysys version of the PR code does not handle this properly but the Aspen version does. The second dew point will be calculated by the Hysys code and the present version.

It is also important to appreciate that the PR method in commercial programs is actually an evolved version of the original method we are discussing here (see for example [15]). Commercial codes make modifications for several species; for example, initial K_i values for hydrogen may be calculated using a modified version of equation 4. Most commercial codes now incorporate the Boston–Mathias [16] modification for the temperature dependence of the alpha term that is needed for high-temperature calculations. Finally, the code described here is based on the PR-EOS algorithm found in [17]; Supertrapp, available at nominal cost from NIST, offers additional convergence checks, starting guesses and physical property calculations.

Conclusions

It is important for students in engineering to have an appreciation and understanding of how thermodynamic databases are constructed and how thermodynamic properties are determined. The use of EOS and EOS mixing rules is a ‘modern’ method of physical property prediction and here we developed the PR-EOS for mixtures. We showed how this EOS can be used to determine whether a stream is in the vapor phase or liquid phase or is a two-phase mixture (bubble point and dew point temperature calculations). We provided modifications to standard textbook dew point, bubble point and flash algorithms to improve the convergence of any cubic EOS. These modifications are provided in flowcharts which can serve as the basis for a semester EOS project or faculty lectures on improving flash calculation convergence.

Calculation of stream enthalpy and entropy utilized species formation from its elements at a reference state (generally ideal gas at $T = 298.15$ K and $P = 1$ bar) as well as species ideal gas capacities and mixture departure functions. Commercial simulators generally produce ‘identical’ values for system enthalpy and vapor fraction but can show widely differing values for system entropy. This discrepancy was traced to some commercial simulators using different reference states for entropy and enthalpy calculations. We do make all developed code freely available from our website, www.cogened.lsu.edu. The code is written in VBA and attached to Excel – it is callable from any Excel spreadsheet, making it easy for student use. Contact the author for access to figures in this article.

Acknowledgements

This work was supported by a grant provided by the National Science Foundation: NSF – CCLI 0716303 (‘Phase II: Integrating a Cogeneration Facility into Engineering Education’). We appreciate the helpful discussions with Dr Marcia L. Huber at NIST concerning their code Supertrapp.

References

[1] B. G. Kyle, *Chemical and Process Thermodynamics*, 3rd edition (Prentice Hall, Upper Saddle River, NJ, 1999).

[2] S. I. Sandler, *Chemical and Engineering Thermodynamics*, 4th edition (Wiley, New York, 2006), p. 562.

[3] D. Y. Peng and D. B. Robinson, 'A new two-constant equation of state', *Ind. Eng. Chem. Fundam.*, **15** (1976), 59–64.

[4] H. Nishiumi, T. Arai and K. Takeuchi, 'Generalization of the binary interaction parameter of the Peng–Robinson equation of state by component family', *Fluid Phase Equilibria*, **42** (1988), 43–62.

[5] J. O. Valderrama and L. R. Reyes, 'Vapor–liquid–equilibrium of hydrogen-containing mixtures', *Fluid Phase Equilibria*, **13** (1983), 195–202.

[6] M. R. Spiegel and J. Liu, *Mathematical Handbook of Formulas and Tables*, 2nd edition (Schaum's Outline Series, McGraw Hill, New York, 1999), p. 10.

[7] H. C. Van Ness and M. M. Abbott, *Classical Thermodynamics of Nonelectrolyte Solutions – With Applications to Phase Equilibria* (McGraw Hill, New York, 1982), p. 284.

[8] F. G. Elliott, R. Kurz, C. Etheridge and J. P. O'Connell, 'Fuel system stability considerations for industrial gas turbines', *J. Enging. Gas Turbines Power*, **126** (January 2004), 119–126.

[9] V. S. Parekh and P. M. Mathias, 'Efficient flash calculations for chemical process design – extension of the Boston–Britt "inside-out" flash algorithm to extreme conditions and new flash types', *Computers Chemical Enging.*, **22**(10) (1998), 1371–1380.

[10] W. B. Kay, 'Liquid–vapor phase equilibrium relations in the ethane-n-heptane system', *Ind. Eng. Chem.*, **30**(4) (1938), 459–465.

[11] R. A. Heidemann and M. L. Michelsen, 'Instability of successive substitution', *Ind. Eng. Chem. Res.*, **34**(3) (1995), 958–966.

[12] D. G. Friend and M. L. Huber, 'Thermophysical property standard reference data from NIST', *Int. J. Thermophysics*, **15**(6) (1994), 1279–1288.

[13] R. M. Pratt, 'Thermodynamic properties involving derivatives using the Peng–Robinson equation of state', *Chemical Enging. Educ.*, **35**(2) (2001), 112–115.

[14] *TRC Thermodynamic Tables – Hydrocarbons* (Thermodynamic Research Center, Texas A&M University System, College Station, TX, 1983).

[15] E. Neau, O. Hernandez-Garduza, J. Escandell, C. Nicolas and I. Raspo, 'The Soave, Twu and Boston–Mathias alpha functions in cubic equations of state – Part 1. Theoretical analysis of their variations according to temperature', *Fluid Phase Equilibria*, **276** (2009), 87–93.

[16] J. F. Boston and P. M. Mathias, 'Phase equilibria in a third-generation process simulator', in *Proc. 2nd International Conference on Phase Equilibria and Fluid Properties in the Chemical Process Industries* (West Berlin, 1980), pp. 823–849.

[17] M. L. Huber, *NIST Thermophysical Properties of Hydrocarbon Mixtures Database (Supertrapp) Version 3.2 – User's Guide* (US Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, 2007).