

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@engin.umich.edu), Chemical Engineering Department, University of Michigan, Ann Arbor, MI 48109-2136.

USE OF AN INTEGRATION TECHNIQUE TO TRACE PHASE EQUILIBRIA CURVES

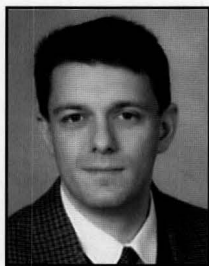
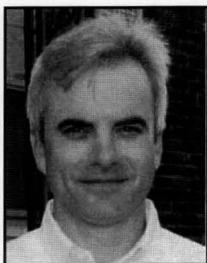
JAMES P. RUSSUM, DONALD P. VISCO, JR.
Tennessee Technological University • Cookeville, TN

The evaluation of phase equilibria from equations of state (EOS) is a classical problem traditionally taught in thermodynamics courses as part of the chemical engineering curriculum, both at the undergraduate and graduate levels. It can appear within a thermodynamics class or as a practical example in a numerical methods course.

The solution methodology normally taught (of the several available^[1]) for extracting coexistence information from an EOS is based on solving for unknown variables in the EOS, subject to the constraints of equilibrium. The technique is implemented through an iterative procedure^[2-3] or by using a multidimensional root-finding algorithm.

In this work, we introduce a novel method to solve this

Jim Russum graduated from Tennessee Technological University in May of 2001 with a BS in Chemical Engineering. He is currently enrolled in the graduate school of the Georgia Institute of Technology where he is pursuing a Ph.D. in Chemical Engineering. He is currently conducting research in catalysis and mini-emulsions.



Don Visco has held the position of Assistant Professor in Chemical Engineering at Tennessee Technological University since his graduation from the State University of New York at Buffalo in 1999. His research interests are in computational thermodynamics, bioinformatics and impulse characterization in granular media.

problem by writing a differential form for the equilibrium conditions that requires the numerical integration of these coupled differential equations to trace out the coexistence curve. For a simple example problem, we show that this method produces reasonable results around thirty times faster when compared to a root-finding algorithm. We also revisit a combined algorithm, discussed by both Asselineau, *et al.*,^[5] and Michelsen,^[6] that uses the best features of both approaches.

PROBLEM STATEMENT

Determine the coexistence curve for propane from $T = 200\text{K}$ to the predicted critical point. Use the Redlich-Kwong equation of state.^[7]

SOLUTION METHODOLOGY

Direct Method

The direct method involves writing the three equilibrium conditions,^[8] namely

$$T^l(P^l, \rho^l) = T^v(P^v, \rho^v) \quad (1)$$

$$P^l(T^l, \rho^l) = P^v(T^v, \rho^v) \quad (2)$$

$$\mu^l(T^l, \rho^l) = \mu^v(T^v, \rho^v) \quad (3)$$

where T is the temperature, P is the pressure, ρ is the density, and μ is the chemical potential, while the superscript “ l ” refers to liquid and “ v ” refers to vapor.

According to the Gibbs phase rule^[4] for this one-component, two-phase system, one independent intensive variable must be fixed in order to establish the intensive state of the system. Since the Redlich-Kwong equation of state is explicit in pressure, we fix the temperature (*i.e.*, $T^l = T^v$) as per the problem statement. From these specifications we are left with two coupled, nonlinear equations (Eqs. 2 and 3 above) and two unknowns (ρ^l and ρ^v). Here, a nonlinear root-finding algorithm (Newton-Raphson^[9] with a forward-difference Jacobian) is used to solve (root tolerance = 10^{-8} for convergence) for the two unknowns at the system specifications (fixed temperature). To reach the critical point, steps in the temperature are taken starting from 200K. There is a finite-sized temperature step, however, that one can take using this solution methodology, above which this method will not converge. For example, given the converged solution (liquid and vapor density) at 200K, if we step to 200.03K (*i.e.*, a step size of 0.03K) and use the converged solution at 200K as the guess into the Newton-Raphson routine for 200.03K, the method will fail (a converged solution will not be reached). But if we use a smaller step size (say, 0.02K), the routine runs without problem up to the critical point. For a step size of 0.02K, this method took about 1.5 seconds to run on a Dell Dimension (600 MHz, 256Kb RAM, DIGITAL Visual FORTRAN).

Integrate Method

Although the problem specification is for a two-phase, one-component mixture, we will introduce the integrate method for a two-phase, n -component mixture (keeping in mind that what follows is applicable for three-phase or higher equilibrium). We do this to show the compactness of the resulting expressions for a mixture, although the problem we are working through is for a pure component (propane).

We can write the chemical potential and pressure for each component as

$$\mu_i^l = \mu_i^l(T^l, \rho_1^l, \dots, \rho_n^l) \quad (4)$$

$$P^l = P^l(T^l, \rho_1^l, \dots, \rho_n^l) \quad (5)$$

$$\mu_i^v = \mu_i^v(T^v, \rho_1^v, \dots, \rho_n^v) \quad (6)$$

$$P^v = P^v(T^v, \rho_1^v, \dots, \rho_n^v) \quad (7)$$

where ρ_i indicates a component density of species i , while the differentials are

$$d\mu_i^l = \left(\frac{\partial \mu_i^l}{\partial T^l} \right)_{\rho^l} dT^l + \sum_{j=1}^n \left(\frac{\partial \mu_i^l}{\partial \rho_j^l} \right)_{T^l, \rho_k^l (k \neq j)} d\rho_j^l \quad (8)$$

$$dP^l = \left(\frac{\partial P^l}{\partial T^l} \right)_{\rho^l} dT^l + \sum_{i=1}^n \left(\frac{\partial P^l}{\partial \rho_i^l} \right)_{T^l, \rho_j^l (j \neq i)} d\rho_i^l \quad (9)$$

$$d\mu_i^v = \left(\frac{\partial \mu_i^v}{\partial T^v} \right)_{\rho^v} dT^v + \sum_{j=1}^n \left(\frac{\partial \mu_i^v}{\partial \rho_j^v} \right)_{T^v, \rho_k^v (k \neq j)} d\rho_j^v \quad (10)$$

$$dP^v = \left(\frac{\partial P^v}{\partial T^v} \right)_{\rho^v} dT^v + \sum_{i=1}^n \left(\frac{\partial P^v}{\partial \rho_i^v} \right)_{T^v, \rho_j^v (j \neq i)} d\rho_i^v \quad (11)$$

If our system is at equilibrium, then the following constraints exist regarding the chemical potential and pressure of the system, respectively,^[10]

$$d\mu_i^l = d\mu_i^v \quad (12)$$

$$dP^l = dP^v \quad (13)$$

We next divide each differential by an infinitesimal change in the temperature and take these partial derivatives constrained to a path that satisfies the equilibrium of chemical potential and pressure (the symbol σ indicates that the derivative is evaluated along that path).

$$\left(\frac{\partial \mu_i^l}{\partial T} \right)_{\sigma} = \left(\frac{\partial \mu_i^v}{\partial T} \right)_{\sigma} \quad (14)$$

$$\left(\frac{\partial P^l}{\partial T} \right)_{\sigma} = \left(\frac{\partial P^v}{\partial T} \right)_{\sigma} \quad (15)$$

From the Gibbs phase rule we are able to specify n independent variables in our system. We can fix the temperature and the $n-1$ independent mole fractions of the liquid phase to completely specify our system. Doing this, we can write the differential equations for the liquid phase as

$$\left(\frac{\partial \mu_i^l}{\partial T} \right)_{\sigma} = \left(\frac{\partial \mu_i^l}{\partial T^l} \right)_{\rho^l} \left(\frac{\partial T}{\partial T} \right)_{\sigma} + \sum_{j=1}^n \left(\frac{\partial \mu_i^l}{\partial \rho_j^l} \right)_{T^l, \rho_k^l (k \neq j)} x_j \left(\frac{\partial \rho_j^l}{\partial T} \right)_{\sigma} \quad (16)$$

$$\left(\frac{\partial P^l}{\partial T} \right)_{\sigma} = \left(\frac{\partial P^l}{\partial T^l} \right)_{\rho^l} \left(\frac{\partial T}{\partial T} \right)_{\sigma} + \sum_{i=1}^n \left(\frac{\partial P^l}{\partial \rho_i^l} \right)_{T^l, \rho_j^l (j \neq i)} x_j \left(\frac{\partial \rho_j^l}{\partial T} \right)_{\sigma} \quad (17)$$

For the vapor phase, we arrive at

$$\left(\frac{\partial \mu_i^v}{\partial T} \right)_{\sigma} = \left(\frac{\partial \mu_i^v}{\partial T^v} \right)_{\rho^v} \left(\frac{\partial T}{\partial T} \right)_{\sigma} + \sum_{j=1}^n \left(\frac{\partial \mu_i^v}{\partial \rho_j^v} \right)_{T^v, \rho_k^v (k \neq j)} \left(\frac{\partial \rho_j^v}{\partial T} \right)_{\sigma} \quad (18)$$

$$\left(\frac{\partial P^v}{\partial T}\right)_\sigma = \left(\frac{\partial P^v}{\partial T^v}\right)_{\rho^v} \left(\frac{\partial T}{\partial T}\right)_\sigma + \sum_{i=1}^n \left(\frac{\partial P^v}{\partial \rho_i^v}\right)_{T^v, \rho_j^v (j \neq i)} \left(\frac{\partial \rho_i^v}{\partial T}\right)_\sigma \quad (19)$$

where x_i are the mole fractions of component i in the liquid phase.

Substituting Eqs. (16) and (18) into Eq. (14) and noting that

$$\left(\frac{\partial T}{\partial T}\right)_\sigma = 1 \quad (20)$$

yields

$$\left(\frac{\partial \mu_i^l}{\partial T^l}\right)_{\rho^l} + \sum_{j=1}^n \left(\frac{\partial \mu_i^l}{\partial \rho_j^l}\right)_{T^l, \rho_k^l (k \neq j)} x_j \left(\frac{\partial \rho_j^l}{\partial T}\right)_\sigma = \left(\frac{\partial \mu_i^v}{\partial T^v}\right)_{\rho^v} + \sum_{j=1}^n \left(\frac{\partial \mu_i^v}{\partial \rho_j^v}\right)_{T^v, \rho_k^v (k \neq j)} \left(\frac{\partial \rho_j^v}{\partial T}\right)_\sigma \quad (21)$$

or

$$\left(\frac{\partial \mu_i^l}{\partial T^l}\right)_{\rho^l} - \left(\frac{\partial \mu_i^v}{\partial T^v}\right)_{\rho^v} = \sum_{j=1}^n \left(\frac{\partial \mu_i^v}{\partial \rho_j^v}\right)_{T^v, \rho_k^v (k \neq j)} \left(\frac{\partial \rho_j^v}{\partial T}\right)_\sigma - \sum_{j=1}^n \left(\frac{\partial \mu_i^l}{\partial \rho_j^l}\right)_{T^l, \rho_k^l (k \neq j)} x_j \left(\frac{\partial \rho_j^l}{\partial T}\right)_\sigma \quad (22)$$

and, similarly for the pressure

$$\left(\frac{\partial \rho^l}{\partial T^l}\right)_{\rho^l} - \left(\frac{\partial P^v}{\partial T^v}\right)_{\rho^v} = \sum_{i=1}^n \left(\frac{\partial P^v}{\partial \rho_i^v}\right)_{T^v, \rho_j^v (j \neq i)} \left(\frac{\partial \rho_i^v}{\partial T}\right)_\sigma - \sum_{i=1}^n \left(\frac{\partial P^l}{\partial \rho_i^l}\right)_{T^l, \rho_j^l (j \neq i)} x_i \left(\frac{\partial \rho_i^l}{\partial T}\right)_\sigma \quad (23)$$

Recognizing that Eqs. (22) and (23) can be written in linear form, we can use matrix notation to compactly represent the system of equations in terms of a coefficient matrix (**A**), a solution vector (**b**), and an unknown vector (**x**); **Ax = b**.

$$\begin{pmatrix} -\sum_{j=1}^n x_j \left(\frac{\partial \mu_i^l}{\partial \rho_j^l}\right)_{T^l, \rho_k^l (k \neq j)} & \left(\frac{\partial \mu_i^v}{\partial \rho_1^v}\right)_{T^v, \rho_k^v (k \neq 1)} & \dots & \left(\frac{\partial \mu_i^v}{\partial \rho_n^v}\right)_{T^v, \rho_k^v (k \neq n)} \\ \vdots & \vdots & \vdots & \vdots \\ -\sum_{j=1}^n x_j \left(\frac{\partial \mu_n^l}{\partial \rho_j^l}\right)_{T^l, \rho_k^l (k \neq j)} & \left(\frac{\partial \mu_n^v}{\partial \rho_1^v}\right)_{T^v, \rho_k^v (k \neq 1)} & \dots & \left(\frac{\partial \mu_n^v}{\partial \rho_n^v}\right)_{T^v, \rho_k^v (k \neq n)} \\ -\sum_{i=1}^n x_j \left(\frac{\partial P^l}{\partial \rho_i^l}\right)_{T^l, \rho_j^l (j \neq i)} & \left(\frac{\partial P^v}{\partial \rho_1^v}\right)_{T^v, \rho_j^v (j \neq 1)} & \dots & \left(\frac{\partial P^v}{\partial \rho_n^v}\right)_{T^v, \rho_j^v (j \neq n)} \end{pmatrix} \times \begin{pmatrix} \left(\frac{\partial \rho^l}{\partial T}\right)_\sigma \\ \left(\frac{\partial \rho_1^v}{\partial T}\right)_\sigma \\ \vdots \\ \left(\frac{\partial \rho_n^v}{\partial T}\right)_\sigma \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial \mu_i^l}{\partial T^l}\right)_{\rho^l} - \left(\frac{\partial \mu_i^v}{\partial T^v}\right)_{\rho^v} \\ \vdots \\ \left(\frac{\partial \mu_n^l}{\partial T^l}\right)_{\rho^l} - \left(\frac{\partial \mu_n^v}{\partial T^v}\right)_{\rho^v} \\ \left(\frac{\partial P^l}{\partial T^l}\right)_{\rho^l} - \left(\frac{\partial P^v}{\partial T^v}\right)_{\rho^v} \end{pmatrix} \quad (24)$$

In order to implement the integrate method, a converged starting point must be used. To do this, one would use the direct method (or an iterative method), subject to the equilibrium conditions, at a specified temperature and liquid-phase mole fraction to find the *first* point (*i.e.*, the liquid phase density and the vapor phase component densities). For an equation of state that has the mole fraction, density and temperature as independent variables, all of the elements of the matrix **A** and the solution vector **b** can be solved from the equation of state, either analytically or numerically. Thus, the unknown vector **x** is given as **x = A⁻¹b**. The $n+1$ coupled differential equations are then numerically integrated to yield the overall liquid phase density and the component densities of the vapor phase. The mole fractions can be extracted from the overall density and the individual component densities. A step in the temperature is taken next and the process is repeated until the critical point is reached. A block diagram (Figure 1) may be helpful in illustrating the technique.

For the purpose of the problem at hand, we have used a fourth-order Runge-Kutta method^[11] to integrate numerically the coupled pair of differential equations. We find that for the same step size used in the direct method (0.02K), we achieve reliable results for the phase densities in about the same amount of time using the integrate method, as seen in Figure 2 and Table 1, respectively. But, if we increase the integration step size to 1K, the integrate method provides reliable results for the coexistence densities with a computational speedup of

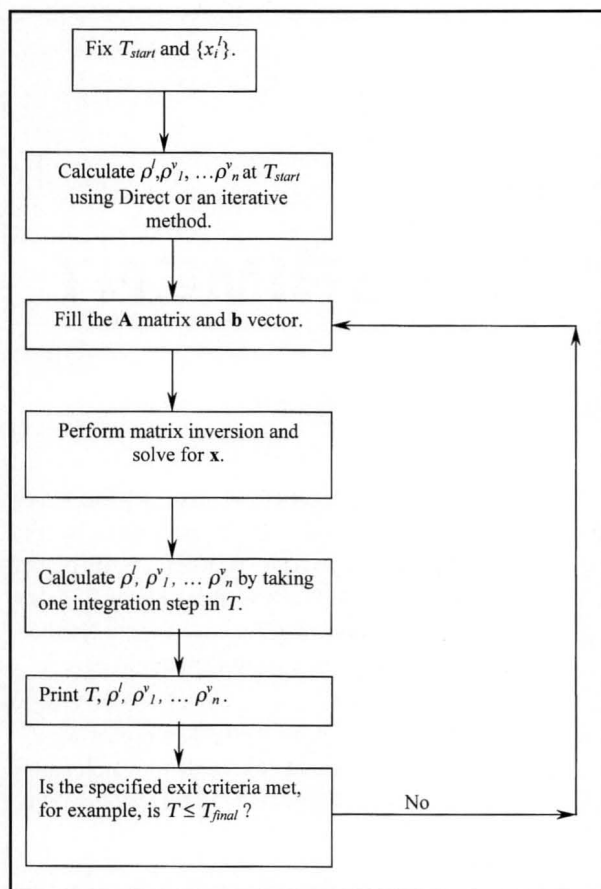
TABLE 1
CPU Time Required to Solve the Problem for Each Method at Various Step Sizes

Method	Step Size (K)	CPU Time (sec)
Direct	0.02	1.5
Integrate	0.02	1.5
Integrate	1.0	0.05
Integrate	10.0	0.02
Integrate+Direct	10.0	0.02

around 30 compared to the direct method. At a step size of 10K, the integrate method starts to fail (*i.e.*, produce inaccurate coexistence densities), as seen in Figure 2.

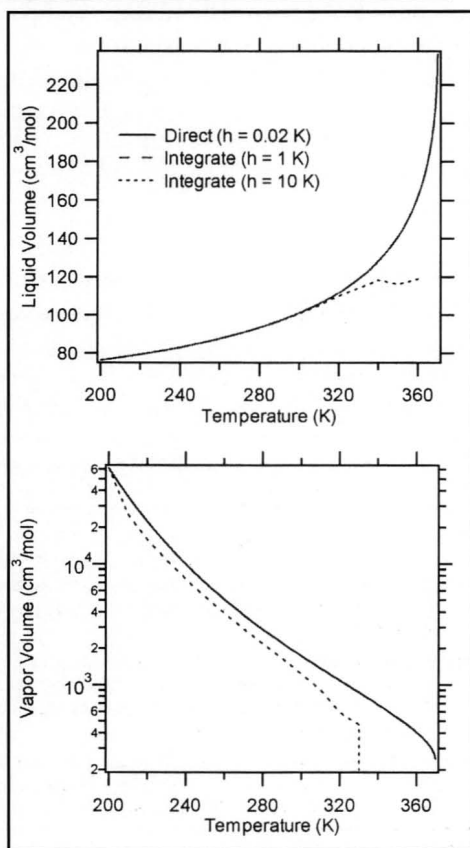
Integrate + Direct Method

The deficiency in the direct method is that above a certain step size in temperature, the method will not converge, while the advantage is that equilibrium is *ensured* upon convergence (assuming the roots are not repeated). Mirroring this is the integrate method, whose deficiency is that equilibrium is not ensured at each step (owing to



▲ **Figure 1.** A flowchart illustrating the integrate method algorithm.

► **Figure 2.** The saturated liquid volume (top) and saturated vapor volume (bottom) predicted from the direct method compared to that predicted from the integrate method at various step sizes.



the numerical integration scheme), while the advantage is that relatively large steps in temperature can be taken. A combination of the two methods, wherein first the integrate method *predicts* a guess value for the coexistence densities at the next temperature (as opposed to using the previously converged values), while the direct method uses these better guesses to converge to a solution, would seem to allow for the use of a larger step size in temperature. Such an approach has been suggested before.^[5,6,12] To this end, a combined *integrate+direct* method provided equilibrium densities in this problem for a step size of 10K with a computational speed of 75 as compared to the direct method alone.

CONCLUSIONS

A novel integration technique, here called *integrate* has been presented to solve phase equilibrium problems using equations of state. This method was shown to result in a computational speedup of around 30 relative to the direct method, owing to the larger step size the integrate method allows. Additionally, a combined *integrate+direct* method proved most useful in using the best features of both approaches. Future work will look at both the integrate and a combined *integrate+direct* method in the solution of more computationally demanding thermodynamic problems, such as tracing out liquid-liquid miscibility gaps or in determining the P-T diagram for retrograde systems for complicated equations of state.

ACKNOWLEDGMENTS

The idea for this work has its roots in a special projects course taken by DPVJ under the advisement of Professor David A. Kofke. Helpful suggestions by Dr. Paul Mathias are acknowledged.

REFERENCES

1. Fotouh, K., and K. Shukla, *Chem. Eng. Sci.*, **51**, p. 3763 (1996)
2. Elliott, J.R., and C.T. Lira, *Introduction to Chemical Engineering Thermodynamics*, Prentice Hall PTR, Upper Saddle River, NJ (1999)
3. Sandler, S.I., *Chemical and Engineering Thermodynamics*, 3rd ed., John Wiley & Sons, Inc. (1999)
4. Smith, J.M., H.C. Van Ness, and M.M. Abbott, *Introduction to Chemical Engineering Thermodynamics*, 5th ed., McGraw-Hill Company, New York, NY (1996)
5. Asselineau, L., G. Bogdanic, and J. Vidal, "A Versatile Algorithm for Calculating Vapor-Liquid Equilibria," *Fluid Phase Equil.*, **3**, p. 273 (1979)
6. Michelsen, M.L., "Calculation of Phase Envelopes and Critical Points for Multi-Component Mixtures," *Fluid Phase Equil.*, **4**, 1 (1980)
7. Redlich, O., and J.N.S. Kwong, *Chem. Rev.*, **44**, p. 233 (1949)
8. Prausnitz, J.M., R.N. Lichtenthaler, and E.G. de Azevedo, *Molecular Thermodynamics of Fluid-Phase Equilibria*, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ (1999)
9. Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, New York, NY (1992)
10. Modell, M., and R.C. Reid, *Thermodynamics and Its Applications*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ (1983)
11. Hoffman, J.D., *Numerical Methods for Engineers and Scientists*, 2nd ed., Marcel Dekker, Inc., New York, NY (2001)
12. Heidemann, R.A., "Computation of High Pressure Phase Equilibria," *Fluid Phase Equil.*, **14**, p. 55 (1983) □