Simple Phase Stability-Testing Algorithm in the Reduction Method

Hussein Hoteit and Abbas Firoozabadi
Reservoir Engineering Research Institute, Palo Alto, CA 94306

DOI 10.1002/aic.10908
Published online June 15, 2006 in Wiley InterScience (www.interscience.wiley.com).

Stability analysis is generally faster than phase-split calculations because the Rachford–Rice equation is not part of stability testing. However, there are occasions in which one may encounter computational inefficiency and indeed divergence in the single-phase region far away from the critical point. The computational inefficiency is found in various formulations of the stability analysis calculations. In this work, the cause of the problem is explained. A simple algorithm is also presented for stability analysis calculations in the reduction method. As in the past, only the Newton method is used in the solution of nonlinear equations except in some isolated iterations when one single successive substitution (SSI) iteration may be required to avoid nonphysical conditions. Results show that the stability analysis for the proposed algorithm is fast and robust. Results also show that the simple algorithm in the reduction method is superior to the combined SSI–Newton and quasi-Newton methods with conventional variables. Results are also presented showing that the stability analysis in two-phase is faster than phase-split calculations even when the initial guess is from the stability testing. © 2006 American Institute of Chemical Engineers *AIChE J,* 52: 2909–2920, 2006
Keywords: phase stability, Gibbs free energy, tangent plane distance, reduction method, Newton method

Introduction

Phase stability testing is an integral part of the phase-split calculations when a large number of such calculations must be performed.[1] In such calculations, one is interested to perform intensive computations based on given temperature, pressure, and composition. There are two approaches to perform phase-split calculations that can produce nearly identical results. In the conventional approach, the mole fractions or the mole numbers are the primary variables.[2-7] In the reduction method, stability analysis is performed using transformed equations, where the number of primary unknowns is less than the number of components. Even in the conventional space, different algorithms can be used to perform phase stability analysis.[8]

Two approaches are generally used to formulate the stability analysis. Based on the work of Baker et al.,[2] Michelsen[3] formulated the stability analysis as a nonlinear-unconstrained problem where a two-stage initialization is implemented to locate the stationary points of the tangent-plane distance (TPD). This approach is very efficient and is widely implemented. A second approach is based on the global minimization of Gibbs free energy or the global minimization of the TPD. McDonald and Floudas[5] and Harding and Floudas[9] presented a global optimization approach that guarantees locating of the global minimum of the TPD. Hua et al.[6,10] used the interval-Newton method to locate all stationary points of the TPD. This method guarantees convergence and is independent of the initial guess. Although the global minimization methods are reliable, they are also costly and limited to systems with a small number of components. In this work, our objective is to provide a very efficient algorithm that can be applied in flow simulators with a large number of components.

The reduction method for phase-split calculations was first

A. Firoozabadi is also affiliated with Yale University, New Haven, CT 06520.
Correspondence concerning this article should be addressed to A. Firoozabadi at abbas.firoozabadi@yale.edu.

introduced by Michelsen,[11] where zero binary interaction coefficients were assumed. Later, Jensen and Fredenslund[12] extended the method for nonzero binary interaction coefficients between one component and the remaining ones. Hendricks[13] and Hendricks and van Bergen[14] introduced a general formulation of the reduction method that allows nonzero binary interaction coefficients. In the reduction space, the reduced variables are related to compositional space where the eigenvalues derived from the interaction coefficients result in a substantial reduction of variables.[14] The transformed variables from the reduction space also make the thermodynamics functions such as the TPD surface smooth.[1] There are computational benefits from smoothness of the TPD surface including the use of the Newton method with a poor initial guess.

To the best of our knowledge, two reports in the literature discuss the use of the reduction variables to carry out thermodynamics stability analysis. Firoozabadi and Pan[1] used the minimization with constraints and Lagrange multipliers. Later, Nichita et al.[15] suggested a direct approach to minimize the TPD. Since the publication of our article[1] on stability analysis in the reduction space, we have further tested stability computations and found that in some narrow regions in single-phase, there may be a substantial increase in the number of iterations in the stability testing. Because the problem occurs in single phase outside the two-phase envelope, one should not attempt to relate the problem to the intrinsic stability limit. When stability testing is performed, one is not aware whether the mixture is in single-phase or in two-phase and, therefore, convergence problems can be encountered in the single-phase region. Based on extensive testing of the stability computations in the reduction space, Firoozabadi and Pan[1] recommended the number of iterations to be limited to 50. In case there was no convergence in 50 iterations, the fluid was assumed to be single-phase. Occasionally, there was no convergence in 50 iterations. Once the number of iterations in the Newton method increases to more than say 15, there is a definite need for investigation.

Recently, we also found that the problem of convergence in stability analysis exists in the use of other methods such as the combined successive substitution iteration (SSI)–Newton and quasi-Newton methods for the conventional variables. The goal of this work is to examine the convergence problem and to offer a solution.

This article is structured as follows. We first present various formulations of stability testing. Then, we offer the reason for numerical inefficiency, which is often limited to a narrow region away from the critical point in the single-phase state. A simple algorithm is presented for an efficient calculation of the minimum of the TPD. Two numerical examples are presented to show the robustness and efficiency of the proposed algorithm. The work is concluded with few remarks.

Basic Equations of the Stability Analysis

The tangent-plane-distance (TPD) analysis is used to determine the stability of a phase.[2,3,7] Consider a c-component mixture of mole fractions z_i ($i = 1, \ldots, c$) at a given temperature T and pressure p. For a trial phase composition y_i ($i = 1, \ldots, c$), the TPD is expressed by

$$\overline{TPD}(y) = \sum_{i=1}^{c} y_i[\mu_i(y) - \mu_i(z)] \tag{1}$$

where μ_i is the chemical potential of component i. The above expression is commonly written in a dimensionless form in terms of the fugacity coefficients φ_i:

$$TPD(y) = \frac{\overline{TPD}(y)}{RT} = \sum_{i=1}^{c} y_i[\ln \varphi_i(y) + \ln y_i - d_i(z)] \tag{2}$$

where

$$d_i(z) = \ln z_i + \ln \varphi_i(z) \tag{3}$$

The mixture is considered to be stable if the TPD is positive or zero for any trial phase composition y, that is,

$$\begin{cases} TPD(y) \geq 0 \quad \forall\, y = (y_1, y_2, \ldots, y_c) \in \Re_+^c, \quad \text{such that} \\ \sum_{i=1}^{c} y_i = 1 \end{cases} \tag{4}$$

Two standard approaches are generally used to solve Eq. 4. In one approach, the TPD is examined in the context of a constrained minimization problem and the positivity of the TPD is checked at its global minimum. In the second approach, the stationary points of the TPD are calculated. The mixture is stable if the tangent plane distance is nonnegative at all the local minima.[3] The second approach is widely used because of its simplicity and efficiency. At the stationary points, the first derivative of the TPD with respect to the independent variables y_i ($i = 1, \ldots, c - 1$) vanishes. The stationary conditions can then be written as

$$\ln \varphi_i(Y) + \ln Y_i - d_i(z) = 0 \qquad i = 1, \ldots, c \tag{5}$$

where the introduced variable Y_i in Eq. 5 is defined by

$$Y_i = y_i e^{-K} \tag{6}$$

and

$$y_i = \frac{Y_i}{\sum_{i=1}^{c} Y_i} \tag{7}$$

The system is assumed stable if the following condition is satisfied:

$$Y_T = \sum_{i=1}^{c} Y_i \leq 1 \tag{8}$$

Alternative formulation

Michelsen[3] introduced the formulation in terms of an unconstrained minimization of Eq. 4 and defined a modified TPD function *TPD** in terms of the variable Y_i:

$$TPD^*(Y) = 1 + \sum_{i=1}^{c} Y_i[\ln \varphi_i(Y) + \ln Y_i - d_i(z) - 1] \quad (9)$$

He showed that the stationary points of *TPD** correspond to the stationary points of *TPD* and that the functions *TPD** and *TPD* have the same signs at the stationary points. Therefore, to determine the stability of a mixture, it is sufficient to check the positivity of the modified function *TPD** at all its stationary points. If all are positive, this implies that the values of the original *TPD* function at its stationary points are positive as well. The mixture is thus stable; otherwise, if *TPD** is strictly negative at one of the stationary points, the mixture is then considered to be unstable.

Conventional Methods

In this section, we review two numerical algorithms: the first algorithm uses the successive substitution iteration (SSI) combined with the Newton method; the second is a quasi-Newton method with the Broyden–Fletcher–Golfarb–Shanno (BFGS) update of the Hessian matrix.[16,17] The two methods seek to locate the local stationary points of the Gibbs free energy and then evaluate the TPD at those points. There are variations of the above methods. For example, the accelerated successive substitution (ASS) method can replace the successive substitution. In our experience, despite the efficiency of the ASS, it does not have the robustness of the SSI, and it may diverge in some complicated near-critical problems.

A simple numerical algorithm to solve the system of nonlinear Eqs. 5 is the SSI method:

$$Y_i^k = \exp[d_i(z) - \ln \varphi_i(Y^{k-1})] \qquad i = 1, \ldots, c \quad (10)$$

where k denotes the iteration counter.

The computational cost of each SSI iteration is low. However, excessive iterations may be required in the vicinity of the critical point. This could eventually result in computational inefficiency.

The Newton method can be used to solve the nonlinear Eqs. 5 by using the independent unknowns $[Y_i\ (i = 1, \ldots, c)]$. The differentiation of Eq. 5 with respect to Y_j yields

$$\frac{\partial}{\partial Y_j}\left[\frac{\partial}{\partial Y_i} TPD^*(Y)\right] = \frac{\delta_{ij}}{Y_i} + \frac{\partial \ln \varphi_i(Y)}{\partial Y_j} \qquad i, j = 1, \ldots, c \quad (11)$$

where

$$\frac{\partial \ln \varphi_i(Y)}{\partial Y_j} = \frac{1}{Y_T} \sum_{k=1}^{c-1} \frac{\partial \ln \varphi_i(y)}{\partial y_k} (\delta_{kj} - y_k) \qquad i, j = 1, \ldots, c$$

The Newton method, when it converges, has a quadratic convergence rate. However, it is well known that it requires a good initial guess, or else it may diverge, especially in the critical region. The combined SSI–Newton method is a natural approach that may take advantages of the robustness of the SSI and the efficiency of the Newton method. The SSI method is performed first to attain a "sufficiently close" initial guess to be used in the Newton method. The main steps of the conventional SSI–Newton algorithm are presented in Appendix A.

Quasi-Newton Method with BFGS Update

Quasi-Newton methods are generally used when the analytical expression of the Hessian matrix is either unavailable or too expensive to compute. They provide an inexpensive approximation for the Hessian matrix based on the information from the previous iterations. These methods, if they converge, are often used to achieve a superlinear convergent rate (~1.6). Although, in our case, the Hessian matrix (Eq. A8) can be readily computed, the computational cost of the Newton method is expensive per iteration mainly because of the construction and the solution of a linear system of size c. Using a special quasi-Newton method can reduce the computational cost per iteration with the hope that the method converges with a number of iterations comparable to that of the Newton method.

The quasi-Newton methods primarily differ in the way the Hessian matrix is approximated. Let us denote the approximate Hessian matrix by $B^k = B(\alpha^k)$ at the kth iteration. A typical iteration of the quasi-Newton method is

$$B(\alpha^{k+1})[\alpha^{k+1} - \alpha^k] = -F(\alpha^k) \quad (12)$$

The main idea to approximate the Hessian matrix is to use two successive iterates α^{k+1} and α^k with the gradients $F(\alpha^{k+1})$ and $F(\alpha^k)$, such that the following condition is satisfied:

$$B^{k+1}s^k = g^k \quad (13)$$

where $s^k = \alpha^{k+1} - \alpha^k$ and $g^k = F(\alpha^{k+1}) - F(\alpha^k)$.

To approximate B^{k+1}, Michelsen[3] recommended use of the BFGS update formula[16,17] and to reset the Hessian matrix to the identity matrix after each iteration. Therefore, there is no need to use a continued update of the Hessian matrix. Equation 13 becomes (see Appendix B)

$$\alpha^{k+1} = \alpha^k + \tilde{F}(g^k, s^k, F^k) \quad (14)$$

where

$$\tilde{F}(g^k, s^k, F^k) = -\left[F + \frac{1}{\langle s, g\rangle}\left(\langle s - g, F\rangle + \frac{\langle g, g\rangle\langle s, F\rangle}{\langle s, g\rangle}\right)s - \frac{\langle s, F\rangle}{\langle s, g\rangle} g\right] \quad (15)$$

Unlike the conventional Newton method, the above-cited quasi-Newton method avoids solving a linear system at each iteration.

When the trial phase solution approaches the trivial solution, the quantity

Table 1. Relevant Properties for Mixture 1

Composition	Mol Fraction (%)	T_c (K)	p_c (bar)	ω	MW (g/mol)	Shift Para.	$\delta C_1 - C_i$
C1	35	190.6	45.4	0.008	16.04	−0.154	0.0
C2	3	305.4	48.2	0.098	30.07	−0.100	0.0
C3	4	369.8	41.9	0.152	44.10	−0.085	0.0
nC4	6	425.2	37.5	0.193	58.12	−0.064	0.02
nC5	4	469.6	33.3	0.251	72.15	−0.041	0.02
nC6	3	507.5	30.1	0.305	86.18	−0.014	0.025
nC7	5	540.3	27.4	0.350	100.20	0.002	0.025
nC8	5	568.8	24.9	0.396	114.23	0.022	0.035
nC10	30	617.9	21.0	0.484	142.29	0.083	0.045
nC14	5	691.9	15.2	0.747	198.10	0.130	0.045

$$r = 2\,\frac{TPD^*(Y)}{\beta} \tag{16}$$

approaches unity. In Eq. 16, β is defined as

$$\beta = \sum_{i=1}^{c} (Y_i - z_i)\,\frac{\partial TPD^*(Y)}{\partial Y_i}$$

This criterion is useful to terminate the iterations when one approaches the trivial solution.[3] The main steps of the quasi-Newton method with the BFGS-update algorithm are given in Appendix B.

Basic Equations of the Reduction Method

The theory of the reduction method for the stability analysis is described by Firoozabadi and Pan.[1] The main idea is to write the TPD function in terms of a new set of independent variables Q_i ($i = 1, \ldots, M$), where M is often much smaller than c, instead of the conventional variables Y_i ($i = 1, \ldots, c$). The increase in computational speed in the reduction method is from (1) smoothness in the TPD surface and (2) reduction in the number of variables based on the number of significant eigenvalues. In some applications for facility design in oil production sometimes in excess of 40 species are required for accurate calculations. The savings from use of the CPU reduction method will be substantial.

In the reduction method, the expressions of "a" and "B" in the Peng–Robinson equation of state (PR-EOS) are given by (see Appendix C)

$$a = \sum_{k=1}^{m} \lambda_k Q_k^2 \tag{17}$$

and

$$B = \sum_{i=1}^{c} z_i B_i \tag{18}$$

where λ_k ($k = 1, \ldots, m$) represents the significant eigenvalues of the matrix $\bar{\beta} = [1 - \delta_{ij}]_{i,j=1,\ldots,c}$, where generally $m \ll c$. In Eq. 17, Q_k ($k = 1, \ldots, m$) values are defined as

$$Q_k = \sum_{i=1}^{c} z_i q_{ki} \tag{19}$$

where $q_{ki} = a_i^{1/2} q'_{ki}$ and q'_{ki} ($i = 1, \ldots, c$) are related to the entities of the eigenvector corresponding to λ_k. More details are provided in Appendix C.

The primary unknowns in the reduction method are, therefore, $Q_1, Q_2, \ldots, Q_m, B$. It can be readily shown that the compressibility factor Z and $\ln \varphi_i$ are functions of $Q_1, Q_2, \ldots, Q_m, B$ (see Appendix C) and so is the TPD.

Solution Method

The numerical approach for stability analysis, introduced by Firoozabadi and Pan,[1] is based on the constrained minimization of the TPD in the reduction method using Lagrange multipliers. The numerical method presented here is similar to the conventional methods where the stability is determined by confirming the positivity of the TPD at the local minima.

Let y_i ($i = 1, \ldots, c$) be the composition of the trial phase. The constrained equations given in Eqs. 17 and 18 can be written in a compact form:

$$Q_\alpha = \sum_{i=1}^{c} y_i q_{\alpha i} \qquad \alpha = 1, \ldots, m+1 \tag{20}$$

where $q_{\alpha i} = b_i$ and $Q_\alpha = b$, for $\alpha = M = m + 1$ ($i = 1, \ldots, c$).

Replace the mole numbers from Eq. 7 in Eq. 20 and define the functions $h_\alpha(Q)$ such that

$$h_\alpha(Q) \equiv Y_T Q_\alpha - \sum_{i=1}^{c} Y_i q_{\alpha i} = 0 \qquad \alpha = 1, \ldots, M \tag{21}$$

where $Y_T = \sum_{i=1}^{c} Y_i$ and $M = m + 1$.

At the stationary points, the derivative of the TPD function with respect to Y_i ($i = 1, \ldots, c$) is zero. From Eq. 5, we get

$$\ln Y_i = d_i(z) - \ln \varphi_i(Q) = 0 \qquad i = 1, \ldots, c \tag{22}$$

then

Table 2. Relevant Properties for Mixture 4

Composition	Mol Fraction (%)	T_c (K)	p_c (bar)	ω	MW (g/mol)	Shift Para.	$\delta N_2 - C_i$	$\delta CO_2 - C_i$	$\delta C_1 - C_i$
N2	0.190	126.2	33.5	0.040	28.01	−0.166	—	—	—
CO2	1.01	304.2	72.8	0.225	44.01	−0.100	−0.02	—	—
C1	86.49	190.6	45.4	0.008	16.04	−0.154	0.12	0.093	—
C2	2.480	305.4	48.2	0.098	30.07	−0.100	0.12	0.128	0.000
C3	1.280	369.8	41.9	0.152	44.10	−0.085	0.12	0.123	0.000
iC4	0.720	408.1	36.0	0.176	58.12	−0.079	0.12	0.136	0.020
nC4	0.370	425.2	37.5	0.193	58.12	−0.064	0.12	0.125	0.020
iC5	0.220	460.4	33.4	0.227	72.15	−0.043	0.12	0.131	0.025
nC5	0.140	469.6	33.3	0.251	72.15	−0.041	0.12	0.120	0.025
C6–C9	0.9978	547.43	30.3	0.4099	103.56	0.015	0.12	0.120	0.035
C10–C14	1.2590	643.77	22.9	0.6714	161.99	0.101	0.12	0.120	0.038
C15–C19	1.2321	724.23	17.0	0.9296	233.97	0.167	0.12	0.120	0.038
C20–C24	0.9024	777.38	13.4	1.1603	302.66	0.209	0.12	0.120	0.038
C25+	2.7087	849.61	9.9	1.6043	437.75	0.267	0.12	0.120	0.038

$$Y_i = \frac{\tilde{d}_i}{\varphi_i(Q)} \qquad i = 1, \ldots, c \tag{23}$$

where $\tilde{d}_i = \exp[d_i(z)]$.

Note that the fugacity coefficient in Eqs. 22 and 23 can be calculated in terms of the reduced variables Q_α ($\alpha = 1, \ldots, M$) (see Appendix C).

By combining Eqs. 23 and 21, we can formulate the problem as follows.

We seek Q_α ($\alpha = 1, \ldots, M$) such that

$$h_\alpha(Q) \equiv \sum_{i=1}^{c} \frac{\tilde{d}_i(Q_\alpha - q_{\alpha i})}{\varphi_i(Q)} = 0 \qquad \alpha = 1, \ldots, M \tag{24}$$

From Eq. 20, one can readily show that the reduced variables Q_α ($\alpha = 1, \ldots, M$) are constrained by

$$\min_{i=1,\ldots,c} q_{\alpha i} \le Q_\alpha \le \max_{i=1,\ldots,c} q_{\alpha i} \qquad \alpha = 1, \ldots, M \tag{25}$$

The system of nonlinear expressions given by Eq. 24 is solved by the Newton method. The Jacobian matrix is given in Appendix C. The main steps of the algorithm are presented in Appendix D.

The constraints in Eq. 25 ensure a positive root for the cubic EOS to prevent overflow in the numerical procedure. If at a certain iteration one of these constraints is not satisfied, the algorithm overpasses the Newton iteration (step 5.2.1 in Appendix D) and instead is updated by an SSI iteration (step 5.2.2). Therefore, the algorithm couples the SSI and the Newton methods, where one SSI step is performed in case of failure in the Newton iteration. We found that most cases where the Newton iteration violates the constraints in Eq. 25 correspond to single-phase systems with high pressure. Usually only one SSI step (step 5.2.2) is sufficient to retain convergence.

The algorithm presented above is similar to the one proposed by Nichita et al.[15] with two main differences.

The algorithm by Nichita et al.[15] may lead to nonphysical values of the variables in the Newton iterations. We found that the control step (step 5.2) given in Eq. 25 is crucial to the

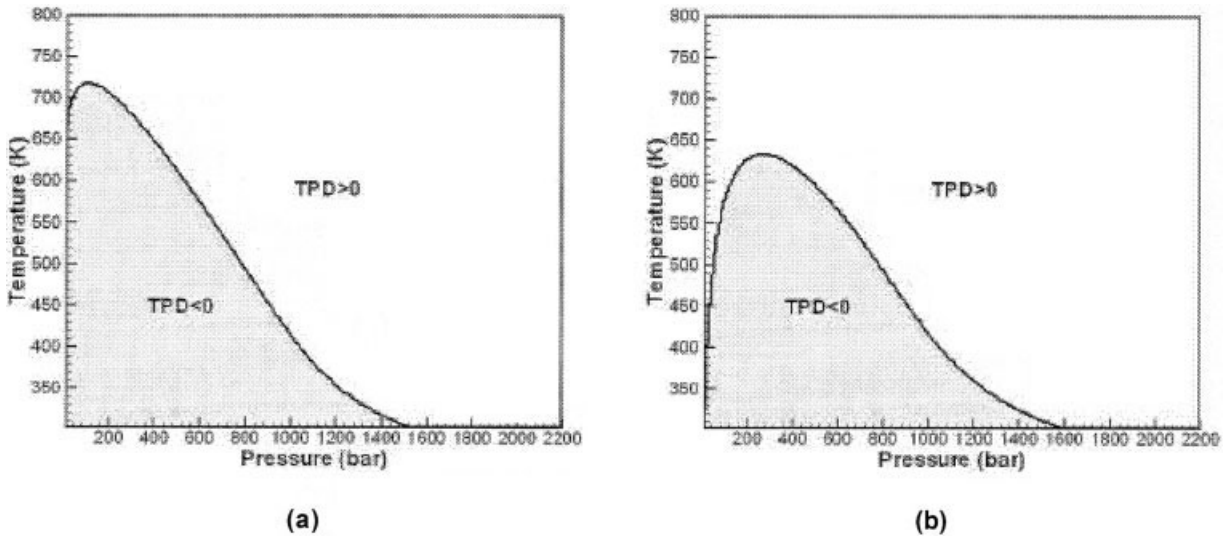

Figure 1. Minimum of the tangent-plane-distance (TPD) by the three methods.

Mixture 4, (a) stability of vapor-like phase, (b) stability of liquid-like phase.

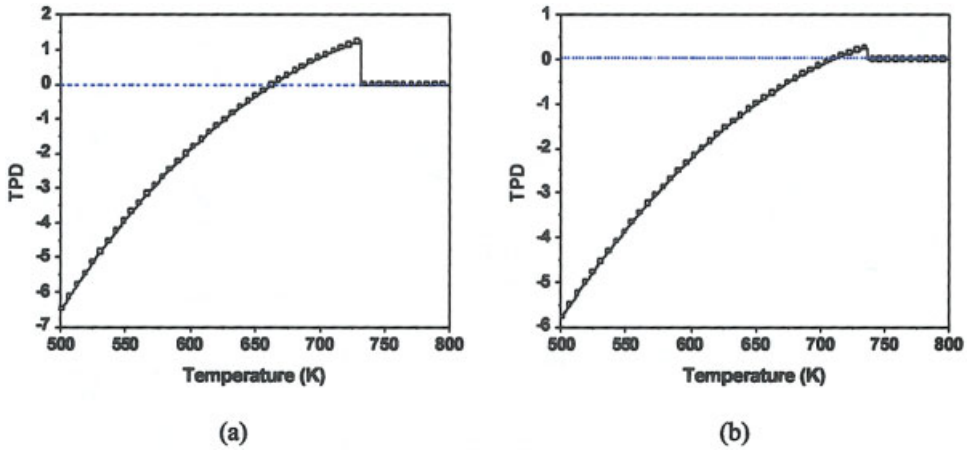

Figure 2. Minimum of the TPD vs. temperature.

Mixture 4, (a) stability of vapor-like phase at $p = 10$ bar, (b) stability of vapor-like phase at $p = 50$ bar. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

numerical stability of the procedure. Nichita et al.[15] also proposed using the Newton method with line search backtracking. We found their proposal has a minor effect on the robustness of the algorithm but adds to computational cost. Their approach does not ensure the constraints in Eq. 25.

The second difference is in the initial guesses. Nichita et al. proposed performing the stability with two initial guesses:

$$Q_\ell^{(0)} = (q_{1\ell}, q_{2\ell}, \ldots, q_{M\ell}) \tag{26}$$

and

$$Q_h^{(0)} = (q_{1h}, q_{2h}, \ldots, q_{Mh}) \tag{27}$$

where the indexes ℓ and h correspond to the lightest and the heaviest components, respectively.

Let us consider the following two initial guesses for the composition of the trial phase:

$$y_i = \begin{cases} 1 & \text{if } i = \ell \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, c \tag{28}$$

and

$$y_i = \begin{cases} 1 & \text{if } i = h \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, c \tag{29}$$

One can readily find that the initial guesses in Eqs. 26 and 27 can be obtained by replacing Eqs. 28 and 29 in Eq. 20. It is expected that these initial guesses may not be as good as those from Wilson's correlation, especially when there is no predominance of one of the K-values (that is, the K-values are comparables).

Phase-Split Initialization from the Stability Analysis

The stability analysis can provide the initial guess for the K-values in phase-split calculations if the system is unstable

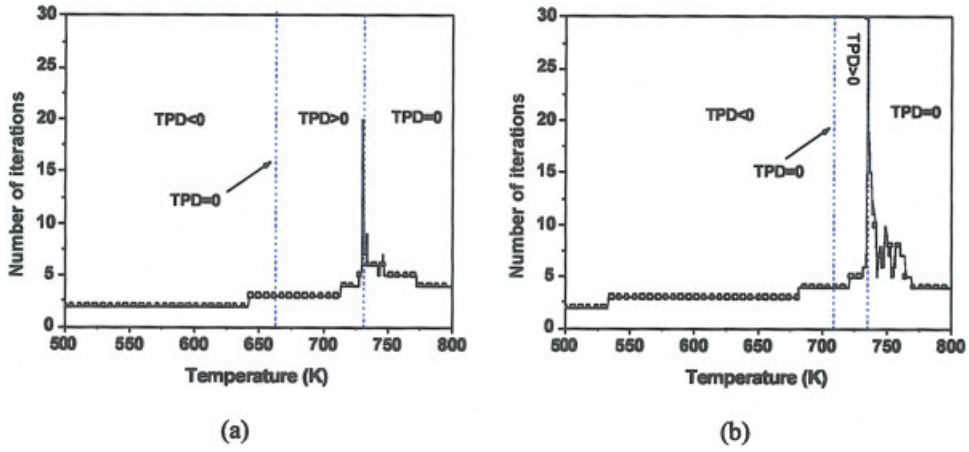

Figure 3. Number of iterations in the reduction method vs. temperature.

Mixture 4, (a) stability of vapor-like phase at $p = 10$ bar, (b) stability of vapor-like phase at $p = 50$ bar. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

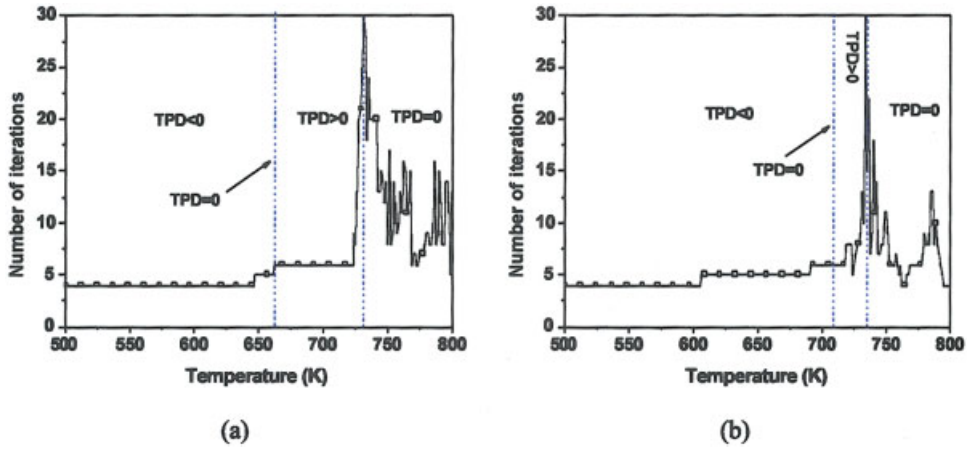

Figure 4. Number of iterations in the BFGS–quasi-Newton method vs. temperature.

Mixture 4, (a) stability of vapor-like phase at $p = 10$ bar, (b) stability of vapor-like phase at $p = 50$ bar. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

and to avoid such computations if the system is stable. When the TPD is negative and the trial phase is vapor-like, the K-values are updated from

$$K_i = \frac{\varphi_i(z)}{\varphi_i(Y)} = \frac{Y_i}{z_i} \tag{30}$$

When the TPD is negative and the trial phase is liquid-like, the K-values are updated from

$$K_i = \frac{\varphi_i(Y)}{\varphi_i(z)} = \frac{z_i}{Y_i} \tag{31}$$

When the TPD is negative for both liquid- and vapor-like trial phases, the K-values that correspond to the smaller TPD are selected. Note that the K-values should not be calculated from the normalized mole fractions y_i instead of the mole numbers Y_i in Eqs. 30 and 31. A numerical example to be presented later will reveal that using the mole fractions y_i instead of Y_i gives poor estimation of the K-values for low pressures. The K-values from Eq. 30 or Eq. 31 provide a good initial guess for the phase-split computations for all cases (whatever the value of the TPD is), except if there is a failure in the stability analysis code and the K-values are all >1 or all <1. In this case, although we have not encountered such a case, the K-values can instead be calculated from Wilson's correlation.

Results

The reduction method is compared with the conventional methods, SSI–Newton and quasi-Newton with BFGS-update. The stability of two mixtures is tested by using the three algorithms presented earlier in the article. These mixtures are cited in Firoozabadi and Pan[1]:

- Mixture 4 (14-component mixture)
- Mixture 1 (10-component mixture)

The composition, critical properties, and the binary interaction coefficients are given in Tables 1 and 2. The stability of each mixture is checked at various temperatures and pressures. The

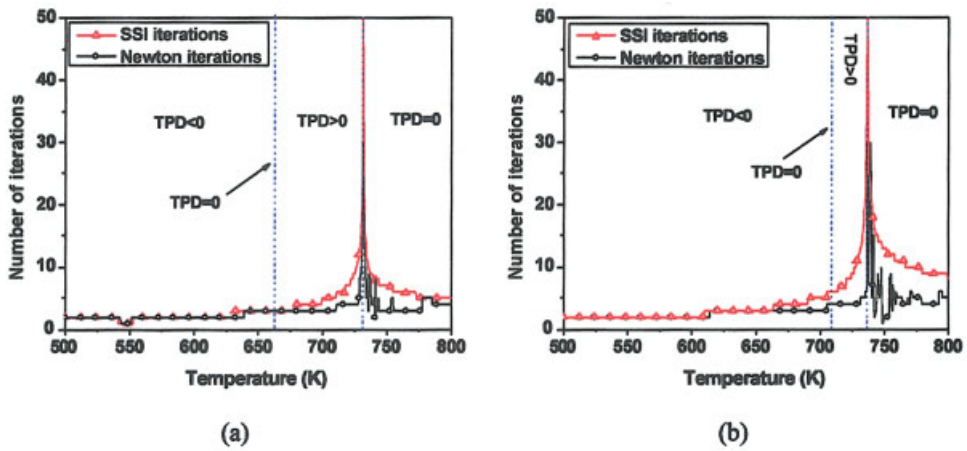

Figure 5. Number of SSI and Newton iterations in the SSI–Newton method vs. temperature.

Mixture 4, (a) stability of vapor-like phase at $p = 10$ bar, (b) stability of vapor-like phase at $p = 50$ bar. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

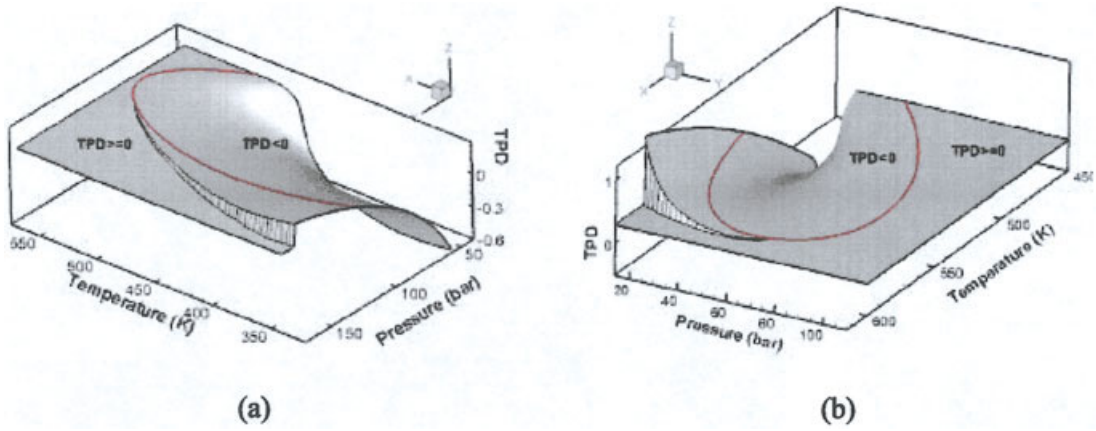

Figure 6. 3D plot of the minimum of the TPD vs. pressure and temperature.

Mixture 1, (a) stability of vapor-like phase, (b) stability of liquid-like phase. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

composition is kept fixed. The pressure is varied between 5 and 2400 bar with a step interval equal to 11 bar (200 points). The temperature is varied between 300 and 800 K with a step interval equal to 2.5 K (200 points). The three codes are tested for 40,000 points for each mixture. The number of reduced variables for Mixture 1 and Mixture 4 is four and six, respectively.

The switching criterion from the SSI to the Newton method in the conventional code is $\varepsilon_{SSI} = 10^{-2}$. The stopping criterion for the three codes is $\varepsilon = 10^{-9}$, except for the BFGS–quasi-Newton code that has an additional stopping criterion when the solution approaches the trivial solution (Eq. 16).

Figure 1 shows regions with positive and negative TPD for Mixture 4 by varying the temperature and pressure. The two-phase region corresponds to the zones where the minimum of the TPD is negative.

Figure 2 presents the minimum of the TPD vs. temperature for Mixture 4 at fixed pressures of $p = 10$ bar and $p = 50$ bar. These plots should not be confused with the TPD functions. They just represent the minimum of the TPD. In other words, at a given point (p, T) we performed stability analysis to calculate the value of the minimum of the TPD function. The two-phase region corresponds to the region where the TPD is negative either for vapor-like or liquid-like phases. Note that there are some regions of discontinuity in Figure 2. The sharp variation in the minimum TPD from a strictly positive value to zero corresponds to the region where the numerical method at one condition (such as pressure or temperature) converges to a nontrivial solution. With a small perturbation on that condition, the method converges to the trivial solution. We have found that the three methods need additional iterations close to the discontinuity regions. This discontinuity in the minimum of TPD corresponds to the dashed line that separates regions C and D in the work of Rasmussen et al.,[18] who in this work, and in previous work reported in the literature, neither state the existence of the

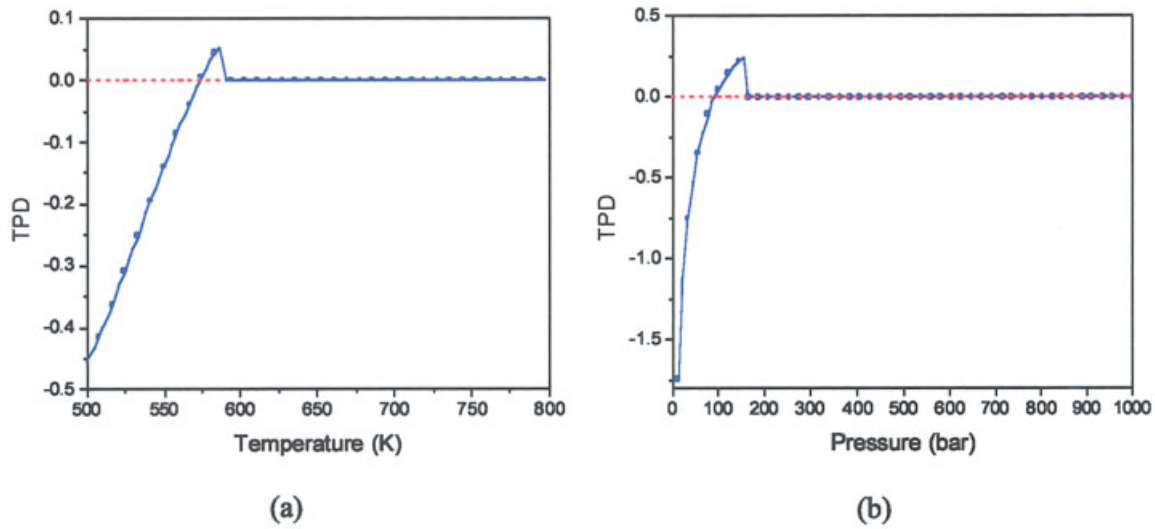

Figure 7. Minimum of the TPD vs. temperature and pressure.

Mixture 1, (a) stability of vapor-like phase at $p = 25$ bar, (b) stability of vapor-like phase at $T = 302.5$ K. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

Table 3. Total CPU Time (seconds) in Stability Calculations for 40,000 Points: For Both Vapor and Liquid Phases

	Reduction Method	SSI–Newton Method	Quasi-Newton Method
Mixture 4	1.4	40.1	4.5
Mixture 1	0.9	13.4	3.2

discontinuity in the minimum of TPD nor report the (high) number of iterations in stability testing in that region. This behavior is different from the usual thinking that the computational cost is higher at the phase boundary.

In Figure 3, we plot the number of Newton iterations by the reduction method vs. temperature at fixed pressures of $p = 10$ bar and $p = 50$ bar. In Figures 4 and 5, we show the number of iterations by the BFGS–quasi-Newton and SSI–Newton methods. Similarly to the reduction method, an increase in the number of iterations occurs close to the discontinuities. An excessive number of SSI iterations can be seen in Figure 5. For some narrow regions, there can be more than 800 SSI iterations to reduce the residual to the switching criterion from SSI to the Newton method. Figures 6 and 7 show the minimum of the TPD for Mixture 1. For Mixtures 1 and 4, the minimum of the TPD has a similar behavior. The discontinuity region is from the positive side of the TPD. The SSI–Newton and the BFGS–quasi-Newton methods failed to converge for some of the points in the discontinuity zone. The reduction method also failed to converge within 30 iterations for some of the points in that region. These points represented $<0.2\%$ of the points examined. However, we have noticed that for all the pointes we tested ($>10^5$ points), the reduction method converged with fewer than 12 iterations in the two-phase region. Thus, 15 iterations in the reduction method should be enough for convergence. If the method does not converge within this limit, the mixture is assumed to be single phase. We did not find a reasonable limit for the iterations in the quasi-Newton and SSI–Newton methods with the conventional variables.

CPU times for the three methods are given in Table 3. Computations are performed on a 2.4-GHz Pentium PC. The reduction method is about one order of magnitude faster than the SSI–Newton method and three times faster than the BFGS–quasi-Newton method.

To have an idea about the cost of the phase-split computations compared to the stability analysis computations, we performed phase-split calculations using our code, which is based on the SSI–Newton procedure for the points for Mixtures 1 and 4 in a two-phase state. The CPU time for the phase-split computations is about 31 s for 29% of points for Mixture 4 and about 1 s for roughly 3% of the points in Mixture 1 (where the data are in two-phase). Comparing these results to the CPU time for the stability analysis by the conventional SSI–Newton method, we find that the phase-split code is three times slower than the stability testing code for each run. The stability code by the reduction method is then at least one order of magnitude faster than the phase-split code for each run.

Discussion and Concluding Remarks

Figures 2 and 7 show that in the single-phase state, away from the phase boundary, the minimum of the TPD has a discontinuity toward the trivial solution. In this region, we may encounter convergence problems in the computation of the minimum of the TPD. Three methods, two based on the compositional variables and one based on the reduction variables, encountered convergence problems in our test examples. The severity of convergence was very pronounced in the SSI–Newton method with conventional variables, and it is the most costly. The Newton method with reduction variables is the least costly. Our preferred approach for robust stability analysis is the use of reduced space in the Newton method.

Acknowledgments

We sincerely thank Dr. Fredrik Saaf from Schlumberger for bringing the problem to our attention and for his help. This work was supported by the member companies of the Reservoir Engineering Research Institute (RERI) whose support is greatly appreciated.

Literature Cited

1. Firoozabadi A, Pan H. Fast and robust algorithm for the compositional modeling: Part I. Stability analysis testing. *SPE J.* 2002;March:78-89.
2. Baker LE, Pierce AC, Luks KD. Gibbs energy analysis of phase equilibria. *SPE J.* 1982;22:731-742.
3. Michelsen ML. The isothermal flash problem: Part I. Stability. *Fluid Phase Equilib.* 1982;9:1-19.
4. Sun AC, Seider WD. Homotopy-continuation method for stability analysis in the global minimization of the Gibbs free energy. *Fluid Phase Equilib.* 1995;103:213-249.
5. McDonald CM, Floudas CA. Global optimization for the phase stability problem. *AIChE J.* 1995;41:1798-1814.
6. Hua JZ, Brennecke JF, Stadtherr MA. Reliable prediction of phase stability using an interval Newton method. *Fluid Phase Equilib.* 1996; 116:52-59.
7. Firoozabadi A. *Thermodynamics of Hydrocarbon Reservoirs.* New York, NY: McGraw-Hill; 1999.
8. Guo M, Wang S, Repke JU, Wozny G. A simulation method for two- and three-liquid-phase stability determination. *AIChE J.* 2004;50: 2571-2582.
9. Harding ST, Floudas CA. Phase stability with cubic equations of state: A global optimization approach. *AIChE J.* 2000;7:1422-1440.
10. Hua JZ, Brennecke JF, Stadtherr MA. Reliable computation of phase stability using interval analysis: Cubic equation of state models. *Comput Chem Eng.* 1998;22:1207-1214.
11. Michelsen ML. Simplified flash calculations for cubic equations of state. *Ind Eng Chem Process Des Dev.* 1986;25:184-188.
12. Jensen BH, Fredenslund A. A simplified flash procedure for multicomponent mixtures containing hydrocarbons and one non-hydrocarbon using two-parameter cubic equation of state. *Ind Eng Chem Res.* 1987;26:2129-2134.
13. Hendricks EM. Reduction theory for phase equilibrium problems. *Ind Eng Chem Res.* 1988;27:1728-1732.
14. Hendricks EM, van Bergen AR. Application of a reduction method to phase equilibria calculations. *Fluid Phase Equilib.* 1992;74:17-34.
15. Nichita DV, Broseta D, Hemptinne JC. Multiphase equilibrium calculations using reduced variables. SPE 89439. Proceedings of the 2004 SPE/DOE Meeting, Tulsa, OK; 2004.
16. Dennis J, Moré J. Quasi-Newton methods, motivation and theory. *SIAM Rev.* 1977;19:46-89.
17. Dennis J, Schnabel R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Philadelphia, PA: Society for Industrial and Applied Mathematics; 1996.
18. Rasmussen CP, Krejbjerg K, Michelsen ML, Bjurstrom KE. Increasing computational speed of flash calculations with application for compositional. *SPERE.* 2006;February:32-38.

Appendix A

Algorithm: SSI–Newton

1. Initialize the K-values from Wilson's correlation

$$K_i = \frac{p_{ci}}{p} \exp\left[5.37(1.0 + \omega_i)\left(1.0 - \frac{Tc_i}{T}\right)\right] \quad \text{(A1)}$$

2. For given z_i, calculate $d_i(z)$ from Eq. 3.
3. Calculate Y_i:

$$Y_i = \begin{cases} \dfrac{z_i}{K_i} & \text{if the trial phase is liquid} \\ z_i K_i & \text{if the trial phase is gas} \end{cases} \quad \text{(A2)}$$

4. Repeat SSI steps:
   4.1. $Y_i^k = \exp[d_i(z) - \ln \varphi_i(Y^{k-1})]$, $i = 1, \ldots, c$.
   4.2. Until $\| \Delta Y \| < \varepsilon_{SSI}$, where $\Delta Y = Y^k - Y^{k-1}$.
5. Repeat Newton steps:
   5.1. Solve the linear system

$$\left[\frac{\partial}{\partial Y_i Y_j} TPD^*(Y^k)\right]_{\substack{i=1,\ldots,c \\ j=1,\ldots,c}} \Delta Y = -\left[\frac{\partial}{\partial Y_i} TPD^*(Y^k)\right]_{i=1,\ldots,c} \quad \text{(A3)}$$

   5.2. until $\| \Delta Y \| < \varepsilon_{NM}$.
6. Evaluate the TPD at the located stationary point: $TPD = -\ln(Y_T)$.

The algorithm does not locate all the stationary points. To ensure that the negative stationary points are detected, Michelsen[3] suggested using two initial guesses for the vapor–liquid stability analysis in which the feed is assumed to be liquid-like or vapor- like, in the hope that these searches will cover at least the negative minima. If the feed is assumed to be liquid-like, the trial phase is then vapor-like and if the feed is assumed to be vapor-like, the trial phase is then liquid-like. The initial guesses for Y_i are selected as given in step 3 in the algorithm above.

Michelsen[3] also recommended use of a transformed iteration variable α_i, such that

$$\alpha_i = 2\sqrt{Y_i} \quad \text{(A4)}$$

The differentiation of TPD^* with respect to α_i becomes

$$\frac{\partial}{\partial \alpha_i} TPD^*(Y) = \frac{\partial}{\partial Y_i} TPD^* \frac{\partial Y_i}{\partial \alpha_i} = \sqrt{Y_i}\,[\ln \varphi_i(Y) + \ln Y_i - d_i(z)] \qquad i = 1, \ldots, c \quad \text{(A5)}$$

and the Hessian elements,

$$\frac{\partial}{\partial \alpha_i}\left[\frac{\partial}{\partial \alpha_j} TPD^*(Y)\right] = \delta_{ij} + \sqrt{Y_i Y_j}\frac{\partial \ln \varphi_i(Y)}{\partial Y_j} + \frac{1}{2}\delta_{ij}[\ln \varphi_i(Y) + \ln Y_i - d_i(z)] \qquad i, j = 1, \ldots, c \quad \text{(A6)}$$

From Eq. 5, the last term in Eq. A6 should vanish at the stationary points. Equation A6 is then reduced to

$$\frac{\partial}{\partial \alpha_i}\left[\frac{\partial}{\partial \alpha_j} TPD^*(Y)\right] = \delta_{ij} + \sqrt{Y_i Y_j}\frac{\partial \ln \varphi_i(Y)}{Y_j} \qquad i, j = 1, \ldots, c \quad \text{(A7)}$$

Step 5.1 in the SSI–Newton algorithm can be replaced by

$$H(\alpha^k)\Delta\alpha = -F(\alpha^k) \quad \text{(A8)}$$

where

$$H = \left[\frac{\partial}{\partial \alpha_i \alpha_j} TPD^*(Y^k)\right]_{\substack{i=1,\ldots,c \\ j=1,\ldots,c}} \quad \text{and} \quad F(\alpha^k) = \left[\frac{\partial}{\partial \alpha_i} TPD^*(Y^k)\right]_{i=1,\ldots,c}$$

The variable Y_i can then be updated from Eq. A4. We did not find the transformation to add computational efficiency to the Newton method. However, it is useful for the quasi-Newton method.

Appendix B

BFGS-update and the BFGS–quasi-Newton algorithm

The approximate Hessian matrix is required to be symmetric and positive definite. Different formulas are developed to update B^{k+1} in terms of B^k, s^k, and g^k. The most successful update formula is that of Broyden–Fletcher–Golfarb–Shanno (BFGS).[16,17] The two-rank BFGS update can be written as

$$B^{k+1} = \left[B + \frac{gg^T}{\langle s, g\rangle} - \frac{Bss^TB}{\langle s, Bs\rangle}\right]^k \quad \text{(B1)}$$

where the superscript k on the right-hand side in Eq. B1 applies to all the variables in brackets. The notation $\langle x, y\rangle = x^T y$ is the scalar product. The BFGS-update satisfies 13, given that

$$\begin{aligned} B^{k+1}s^k &= \left[B + \frac{gg^T}{\langle s, g\rangle} - \frac{Bss^TB}{\langle s, Bs\rangle}\right]^k s^k \\ &= \left[Bs + \frac{gg^Ts}{\langle s, g\rangle} - \frac{Bss^TBs}{\langle s, Bs\rangle}\right]^k \\ &= \left[Bs + \frac{\langle s, g\rangle}{\langle s, g\rangle} g - \frac{\langle s, Bs\rangle}{\langle s, Bs\rangle} Bs\right]^k = g^k \end{aligned} \quad \text{(B2)}$$

Moreover, B^{k+1} inherits the symmetry and positive definiteness properties from B^k.

In the analytical Hessian matrix (Eq. A6), the first term equals the identity matrix and the second term $\sqrt{Y_i Y_j}\{[\partial \ln \varphi_i(Y)]/\partial Y_j\}$ is of low effective rank.[3] The initial Hessian matrix in Eq. B1 is chosen to be equal to the identity matrix, that is, $B^0 = I$ is reasonable. Michelsen[3] recommended resetting the Hessian matrix to the identity matrix after each iteration. Equation B1 then simplifies to

$$B^{k+1} = \left[I + \frac{gg^T}{\langle s, g\rangle} - \frac{ss^T}{\langle s, s\rangle} \right]^k \tag{B3}$$

By using the Sherman–Morrison–Woodbury (SMW) formula, the inverse matrix $\hat{B}^{k+1}$ of B^{k+1} becomes

$$\hat{B}^{k+1} = [B^{k+1}]^{-1} = \left[I + \left(1 + \frac{\langle g, g\rangle}{\langle s, g\rangle} \right) \frac{ss^T}{\langle s, g\rangle} - \left(\frac{sg^T + gs^T}{\langle s, g\rangle} \right) \right]^k \tag{B4}$$

By replacing Eq. B4 in Eq. 12, we obtain

$$\begin{aligned}
[\alpha^{k+1} - \alpha^k] &= -\left[F + \left(1 + \frac{\langle g, g\rangle}{\langle s, g\rangle} \right) \frac{ss^TF}{\langle s, g\rangle} - \left(\frac{sg^TF + gs^TF}{\langle s, g\rangle} \right) \right]^k \\
&= -\left[F + \left(1 + \frac{\langle g, g\rangle}{\langle s, g\rangle} \right) \frac{\langle s, F\rangle}{\langle s, g\rangle} s - \left(\frac{s\langle g, F\rangle + g\langle s, F\rangle}{\langle s, g\rangle} \right) \right]^k \\
&= -\left[F + \frac{1}{\langle s, g\rangle} \left(\langle s, F\rangle + \frac{\langle g, g\rangle\langle s, F\rangle}{\langle s, g\rangle} - \langle g, F\rangle \right) s - \frac{\langle s, F\rangle}{\langle s, g\rangle} g \right]^k \\
&= -\left[F + \frac{1}{\langle s, g\rangle} \left(\langle s - g, F\rangle + \frac{\langle g, g\rangle\langle s, F\rangle}{\langle s, g\rangle} \right) s - \frac{\langle s, F\rangle}{\langle s, g\rangle} g \right]^k
\end{aligned} \tag{B5}$$

Algorithm: BFGS–quasi-Newton

1. Initialize the K-values from Wilson's correlation (Eq. A1).
2. For given z_i, calculate $d_i(z)$ from Eq. 3.
3. Initialize Y_i from Eq. A2.
4. Initialize the variables:

- $\alpha_i^{old} = 2\sqrt{Y_i}$ $(i = 1, \ldots, c)$ from Eq. A4.
- $F_i^{old} = (\alpha_i^{old}/2)[\ln \varphi_i(Y) + \ln Y_i - d_i(z)]$ $(i = 1, \ldots, c)$ from Eq. A5.
- $\alpha_i^{new} = 2 \exp\{[\ln \varphi_i(Y) - d_i(z)]/2\}$ $(i = 1, \ldots, c)$ from Eq. A4 and Eq. 5.

5. Repeat
   - 5.1. For given α^{old}, update Y from Eq. A4: $Y_i = (1/4)(\alpha_i^{old})^2$ $(i = 1, \ldots, c)$.
   - 5.2. Normalize the mole numbers from $y_i = (Y_i/\Sigma_{i=1}^c Y_i)$ and calculate $\ln \varphi_i(Y)$ from the EOS.
   - 5.3. Update the variables F^{new}, s, and g:
     - $F_i^{new} = (\alpha_i^{old}/2)[\ln \varphi_i(Y) + \ln Y_i - d_i(z)]$ $(i = 1, \ldots, c)$.
     - $g = F^{new} - F^{old}$ and set $F^{old} \leftarrow F^{new}$.
     - $s = \alpha^{new} - \alpha^{old}$ and set $\alpha^{old} \leftarrow \alpha^{new}$.
   - 5.4. Apply the BFGS-update from Eq. 14: $\alpha^{new} \leftarrow \alpha^{old} + \tilde{F}(g, s, F^{old})$.
   - 5.5. Compute: $r = 2[TPD^*(Y)/\beta]$ from Eq. 16.
   - 5.6. Until $\| \alpha_{new} - \alpha^{old} \| < \varepsilon$ or $| r - 1 | < 0.2$.
6. Check the sign of $TPD^*(Y)$.

Appendix C

The expression of "a" in the PR-EOS is given by (see, for example, Firoozabadi[7])

$$a = \sum_{i=1}^{c} \sum_{j=1}^{c} z_i z_j a_i^{1/2} a_j^{1/2} (1 - \delta_{ij}) \tag{C1}$$

where z_i $(i = 1, \ldots, c)$ are mole fractions and δ_{ij} is the binary interaction coefficient for components i and j.

The matrix $\bar{\beta} = [1 - \delta_{ij}]_{i,j=1,\ldots,c}$ is symmetric, so it can be factorized into

$$\bar{\beta} = SDS^T \tag{C2}$$

where D is a diagonal matrix with diagonal entries equal to the eigenvalues λ_i $(i = 1, \ldots, c)$, and $S = [q'^{(1)}, q'^{(2)}, \ldots, q'^{(c)}]$ is an orthogonal matrix with the associated eigenvectors $q'^{(i)} = [q'_{i1}, q'_{i2}, \ldots, q'_{ic}]^T$ $(i = 1, \ldots, c)$.

From Eq. C2, we have

$$\bar{\beta}_{ij} = (1 - \delta_{ij}) = \sum_{k=1}^{c} \lambda_k q'_{ki} q'_{kj} \tag{C3}$$

Replacing Eq. C3 in Eq. C1 yields

$$a = \sum_{k=1}^{c} \lambda_k Q_k^2 \tag{C4}$$

where

$$Q_k = \sum_{i=1}^{c} z_i q_{ki} \tag{C5}$$

and $q_{ki} = a_i^{1/2} q'_{ki}$.

A key step in the reduction method is to simplify Eq. C4 by

$$a = \sum_{k=1}^{c} \lambda_k Q_k^2 \approx \sum_{k=1}^{m} \lambda_k Q_k^2 \tag{C6}$$

where λ_k $(k = 1, \ldots, m)$ are the significant eigenvalues where generally $m \ll c$.

Fugacity coefficient and the derivatives

The PR-EOS can be written as

$$Z^3 - [1 - B]Z^2 + [A - 3B^2 - 2B]Z - [AB - B^2 - B^3] = 0 \tag{C7}$$

where $A = ap/(RT)^2$. B was defined in the text.

The expression for the fugacity coefficient is given by

$$\ln \varphi_i(Q) = \frac{B_i}{B}(Z-1) - \ln(Z-B) - \frac{A}{2\sqrt{2}\,B}\left[\frac{2\sum_{j=1}^{c} y_j a_i^{1/2} a_j^{1/2}(1-\delta_{ij})}{a} - \frac{B_i}{B}\right]\ln\frac{Z+(1+\sqrt{2})B}{Z+(1-\sqrt{2})B} \tag{C8}$$

In Eq. C8, we can rewrite the term

$$\sum_{j=1}^{c} y_j a_i^{1/2} a_j^{1/2}(1-\delta_{ij}) = \sum_{k=1}^{c} \lambda_k Q_k q_{ki} \approx \sum_{\alpha=1}^{m} \lambda_\alpha Q_\alpha q_{\alpha i} \tag{C9}$$

Note that $B = Q_M$, $A = [p/(RT)^2]a = [p/(RT)^2]a(Q_i)$ ($i = 1, \ldots, m$), and $Z = Z(A, B)$.

Calculation of Jacobian elements $[\partial \ln \varphi_i(Q)]/\partial Q_\gamma$

Let us first calculate $[\partial \ln \varphi_i(Q)]/\partial\beta$ (that is, $\gamma = M$). We have

$$\frac{\partial \ln \varphi_i(Q)}{\partial B} = \frac{\partial Z}{\partial B}\frac{\partial \ln \varphi_i(Q)}{\partial Z} + \left.\frac{\partial \ln \varphi_i(Q)}{\partial B}\right|_Z \tag{C10}$$

where we can readily find

$$\frac{\partial Z}{\partial B} = \frac{-Z^2 + 2(3B+1)Z + A - 2B - 3B^2}{3Z^2 - 2(1-B)Z + (A - 2B - 3B^2)} \tag{C11}$$

$$\frac{\partial \ln \varphi_i(Q)}{\partial Z} = \frac{B_i}{B} - \frac{1}{Z-B} - \frac{A}{2\sqrt{2}\,B} \times \left[\frac{2\sum_{k=1}^{m}\lambda_k Q_k q_{k,i}}{A} - \frac{B_i}{B}\right]\left(\frac{2\sqrt{2}\,B}{B^2 - 2BZ - Z^2}\right) = \frac{B_i}{B} - \frac{1}{Z-B} - \left(\frac{A}{B^2 - 2BZ - Z^2}\right)\left[\frac{2A_i}{A} - \frac{B_i}{B}\right] \tag{C12}$$

and

$$\left.\frac{\partial \ln \varphi_i(Q)}{\partial B}\right|_{Z=cnt} = -\frac{B_i}{B^2}(Z-1) + \frac{1}{Z-B} + \frac{(AB_i - 2BA_i)Z}{B^2(-B^2 + 2BZ + Z^2)} + \frac{(BA_i - AB_i)}{\sqrt{2}\,B^3}\ln\frac{Z+(1+\sqrt{2})B}{Z+(1-\sqrt{2})B} \tag{C13}$$

with $A_i = \sum_{k=1}^{m} \lambda_k Q_k q_{ki}$.

Similarly, the derivatives $[\partial \ln \varphi_i(Q)]/\partial Q_\gamma$ ($\gamma = 1, \ldots, m$) are

$$\frac{\partial \ln \varphi_i(Q)}{\partial Q_\gamma} = \frac{\partial \ln \varphi_i(Q)}{\partial Z}\frac{\partial Z}{\partial Q_\gamma} + \left.\frac{\partial \ln \varphi_i(Q)}{\partial Q_\gamma}\right|_Z \tag{C14}$$

where

$$\frac{\partial Z}{\partial Q_\gamma} = \frac{\partial Z}{\partial A}\frac{\partial A}{\partial Q} = \frac{(B-Z)}{3Z^2 - 2(1-B)Z + (A - 2B - 3B^2)}(2\lambda Q_\gamma) \tag{C15}$$

The derivative $[\partial \ln \varphi_i(Q)]/\partial Z$ is given in Eq. C12. The remaining derivative

$$\left.\frac{\partial \ln \varphi_i(Q)}{\partial Q_\gamma}\right|_{\substack{Z \\ \gamma=1,\ldots,m}}$$

is given by

$$\left.\frac{\partial \ln \varphi_i(Q)}{\partial Q_\gamma}\right|_Z = -\frac{1}{2\sqrt{2}\,B} \times \left[2\lambda_\gamma q_{\gamma,i} - \frac{2\lambda_\gamma Q_\gamma B_i}{B}\right]\ln\frac{Z+(1+\sqrt{2})B}{Z+(1-\sqrt{2})B} \tag{C16}$$

Appendix D

Algorithm: reduction method

1. Initialize the K-values from Wilson's correlation (Eq. A1).
2. For given z_i, calculate $d_i(z)$ from Eq. 3.
3. Initialize Y_i from Eq. A2.
4. Repeat steps 4.1 and 4.2 twice.
   - 4.1. $Q_\alpha = (1/Y_T)\sum_{i=1}^{c} Y_i q_{\alpha i}$ ($\alpha = 1, \ldots, M$) from Eq. 24.
   - 4.2. $Y_i = \tilde{d}_i/\varphi_i(Q)$ ($i = 1, \ldots, c$) from Eq. 23.
5. Repeat
   - 5.1. Calculate the Jacobian matrix and solve the system.

$$\left[\frac{h_\alpha(Q)}{\partial Q_\gamma}\right]_{\alpha,\gamma=1,\ldots,M} \Delta Q \equiv -[h_\alpha(Q)]_{\alpha=1,\ldots,M}$$

   - 5.2. Check for feasibility:
     - 5.2.1. If $\min_{i=1,\ldots,c} q_{\alpha i} \le Q_\alpha + \Delta Q_\alpha \le \max_{i=1,\ldots,c} q_{\alpha i}$ ($\alpha = 1, \ldots, M$) are satisfied
       Update Q from: $Q_\alpha = Q_\alpha + \Delta Q_\alpha$ ($\alpha = 1, \ldots, M$).
     - 5.2.2. Otherwise,
       Update Q from: $Q_\alpha = (1/Y_T)\sum_{i=1}^{c} Y_i q_{\alpha i}$ ($\alpha = 1, \ldots, M$).
   - 5.3. Update Y from: $Y_i = \tilde{d}_i/\varphi_i(Q)$ ($i = 1, \ldots, c$).
   - 5.4. Until $\| \Delta Q \| < \varepsilon$.
6. Check the sign of $TPD(Y)$.

Manuscript received Sept. 26, 2005, and revision received Apr. 18, 2006.