Volume 241, issues 1–2, March 15 2006 ISSN 0378-3812

ELSEVIER

FLUID PHASE EQUILIBRIA

AN INTERNATIONAL JOURNAL

Special Issue
A Festschrift in Honor of John M. Prausnitz

Editors:
J.P. O'Connell
P.T. Cummings

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Fluid Phase Equilibria 241 (2006) 186–195

FLUID PHASE EQUILIBRIA

www.elsevier.com/locate/fluid

An efficient and robust algorithm for the calculation of gas–liquid critical point of multicomponent petroleum fluids

Hussein Hoteit, Erik Santiso, Abbas Firoozabadi *

Research Engineering Research Institute (RERI), 385 Sherman Av. Suite 5, Palo Alto, CA 94306, USA

Received 28 August 2005; received in revised form 1 December 2005; accepted 2 December 2005
Available online 24 January 2006

Abstract

The calculation of critical points in petroleum fluids and other multicomponent mixtures is important for practical and theoretical reasons. For certain applications, there is a need for the calculation of critical point as many 10^8 times with the change in composition. There is currently no algorithm that enjoys both robustness and efficiency (that is, speed) to be used for such a purpose. In this work, we propose a simple algorithm that has remarkable efficiency and robustness. It is orders of magnitude faster than the existing algorithms which are known to be robust.

Keywords: Thermodynamics; Critical point calculation; Multicomponent hydrocarbons; Criticality in mixtures; Brent's method in thermodynamics

1. Introduction

John Prausnitz and the senior author co-taught annually an advanced course on thermodynamics of reservoirs fluids in the 1990–2000 period. The participants were scientists and engineers from the petroleum industry worldwide, most with advanced degrees. This was an opportunity for the exchange of ideas and to watch how John has shaped the mechanics of phase equilibria in mixtures and to learn from his approach in solving complicated problems often definitively and the presentation of solution with unique clarity. John has contributed in most areas of phase equilibria including the subject of this article. Jiang and Prausnitz [1] used an equation of sate with renormalization-group theory correction to calculate the critical point of hydrocarbon mixtures. Their major goal was the accuracy of the calculated critical point. There are various motivations in the calculation of gas–liquid critical point in multicomponent mixtures. The motivation for our work is the use of the critical temperature to recognize the gas and liquid states in single phase using a highly efficient computation scheme (see Fig. 1). In this pressure–temperature plot for a fixed composition, the fluid to the right of T_c (in the single phase) can be defined as a gas and the fluid to the left of T_c (in the single phase) can be referred to as a liquid.

This type of phase recognition is crucial in the calculation of the flux between adjacent numerical grids in multicomponent gas–liquid flow simulation in porous media. When the two fluids in the adjacent numerical grids are in the same phase state, the flux is given by the pressure gradient of the same phase. When the two fluids are different (that is, in one grid cell there is gas, in the other grid cell there may be gas and liquid), then the relative permeability concept should be invoked and different phase pressure gradients have to be used. In such type of calculation, with say 10,000 to 1,000,000 grid cells and with 1000 time steps, there may be a need to perform 1,000,000 to over 100,000,000 critical point calculations (assuming that we are interested in determining the phase state in 10% of gridcells). The computational time is, therefore, enormous for an inefficient computational scheme. In addition, the procedure for the calculation of the critical point should also be robust. The purpose of this work is to develop a highly efficient and robust method for the calculation of the critical point in multicomponent mixtures. This is a key requirement in compositional modeling of hydrocarbon reservoirs [2]. In the following, after a brief literature review on the calculation of critical point(s) in multicomponent mixtures, we define the problem for the purpose of this work.

* Corresponding author. Tel.: +1 650 326 9259; fax: +1 650 326 9277.
E-mail addresses: AF@rerinst.org, abbas.firoozabadi@yale.edu (A. Firoozabadi).

0378-3812/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.fluid.2005.12.019

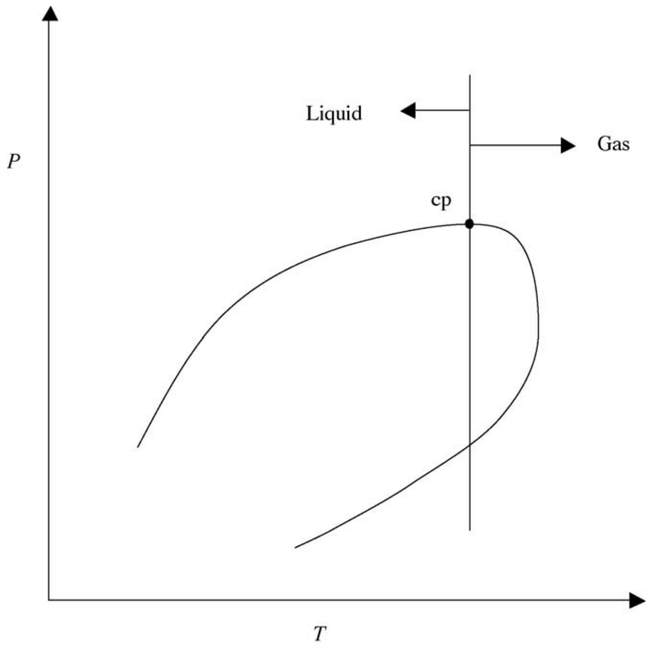

Fig. 1. Definition of gas and liquid phases.

The calculation of gas–liquid critical point for multicomponent mixtures was first presented in 1977 by Peng and Robinson [3]. These authors used the criteria of criticality derived from

$$y^{(c)}_{c+1,c+1} = 0, \tag{1}$$

$$y^{(c)}_{c+1,c+1,c+1} = 0, \tag{2}$$

in the form of two determinants based on the Gibbs free energy G. In Eqs. (1) and (2), $y^{(c)}$ is the cth Legendre transformation of $y^{(0)}$, and the index "$c+1$" represents the derivative of $y^{(c)}$ with respect to variable $(c+1)$. Details are presented in Firoozabadi [4]. Shortly after the work of Peng and Robinson, Baker and Luks [5] used the Helmholtz free energy function A instead of the Gibbs free energy function G to calculate the critical point. In the same year, Heidemann and Khalil [6] introduced a new procedure for the calculation of the critical point. The new approach, which will be described in the next section, requires numerical solution of two 1D nonlinear equations. Heidemann and Khalil used the Newton–Raphson method for the solution of the two nonlinear equations. There have been a number of other authors who have introduced various alternatives for the calculation of the gas–oil critical points [7,8]. Stradi et al. used the formulation of Heidemann and Khalil but instead of the Newton–Raphson method, they applied an interval Newton/generalized bisection technique to find the solution(s) of the nonlinear equations. The technique avoids the need for an initial guess near enough to the solution of nonlinear equations required by the Newton–Raphson method, but at the expense of computational efficiency. For a ternary mixture, the CPU time on the Sun Ultra 30 workstation is reported to be 30–3000 s (Stradi et al.). The purpose of Stradi et al. was to ensure locating all critical points regardless of the CPU cost of the algorithm. For multicomponent petroleum fluid mixtures with 10–15 components, the CPU time required by the algorithm is expected to be excessive. To the best of our knowledge, the critical-point calculation methods have not been used for the purpose of phase recognition (which requires the critical temperature) in compositional hydrocarbon numerical simulation. Here, the main requirements are computational speed and robustness. One may need to calculate the critical temperature with known compositions some 10^6–10^8 times as was mentioned above. Failure to converge in a single case seriously hampers the simulation or causes it to fail. Current methods are mainly based on decreasing the pressure in some increments and carrying out phase-split calculations. These ad hoc methods are neither efficient nor robust. The purpose of our work is the calculation of gas–liquid critical temperature(s) in multicomponent mixtures (as well as critical pressure and volume) in an efficient and robust manner.

In the following, after mathematical formulation of the problem, we present the results for several examples of various complexities and analyze the performance of our method, then draw conclusions from the work.

2. Mathematical formulation

The criteria for criticality can be expressed in different ways. Heidemann and Khalil expressed the criticality condition in the following forms [6,4],

$$\sum_{i=1}^{c}\sum_{j=1}^{c}(\mathcal{A}_{ij})(\Delta N_i \Delta N_j) = 0, \tag{3}$$

$$\sum_{i=1}^{c}\sum_{j=1}^{c}\sum_{k=1}^{c}(\mathcal{A}_{ijk})(\Delta N_i \Delta N_j \Delta N_k) = 0, \tag{4}$$

where $\mathcal{A}_{ij} = (\partial^2 \mathcal{A}/\partial N_i \partial N_j)$ and $\mathcal{A}_{ijk} = (\partial^3 \mathcal{A}/\partial N_i \partial N_j \partial N_k)$; $\mathcal{A}$ is the Helmholtz free energy.

The condition given by Eq. (3) can be expressed as:

$$\det(H) = 0, \tag{5}$$

and

$$H\Delta N = 0, \tag{6}$$

where H is the $c \times c$ Hessian matrix with elements $\mathcal{A}_{ij}$ given above, and ΔN is the column vector of nonzero perturbation in the mole numbers $(\Delta N_1, \Delta N_2, \ldots, \Delta N_c)$. The derivatives in the above equations are at constant T, V, and mole numbers other than those used in the differentiation. There are $(c+2)$ unknowns T_c, V_c, and ΔN_i; $i = 1, \ldots, c$. There are also $(c+2)$ equations given by Eqs. (4)–(6). One of these equations is dependent since the system given by Eq. (6) is singular due to the condition given by Eq. (5). In order to evaluate ΔN it is necessary to impose an additional condition by setting the value of one of its components, say ΔN_1, or the magnitude of the vector ΔN. In this work, our goal is to solve Eqs. (4)–(6) efficiently using a simple approach. After solving for V_c and T_c from Eqs. (4) and (5), the critical pressure can be computed from an equation of state. In this work, the Helmholtz free energy $\mathcal{A}$ is also computed from an equation of state. We use the Peng–Robinson equation of state [15]. The energy parameter A of the PR-EOS, which is given by the following expression:

$$A = \sum_{i=1}^{c}\sum_{j=1}^{c} N_i N_j a_i^{1/2} a_i^{1/2}(1-\delta_{ij}), \tag{7}$$

is calculated from [2]

$$A = \sum_{i=1}^{m} \lambda_i Q_i^2, \tag{8}$$

where Q_i is given by

$$Q_i = \sum_{j=1}^{m} N_j a_j^{1/2} q_{ji}. \tag{9}$$

In Eq. (8), λ_i is the ith significant eigenvalue of the matrix B with elements defined by:

$$B_{ij} = 1 - \delta_{ij}, \tag{10}$$

where δ_{ij} is the binary interaction coefficient. The λ_i's are chosen such that $|\lambda_i| > \varepsilon$. In this work, we set $\varepsilon = 10^{-10}$. The parameter m in Eqs. (8) and (9) is the number of significant eigenvalues of the matrix B, and q_{ij} is the orthogonal matrix with the eigenvectors of B as its columns. This procedure makes the calculation more efficient for multicomponent mixtures because m is often 2–4, while the number of components can be much higher [2].

3. Numerical approach

Our approach for the solution of Eqs. (5), (6) and (4) is similar to that of Heidemann and Khalil [6], but with some important differences in the calculations. Similar to Heidemann and Khalil, we solve the equations by an iterative algorithm that implements two nested one-dimensional searches. For an assigned value of V_c, a temperature T_c is computed so that the Hessian matrix H of A is singular. Then, we compute ΔN, which is the eigenvector corresponding to the zero eigenvalue of H. In the final step, we check to examine that Eq. (4) is satisfied for (V_c, T_c, ΔN).

The step-by-step procedure for the algorithm is as follows:

1. Provide an initial guess for V_c.
   1.1. Provide an initial guess for the critical temperature T_c.
   1.2. For the current V_c and T_c, calculate the elements of H.
   1.3. Obtain the determinant of H. If its value is greater than a given tolerance, assume a new T_c and return to step 1.2. Otherwise continue to step 2.
2. Solve the system of Eq. (6) for ΔN.
3. Calculate the elements A_{ijk}.
4. Evaluate Eq. (4). If its value is greater than a given tolerance, assume a new V_c and return to step 1.1. Otherwise end.

Our main goal in this work is to provide a reliable and efficient method for the different steps in the algorithm.

4. One-dimensional search methods

The critical temperature T_c and critical volume V_c should satisfy the nonlinear Eqs. (5) and (4), respectively. For a given value of V_c, the zero(s) of Eq. (5) can be evaluated by using a 1D Newton–Raphson method. However, this approach may not be efficient from a computational point of view because the calculation of the derivative of Eq. (5) may require excessive computational time. The Newton–Raphson method may also fail to converge; that would hamper the robustness of the algorithm.

Many methods can be used to find the zero(s) of nonlinear functions without the need to evaluate the function derivative. Indeed, each method has some advantages and disadvantages. A reliable method such as the bisection method is often too slow, whereas, faster methods like Regula Falsi or secant methods may not converge. To have both efficiency and robustness, a hybrid method based on Brent's idea [9] can be very efficient. The Brent approach combines three methods:

1. Bisection method.
2. Secant method.
3. Inverse quadratic method.

The bisection method starts from an interval where the function changes sign and refines this interval by evaluating the function at the interval's midpoint. Then, by replacing whichever limit has the same sign, it can halve the size of the bracketing interval in each iteration. Despite the fact that the bisection method is slow, its convergence is, however, guaranteed even for non-smooth functions.

The secant method which is closely related to the Newton–Raphson method replaces the function derivative by a finite difference approximation. Generally, it is faster than the bisection method which only has a linear rate of convergence. The secant method has a superlinear convergence rate of order ∼1.62 [14].

In the inverse quadratic method, the idea is to use three prior points of the function to fit a quadratic polynomial. The zero of this polynomial is taken at the next estimate of the sought root. The inverse quadratic method has a superlinear convergence rate of order ∼1.84 [14].

The secant and inverse quadratic methods may not converge unless the starting point is close to the solution. In our work, we take advantage of the reliability of the bisection method and the speed of secant and inverse quadratic methods.

The procedure used to evaluate T_c, the zero(s) of Eq. (5) is as follows:

1. Obtain an initial bracketing interval $[T_{c,\,\mathrm{inf}}, T_{c,\,\mathrm{sup}}]$ (this step will be discussed later).
2. Evaluate the objective functions $f_{T,\,\mathrm{inf}}$ and $f_{T,\,\mathrm{sup}}$ corresponding to $T_{c,\,\mathrm{inf}}$ and $T_{c,\,\mathrm{sup}}$.
3. Set $T_c = T_{c,\,\mathrm{inf}}$ and $f_T = f_{T,\,\mathrm{inf}}$.
4. Obtain a new estimate of T_c by using one of the following steps:

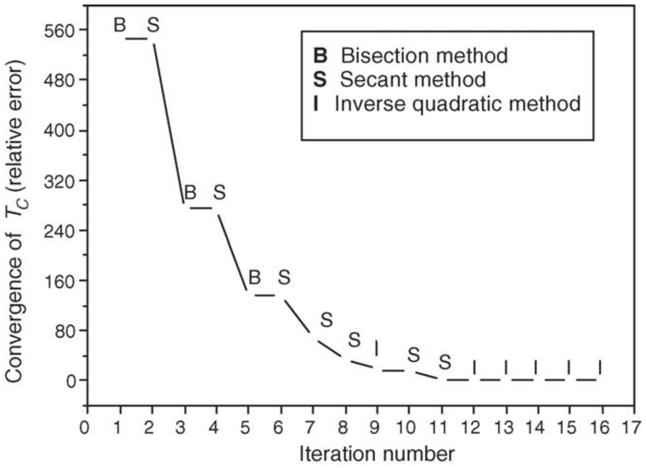

Fig. 2. Switch between the three methods on each iteration.

4.1. Evaluate T_c by using the inverse quadratic method. If T_c belongs to the bracketing interval, go to step 5.
4.2. Evaluate T_c by using the secant method. If T_c lies in the bracketing interval, go to step 5.
4.3. Evaluate T_c by using the bisection method.

5. Evaluate the objective function f_T corresponding to T_c.
6. Update the bracketing interval by checking the sign of f_T.

If the relative difference between $T_{c,\,inf}$ and $T_{c,\,sup}$ is greater than a predefined tolerance, go back to step 4. Otherwise end.

With the above scheme the algorithm switches between fast and robust methods. It maintains brackets on the root by checking where the latest approximate will lead. Fig. 2 shows an example where the algorithm switches between the three methods to converge to the critical temperature for a six-component mixture to be discussed later (mixture 28, Table 2). The bisection method is used first where the initial guess may be far from the solution. Then, it may or may not switch to faster methods: the secant or inverse quadratic methods. Note that the algorithm overhead is low since only one evaluation of the function is needed per iteration.

5. Initial guess

The proposed algorithm is based on searching for a sign change at the critical point. In order to carry out the iterations, initial bracketing intervals $[T_{c,\,inf}, T_{c,\,sup}]$ and $[V_{c,\,inf}, V_{c,\,sup}]$ for the critical temperature and critical volume, respectively, must be provided. The interval boundaries can be estimated as:

$$V_{c,inf} = m_{V,inf}b, \quad \text{and} \quad V_{c,sup} = m_{V,sup}b, \tag{11}$$

with

$$B = \sum_{i=1}^{c} N_i \frac{RT_{ci}}{P_{ci}}.$$

$$T_{c,inf} = m_{T,inf}\min(T_{ci}), \quad T_{c,sup} = m_{T,sup}max(T_{ci}), \tag{12}$$

where T_{ci} and P_{ci} are the critical temperature and critical pressure of component i, respectively. The coefficients $m_{T,\,inf}$, $m_{T,\,sup}$, $m_{V,\,inf}$ and $m_{V,\,sup}$ are selected such that the intervals cover all possible physical solutions for the critical temperature and volume. The idea used to find a sign change bracketing interval is to subdivide uniformly the initial interval into n_{sub} subintervals.

The objective function is then evaluated at the boundaries of each subinterval until a sign change occurs.

The procedure used to estimate a bracketing interval for T_c is as follows:

1. Subdivide the interval $[T_{c,\,inf}, T_{c,\,sup}]$ into n_{sub} subintervals by defining a spacing step $\Delta T_c = (T_{c,\,sup} - T_{c,\,inf})/n_{sub}$.
2. Evaluate the objective function f_T corresponding to $T_{c,\,inf}$.
3. Set $T_{c,\,inf} = T_{c,\,inf} + \Delta T_c$ and $f_{T,\,inf} = f_T$.
4. Evaluate the objective function f_T corresponding to $T_{c,\,inf}$.
5. If f_T and $f_{T,\,inf}$ have the same sign, go to step 3. Otherwise end.

A similar procedure is used to obtain an initial bracketing interval for the critical volume. With this algorithm, at most one critical point can be found. However, where there exists more than one critical point, one needs to detect all the zeros of the volume objective function. In this case, we specify all the bracketing intervals that have a sign change of the volume objective function. This can be readily implemented by a simple modification of the above procedure.

6. Algorithm performance

The CPU time is a function of the number of components c. In order to evaluate the CPU performance of the algorithm, we calculate the CPU time for 26 mixtures $\{M_c | c = 2i;\ i = 1, 2, \ldots, 26\}$, where the mixture M_c has c components. Starting from a binary mixture of $C_1(0.9)$; $nC_{10}(0.1)$: (mole fractions), M_c is composed by dividing the binary mixture into a c-component mixture. Numerical results in Fig. 3 show performance of order $O(c^2)$ for the algorithm. The execution time t_c for the mixture M_c is given approximately by:

$$t_c = \frac{c^{2.1}}{16} t_4, \tag{13}$$

where t_4 is the execution time for the four-component mixture. For a 500 MHz Pentium, the CPU time for M_4 is: $t_4 = 0.93$ ms.

Our proposed algorithm has nearly a quadratic scaling with the number of components. It is much faster than the algorithm of Heidemann and Khalil which is of order $O(c^3)$. We believe that the gains in CPU time are due to three main improvements.

1. Use of Brent's method instead of Newton–Raphson method.
2. Use of the LDL^T matrix factorization instead of the Gaussian elimination.
3. Simplified form of the volume objective function from Eq. (4).

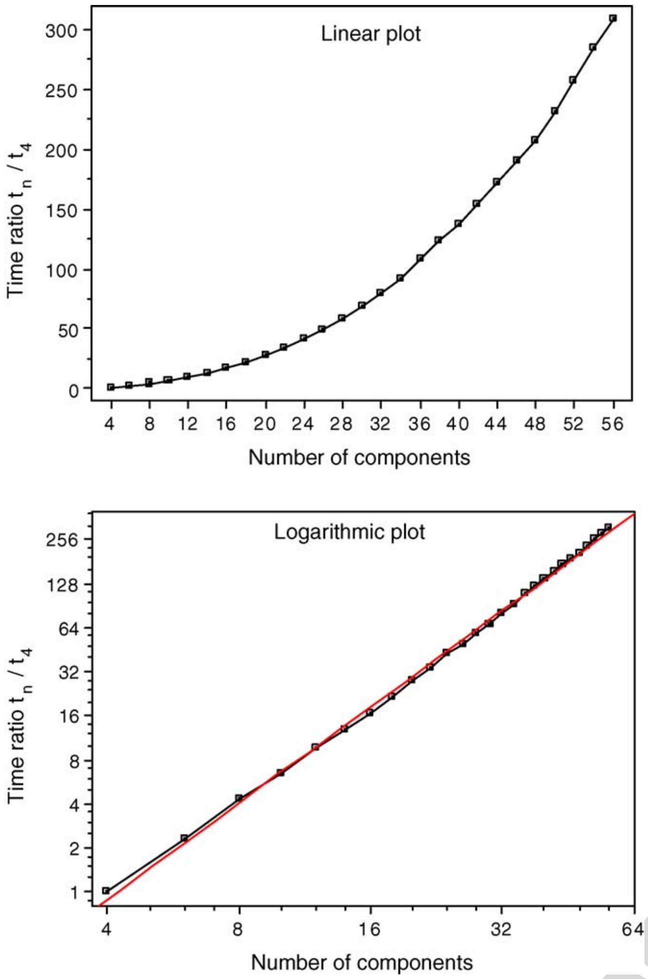

Fig. 3. CPU time as a function of the number of components.

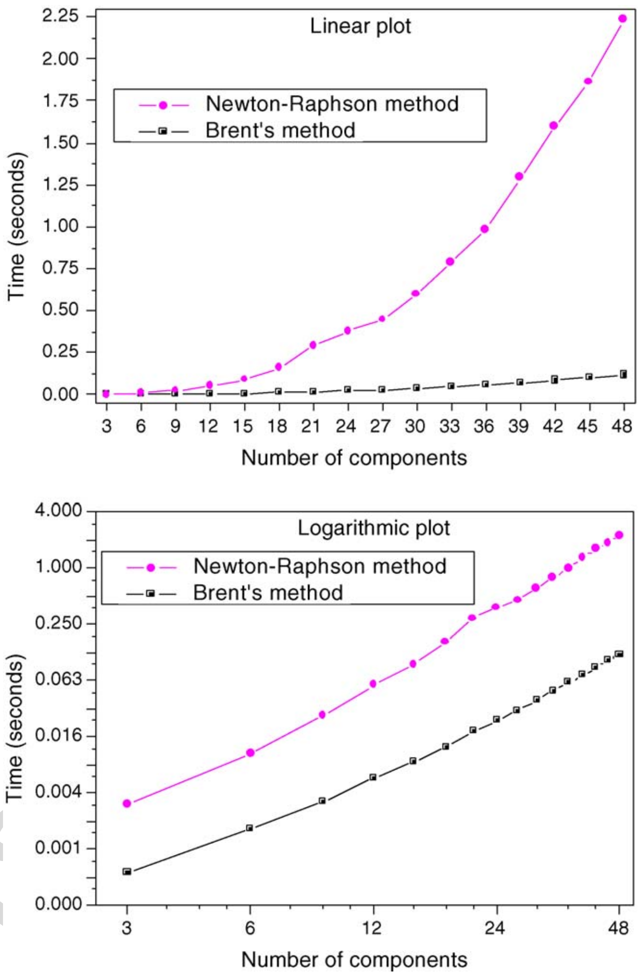

Fig. 4. CPU time performance of the Brent method and the Newton–Raphson method.

Brent's method—our numerical approach uses Brent's method to solve the two nonlinear systems in Eqs. (4) and (5). The Newton–Raphson method, as proposed by Heidemann and Khalil, may be used to approximate the zero of Eq. (4). The Newton–Raphson method converges faster than the local root-finding methods such as the bisection and the secant methods. However, the Newton–Raphson method is more sensitive to the initial guess and to the smoothness of the function. It can be wildly unstable if we start far from the root or if we pass through a point where the function derivative is small. In such cases, it is possible to diverge entirely or to be stranded near an inflection point. Furthermore, it can be also expensive because the function and its derivative have to be evaluated at each iteration. Brent's method overcomes such drawbacks due to the use of the bisection method which guarantees the convergence of the solution.

Our numerical experiments show that Brent's method is more efficient than the Newton–Raphson method. A number of mixtures $\{M_c|\, c=3i;\, i=1, 2, \ldots, 16\}$ are used to compare the CPU times for both methods. These mixtures are composed by dividing a 3-component mixture consisting of C_2(0.429), nC_4(0.373), nC_7(0.198): (mole fractions). Results presented in Fig. 4 show the significant gain in CPU time. The performance of the code with the Newton–Raphson method is found to be of order $O(c^{26})$.

Linear solver—the objective function for the critical temperature, given in Eq. (5), is the determinant of the matrix H. In order to take advantage of the symmetric structure of H, we perform a LDL^{T} factorization with diagonal pivoting: L is a unit lower triangular matrix (L^T is the transpose of L) and D is a diagonal matrix. In this case, the determinant is the product of the diagonal elements of D.

The objective function for the critical volume in Eq. (4) uses the LDL^{T} factorization from the last temperature iteration. If there is a zero pivot in row i, then the diagonal element on that row is set to 1 and the ith right-hand side vector component is set to 1. This is equivalent to assuming a value of 1 for this component of the vector ΔN. Then, ΔN is found by solving lower diagonal and upper linear systems.

The LDL^{T} algorithm is roughly two times faster than the Gauss elimination [10]. In Fig. 5, we compare the execution time of the code by using the two linear solvers, i.e., the Gaussian elimination and the LDL^{T} methods. The gain in the total CPU time of the code due to the latter is about 30%.

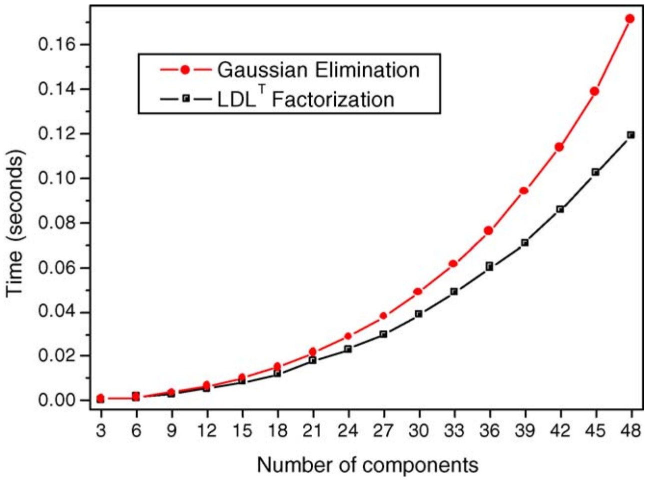

Fig. 5. CPU time performance of LDL^Tfactorization and Gaussian elimination.

Computation of the volume objective function—the cubic form given in Eq. (4) can be simplified to:

$$\sum_{i=1}^{c}\sum_{j=1}^{c}\sum_{k=1}^{c}\mathcal{A}_{ijk}\Delta N_i\Delta N_j\Delta N_k$$
$$= F_1\chi^3 + \left(F_2\sum_{i=1}^{c}\Delta N_i + F_3\sum_{i=1}^{c}A'_i\Delta N_i\right)\chi^2$$
$$+F_4\left(2\sum_{i=1}^{c}\sum_{j=1}^{c}a_{ij}\Delta N_i\Delta N_j + \sum_{i=1}^{c}a_{ij}\Delta N_i^2\right)\chi$$
$$-RT\sum_{i=1}^{c}\frac{\Delta N_i^3}{N_i^2}, \qquad (14)$$

where, $\chi = \sum_{i=1}^{c} b_i\Delta N_i$ and $F_1,\ldots, F_4$ coefficients are independent of the indices i, j and k; they are defined in Appendix A.

This simplified form is very similar to the form given by Stradi et al. [8] and that of Michelsen and Heidemann [11]. However, our simplified form is more efficient because we account for the symmetric form of the double summation:

$$\sum_{i=1}^{c}\sum_{j=1}^{i-1}a_{ij}\Delta N_i\Delta N_j = 2\sum_{i=1}^{c}\sum_{j=1}^{c}a_{ij}\Delta N_i\Delta N_j + \sum_{i=1}^{c}a_{ii}\Delta N_i^2. \quad (15)$$

We note also that the form given by Stradi et al. contains two typos in the first and last terms.

The CPU time required to evaluate Eq. (14) is of order $(5c+c^2/2)$ due to five summations of order c and one double summation of order $c^2/2$. Thus, this simplified form is faster than the non-simplified form in Eq. (4). In order to evaluate the gain in CPU time due to this improvement, the performance of the code is evaluated with the use of simplified and non-simplified forms. In Fig. 6, we note that the execution time for a 48-component mixture is two times faster than the non-simplified form.

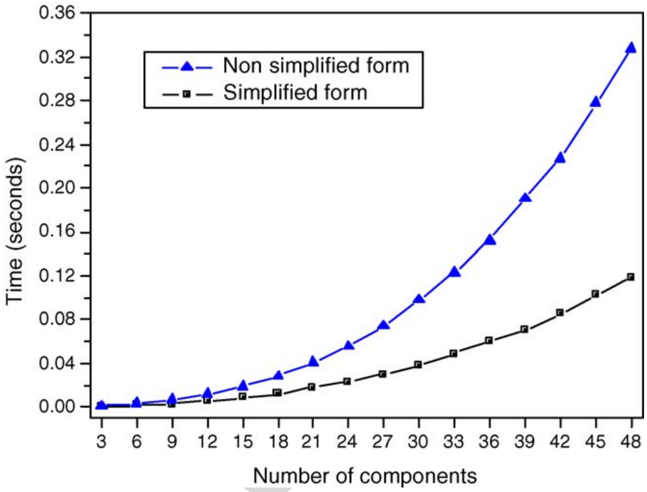

Fig. 6. CPU time performance due to simplification of the volume objective function.

7. Results

We now present the results from the use of the proposed algorithm for a wide variety of mixtures of various degrees of complexity. The mixtures are selected from binaries to petroleum fluids. They include mixtures from Peng and Robinson [3], Heidemann and Khalil [6], Stradi et al. [8], Firoozabadi and Pan [2], Pan and Firoozabadi [13], Ghorayeb et al. [12], and Jiang and Prausnitz [1]. Most of the mixtures have only one positive critical point (in this work, we do not consider the root(s) of the cubic equation for which $v<b$, as a result, we disregard negative critical pressure, see Fig. 3 of Firoozabadi [4]). Some mixtures have no critical point, and some may have more than one critical point.

We used the following convergence criteria for all the mixtures.

- For the inner iterative process that finds the roots of Eq. (5), the relative error between two consecutive steps m and $m+1$ is:

$$\frac{|T^{m+1}-T^m|}{T^m} \leq 10^{-12}$$

It is found that this tolerance has to be sufficiently small in order to ensure the singularity of H.

- The convergence criterion for the cubic form in Eq. (4) can be more relaxed. In all our calculations, we set:

$$\frac{|V^{k+1}-V^k|}{V^k} \leq 10^{-10}$$

where k denotes the steps counter.

When we have no idea about the critical point, the initial bracketing intervals are chosen large enough in order to cover all possible physical solutions. The initial guesses are selected such that:

Table 1
Comparison of our results with those from Stradi et al. [8]

Mixture	Number of components	Critical volume (cm^3/mol)	Critical temperature (K)	Critical pressure (bar)	CPU time from this work (ms)[a]	CPU time from Stradi et al. (ms)[b]
1	2	NCP	NCP	NCP	1.2	1.45×10^4
2	2	54.21	270.2	145.0	0.98	3.5×10^4
		48.02	254.82	149.7		
3	2	31.33	204.74	208.41	1.4	2.64×10^4
		39.39	228.25	235.57		
		61.22	287.49	142.60		
4	3	170.16	394.73	83.02	0.83	6.65×10^5
5	3	214.25	424.84	70.52	0.83	7.82×10^5
6	3	211.25	419.63	69.50	0.89	7.09×10^5

[a] Pentium 500 MHz.
[b] Sun Altra 30.

Table 2
Computed critical points and CPU times for a variety of mixtures.

Mixture	Number of components	Critical volume (cm^3/mol)	Critical temperature (K)	Critical pressure (bar)	CPU time (ms)	Reference
1	3	238.85	440.21	62.96	0.52	Peng and Robinson [3]
2	3	174.79	389.78	74.93	0.53	Peng and Robinson [3]
3	3	206.59	404.46	62.36	0.52	Peng and Robinson [3]
4	3	168.40	395.38	83.82	0.53	Peng and Robinson [3]
5	3	204.24	424.82	73.10	0.55	Peng and Robinson [3]
6	3	200.01	419.37	72.24	0.54	Peng and Robinson [3]
23	5	479.26	540.42	30.79	0.93	Peng and Robinson [3]
24	5	101.14	322.22	144.29	1.21	Peng and Robinson [3]
25	5	101.74	320.39	144.08	1.18	Peng and Robinson [3]
26	5	196.80	394.40	69.83	1.19	Peng and Robinson [3]
27	5	206.85	388.89	55.74	1.22	Peng and Robinson [3]
28	6	189.87	381.34	64.33	1.58	Peng and Robinson [3]
I	10	371.13	571.55	79.32	9.9	Firoozabadi and Pan [2]
II	11	NCP	NCP	NCP	17.9	Firoozabadi and Pan [2]
III	11	81.42	397.73	563.43	9.4	Firoozabadi and Pan [2]
IV	14	86.49	470.18	735.24	7.9	Firoozabadi and Pan [2]
V	15	84.20	455.28	707.48	8.7	Pan and Firoozabadi [13]
1	13	74.60	332.08	401.54	7.7	Ghorayeb et al. [12], Table 5
2	13	151.22	549.19	330.76	6.3	Ghorayeb et al. [12], Table 5
3	13	209.93	618.88	253.90	7.8	Ghorayeb et al. [12], Table 5
4	13	NCP	NCP	NCP	27.6	Ghorayeb et al. [12], Table 5
5	13	NCP	NCP	NCP	29.2	Ghorayeb et al. [12], Table 5
6	5	NCP	NCP	NCP	4.0	Heidemann and Khalil [6]

NCP: no critical point was found.

- volume interval = [1.01b, 4b];
- temperature interval = [0.5 min(T_{ci}), 1.5 max(T_{ci})].

Next we present the calculated critical points and the CPU time for a variety of mixtures.

In Table 1, our results computed for binary and ternary mixtures are compared to those obtained by Stradi et al. [8]. The results for the other mixtures are presented in Table 2. The composition for the mixtures of Table 1 is shown in Table 3. The critical properties and binary interaction parameters for components of mixtures from Table 1 and some of the components from Table 2 mixtures are shown in Table 4. Some of these mixtures have no critical points and some have one, two or three critical points. The computed critical pressure, critical temperature, and critical volume presented in Table 1 show good agreement with those obtained by Stradi et al. [8]. However, the CPU time required to solve these problems using our approach is four to five orders less than the approach used by Stradi et al. (see Table 1). Note that the speed of the two machines is comparable. The method used by Stradi et al. is based on an interval Newton/generalized bisection algorithm. This algorithm guar-

Table 3
Composition (mole%) for mixtures 1–6 of Peng and Robinson [3] from Table 2

Mixture	H_2S	C_1	C_2	C_5	C_7
1	25	75	–	–	–
2	48	52	–	–	–
3	51	49	–	–	–
4	–	–	80.1	6.4	13.5
5	–	–	61.2	27.1	11.7
6	–	–	61.5	29.6	8.9

Table 4
Critical properties and interaction parameters for mixtures 1–6 of Peng and Robinson [3] and additional components from mixtures of Table 2

Comp.	T_c (K)	P_c (bar)	ω	Interaction parameters			
				CO_2-Comp$_i$	H_2S-Comp$_i$	C_5-Comp$_i$	C_7-Comp$_i$
C_1	190.6	45.4	0.008	0.09	0.08	–	–
C_2	305.4	48.2	0.098	0.13	0.06	0.0078	0.0067
C_3	369.8	41.9	0.152	0.12	0.06	–	–
C_5	469.7	33.6	0.256	0.12	0.06	–	0.0074
C_7	540.2	27.3	0.351	0.12	0.06	0.0074	
H_2S	304.2	72.8	0.225	0.10	–	–	–
CO_2	373.2	88.2	0.100	–	0.10	–	–

antees to find all the critical points but at the expense of high CPU requirement. Moreover, it is expected that the CPU time becomes very excessive as the number of components increases.

The strategy used in our algorithm to detect all the critical points is to discretize the initial volume interval into n_{sub} subintervals. The length of each subinterval is approximately equal to $\Delta v = 3b/n_{sub}$. If the distance between the two critical volumes is less than Δv, then one of them will be missed. However, one can choose n_{sub} large enough so that all the critical points can be detected. Generally, $n_{sub} = 5$ was sufficient to detect the critical points for all mixtures given in Tables 1 and 2. However, n_{sub} is taken 50 for mixtures where no critical point was found (mixture 1 in Table 1, mixture II, and mixtures 4, 5 and 6 of Ghorayeb et al. in Table 2). Indeed, higher n_{sub} requires more computations but the CPU time stays within tens of milliseconds. When there is no critical point, the cubic function given in Eq. (4) does not vanish. Depending on the choice of ΔN (since both ΔN and $-\Delta N$ satisfy Eq. (4)) the cubic function is always strictly positive or strictly negative. Fig. 7 shows the behavior of the cubic function for mixture 1 of Table 1. Alternatively, the zeros of this function are the critical points. The cubic function for mixture 3 of Table 1, which has three critical points, is depicted in Fig. 8.

The results for mixtures 1 through 28 in Table 2 are comparable with the critical point obtained by Peng and Robinson [3]. These are mixtures of 3, 5 and 6 components having one critical point each. The CPU time to compute these critical points is of the order of 1 ms. We also computed the critical points for petroleum fluids with up to 15 components. These mixture are taken from Firoozabadi and Pan [2], Pan and Firoozabadi [13] and Ghorayeb et al. [12]. The composition and the critical properties for these mixtures are given in Tables 5 and 6. As Table 2 shows the CPU time for the calculation of critical point is less than 10 ms for the petroleum fluids which are described by 11–15 component/pseudocomponents. The CPU time is somewhat higher when there is no critical point. Table 2 also includes the CPU time for a 5-component mixture from Heidemann and Khalil [6], with no critical point. We like also comment on the accuracy of critical point cal-

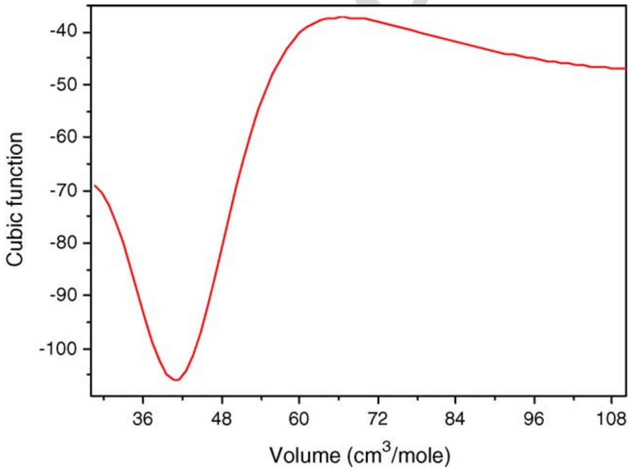

Fig. 7. Behavior of the cubic function for mixture 1 of Table 1.

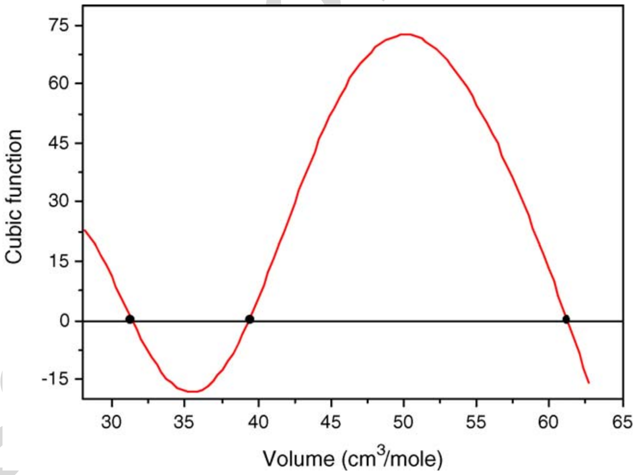

Fig. 8. Behavior of the cubic function for mixture 3 of Table 1.

Table 5
Composition of mixtures 1–5 of Ghorayeb et al. from Table 2

Comp.	Mixture 1	Mixture 2	Mixture 3	Mixture 4	Mixture 5
C_{30+}	0.49	0.70	0.40	0.03	0.01
C_{25}–C_{29}	0.34	1.91	3.89	0.09	0.01
C_{20}–C_{24}	0.59	2.03	3.34	0.16	0.03
C_{16}–C_{19}	0.74	1.82	2.60	0.23	0.07
C_{13}–C_{14}–C_{15}	0.88	1.72	2.23	0.33	0.12
C_{10}–C_{11}–C_{12}	1.26	2.02	2.42	0.58	0.27
C_7–C_8–C_9	2.97	4.04	4.49	1.69	1.00
C_6	0.65	0.80	0.86	0.42	0.29
nC_5–iC_5	0.93	1.08	1.11	0.65	0.48
nC_4–iC_4	2.28	2.48	2.47	1.68	1.33
C_3	3.78	3.89	3.82	3.08	2.67
C_2	8.15	8.06	7.84	7.49	7.05
C_1–CO_2–N_2	76.96	69.45	64.53	83.56	86.66

Table 6
Critical properties and interaction parameters for mixtures 1–5 of Ghorayeb et al. of Table 2

Comp.	T_c (K)	P_c (bar)	ω	Interaction parameters (C_1–CO_2–N_2)-Comp$_i$
C_{30+}	830.00	6.86	1.505	0.024
C_{25}–C_{29}	818.92	11.08	1.399	0.020
C_{20}–C_{24}	778.92	13.28	1.168	0.020
C_{16}–C_{19}	726.96	16.79	0.940	0.019
C_{13}–C_{14}–C_{15}	679.05	20.31	0.776	0.015
C_{10}–C_{11}–C_{12}	622.29	24.53	0.611	0.010
C_7–C_8–C_9	554.13	29.69	0.428	0.009
C_6	506.35	33.96	0.299	0
nC_5–iC_5	469.60	33.30	0.251	0
nC_4–iC_4	425.18	37.99	0.193	0
C_3	369.80	41.90	0.152	0
C_2	305.40	48.83	0.098	0
C_1–CO_2–N_2	190.60	46.00	0.008	–

culations. It widely known that cubic equations of state such as the PR-EOS have deficiency in the critical point region for complicated mixtures such as reservoir fluids. However, we are not aware of suitable equations for such mixtures in the critical region. Table 7 shows that the accuracy of the results from our calculations using the PR-EOS are comparable to the results from the six-component hydrocarbon mixtures by Jiang and Prausnitz [1].

Table 7
Comparison with experimental data and calculated results from Jiang and Prausnitz [1]

Mixture	T_c (K)			P_c (MPa)		
	Data	This work	Jiang and Prausnitz [1]	Data	This work	Jiang and Prausnitz [1]
1	227.6	226.7	230.8	6.89	6.97	7.20
2	255.4	250.5	260.6	8.96	9.09	9.43
3[a]	313.7	320.1	321.7	8.96	8.15	8.87
4	354.3	358.8	350.7	7.64	7.21	7.21
5	331.5	336.1	333.3	9.72	9.02	9.49
6	323.7	331.8	328.4	5.79	5.87	5.83
7	324.5	328.7	326.4	5.48	5.58	5.57
8	310.9	308.8	313.6	10.34	10.07	10.55
9	310.9	310.9	307.4	6.89	6.81	6.67
10	306.3	306.2	306.2	4.90	4.90	4.92
11	307.3	307.2	307.1	4.93	4.93	4.95
12	307.6	307.8	307.8	4.96	4.93	4.97
13	308.6	308.6	308.5	4.96	4.94	4.99
14	397.2	404.3	399.9	5.60	5.54	5.77
15	428.8	430.6	428.0	4.19	4.17	4.25
16	436.3	438.9	435.3	3.85	3.99	3.98
17	442.6	443.8	440.1	3.90	3.95	3.94
18	449.4	452.7	449.4	3.81	3.85	3.85
19	450.2	450.6	450.8	3.88	3.79	3.88
20	405.9	410.6	408.1	5.11	5.06	5.36
21	417.9	419.6	420.3	4.51	4.42	4.66
22	387.0	394.4	391.2	7.22	7.00	7.65
23	385.4	388.8	388.0	5.62	5.57	6.04

[a] Mixture also contains a small amount of nitrogen.

8. Concluding remarks

The work in the past for the calculation of critical point in multicomponent mixtures can be divided into two groups. Those methods which are not robust but enjoy relative computational speed, and methods such as the one of Stradi et al. [8] which guarantee to calculate all the critical points. In this work, we provide a method which is very robust, it is also as reliable as the method of Stradi et al. [8]. Our approach is about five orders of magnitude faster than the approach used by Stradi et al. [8] for the three-component mixtures. It is even faster for multicomponent mixtures with greater number of components due to its more favorable scaling. We expect our approach to be ideal for algorithms that require extensive critical point calculation in multicomponent mixtures. The computational speed is of the same order as vapor–liquid equilibria calculations.

List of symbols

$\mathcal{A}_{ij}$ ($\mathcal{A}_{ijk}$)	second (third) order derivative of the Helmholtz free energy with respect to the number of components i and j (i, j and k)
b	covolume in the EOS
c	number of components
G	Gibbs free energy
m	number of significant eigenvalues
N	number of moles
ΔN_i	perturbation in the number of moles of component i
P_c	critical pressure
R	gas constant
T_c	critical temperature
v	molar volume
V_c	critical volume
V	volume
ω	acentric factor
δ_{ij}	binary interaction parameter between component i and component j
λ_i	ith significant eigenvalue

Acknowledgment

This work was supported by member companies of the Reservoir Engineering Research Institute (RERI). Their support is greatly appreciated.

Appendix A

The coefficients A_{ijk} in the cubic equation Eq. (4) are explicitly derived from the PR-EOS [4]:

$$A_{ijk} = -\frac{RT\delta_{ij}\delta_{jk}}{N_k^2} + \frac{RT}{(V-B)^3}[(V-B)(b_ib_j + b_ib_k + b_jb_k) + 2Nb_ib_jb_k] + \frac{1}{2\sqrt{2}B^4}\Bigg\{[B^2(b_ia_{jk} + b_ja_{ik} + b_ka_{ij}) - B(A'_ib_jb_k + A'_jb_ib_k + A'_kb_ib_j) + 3Ab_ib_jb_k]$$

$$\times \left[2\ln \frac{V+(\sqrt{2}+1)B}{V-(\sqrt{2}-1)B} - \frac{4\sqrt{2}BV}{V^2+2BV-B^2} \right]$$

$$- \frac{4\sqrt{2}B^2V}{(V^2+2BV-B^2)^2} [B(B-V)(A'_i b_j b_k + A'_j b_i b_k$$

$$+ A'_k b_i b_j) + A b_i b_j b_k (3V-2B)]$$

$$\left. - \frac{16\sqrt{2}AB^3(V-B)^2 V b_i b_j b_k}{(V^2+2BV-B^2)^3} \right\},$$

where $B = Nb = \sum_{i=1}^{c} N_i b_i$, $A = N^2 a = \sum_{i=1}^{c}\sum_{j=1}^{c} N_i N_j a_{ij}$, and $A'_i = (\partial A)/(\partial N_i)$.

The coefficients in the triple summation in Eq. (14), $F_1, \ldots F_4$ are independent of the indices i, j and k; they are defined as:

$$F_1 = \frac{2RT}{(V-B)^3} + \frac{3AF_6}{2\sqrt{2}B^4} - \frac{2VA(3V-2B)}{B^2F_5^2} - \frac{8VA(V-B)^2}{BF_5^3},$$

$$F_2 = \frac{3RT}{(V-B)^2},$$

$$F_3 = \frac{3F_6}{2\sqrt{2}B^3} - \frac{6V(B-V)}{BF_5^2},$$

$$F_4 = \frac{3F_6}{2\sqrt{2}B^2},$$

with F_5 and F_6 are given by:

$$F_5 = (V^2 + 2VB - B^2),$$

$$F_6 = 2\ln\frac{V+(\sqrt{2}+1)B}{V-(\sqrt{2}-1)B} - \frac{4\sqrt{2}BV}{F_5}.$$

References

[1] J. Jiang, J.M. Prausnitz, Critical temperature and pressure for hydrocarbon mixtures from an equation of sate with renormalization-group theory corrections, Fluid Phase Equilibria 169 (2000) 127.

[2] A. Firoozabadi, H. Pan, Fast and robust algorithm for compositional modeling: part 1—stability analysis testing, SPE J. 7 (1) (2002) 78.

[3] D. Peng, D. Robinson, A rigorous method for predicting the critical properties of multicomponent systems from an equation of state, AIChE J. 23 (2) (1977) 137.

[4] A. Firoozabadi, Thermodynamics of Hydrocarbon Reservoirs, McGraw-Hill, New York, 1999.

[5] L.E. Baker, K.D. Luks, Critical point and saturation pressure calculations for multicomponent mixtures, SPE J. 15 (1980).

[6] R. Heidemann, A. Khalil, The calculation of critical points, AIChE J. 26 (5) (1980) 769.

[7] N.R. Nagarajan, A.S. Cullick, A. Griewank, New strategy for phase equilibrium and critical point calculations by thermodynamic energy analysis. Part II. Critical point calculations, Fluid Phase Equilibria 62 (1991) 211.

[8] B. Stradi, J. Brennecke, J. Kohn, M. Stadtherr, Reliable computation of mixture critical points, AIChE J. 47 (1) (2001) 212.

[9] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes in FORTRAN: The Art of Scientific Computing, Cambridge University Press, 1992.

[10] G. Golub, C. Van Loan, Matrix Computations, The Johns Hopkins University Press, 1993.

[11] M. Michelsen, R. Heidemann, Calculation of critical points from cubic two-constant equations of state, AIChE J. 27 (1981) 521.

[12] K. Ghorayeb, T. Anraku, A. Firoozabadi, Interpretation of the unusual fluid distribution in the Yufutsu gas-condensate field, SPE J. 8 (2) (2003) 114.

[13] H. Pan, A. Firoozabadi, Fast and robust algorithm for compositional modeling: Part II. Two-Phase Flash Computations, SPE J. 8 (4) (2003) 380.

[14] G. Arfken, Mathematical methods for Physicists, Academic Press, Orlando, FL, 1985.

[15] D. Peng, D. Robinson, A new two constant equation of state, Ind. Eng. Chem. Fund. 15 (1976) 59.