Reliable Computation of Phase Stability and Equilibrium from the SAFT Equation of State

Gang Xu, Joan F. Brennecke and Mark A. Stadtherr*

Department of Chemical Engineering
University of Notre Dame
182 Fitzpatrick Hall
Notre Dame, IN 46556
USA

February 2001
(revised, May 2001)

Keywords: SAFT; Phase Stability; Phase Equilibrium; Computational Method; Interval Analysis.

(for submission to *Industrial & Engineering Chemistry Research*,
special issue in honor of Dr. Hasan Orbey)

*Author to whom all correspondence should be addressed. Fax: (219) 631-8366; E-mail: markst@nd.edu

Abstract

In recent years, molecularly-based equations of state, as typified by the SAFT (statistical associating fluid theory) approach, have become increasingly popular tools for the modeling of phase behavior. However, whether using this, or even much simpler models, the reliable calculation of phase behavior from a given model can be a very challenging computational problem. A new methodology is described that is the first *completely reliable* technique for computing phase stability and equilibrium from the SAFT model. The method is based on interval analysis, in particular an interval Newton/generalized bisection algorithm, which provides a mathematical and computational *guarantee* of reliability, and is demonstrated using nonassociating, self-associating, and cross-associating systems. New techniques are presented that can also be exploited when conventional point-valued solution methods are used. These include the use of a volume-based problem formulation, in which the core thermodynamic function for phase equilibrium at constant temperature and pressure is the Helmholtz energy, and an approach for dealing with the internal iteration needed when there are association effects. This provides for direct, as opposed to iterative, determination of the derivatives of the internal variables.

1 Introduction

In recent years, molecularly-based equations of state (EOS) have become increasingly popular tools for the modeling of phase behavior. These models, as typified by the SAFT (statistical associating fluid theory) approach,[1–8] characteristically have parameters with well-defined physical meaning, based on molecular structure (e.g., chain length) and interactions (e.g., association energy), and thus may be more reliable in making extrapolations to different molecules or thermodynamic conditions. Unfortunately, even when good EOS models are available, it is often very difficult to actually calculate the equilibrium phase behavior reliably from the given models.

Since the number of phases present at equilibrium may not be known *a priori*, the computation of phase equilibrium is often considered in two stages, as outlined by Michelsen.[9,10] The first involves the *phase stability* problem, that is, to determine whether or not a given mixture will split into multiple phases. The second involves the *phase split* problem, that is to determine the amounts and compositions of the phases assumed to be present. After a phase split problem is solved, it is then necessary to do phase stability analysis on the results to determine whether the postulated number of phases was correct, and if not to repeat the phase split problem. Both the phase stability and phase split problems can be formulated as minimization problems, or as equivalent nonlinear equation solving problems. There are several versions of this two-stage approach. The recent version of Lucia et al.[11] has been applied to a wide variety of practical problems and appears to be particularly useful. With the choice of the proper thermodynamic state functions,[12] the two-stage framework can be applied to phase equilibrium problems with various types of specficiations (e.g., constant temperature and pressure, constant temperature and density, isentropic, isenthalpic, etc.).

For determining phase equilibrium at constant temperature and pressure, the case considered here, a minimum in the total Gibbs energy of the system is sought. Phase stability analysis may be interpreted as a global optimality test that determines whether the phase being tested corresponds to a global minimum in the total Gibbs energy of the system. If it is determined that a phase will split, then a phase split problem is solved, which can be interpreted as finding a *local* minimum in the total Gibbs energy of the system. This local minimum can then be tested for global optimality using phase stability analysis. If necessary the phase split calculation must then be repeated, perhaps changing the number of phases assumed to be present, until a solution is found that meets the global optimality test. Clearly the correct solution of the phase stability problem, itself a global optimization problem, is the key in this two-stage global optimization procedure for phase equilibrium. Assuming that the phase stability problem is correctly solved, this type of two-stage procedure can be shown to converge in a finite number of steps to the equilibrium solution (e.g., McKinnon et al.[13]).

Conventional minimization or equation solving techniques for solving the phase stability problem are initialization dependent, and may fail by converging to trivial or nonphysical solutions or to a point that is a local but not a global minimum, even for very simple models (e.g., Green et al.[14]); thus there is no guarantee that the phase stability problem has been correctly solved. Furthermore, two-stage methods that do not solve the phase stability problem to global optimality (e.g., Michelsen,[9,10] Lucia et al.[11]) provide no guarantee that the phase equilbrium problem has been solved correctly. Thus, there has been significant interest in the development of completely reliable methods, as reviewed in more detail by Hua et al.[15] Particularly noteworthy is the work of McDonald and Floudas[16–19] and Harding and Floudas,[20] who have shown that for certain activity coefficient models and cubic EOS models, that the phase stability and equilibrium problems can be made amenable, through the formulation of convex underestimating functions, to solution by powerful global optimization techniques using branch and bound, which provide a mathematical guarantee of reliability.

An alternative approach for solving the phase stability problem is the use of interval analysis. This provides not only a mathematical guarantee of global optimality, but also a computational guarantee,[15] since it deals automatically with the rounding error issues that are inherent in floating-point arithmetic and that lead to the loss of mathematical guarantees. This method, based on an interval Newton/generalized bisection algorithm, was originally suggested by Stadtherr et al.,[21] who applied it to problems involving excess Gibbs energy models, as later done also by McKinnon et al..[13] Subsequently, Hua et al.[22,23] applied this method to problems modeled with cubic equations of state, and efforts were made to improve the computational efficiency of the approach, both for EOS models[15] and for excess Gibbs energy models.[24]

We demonstrate here the use of the interval approach for the completely reliable solution of phase stability and equilibrium problems when the SAFT EOS model is used. Since the SAFT EOS is most conveniently expressed in terms of the Helmholtz energy, we use a "volume-based" formulation of the phase stability and equilibrium problems at constant temperature and pressure, in which the core thermodynamic function is the Helmholtz energy. This problem formulation was developed by Nagarajan et al.,[25] but as noted by Michelsen,[12] its use has not yet been widely investigated. In the course of applying the interval approach, we also address other numerical issues concerning SAFT; in particular, we present an approach for dealing with the "internal" iteration required when association effects are included in the SAFT model. The final result is a new computational procedure, with a *mathematical and computational guarantee* of reliability, for solving phase stability and equilibrium problems when the SAFT EOS model is used.

In the next section, we present the mathematical formulation of the problem, summarizing the SAFT EOS model and the volume-based approach for solving the phase stability and equilibrium problems, as well as providing some additional detail about the two-stage procedure for determining phase equilibrium. In

Section 3, we describe the actual solution methodology used, including a summary of the interval approach that provides the guarantee of reliability. Then, in Section 4, we present results for several test problems, and, in Section 5, summarize the conclusions of this study.

2 Problem Formulation

2.1 SAFT Equation of State

Many details concerning the SAFT EOS and it variants can be found elsewhere.[1–8] A good description of the key concepts underlying SAFT has been provided by Fu and Sandler.[7] In this section we summarize the equations and parameters that make up the SAFT EOS for a mixture of species.

In the SAFT model, molecules of each species are treated as a chain composed of equal-size, spherical segments, with molecules of each different species i having different number of segments, m_i, temperature-independent segment molar volume (at close packing), v_i^{00}, and temperature-independent segment interaction energy, u_i^0. For molecules that may associate, for instance due to hydrogen bonding, an association energy parameter ϵ^{AB} and dimensionless association volume parameter κ^{AB} are used to characterize the interaction between association sites A and B. Values of these parameters for a wide range of compounds are given by Huang and Radosz.[4]

The SAFT EOS is generally expressed in terms of the residual Helmholtz energy per unit mole of mixture a^{res} (this is the Helmholtz energy relative to an ideal gas mixture of the same composition and at the same temperature and density). There are hard sphere, dispersion, chain and association contributions that must be accounted for, so

$$a^{\mathrm{res}} = a^{\mathrm{hs}} + a^{\mathrm{disp}} + a^{\mathrm{chain}} + a^{\mathrm{assoc}}. \tag{1}$$

There are some variations in the SAFT model, depending on the expressions used to represent each of these contributions. We use the model as described by Huang and Radosz.[4–6]

The hard sphere contribution a^{hs} is based on the hard sphere EOS as expressed by Boublík,[26] and is given by

$$\frac{a^{\mathrm{hs}}}{RT} = \frac{6}{\pi N_{\mathrm{A}}\varrho}\left[\frac{(\zeta_2)^3 + 3\zeta_1\zeta_2\zeta_3 - 3\zeta_1\zeta_2(\zeta_3)^2}{\zeta_3(1-\zeta_3)^2} - \left(\zeta_0 - \frac{(\zeta_2)^3}{(\zeta_3)^2}\right)\ln(1-\zeta_3)\right] \tag{2}$$

where

$$\zeta_j = \frac{\pi N_{\mathrm{A}}\varrho}{6}\sum_{i=1}^{N} x_i m_i d_{ii}^j\,, \qquad j = 0, 1, 2, 3. \tag{3}$$

Here ϱ is the total molar density of the mixture, x_i is the mole fraction of compound i in the mixture, N is the number of components in the mixture, N_{A} is Avogadro's number, j is an exponent, and d_{ii} is the temperature-dependent segment diameter given by

$$d_{ii} = \sigma_i \left[1 - 0.12 \exp\left(\frac{-3u^0}{kT}\right)\right], \tag{4}$$

where

$$\sigma_i = \left(v_i^{00} \frac{6\tau}{\pi N_\mathrm{A}}\right)^{1/3}, \tag{5}$$

k is Boltzmann's constant, and $\tau = \pi\sqrt{2}/6$ is the packing fraction for close-packed spheres.

The dispersion term a^disp used by Huang and Radosz[4,5] is

$$\frac{a^\mathrm{disp}}{RT} = m \sum_{i=1}^{4} \sum_{j=1}^{9} D_{ij} \left(\frac{u}{kT}\right)^i \left(\frac{\eta}{\tau}\right)^j. \tag{6}$$

To determine the average segment number m, average segment energy u and average reduced density (packing fraction) η, the van der Waals one-fluid (vdW1) approach, that treats the mixture as a hypothetical pure fluid having the same residual properties as the mixture, is used (as in Eqs. (2–3) above). Thus, with the vdW1 mixing rule,

$$m = \sum_{i=1}^{N} x_i m_i, \tag{7}$$

$$\eta = \frac{\pi N_\mathrm{A} \varrho}{6} \sum_{i=1}^{N} x_i m_i d_{ii}^3 = \zeta_3 \tag{8}$$

and

$$\frac{u}{kT} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j m_i m_j v_{ij}^0 (u_{ij}/kT)}{\sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j m_i m_j v_{ij}^0}, \tag{9}$$

where

$$v_{ij}^0 = \left\{1/2 \left[(v_i^0)^{1/3} + (v_j^0)^{1/3}\right]\right\}^3, \tag{10}$$

$$u_{ij} = (1 - k_{ij})(u_{ii} u_{jj})^{1/2}, \tag{11}$$

$$v_i^0 = v_i^{00} \left[1 - 0.12 \exp\left(\frac{-3u^0}{kT}\right)\right]^3 \tag{12}$$

and

$$u_{ii} = u_i^0 \left(1 + \frac{e}{kT}\right). \tag{13}$$

The D_{ij} are the Chen and Kreglewski[27] constants, which are listed in Table 1, and $e/k = 10$, with some exceptions as discussed by Huang and Radosz.[4] The k_{ij} are binary interaction parameters that can be fit to experimental data. It should be noted that under the one-fluid approach the upper limit on the reduced density η will be given by the closest packing of equal diameter spheres, which is the reduced density given by $\tau = \pi\sqrt{2}/6$. It should also be noted that different versions of SAFT, for instance the SSAFT (simplified

SAFT) model of Fu and Sandler,[7] may use different expressions for the dispersion term. Furthermore, different mixing rules may be used; for example, Huang and Radosz[5] also discuss mixing rules based on volume fraction. While we use the expressions given above for the dispersion term, the computational method described here is general purpose and can be applied in connection with any variation of the SAFT EOS.

The chain term, a^{chain}, as derived by Chapman et al.,[3] is based on the pair correlation function, $g_{ii}^{\mathrm{hs}}(d_{ii})$, in the form of Boublík,[26] and is given by

$$\frac{a^{\mathrm{chain}}}{RT} = \sum_{i=1}^{N} x_i(1-m_i)\ln\left(g_{ii}^{\mathrm{hs}}(d_{ii})\right) \tag{14}$$

where

$$g_{ii}^{\mathrm{hs}}(d_{ii}) = \frac{1}{1-\zeta_3} + \frac{3d_{ii}}{2}\frac{\zeta_2}{(1-\zeta_3)^2} + 2\left[\frac{d_{ii}}{2}\right]^2 \frac{\zeta_2^2}{(1-\zeta_3)^3}. \tag{15}$$

This is a special case of the more general pair correlation function,[26] needed below, which is

$$g_{ij}^{\mathrm{hs}}(d_{ij}) = \frac{1}{1-\zeta_3} + \frac{3d_{ii}d_{jj}}{d_{ii}+d_{jj}}\frac{\zeta_2}{(1-\zeta_3)^2} + 2\left[\frac{d_{ii}d_{jj}}{d_{ii}+d_{jj}}\right]^2 \frac{(\zeta_2)^2}{(1-\zeta_3)^3} \tag{16}$$

with d_{ij} representing $d_{ii}d_{jj}/(d_{ii}+d_{jj})$.

Finally, the association term, a^{assoc}, is written as

$$\frac{a^{\mathrm{assoc}}}{RT} = \sum_{i=1}^{N} x_i \left[\sum_{\mathrm{A}_i}\left[\ln X^{\mathrm{A}_i} - \frac{X^{\mathrm{A}_i}}{2}\right] + \frac{M_i}{2}\right], \tag{17}$$

where the summation over A_i indicates summation over all association sites on component i. Here M_i is the number of association sites on component i, and X^{A_i} is the mole fraction of molecules of i which are *not* bonded at the association site A_i. This is determined from

$$X^{\mathrm{A}_i} = \left[1 + N_{\mathrm{A}}\sum_{j=1}^{N}\sum_{\mathrm{B}_j}\rho_j X^{\mathrm{B}_j}\Delta^{\mathrm{A}_i\mathrm{B}_j}\right]^{-1}, \tag{18}$$

where the summation over B_j indicates summation over all association sites on component j. Note that this means that X^{A_i} is, in general, given only implicitly by Eq. (18), since X^{A_i} appears not only on the left-hand side, but also in one of the terms in the double summation on the right-hand side. Thus, when there are association effects to be accounted for, evaluation of the SAFT EOS requires an "internal" iteration to solve for the "internal" variables X^{A_i}, except in some special cases in which it is possible to solve for these variables explicitly. In Eq. (18) the association strength function $\Delta^{\mathrm{A}_i\mathrm{B}_j}$ is

$$\Delta^{\mathrm{A}_i\mathrm{B}_j} = g_{ij}^{\mathrm{hs}}(d_{ij})\left[\exp\left(\epsilon^{\mathrm{A}_i\mathrm{B}_j}/kT\right) - 1\right]\sigma_{ij}^3\kappa^{\mathrm{A}_i\mathrm{B}_j} \tag{19}$$

with

$$\sigma_{ij} = (\sigma_i + \sigma_j)/2. \tag{20}$$

The SAFT parameters $\epsilon^{\mathrm{A}_i\mathrm{B}_j}$ and $\kappa^{\mathrm{A}_i\mathrm{B}_j}$ are the association energy and dimensionless association volume, respectively, used to characterize interactions between sites A_i and B_j. In a mixture, there may be self-association interactions ($i = j$) involving the same types of sites ($\mathrm{A} = \mathrm{B}$) or different types of sites on molecules of the same species, and there may be cross-association interactions ($i \neq j$) between sites on molecules of different species.

Eqs. (1), (2), (6), (14) and (17) provide an expression for the SAFT EOS in terms of the molar residual Helmholtz energy $a^{\mathrm{res}}(\mathbf{x}, \varrho)$ given (at constant temperature) as a function of composition (mole fraction) $\mathbf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ and total molar density ϱ. In pressure-explicit form, the EOS may be written

$$P = \varrho^2 \left(\frac{\partial a^{\mathrm{res}}(\mathbf{x}, \varrho)}{\partial \varrho} \right)_{\mathbf{x},T} + \varrho RT \tag{21}$$

where the last term is the ideal gas contribution. For ease of dealing with the function $a^{\mathrm{res}}(\mathbf{x}, \varrho)$ and its derivatives, the parameterization given by Huang and Radosz,[5,6] following the approach of Topliss,[28] is very useful.

2.2 Phase Stability Problem

As discussed above, the determination of phase stability, i.e., whether or not a given mixture can split into multiple phases, is a key step in phase equilibrium computations, providing a global optimality test for the minimization of the total Gibbs energy. The determination of phase stability is typically done using tangent plane analysis. For a mixture at constant T, P, and composition (mole fractions) $\mathbf{x}_0$, the tangent plane condition is usually expressed[9,29] in terms of the molar Gibbs energy of the mixture as a function of molar composition and volume (or density). The mixture is not stable if the molar Gibbs energy surface $g(\mathbf{x}, \varrho)$ ever falls below a plane tangent to the surface at $\mathbf{x}_0$.

However, since the SAFT EOS is most conveniently expressed in terms of the Helmholtz energy, as detailed above, we choose here to use a “volume-based” formulation of tangent plane analysis[25] in which the Helmholtz energy is the core function. For this case, the tangent plane condition is expressed in terms of the Helmholtz energy density (Helmholtz energy per unit volume of mixture) as a function of the molar component density vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)^{\mathrm{T}}$. A mixture (the “feed” or “test phase”) at constant T, P, and composition (molar densities) $\boldsymbol{\rho}_0$ is not stable if the Helmholtz energy density surface $\tilde{a}(\boldsymbol{\rho})$ ever falls below a plane tangent to the surface at $\boldsymbol{\rho}_0$. That is, if the tangent plane distance function

$$D(\boldsymbol{\rho}) = \tilde{a}(\boldsymbol{\rho}) - [\tilde{a}(\boldsymbol{\rho}_0) + \nabla\tilde{a}(\boldsymbol{\rho}_0) \cdot (\boldsymbol{\rho} - \boldsymbol{\rho}_0)] \tag{22}$$

is negative for any composition $\boldsymbol{\rho}$, the mixture is not stable.[25] Here ∇ is used to indicate the gradient vector, and the Helmholtz energy density $\tilde{a}(\boldsymbol{\rho})$ can be obtained from the molar residual Helmholtz energy $a^{\text{res}}(\mathbf{x}, \varrho)$ given by Eq. (1) by first defining the residual Helmholtz energy density $\tilde{a}^{\text{res}}(\boldsymbol{\rho}) = \varrho a^{\text{res}}(\mathbf{x}, \varrho)$ and then adding the ideal gas energy of mixing term, yielding

$$\tilde{a}(\boldsymbol{\rho}) = \tilde{a}^{\text{res}}(\boldsymbol{\rho}) + RT \sum_{i=1}^{N} \rho_i \ln [\rho_i RT]. \tag{23}$$

To simplify dealing with the function $\tilde{a}(\boldsymbol{\rho})$ and its derivatives, we use a parameterization of $\tilde{a}^{\text{res}}(\boldsymbol{\rho})$ similar to the parameterization that Huang and Radosz[5,6] used for $a^{\text{res}}(\mathbf{x}, \varrho)$. This new parameterization is given in Appendix A.

To determine if $D(\boldsymbol{\rho})$ ever becomes negative, an unconstrained minimization of $D(\boldsymbol{\rho})$ can be done. Clearly, it is important to find the *global* minimum, or one may incorrectly conclude that the mixture is stable. Alternatively, this may be treated as an equation solving problem. It is easily seen that the stationary points in the optimization problem can be found by solving the nonlinear equation system

$$\nabla \tilde{a}(\boldsymbol{\rho}) - \nabla \tilde{a}(\boldsymbol{\rho}_0) = \mathbf{0}. \tag{24}$$

This equation system has a trivial root at $\boldsymbol{\rho} = \boldsymbol{\rho}_0$ and, because $\tilde{a}(\boldsymbol{\rho})$ is likely to be nonconvex, may have multiple nontrivial roots as well, corresponding to multiple local extrema in the optimization problem. If this approach is used, it is important that no roots of the equation system be missed, since that may result in failure to find the one corresponding to the global minimum. Thus, whether solved directly by some minimization algorithm, or indirectly by some nonlinear equation solver, this can be a very difficult problem to solve reliably. As explained in more detail below, we will use here an interval-based methodology that provides a mathematical and computational guarantee that no roots of Eq. (24) will be missed, thus ensuring that the global minimum in $D(\boldsymbol{\rho})$ will be found, and that the phase stability problem is solved correctly.

Since in typical problem specifications the mixture composition is given in terms of the mole fraction vector $\mathbf{x}_0$ (or equivalently in terms of mole numbers), in order to use Eqs. (22) or (24) for phase stability analysis it is necessary to first determine the feed composition in terms of the molar density vector $\boldsymbol{\rho}_0$. This can be done by substituting $\mathbf{x} = \mathbf{x}_0$ into the EOS, Eq. (21), and solving for the mixture density ϱ. Then $\boldsymbol{\rho}_0 = \varrho \mathbf{x}_0$. A difficulty here is that the SAFT EOS is high order in ϱ and thus there may be a large number of real roots (e.g., Koak et al.,[30] Lucia and Luo[31]). The real density root that yields the smallest molar Gibbs energy $g = (\tilde{a} + P)/\varrho$ must then be used to compute $\boldsymbol{\rho}_0$. Thus, in solving the EOS for ϱ, it is important that no real density roots be missed, since that may result in failure to find the root yielding the smallest Gibbs energy. Again, by using the interval approach described below, we can provide a guarantee that no roots of the EOS will be missed.

The fact that for a given $\mathbf{x}$ there may be multiple real values of ϱ that satisfy the EOS highlights an advantage of the volume-based approach over the usual molar approach, at least in the context of conventional point-valued numerical techniques. Note that the usual approach to phase stability analysis is based on the molar Gibbs free energy function $g(\mathbf{x}, \varrho)$, which will be multivalued if the EOS has multiple real density roots. In this case, the actual function that must be used is $\min_{\varrho} g(\mathbf{x}, \varrho)$, which may not be a smooth function. Thus, not only are there difficulties in ensuring that all the real density roots are found, but also in dealing with the optimization of a nonsmooth function. In contrast, the volume-based approach uses the function $\tilde{a}(\boldsymbol{\rho})$, which is a smooth function of the component densities and is not multivalued with respect to the component densities. This point is explained in some detail by Nagarajan et al.[25] for the case of a cubic EOS. However, since the interval approach used here can deal with multiple real roots of the EOS, and the resulting nonsmooth Gibbs energy function, without any special effort, as shown by Hua et al.[15,22,23] for the case of cubic EOS models, this advantage of the volume-based approach does not come into play here. Nevertheless, we wish to emphasize that this approach may have significant advantages when applied in connection with conventional point-valued methods for computing phase behavior from the SAFT EOS.

The phase stability problem represents one stage in the two-stage procedure outlined above for computing phase equilibrium. If solution of the phase stability problem indicates that the mixture will split into multiple phases, then the second stage of procedure, the phase split problem, must be considered.

2.3 Phase Split Problem

In solving the phase split problem, it is desired to compute the amounts and compositions of the phases present at equilibrium. For constant T and P, this means that the minimum in the total Gibbs energy $G = A + PV$ must be sought, either by direct optimization, or, equivalently, by solution of the first-order optimality conditions (equifugacity conditions).

Following again the volume-based approach of Nagarajan et al.,[25] the problem is to minimize

$$G = \left[V^{\mathbf{I}}\tilde{a}(\boldsymbol{\rho})^{\mathbf{I}} + V^{\mathbf{II}}\tilde{a}(\boldsymbol{\rho})^{\mathbf{II}} + \cdots + V^{\boldsymbol{\Psi}}\tilde{a}(\boldsymbol{\rho})^{\boldsymbol{\Psi}}\right] + P(V^{\mathbf{I}} + V^{\mathbf{II}} + \cdots + V^{\boldsymbol{\Psi}}) \tag{25}$$

subject to the material balance constraint

$$V^{\mathbf{I}}\boldsymbol{\rho}^{\mathbf{I}} + V^{\mathbf{II}}\boldsymbol{\rho}^{\mathbf{II}} + \cdots + V^{\boldsymbol{\Psi}}\boldsymbol{\rho}^{\boldsymbol{\Psi}} = \mathbf{x}_0. \tag{26}$$

Here it is assumed that there are $\boldsymbol{\Psi}$ phases, labeled $\mathbf{I}, \mathbf{II}, \ldots, \boldsymbol{\Psi}$, and that the basis for the material balance is a unit mole system. The first-order optimality (equifugacity) conditions for this constrained minimization problem require that

$$\nabla\tilde{a}(\boldsymbol{\rho}^{\mathbf{I}}) = \nabla\tilde{a}(\boldsymbol{\rho}^{\mathbf{II}}) = \cdots = \nabla\tilde{a}(\boldsymbol{\rho}^{\boldsymbol{\Psi}}) \tag{27}$$

and that

$$P = \mathcal{F}(\boldsymbol{\rho}^{\mathbf{I}}) = \mathcal{F}(\boldsymbol{\rho}^{\mathbf{II}}) = \cdots = \mathcal{F}(\boldsymbol{\rho}^{\boldsymbol{\Psi}}), \tag{28}$$

where the pressure-explicit equation of state, $P = \mathcal{F}(\boldsymbol{\rho})$, can be expressed[25] in terms of $\tilde{a}(\boldsymbol{\rho})$ by

$$P = \mathcal{F}(\boldsymbol{\rho}) = \nabla\tilde{a}(\boldsymbol{\rho}) \cdot \boldsymbol{\rho} - \tilde{a}(\boldsymbol{\rho}). \tag{29}$$

While global optimization techniques may be applied directly to the minimization problem here, the two-stage procedure being used requires only that a local minimum be sought, either by direct minimization or by solution of the equifugacity conditions. As noted by Sun and Seider,[32] good initial guesses for the local solution of the phase split problem can typically be generated from the local minima in the tangent plane distance function D found by solving Eq. (24) as part of the phase stability problem. For example, if a two-phase split problem is being solved, and three local minima in D were identified in the phase stability problem just completed, then taken pairwise (since two phases have been assumed) there are three different initial guesses that could be generated for the two-phase split problem.

Once a local solution to the phase split problem has been found, it is then checked for global optimality by solving a phase stability problem, testing any one of the phases computed, since they must all share the same tangent plane (a necessary condition for equifugacity). If the tested phase proves not to be stable, then another local solution must be sought, either by using a different initial guess with the same number of phases, or by increasing the number of phases. If a local solution to the phase split problem is found that does prove to be stable when phase stability analysis is done, then the final molar compositions can be determined by computing for each phase $\varrho = \sum_{i=1}^{N} \rho_i$ and $x_i = \rho_i/\varrho$.

The type of two-stage process outlined here for computing phase equilibrium is widely used. However, its reliability cannot be guaranteed unless one can guarantee that the phase stability problem is solved correctly. To provide this guarantee we use the interval approach discussed in the next section.

3 Problem Solving Methodology

In this section we provide details about the actual problem solving methodology used to solve the phase stability and phase split problems when the SAFT EOS model is used. Since, as emphasized above, the key to the solving the phase equilibrium problem correctly is the ability to solve the phase stability problem with complete certainty, we first summarize the interval approach used for this purpose.

3.1 Interval Approach

We apply here interval mathematics, in particular an interval Newton/generalized bisection (IN/GB) technique, to find enclosures for all solutions to the nonlinear equation solving problems that arise in the phase stability problem. We will briefly outline these topics here, and then, in section 3.2, explain how they were applied to solve the phase stability problem. Recent monographs which more thoroughly introduce interval analysis, as well as interval arithmetic and other aspects of computing with intervals, include those of Neumaier,[33] Hansen[34] and Kearfott.[35] It should be emphasized that, when machine computations with interval arithmetic operations are done, as in the procedures outlined below, the endpoints of an interval are computed with a directed outward rounding. That is, the lower endpoint is rounded down to the next machine-representable number and the upper endpoint is rounded up to the next machine-representable number. In this way, through the use of interval, as opposed to floating point arithmetic, any potential rounding error problems are eliminated.

A key concept in the interval methodology used is that of an *interval extension*. For an arbitrary function $\mathbf{f}(\mathbf{x})$, the interval extension over the interval $\mathbf{X}$, denoted $\mathbf{F}(\mathbf{X})$, is an interval enclosing all values of $\mathbf{f}(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$; that is, it encloses the *range* of $\mathbf{f}(\mathbf{x})$ over $\mathbf{X}$ (interval quantities are indicated in upper case, point quantities in lower case). It is often computed by substituting the given interval $\mathbf{X}$ into the function $\mathbf{f}(\mathbf{x})$ and then evaluating the function using interval arithmetic. This so-called "natural" interval extension is often wider than the actual range of function values, though it always includes the actual range. This overestimation of the function range is due to the "dependency" problem, which may arise when a variable occurs more than once in a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when the natural interval extension is computed. In effect, when the natural interval extension is used, the range computed for the function is the range that would occur if each instance of a particular variable were allowed to take on a different value in its interval range.

Consider the solution of a nonlinear equation system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{x} \in \mathbf{X}^{(0)}$ and the goal is to find (or, more precisely, enclose within very narrow intervals) *all* the roots of the equation system in $\mathbf{X}^{(0)}$. The solution algorithm is applied to a sequence of intervals, beginning with the initial interval vector (box) $\mathbf{X}^{(0)}$ specified by the user. This initial interval can be chosen to be sufficiently large to enclose all physically feasible behavior. For an interval $\mathbf{X}^{(k)}$ in the sequence, the first step in the solution procedure is the *function range test*. Here the interval extension $\mathbf{F}(\mathbf{X}^{(k)})$ of $\mathbf{f}(\mathbf{x})$ over the current interval $\mathbf{X}^{(k)}$ is computed and tested to see whether it contains zero. If not, then clearly there is no root of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in this interval and it can be discarded. If $\mathbf{X}^{(k)}$ passes the function range test, then the next step is the *interval Newton test*. This requires an interval extension of the Jacobian matrix of $\mathbf{f}(\mathbf{x})$, and involves setting up and solving the interval Newton equation (a system of linear interval equations) for a new interval, referred to here as the *image*. Comparison of the image to the current interval being tested provides a powerful existence and uniqueness test for roots of the equation system.[36] In particular, if the image is a proper subset of the current interval, then this is mathematical proof that the current interval contains a *unique* solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, which can be tightly enclosed with quadratic convergence by continued application of the interval Newton equation. Furthermore, any root lying in the current interval must also lie in the image. Thus, if the image and the current interval do not intersect, this is mathematical proof that the current interval contains no roots of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. If the number of roots in the current interval cannot be proven to be zero or one in this way, then in most cases the current interval is bisected, and the resulting two intervals added to the sequence of intervals to be tested.

These are the basic ideas of interval Newton/generalized bisection (IN/GB) methods. More detailed descriptions of the technique used are available from Schnepper and Stadtherr[37] and Hua et al.[15] As a framework for our implementation of the IN/GB method, we use appropriately modified FORTRAN-77 routines from the packages INTBIS[38] and INTLIB.[39] In addition, for solving the interval Newton equation, the hybrid preconditioning technique of Gau et al.[40] is employed. Overall, the IN/GB method described above provides a procedure that is mathematically *and* computationally guaranteed to enclose *all* solutions of the nonlinear equation systems that must be solved to perform phase stability analysis using the SAFT EOS.

3.2 Phase Stability Problem

To solve the phase stability problem, we use the interval approach outlined above to solve the $N \times N$ nonlinear equation system given by Eq. (24) to enclose *all* the stationary points of the tangent plane distance function D. This guarantees that the global minimum of D will be located. If the global minimum of D is negative then the phase being tested is not stable. However, it should be noted that finding *all* the stationary points is not always necessary, as discussed in more detail below. To initialize $\boldsymbol{\rho}$, the initial interval used for each component ρ_i, $i = 1, \ldots, N$, is

$$\rho_i \in \left[\xi, \frac{\sqrt{2}}{N_{\mathrm{A}} m_i d_{ii}^3}\right] \tag{30}$$

The lower bound ξ is some arbitrary small positive number, taken here to be $\xi = 10^{-15}$ (mol/mL), corresponding to the implicit assumption in phase stability analysis that every component is present in at least trace amounts. The upper bound comes from the closest packing limit for pure component i, which says that the reduced density η cannot exceed $\tau = \pi\sqrt{2}/6$. Note that no initial point guess is needed, and that the initial interval used covers all physically feasible possibilities. In some cases, especially at low reduced temperatures, it may be appropriate to consider "non-physical" density roots (e.g., Koak et al.[30]), i.e., density roots that exceed the closest packing limit. If this is desired, the upper limit in Eq. (30) should be determined based on η not exceeding one, resulting in the upper limit $6/(\pi N_{\mathrm{A}} m_i d_{ii}^3)$. Expressions for the derivatives appearing in Eq. (24) and its Jacobian are given by Eqs. (41–42) in Appendix A.

If the initial mixture composition is given in terms of mole fractions $\mathbf{x}_0$ instead of component densities $\boldsymbol{\rho}_0$, then, as explained in more detail above, it is first necessary to compute the mixture density ϱ in order to formulate Eq. (24). This is done by using the IN/GB approach to solve Eq. (21) for ϱ, given $\mathbf{x} = \mathbf{x}_0$. The initial interval used for ϱ is

$$\varrho \in \left[\frac{P}{Z_{\max} RT}, \frac{\sqrt{2}}{N_{\mathrm{A}} \sum\limits_{i=1}^{N} x_{i0} m_i d_{ii}^3} \right] \tag{31}$$

The lower bound comes from setting an arbitrary upper limit of $Z_{\max}$ on the compressibility factor $Z = P/\varrho RT$. A value of $Z_{\max} = 2$ is used here; this should be more than adequate for the reduced pressures considered in the examples used below; however, this value can be adjusted as needed in solving other types of problems. The upper bound comes again from the closest packing limit. While in general the EOS may have a large number of real density roots, in our experience, many of these can be eliminated by enforcing this upper bound on ϱ. However, there may still be multiple real density roots in the interval given by Eq. (31). As explained above, the one corresponding to the lowest Gibbs energy must then be used to determine $\boldsymbol{\rho}_0$. Once $\boldsymbol{\rho}_0$ has been determined one can then proceed to solve Eq. (24). As discussed previously, if one wishes to consider non-physical density roots, then the upper limit in Eq. (31) should be adjusted to correspond to η not exceeding one.

In order to apply the interval approach to solve Eq. (24), it is necessary to compute interval extensions of this function and its Jacobian for a given $\boldsymbol{\rho}$ interval. To do this it is first necessary to deal with the "internal" iteration defined by Eq. (18) for the "internal" variables X^{A_i}. This is an $N_s \times N_s$ nonlinear equation system, where N_s indicates the total number of association sites in the model. The goal is to determine an interval value, preferably with as small as width as possible, for the X^{A_i}, such that it contains *all* values of the X^{A_i} that satisfy Eq. (18) for the current $\boldsymbol{\rho}$ interval. There are several possible ways to do this; an ϵ-expansion approach[35] is used here. First a point-valued version of Eq. (18) is formulated using the midpoints of the current ρ_i intervals, and its (unique in [0,1]) solution found. The uniqueness of the point solution in [0,1] can be seen by rewriting Eq. (18) as

$$f_{\mathrm{A}_i} = X^{\mathrm{A}_i} + N_{\mathrm{A}} \sum_{j=1}^{N} \sum_{\mathrm{B}_j} \left(\rho_{\mathrm{j}} X^{\mathrm{A}_i} X^{\mathrm{B}_j} \Delta^{\mathrm{A}_i \mathrm{B}_j} \right) - 1, \tag{32}$$

which has the Jacobian elements

$$\frac{\partial f_{\mathrm{A}_i}}{\partial X^{\mathrm{A}_i}} = 1 + N_{\mathrm{A}} \sum_{j=1}^{N} \sum_{\mathrm{B}_j} \left(\rho_j X^{\mathrm{B}_j} \Delta^{\mathrm{A}_i \mathrm{B}_j} \right) + N_{\mathrm{A}} \rho_i X^{\mathrm{A}_i} \Delta^{\mathrm{A}_i \mathrm{A}_i} \tag{33}$$

$$\frac{\partial f_{\mathrm{A}_i}}{\partial X^{\mathrm{B}_j}} = N_{\mathrm{A}} \rho_j X^{\mathrm{A}_i} \Delta^{\mathrm{A}_i \mathrm{B}_j}. \tag{34}$$

Noting that the slopes in Eqs. (33–34) are positive, that the Jacobian matrix is diagonally dominant and thus nonsingular, and that $f_{\mathrm{A}_i} = 0$ is bracketed by $X^{\mathrm{A}_i} = 0$ and $X^{\mathrm{A}_i} = 1$ for all association sites, one can conclude that there is a unique solution in [0,1]. Once a point solution is found it is gradually expanded as an interval until the image computed by applying the interval Newton equation to Eq. (32) is contained in the expanded X^{A_i} interval. This interval for the X^{A_i} has now been verified to contain all values of the X^{A_i} that satisfy Eq. (18) for the current $\boldsymbol{\rho}$ interval. If the current $\boldsymbol{\rho}$ interval is relatively wide, it is possible that the result for one or more of the X^{A_i} will be no smaller than [0,1]. In this case, processing of the current $\boldsymbol{\rho}$ interval is stopped and it is bisected.

In order to evaluate Eq. (24) and its Jacobian, not only are intervals for the X^{A_i} needed, but also intervals for the first and second derivatives of the X^{A_i} with respect to the ρ_i. From differentiation of Eq. (24), the first derivatives $[X^{\mathrm{A}_i}]_k = \partial X^{\mathrm{A}_i}/\partial \rho_k$ are

$$[X^{\mathrm{A}_i}]_k = -N_{\mathrm{A}}(X^{\mathrm{A}_i})^2 \left\{ \sum_{j=1}^{N} \sum_{\mathrm{B}_j} \rho_j \Delta^{\mathrm{A}_i\mathrm{B}_j} [X^{\mathrm{B}_j}]_k + \sum_{j=1}^{N} \sum_{\mathrm{B}_j} \rho_j X^{\mathrm{B}_j} [\Delta^{\mathrm{A}_i\mathrm{B}_j}]_k + \sum_{\mathrm{B}_k} X^{\mathrm{B}_k} \Delta^{\mathrm{A}_i\mathrm{B}_j} \right\} \qquad (35)$$

where $[\Delta^{\mathrm{A}_i\mathrm{B}_j}]_k = \partial \Delta^{\mathrm{A}_i\mathrm{B}_j}/\partial \rho_k$. This is not an explicit expression for $[X^{\mathrm{A}_i}]_k$ since $[X^{\mathrm{A}_i}]_k$ also appears in a term in the first double summation on the right-hand side. While it appears to be common practice to solve this iteratively for the derivatives $[X^{\mathrm{A}_i}]_k$, we show here how this can be done directly without iteration. This is possible because Eq. (35) is in fact linear in the $[X^{\mathrm{A}_i}]_k$ and can be rearranged

$$Q\,\mathbf{y}_k = \mathbf{c}_k. \qquad (36)$$

Here $\mathbf{y}_k$ is the $N_s \times 1$ vector comprising the first derivatives $[X^{\mathrm{A}_i}]_k$ that are to be solved for, Q is an $N_s \times N_s$ matrix with coefficients

$$q_{ii} = 1 + (X^{\mathrm{A}_i})^2 N_{\mathrm{A}} \rho_i \Delta^{\mathrm{A}_i\mathrm{A}_i}$$

$$q_{ij} = (X^{\mathrm{A}_i})^2 N_{\mathrm{A}} \rho_j \Delta^{\mathrm{A}_i\mathrm{B}_j},$$

and $\mathbf{c}_k$ is an $N_s \times 1$ vector with elements given by

$$c_k^{\mathrm{A}_i} = -(X^{\mathrm{A}_i})^2 N_{\mathrm{A}} \left\{ \sum_{j=1}^{N} \sum_{\mathrm{B}_j} \rho_j X^{\mathrm{B}_j} [\Delta^{\mathrm{A}_i\mathrm{B}_j}]_k + \sum_{\mathrm{B}_k} X^{\mathrm{B}_k} \Delta^{\mathrm{A}_i\mathrm{B}_j} \right\}.$$

Once the X^{A_i} are determined from Eq. (18), then Eq. (36) is just an $N_s \times N_s$ system of linear equations that can be solved for the vector $\mathbf{y}_k$ of first derivatives of X^{A_i} with respect to ρ_k. This is repeated for each species $k = 1, \ldots, N$. Since in the algorithm being used here the X^{A_i} are interval valued, Eq. (36) is a linear interval equation system. There are various approaches for bounding the solution set of such a system.[33–35,41,42] We first precondition Eq. (36) using a standard inverse-midpoint preconditioner (that is, the inverse of the point-valued matrix formed from the midpoints of the elements of Q). This system is then solved using the approach first described by Hansen[41] for bounding the elements of Q^{-1}. Since all the linear systems to be solved for the first derivatives of the X^{A_i} (and for the second derivatives as seen below) have the same coefficient matrix Q, the inverse can be saved and re-used. If the current $\boldsymbol{\rho}$ interval is relatively wide, it is possible that useful bounds on the solution set of Eq. (36) cannot be obtained. In this case, the processing of the current $\boldsymbol{\rho}$ interval is stopped and it is bisected. It should be emphasized that this procedure for the *direct* computation of the first derivatives of the X^{A_i} can also be exploited when conventional point-valued methods are used to compute phase behavior from SAFT.

The second derivatives of the X^{A_i} are needed if the current $\boldsymbol{\rho}$ interval has passed the range test and an interval Newton test is required. From differentiation of Eq. (35), the second derivatives $[X^{\mathrm{A}_i}]_{kl} = \partial^2 X^{\mathrm{A}_i}/\partial\rho_k\partial\rho_l$ are

$$\begin{aligned}
[X^{\mathrm{A}_i}]_{kl} &= \frac{2[X^{\mathrm{A}_i}]_k[X^{\mathrm{A}_i}]_l}{X^{\mathrm{A}_i}} - N_{\mathrm{A}}(X^{\mathrm{A}_i})^2\left\{\sum_{j=1}^{N}\sum_{\mathrm{B}_j}\rho_j\Delta^{\mathrm{A}_i\mathrm{B}_j}[X^{\mathrm{B}_j}]_{kl}\right. \qquad (37)\\
&+ \sum_{j=1}^{N}\sum_{\mathrm{B}_j}\rho_j\left([\Delta^{\mathrm{A}_i\mathrm{B}_j}]_l[X^{\mathrm{B}_j}]_k + [\Delta^{\mathrm{A}_i\mathrm{B}_j}]_k[X^{\mathrm{B}_j}]_l + X^{\mathrm{B}_j}[\Delta^{\mathrm{A}_i\mathrm{B}_j}]_{kl}\right)\\
&+ \left.\sum_{\mathrm{B}_k}\left([X^{\mathrm{B}_k}]_l\Delta^{\mathrm{A}_i\mathrm{B}_k} + X^{\mathrm{B}_k}[\Delta^{\mathrm{A}_i\mathrm{B}_k}]_l\right) + \sum_{\mathrm{B}_l}\left([X^{\mathrm{B}_l}]_k\Delta^{\mathrm{A}_i\mathrm{B}_l} + X^{\mathrm{B}_l}[\Delta^{\mathrm{A}_i\mathrm{B}_l}]_k\right)\right\}
\end{aligned}$$

where $[\Delta^{\mathrm{A}_i\mathrm{B}_j}]_{kl} = \partial^2\Delta^{\mathrm{A}_i\mathrm{B}_j}/\partial\rho_k\partial\rho_l$. Again this is not an explicit expression for $[X^{\mathrm{A}_i}]_{kl}$ and it appears to be common practice to solve Eq. (37) iteratively for the second derivatives $[X^{\mathrm{A}_i}]_{kl}$. However, this also can be done directly by solution of a linear equation system. Note that Eq. (37) can be rearranged

$$Q\,\mathbf{y}_{kl} = \mathbf{c}_{kl}, \qquad (38)$$

where $\mathbf{y}_{kl}$ is the $N_s \times 1$ vector comprising the second derivatives $[X^{\mathrm{A}_i}]_{kl}$ that are to be solved for, Q is the same $N_s \times N_s$ matrix defined above, and and $\mathbf{c}_{kl}$ is an $N_s \times 1$ vector with elements given by

$$\begin{aligned}
\mathbf{c}_{kl} &= \frac{2[X^{\mathrm{A}_i}]_k[X^{\mathrm{A}_i}]_l}{X^{\mathrm{A}_i}} - \qquad (39)\\
&(X^{\mathrm{A}_i})^2 N_{\mathrm{A}}\left\{\sum_{j=1}^{N}\sum_{\mathrm{B}_j}\rho_j\left([\Delta^{\mathrm{A}_i\mathrm{B}_j}]_l[X^{\mathrm{B}_j}]_k + [\Delta^{\mathrm{A}_i\mathrm{B}_j}]_k[X^{\mathrm{B}_j}]_l + X^{\mathrm{B}_j}[\Delta^{\mathrm{A}_i\mathrm{B}_j}]_{kl}\right) + \right.\\
&\left.\sum_{\mathrm{B}_k}\left([X^{\mathrm{B}_k}]_l\Delta^{\mathrm{A}_i\mathrm{B}_k} + X^{\mathrm{B}_k}[\Delta^{\mathrm{A}_i\mathrm{B}_k}]_l\right) + \sum_{\mathrm{B}_l}\left([X^{\mathrm{B}_l}]_k\Delta^{\mathrm{A}_i\mathrm{B}_l} + X^{\mathrm{B}_l}[\Delta^{\mathrm{A}_i\mathrm{B}_l}]_k\right)\right\}.
\end{aligned}$$

Once the X^{A_i} are determined from Eq. (18) and the first derivatives $[X^{\mathrm{A}_i}]_k$ from Eq. (36), then Eq. (38) is just an $N_s \times N_s$ system of linear equations that can be solved for the vector $\mathbf{y}_{kl}$ of second derivatives of X^{A_i} with respect to ρ_k and ρ_l. This is repeated for each pair of species $k = 1, \ldots, N$ and $l = 1, \ldots, N$. Since in the algorithm being used here, the X^{A_i} and $[X^{\mathrm{A}_i}]_k$ are interval valued, we again require the use of an interval linear equation solver to determine interval-valued results for the $[X^{\mathrm{A}_i}]_{kl}$. This can be done using the same approach as discussed above for determining intervals for the $[X^{\mathrm{A}_i}]_k$, and, in fact, since the matrix Q in Eq. (38) is the same as in Eq. (36), the bounds on Q^{-1} generated in solving Eq. (36) can be used for solving Eq. (38) as well. As in the case of the first derivatives, we emphasize that this procedure for the *direct* computation of the second derivatives of the X^{A_i} can also be taken advantage of when conventional point-valued methods are used to compute phase behavior from SAFT. Since in point-valued methods a problem formulation in terms of $\mathbf{x}$ and ϱ may be used, and since this formulation is required here in order to solve Eq. (21) for the feed density ϱ, we give in Appendix B the linear equation systems that can be solved for the first and second derivatives of the X^{A_i} when this formulation is used.

Because of the complexity of Eq. (24) and its Jacobian elements, it is difficult in computing interval extensions to avoid overestimation due to the dependency problem discussed above. However, some steps can be taken in an attempt to ameliorate this difficulty. For example, density weighted averages are frequently occurring quantities in the model. These are quantities of the form $\bar{s} = \sum_{i=1}^{N} \rho_i s_i$, where the s_i are constants. There are two inequality constraints on the ρ_i that, when active, can be used to more tightly bound expressions of this form. These inequalities are

$$\sum_{i=1}^{N} \rho_i \geq \varrho_{\min} = \frac{P}{Z_{\max} RT},$$

indicating simply that the sum of the component densities must be no less than the minimum total density ϱ set in Eq. (31), and

$$\eta = \zeta_3 = \frac{\pi N_{\mathrm{A}}}{6} \sum_{i=1}^{N} \rho_i m_i d_{ii}^3 \leq \tau = \frac{\pi\sqrt{2}}{6},$$

which indicates that the reduced density from Eq. (8) cannot exceed the closest packing value. If nonphysical density roots are being allowed, as discussed above, then the upper bound on η should be set to one. For any interval $\boldsymbol{\rho}$ for which either of these constraints is active, the bounds on $\bar{s}$ can be improved using the procedure explained by Hua et al.[15] for the case of mole fraction weighted averages.

Another useful procedure for reducing overestimations due to the dependency problem is the use of high-order Taylor models, as explained by Makino and Berz.[43] This is particularly useful in dealing with the terms in Eqs. (40–42) in Appendix A that arise from the dispersion term given by Eq. (6). For example, consider the quantity

$$\mathcal{F}(G,\zeta) = \sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij} \left(\frac{1}{\tau}\right)^{j} iG^{(i-1)}\zeta^{j}$$

that occurs in both Eqs. (41) and (42). Here G and ζ are parameters defined in Appendix A, and are both interval valued. A Taylor model of $\mathcal{F}(G,\zeta)$ can be constructed with respect to one or both of the variables G and ζ. Doing this for ζ and using a ninth-order Taylor expansion around some point $\zeta_p \in \zeta$, taken here to be the midpoint of the ζ interval, yields

$$\begin{aligned}
\mathcal{F}(G,\zeta) &= \mathcal{F}(G,\zeta_p) + (\zeta-\zeta_p)\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^{j} iG^{(i-1)} j\zeta_p^{(j-1)} \\
&+ \frac{(\zeta-\zeta_p)^2}{2!}\sum_{i=1}^{4}\sum_{j=2}^{9} D_{ij}\left(\frac{1}{\tau}\right)^{j} iG^{(i-1)} j(j-1)\zeta_p^{(j-2)} \\
&+ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
&+ \frac{(\zeta-\zeta_p)^9}{9!}\sum_{i=1}^{4}\sum_{j=9}^{9} D_{ij}\left(\frac{1}{\tau}\right)^{j} iG^{(i-1)} j(j-1)(j-2)\cdots(j-8)\zeta_p^{(j-9)}.
\end{aligned}$$

Note that since $\mathcal{F}(G,\zeta)$ is a ninth-order polynomial in ζ and we have used a ninth-order Taylor model, the remainder bound in the Taylor series is zero. Using such high-order Taylor models for this and the other similar quantities in Eqs. (40–42) that arise from the dispersion term leads to substantial computational savings.

3.3 Phase Split Problem

For solving the phase split problem, it is sufficient to use a local, point-valued method, since the results will subsequently be checked for global optimality using phase stability analysis. Thus, modelers may use their favorite methodology, whether it is based on direct optimization of the total Gibbs energy or based on solution of the equifugacity conditions. To solve the phase split problem, we follow the equifugacity approach, and use the nonlinear equation solving code NEQLU described by Chen and Stadtherr[44] to solve the equation system given by Eqs. (26–28). This code is based on the “dogleg” approach. For the special case $N = \Psi$, the phase rule dictates that the component densities in each phase can be determined by solving only the system given by Eqs. (27–28), with Eq. (26) used subsequently to compute the phase volumes. Initial guesses for the phase split problem are determined from results of the prior phase stability analysis, as discussed above.

4 Test Problems and Results

To test our initial implementation of the interval methodology for reliable computation of phase stability and equilibrium from SAFT, several different binary mixtures have been used. The first two mixtures are self-associating systems, the next two are cross-associating, and the final one is nonassociating. For all problems, the SAFT parameters used for each component were taken from Huang and Radosz[4] and are listed in Table 2. The nomenclature for association type in Table 2 is that used by Huang and Radosz[4] and is explained further in the examples below. All computations were performed on a Sun Ultra 10/440 workstation.

4.1 Problem 1. 2B Self-Associating System

This is a mixture of n-heptane(1) and 1-propanol(2). There are no association sites on n-heptane molecules, but 1-propanol molecules self-associate. 1-Propanol is assumed to be of association type 2B. This means that there are two association sites: A_2, on the hydrogen in the hydroxyl group, and B_2, on the oxygen in the hydroxyl group, and that there is no association between sites of the same type; that is, $\Delta^{\mathrm{A}_2\mathrm{A}_2} = 0$, $\Delta^{\mathrm{B}_2\mathrm{B}_2} = 0$, and $\Delta^{\mathrm{A}_2\mathrm{B}_2} = \Delta^{\mathrm{B}_2\mathrm{A}_2} \neq 0$. From Table 2, $\epsilon^{\mathrm{A}_2\mathrm{B}_2}/k = 2619$ K, and $\kappa^{\mathrm{A}_2\mathrm{B}_2} = 0.01968$. For this special case, the internal variables X^{A_2} and X^{B_2} are equal and can be solved for explicitly from Eq. (18) without iteration. However, in order to emphasize the generality of the approach used here, we have not exploited this in solving this problem. For the binary interaction parameter, we use $k_{ij} = 0.018$, as given by Fu and Sandler.[7]

Table 3 shows results for five different feed compositions $\mathbf{x}_0$ at $T = 333$ K and $P = 0.35$ bar. For each feed, the results of solving the EOS for the feed density ϱ are shown first. While mathematically the EOS may in some cases have other real density roots, those shown are all the real density roots within the physically feasible bounds specified above. The density root corresponding to the minimum Gibbs energy is indicated with an asterisk. This is the root used in the calculation of $\boldsymbol{\rho}_0$ to set up the phase stability problem. Shown next in the Table are the results of solving the initial phase stability problem. Each stationary point found, along with the corresponding value of the tangent plane distance D is given. Then for the cases in which the feed is not stable (i.e., there is a negative value for D), the final results of the phase equilibrium calculation are given, in terms of the composition (mole fractions) in each phase, and the molar phase fraction ϕ for each phase. The computational results for phase stability and equilibrium agree well with those that were obtained by Fu and Sandler[7] using the same SAFT model, but with slightly different (unpublished) model parameters. It should be noted that, while rounded point approximations are given in Table 3, we have actually determined verified enclosures of each stationary point and computed D for this enclosure. Similarly, the ϱ results are rounded point approximations of verified enclosures. Each such enclosure is a very narrow interval that is known to contain a unique root based on the interval Newton uniqueness test.

In solving the phase stability problem, for this example, and for the subsequent examples, we have located the global minimum in D by solving Eq. (24) for *all* the stationary points in D. It should be noted, however, that the interval methodology can be implemented so that it is not necessary to find enclosures of all the stationary points. This can be done by making use of the underlying global minimization problem, and incorporating a branch-and-bound strategy into the overall IN/GB algorithm. For example, since the objective function D has a known value of zero at $\boldsymbol{\rho} = \boldsymbol{\rho}_0$, any $\boldsymbol{\rho}$ interval for which the interval extension of D has a lower bound greater than zero cannot contain the global minimum and thus can be discarded, even though it may contain a stationary point (at which D will be positive and thus not of interest). However, the addition of such objective range tests requires the additional work of computing an interval extension of D, so will not necessarily reduce computation time requirements. Computation times can often be reduced by noting that in order to demonstrate that a mixture is not stable, it is not really necessary to find the global minimum of D, but only to find *any* point $\boldsymbol{\rho}$ for which D is negative. This can often be done using a fast local method, followed by verification of the negativity of D using interval arithmetic. However, this shortcut is not suitable for the initial phase stability test, at least if one plans to use the results to initialize the subsequent phase split calculation. These ideas for incorporating branch-and-bound strategies and local methods into the algorithm for solving the phase stability problem have been discussed in more detail by Hua et al.[15] in the context of cubic EOS models.

For the feed mixtures that split into two phases, the CPU time requirements are roughly twice that for the mixtures that do not split. This is because, for the mixtures that split, the phase stability problem is solved rigorously twice, once at the beginning to determine that the mixture is not stable and to generate initial guesses for the phase split calculation, and then once at the end to verify that the two-phase solution found is in fact the global minimum in the total Gibbs energy. However, if there is some alternative means to get initial guesses for the phase split problem, then, for the initial phase stability test, a local method could be used as a shortcut, as discussed above, to try to establish that the mixture is not stable. Using this approach, the CPU times for the feeds that split could be roughly halved.

The computation time requirements for this problem and those that follow are relatively large. This is not surprising considering the complexity of the model and the fact that a mathematical and computational guarantee of reliability has been provided. Clearly, there is a trade-off between computational speed and reliability. However, since this is our first implementation of IN/GB for the phase stability problem with the SAFT EOS, we anticipate that there are opportunities for improving its computational efficiency. As discussed further below, the decision to treat the internal iteration with generality, rather than take advantage of the special properties of a given system, comes at a significant computational cost. As with all computation time results, it is also important to view such results in the context of the ongoing advances in computer processor speed, which continues to roughly double every 18 months.

4.2 Problem 2. 1A Self-Associating System

This is a mixture of acetic acid(1) and benzene(2). There are no association sites on benzene molecules, but acetic acid molecules self associate. In SAFT, the carboxylic acid group is typically treated as a single strong association site (A_1) that can self associate. Thus, acetic acid is considered as having a 1A association type, with $\epsilon^{\mathrm{A}_1\mathrm{A}_1}/k = 3941$ K and $\kappa^{\mathrm{A}_1\mathrm{A}_1} = 0.03926$. Again, for this special case, the internal variable X^{A_1}

can be solved for explicitly from Eq. (18) without iteration, but to emphasize the generality of our approach, we have not exploited this in solving this problem. For the binary interaction parameter, we use $k_{ij} = 0.031$, as given by Fu and Sandler.[7]

Three feed compositions were considered at $T = 323$ K and $P = 0.25$ bar, as indicated in Table 4. The interval methodology worked well in all cases, demonstrating stability of the feeds at $x_{1,0} = 0.125$ and $x_{1,0} = 0.75$, and a phase split for the feed at $x_{1,0} = 0.25$. Again, these computed results for phase stability and equilibrium agree well with those that were obtained by Fu and Sandler[7] using the same SAFT model, but with slightly different (unpublished) model parameters.

4.3 Problem 3. 2B + 2B Cross-Associating System

This is a mixture of 1-butanol(1) and ethanol(2). Both components are assumed to be of association type 2B, as described above. Here there is no association between sites of the same type, so $\Delta^{\mathrm{A_1A_1}} = \Delta^{\mathrm{A_1A_2}} = \Delta^{\mathrm{A_2A_1}} = \Delta^{\mathrm{A_2A_2}} = 0$ and $\Delta^{\mathrm{B_1B_1}} = \Delta^{\mathrm{B_1B_2}} = \Delta^{\mathrm{B_2B_1}} = \Delta^{\mathrm{B_2B_2}} = 0$. From Table 2, $\epsilon^{\mathrm{A_1B_1}}/k = 2605$ K, $\epsilon^{\mathrm{A_2B_2}}/k = 2759$ K, $\kappa^{\mathrm{A_1B_1}} = 0.01639$, and $\kappa^{\mathrm{A_2B_2}} = 0.02920$. For the cross associating parameters $\epsilon^{\mathrm{A_1B_2}} = \epsilon^{\mathrm{A_2B_1}}$ and $\kappa^{\mathrm{A_1B_2}} = \kappa^{\mathrm{A_2B_1}}$, we use the "mixing rules" of Fu and Sandler:[7]

$$\epsilon^{\mathrm{A_1B_2}} = \epsilon^{\mathrm{A_2B_1}} = \sqrt{\epsilon^{\mathrm{A_1B_1}}\epsilon^{\mathrm{A_2B_2}}}$$

$$\kappa^{\mathrm{A_1B_2}} = \kappa^{\mathrm{A_2B_1}} = \frac{\kappa^{\mathrm{A_1B_2}} + \kappa^{\mathrm{A_1B_2}}}{2}.$$

It should be noted that alternative mixing rules for cross-association are available (e.g., Wolbach and Sandler[45]). For the binary interaction parameter, we use $k_{ij} = 0.010$, as in Fu and Sandler,[7] who solve the problem with the same SAFT model but slightly different (unpublished) parameters.

Three feed compositions were considered at $T = 343$ K and $P = 0.35$ bar, as indicated in Table 5. The interval methodology worked well in all cases, demonstrating the stability of the feeds at $x_{1,0} = 0.265$ and $x_{1,0} = 0.9$, and a phase split for the feed at $x_{1,0} = 0.65$. The increase in CPU times compared to the first two problems apparently arises from the increased difficulty of the internal iteration problem due to the cross association terms. Once again the computed results are in good agreement with the computed results of Fu and Sandler.[7]

4.4 Problem 4. 1A + 2B Cross-Associating System

This is a mixture of acetic acid (1) and 1-butanol (2). Acetic acid is treated as having association type 1A, and 1-butanol as having type 2B, both as described above. Thus there are a total of three association sites, $\mathrm{A_1}$ on acetic acid, and $\mathrm{A_2}$ and $\mathrm{B_2}$ on 1-butanol. For the nonzero association parameters, the mixing rules given by Fu and Sandler[7] are used:

$$\epsilon^{\mathrm{A_1A_2}} = \epsilon^{\mathrm{A_1B_2}} = \sqrt{\epsilon^{\mathrm{A_1A_1}}\epsilon^{\mathrm{A_2B_2}}}$$

$$\kappa^{\mathrm{A_1A_2}} = \kappa^{\mathrm{A_1B_2}} = \frac{\kappa^{\mathrm{A_1A_1}} + \kappa^{\mathrm{A_1B_2}}}{2}.$$

For the binary interaction parameter, we use $k_{ij} = -0.046$, as in Fu and Sandler.[7]

Three feed compositions at $T = 308$ K and $P = 0.02664$ bar were considered, and the results shown in Table 6. Again the IN/GB approach worked well for each feed, demonstrating the stability of the feeds at $x_{1,0} = 0.995$ and $x_{1,0} = 0.80$, and a phase split for the feed at $x_{1,0} = 0.95$. Again, these computed results match closely the results computed from the SAFT model by Fu and Sandler.[7]

4.5 Problem 5. Nonassociating System

This is a mixture of ethene(1) and n-eicosane(2). There are no association sites on either molecule so $a^{\mathrm{assoc}} = 0$ in Eq. (1). The binary interaction parameter value used was $k_{ij} = 0.076$, from Huang and Radosz.[5] Three feed compositions at $T = 423$ K and $P = 20$ bar were considered, and three more at $T = 423$ K and $P = 250$ bar, with the results shown in Table 7. The computed results match the calculations of Huang and Radosz,[5] who used the same SAFT model.

The CPU times required for these mixtures are significantly less than in the previous cases, suggesting, not surprisingly, that the additional model complexity resulting from the association term contributes substantially to the difficulty of solving the problem. It should be emphasized that, in the implementation of the interval approach tested here, we have chosen to treat the internal iteration resulting from the association term with complete generality. Undoubtedly, if one took advantage of the special properties of each type of association system, for example the ability to solve the internal iteration explicitly in some cases and the use of other special relationships between the internal variables, the computational effort required by the first four problems could be greatly reduced.

5 Concluding Remarks

We have described here a new methodology that is the first *completely reliable* technique for computing phase stability and equilibrium from the SAFT EOS model. The method is based on interval analysis, in particular an interval Newton/generalized bisection algorithm, which provides a mathematical and computational *guarantee* of reliability. The method was demonstrated using a number of test problems, including nonassociating, self-associating, and cross-associating systems. While we have demonstrated the technique here using the Huang and Radosz[4,5] version of SAFT, with binary mixtures as examples, the methodology is general purpose and can be applied to other versions of SAFT, as well as to multicomponent systems, and multiple fluid phases of any type (i.e., not just VLE and LLE, but VLLE, etc.). For the case of more than two phases, computational times will not differ significantly from the case of two phases, at least if the rigorous phase stability analysis is not used to generate initial guesses for phase split calculations and thus needs to be done only once, to verify the final result. For multicomponent systems, our experience (e.g., Tessier et al.[24]) is that computation times will increase with the number of components, though perhaps not in entirely predictable ways.

The use of a volume-based problem formulation, in which the core thermodynamic function for phase equilibrium at constant temperature and pressure is the Helmholtz energy, proved to be very convenient. Moreover, since it lessens difficulties due to multiple real roots of the EOS, this problem formulation may have significant advantages when applied in connection with conventional point-valued methods. A new approach for dealing with the internal iteration needed when there are association effects was also developed. This provides for direct, as opposed to iterative, determination of the derivatives of the internal variables, and can be applied not only in the context of the interval methodology described here, but also when conventional point-valued methods are used.

Acknowledgments

This work has been supported in part by the donors of The Petroleum Research Fund, administered by the ACS, under Grant 35979-AC9, by the National Science Foundation Grant EEC97-00537-CRCD, and by the Environmental Protection Agency Grant R826-734-01-0. We also thank Prof. Walter G. Chapman for his helpful comments and advice.

References

(1) Chapman, W. G.; Jackson, G.; Gubbins, K. E. Phase Equilibria of Associating fluids: Chain Molecules with Multiple Bonding Sites. *Mol. Phys.* **1988**, *65*, 1057.

(2) Chapman, W. G.; Gubbins, K. E.; Jackson, G.; Radosz, M. SAFT: Equation-of-State Solution Model for Associating Fluids. *Fluid Phase Equilib.* **1989**, *52*, 31.

(3) Chapman, W. G.; Gubbins, K. E.; Jackson, G.; Radosz, M. New Reference Equation of State for Associating Liquids. *Ind. Eng. Chem. Res.* **1990**, *29*, 1709.

(4) Huang S. H.; Radosz, M. Equation of State for Small, Large, Polydisperse, and Associating Molecules. *Ind. Eng. Chem. Res.* **1990**, *29*, 2284.

(5) Huang S. H.; Radosz, M. Equation of State for Small, Large, Polydisperse, and Associating Molecules: Extension to Fluid Mixtures. *Ind. Eng. Chem. Res.* **1991**, *30*, 1994.

(6) Huang S. H.; Radosz, M. Additions and Corrections. *Ind. Eng. Chem. Res.* **1993**, *32*, 762.

(7) Fu, Y.; Sandler, S. I. A Simplified SAFT Equation of State for Associating Compounds and Mixtures. *Ind. Eng. Chem. Res.* **1995**, *34*, 1897.

(8) Blas, F. J.; Vega L. F. Prediction of Binary and Ternary Diagrams Using the Statistical Associating Fluid Theory (SAFT) Equation of State. *Ind. Eng. Chem. Res.* **1998**, *37*, 660.

(9) Michelsen, M. L. The Isothermal Flash Problem. Part I: Stability. *Fluid Phase Equilib.* **1982**, *9*, 1.

(10) Michelsen, M. L. The Isothermal Flash Problem. Part II: Phase-Split Calculation. *Fluid Phase Equilib.* **1982**, *9*, 21.

(11) Lucia, A.; Padmanabhan, L.; Venkataraman, S. Multiphase Equilbrium Flash Calculations. *Comput. Chem. Eng.* **2000**, *24*, 2557.

(12) Michelsen, M. L. State Function Based Flash Specifications. *Fluid Phase Equilib.* **1999**, *160*, 617.

(13) McKinnon, K. I. M.; Millar, C. G.; Mongeau M. Global Optimization for the Chemical and Phase Equilibrium Problem Using Interval Analysis. In *State of the Art in Global Optimization: Computational Methods and Applications*; Floudas, C. A.; Pardalos, P.M., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.

(14) Green, K. A.; Zhou, S.; Luks, K. D. The Fractal Response of Robust Solution Techniques to the Stationary Point Problem. *Fluid Phase Equilib.*, **1993**, *84*, 49.

(15) Hua, J. Z.; Brennecke, J. F.; Stadtherr, M. A. Enhanced Interval Analysis for Phase Stability: Cubic Equation of State Models. *Ind. Eng. Chem. Res.* **1998**, *37*, 1519.

(16) McDonald, C. M.; Floudas, C. A. Global Optimization for the Phase Stability Problem. *AIChE J.* **1995a**, *41*, 1798.

(17) McDonald, C. M.; Floudas, C. A. Global Optimization for the Phase and Chemical Equilibrium Problem: Application to the NRTL Equation. *Comput. Chem. Eng.* **1995b**, *19*, 1111.

(18) McDonald, C. M.; Floudas, C. A. Global Optimization and Analysis for the Gibbs Free Energy Function using the UNIFAC, Wilson, and ASOG Equations. *Ind. Eng. Chem. Res.* **1995c**, *34*, 1674.

(19) McDonald, C. M.; Floudas, C. A. GLOPEQ: A New Computational Tool for the Phase and Chemical Equilibrium Problem. *Comput. Chem. Eng.* **1997**, *21*, 1.

(20) Harding, S. T.; Floudas, C. A. Phase Stability with Cubic Equations of State: Global Optimization Approach. *AIChE J.* **2000** *46*, 1422.

(21) Stadtherr, M. A.; Schnepper, C. A.; Brennecke, J. F. Robust Phase Stability Analysis Using Interval Methods. *AIChE Symp. Ser.* **1995**, *91*(304), 356.

(22) Hua, J. Z.; Brennecke, J. F.; Stadtherr, M. A. Reliable Prediction of Phase Stability Using an Interval-Newton Method. *Fluid Phase Equilib.* **1996**, *116*, 52.

(23) Hua, J. Z.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of Phase Stability Using Interval Analysis: Cubic Equation of State Models. *Comput. Chem. Eng.* **1998**, *22*, 1207.

(24) Tessier, S. R.; Brennecke, J. F.; Stadtherr, M. A. Reliable Phase Stability Analysis for Excess Gibbs Energy Models. *Chem. Eng. Sci.* **2000**, *55*, 1785.

(25) Nagarajan, N. R.; Cullick, A. S.; Griewank, A. New Strategy for Phase Equilibrium and Critical Point Calculations by Thermodynamic Energy Analysis. Part I. Stability Analysis and Flash. *Fluid Phase Equilib.* **1991**, *62*, 191.

(26) Boublík, T. Hard-Sphere Equation of State. *J. Chem. Phys.* **1970**, *53*, 471.

(27) Chen, S. S.; Kreglewski, A. Applications of the Augmented van der Waals Theory of Fluids. I. Pure Fluids. *Ber. Bunsen-Ges. Phys. Chem.* **1977**, *81*, 1048.

(28) Topliss, R. J. Techniques to Facilitate the Use of Equations of State for Complex Fluid Phase Equilibrium. Ph. D. Thesis, University of California, Berkeley, 1985.

(29) Baker, L. E.; Pierce, A. C.; Luks, K. D. Gibbs Energy Analysis of Phase Equilibria. *Soc. Petrol. Engrs. J.* **1982**, *22*, 731.

(30) Koak, N.; de Loos, T. W.; Heidemann, R. A. Effect of the Power Series Dispersion Term on the Pressure-Volum Behavior of Statistical Associating Fluid Theory. *Ind. Eng. Chem. Res.* **1999**, *38*, 1718.

(31) Lucia, A.; Luo, Q. Binary Refrigerant-Oil Phase Equilibrium Using the Simplified SAFT Equation. *Adv. Environ. Res.* **2001**, in press.

(32) Sun, A. C.; Seider, W. D. Homotopy-continuation Method for Stability Analysis in the Global Minimization of the Gibbs Free Energy. *Fluid Phase Equilib.* **1995**, *103*, 213.

(33) Neumaier, A. *Interval Methods for Systems of Equations*; Cambridge University Press: Cambridge, England, 1990.

(34) Hansen, E. R.*Global Optimization Using Interval Analysis*; Marcel Dekkar: New York, 1992.

(35) Kearfott, R. B. *Rigorous Global Search: Continuous Problems*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.

(36) Moore, R. E. *Interval Analysis*, Prentice-Hall: Englewood Cliffs, New Jersey, 1966.

(37) Schnepper, C. A.; Stadtherr, M. A. Robust Process Simulation using Interval Methods. *Comput. Chem. Eng.* **1996**, *20*, 187.

(38) Kearfott, R. B.; Novoa, M. Algorithm 681: INTBIS, a Portable Interval Newton/Bisection Package. *ACM Trans. Math. Software* **1990**, *16*, 152.

(39) Kearfott, R. B.; Dawande, M.; Du, K.-S.; Hu, C.-Y. Algorithm 737: INTLIB, A Portable FORTRAN 77 Interval Standard Function Library. *ACM Trans. Math. Software* **1994**, *20*, 447.

(40) Gau, C.-Y.; Stadtherr, M. A. New Interval Methodologies for Reliable Process Modeling. *Comput. Chem. Eng.* **2001**, submitted for publication.

(41) Hansen, E. Interval Arithmetic in Matrix Computations, Part I. *J. SIAM Numer. Anal.* **1965**, *2*, 308.

(42) Hansen, E.; Smith, R. Interval Arithmetic in Matrix Computations, Part II. *J. SIAM Numer. Anal.* **1967**, *4*, 1.

(43) Makino, K.; Berz, M. Efficient Control of the Dependency Problem Based on Taylor Model Methods. *Reliable Computing* **1999**, *5*, 3.

(44) Chen, H. S.; Stadtherr, M. A. A Modification of Powell's Dogleg Method for Solving Systems of Nonlinear Equations. *Comput. Chem. Eng.* **1981**, *5*, 143.

(45) Wolbach, J. P.; Sandler, S. I. Using Molecular Orbital Calculations To Describe the Phase Behavior of Cross-associating Mixtures. *Ind. Eng. Chem. Res.* **1998**, *37*, 2917.

Appendix A. SAFT Parameterization

Parameters

Eight parameters, A through H, are used here, which are similar to those used by Huang and Radosz,[5,6] who gave a parameterization of $a^{\text{res}}(\mathbf{x}, \varrho)$, the residual molar Helmholtz energy in terms of mole fraction and total density. However, the parameterization here is for $\tilde{a}^{\text{res}}(\boldsymbol{\rho})$, the residual Helmholtz energy density in terms of molar component densities. Note that here d_{ii} is abbreviated as d_i, ζ_3 as ζ, and $g_{ij}^{\text{hs}}(d_{ij})$ as g_{ij}.

$$A = \sum_{i=1}^{N} \rho_i m_i d_i^0 \; ; \qquad \zeta_0 = \left(\frac{\pi N_{\text{A}}}{6}\right) A$$

$$B = \sum_{i=1}^{N} \rho_i m_i d_i^1 \; ; \qquad \zeta_1 = \left(\frac{\pi N_{\text{A}}}{6}\right) B$$

$$C = \sum_{i=1}^{N} \rho_i m_i d_i^2 \; ; \qquad \zeta_2 = \left(\frac{\pi N_{\text{A}}}{6}\right) C$$

$$D = \sum_{i=1}^{N} \rho_i m_i d_i^3 \; ; \qquad \zeta = \left(\frac{\pi N_{\text{A}}}{6}\right) D$$

$$E = A = \sum_{i=1}^{N} \rho_i m_i \; ;$$

$$F = a^{\text{chain}}/RT = \sum_{i=1}^{N} \rho_i (1 - m_i) \ln(g_{ii}) \; ;$$

$$G = \frac{u}{kT} = \frac{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \rho_i \rho_j m_i m_j [u_{ij}/kT] v_{ij}^0}{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \rho_i \rho_j m_i m_j v_{ij}^0} \; ;$$

$$H = \frac{a^{\text{assoc}}}{RT} = \sum_{i=1}^{N} \rho_i \left[\sum_{\text{A}_i} \left[\ln X^{\text{A}_i} - \frac{X^{\text{A}_i}}{2} \right] + \frac{M_i}{2} \right] .$$

After parameterization, the residual Helmholtz energy density is

$$\frac{\tilde{a}^{\text{res}}}{RT} = \frac{3\frac{BC}{D}\zeta - \frac{C^3}{D^2}}{1 - \zeta} + \frac{\frac{C^3}{D^2}}{(1 - \zeta)^2} + \left[\frac{C^3}{D^2} - A\right] \ln(1 - \zeta) + F + E \sum_{i=1}^{4} \sum_{j=1}^{9} D_{ij} G^i \left[\frac{\zeta}{\tau}\right]^j + H. \qquad (40)$$

First Derivatives with Respect to Component Densities

The notation $[\,\cdot\,]_k$ is used here to indicate $\partial[\,\cdot\,]/\partial\rho_k$.

$$\begin{aligned}
[g_{ij}]_k &= \left(\frac{\pi N_{\mathrm{A}}}{6}\right)\frac{m_k(d_k)^2}{(1-\zeta)^2}\left\{ d_k + \frac{3d_id_j}{d_i+d_j}\left[1+\frac{2d_k\zeta_2}{1-\zeta}\right]\right.\\
&+ \left. 2\left[\frac{d_id_j}{d_i+d_j}\right]^2\left[\frac{2\zeta_2}{1-\zeta}+\frac{3d_k(\zeta_2)^2}{(1-\zeta)^2}\right]\right\} ;
\end{aligned}$$

$$[\ln(g_{ij})]_k = \frac{1}{g_{ij}}\,[g_{ij}]_k \ ;$$

$$[A]_k = m_k \ ; \qquad [B]_k = m_kd_k \ ; \qquad [C]_k = m_kd_k^2 \ ;$$

$$[D]_k = m_kd_k^3 \ ; \qquad [\zeta]_k = \left(\frac{\pi N_{\mathrm{A}}}{6}\right)D_k \ ; \qquad [E]_k = m_k \ ;$$

$$[F]_k = (1-m_k)\ln(g_{kk}) + \sum_{l=1}^{N}\rho_l(1-m_l)[g_{ll}]_k/g_{ll} \ ;$$

$$[G]_k = \frac{2\sum\limits_{j=1}^{N}\rho_jm_km_j\left(\frac{u_{kj}}{kT}\right)v^0_{kj} - 2G\sum\limits_{j=1}^{N}\rho_jm_km_jv^0_{kj}}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\rho_i\rho_jm_im_jv^0_{ij}} \ ;$$

$$[H]_k = \sum_{\mathrm{A}_k}\left[\ln X^{\mathrm{A}_k} - \frac{X^{\mathrm{A}_k}}{2}\right] + \frac{M_k}{2} + \sum_{i=1}^{N}\rho_i\left[\sum_{\mathrm{A}_i}\left[\frac{1}{X^{\mathrm{A}_i}}-\frac{1}{2}\right][X^{\mathrm{A}_i}]_k\right].$$

This leads to

$$\begin{aligned}
\frac{1}{RT}\frac{\partial\tilde{a}^{res}}{\partial\rho_k} &= \frac{3[B]_kC\zeta + 3B[C]_k\zeta - 3C^2[C]_k/D + 2C^3[D]_k/D^2}{D(1-\zeta)}\\
&+ \frac{1}{D(1-\zeta)^2}\left[\left(3BC\zeta - \frac{C^3}{D}\right)[\zeta]_k + \frac{3C^2[C]_k}{D} - \frac{2C^3[D]_k}{D^2}\right] + \frac{2C^3[\zeta]_k}{D^2(1-\zeta)^3}\\
&+ \left[\frac{3C^2[C]_k}{D^2} - \frac{2C^3[D]_k}{D^3} - [A]_k\right]\ln(1-\zeta) - \left[\frac{C^3}{D^2}-A\right]\frac{[\zeta]_k}{1-\zeta}\\
&+ [F]_k + [E]_k\sum_{i=1}^{4}\sum_{j=1}^{9}D_{ij}G^i\left[\frac{\zeta}{\tau}\right]^j\\
&+ E\left[[G]_k\sum_{i=1}^{4}\sum_{j=1}^{9}D_{ij}\left[\frac{1}{\tau}\right]^j iG^{(i-1)}\zeta^j + [\zeta]_k\sum_{i=1}^{4}\sum_{j=1}^{9}D_{ij}\left[\frac{1}{\tau}\right]^j jG^i\zeta^{(j-1)}\right]\\
&+ [H]_k. \qquad (41)
\end{aligned}$$

Second Derivatives with Respect to Component Densities

The notation $[\,\cdot\,]_{kl}$ is used here to indicate $\partial^2[\,\cdot\,]/\partial\rho_k\partial\rho_l$.

$$\begin{aligned}[g_{ij}]_{kl} &= \left(\frac{\pi N_{\mathrm{A}}}{6}\right)^2 \frac{2m_k m_l (d_k d_l)^2}{(1-\zeta)^3}\left\{ d_k d_l + 3\frac{d_i d_j}{d_i + d_j}\left[d_l + d_k + 3\frac{d_k d_l \zeta_2}{1-\zeta}\right]\right. \\ &+ \left. 2\left(\frac{d_i d_j}{d_i + d_j}\right)^2 \left[1 + 3\frac{d_l \zeta_2}{1-\zeta} + 3\frac{d_k\zeta_2}{1-\zeta} + 6\frac{d_k d_l \zeta_2^2}{(1-\zeta)^2}\right]\right\} ;\end{aligned}$$

$$[\ln g_{ij}]_{kl} = \frac{1}{g_{ij}}[g_{ij}]_{kl} - \frac{1}{(g_{ij})^2}[g_{ij}]_l [g_{ij}]_k \ ;$$

$$[F]_{kl} = (1-m_k)[\ln g_{kk}]_l + (1-m_l)[\ln g_{ll}]_k + \sum_{i=1}^{N} \rho_i (1-m_i)[\ln g_{ii}]_{kl} \ ;$$

$$\begin{aligned}[G]_{kl} &= \frac{2}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} \rho_i\rho_j m_i m_j v_{ij}^0}\left[m_k m_l \left(\frac{u_{kl}}{kT}\right) v_{kl}^0 \right. \\ &- \left. [G]_l \sum_{i=1}^{N} \rho_i m_i m_k v_{ki}^0 - [G]_k \sum_{i=1}^{N} \rho_i m_i m_l v_{li}^0 - G m_k m_l v_{kl}^0 \right] ;\end{aligned}$$

$$\begin{aligned}[H]_{kl} &= \sum_{\mathrm{A}_k}\left(\frac{1}{X^{\mathrm{A}_k}} - \frac{1}{2}\right)[X^{\mathrm{A}_k}]_l + \sum_{\mathrm{A}_l}\left(\frac{1}{X^{\mathrm{A}_l}} - \frac{1}{2}\right)[X^{\mathrm{A}_l}]_k \\ &+ \sum_{i=1}^{N}\rho_i\left\{\sum_{\mathrm{A}_i}\left[\left(\frac{1}{X^{\mathrm{A}_i}} - \frac{1}{2}\right)[X^{\mathrm{A}_i}]_{kl} - \frac{[X^{\mathrm{A}_i}]_k [X^{\mathrm{A}_i}]_l}{(X^{\mathrm{A}_i})^2}\right]\right\} .\end{aligned}$$

This leads to

$$
\begin{aligned}
\frac{1}{RT}\frac{\partial^2 \tilde{a}^{res}}{\partial\rho_k\partial\rho_l} &= \frac{[\zeta]_l}{D(1-\zeta)^2}\left(3[B]_k C\zeta + 3B[C]_k\zeta - 3\frac{C^2[C]_k}{D} + 2\frac{C^3[D]_k}{D^2}\right) \\
&+ \frac{1}{D(1-\zeta)}\left(3[B]_k[C]_l\zeta + 3[B]_l[C]_k\zeta - 6\frac{C[C]_l[C]_k}{D}\right. \\
&+ \left. 6\frac{C^2[C]_k[D]_l}{D^2} + 6\frac{C^2[C]_l[D]_k}{D^2} - 6\frac{C^3[D]_k[D]_l}{D^3}\right) \\
&+ \frac{2[\zeta]_l[\zeta]_k}{(1-\zeta)^3}\left(3\frac{BC}{D}\zeta - \frac{C^3}{D^2}\right) + \frac{2[\zeta]_l}{D^2(1-\zeta)^3}\left(3C^2[C]_k - 2\frac{C^3[D]_k}{D}\right) \\
&+ \frac{[\zeta]_k}{D(1-\zeta)^2}\left(3[B]_l C\zeta + 3B[C]_l\zeta - 3\frac{C^2[C]_l}{D} + 2\frac{C^3[D]_l}{D^2}\right) \\
&+ \frac{1}{D^2(1-\zeta)^2}\left[6C[C]_k[C]_l - 6\frac{C^2[C]_k[D]_l}{D} - 6\frac{C^2[C]_l[D]_k}{D} + 6\frac{C^3[D]_l[D]_k}{D^2}\right] \\
&+ \frac{6[\zeta]_l[\zeta]_k C^3}{D^2(1-\zeta)^4} + \frac{6C^2[C]_l[\zeta]_k}{D^2(1-\zeta)^3} - \frac{4C^3[\zeta]_k[D]_l}{D^3(1-\zeta)^3} \\
&- \frac{[\zeta]_l}{1-\zeta}\left[\frac{3C^2[C]_k}{D^2} - \frac{2C^3[D]_k}{D^3} - [A]_k\right] \\
&+ \frac{\ln(1-\zeta)}{D^2}\left[6C[C]_k[C]_l - \frac{6C^2[C]_k[D]_l}{D} - \frac{6C^2[C]_l[D]_k}{D} + \frac{6C^3[D]_k[D]_l}{D^2}\right] \\
&- \frac{[\zeta]_k[\zeta]_l}{(1-\zeta)^2}\left[\frac{C^3}{D^2} - A\right] - \frac{[\zeta]_k}{1-\zeta}\left(\frac{3C^2[C]_l}{D^2} - \frac{2C^3[D]_l}{D^3} - [A]_l\right) + [F]_{kl} \\
&+ [E]_k\left[[G]_l\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j iG^{(i-1)}\zeta^j + [\zeta]_l\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j jG^i\zeta^{(j-1)}\right] \\
&+ [E]_l\left[[G]_k\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j iG^{(i-1)}\zeta^j + [\zeta]_k\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j jG^i\zeta^{(j-1)}\right] \\
&+ E\left[[G]_{kl}\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j iG^{i-1}\zeta^j\right. \\
&+ ([G]_k[\zeta]_l + [G]_l[\zeta]_k)\sum_{i=1}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j ijG^{i-1}\zeta^{j-1} \\
&+ [G]_k[G]_l\sum_{i=2}^{4}\sum_{j=1}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j i(i-1)G^{i-2}\zeta^j \\
&+ \left.[\zeta]_k[\zeta]_l\sum_{i=1}^{4}\sum_{j=2}^{9} D_{ij}\left(\frac{1}{\tau}\right)^j j(j-1)G^i\zeta^{j-2}\right] + [H]_{kl}.
\end{aligned}
\tag{42}
$$

Appendix B. Linear Systems for Derivatives of Internal Variables

The derivatives of the internal variables X^{A_i} can be determined directly without iteration by solving a series of linear equation systems. In the text the linear equation systems to be solved are given for the case in which the volume-based approach is used and the problem formulated in terms of the molar component densities ρ_i. In this Appendix, we give the linear systems for the case in which the independent variables are the component mole fractions x_i and total density ϱ. For the indices k and l below, values in the range $1, \dots, N$ refer to the mole fraction variables and a value of $N+1$ refers to the total density.

First Derivatives

The first derivatives $[X^{\mathrm{A}_i}]_k$ can be determined by solving the linear systems

$$Q\,\mathbf{y}_k = \mathbf{c}_k, \qquad k = 1, \dots, N+1. \tag{43}$$

Here $\mathbf{y}_k$ is the $N_s \times 1$ vector comprising the first derivatives $[X^{\mathrm{A}_i}]_k$ that are to be solved for, Q is an $N_s \times N_s$ matrix with coefficients

$$q_{ii} = 1 + N_{\mathrm{A}}\rho(X^{\mathrm{A}_i})^2 x_i \Delta^{\mathrm{A}_i A_i}$$

$$q_{ij} = N_{\mathrm{A}}\rho(X^{\mathrm{A}_i})^2 x_j \Delta^{\mathrm{A}_i \mathrm{B}_j},$$

and $\mathbf{c}_k$ is an $N_s \times 1$ vector with elements given by

$$c_k^{\mathrm{A}_i} = -N_{\mathrm{A}}\rho(X^{\mathrm{A}_i})^2 \left(\sum_{\mathrm{B}_k} X^{\mathrm{B}_k} \Delta^{\mathrm{A}_i \mathrm{B}_k} + \sum_{j=1}^{N} \sum_{\mathrm{B}_j} x_j X^{\mathrm{B}_j} [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k \right), \qquad k = 1, \dots, N$$

$$c_k^{\mathrm{A}_i} = -N_{\mathrm{A}}(X^{\mathrm{A}_i})^2 \left\{ \sum_{j=1}^{N} \sum_{\mathrm{B}_j} x_j X^{\mathrm{B}_j} \left(\Delta^{\mathrm{A}_i \mathrm{B}_j} + \rho [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k \right) \right\}, \qquad k = N+1.$$

Second Derivatives

The second derivatives $[X^{\mathrm{A}_i}]_{kl}$ can be determined by solving the linear systems

$$Q\,\mathbf{y}_{kl} = \mathbf{c}_{kl}, \qquad k = 1, \dots, N+1, \qquad l = 1, \dots, N+1. \tag{44}$$

Here $\mathbf{y}_{kl}$ is the $N_s \times 1$ vector comprising the second derivatives $[X^{\mathrm{A}_i}]_{kl}$ that are to be solved for and Q is the same $N_s \times N_s$ matrix defined above in this Appendix. On the right-hand side, $\mathbf{c}_{kl}$ is an $N_s \times 1$ vector with elements given by

$$\mathbf{c}_{kl} = \mathbf{r}_{kl} + \mathbf{s}_{kl} + \mathbf{t}_{kl},$$

where for $k = 1, \ldots, N$ and $l = 1, \ldots, N$,

$$\mathbf{r}_{kl} = \frac{2[X^{\mathrm{A}_i}]_k [X^{\mathrm{A}_i}]_l}{X^{\mathrm{A}_i}},$$

$$\mathbf{s}_{kl} = -N_{\mathrm{A}} \rho (X^{\mathrm{A}_i})^2 \left\{ \sum_{j=1}^{N} \sum_{\mathrm{B}_j} x_j \left(X^{\mathrm{B}_j} [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_{kl} + [X^{\mathrm{B}_j}]_k [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_l + [X^{\mathrm{B}_j}]_l [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k \right) \right\},$$

$$\begin{aligned} \mathbf{t}_{kl} &= -N_{\mathrm{A}} \rho (X^{\mathrm{A}_i})^2 \left\{ \sum_{\mathrm{B}_k} \left([\Delta^{\mathrm{A}_i \mathrm{B}_k}]_l X^{\mathrm{B}_k} + \Delta^{\mathrm{A}_i \mathrm{B}_k} [X^{\mathrm{B}_k}]_l \right) \right. \\ &+ \left. \sum_{\mathrm{B}_l} \left([\Delta^{\mathrm{A}_i \mathrm{B}_l}]_k X^{\mathrm{B}_l} + \Delta^{\mathrm{A}_i \mathrm{B}_l} [X^{\mathrm{B}_l}]_k \right) \right\}, \end{aligned}$$

for $k = 1, \ldots, N$ and $l = N + 1$, $\mathbf{r}_{kl}$ and $\mathbf{s}_{kl}$ are the same as above, and

$$\begin{aligned} \mathbf{t}_{kl} &= -N_{\mathrm{A}} (X^{\mathrm{A}_i})^2 \left\{ \rho \sum_{\mathrm{B}_k} \left([\Delta^{\mathrm{A}_i \mathrm{B}_k}]_l X^{\mathrm{B}_k} + \Delta^{\mathrm{A}_i \mathrm{B}_k} [X^{\mathrm{B}_k}]_l \right) + \sum_{\mathrm{B}_k} \Delta^{\mathrm{A}_i \mathrm{B}_k} X^{\mathrm{B}_k} \right. \\ &+ \left. \sum_{j=1}^{N} \sum_{\mathrm{B}_j} x_j \left(\Delta^{\mathrm{A}_i \mathrm{B}_j} [X^{\mathrm{B}_j}]_k + [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k X^{\mathrm{B}_j} \right) \right\}, \end{aligned}$$

and for $k = N + 1$,

$$\mathbf{r}_{kk} = \frac{2[X^{\mathrm{A}_i}]_k^2}{X^{\mathrm{A}_i}},$$

$$\begin{aligned} \mathbf{s}_{kk} &= -N_{\mathrm{A}} (X^{\mathrm{A}_i})^2 \left\{ \sum_{j=1}^{N} \sum_{\mathrm{B}_j} x_j \left(2\Delta^{\mathrm{A}_i \mathrm{B}_j} [X^{\mathrm{B}_j}]_k + 2[\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k X^{\mathrm{B}_j} \right. \right. \\ &+ \left. \left. 2\rho [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_k [X^{\mathrm{B}_j}]_k + \rho X^{\mathrm{B}_j} [\Delta^{\mathrm{A}_i \mathrm{B}_j}]_{kk} \right) \right\}, \end{aligned}$$

$$\mathbf{t}_{kk} = 0.$$

Table 1: Chen and Kreglewski[27] constants D_{ij} used in Eq. (6).

D_{ij}	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$j = 1$	-8.8043	2.9396	-2.8225	0.34
$j = 2$	4.1646270	-6.0865383	4.7600148	-3.1875014
$j = 3$	-48.203555	40.137956	11.257177	12.231796
$j = 4$	140.43620	-76.230797	-66.382743	-12.110681
$j = 5$	-195.23339	-133.70055	69.248785	0.0
$j = 6$	113.51500	860.25349	0.0	0.0
$j = 7$	0.0	-1535.3224	0.0	0.0
$j = 8$	0.0	1221.4261	0.0	0.0
$j = 9$	0.0	-409.10539	0.0	0.0

Table 2: SAFT parameters[4] for compounds in Problems 1–5.

Compound	v^{00} (mL/mol)	m	u^0/k (K)	ϵ/k (K)	$10^2\kappa$	association type
Acetic Acid	14.5	2.132	290.73	3941	3.926	1A
Benzene	11.421	3.749	250.19	–	–	–
1-Butanol	12.0	3.971	225.96	2605	1.639	2B
n-Eicosane	12.0	13.940	211.25	–	–	–
Ethanol	12.0	2.457	213.48	2759	2.920	2B
Ethene	18.157	1.464	212.06	–	–	–
n-Heptane	12.282	5.391	204.61	–	–	–
1-Propanol	12.0	3.240	225.68	2619	1.968	2B

Table 3: Results for Problem 1: n-heptane(1) and 1-propanol(2) at $T = 333$ K and $P = 0.35$ bar, with $k_{ij} = 0.018$.

Feed Composition $(x_{1,0}, x_{2,0})$	Roots ϱ (mol/L) from Eq. (21)	Roots (ρ_1, ρ_2) (mol/L) from Eq. (24)	D/RT value from Eq. (22)	Equil. Phase I (x_1, x_2, ϕ)	Equil. Phase II (x_1, x_2, ϕ)	Total CPU time (seconds)†
$(0.975, 0.025)$	(6.84*; 0.90; $0.013)$	$(6.667, 0.171)$ $(2.768, 0.069)$ $(0.010, 0.002)$	0.0 0.005 0.58×10^{-6}	$(0.975, 0.025, 1.0)$	–	143.3
$(0.875, 0.125)$	(0.0128*; 7.169; $0.968)$	$(0.011, 0.002)$ $(2.708, 0.049)$ $(6.736, 0.112)$	0.0 0.004 -0.597×10^{-3}	$(0.959, 0.041, 0.507)$	$(0.789, 0.211, 0.493)$	289.1
$(0.60, 0.40)$	(0.0128*; 1.114; $8.344)$	$(0.0077, 0.0051)$ $(2.817, 0.298)$ $(5.310, 2.626)$	0.0 0.005 0.001	$(0.60, 0.40, 1.0)$	–	194.3
$(0.45, 0.55)$	(0.0128*; 1.173; $9.175)$	$(0.006, 0.007)$ $(2.706, 0.726)$ $(1.511, 10.12)$	0.0 0.006 -0.176×10^{-3}	$(0.143, 0.857, 0.037)$	$(0.462, 0.538, 0.963)$	386.5
$(0.10, 0.90)$	(11.92*; 0.0128; $1.250)$	$(1.192, 10.72)$ $(2.667, 0.960)$ $(0.005, 0.007)$	0.0 0.006 0.896×10^{-6}	$(0.10, 0.90, 1.0)$	–	189.2

* feed density root corresponding to lowest Gibbs energy for feed mixture
† CPU time on Sun Ultra 10/440 workstation

Table 4: Results for Problem 2: acetic acid(1) and benzene(2) at $T = 323$ K and $P = 0.25$ bar, with $k_{ij} = 0.031$.

Feed Composition $(x_{1,0}, x_{2,0})$	Roots ϱ (mol/L) from Eq. (21)	Roots (ρ_1, ρ_2) (mol/L) from Eq. (24)	D/RT value from Eq. (22)	Equil. Phase I (x_1, x_2, ϕ)	Equil. Phase II (x_1, x_2, ϕ)	Total CPU time (seconds)†
$(0.125, 0.875)$	$(0.0095^*; 1.452; 11.86)$	$(0.001, 0.008)$	0.0	$(0.125, 0.875, 1.0)$	–	343.0
		$(1.207, 4.366)$	0.01			
		$(4.134, 8.415)$	0.003			
		$(32.71, 0.232 \times 10^{-5})$	0.355			
$(0.25, 0.75)$	$(0.0096^*; 1.470; 12.38)$	$(0.0024, 0.0072)$	0.0	$(0.241, 0.759\ 0.98)$	$(0.707, 0.293, 0.02)$	719.2
		$(3.015, 2.774)$	0.009			
		$(10.86, 4.108)$	-3.06×10^{-4}			
		$(32.69, 0.970 \times 10^{-6})$	0.335			
$(0.75, 0.25)$	$(15.12^*; 1.471; 1.116 \times 10^{-2})$	$(11.34, 3.78)$	0.0	$(0.75, 0.25, 1.0)$	–	357.9
		$(3.252, 2.648)$	0.0095			
		$(0.0024, 0.0066)$	0.57×10^{-6}			
		$(32.69, 0.890 \times 10^{-6})$	0.335			

* feed density root corresponding to lowest Gibbs energy for feed mixture

† CPU time on Sun Ultra 10/440 workstation

Table 5: Results for Problem 3: 1-Butanol(1) and Ethanol(2) at $T = 343$ K and $T = 0.35$ bar with $k_{ij} = 0.010$.

Feed Composition $(x_{1,0}, x_{2,0})$	Roots ϱ (mol/L) from Eq. (21)	Roots (ρ_1, ρ_2) (mol/L) from Eq. (24)	D/RT value from Eq. (22)	Equil. Phase I (x_1, x_2, ϕ)	Equil. Phase II (x_1, x_2, ϕ)	Total CPU time (seconds)†
$(0.90, 0.10)$	$(10.86$*; 0.012; $1.081)$	$(9.774, 1.086)$ $(3.953, 0.529)$ $(0.0044, 0.0028)$	0.0 0.0087 0.5×10^{-5}	$(0.90, 0.10, 1.0)$	–	858.7
$(0.65, 0.35)$	$(12.00$*; 1.227; $0.0125)$	$(7.802, 4.201)$ $(3.028, 1.990)$ $(0.0033, 0.0097)$	0.0 0.008 -0.48×10^{-6}	$(0.672, 0.328, 0.945)$	$(0.270, 0.730, 0.055)$	1816.1
$(0.265, 0.735)$	$(0.0125$*; 1.536; $14.34)$	$(0.0033, 0.0092)$ $(3.106, 1.886)$ $(7.949, 3.960)$	0.0 0.0085 0.111×10^{-3}	$(0.265, 0.735, 1.0)$	–	907.3

* feed density root corresponding to lowest Gibbs energy for feed mixture

† CPU time on Sun Ultra 10/440 workstation

Table 6: Results for Problem 4: Acetic acid(1) and 1-butanol(2) at $T = 308$ K and $P = 0.02664$ bar, with $k_{ij} = -0.046$.

Feed Composition $(x_{1,0}, x_{2,0})$	Roots ϱ (mol/L) from Eq. (21)	Roots (ρ_1, ρ_2) (mol/L) from Eq. (24)	D/RT value from Eq. (22)	Equil. Phase I (x_1, x_2, ϕ)	Equil. Phase II (x_1, x_2, ϕ)	Total CPU time (seconds)†
$(0.995, 0.005)$	$(0.0012^{*}; 31.38; 17.34; 1.279)$	$(0.001, 0.58 \times 10^{-5})$ $(31.8, 0.5 \times 10^{-3})$ $(16.02, 0.805)$ $(7.310, 0.216)$	0.0 0.279 0.003 0.014	$(0.995, 0.005, 1.0)$	–	818.6
$(0.95, 0.05)$	$(0.0012^{*}; 1.217; 16.95; 30.83)$	$(0.001, 0.58 \times 10^{-4})$ $(5.520, 1.076)$ $(12.73, 3.062)$ $(31.79, 0.005)$	0.0 0.0131 -0.0005 0.280	$(0.824, 0.176, 0.068)$	$(0.959, 0.041, 0.932)$	1766.4
$(0.80, 0.20)$	$(15.72^{*}; 29.21; 1.060; 0.0011)$	$(12.58, 3.144)$ $(31.79, 0.0057)$ $(5.494, 1.127)$ $(0.001, 0.58 \times 10^{-4})$	0.0 0.281 0.0133 0.415×10^{-7}	$(0.80, 0.20, 1.0)$	–	887.0

* feed density root corresponding to lowest Gibbs energy for feed mixture

† CPU time on Sun Ultra 10/440 workstation

Table 7: Results for Problem 5: ethene(1) and n-Eicosane(2) at $T = 423$ K, $P = 20$ bar and $P = 250$ bar, with $k_{ij} = 0.076$.

Feed Composition $(x_{1,0}, x_{2,0})$; P	Roots ϱ (mol/L) from Eq. (21)	Roots (ρ_1, ρ_2) (mol/L) from Eq. (24)	D/RT value from Eq. (22)	Equil. Phase I (x_1, x_2, ϕ)	Equil. Phase II (x_1, x_2, ϕ)	Total CPU time (seconds)†
$(0.9999, 0.0001)$; $P = 20$ bar	0.589	$(0.608, 2.433)$ $(1.321, 0.943)$ $(0.589, 0.589 \times 10^{-4})$	4.77×10^{-4} 3.28×10^{-3} 0.0	$(0.9999, 0.0001, 1.0)$	–	38.9
$(0.50, 0.50)$; $P = 20$ bar	4.184	$(3.717, 0.857)$ $(2.092, 2.092)$ $(1.988, 0.180 \times 10^{-3})$	1.09×10^{-3} 0.0 -1.21×10^{-3}	$(0.99988, 0.00012\ 0.38)$	$(0.192, 0.808, 0.62)$	88.5
$(0.18, 0.82)$; $P = 20$ bar	3.00	$(0.541, 2.464)$ $(1.214, 0.918)$ $(0.546, 0.694 \times 10^{-4})$	0.0 3.17×10^{-3} 0.40×10^{-4}	$(0.18, 0.82, 1.0)$	–	38.2
$(0.85, 0.15)$; $P = 250$ bar	7.968	$(6.773, 1.195)$ $(8.765, 0.152)$ $(8.773, 0.330)$	0.0 0.989×10^{-4} 0.103×10^{-3}	$(0.85, 0.15, 1.0)$	–	64.8
$(0.95, 0.05)$; $P = 250$ bar	9.166	$(8.708, 0.458)$ $(8.840, 0.128)$ $(7.234, 1.063)$	0.0 -0.196×10^{-4} -0.358×10^{-4}	$(0.876, 0.124, 0.330)$	$(0.987, 0.013, 0.670)$	131.3
$(0.99, 0.01)$; $P = 250$ bar	8.879	$(8.790, 0.089)$	0.0	$(0.99, 0.01, 1.0)$	–	60.6

† CPU time on Sun Ultra 10/440 workstation