

# Reliable Phase Stability Analysis for Excess Gibbs Energy Models

Stephen R. Tessier\*, Joan F. Brennecke, and Mark A. Stadtherr†  
Department of Chemical Engineering  
University of Notre Dame  
Notre Dame, IN 46556 USA

Keywords: Separations, Phase Stability, Phase Equilibrium, Thermodynamics,  
Interval Analysis, Mathematical Modeling

(August 1998; revised, June 1999)

\*Currently at AlliedSignal Inc., Troy, MI, USA

† Author to whom all correspondence should be addressed. Phone: (219)631-9318; Fax: (219)631-8366; E-mail: markst@nd.edu

## **Abstract**

Because models used to represent the Gibbs energy of mixing are typically highly nonlinear, the reliable prediction of phase stability from such models is a challenging computational problem. The phase stability problem can be formulated either as a minimization problem or as an equivalent nonlinear equation solving problem. However, conventional solution methods are initialization dependent, and may fail by converging to trivial or non-physical solutions or to a point that is a local but not global minimum. Since the correct prediction of phase stability is critical in the design and analysis of separation processes, there has been considerable recent interest in developing more reliable techniques for stability analysis. Recently we have demonstrated a technique that can solve the phase stability problem with complete reliability. The technique, which is based on interval analysis, is initialization independent, and if properly implemented provides a mathematical guarantee that the correct solution to the phase stability problem has been found. In this paper, we demonstrate the use of this technique in connection with excess Gibbs energy models. The NRTL and UNIQUAC models are used in examples, and larger problems than previously considered are solved. We also consider two means of enhancing the efficiency of the method, both based on sharpening the range of interval function evaluations. Results indicate that by using the enhanced method, computation times can be substantially reduced, especially for the larger problems.

# 1 Introduction

The basic problem in computing phase equilibrium is the solution of a nonlinear programming (NLP) problem representing the minimization of the total Gibbs energy subject to material balance constraints. This may also be represented as an equivalent set of nonlinear equations, including the equifugacity conditions. In an attempt to assure that a global minimum in the Gibbs energy is found, some type of two-stage approach is often used (e.g., Michelsen, 1982a,b; Sun and Seider, 1995; McKinnon *et al.*, 1996; McDonald and Floudas, 1997; Hua *et al.*, 1997).

In the first stage, a local (and potentially global) minimum of the NLP problem is located. This represents a candidate equilibrium solution. Generally, this is initially the trivial case of a single phase whose composition is the same as a specified overall feed composition. If it is later (in the second stage) determined that there is no single phase equilibrium solution, then in this stage a phase split calculation is performed, generally based on some presumed number of phases. This can be done by using a local method to solve the NLP problem or to solve the equivalent nonlinear equation system. Whether based on the trivial single phase case, or obtained from a phase split calculation, the phase compositions corresponding to the local minimum in the NLP problem can be used to define a tangent plane. This is a (hyper)plane tangent to the (reduced) Gibbs energy surface  $g(\mathbf{x})$  at each phase composition. It is well known (e.g. Baker *et al.*, 1982) that the candidate solution corresponds to a global minimum in the NLP problem, and thus the true equilibrium solution, only if the tangent plane  $g_{tan}(\mathbf{x})$  never intersects (i.e., lies above) the Gibbs energy surface  $g(\mathbf{x})$ .

In the second stage of this two-stage approach, this tangent plane criterion is used as a global optimality test on the local solution identified in the first stage. This is the phase stability problem. One way to determine whether  $g_{tan}(\mathbf{x})$  ever lies above  $g(\mathbf{x})$  is to consider whether the tangent plane distance function  $D(\mathbf{x}) = g(\mathbf{x}) - g_{tan}(\mathbf{x})$  is negative for any composition  $\mathbf{x}$ . A common approach for determining if  $D(\mathbf{x})$  is ever negative is to minimize it subject to the mole fractions summing to one. This optimization problem can either be solved directly or by solving an equivalent set of nonlinear equations for the stationary points

of  $D(\mathbf{x})$ . If there is no stationary point found for which  $D(\mathbf{x}) < 0$ , that is, if the global minimum of  $D(\mathbf{x})$  is 0 (at the tangent points), then the local solution being tested is globally optimal and represents the true equilibrium solution. If, on the other hand, any of these yield a negative tangent plane distance, indicating that the tangent plane intersects the Gibbs energy surface, the phase is unstable and can split (in this context, unstable refers to both the metastable and classically unstable cases). In this case, the first stage (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 second stage global optimality (phase stability) test. Assuming that the tangent plane distance is minimized globally in the phase stability analysis, 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.*, 1996).

Clearly, determining a global minimum in the phase stability problem is the key in this two-stage procedure for computing phase equilibrium. Conventional minimization or equation solving techniques 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 (e.g., Green *et al.*, 1993). Thus there is no guarantee that the phase equilibrium problem has been correctly solved. Because of the difficulties that may arise in solving such problems by standard methods (e.g., Michelsen, 1982a,b), there has been significant interest in the development of more reliable methods, as reviewed in more detail by Hua *et al.* (1998). Particularly noteworthy is the work of McDonald and Floudas (1995a,b,c; 1997), who have shown that for certain activity coefficient models, 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. McDonald and Floudas (1997) also demonstrate that while it is possible to apply rigorous global optimization techniques directly to the NLP problem for the phase equilibrium problem, it is computationally more efficient to use a two-stage approach such as outlined above, since the dimensionality of the global optimization problem (phase stability) that must be solved is less than that of the full equilibrium problem.

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 (Hua *et al.*, 1998), since it deals automatically with rounding error. This method, based on an interval Newton/generalized bisection algorithm, was originally suggested by Stadtherr *et al.* (1995), who applied it to small problems (binary and ternary) involving excess Gibbs energy models, as later done also by McKinnon *et al.* (1996). More recently Hua *et al.* (1996a,b; 1998) extended this method to problems modeled with cubic equations of state, in particular the Van der Waals (VDW), Peng-Robinson (PR) and Soave-Redlich-Kwong (SRK) models with standard mixing rules. In this paper we return to problems involving excess Gibbs energy models. Though the technique applied here is general-purpose, the applications presented here focus on the NRTL and UNIQUAC models for excess Gibbs energy. We consider larger problems using these two models than previously attempted using this method, and in doing so consider two means of enhancing the efficiency of the technique, both based on sharpening the range of interval function evaluations.

## 2 Background

### 2.1 Phase Stability Analysis

As discussed above, the determination of phase stability is often done using tangent plane analysis (Baker *et al.*, 1982; Michelsen, 1982a). A phase at specified temperature  $T$ , pressure  $P$ , and feed mole fraction  $\mathbf{z}$  is unstable if the molar Gibbs energy of mixing surface  $m(\mathbf{x}) = \Delta g_{mix} = \Delta \hat{G}_{mix}/RT$  ever falls below a plane tangent to the surface at  $\mathbf{z}$ . That is, if the tangent plane distance

$$D(\mathbf{x}) = m(\mathbf{x}) - m_0 - \sum_{i=1}^n \left( \frac{\partial m}{\partial x_i} \right)_0 (x_i - z_i) \quad (1)$$

is negative for any composition  $\mathbf{x}$ , the phase is unstable. The subscript zero indicates evaluation at  $\mathbf{x} = \mathbf{z}$ ,  $n$  is the number of components, and  $0 < x_i < 1$ ,  $i = 1, \dots, n$ . A common approach for determining if  $D$  is ever negative is to minimize  $D$  subject to the mole fractions summing to one. It is readily shown that the

stationary points in this optimization problem can be found by solving the system of nonlinear equations:

$$\left[ \left( \frac{\partial m}{\partial x_i} \right) - \left( \frac{\partial m}{\partial x_n} \right) \right] - \left[ \left( \frac{\partial m}{\partial x_i} \right) - \left( \frac{\partial m}{\partial x_n} \right) \right]_0 = 0, \quad i = 1, \dots, n-1 \quad (2)$$

$$1 - \sum_{i=1}^n x_i = 0 \quad (3)$$

The  $n \times n$  system given by equations (2)–(3) above has a trivial root at  $\mathbf{x} = \mathbf{z}$  and frequently has multiple nontrivial roots as well, even for a very simple Margules-type model of excess Gibbs energy (Green *et al.*, 1993). We use here an interval Newton/generalized bisection method for solving the system of equations (2)–(3). The method requires no initial guess, and will find with certainty enclosures of all the stationary points of the tangent plane distance  $D$ . The method can also be easily modified so that only the stationary point corresponding to the global minimum in  $D$  is found.

The technique is general-purpose and can be applied using any model for the reduced Gibbs energy of mixing  $m(\mathbf{x})$ . In this paper we concentrate on the use of excess Gibbs energy models, thus the reduced Gibbs energy of mixing is given by

$$m(\mathbf{x}) = \sum_{i=1}^n x_i \ln x_i + g^E(\mathbf{x}) \quad (4)$$

where  $g^E = \hat{G}^E/RT$  is the reduced molar excess Gibbs energy. In the example presented here we use the NRTL and UNIQUAC models for  $g^E$ .

## 2.2 NRTL Model

The Non-Random Two Liquid model is a local composition model. For an  $n$  component system, the NRTL equation for the reduced molar excess Gibbs energy is given by:

$$g^E(\mathbf{x}) = \frac{\hat{G}^E(\mathbf{x})}{RT} = \sum_{k=1}^n x_k \frac{(\overline{\tau G})_k}{\overline{G}_k} \quad (5)$$

where  $x_k$  is the mole fraction for species  $k$ , and  $\overline{G}_k$  and  $(\overline{\tau G})_k$  are mole fraction weighted averages involving the binary interaction parameters  $\tau_{ik}$  and  $G_{ik}$ :

$$\overline{G}_k = \sum_{i=1}^n G_{ik} x_i \quad (6)$$

$$(\overline{\tau G})_k = \sum_{i=1}^n \tau_{ik} G_{ik} x_i. \quad (7)$$

The presence of mole fraction weighted averages is emphasized here since this will be exploited below in improving the efficiency of the interval approach. The parameter  $G_{ik}$  is a function of  $\tau_{ik}$  and the more commonly available parameter  $\alpha_{ik} = \alpha_{ki}$ , and is given by  $G_{ik} = \exp(-\alpha_{ik}\tau_{ik})$ .

For this model, equation (2) becomes

$$\ln\left(\frac{x_i}{x_n}\right) - \ln\left(\frac{z_i}{z_n}\right) + s_i(\mathbf{x}) - s_i(\mathbf{z}) = 0 \quad i = 1, \dots, n-1 \quad (8)$$

where

$$s_i(\mathbf{x}) = \frac{(\overline{\tau G})_i}{\overline{G}_i} - \frac{(\overline{\tau G})_n}{\overline{G}_n} + \sum_{k=1}^n x_k \left[ \frac{\tau_{ik} G_{ik} - \tau_{nk} G_{nk}}{\overline{G}_k} + \frac{(\overline{\tau G})_k (-G_{ik} + G_{nk})}{\overline{G}_k^2} \right]. \quad (9)$$

Equations (8) and (3) represent an  $n \times n$  equation system whose solutions are the stationary points of  $D$  when the NRTL model is used.

### 2.3 UNIQUAC model

The **UNI**versal **QU**asi-Chemical model is another local composition model. For an  $n$  component system, the UNIQUAC equation for the reduced molar excess Gibbs energy is given by:

$$g^E(\mathbf{x}) = \frac{\hat{G}^E(\mathbf{x})}{RT} = \sum_{k=1}^n x_k \left[ \ln\left(\frac{r_k}{\bar{r}}\right) + \frac{\zeta}{2} q_k \ln\left(\frac{q_k \bar{r}}{r_k \bar{q}}\right) - q'_k \ln\left(\frac{\overline{q' \tau_k}}{q'_k}\right) \right] \quad (10)$$

where  $x_k$  is the mole fraction for species  $k$ , and

$$\bar{r} = \sum_{i=1}^n r_i x_i \quad (11)$$

$$\bar{q} = \sum_{i=1}^n q_i x_i \quad (12)$$

$$\bar{q}' = \sum_{i=1}^n q'_i x_i \quad (13)$$

$$\overline{q' \tau_k} = \sum_{i=1}^n q'_i \tau_{ik} x_i \quad (14)$$

are mole fraction weighted averages involving the pure component parameters  $r_i$ ,  $q_i$  and  $q'$  and the binary interaction parameter  $\tau_{ik}$ . The coordination number  $\zeta$  is 10.

For this model, equation (2) becomes

$$\ln\left(\frac{x_i}{x_n}\right) - \ln\left(\frac{z_i}{z_n}\right) + s_i(\mathbf{x}) - s_i(\mathbf{z}) = 0 \quad i = 1, \dots, n-1 \quad (15)$$

where

$$\begin{aligned} s_i(\mathbf{x}) = & \frac{(r_n - r_i)(-5\bar{q} + \sum_{i=1}^n x_i)}{\bar{r}} - 5(q_i - q_n) \ln \frac{\bar{q}}{\bar{r}} \\ & + q'_i \ln \frac{\bar{q}'}{q' \tau_i} - q'_n \ln \frac{\bar{q}'}{q' \tau_n} - \sum_{k=1}^n x_k \left[ \frac{q'_k (q'_i \tau_{ik} - q'_n \tau_{nk})}{q' \tau_k} \right]. \end{aligned} \quad (16)$$

Equations (15) and (3) represent an  $n \times n$  equation system whose solutions are the stationary points of  $D$  when the UNIQUAC model is used.

### 3 Methodology

#### 3.1 Interval Computations

We apply here an interval Newton/generalized bisection (IN/GB) technique. This technique provides the power to find, with confidence, *enclosures* of all solutions of a system of nonlinear equations (e.g., Neumaier, 1990; Kearfott, 1996), and to find with total reliability the global minimum of a nonlinear objective function (e.g., Hansen, 1992; Kearfott, 1996), provided only that upper and lower bounds are available for all variables. A detailed step-by-step description of the basic IN/GB algorithm used here is given by Schnepper and Stadtherr (1996), and additional details are provided by Hua *et al.* (1998). Our implementation of the IN/GB method for the phase stability problem is based on appropriately modified routines from the packages INTBIS (Kearfott and Novoa, 1990) and INTLIB (Kearfott *et al.*, 1994)

#### 3.2 Enhancements

The efficiency of the IN/GB method may depend significantly on how tightly one can compute *interval*

extensions  $F(\mathbf{X})$  of real expressions  $f(\mathbf{x})$ . Denoting  $F^R(\mathbf{X}) = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$  as the exact range of  $f(\mathbf{x})$  over the interval  $\mathbf{X}$ , then an interval extension of  $f(\mathbf{x})$  is an enclosure for  $F^R$ , that is,  $F(\mathbf{X}) \supseteq F^R(\mathbf{X})$ .

The most common approach to enclosing  $F^R$  is to use the *natural interval extension* of  $f(\mathbf{x})$ , which is obtained from the expression  $f(\mathbf{x})$  by simply replacing each occurrence of the variable  $\mathbf{x}$  by the interval  $\mathbf{X}$  and evaluating the real arithmetic operations using the corresponding interval arithmetic operations (Moore, 1966). If each variable in an expression appears only once, then the natural interval extension will always yield the true (though, in computational practice, outwardly rounded) range  $F^R$ . Unfortunately for most of the functions of interest in phase stability analysis it is not possible to perform rearrangements that eliminate all but one occurrence of each variable. In this case, use of the natural interval extension can lead to substantial overestimation of  $F^R$ , and means of computing tighter interval extensions should be considered. Two techniques considered here are the use of constraint information, and the use of monotonicity, as outlined very briefly below. More details concerning these techniques, and more discussion and examples related to the computation of interval extension, are available from Tessier (1997) and from Hua *et. al.* (1998), who use the techniques in the context of phase stability analysis for equation of state models.

### 3.2.1 Mole Fraction Weighted Averages

As emphasized above, mole fraction weighted averages, of the form  $f(\mathbf{x}) = \bar{a} = \sum_{i=1}^n x_i a_i$ , where the  $a_i$  are scalar constants and  $\sum_{i=1}^n x_i = 1$ , appear frequently in the equation system to be solved in phase stability analysis when excess Gibbs energy models are used. The natural interval extension of  $\bar{a}$  will yield the true range of the expression in the space in which all the mole fraction variables  $x_i$  are independent. However, the range can be tightened by using the constraint that the mole fractions must sum to one. Thus, Tessier (1997) developed a *constrained space interval extension*  $F^{CS}$  of  $f(\mathbf{x}) = \bar{a}$ .

One approach to using the constraint is to use a constraint propagation approach. However, this may not lead to the sharpest possible bounds on  $\bar{a}$  in the constrained space. Tessier (1997) showed how the true upper and lower bounds on  $\bar{a}$  in the constrained space can be found by thinking of the problem in terms of

linear programming, which led to an  $O(n)$  algorithm for constructing the desired upper and lower bounds for  $F^{CS}$  without actually using an LP problem solver. Details of this algorithm, as well as examples of its use have been given by Tessier (1997) and Hua *et. al.* (1998).

For computing the range of mole fraction weighted averages of scalar quantities, such as  $\overline{G}_k$  and  $(\overline{\tau G})_k$  in the NRTL model and  $\overline{r}$ ,  $\overline{q}$ ,  $\overline{q}^l$ , and  $\overline{q}^l \tau_k$  in the UNIQUAC model, this algorithm can be applied directly. This approach can also be extended to determining the range of a mole fraction weighted average of interval quantities. This is potentially useful, for example, in bounding the summations in equations (9) and (16), which are mole fraction weighted averages of the quantities in brackets, which we denote as  $a_i(\mathbf{x})$ . To evaluate the range of such a summation over a desired interval, one can first determine the range  $A_i = [a_i^L, a_i^U]$  of  $a_i(\mathbf{x})$  over the interval. Then the problem becomes one of determining the range of  $\overline{A} = \sum_{i=1}^n x_i A_i = \sum_{i=1}^n x_i [a_i^L, a_i^U] = [\sum_{i=1}^n x_i a_i^L, \sum_{i=1}^n x_i a_i^U]$ . Note that these upper and lower bounds are themselves mole fraction weighted averages of the known scalar values  $a_i^L$  and  $a_i^U$ . Thus, to get the range of  $\overline{A}$  we get its lower bound by using the algorithm referred to above to determine the lower bound of  $\sum_{i=1}^n x_i a_i^L$  and its upper bound by using this procedure to determine the upper bound of  $\sum_{i=1}^n x_i a_i^U$ . When this interval form of the constrained space extension is used, it is denoted  $F^{CSI}$ .

Using the algorithm to determine  $F^{CS}$  or  $F^{CSI}$  represents a computational overhead that, it is hoped, will be offset by the improvements due to tighter function bounds. Since the algorithm is applied twice in determining  $F^{CSI}$ , the overhead will be roughly twice that incurred in determining  $F^{CS}$ .

### 3.2.2 Monotonicity

If a function is known to be monotonic on a given interval, then clearly its range can be tightly bounded simply by evaluating the function at the endpoints of the interval. Thus, the use of a *monotonic interval extension*  $F^M$  is a common approach used in trying to more tightly bound function ranges (e.g., Hansen, 1992). If for every variable the function is either monotonically nonincreasing or monotonically nondecreasing, then  $F^M = F^R$ ; otherwise,  $F^M$  is no less sharp than the natural extension. Some implementation

issues not discussed by Hua *et. al.* (1998) are mentioned briefly here.

In order to ensure proper outward rounding in computing  $F^M$ , its upper and lower bounds should be computed using interval arithmetic. Thus, an evaluation of  $F^M$  requires two interval function evaluations, one for each bound, as opposed to only one if the natural interval extension is used. This is a computational overhead that must be traded-off against hopefully tighter function bounds when  $F^M$  is computed. Since, for purposes of using the constrained space extensions discussed above we evaluate the functions to be solved in terms of various mole fraction weighted averages, to use the monotonic interval extension in this context requires that partial derivatives with respect to the mole fraction weighted averages be bounded. This is an additional computational overhead which must be weighed against improvements due to sharper function ranges. It should also be noted that since the natural interval extension is used bound these partial derivatives, these bounds may be overestimates and thus monotonicity may not be detected even when present.

An important feature in the implementation of  $F^M$  is that monotonicity can be “inherited.” That is, if a function is known to be monotonic with respect to one or more variables over a particular interval  $\mathbf{X}$ , then any subintervals of  $\mathbf{X}$ , formed by bisection or by the interval-Newton iteration, will have the same monotonicity properties with respect to these variables. By using this property of inheritance, the unnecessary evaluation of partial derivative bounds can be avoided.

## 4 Results and Discussion

The methodology outlined above is now applied to solve several phase stability problems. These will be used to evaluate the enhancements in computational efficiency expected to be provided by using the constrained space and monotonic interval extensions. The examples will also demonstrate the performance of the method on larger NRTL and UNIQUAC problems than previously solved using the method.

The results for each problem presented below include the roots (stationary points) enclosed, as well as the value of the tangent plane distance  $D$  at each root. It should be noted that, while rounded point approximations are reported here, we have actually determined verified enclosures of each root and computed  $D$  for

this enclosure. Each such enclosure is known to contain a *unique* root, based on the interval-Newton uniqueness test (see Hua *et al.*, 1998). Thus, for example, in Table 2, for the first ( $z_1 = 0.148$ ) feed, the  $x_1$  value for the first stationary point was actually the interval  $x_1 = [0.11433639929296194, 0.11433639934254627]$ . Similarly narrow enclosures were found in all cases.

In order to emphasize the power of the method, in these studies we have found *all* the stationary points. However, it should be emphasized that, for making a determination of phase stability or instability, finding *all* the stationary points is not necessary, since at most the stationary point corresponding to the global minimum in  $D$  is needed. The method applied here can easily be modified, as discussed in more detail below, so that only this point is found. Nevertheless, it may in fact be desirable to determine all the stationary points (at least all the local minima), since the local minima in  $D$  are known to provide good initializations for the local methods used to solve the phase split stage of the phase equilibrium problem (e.g., Michelsen, 1982a,b; Sun and Seider, 1995).

For each problem, results on computational efficiency are also presented. These include data for the number of intervals on which the root inclusion test must be performed and for the CPU time on a Sun Ultra 2/1300 workstation. Results for four different ways to compute interval function extensions are given. In the tables of results, these are denoted  $F$  for the natural interval extension without any of the enhancements suggested above,  $F^{CS}$  for the scalar form of the constrained space interval extension,  $F^{CSI}$  for both interval and scalar forms of the constrained space extension, and  $F^{CSM}$  for use of monotonicity along with both interval and scalar forms of the constrained space extension. In all cases, the results found for the stationary points is the same.

#### 4.1 Problem 1

This problem involves the three-component system  $n$ -propanol (1),  $n$ -butanol (2) and water (3). The NRTL model is used with parameters (McDonald and Floudas, 1995a) listed in Table 1. Results are shown in Tables 2 and 3. The first feed listed was considered by McDonald and Floudas (1995a), and is interesting

because it is very close to the plait point and thus is particularly challenging. The results obtained for the global minimum in  $D$  match (after accounting for their misnumbering of components) those of McDonald and Floudas (1995a). Three other feeds were also considered. The improved bounds on function ranges due to the constrained space and monotonic interval extensions is apparent in the reduction in the number of intervals that must be tested, which is about 36% from  $F$  to  $F^{CS}$ , 55% from  $F$  to  $F^{CSI}$ , and 60% from  $F$  to  $F^{CSM}$ . This is not fully reflected in the CPU times, however, because of the overhead involved in implementing the enhanced interval extensions. For example, in the  $F^{CSM}$  case, the CPU time actually increases relative to  $F^{CSI}$  even though the number of root inclusion tests required is somewhat less. We would expect this overhead to perhaps be somewhat less of a factor as larger problems are considered.

## 4.2 Problem 2

This problem involves the four-component system  $n$ -propanol (1),  $n$ -butanol (2), benzene (3) and water (4). The NRTL model is used with parameters (Gmehling *et al.*, 1977-1990) listed in Table 4. Results are shown in Tables 5 and 6. All but the second feed listed are unstable. By far the largest improvement due to enhancements comes in going from  $F$  to  $F^{CS}$ , about a 68% reduction in root inclusion tests and a 60% reduction in CPU time. Using  $F^{CSI}$  and  $F^{CSM}$  results in relatively small improvements in the number of root inclusion tests, which are insufficient to overcome the overhead incurred, as seen in the CPU time results.

## 4.3 Problem 3

This problem involves the five-component system  $n$ -propanol (1),  $n$ -butanol (2), benzene (3), ethanol (4), water (5). The NRTL model is used with parameters listed in Table 7. Results are given in Tables 8 and 9. All these feeds are unstable, and some have five stationary points. Once again the biggest improvement in performance comes in changing from use of  $F$  to use of  $F^{CS}$ . In this case, the improvements due to using  $F^{CSI}$  and  $F^{CSM}$  are sufficient to somewhat overcome the overhead, resulting in an average CPU

time savings of over 75% on these problems.

#### 4.4 Problem 4

This problem involves the three-component system ethylene glycol (1), lauryl alcohol (2), and nitromethane (3). The UNIQUAC model is used with the parameters (McDonald and Floudas, 1995a) listed in Table 10. Results are shown in Tables 11 and 12. The first two feeds listed were also considered by McDonald and Floudas (1995a), and the results obtained here for the global minimum in  $D$  match their results in both cases. Two other feeds were also considered. For this problem, only the use of the  $F^{CS}$  extension proved worthwhile in terms of computation time, again due to the additional overhead required for  $F^{CSI}$  and  $F^{CSM}$ .

#### 4.5 Problem 5

This problem involves the four-component problem acetic acid (1), benzene (2), furfural (3) and cyclohexane (4). The UNIQUAC model is used with the parameters (Sørensen and Arlt, 1979–1987) listed in Table 13. Results are shown in Tables 14 and 15. The first feed is a stable phase, while the remaining four are unstable. This problem is unusual, in our experience, in that for none of the feeds was there any improvement in number of root inclusion tests due to use of monotonicity. Apparently, for this particular Gibbs energy surface, the use of the natural interval extension to bound the partial derivatives that determine monotonicity results in overestimations sufficient to prevent monotonicity from being detected when present, or at least to not be detected often enough to make any difference. Overall, the computational savings due to the use of the  $F^{CS}$  extension was impressive on this problem, averaging about 83%.

#### 4.6 Problem 6

This problem involves the five-component problem acetic acid (1), benzene (2), furfural (3), cyclohexane (4) and water (5). The UNIQUAC model is used with the parameters (Sørensen and Arlt, 1979–1987) listed

in Table 16. Results are shown in Tables 17 and 18. All of the feeds considered prove to be unstable. This is the most difficult of the problems solved in terms of computation time. Here again the use of  $F^{CSI}$  relative to  $F^{CS}$  resulted in only modest reduction in number of root inclusion tests, insufficient to overcome the computation time overhead. However, this is the only problem in which then using monotonicity in  $F^{CSM}$  resulted in a large reduction in root inclusion tests, well beyond what was needed to offset overhead, with computational savings ranging from about 75% to 85% relative to the natural interval extension. This suggests that there may be large regions of monotonicity in the functions whose zeros are sought in finding the stationary points.

## 4.7 Discussion

The results clearly show that substantial computational savings can be obtained by using the scalar form of the constrained space interval extension  $F^{CS}$  instead of the natural interval extension  $F$ . The procedure (Tessier, 1997; Hua *et al.*, 1998) for computing  $F^{CS}$  efficiently determines the exact (without roundout) bounds on the mole fraction weighted averages occurring in the equation system to be solved. This in turn leads to tighter bounds on expressions containing these parameters. The tightening of bounds on function ranges allows intervals to be eliminated more readily than if bounds are not sharp. For instance, if the true range  $F^R$  of a function over an interval does not contain zero, then that interval can be eliminated from further consideration, but an overestimate of  $F^R$ , such as might be obtained from the natural interval extension, might well contain zero, leading to further work on this interval and perhaps its bisection, creating even more work. While the interval form of the constrained space extension  $F^{CSI}$ , and the monotonic extension  $F^{CSM}$  derived from it, can even further tighten function bounds in most cases, this comes at the cost of some additional computational overhead, which cannot always be overcome by the reduction in the number of intervals that must be tested. However, even in the worst case (Problem 5), the  $F^{CSM}$  extension is not significantly more costly than  $F^{CS}$  (and is still a significant improvement over the natural interval extension  $F$ ), and there is potential for substantial savings as seen in Problem 6. Thus, since its benefit is

potentially significant, and its disadvantages are relatively small, the routine use of the  $F^{CSM}$  extension may be worthwhile, at least on problems of size equal to or greater than those considered here.

The computation times required by the method used here are quite good for a general-purpose approach offering a verified solution. All the problems used here were also solved with the commercial package Numerica (ILOG), which also offers a verified solution. This code (van Hentenryck, *et al.*, 1997) combines ideas from interval analysis, such as used here, with techniques from constraint satisfaction programming (CSP). In all cases, Numerica found the same results as reported in Tables 2,5,8,11,14 and 17, but at a computational cost roughly two to three orders of magnitude greater than for the method used here. For example, on the first feed listed for Problem 6, Numerica required 34448 seconds (over nine and a half hours) of CPU time, while (on the same machine) the method used here required only about 144 seconds (less than two and a half minutes).

While we have focused here on improving the interval function evaluations that underlie the interval approach for phase stability analysis, we can also use several other simple ways to make improvements in efficiency. For example, in the problems above we found enclosures of *all* the stationary points. However, for making a determination of phase stability or instability, finding *all* the stationary points is not always necessary. For example if an interval is encountered over which the interval evaluation of  $D$  has a negative upper bound, this guarantees that there is a point at which  $D < 0$ , and so one can immediately conclude that the mixture is unstable without determining all the stationary points. It is also possible to make use of the underlying global minimization problem. Since the objective function  $D$  has a known value of zero at the mixture feed composition, any interval over which the interval value of  $D$  has a lower bound greater than zero cannot contain the global minimum and can be discarded, even though it may contain a stationary point (at which  $D$  will be positive and thus not of interest). Thus, we can essentially combine the interval-Newton technique with an interval branch and bound procedure in which lower bounds are generated using interval techniques as opposed to convex underestimators. Also, it should be noted that the method described here can easily be combined with existing local methods for determining phase stability. First, the (fast) local

method is used. If it indicates instability then this is the correct answer as it means a point at which  $D < 0$  has been found. If the local method indicates stability, however, this may not be the correct answer since the local method may have missed the global minimum in  $D$ . Applying the method described here can then be used to confirm that the mixture is stable if that is the case, or to correctly determine that it is really unstable if that is the case (Hua *et al.*, 1997).

## 5 Concluding Remarks

Results presented here demonstrate that interval analysis provides a completely reliable technique for solving phase stability problems involving excess Gibbs energy models. The technique is initialization independent, immune from rounding error, and provides both mathematical and computational guarantees that all stationary points in the tangent plane distance function are enclosed, thus eliminating computational problems that may occur in other currently used techniques. Though applied here in connection with the NRTL and UNIQUAC models, the technique used is general-purpose, straightforward to use, and can be applied in connection with any model. The enhancements described for improving the evaluation of interval extensions provides significant savings in computation time, approaching an order of magnitude reduction in some cases. For problems of the size considered here (up to five components), this means that computation times are quite reasonable. Of course, the computation time requirements are much greater than in the case of local methods that do not guarantee that the correct solution is found. Thus, as might be expected, there is a price to pay for knowing with certainty that the phase stability problem is correctly solved.

**Acknowledgments** – This work has been supported in part by the donors of The Petroleum Research Fund, administered by the ACS, under Grant 30421-AC9, by the National Science Foundation Grants CTS95-22835, DMI96-96110, and EEC97-00537-CRCD, by the Environmental Protection Agency Grant R824731-01-0, by the Department of Energy Grant DE-FG07-96ER14691, and by a grant from Sun Microsystems, Inc.

## References

- Baker, L. E., Pierce, A. C., and Luks, K. D. (1982) Gibbs energy analysis of phase equilibria. *Soc. Petrol. Engrs. J.* **22**, 731–742.
- Gmehling, J., Onken, U., and Arlt, W. (1977–1990) *Vapor-Liquid Equilibrium Data Collection*, volume I of *Chemistry Data Series*. DECHEMA, Frankfurt/Main, Germany.
- Green, K. A., Zhou, S., and Luks, K. D. (1993) The fractal response of robust solution techniques to the stationary point problem. *Fluid Phase Equilib.* **84**, 49–78.
- Hansen, E. R. (1992) *Global Optimization Using Interval Analysis*. Marcel Dekkar, Inc., New York.
- Hua, J. Z., Brennecke, J. F., and Stadtherr, M. A. (1996a) Reliable prediction of phase stability using an interval-Newton method. *Fluid Phase Equilib.* **116**, 52–59.
- Hua, J. Z., Brennecke, J. F., and Stadtherr, M. A. (1996b) Reliable phase stability analysis for cubic equation of state models. *Comput. Chem. Eng.* **20**, S395–S400.
- Hua, J. Z., Brennecke, J. F., and Stadtherr, M. A. (1997) Combined local and global approach to reliable computation of phase equilibria. Paper 80b, presented at AIChE Annual Meeting, Los Angeles, CA, November 16–21.
- Hua, J. Z., Brennecke, J. F., and Stadtherr, M. A. (1998) Enhanced interval analysis for phase stability: Cubic equation of state models. *Ind. Eng. Chem. Res.* **37**, 1519–1527.
- Kearfott, R. B. (1996) *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Kearfott, R. B., Dawande, M., Du, K.-S., and Hu, C.-Y. (1994) Algorithm 737: INTLIB, a portable FORTRAN 77 interval standard function library. *ACM Trans. Math. Software* **20**, 447–459.

- Kearfott, R. B. and Novoa, M. (1990) Algorithm 681: INTBIS, a portable interval Newton/bisection package. *ACM Trans. Math. Software* **16**, 152–157.
- McDonald, C. M. and Floudas, C. A. (1995a) Global optimization for the phase stability problem. *AIChE J.* **41**, 1798–1814.
- McDonald, C. M. and Floudas, C. A. (1995b) Global optimization for the phase and chemical equilibrium problem: Application to the NRTL equation. *Comput. Chem. Eng.* **19**, 1111–1139.
- McDonald, C. M. and Floudas, C. A. (1995c) Global optimization and analysis for the Gibbs free energy function using the UNIFAC, Wilson, and ASOG equations. *Ind. Eng. Chem. Res.* **34**, 1674–1687.
- McDonald, C. M. and Floudas, C. A. (1997) GLOPEQ: A new computational tool for the phase and chemical equilibrium problem. *Comput. Chem. Eng.* **21**, 1–23.
- McKinnon, K. I. M., Millar, C. G., and Mongeau, M. (1996) Global optimization for the chemical and phase equilibrium problem using interval analysis. In C. A. Floudas and P. M. Pardalos, editors, *State of the Art in Global Optimization: Computational Methods and Applications*, pp. 365–382, Dordrecht, The Netherlands. Kluwer Academic Publishers.
- Michelsen, M. L. (1982a) The isothermal flash problem. Part I: Stability. *Fluid Phase Equilib.* **9**, 1–19.
- Michelsen, M. L. (1982b) The isothermal flash problem. Part II: Phase-split calculation. *Fluid Phase Equilib.* **9**, 21–40.
- Moore, R. E. (1966) *Interval Analysis*. Prentice-Hall, Englewood Cliffs.
- Neumaier, A. (1990) *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, England.
- Schnepper, C. A. and Stadtherr, M. A. (1996) Robust process simulation using interval methods. *Comput. Chem. Eng.* **20**, 187–199.

- Sørensen, J. M. and Arlt, W. (1979–1987) *Liquid-Liquid Equilibrium Data Collection*, volume V of *Chemistry Data Series*. DECHEMA, Frankfurt/Main, Germany.
- Stadtherr, M. A., Schnepfer, C. A., and Brennecke, J. F. (1995) Robust phase stability analysis using interval methods. *AIChE Symp. Ser.* **91**(304), 356–359.
- Sun, A. C. and Seider, W. D. (1995) Homotopy-continuation method for stability analysis in the global minimization of the Gibbs free energy. *Fluid Phase Equilib.* **103**, 213–249.
- Tessier, S. R. (1997) *Enhanced Interval Analysis for Phase Stability: Excess Gibbs Energy Models*. Master's thesis, University of Notre Dame, Notre Dame, IN, USA.
- van Hentenryck, P., Michel, L., and Deville, Y. (1997) *Numerica: A Modeling Language for Global Optimization*. MIT Press, Cambridge, Mass.

Table 1: Problem 1: NRTL parameters (McDonald and Floudas, 1995a)

		$G_{ij}$		
$i \backslash j$	1	2	3	
1	1.0	1.2017478	1.0216786	
2	0.8066060	1.0	0.6490629	
3	0.4392221	0.1852084	1.0	

  

		$\tau_{ij}$		
1	0.0	-0.61259	-0.07149	
2	0.71640	0.0	0.90047	
3	2.7425	3.51307	0.0	

Table 2: Stationary points for *n*-propanol(1), *n*-butanol(2), water(3) system at a number of feeds using NRTL model.

<b>Feed</b> $(z_1, z_2, z_3)$	<b>Stationary Points</b> $(x_1, x_2, x_3)$	<i>D</i>
(0.148, 0.052, 0.80)	(0.144, $4.99 \times 10^{-2}$ , 0.807)	$4.5711 \times 10^{-8}$
	(0.148, 0.052, 0.800)	0.00000
	(0.114, 0.036, 0.850)	$-9.9851 \times 10^{-6}$
(0.12, 0.08, 0.80)	(0.130, $8.90 \times 10^{-2}$ , 0.781)	$-3.0693 \times 10^{-6}$
	( $5.97 \times 10^{-2}$ , $2.82 \times 10^{-2}$ , 0.912)	$-7.4818 \times 10^{-4}$
	(0.120, 0.080, 0.800)	0.00000
(0.13, 0.07, 0.80)	(0.130, 0.070, 0.800)	0.00000
	(0.138, $7.56 \times 10^{-2}$ , 0.787)	$-8.6268 \times 10^{-7}$
	( $7.38 \times 10^{-2}$ , $3.03 \times 10^{-2}$ , 0.896)	$-3.2762 \times 10^{-4}$
(0.12, 0.05, 0.83)	(0.158, $7.29 \times 10^{-2}$ , 0.770)	$-5.7360 \times 10^{-5}$
	( $9.40 \times 10^{-2}$ , $3.49 \times 10^{-2}$ , 0.871)	$-3.0088 \times 10^{-5}$
	(0.120, 0.050, 0.830)	0.00000

Table 3: Computational efficiency on  $n$ -propanol(1),  $n$ -butanol(2), water(3) system with NRTL model.

Feed $(z_1, z_2, z_3)$	Number of Root				CPU Time (sec)			
	Inclusion Tests				Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
(0.148, 0.052, 0.800)	8543	6125	4375	3982	0.739	0.670	0.580	0.654
(0.12, 0.08, 0.80)	4936	3039	2133	1816	0.445	0.358	0.305	0.320
(0.13, 0.07, 0.80)	5583	3624	2507	2195	0.498	0.409	0.350	0.385
(0.12, 0.05, 0.83)	5111	3027	2111	1756	0.454	0.348	0.303	0.312

Table 4: Problem 2: NRTL parameters (Gmehling *et al.*, 1977-1990)

		$G_{ij}$			
$i \setminus j$		1	2	3	4
1		1.0	0.34320	0.93449	0.96384
2		1.80967	1.0	1.02932	0.93623
3		0.56132	0.59659	1.0	0.32322
4		0.51986	0.22649	0.31656	1.0

  

		$\tau_{ij}$			
1		0.0	2.16486	0.23689	0.13060
2		-1.2007	0.0	-0.09730	0.19154
3		2.01911	1.73912	0.0	4.01932
4		2.31985	4.31706	4.09334	0.0

Table 5: Stationary points for *n*-propanol(1), *n*-butanol(2), benzene(3), water(4) system at a number of feeds using NRTL model.

<b>Feed</b> ( $z_1, z_2, z_3, z_4$ )	<b>Stationary Points</b> ( $x_1, x_2, x_3, x_4$ )	<i>D</i>
(0.148, 0.052, 0.600, 0.200)	( $4.61 \times 10^{-2}, 1.89 \times 10^{-2}, 0.916, 1.87 \times 10^{-2}$ ) (0.148, 0.052, 0.600, 0.200) ( $1.81 \times 10^{-2}, 6.20 \times 10^{-4}, 4.48 \times 10^{-3}, 0.977$ )	-0.03365 0.00000 -0.33982
(0.25, 0.25, 0.25, 0.25)	( $3.53 \times 10^{-2}, 5.73 \times 10^{-3}, 6.75 \times 10^{-3}, 0.952$ ) (0.133, $8.02 \times 10^{-2}, 5.20 \times 10^{-2}, 0.735$ ) (0.250, 0.250, 0.250, 0.250)	0.03079 0.06532 0.00000
(0.148, 0.052, 0.700, 0.100)	( $8.20 \times 10^{-2}, 3.07 \times 10^{-2}, 0.854, 3.29 \times 10^{-2}$ ) (0.148, 0.052, 0.700, 0.100) ( $2.41 \times 10^{-2}, 7.86 \times 10^{-4}, 4.74 \times 10^{-3}, 0.970$ )	$-3.1279 \times 10^{-3}$ 0.00000 -0.31097
(0.25, 0.15, 0.40, 0.20)	( $3.67 \times 10^{-2}, 2.98 \times 10^{-3}, 7.37 \times 10^{-3}, 0.953$ ) (0.195, $7.86 \times 10^{-2}, 0.114, 0.613$ ) (0.250, 0.150, 0.400, 0.200)	-0.03867 0.02268 0.00000
(0.25, 0.15, 0.35, 0.25)	( $3.32 \times 10^{-2}, 2.69 \times 10^{-3}, 6.71 \times 10^{-3}, 0.957$ ) (0.206, $9.47 \times 10^{-2}, 0.140, 0.560$ ) (0.250, 0.150, 0.350, 0.250)	-0.07363 0.01066 0.00000

Table 6: Computational efficiency on *n*-propanol(1), *n*-butanol(2), benzene(3), water(4) system with NRTL model.

Feed $(z_1, z_2, z_3, z_4)$	Number of Root Inclusion Tests				CPU Time (sec) Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
	(0.148, 0.052, 0.600, 0.200)	14,918	4774	4059	3392	2.599	1.058	1.124
(0.25, 0.25, 0.25, 0.25)	22,554	7824	6181	5075	3.944	1.727	1.722	1.655
(0.148, 0.052, 0.700, 0.100)	23,962	7858	6623	5383	4.138	1.719	1.834	1.748
(0.25, 0.15, 0.40, 0.20)	39,237	12,198	9980	7930	6.807	2.717	2.798	2.615
(0.25, 0.15, 0.35, 0.25)	40,532	12,042	9903	7824	7.059	2.714	2.802	2.592

Table 7: Problem 3: NRTL parameters (Gmehling *et al.*, 1977-1990)

		$G_{ij}$				
$i \backslash j$	1	2	3	4	5	
1	1.0	0.34320	0.93450	0.35902	0.96384	
2	3.91873	1.0	1.02931	1.41931	1.06238	
3	0.56132	0.59670	1.0	0.57907	0.36864	
4	1.03030	0.70216	0.86880	1.0	1.04035	
5	0.51986	0.21196	0.17857	0.55537	1.0	

  

		$\tau_{ij}$				
1	0.0	2.16486	0.23686	3.78001	0.13060	
2	-1.20070	0.0	-0.09730	-1.15187	-0.20374	
3	2.01911	1.73912	0.0	1.85228	3.73758	
4	-0.10979	1.16315	0.47676	0.0	-0.14651	
5	2.31985	5.22337	6.45226	2.17820	0.0	

Table 8: Stationary points for *n*-propanol(1), *n*-butanol(2), benzene(3), ethanol(4), water(5) system at a number of feeds using NRTL model.

<b>Feed</b> $(z_1, z_2, z_3, z_4, z_5)$	<b>Stationary Points</b> $(x_1, x_2, x_3, x_4, x_5)$	<i>D</i>
(0.148, 0.052, 0.500, 0.100, 0.200)	$(6.98 \times 10^{-2}, 2.26 \times 10^{-2}, 0.811, 5.15 \times 10^{-2}, 4.52 \times 10^{-2})$ $(2.43 \times 10^{-2}, 5.45 \times 10^{-4}, 1.73 \times 10^{-3}, 3.55 \times 10^{-2}, 0.938)$ (0.148, 0.052, 0.500, 0.100, 0.200)	$-4.2107 \times 10^{-3}$ -0.10430 0.00000
(0.148, 0.052, 0.540, 0.08, 0.180)	$(6.90 \times 10^{-2}, 2.62 \times 10^{-2}, 0.822, 4.30 \times 10^{-2}, 4.33 \times 10^{-2})$ (0.148, 0.052, 0.540, 0.08, 0.180) $(2.31 \times 10^{-2}, 4.81 \times 10^{-4}, 1.42 \times 10^{-3}, 2.89 \times 10^{-2}, 0.946)$	$-4.4793 \times 10^{-3}$ 0.00000 -0.12840
(0.148, 0.052, 0.560, 0.08, 0.160)	$(7.99 \times 10^{-2}, 2.68 \times 10^{-2}, 0.794, 4.85 \times 10^{-2}, 5.04 \times 10^{-2})$ (0.148, 0.052, 0.560, 0.08, 0.160) $(2.49 \times 10^{-2}, 5.52 \times 10^{-4}, 1.59 \times 10^{-3}, 3.14 \times 10^{-2}, 0.942)$	$-1.9581 \times 10^{-3}$ 0.00000 -0.10682
(0.148, 0.052, 0.500, 0.120, 0.180)	(0.108, $3.68 \times 10^{-2}$ , 0.684, $8.60 \times 10^{-2}$ , $8.60 \times 10^{-2}$ ) (0.119, $4.13 \times 10^{-2}$ , 0.639, $9.46 \times 10^{-2}$ , 0.105) $(2.95 \times 10^{-2}, 8.25 \times 10^{-4}, 2.71 \times 10^{-3}, 4.93 \times 10^{-2}, 0.918)$ (0.136, $3.92 \times 10^{-2}$ , 0.152, 0.156, 0.518) (0.148, 0.052, 0.500, 0.120, 0.180)	$1.0396 \times 10^{-4}$ $1.1075 \times 10^{-4}$ -0.04748 $6.2085 \times 10^{-3}$ 0.00000
(0.148, 0.052, 0.520, 0.100, 0.180)	$(7.96 \times 10^{-2}, 2.63 \times 10^{-2}, 0.784, 5.77 \times 10^{-2}, 5.24 \times 10^{-2})$ (0.148, 0.052, 0.520, 0.100, 0.180) $(2.60 \times 10^{-2}, 6.18 \times 10^{-4}, 1.93 \times 10^{-3}, 3.83 \times 10^{-2}, 0.933)$ (0.163, $5.64 \times 10^{-2}$ , 0.397, 0.116, 0.267) (0.162, $5.32 \times 10^{-2}$ , 0.271, 0.128, 0.385)	$-1.9019 \times 10^{-3}$ 0.00000 -0.08658 $-1.0107 \times 10^{-4}$ $8.2373 \times 10^{-5}$

Table 9: Computational efficiency on *n*-propanol(1), *n*-butanol(2), benzene(3), ethanol(4), water(5) system with NRTL model.

Feed $(z_1, z_2, z_3, z_4, z_5)$	Number of Root Inclusion Tests				CPU Time (sec) Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
(0.148, 0.052, 0.50, 0.10, 0.20)	493,833	98,057	73,756	58,733	147.98	37.77	36.16	32.97
(0.148, 0.052, 0.54, 0.08, 0.18)	387,526	77,118	58,174	45,968	115.50	29.44	28.24	25.51
(0.148, 0.052, 0.56, 0.08, 0.16)	569,988	112,822	84,689	66,897	170.56	43.56	41.57	37.54
(0.148, 0.052, 0.50, 0.12, 0.18)	1,612,909	437,783	310,894	240,244	471.00	166.11	148.45	131.99
(0.148, 0.052, 0.52, 0.10, 0.18)	857,095	173,687	129,540	103,053	258.50	67.97	64.64	58.98

Table 10: Problem 4: UNIQUAC parameters (McDonald and Floudas, 1995a)

$i \backslash j$	$\tau_{ij}$		
	1	2	3
1	1.0	0.432589	0.830749
2	0.789593	1.0	0.354992
3	0.204736	0.636678	1.0

  

$r_j$		
2.4088	8.8495	2.0086

  

$q_j = q'_j$		
2.2480	7.3720	1.8680

Table 11: Stationary points for ethylene glycol(1), lauryl alcohol(2), nitromethane(3) system at a number of feeds using UNIQUAC model.

<b>Feed</b> $(z_1, z_2, z_3)$	<b>Stationary Points</b> $(x_1, x_2, x_3)$	<i>D</i>
(0.27078, 0.47302, 0.25620)	(0.620, $5.62 \times 10^{-3}$ , 0.374)	$-4.6119 \times 10^{-6}$
	(0.369, $1.53 \times 10^{-2}$ , 0.615)	$7.9578 \times 10^{-3}$
	( $2.33 \times 10^{-2}$ , $1.73 \times 10^{-3}$ , 0.975)	-0.05876
	(0.27078, 0.47302, 0.25620)	0.00000
	(0.347, 0.157, 0.496)	0.03454
(0.40, 0.30, 0.30)	(0.754, $2.22 \times 10^{-3}$ , 0.244)	-0.11395
	(0.190, $1.03 \times 10^{-2}$ , 0.799)	-0.01094
	( $3.59 \times 10^{-2}$ , $2.05 \times 10^{-3}$ , 0.962)	-0.02711
	(0.409, 0.284, 0.308)	$5.1978 \times 10^{-6}$
	(0.400, 0.300, 0.300)	0.00000
(0.30, 0.40, 0.30)	(0.629, $4.70 \times 10^{-3}$ , 0.366)	-0.04575
	(0.359, $1.31 \times 10^{-2}$ , 0.628)	-0.03536
	( $2.34 \times 10^{-2}$ , $1.55 \times 10^{-3}$ , 0.975)	-0.10136
	(0.349, 0.206, 0.444)	$-7.9580 \times 10^{-3}$
	(0.300, 0.400, 0.300)	0.00000
(0.30, 0.30, 0.40)	( $2.00 \times 10^{-2}$ , $1.32 \times 10^{-3}$ , 0.979)	-0.15664
	(0.298, 0.315, 0.387)	$-3.8209 \times 10^{-6}$
	(0.300, 0.300, 0.400)	0.00000

Table 12: Computational efficiency on ethylene glycol(1), lauryl alcohol(2), nitromethane(3) system with UNIQUAC model.

Feed $(z_1, z_2, z_3)$	Number of Root				CPU Time (sec)			
	Inclusion Tests				Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
(0.27078, 0.47302, 0.25620)	9651	4307	4022	3535	1.093	0.681	0.703	0.829
(0.40, 0.30, 0.30)	18,577	10,105	9563	6618	2.082	1.380	1.575	1.459
(0.30, 0.40, 0.30)	10,183	4553	4228	3623	1.154	0.651	0.738	0.847
(0.30, 0.30, 0.40)	15,055	7781	7338	5635	1.689	1.073	1.223	1.262

Table 13: Problem 5: UNIQUAC parameters (Sørensen and Arlt, 1979–1987)

$i \setminus j$	$\tau_{ij}$			
	1	2	3	4
1	1.0	1.26362	3.36860	0.85128
2	0.99972	1.0	1.02041	0.89333
3	0.31633	0.79027	1.0	0.96249
4	0.49739	1.09619	0.26222	1.0

  

$r_j$			
2.2024	3.1878	3.1680	4.0464

  

$q_j = q'_j$			
2.072	2.400	2.484	3.240

Table 14: Stationary points for acetic acid(1), benzene(2), fufural (3), cyclohexane(4) system at a number of feeds using UNIQUAC model.

<b>Feed</b> $(z_1, z_2, z_3, z_4)$	<b>Stationary Points</b> $(x_1, x_2, x_3, x_4)$	$D$
(0.25, 0.25, 0.25, 0.25)	(0.25, 0.25, 0.25, 0.25)	0.0000
(0.05, 0.20, 0.35, 0.40)	$(1.75 \times 10^{-2}, 0.200, 0.134, 0.649)$ $(6.14 \times 10^{-2}, 0.187, 0.430, 0.321)$ (0.050, 0.200, 0.350, 0.400)	$-4.9318 \times 10^{-3}$ $-2.2164 \times 10^{-4}$ 0.00000
(0.05, 0.21, 0.34, 0.40)	$(2.04 \times 10^{-2}, 0.212, 0.149, 0.618)$ $(5.98 \times 10^{-2}, 0.199, 0.407, 0.334)$ (0.050, 0.210, 0.340, 0.400)	$-2.8447 \times 10^{-3}$ $-1.1141 \times 10^{-4}$ 0.00000
(0.05, 0.22, 0.33, 0.40)	$(2.42 \times 10^{-2}, 0.225, 0.169, 0.582)$ $(5.75 \times 10^{-2}, 0.213, 0.378, 0.351)$ (0.050, 0.220, 0.330, 0.400)	$-1.3244 \times 10^{-3}$ $-3.7713 \times 10^{-5}$ 0.00000
(0.05, 0.23, 0.32, 0.40)	$(2.97 \times 10^{-2}, 0.263, 0.197, 0.537)$ $(5.35 \times 10^{-2}, 0.227, 0.342, 0.378)$ (0.050, 0.230, 0.320, 0.400)	$-3.6123 \times 10^{-4}$ $-2.5032 \times 10^{-6}$ 0.00000

Table 15: Computational efficiency on acetic acid(1), benzene(2), furfural(3), cyclohexane(4) system with UNIQUAC model.

Feed $(z_1, z_2, z_3, z_4)$	Number of Root Inclusion Tests				CPU Time (sec) Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
	(0.25, 0.25, 0.25, 0.25)	33,971	4976	4209	4209	7.517	1.162	1.240
(0.05, 0.20, 0.35, 0.40)	69,092	8190	7095	7095	14.59	1.906	2.103	2.191
(0.05, 0.21, 0.34, 0.40)	78,789	9919	8491	8491	16.61	2.311	2.512	2.617
(0.05, 0.22, 0.33, 0.40)	93,380	12,657	10,955	10,955	19.77	2.958	3.278	3.411
(0.05, 0.23, 0.32, 0.40)	126,174	30,511	26,947	26,947	26.65	7.092	7.927	8.269

Table 16: Problem 6: UNIQUAC parameters (Sørensen and Arlt, 1979–1987)

$i \backslash j$	$\tau_{ij}$				
	1	2	3	4	5
1	1.0	1.26362	3.36860	0.85128	1.54662
2	0.99972	1.0	1.02041	0.89333	0.09441
3	0.31633	0.79027	1.0	0.96249	0.60488
4	0.49739	1.09619	0.26222	1.0	0.08839
5	2.44225	0.13507	0.69066	0.19491	1.0

  

$r_j$				
2.2024	3.1878	3.1680	4.0464	0.9200

  

$q_j = q_j'$				
2.072	2.400	2.484	3.240	1.400

Table 17: Stationary points for acetic Acid(1), benzene(2), furfural(3), cyclohexane(4), water(5) system at a number of feeds using UNIQUAC model.

<b>Feed</b> $(z_1, z_2, z_3, z_4, z_5)$	<b>Stationary Points</b> $(x_1, x_2, x_3, x_4, x_5)$	<i>D</i>
(0.20, 0.20, 0.20 0.20, 0.20)	$(3.04 \times 10^{-2}, 0.302, 3.44 \times 10^{-2}, 0.622, 1.06 \times 10^{-2})$ $(0.227, 3.34 \times 10^{-3}, 4.75 \times 10^{-2}, 1.58 \times 10^{-3}, 0.720)$ (0.200, 0.200, 0.200, 0.200, 0.200)	-0.11354 -0.17765 0.00000
(0.20, 0.25, 0.20 0.15, 0.20)	$(0.217, 3.66 \times 10^{-3}, 4.08 \times 10^{-2}, 1.04 \times 10^{-3}, 0.737)$ $(4.35 \times 10^{-2}, 0.412, 5.50 \times 10^{-2}, 0.475, 1.47 \times 10^{-2})$ (0.200, 0.250, 0.200, 0.150, 0.200)	-0.17697 -0.07454 0.00000
(0.20, 0.25, 0.25 0.15, 0.15)	$(0.270, 1.43 \times 10^{-2}, 0.118, 4.83 \times 10^{-3}, 0.593)$ $(6.89 \times 10^{-2}, 0.393, 9.84 \times 10^{-2}, 0.420, 2.03 \times 10^{-2})$ (0.200, 0.250, 0.250, 0.150, 0.150)	-0.06565 -0.02410 0.00000
(0.10, 0.25, 0.25 0.15, 0.25)	$(5.69 \times 10^{-2}, 1.38 \times 10^{-4}, 7.32 \times 10^{-3}, 4.54 \times 10^{-5}, 0.936)$ $(1.10 \times 10^{-2}, 0.433, 6.78 \times 10^{-2}, 0.471, 1.73 \times 10^{-2})$ (0.100, 0.250, 0.250, 0.150, 0.250)	-0.44049 -0.10803 0.00000
(0.15, 0.25, 0.25 0.10, 0.25)	$(0.117, 7.69 \times 10^{-4}, 1.81 \times 10^{-2}, 1.61 \times 10^{-4}, 0.864)$ $(2.63 \times 10^{-2}, 0.497, 8.43 \times 10^{-2}, 0.375, 1.72 \times 10^{-2})$ (0.150, 0.250, 0.250, 0.100, 0.250)	-0.24901 -0.09182 0.00000

Table 18: Computational efficiency on acetic Acid(1), benzene(2), furfural(3), cyclohexane(4), water(5) system with UNIQUAC model.

Feed $(z_1, z_2, z_3, z_4, z_5)$	Number of Root				CPU Time (sec)			
	Inclusion Tests				Sun Ultra 2/1300			
	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$	$F$	$F^{CS}$	$F^{CSI}$	$F^{CSM}$
(0.20, 0.20, 0.20, 0.20, 0.20)	1,999,134	618,759	562,050	311,745	648.64	196.70	233.93	143.68
(0.20, 0.25, 0.20, 0.15, 0.20)	2,086,106	712,531	649,510	352,054	657.67	224.83	268.29	161.25
(0.20, 0.25, 0.25, 0.15, 0.15)	3,704,189	1,285,709	1,179,217	648,875	1201.8	406.69	488.56	296.40
(0.10, 0.25, 0.25, 0.15, 0.25)	1,294,290	255,134	224,104	114,753	423.12	85.17	98.38	55.28
(0.15, 0.25, 0.25, 0.10, 0.25)	1,645,807	456,376	408,388	214,395	536.58	183.85	172.46	100.63