Global Optimization for the Chemical and Phase Equilibrium Problem using Interval Analysis

K.I.M. McKinnon, C. Millar and M. Mongeau

March 1995

MS 95-001a

Supported by EPSRC research grant RR31817/18

A revised version of MS 95-001. Given in the Princeton conference on global optimization, 1995, and publised in "Global optimization" Editors C. Floudas and P.M.Pardalos, Kluwer Academic Press, pp365-382, 1995.

Department of Mathematics and Statistics
University of Edinburgh, The King's Buildings, Edinburgh EH9 3JZ
Tel. (33) 131 650 5075 E-Mail : ken@maths.ed.ac.uk, mongeau@cict.fr

Global Optimization for the Chemical and Phase Equilibrium Problem using Interval Analysis

K.I.M MCKINNON, C. MILLAR, M. MONGEAU ken@maths.ed.ac.uk, mongeau@cict.fr
Dept of Mathematics and Statistics, University of Edinburgh EH9 3JZ, UK
Labo Approximation & Optimisation, Université Paul Sabatier, 31062 Toulouse cedex, France

Abstract. This paper addresses the problem of minimizing the Gibbs free energy in the n_c-component, multi-phase chemical and phase equilibrium problem involving different thermodynamic models. The algorithmic approach used is based on the tangent-plane criterion of Gibbs: the global optimization problem considered, which involves a search space of $n(n+1)$ dimensions, is reduced to a finite sequence of *global* optimization steps in n-space, and *local* optimization steps in nK-space, $K \leq n+1$.

We describe an algorithm performing the global optimization step involved in this lower-dimensional search space using techniques from interval analysis. We report good numerical results on instances of the Gibbs free energy minimization problem.

Keywords: Global optimization, interval analysis, tangent-plane criterion, Gibbs free energy, chemical and phase equilibrium, non-convex optimization.

1. Introduction

The problem of phase equilibrium is: Given the total quantities (in number of moles) of n_c components, $Z_i > 0$, $i = 1, \ldots, n_c$, determine the number, K, and *type* (e.g. the state; liquid, vapour) of phases existing at equilibrium and the composition (number of moles of each component), $\{z_{ik}\}_{i=1}^{n_c}$, of each phase k, $1 \leq k \leq K$. In the problem of chemical phase equilibrium (i.e. involving reaction), the n_c components are constituted of m chemical elements, and the $\{Z_i\}_{i=1}^{m}$ are the total quantities of each of the chemical elements [12].

These equilibrium quantities can be obtained by minimizing globally a thermodynamic function, the Gibbs free energy (GFE). This is a problem which is known to be numerically difficult to solve even in the case where the number of phases K present at equilibrium is known, and even for instances involving as few as $n_c = 2$ components and $K = 2$ phases [13].

We consider n_c-component chemical and phase equilibrium problems which involve different thermodynamic functions to model the different types of phase that may be present at equilibrium. Also, this will require us to make use of an extension, presented in [8], of the tangent-plane criterion [1], [2], [3], [5]. The algorithmic approach we use is also taken from [8]: the problem is reduced to a finite sequence of *local* optimization steps, and of *global* optimization steps in a lower-dimensional space. These results are summarized in section 2.

Related work has been done by Michelsen [9], [10] who uses alternately a local method to obtain a stationary point of the GFE corresponding to a given number

of phases, and a stability test derived from the tangent-plane-criterion. He however uses a *local* method to attempt to determine whether a phase is stable, making use of several initial estimates. Also, Michelsen concentrates on phase equilibrium problems in which all phases are described with the same thermodynamic model. In our approach, different thermodynamic models for the different types of phase present at equilibrium are allowed. Moreover, the way we make use of the tangent-plane criterion is different. We use the tangent-plane criterion either to add an extra phase, to improve the Gibbs free energy without changing the number of phases, or to establish global optimality.

The unspecified global optimization subroutine of the generic algorithm, GILO (Global Improvement - Local Optimization), in [8] for the global minimization step in the lower-dimensional space will here be implemented using techniques from interval analysis [4], [11]. We survey, in section 3, the techniques we use for the global step. Section 4 reports numerical results obtained from the implementation of our algorithm on 2-component and 3-component phase equilibrium problems, and we compare our results with those obtained by McDonald & Floudas in [7]. We present some conclusions in Section 5.

We consider the following global optimization problem, which we shall refer to as problem P:

$$\inf \quad F(K,t,Y) \equiv \sum_{k=1}^{K} y_k f_{t_k}(x_k)$$

$$\begin{aligned} \text{subject to} \quad \sum_{k=1}^{K} y_k A x_k &= b \\ \sum_{k=1}^{K} y_k &= 1 \\ y_k &\geq 0, \quad k = 1,2,\ldots,K \\ x_k &\in S, \quad k = 1,2,\ldots,K \\ t_k &\in \tau, \quad k = 1,2,\ldots,K \\ K &\in I\!N^+, \end{aligned} \tag{1}$$

where $Y = (y_1,\ldots,y_K,x_1^T,\ldots,x_K^T) \in I\!R^{K(n+1)}$, A is an $m \times n$ real matrix of rank $m \leq n$, $0 \neq b \in I\!R^m$, S is an open convex subset of $(I\!R^+)^n$, τ is a finite index set, and, for $k = 1,2,\ldots,K$, $t_k \in \tau$, $y_k \in I\!R$, $x_k \in I\!R^n$, and f_{t_k} is a differentiable real-valued function defined over S satisfying

$$\lim_{x \to \bar{x}} d^T \nabla f(x) = \infty, \tag{2}$$

for any $\bar{x}$ in the boundary of X, where d is a direction from the interior of S and pointing towards $\bar{x}$. Note that in the above optimization problem we are interested

in a solution for which K is smallest, amongst all the possible global minima (to avoid obtaining a solution with several phases having identical compositions).

The n_c-component phase equilibrium problem (PEP) and the n_c-component, m-element chemical equilibrium problem (CEP), can be shown to be special cases of problem P (see [5]). The ith component of vector x_k:

$$x_{ik} \equiv \frac{z_{ik}}{\sum_{i=1}^{n_c} z_{ik}} \quad i = 1, 2, \ldots, n_c;\ k = 1, 2, \ldots, K, \tag{3}$$

denotes the *molar fraction* of component i in phase k. Since $\sum_{i=1}^{n_c} x_{ik} = 1$, $k = 1, 2, \ldots, K$, and here $x_{ik} > 0$, $i = 1, 2, \ldots, n_c$; $k = 1, 2, \ldots, K$ is assumed (see [5]), we eliminated the variables $x_{n_c k}$, $k = 1, 2, \ldots, K$, and considered the feasible set

$$S = \{x_k \in I\!R^n : \sum_{i=1}^{n} x_{ik} < 1,\ x_{ik} > 0\},$$

where $n = n_c - 1$. Let the total number of moles be defined by

$$Z \equiv \sum_{i=1}^{n_c} Z_i, \tag{4}$$

where Z_i is the number of moles of component i in the PEP case, and by $Z \equiv \sum_{i=1}^{m} Z_i$, where Z_i is the number of moles of element i in the CEP case. The set τ is the set of phase types, and, for $k = 1, 2, \ldots, K$: t_k is the type of phase k, y_k is the molar fraction of phase k (given by $y_k = \frac{\sum_{i=1}^{n_c} z_{ik}}{Z}$ for the PEP), F is the GFE of the system with K phases of types $t_1, t_2, \ldots, t_K$, and f_t is the molar GFE of a phase of type t. In the PEP, $m = n$, $A = I$, and the ith entry of vector b is the total molar fraction of component i, $i = 1, 2, \ldots, n$. In the CEP, A is a matrix related to the *formula matrix* whose entries give the number of atoms of each chemical element per molecule of each component, and the ith entry of vector b is the total molar fraction of *element* i, $i = 1, 2, \ldots, m$.

We assume that the problem has at least one feasible solution. For the PEP and the CEP, we have that (2) holds. This implies we can assume [5] that any local *infimum* of P is a local *minimum*.

Note finally that, when using the z_{ik} variables (number of moles) instead of our molar-fraction variables, problem P can be formulated with linear constraints rather than with bilinear constraints (as in this paper). We shall see in the next section that the convexity structure of the latter formulation gives a clearer geometric interpretation of the problem. In the practical implementation these bilinear constraints will not be dealt with explicitly. They are transformed back into the number-of-mole variables formulation in the case of the local optimization subproblem. On the other hand, the global optimization subproblem will only involve box constraints and one linear inequality constraint (corresponding to the set S above).

2. GILO: A Generic Global Minimization Algorithm for the Chemical and Phase Equilibrium Problem

In this section, we summarize the results of [8] which produced the GILO (Global Improvement - Local Optimization) algorithm.

The number of phases, K, that may exists at equilibrium is unknown a priori. Let $P_{\tilde{t}}^{\tilde{K}}$ denote problem P in which K is fixed to a positive integer $\tilde{K}$ and t is fixed to $\tilde{t} \in \tau^{\tilde{K}}$, and let $F_{\tilde{t}}^{\tilde{K}} : I\!R^{\tilde{K}(n+1)} \to I\!R$ be the corresponding objective function.

Lemma 1 *If Y^* is a Kuhn-Tucker point for $P_{t^*}^{K^*}$, for some positive integer K^* and some $t^* \in \tau^{K^*}$, then there exists an m-dimensional hyperplane Θ^* which is tangent to $f_{t_k^*}$ at the points $\{x_k^* : 1 \leq k \leq K^* \text{ and } y_k^* > 0\}$.*

Let $f_M : S \to I\!R$ be the (nonsmooth) function defined by

$$f_M(x) \equiv \min_{t \in \tau} f_t(x).$$

Lemma 2 *Let (K^+, t^+, Y^+), be a global minimum for P, $K^+ \in I\!N^+$, $t^+ \in \tau^{K^+}$, and $Y^+ \in I\!R^{K^+(n+1)}$, and Θ^+ be the corresponding tangent hyperplane described by Lemma 1. Then we have that, at the points $\{x_k^+ : 1 \leq k \leq K^+ \text{ and } y_k^+ > 0\}$, f_M is differentiable and Θ^+ is tangent to f_M.*

We have the following characterization of global minima for P in terms of the function f_M.

Theorem 1 (Tangent-Plane Criterion) *Let $K^+ \in I\!N^+$, $t^+ \in \tau^{K^+}$, and $Y^+ \in I\!R^{K^+(n+1)}$.*

If (K^+, t^+, Y^+) is a global minimum for P, then Y^+ is a Kuhn-Tucker point for $P_{t^+}^{K^+}$ and

$$f_M(x) \geq \Theta_M^+(x), \quad \text{for all } x \in S, \tag{5}$$

where Θ_M^+ is the tangent hyperplane corresponding to (K^+, t^+, Y^+), as described by Lemma 2.

Conversely, if Y^+ is a Kuhn-Tucker point for $P_{t^+}^{K^+}$ and

$$f_M(x) \geq \Theta^+(x), \quad \text{for all } x \in S, \tag{6}$$

where Θ^+ is the tangent hyperplane corresponding to (K^+, t^+, Y^+) as described by Lemma 1, then (K^+, t^+, Y^+) is a global minimum for P and $\Theta^+ = \Theta_M^+$.

We know from the following proposition that we do not need to go beyond $K = m + 1$ phases in order to solve P.

Proposition 1 *If $Y \in I\!R^{K(n+1)}$ is a Kuhn-Tucker point for P_t^K, for some fixed positive integer K and some $t \in \tau^K$, then there exists a solution $\tilde{Y} \in I\!R^{(m+1)(n+1)}$ for $P_{\tilde{t}}^{m+1}$, for some $\tilde{t} \in \tau^{m+1}$, such that $F_{\tilde{t}}^{m+1}(\tilde{Y}) = F_t^K(Y)$.*

Let LO be a local minimization subroutine which, given the current feasible solution, (K, t, Y), for P, returns a Khun-Tucker point $Y^* := \mathrm{LO}(K, t, Y) \in I\!R^{K(n+1)}$, for P_t^K with at least as good a function value as the initial point. Let GI be a global optimization subroutine (for a lower-dimensional search space) which, given a linear function, $\Theta^* : I\!R^n \to I\!R$, returns $(t_0, x_0) := \mathrm{GI}(\Theta^*)$, where either $f_M(x_0) < \Theta(x_0)$ or x_0 is a global minimum for the subproblem

$$\min_{x \in S} \; f_M(x) - \Theta^*(x), \tag{7}$$

and where $t_0 \in \tau$ is such that $f_{t_0}(x_0) = f_M(x_0)$, in which case $f_M(x_0) = \Theta(x_0)$.

GILO Algorithm

Step 0: [Initialization]
Let $K^* = 1$, and let t^* be an arbitrary element of τ.
Let Y^* be a feasible solution to $P_{t^*}^{K^*}$.
Step 1: [Local Optimization]
$Y^* := \mathrm{LO}(K, t, Y)$.
$t_k^* := t_k, \quad 1 \le k \le K$.
$K^* :=$ number of non-zero *distinct* phases (with the rest renumbered if necessary).
Step 2: [Global Improvement]
Let $1 \le j \le K^*$ be arbitrary.
Let $\Theta^* : I\!R^n \to I\!R$ be defined by $\Theta^*(x) := \nabla f_{t_j^*}(x_j^*)^T (x - x_j^*) + f_{t_j^*}(x_j^*)$,
$(t_0, x_0) := \mathrm{GI}(\Theta^*)$.
Step 3: [Global Optimality Condition]
If $f_M(x_0) \ge \Theta^*(x_0)$, then stop: (K^*, t^*, Y^*) is a global minimum for P.
Step 4: [Force improvement]
Solve

$$\sum_{k=1}^{K^*} \alpha_k = 1, \;\; \sum_{k=1}^{K^*} \alpha_k A x_k^* = A x_0 \tag{8}$$

for $\{\alpha_k\}_{k=1}^{K^*}$.

- (i) [Interchange a phase]
  If (8) has a solution, $\{\alpha_k\}_{k=1}^{K^*}$:
  The new feasible solution to P is (K, t, Y) given by

$$\begin{aligned} t_i &:= t_0, \\ y_i := \frac{y_i^*}{\alpha_i} \;\; &, \;\; x_i := x_0, \\ t_k := t_k^*, \quad y_k := y_k^* - \frac{y_i^*}{\alpha_i}\alpha_k \;\; &, \;\; x_k := x_k^*, \quad 1 \le k \le K^*, \; k \ne i, \end{aligned}$$

where

$$i \equiv \text{argmin}\{\frac{y_k^*}{\alpha_k} : \alpha_k > 0, 1 \leq k \leq K^*\}.$$

$K :=$ number of phases with non-zero y_k^* (with the rest renumbered if necessary).

- (ii) [Split a phase]

  Otherwise (the system (8) has no solution):

  Let $1 \leq l \leq K^*$ be arbitrary.

  Let $\epsilon_0 > 0$ be small enough so that strict decrease in F is achieved and such that the new solution (K, t, Y) given below is feasible for P.

$$\begin{aligned}
K &:= K^* + 1 \quad , \quad t_{K^*+1} := t_0, \\
y_{K^*+1} &:= \frac{\epsilon_0}{\epsilon_0 + 1} y_l^* \quad , \quad x_{K^*+1} := x_0, \\
y_l &:= \frac{1}{\epsilon_0 + 1} y_l^* \quad , \quad x_l := \epsilon_0 (x_l^* - x_0) + x_l^*, \\
t_k &:= t_k^* \quad , \quad 1 \leq k \leq K^*, \\
y_k &:= y_k^* \quad , \quad x_k := x_k^*, \quad 1 \leq k \leq K^*, \; k \neq l.
\end{aligned}$$

  Go to Step 1.

Note that if $m \geq n$, then finding a single-phase starting solution Y^* (Step 0), reduces to solving the equations to find the unique feasible solution (in the case with no reaction [$m = n$ and $A = I$], Y^* is straightforwardly given by $y_1^* = 1$ and $x_1^* = b$). This starting solution, Y^*, will then be a local minimum of problem $P_{t^*}^{K^*}$, and Step 1 does not need to be performed in the first iteration.

The phase-stability test (Steps 2 and 3) uses the tangent-plane criterion to determine whether the current solution is optimal, and, if it is not, it finds an improved feasible solution either with the same number or fewer phases [Step 4 (i)], or with one added phase [Step 4 (ii)], and it also determines what type of phase is to be added. The number of phases may indeed be *reduced* at the end of Step 4 (i) (or at the end of Step 1) by eliminating the phases corresponding to null y_k's (and renumbering the remaining phases). Also, at the end of Step 1, *spurious* phases (i.e the proportions of the components of two phases of same type are identical), are *merged* into one phase, and the number of phases is hence *decreased*. Step 4 (ii) corresponds physically to *splitting* a phase (phase x_l) into two phases (the new phase x_l and phase x_0). Hence, after performing this step, K is always increased by one. In Step 4 (i) we *interchange* the phase x_i for the new phase x_0. In both cases, a strict decrease of F is obtained [if ϵ_0 is chosen small enough, for Step 4 (ii)] and we leave the current local minimum [8].

Theorem 2 (Convergence) *The GILO Algorithm converges to a global minimum of P, from any point (K^0, t^0, Y^0) which is feasible for P and such that $K^0 = 1$, in a finite number of steps, provided that the following three conditions hold:*
(i) P_t^K has a finite number of distinct objective function values at Kuhn-Tucker points, for any $1 \leq K \leq m+1$ and any $t \in \tau^K$.
(ii) The subroutine GI (Global Improvement) returns any *point x_0 such that*

$$f_t(x_0) < \Theta^*(x_0),$$

for some $t \in \tau$ (or such that $f_M(x_0) < \Theta^*(x_0)$) if such a point exists, or it proves that such a point does not exist.
(iii) The subroutine LO (local optimization) returns a Kuhn-Tucker point whose objective function value is not greater than at its starting point.

Note that it is possible (though unlikely) for a local minimization subroutine to terminate at a Khun-Tucker point which is not a local minimum. Theorem 2 shows that convergence will occur even if this occurs provided there is finite number of distinct objective function values at Khun-Tucker points. Also the GI (Global Improvement) step can terminate when it finds *any* point x_0 such that $f_t(x_0) < \Theta^*(x_0)$, for some $t \in \tau$ [or such that $f_M(x_0) < \Theta^*(x_0)$], so in practise the only global optimization which has to be completed is the final one. In most problems this is a considerable saving.

Theorem 2 is a minor generalization of the convergence theorem in [8] and the proof follows along the same lines.

3. Global Optimization using Interval Analysis

In this section we describe a deterministic method for global optimization, guaranteed to find an *ϵ-global solution* (i.e. whose objective function value is within $\epsilon > 0$ of a globally optimal objective function value), which is based on interval analysis. With this approach, one recursively partitions the search space until each subdomain either can be shown not to contain a global solution, or to contain only ϵ-global solutions. To achieve this aim, one makes use of bounds on the function, gradient and/or Hessian values over the subdomains within a branch-and-bound (B&B) scheme.

The interval-analysis approach is an attractive method to address the global optimization subproblem (7) for many reasons. Firstly, as we mentioned above, Step 2 of the GILO Algorithm does not need to perform a *thorough* global search in order to escape the current local optimum: we can use interval-analysis techniques only until we obtain any point x_0 violating the tangent-plane criterion. We do not need a global minimum for the subproblem (7), but only any point such that the objective function value of this problem is negative. Since the objective function takes the value zero at each of the current tangent points, the B&B search can be started with an upper bound of zero on the optimal solution. Any subdomain over which this objective function is shown to be non-negative will then be rejected

at once. Secondly, since the search space for the global optimization subproblem is low dimensional (n-space versus $(m+1)n$-space for the original problem), the subdividing step for its recursive partition should not be prohibitive. Thirdly, most of the work in the B&B is the same from one global optimization subproblem to another (as the corresponding objective functions differ only, through the tangent planes, by an affine amount): in subsequent searches the gradients differ by a constant amount, and the Hessians are the same. This feature could be exploited by storing these quantities from one B&B search to another.

Let I be a bounded interval. The lower and upper bounds on I are denoted by $\underline{I}$ and $\bar{I}$, so $I = [\underline{I}, \bar{I}]$. We call a vector of intervals a *box*. Let f be a continuous real-valued function defined over $\mathbb{R}$, and $X \subset \mathbb{R}^n$ be a box. We define the *range* of f for $x \in X$, denoted by $f^R(X)$, to be the interval

$$f^R(X) \equiv [\inf_{x \in X} f(x), \sup_{x \in X} f(x)]$$

(where this is understood as a semi-infinite interval or as an infinite interval, according to whether one or both bounds are $\pm\infty$). When using the *interval arithmetic* [4], [11] (e.g. if X and Y are bounded intervals, $X - Y = [\underline{X} - \bar{Y}, \bar{X} - \underline{Y}]$), we always obtain a *valid* bound, B, for a function f in a box X, i.e such that $f^R(X) \subseteq B$. However, in general we do not necessarily get *sharp* bounds (we do not necessarily have $f^R(X) = B$). We call an interval-valued function F having as argument a box of $\mathbb{R}^n$, an *(interval-)extension* of f if $F([x_1, x_1], \ldots, [x_n, x_n]) = f(x_1, \ldots, x_n)$. The extension is not unique: for example both

$$F_1 = X^2 - 2X \quad \text{and} \quad F_2(X) = X(X - 2)$$

are extensions of $f(x) = x^2 - 2x$. Also, depending on which interval extension of a function one chooses, different bounds can be obtained for f over X. In this example, with $X = [0, 3]$, we have

$$F_1([0, 3]) = [0, 9] - [0, 6] = [-6, 9]$$

(i.e. $x \in [0, 3]$ implies $f(x) \in [-6, 9]$), whereas

$$F_2([0, 3]) = [0, 3].[-2, 1] = [-6, 3],$$

which yields a better bound for f over $[0, 3]$. Note that $f^R([0, 3]) = [-1, 3]$.

Different techniques can be used in order to obtain as tight bounds as possible, the aim being to find where is achieved a global minimum, x^+, of f over some specified domain:

1. A *reformulation* of the function (see the example above) can yield better bounds (another example of a reformulation is to use a Taylor expansion, for instance around the mid point of the box).

2. In the case where f is differentiable, one can use bounds on the gradient of f in order to eliminate subboxes. If we denote by $F'(X^k)$ a bound obtained on the gradient of f over some subbox X^k, and that we have $0 \notin F'(X^k)$, then we can eliminate this subbox (recall that from (2), a global minimum of the GFE cannot occur on the boundary of the feasible domain). We shall refer to this technique as a *gradient test*.

3. In the twice-differentiable case, a box can be eliminated if it can be shown that the Hessian is not positive semi-definite anywhere in the box (*Hessian test*). We use the computationally cheap condition for this of testing the diagonal Hessian elements. If any of these are negative through the box, the Hessian is nowhere positive semi-definite in the box so the box can be eliminated.

4. The *interval Newton step*. Given an initial box subdomain containing a global optimum, the interval version of Newton's method produces a smaller subdomain which also contains a global optimum, reducing all dimensions ultimately at a quadratic rate. This involves the solution of a system of equations with interval coefficients. The (real-valued) Hessian at the mid point of the box is found. This is used as a preconditioner for the interval Newton equations, and one sweep of *interval Gauss-Seidel* is then applied in an attempt to obtain tighter bounds [4].

5. *Interval subdivision* involves a recursive partition of the domain into subboxes over each of which bounds are computed. This being too costly in high-dimensional spaces, we integrate this into a B&B scheme. Any evaluation of f at a feasible point gives a lower bound for f over the original domain. In the next section, we shall refer to *mid-point evaluations* when we systematically evaluate f at the center point of each subbox obtained by subdivision. The smallest lower (respectively upper) bound obtained over all subboxes resulting from the subdivision process yields a lower (respectively upper) bounds for $f(x^+)$. The aim of the overall B&B process is to eliminate subboxes which can be shown not to contain a global minimum of f. In addition to the use of bounds, one can also apply techniques 1, 2, 3 and 4 above to eliminate a given subbox.

The reader is referred to [4] and [11] for a more complete description of global optimization using interval analysis.

We shall stop this global optimization process whenever either each remaining box has width less than ϵ_w, or a lower bound for the objective function is within ϵ_f of the upper bound (over the remaining boxes) for the objective function, where ϵ_w and ϵ_f are sufficiently small positive constants. Moreover, in the current GFE minimization application, we also stop whenever the mid point of a box is found that lies at a distance of ϵ_θ or more below the tangent plane [i.e whenever we find a point yielding a sufficiently negative objective function value for problem (7)], for some small positive constant ϵ_θ.

4. Computational Experiments

In this section we report numerical results obtained on two instances of the GFE global minimization problem without reaction (i.e. here $A = I$). The objective function we are using models the GFE for *ideal* vapour phases, and for liquid phases which can be described by the NRTL equation [7]. More precisely, the objective function has the following form:

$$G(K, t, x, y) = Z \sum_{k=1}^{K} y_k G^k_{t_k}(x), \tag{9}$$

where $1 \leq K \leq n_c$, $t_k \in \tau$, $k = 1, 2, \ldots, K$, τ is either $\{L\}$ (Problem 1) or $\{L, V\}$ (Problem 2), Z is defined by (4), y_k is the molar fraction of phase k, given by

$$y_k \equiv \frac{\sum_{i=1}^{n_c} z_{ik}}{Z}, \quad k = 1, 2, \ldots, K,$$

and x_{ik} is defined by (3). In the case of an instance involving only liquid phases ($\tau = \{L\}$), the function G^k_L is defined by

$$G^k_L = \sum_{i=1}^{n_c} x_{ik} \left(\sum_{j=1}^{n_c} \frac{x_{jk} m_{ij}}{\sum_{l=1}^{n_c} x_{lk} \mathcal{G}_{lj}} + \ln x_{ik} \right), \tag{10}$$

where m_{ij} and $\mathcal{G}_{ij}$ are given data. When both liquid and vapour phases are allowed ($\tau = \{L, V\}$), the functions G^k_L and G^k_V are defined as follows:

$$G^k_L = \sum_{i=1}^{n_c} x_{ik} \left(\sum_{j=1}^{n_c} \frac{x_{jk} m_{ij}}{\sum_{l=1}^{n_c} x_{lk} \mathcal{G}_{lj}} + \ln x_{ik} + \ln P^{\text{sat}}_i \right), \tag{11}$$

and

$$G^k_V = \sum_{i=1}^{n_c} x_{ik} (\ln x_{ik} + \ln P_i), \tag{12}$$

where m_{ij}, $\mathcal{G}_{ij}$, P^{sat}_i, and P_i are given data.

We shall describe below the behaviour of the our algorithm on three instances with data taken from [7]. The following equations are required to obtain the data involved in the formulation of the three problems:

$$\mathcal{G}_{ij} = \exp(-\alpha_{ij} t_{ij}), \tag{13}$$

$$m_{ij} = t_{ij} \mathcal{G}_{ij}, \tag{14}$$

$$\alpha_{ij} = \alpha_{ji}, \tag{15}$$

$$\alpha_{ii} = 0, \tag{16}$$

$$t_{ii} = 0, \tag{17}$$

$1 \leq i, j \leq n_c$.

The local optimization subroutine LO (referred to in the GILO algorithm) is implemented using the NAG E04UCF subroutine (sequential quadratic programming). The problem is formulated using non-degenerate linear constraints and the initial point for the local search is feasible. The subroutine follows a feasible path and the objective function is therefore non-increasing. The subroutine reports if a Kuhn-Tucker point is not reached in a specified number of iterations, and if this occurs it can be restarted with a steepest-edge step. Setting an upper limit on the number of iterations allowed between steepest-descent steps guarantees finite convergence to a Kuhn-Tucker point. In all the examples we have tried, rapid convergence occurs and so the return to steepest descent is not required.

The global optimization subroutine GI of the GILO algorithm will here be the implementation of the interval-analysis techniques described in section 3. This yields an algorithm which we call Interval-GILO. The specific interval-analysis techniques implemented, within the interval-subdivision B&B process, are the following. Firstly, we reformulated the objective function of the global optimization subproblem (7) (the tangent-plane distance function) using a second-order Taylor expansion around the mid point of each subbox, of width smaller than $\epsilon_{T_f} \equiv 0.07$, obtained during the course of the B&B process. We did the same using a first-order Taylor expansion for subboxes of width smaller than $\epsilon_{T_g} \equiv 0.05$ for the *gradient* of this objective function. We also obtained bounds on the objective function and on its gradient *without* using the Taylor expansion for subboxes of width larger than $\epsilon_{\bar{T}} \equiv 0.005$ (when both interval extensions—with and without Taylor expansion—are used, the best bound is used). Secondly, gradient tests were used in order to eliminate some subboxes. Use of the Hessian test did not improve the results. Thirdly, we used the interval version of the Newton step each time the algorithm encountered a subbox of width less than $\epsilon_N \equiv 0.05$. However, when it can be proved that a box does contain exactly one stationary point (e.g. when the interval Newton step reduces the box in *all* dimensions [4], or an ideal vapour phase is being tested for stability), we use a real (non-interval) local optimization. Mid-point evaluations were used in the first $N_m \equiv 100$ iterations. We used $\epsilon_w = \epsilon_f = \epsilon_\theta = 10^{-5}$ and $\epsilon_0 = 10^{-6}$. Breadth-first search is used in the B&B implementation. The same setting of the parameters are used in all our tests.

The global search takes place in the reduced space of the variables $x_{1k}, \ldots, x_{n_c-1,k}$. In order words, we deal with the equality constraints as follows. Bounds on the remaining variables $x_{n_c k}$ are obtained at each GI iteration from the intersection of its defined bounds, $[0, 1]$, with bounds obtained from the relations

$$x_{n_c k} \equiv 1 - \sum_{i=1}^{n_c - 1} x_{ik}, \quad k = 1, \ldots, K,$$

and the bounds on the reduced-space variables. If the resulting interval is empty, then the current box has no feasible points and can be deleted.

As interval methods are easily programmed in Fortran 90, our techniques were implemented with this new standard for Fortran. All CPU time figures reported

Table 1. Data for Problem 1

i	j	t_{ij}	α_{ij}	Z_i (mol)
1	2	3.00498	0.39195	0.5
2	1	4.69071	0.39195	0.5

Table 2. Solution for Problem 1

i	$t_1^+ = L$ z_{i1}^+	$t_2^+ = L$ z_{i2}^+	G^+
1	7.13609×10^{-4}	0.499286	-2.01983×10^{-2}
2	0.155879	0.344121	

in the next sections were obtained from runs on a Sun Sparc5 70MHz workstation. The number of local-optimization iterations we shall report refers to the number of iterations needed for the NAG subroutine to converge. The number of global-optimization iterations will correspond to the total number of times an interval has been divided in two in the B&B interval-analysis process.

4.1. Problem 1: LL Equilibrium

This problem corresponds to the Illustrative Example (Liquid-Liquid Equilibrium *n*-Butyl-acetate - Water) of [7].

For this problem, $n_c = 2$ and $\tau = \{L\}$. The necessary data for Problem 1 are obtained from Table 1 together with equations (13) to (17).

The program was started with the only feasible point corresponding to a single phase. The algorithm performed successively a global optimization step, a local optimization step, and a second global optimization step. The first global optimization step yielded the introduction of a second liquid phase, while the second global step simply proved that the iterate found by the local optimization step was (ϵ-)globally optimal. The global minimum, (K^+, t^+, z^+), found by the algorithm corresponds to two liquid phases, and it is given by Table 2, where $K^+ = 2$ and the z_{ik}^+'s are in moles. Table 3 reports numbers of iterations and CPU times required to obtain convergence. McDonald & Floudas [7] report 1.23 seconds of CPU time to converge to the same solution, with $\epsilon_f = 0.0005$. The final global step of the Interval-GILO ended with *both* stopping criteria ϵ_f and ϵ_w satisfied (i.e. the last iterate, say $\bar{z}$, is within distance ϵ_w of the global minimum, z^+, and the objective function evaluated at $\bar{z}$ is within ϵ_f of the global minimum function *value*), with $\epsilon_f = \epsilon_w = 10^{-5}$.

Table 3. Results for Problem 1

	# of iterations	CPU time (sec.)
1st global step	8	0.05
1st local step	9	0.07
2nd global step	46	0.16
entire program		0.38

Table 4. Data for Problem 2a

i	j	$g_{ij} - g_{jj}$	$g_{ji} - g_{ii}$	α_{ij}	i	P_i^{sat} (atm)	Z_i (mol)
1	2	3883.2	3849.57	0.24698	1	0.512337709	0.34483
1	3	998.2	65.74	0.88577	2	0.195255785	0.34843
2	3	1262.4	363.57	0.3565	3	0.485404650	0.31034

4.2. Problem 2: LLV and LV Equilibrium

This problem corresponds to Example 6 (Liquid-Liquid-Vapour and Liquid-Vapour Equilibrium for Benzene - Water - Acetonitrile) of [7]. For this problem, $n_c = 3$ and $\tau = \{L, V\}$.

If there is a vapour phase present in the current local minimum solution then the global step will not be looking for a vapour phase. This is because we assume the vapour phase to be ideal. That is to say, G_V^k is convex and hence one can show that no more than one phase of this type (vapour) needs to occur at any minimum of P. The program was started with the only feasible point corresponding to a single vapour phase.

The necessary data are obtained from Tables 4 and 5 together with equations (13) to (17) and:

$$t_{ij} = \frac{g_{ij} - g_{jj}}{RT},$$

$R = 1.9872$ cal/K/mol. For Problem 2a (Example 6A of [7]), $T = 333$ K, and $P_i = 0.769$ atm, $i = 1, 2, 3$. For Problem 2b (Example 6C of [7]), $T = 300$ K, and $P_i = 0.1$ atm, $i = 1, 2, 3$.

For Problem 2a (Liquid-Liquid-Vapour Equilibrium), the algorithm performed successively a global optimization step, a local optimization step, a second global step, a second local step, and a third global step. Again, each global step introduced a new liquid phase, except for the last global step which simply established the global optimality of the local minimum found by the second local step. The global minimum, (K^+, t^+, z^+), found by the algorithm corresponds to one vapour phase and two liquid phases. It is given by Table 6, where $K^+ = 3$ and the z_{ik}^+'s are in

Table 5. Data for Problem 2b

i	j	$g_{ij} - g_{jj}$	$g_{ji} - g_{ii}$	α_{ij}	i	P_i^{sat} (atm)	Z_i (mol)
1	2	3892.44	3952.2	0.23906	1	0.136305578	0.34483
1	3	693.61	92.47	0.67094	2	0.034639285	0.34843
2	3	1016.28	415.38	0.20202	3	0.123950154	0.31034

Table 6. Solution for Problem 2a

i	$t_1^+ = V$ z_{i1}^+	$t_2^+ = L$ z_{i2}^+	$t_3^+ = L$ z_{i3}^+	G^+
1	0.104643	7.31183×10^{-4}	0.239456	-1.40852
2	5.24255×10^{-2}	0.262353	3.36516×10^{-2}	
3	6.16355×10^{-2}	2.16922×10^{-2}	0.227012	

Table 7. Results for Problem 2a

	# of iterations	CPU time (sec.)
1st global step	2	0.04
1st local step	10	0.08
2nd global step	2	0.04
2nd local step	33	0.22
3rd global step	436	3.11
entire program		3.65

moles. Table 7 displays numbers of iterations and CPU times required to obtain convergence.

For Problem 2b (Liquid-Vapour Equilibrium), the algorithm performed successively a global optimization step, a local optimization step, and a second global step. The first global step introduced a new liquid phase, while the second global step simply established the global optimality of the local minimum found by the local step. The global minimum, (K^+, t^+, z^+), found by the algorithm corresponds to one vapour phase and one liquid phase. It is given by Table 8, where $K^+ = 2$ and the z_{ik}^+'s are in moles. Table 9 displays numbers of iterations and CPU times required to obtain convergence.

For both Problem 2a and Problem 2b, the final global step of the Interval-GILO ended with *both* stopping criteria ϵ_f and ϵ_w satisfied, with $\epsilon_f = \epsilon_w = 10^{-5}$. These results, obtained on a Sun Sparc5 70MHz workstation, again compare favourably with those obtained by McDonald & Floudas' GOP algorithm, for which they report, in [7], 766 and 118 seconds of CPU time respectively for Problem 2a and

Table 8. Solution for Problem 2b

i	$t_1^+ = V$ z_{i1}^+	$t_2^+ = L$ z_{i2}^+	G^+
1	0.344827	3.23173×10^{-6}	
2	0.332572	1.58576×10^{-2}	-3.41234
3	0.309850	4.89759×10^{-4}	

Table 9. Results for Problem 2b

	# of iterations	CPU time (sec.)
1st global step	269	0.99
1st local step	11	0.10
2nd global step	297	1.42
entire program		2.60

Problem 2b, on a Hewlett Packard 9000/730 machine to converge to the same solution, with relative convergence tolerances, ϵ_f, of 0.001 and 0.0001 respectively. When McDonald & Floudas assume that it is known that there is one vapour phase and one liquid phase at the global optimum of Problem 2b, they report that their algorithm then requires 0.88 seconds of CPU time to converge. Note also that in a recently submitted paper [6], they report a CPU time of 0.76 seconds to *verify* the global optimality of the global minimum of Problem 2a.

Finally, notice that our algorithm does not rely on having a particular functional form for the thermodynamic model of the GFE function. Preliminary computational experiments on Example 1 of [6], a problem modelled using the UNIFAC equation, yields a CPU time comparable to that reported in [6].

5. Conclusions

This paper describes the implementation of the finite generic algorithm (the GILO algorithm) for phase and chemical equilibria with multiple phase models presented by McKinnon & Mongeau in [8]. The contribution of the current paper is the implementation of interval-analysis techniques in order to solve the global optimization subproblem involved in the GILO algorithm: this yielded the Interval-GILO algorithm. We obtained good numerical results on the three instances of the GFE minimization problem on which the Interval-GILO algorithm was tested.

In successive global minimization steps the objective functions are the same except for an affine term. It would therefore be possible to save some of the calculations from one global minimization to the next and this would have a significant benefit for problems such as Problem 2b in which not only the last global step is expensive.

In this work, we made use of an unmodified NAG subroutine in order to solve the local optimization subproblem involved in the GILO algorithm of [8]. This worked reliably and in all cases found a local minimum without being restarted. However it would be possible to make this numerically difficult step much more efficient by using techniques such as those proposed by Trangenstein in [13] for phase equilibrium problems involving two phases. Trangenstein adapts a modified Newton's method so as taking advantage of the particular structure of the phase equilibrium problem.

Although it is known that every component is present in every phase, there are cases where the amounts of some components in some phases are so low that it is numerically simpler to treat them as being absent. This is the case for example in the modeling of petroleum reservoir fluid flow where one often wishes to consider one phase which contains only water and at least one phase which cannot contain water. Further work is needed to address this issue by extending the algorithm to allow for the use of thermodynamic models in which some components are absent from some of the phases.

The generic GILO algorithm can be used with any global optimization method and it would be interesting to test the global stability test of McDonald & Floudas' GLOPEQ package with it. The Interval-GILO algorithm has the great advantage of generality: it does not rely on having a particular functional form for the thermodynamic model of the GFE function. All that is needed is an algorithm to evaluate the function values. Optionally, algorithms for evaluating higher-order derivatives can also be used to improve efficiency. The NRTL model is a severe test for interval methods as it contains an entropy term which is often almost exactly cancelled by a molecular interaction term. This is a situation where it is difficult to obtain good bounds from interval methods. However, the use of Taylor series improves the bounds dramatically and produces good results.

References

1. L. Baker, A. Pierce, and K. Luks. Gibbs energy analysis of phase equilibria. *Society of Petroleum Engineers Journal*, 22:731–742, 1982.
2. J. Gibbs. Graphical methods in thermodynamics of fluids. *Trans. Connecticut Acad*, 2:311, 1873.
3. J. Gibbs. A method of geometrical representation of the thermodynamic properties of substances by means of surfaces. *Trans. Connecticut Acad*, 2:382, 1873.
4. E. Hansen. *Global Optimization Using Interval Analysis*, volume 165 of *Pure and Applied Mathematics*. Marcel Dekker, New York, 1992.
5. Y. Jiang, W. Smith, and G. Chapman. Global optimality conditions and their geometric interpretation for the chemical and phase equilibrium problem. *SIAM Journal on Optimization*, 1995. To appear.
6. C. McDonald and C. Floudas. GLOPEQ: A new computational tool for the phase and chemical equilibrium problem. 1994. Submitted.
7. C. McDonald and C. Floudas. Global optimization for the phase and chemical equilibrium problem: Application to the NRTL equation. *Computers & Chemical Engineering*, 1995. To appear.

8. K. I. M. McKinnon and M. Mongeau. A generic global optimization algorithm for the chemical and phase equilibrium. Technical Report MS94/1, Dept. of Mathematics and Statistics, University of Edinburgh, U.K., 1994.
9. M. L. Michelsen. The isothermal flash problem. Part I. Stability. *Fluid Phase Equilibria*, 9:1–19, 1982.
10. M. L. Michelsen. The isothermal flash problem. Part II. Phase-split calculation. *Fluid Phase Equilibria*, 9:21–40, 1982.
11. H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization*. Wiley, New York, 1982.
12. W. Smith and R. Missen. *Chemical Reaction Equilibrium Analysis: Theory and Algorithms*. Wiley & Sons, 1982.
13. J. A. Trangenstein. Customized minimization techniques for phase equilibrium computations in reservoir simulation. *Chemical Engineering Science*, 42(12):2847–2863, 1987.