Research Article

Interval Arithmetic for Nonlinear Problem Solving

Benito A. Stradi-Granados

Department of Materials Science and Engineering, Institute of Technology of Costa Rica, Cartago 07050, Costa Rica

Correspondence should be addressed to Benito A. Stradi-Granados; bstradi@itcr.ac.cr

Received 10 February 2013; Accepted 19 May 2013

Academic Editor: A. Nazli Gundes

Copyright © 2013 Benito A. Stradi-Granados. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Implementation of interval arithmetic in complex problems has been hampered by the tedious programming exercise needed to develop a particular implementation. In order to improve productivity, the use of interval mathematics is demonstrated using the computing platform INTLAB that allows for the development of interval-arithmetic-based programs more efficiently than with previous interval-arithmetic libraries. An interval-Newton Generalized-Bisection (IN/GB) method is developed in this platform and applied to determine the solutions of selected nonlinear problems. Cases 1 and 2 demonstrate the effectiveness of the implementation applied to traditional polynomial problems. Case 3 demonstrates the robustness of the implementation in the case of multiple specific volume solutions. Case 4 exemplifies the robustness and effectiveness of the implementation in the determination of multiple critical points for a mixture of methane and hydrogen sulfide. The examples demonstrate the effectiveness of the method by finding all existing roots with mathematical certainty.

1. Introduction

There are a large number of problems that require the computation of stationary points and roots of equations. These are found in the fields of optimization [1], economics and finance [2, 3], thermodynamics [4, 5], applied mathematics [6], and similar contributions over the last thirty years. The application of interval arithmetic involves from algebraic equations with known solutions to more complicated systems representing physical phenomena. Among the earlier problems, second- and third-order polynomial problems serve to illustrate the effectiveness of the implementation. Similarly, more complicated multidimensional problems serve to illustrate stability and robustness of the implementation. In particular, the determination of critical points is of interest. Critical point computations are a well-known highly nonlinear problem that has been studied for a long time. The literature is endowed with some worthy analyses of the problem. Michelsen and Heidemann [7] discuss the calculation of critical points from cubic two-constant equations of state. They numerically determine the critical points of mixtures solving

the highly nonlinear critical point equations. Hoteit et al. [8] claim an efficient algorithm based on bisection, secant, and inverse quadratic programming methods where their method is apparently faster but incapable of handling the presence of multiple critical points without restarting the program after each critical point determination. Nichita and Gomez [9] discuss the use of the tunneling global optimization method to determine all the of global minima where they concentrate on the determination of the critical points of mixtures. They use temperature and molar volume as primary variables. Their method lacks the strength of being able to stop when all critical points have been found, and this follows from the fact that at the beginning of the process the number of critical points to be determined needs to be specified. Michelsen [10] develops some earlier ideas that deal with the computation of phase envelopes with use of finite differences. Michelsen [11] discusses the isothermal flash separation and critical point computations with the use of tangent plane analysis. Michelsen and Kistenmacher [12] describe the disadvantage of using composition-dependent binary interaction coefficients in computations. Michelsen and Heidemann [13]

present the calculation of tricritical points using tangent-plane distance analysis with some extra algebra in the mathematical development. The results are generated using the Peng-Robinson and Soave-Redlich-Kwong (SRK) equations of state. Michelsen [14] illustrates the determination of the critical points and phase boundaries using the real-arithmetic Newton-Raphson method with temperature and pressure as variables; in his analysis rather close initial estimates to the solution are needed for convergence.

There are also other more general contributions such as that of Cai et al. [15] who developed expressions for the spinodal criterion, critical criterion, and various stability tests for systems containing one discrete component and one polydisperse polymer. Lindvig et al. [16] propose the EFV (Entropic-FV) model for predicting the miscibility behavior of paints. Cismonti and Michelsen [17] use a Newton procedure for the calculation of critical lines, critical endpoints, and three-phase lines for binary mixtures. Carstensen and Petković [18] propose the use of a hybrid method combining normal (Nourein's method) and interval arithmetic to arrive to the solution of polynomial equations efficiently. Maranas and Floudas [19] propose an approach based on convex lower bounding and domain partitioning to achieve convergence to all solutions within a domain. Stradi et al. [4] discuss the computation of critical points using the INTBIS library [20].

This paper presents the solution of nonlinear problems using interval arithmetic in INTLAB (INTerval LABoratory) [21] by using the interval-Newton Generalized-Bisection method (IN/GB). The advantage of this approach is that INTLAB is developed over MATLAB [22], a well-known and powerful computational package widely used in mathematics and engineering applications [23–25]. Although, with the cost of interpretation overhead [26], INTLAB facilitates interval computations by using the MATLAB user interface and debugging tools, this makes production times smaller and results generation faster [27]. Section 2 of the paper provides a description of the problems to be solved in the paper, Section 3 proposes the methodology to solve the problems using the IN/GB method, Section 4 presents the results and discussion comparing the IN/GB method with the simple bisection method, and Section 5 highlights the conclusions derived from this study.

2. Problem Description

This section starts with two examples to demonstrate the use and performance of the IN/GB method. The first deals with two second-order equations where the solution is to be found over a relatively large domain. The second illustrates a particularly important case of a repeated root that causes the first derivative to be zero, and consequently it would traditionally create problems during the process of solution for the real-arithmetic Newton-Raphson method [23]. The third and fourth examples are related to the behavior of fluids in the two phase and critical regions.

Consider the general case where all of the roots to the nonlinear problem $f(x) = 0$ have to be found within a given domain in the absence of an initial guess.

Case 1. The first problem is a set of two second-order equations as follows:

$$\begin{aligned}(xy)^2 + 4x - 4 &= 0, \\ y^2 + 4xy - 4 &= 0.\end{aligned}\tag{1}$$

The domain for the search is the same for both variables: $x \in [-100, 100]$, $y \in [-100, 100]$. Three solutions are found for these two equations with mathematical certainty in a single run of the IN/GB method implemented over the INTLAB program.

Case 2. The second problem is a third-order polynomial with three real roots, where one of the roots is repeated:

$$x^3 - 5x^2 + 7x - 3 = 0.\tag{2}$$

This is an important example because the derivative of the function is zero at the repeated root. The traditional real-arithmetic Newton-Raphson method [23] would fail if we needed to apply the method near the root where the derivative is close to zero. The domain of the search is $x \in [-100, 100]$.

The interval Newton is expected to have problems because of its reliance on the interval derivative of the function to find a solution, but the combination with the bisection method should solve the problem.

Case 3. The third problem deals with the determination of volume solutions for the Peng-Robinson equation of state [28] for carbon dioxide at a given temperature and pressure, which is given by the following expression that related pressure, temperature, and volume:

$$P - \frac{RT}{(v-b)} + \frac{a}{(v * (v+b) + b * (v-b))} = 0.\tag{3}$$

This can also be written in the form

$$\begin{aligned}P(v-b)(v * (v+b) + b * (v-b)) \\ - RT(v * (v+b) + b * (v-b)) + a * (v-b) &= 0,\end{aligned}$$

$$\begin{aligned}a &= \frac{0.45724R^2T_C^2}{P_C(1 + (0.3764 + 1.54226w - 0.2699w^2)(1 - T_r^{1/2}))^2}, \\ b &= \frac{0.07780RT_C}{P_C}.\end{aligned}\tag{4}$$

The necessary conditions for this problem are the following:

$$\begin{aligned}T_C &= 304.21 \text{ K}, \\ P_C &= 73.83 \text{ bar}, \\ w &= 0.224, \\ T &= 301 \text{ K}, \\ P &= 68.18 * 10^5 \text{ Pa}, \\ R &= 83.14 \text{ (Pa cm}^3\text{/gmol K)},\end{aligned}$$

and the interval for the volume search is given by $v \in [1.1b, 16b]$.

This is a classic chemical engineering problem where, for a given equation of state, there may be more than one molar volume that satisfies the equation of state at a given temperature and pressure. In particular, if the conditions are such that two phases coexist, like in this case, then there will be three solutions. The traditional interpretation is that the solutions represent the liquid molar volume (smallest volume solution) and the vapor molar volume (largest volume solution). The middle volume solution is an unstable solution, and thus not observable.

Case 4. This is a highly nonlinear problem subject to analysis by different research groups for a number of years [4, 7, 14, 17]. In this case, there is a mixture of methane and hydrogen sulfide where the problem is to determine all of the critical points for a given composition and domain ranges for temperature and molar volume using the Redlich-Kwong equation of state [28]. The problem was solved in a previous development with the use of the INTBIS library and some extensive programming.

The domains of the variables under study are indicated as follows:

$$\begin{aligned} T &\in [180, 600]K, \\ v &\in [1.1, 4]b, \\ \Delta x_1 &\in [0, 1], \\ \Delta x_2 &\in [-1, 1]. \end{aligned}$$

The criticality conditions are written as follows:

$$\begin{aligned} &\sum_{j=1}^C A_{ij} \Delta n_j \\ &= \frac{RT}{n} \left(\frac{\Delta n_i}{y_i} + F_1 (\beta_i \bar{N} + \bar{\beta}) + \beta_i F_1^2 \bar{\beta} \right) \\ &+ \frac{a}{bn} \left(\beta_i \bar{\beta} F_3 - \frac{F_5}{a} \sum_{j=1}^C a_{ij} \Delta n_j + F_6 (\beta_i \bar{\beta} - \alpha_i \bar{\beta} - \bar{\alpha} \beta_i) \right), \\ &i = 1, \dots, C, \end{aligned} \quad (5)$$

$$\begin{aligned} &\sum_{i=1}^C \sum_{j=1}^C \sum_{k=1}^C A_{ijk} \Delta n_i \Delta n_j \Delta n_k \\ &= \frac{RT}{n^2} \left(-\sum_{i=1}^C \frac{\Delta n_i^3}{y_i^2} + 3\bar{N}(\bar{\beta} F_1)^2 + 2(\bar{\beta} F_1)^3 \right) \\ &+ \frac{a}{n^2 b} \left(3\bar{\beta}^2 (2\bar{\alpha} - \bar{\beta}) (F_3 + F_6) - 2\bar{\beta}^3 F_4 - 3\bar{\beta} \bar{\alpha} F_6 \right). \end{aligned} \quad (6)$$

The mass balance for a mixture of constant composition is given by

$$\sum_{i=1}^C \Delta n_i^2 - 1 = 0. \quad (7)$$

Meanings are given in the appendix.

3. Methodology

We employed the interval mathematics platform INTLAB to implement the interval-Newton Generalized-Bisection (IN/GB) method for the solution of nonlinear problems. The advantage of this procedure is that INTLAB is user friendly and runs over MATLAB; this makes INTLAB a very promising platform by putting the tools of MATLAB at the disposal of the INTLAB programmer. In the past, one of the major problems with the use of interval mathematics was the use of libraries that required extensive programming and lacked utilities for fast debugging. As a result, large time investments needed to be made in programming and debugging prior to the implementation and generation of first results, a situation that obviously derives in making interval arithmetic less attractive to a larger scientific audience. However, with INTLAB, the situation is different: program sizes are much smaller, debugging is more efficient, and results are generated more rapidly.

3.1. The Interval-Newton Generalized-Bisection Method. An interval-Newton Generalized-Bisection method (IN/GB) is presented to demonstrate the use of the INTLAB platform for nonlinear problems of different levels of complexity.

The interval IN/GB method is a powerful computational approach that allows for the computation of all of the roots of a nonlinear problem without the need for initial guesses for the variables [1, 29, 30].

The IN/GB method requires only that we specify the domain of the variables of interest. The program tests a sequence of enclosures determined through the IN/GB method. If there is a root, then the IN/GB method determines a very thin slice in which the root lies. On the other hand, if there is no root, then the algorithm also determines with mathematical certainty the absence of solutions.

The most important component part of the IN/GB method is the Newton method in interval arithmetic. This is written as follows:

$$J(X)(N - x) = -f(x). \quad (8)$$

This equation is of the form $AX = b$, and consequently suitable for solution with simultaneous linear equations solvers.

In this equation, capital letters are intervals and small case letters are real numbers. $J(X)$ is the interval extension of the Jacobian matrix, which is determined by evaluating the Jacobian matrix, J , using the interval domain X , rather than the real variable x , where x is generally the midpoint of X . This system of equations is solved using the Gauss-Seidel method [1], or a similar algorithm to determine N , where N is the image X generated by the application of the IN/GB method. If N and X intersect, then the process is repeated, noting that in successive applications it is the intersection of N and X ($N \cap X$) that serves as initial interval. The process continues until a sufficiently thin interval around the solution is derived. Other implementations, when sufficiently closed to the solution, switch to the real-arithmetic Newton-Raphson method to accelerate convergence [20].

TABLE 1: Interval-Newton Generalized-Bisection method results and actions.

Operation	Result	Comment	Action
$N \cap X$	I	I smaller in size	Continue with interval Newton
$N \cap X$	X , other	No reduction in the original domain size	Bisect

TABLE 2: Results for Case 1.

Domain	Equations	Roots	Subset	Method
[-100, 100]	$(x_1x_2)^2 + 4x_1 - 4 = 0$	$x_1 = -0.68443924$	647	N
		$x_2 = 3.79247516$	809	B
	$x_2^2 + 4x_1x_2 - 4 = 0$	$x_1 = 0.465822965$	654	N
		$x_2 = -3.13799232$	814	B
		$x_1 = 0.84600538$	677	N
		$x_2 = 0.92770308$	819	B
			Total	
			685	N
		820	B	

N: IN/GB method, B: Bisection method.

If N is a subset of X , then there is a single root in the domain of interest [29]. Similarly, if N does not intersect X , then there is no root in the domain, and a new subdomain is tested for the presence of roots.

It is important to indicate at this point that there are several problems not generally mentioned in the literature that may occur when using the IN/GB method. If the interval X is too large, then the image N will contain and be larger than the original interval X , generating the problem where the image is larger than the original interval and no convergence is achieved with the interval-Newton portion of the method alone. The other problem is where the image N is equal to X ; in this case the search will not advance, and no solution will be achieved within the permissible execution time. In both cases, the action taken is to proceed with the partition of the box, as shown in Table 1, and restart the search for each subdomain generated. The partition is done with the Bisection method [23] over each dimension of the domain followed by a range test applied to all subdomains generated.

Application of the Bisection method generates two intervals from each interval that is bisected. Consequently, in order to explore all of the possibilities, the combinations of these intervals are needed to determine the existence and values of all possible solutions.

The existence of a root is ascertained by evaluating the original function over the interval, or subdomain, of interest, using interval arithmetic [30]. If the resulting interval contains zero, $0 \in f(X)$, then there may exist at least one root in that domain. This process of testing for the existence of a root in the domain of interest is called *range test*. If the evaluation of the function over the interval of interest, X , results in an interval image that does not contain zero, then there is a mathematical guarantee that there is no root within the interval X .

For example, if three intervals are bisected, then six subintervals are generated, and eight combinations of them

would need to be tested first in the *range test* to ascertain whether there may be a root in the subdomain, and second in the IN/GB method search for the solution.

Generally, multiple applications of the Bisection method, part of the IN/GB method, are needed when large variable domains are used. This situation is particularly true when zeros are contained in diagonal elements of the interval Jacobian expression, J . The presence of these zeros generates interval images, N , in the IN/GB method with infinite spans for some or all variables. Consequently, bisection is needed to divide and eliminate those subdomains where infinite spans are generated. These computational problems are managed efficiently in INTLAB. In INTLAB, the programming environment allows for processing warnings identified by the flags infinite (inf) and not-a-number (NaN) without an error that would stop the program execution. The flag inf would identify quantities that exceed real number representation such as e^{1000} and also results generated by the division of a number by zero: 1.0/0.0. The flag NaN would identify quantities that are mathematically undefined such as 0.0/0.0 or intervals that do not intersect. Consequently, if these flags occur, then bisection is applied because they are indicative of problems with the interval-Newton portion of the algorithm. This capacity to handle numerically undefined quantities saves extensive programming time and debugging effort invested on the solution of a particular problem.

The search for a solution can only end with one of two outcomes, either a root is found within an interval or no root is found with mathematical certainty. The main steps of the algorithm are represented in Figure 8.

4. Results and Discussion

4.1. Case 1. Table 2 presents the results obtained using the IN/GB and the Bisection methods applied to Case 1. The IN/GB method determined all of the roots and searched the

FIGURE 1: Evolution of the relative error (y-axis) with the subinterval number for the Bisection method for Case 1.

entire domain with 685 subdivisions. The Bisection method with interval arithmetic was applied, and convergence to all roots was achieved with 820 subdivisions of the original interval. It is important to mention that, before applying either the IN/GB method or the Bisection method, a range test is applied to the subdomain under consideration to determine whether a root may be found within the subdomain. Three solutions are found for these two equations. The algorithm developed found three solutions without the need for initial guesses over a fairly large domain. This case is a baseline case where the solutions can be calculated also with the real-arithmetic Newton method, but then at least three different initial guesses would need to be provided.

Figure 1 shows the evolution of the error with the subinterval number for the Bisection method; Figure 2 shows the evolution of the error with the subinterval number for the IN/GB method. The error limit is represented with a horizontal line at 1×10^{-6} , and the trends are approximated with a third-order polynomial in a semilog plot for the Bisection and the IN/GB methods. The interval-Newton part of the IN/GB method may generate some very large errors particularly when the derivative, or the Jacobian matrix, contains a zero in the diagonal-element intervals. Consequently, in order to keep the scale to a legible size, large errors described as inf and nondefined quantities described as NaN appear with an error value of 10 on the graph's y-axis. For both the bisection method and the IN/GB method, the solutions are found toward the end of the search of subdomains.

4.2. Case 2. Table 3 presents the results for Case 2, where the IN/GB method determines all of the roots to this third-order equation using 17723 subdivisions to complete the search over the entire domain. The bisection method requires 17981 subdivisions to search the entire domain and find all of the roots. It is important to notice that the first root is found much faster by the IN/GB method by a factor of more than a hundred times over the bisection method. However, the convergence speed is smaller at the second

FIGURE 2: Evolution of the relative error (y-axis) with the subinterval number for the IN/GB method for Case 1.

FIGURE 3: Evolution of the relative error (y-axis) with the subinterval number for the Bisection method for Case 2.

root where both algorithms need closely the same number of subdivisions. For this simple example, the roots are evident and can be determined by inspection. However, if using the real-arithmetic Newton-Raphson method, there would be no immediate way to determine that one of the roots is repeated, and this situation would require additional time to test additional initial guesses or execute a polynomial deflation procedure with the roots already found.

Figure 3 shows the evolution of the error with the subinterval number for the Bisection method. Figure 4 shows the evolution of the error with the subinterval number for the IN/GB method.

In the case of the Bisection method, it is particularly different the convergence profile, compared to Case 1; both cases have been approximated with a third-order polynomial determined with a least-squares fit. The IN/GB method presents some very large errors at the beginning of the search, and a log-log plot is used to better depict this behavior. Large error values are constrained to a value of 10 for graphical

FIGURE 4: Evolution of the relative error (y-axis) with the subinterval number for the IN/GB method for Case 2.

FIGURE 5: Evolution of the relative error (y-axis) with the subinterval number for the Bisection and IN/GB methods for Case 3.

clarity, and they are indicative of inf and NaN results; in these cases, the algorithm switches from the interval-Newton to the Bisection method. In this case, the roots are found by IN/GB method both at the beginning and at the end of the process of testing all generated subintervals as indicated by the values below $1 * 10^{-6}$.

4.3. Case 3. Table 4 presents the results for Case 3. This is a particularly interesting example because it exposes some of the difficulties in interval computations. In this example, the interval-Newton method does not result in an interval, N , with the desired characteristics: (1) that N be of smaller size than the original interval, X , and (2) that N and X intersect. During the search process with the interval-Newton, this portion of the IN/GB method generates NaN or inf messages, and the program switches to the Bisection method. This point is very important because it clearly demonstrates that if for some reason the interval-Newton portion of the algorithm is incapable of dealing with the problem then the other

FIGURE 6: Evolution of the relative error (y-axis) with the subinterval number for the IN/GB method for Case 4.

FIGURE 7: Evolution of the relative error (y-axis) with the subinterval number for the Bisection method for Case 4.

half of the algorithm (i.e., the Bisection method) takes over and provides a solution, making the IN/GB a very efficient method.

Once an interval is bisected, a range test on each subdomain created is performed, and if a root is detected (i.e., $0 \in f(X)$), then that interval is stacked for further processing. This process will continue until either each subdomain is eliminated for not containing zero (i.e., $0 \notin f(X)$) or a root is found within the subdomain with a prescribed error tolerance. In this example, the IN/GB method finds the solutions and scans the entire domain by dividing the domain in 4332 subsets. Consequently, the graph in Figure 5 for the Bisection and the IN/GB methods is the same.

4.4. Case 4. Table 5 presents the results for Case 4. This is a more elaborate example because for a multicomponent mixture there is no procedure to determine *a priori* whether

TABLE 3: Results for Case 2.

Domain	Equation	Roots	Subset	Method
[-100, 100]	$x^3 - 5x^2 + 7x - 3 = 0$	$x = 3.00000000$	46	N
		$x = 2.99997777$	9121	B
		$x = 0.99807270$	12543	N
		$x = 0.99807270$	12801	B
		(double root)		
			Total	
			17723	N
			17981	B

TABLE 4: Results for Case 3.

Domain	Equations	Roots	Subset	Method
[29.31, 426.43]	$\alpha = (\nu * (\nu + b) + b * (\nu - b))$	$\nu = 100.0632$	4031	N
	$\beta = (\nu - b)$	$\nu = 100.0632$	4031	B
	$P * \alpha * \beta - RT * \alpha + a * \beta = 0$	$\nu = 160.6770$	4181	N
	$T = 301K$	$\nu = 160.6770$	4181	B
	$P = 68.18 \text{ bar}$	$\nu = 79.6437$	4238	N
	$b = 26.6520264 \text{ cm}^3/\text{gmol}$	$\nu = 79.6437$	4238	B
			Total	
			4332	N
			4332	B

TABLE 5: Results for Case 4.

Domain	Equations	Roots	Subset	Method
[0.30, 1] T	Redlich-Kwong-Soave equation of state	$T = 0.48143566$	168781	N
[0.275, 1] ν	Mixture of methane and hydrogen sulfide	$\nu = 0.56426799$	829620	B
[0, 1] Δn_1	Find all of the critical points in the domain	$\Delta n_1 = 0.08460040$		
[-1, 1] Δn_2	T scaled by 600K	$\Delta n_2 = -0.99641496$		
$x_1 = 0.49 \text{ CH}_4$	ν scaled by 4b			
$x_2 = 0.51 \text{ H}_2\text{S}$				
		$T = 0.34691664$	237185	N
		$\nu = 0.29124484$	824978	B
		$\Delta n_1 = 0.6814699$		
		$\Delta n_2 = -0.73189541$		
		$T = 0.38612396$	285239	N
		$\nu = 0.36953418$	779696	B
		$\Delta n_1 = 0.52293204$		
		$\Delta n_2 = -0.85237438$		
			Total	
			345733	N
			856970	B

FIGURE 8

there will be none, one, or more than one critical points. The IN/GB method is much faster (Figure 6) than the Bisection method (Figure 7), and the order for finding the solutions differs between the methods. The IN/GB finds the solutions and scans the entire domain by dividing the domain in 345733 subsets, while the Bisection method finds the solutions and scans the entire domain by dividing the domain in 865970 subsets.

It is illustrative to plot the error in the IN/GB method versus the number of subdivisions of the initial domain. There are some very large errors, particularly at the beginning, which are cut to a value of 10 units, in order to preserve a reasonable scaling in the figures. As it was mentioned previously, for those cases with very large errors (inf) or not-a-number warnings (NaN), the algorithm proceeded to

bisecting those domains in order to continue with the search. The relative error used to determine solutions was of 1×10^{-6} , which is indicated in the graphs with a flat line.

The Bisection method converges linearly in a semilog plot toward the solutions which are found at the end of the process, as it is expected. However, the IN/GB finds solutions prior to the end of the process and with fewer subdivisions of the original interval.

5. Conclusions

This paper has introduced the use of INTLAB in computations for the determination of several solutions without the need for initial guesses and with a single run of the algorithm. The IN/GB method was implemented in INTLAB with four examples. The first two examples were algebraic equations that serve as a reference to verify the effectiveness of the algorithm and describe some important findings of the IN/GB method. The third and fourth cases are related to the computation of multiple solutions that represent properties of substances and fluids. Convergence to a solution is faster with the IN/GB method in general, but there are exceptions to this rule particularly when a singular Jacobian matrix may occur.

Appendix

$$\sum_{j=1}^C A_{ij} \Delta n_j = \frac{RT}{n} \left(\frac{\Delta n_i}{y_i} + F_1 (\beta_i \bar{N} + \bar{\beta}) + \beta_i F_1^2 \bar{\beta} \right) + \frac{a}{bn} \left(\beta_i \bar{\beta} F_3 - \frac{F_5}{a} \sum_{j=1}^C a_{ij} \Delta n_j + F_6 (\beta_i \bar{\beta} - \alpha_i \bar{\beta} - \bar{\alpha} \beta_i) \right),$$

$$i = 1, \dots, C,$$

$$\sum_{i=1}^C \sum_{j=1}^C \sum_{k=1}^C A_{ijk} \Delta n_i \Delta n_j \Delta n_k = \frac{RT}{n^2} \left(- \sum_{i=1}^C \left(\frac{\Delta n_i^3}{y_i^2} \right) + 3 \bar{N} (\bar{\beta} F_1)^2 + 2 (\bar{\beta} F_1)^3 \right) + \frac{a}{n^2 b} \left(3 \bar{\beta}^2 (2 \bar{\alpha} - \bar{\beta}) (F_3 + F_6) - 2 \bar{\beta}^3 F_4 - 3 \bar{\beta} \bar{\alpha} F_6 \right),$$

$$(A.1)$$

where the following are considered.

C is the total number of components.

A_{ij} is the second-order derivative of the Helmholtz free energy with respect to the number of moles of species i and j .

Δn_j is the change in total number of moles of species j . This is used in the context of a Taylor series expansion of the Helmholtz free energy in terms of composition.

R is the ideal gas constant.

T is the absolute temperature.

n is the total number of moles.

n_i is the number of moles of species i , analogous meaning for other subindices.

$\bar{N}$ is the parameter used in the equations for the computation of critical points; it is defined as

$$\bar{N} = \sum_{i=1}^C \Delta n_i. \quad (\text{A.2})$$

y_i is the mole fraction of component i , analogous meaning for other subindices.

a is the average energy parameter of the equation of state for the mixture of components. This is computed using standard van der Waals mixing rules:

$$a = \sum_{i=1}^C \sum_{j=1}^C \frac{n_i n_j}{n^2} a_{ij}, \quad (\text{A.3})$$

$$a_{ij} = (a_i a_j)^{0.5} (1 - k_{ij}).$$

a_{ij} is the energy of interaction parameter between species i and j .

a_i is the energy parameter of species i . The meaning is the same for other subindices. It is computed using the following correlation:

$$a_i = \left(\frac{(RT_{c_i})^2 \eta}{P_{c_i}} \right) \left[1 + c_i \left(1 - \left(\frac{T}{T_{c_i}} \right)^{0.5} \right) \right],$$

$$\eta = 0.42748$$

for the Redlich-Kwong-Soave equation of state,

$$\eta = 0.45724$$

for the Peng-Robinson equation of state,

$$c_i = 0.48 + 1.574w_i - 0.176w_i^2$$

for the Redlich-Kwong-Soave equation of state,

$$c_i = 0.37464 + 1.54226w_i - 0.26992w_i^2$$

for the Peng-Robinson equation of state.

w_i is the acentric factor of species i .

T_{c_i} is the critical temperature of species i .

P_{c_i} is the critical pressure of species i .

w_i is the acentric factor of species i .

k_{ij} is the binary interaction parameter between species i and j .

α_k is the parameter used in the equations for the computation of critical points; it is defined as

$$\alpha_k = \frac{\sum_{i=1}^C y_i a_{ik}}{a}. \quad (\text{A.5})$$

$\bar{\alpha}$ is the parameter used in the equations for the computation of critical points; it is defined as

$$\bar{\alpha} = \sum_{i=1}^C \Delta n_i \alpha_i. \quad (\text{A.6})$$

$\bar{a}$ is the parameter used in the equations for the computation of critical points; it is defined as

$$\bar{a} = \frac{1}{a} \sum_{i=1}^C \sum_{j=1}^C \Delta n_i \Delta n_j a_{ij}. \quad (\text{A.7})$$

b is the average van der Waals mole volume of the equation of state. This is computed using standard van der Waals mixing rules:

$$b = \sum_{i=1}^{nc} \left(\frac{n_i}{n} \right) b_i. \quad (\text{A.8})$$

b_i is the van der Waals mole volume of species i . This is computed as follows:

$$b_i = \frac{0.08664RT_{c_i}}{P_{c_i}} \quad (\text{A.9})$$

for the Redlich-Kwong-Soave equation of state, and

$$b_i = \frac{0.07780RT_{c_i}}{P_{c_i}} \quad (\text{A.10})$$

for the Peng-Robinson equation of state.

β_i is the parameter used in the equations for the computation of critical points; it is defined as

$$\beta_i = \frac{b_i}{b}. \quad (\text{A.11})$$

$\bar{\beta}$ is the parameter used in the equations for the computation of critical points; it is defined as

$$\bar{\beta} = \sum_{i=1}^C \Delta n_i \beta_i. \quad (\text{A.12})$$

F_{1-6} are the auxiliary functions used in the computation of critical points; they are defined as

$$\begin{aligned}
 F_1 &= \frac{1}{K-1}, \\
 F_2 &= \frac{2}{D_1-D_2} \left(\frac{D_1}{K+D_1} - \frac{D_2}{K+D_2} \right), \\
 F_3 &= \frac{1}{(D_1-D_2)} \left(\left(\frac{D_1}{K+D_1} \right)^2 - \left(\frac{D_2}{K+D_2} \right)^2 \right), \\
 F_4 &= \frac{1}{(D_1-D_2)} \left(\left(\frac{D_1}{K+D_1} \right)^3 - \left(\frac{D_2}{K+D_2} \right)^3 \right), \quad (\text{A.13}) \\
 F_5 &= \frac{2}{(D_1-D_2)} \ln \left(\frac{K+D_1}{K+D_2} \right), \\
 F_6 &= \left(\frac{2}{(D_1-D_2)} \right) \\
 &\quad \times \left(\left(\frac{D_1}{K+D_1} - \frac{D_2}{K+D_2} \right) - \ln \left(\frac{K+D_1}{K+D_2} \right) \right).
 \end{aligned}$$

K is the dimensionless volume; it is defined as

$$K = \frac{V}{nb}. \quad (\text{A.14})$$

$D_{1,2}$ are the parameters of the equation of state. Their values are calculated with the following formulas:

$$\begin{aligned}
 D_1 &= \frac{u_o + \sqrt{u_o^2 - 4w_o}}{2}, \\
 D_2 &= \frac{u_o - \sqrt{u_o^2 - 4w_o}}{2}, \quad (\text{A.15})
 \end{aligned}$$

where $u_o = 1$, $w_o = 0$ for the Redlich-Kwong-Soave equation of state and $u_o = 2$, $w_o = -1$ for the Peng-Robinson equation of state.

Acknowledgment

The author wishes to acknowledge the helpful comments and review provided by Professor Emmanuel Haven at the University of Leicester, UK.

References

- [1] R. B. Kearfott, *Rigorous Global Search: Continuous Problem*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [2] B. Stradi and E. Haven, "Optimal investment strategy via interval arithmetic," *International Journal of Theoretical and Applied Finance*, vol. 8, no. 2, pp. 185–206, 2005.
- [3] B. A. Stradi-Granados and E. Haven, "The use of interval arithmetic in solving a non-linear rational expectation based multiperiod output-inflation process model: the case of the IN/GB method," *European Journal of Operational Research*, vol. 203, no. 1, pp. 222–229, 2010.
- [4] B. A. Stradi, J. F. Brennecke, P. Kohn, and M. A. Stadtherr, "Reliable computation of mixture critical points," *AIChE Journal*, vol. 47, no. 1, pp. 212–221, 2001.
- [5] H. Gecegormez and Y. Demirel, "Phase stability analysis using interval Newton method with NRTL model," *Fluid Phase Equilibria*, vol. 237, no. 1-2, pp. 48–58, 2005.
- [6] Z. Galias, "Proving the existence of long periodic orbits in 1D maps using interval Newton method and backward shooting," *Topology and its Applications*, vol. 124, no. 1, pp. 25–37, 2002.
- [7] M. L. Michelsen and R. A. Heidemann, "Calculation of critical points from cubic two-constant equations of state," *AIChE Journal*, vol. 27, no. 2, pp. 521–523, 1981.
- [8] H. Hoteit, E. Santiso, and A. Firoozabadi, "An efficient and robust algorithm for the calculation of gas-liquid critical point of multicomponent petroleum fluids," *Fluid Phase Equilibria*, vol. 241, no. 1-2, pp. 186–195, 2006.
- [9] D. V. Nichita and S. Gomez, "Efficient and reliable mixture critical points calculation by global optimization," *Fluid Phase Equilibria*, vol. 291, no. 2, pp. 125–140, 2010.
- [10] M. L. Michelsen, "Calculation of phase envelopes and critical points for multicomponent mixtures," *Fluid Phase Equilibria*, vol. 4, no. 1-2, pp. 1–10, 1980.
- [11] M. L. Michelsen, "Phase equilibrium calculations. What is easy and what is difficult?" *Computers and Chemical Engineering*, vol. 17, no. 5-6, pp. 431–439, 1993.
- [12] M. L. Michelsen and H. Kistenmacher, "On composition-dependent interaction coefficients," *Fluid Phase Equilibria*, vol. 58, no. 1-2, pp. 229–230, 1990.
- [13] M. L. Michelsen and R. A. Heidemann, "Calculation of tri-critical points," *Fluid Phase Equilibria*, vol. 39, no. 1, pp. 53–74, 1988.
- [14] M. L. Michelsen, "Calculation of critical points and phase boundaries in the critical region," *Fluid Phase Equilibria*, vol. 16, no. 1, pp. 57–76, 1984.
- [15] J. Cai, H. Liu, Y. Hu, and J. M. Prausnitz, "Critical properties of polydisperse fluid mixtures from an equation of state," *Fluid Phase Equilibria*, vol. 168, no. 1, pp. 91–106, 2000.
- [16] T. Lindvig, L. L. Hestkjar, A. F. Hansen, M. L. Michelsen, and G. M. Kontogeorgis, "Phase equilibria for complex polymer solutions," *Fluid Phase Equilibria*, vol. 194–197, pp. 663–673, 2002.
- [17] M. Cismonti and M. L. Michelsen, "Global phase equilibrium calculations: critical lines, critical end points and liquid-liquid-vapour equilibrium in binary mixtures," *Journal of Supercritical Fluids*, vol. 39, no. 3, pp. 287–295, 2007.
- [18] C. Carstensen and M. S. Petković, "On iteration methods without derivatives for the simultaneous determination of polynomial zeros," *Journal of Computational and Applied Mathematics*, vol. 45, no. 3, pp. 251–266, 1993.
- [19] C. D. Maranas and C. A. Floudas, "Finding all solutions of nonlinearly constrained systems of equations," *Journal of Global Optimization*, vol. 7, no. 2, pp. 143–182, 1995.
- [20] R. B. Kearfott and M. Novoa III, "Algorithm 681 INTBIS, a portable interval Newton/bisection package," *ACM Transactions on Mathematical Software*, vol. 16, no. 2, pp. 152–157, 1990.
- [21] S. M. Rump, "INTLAB: INTerval LABoratory," in *Developments in Reliable Computing*, T. Csendes, Ed., pp. 77–104, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.
- [22] "Matlab R2011b (64 bit)," Mathworks, Natick, Mass, USA, 2011.
- [23] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, New York, NY, USA, 6th edition, 2010.

- [24] S. Nakamura, *Numerical Analysis and Graphical Visualization*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [25] S. Attaway, *Matlab: A Practical Introduction to Programming and Problem Solving*, Butterworth-Heinemann, New York, NY, USA, 2nd edition, 2011.
- [26] K. Ozaki, T. Ogita, S. M. Rump, and S. Oishi, “Fast algorithms for floating-point interval matrix multiplication,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 7, pp. 1795–1814, 2012.
- [27] S. M. Rump and T. Ogita, “Super-fast validated solution of linear systems,” *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 199–206, 2007.
- [28] R. C. Reid, J. M. Prausnitz, and B. E. Poling, *The Properties of Gases and Liquids*, McGraw-Hill, New York, NY, USA, 4th edition, 1987.
- [29] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK, 1990.
- [30] R. Moore, *Interval Analysis*, Prentice-Hall, Upper Saddle River, NJ, USA, 1966.