A New Algorithm for Rachford-Rice for Multiphase Compositional Simulation

R. Okuno, R.T. Johns, and **K. Sepehrnoori,** SPE, The University of Texas at Austin

Summary

Flash calculations for use in compositional simulation are more difficult and time-consuming as the number of equilibrium phases increases beyond two. Because of its complexity, many simulators do not even attempt to incorporate three or more hydrocarbon phases, even though such cases are important in many low-temperature gasfloods or for high temperatures where hydrocarbons can partition into water. Multiphase flash algorithms typically use successive substitution (SS) followed by Newton's method. For N_P-phase flash calculations, (N_P–1) Rachford-Rice (RR) equations are solved in every iteration step in SS and, depending on the choice of independent variables, in Newton's method. Solution of RR equations determines both compositions and amounts of phases for a fixed overall composition and set of K-values. A robust algorithm for RR is critical to obtain convergence in multiphase compositional simulation and has not been satisfactorily developed, unlike the traditional two-phase flash. In this paper, we develop an algorithm for RR equations for multiphase compositional simulation that is guaranteed to converge to the correct solution independent of the number of phases for both positive and negative flash calculations.

We derive a function whose gradient vector consists of RR equations. This correct solution to the RR equations is formulated as a minimization of the nonmonotonic convex function using the independent variables of (N_P–1) phase mole fractions. The key to obtaining a robust algorithm is that we specify nonnegative constraints for the resulting equilibrium phase compositions, which are described by a very small region with no poles. The minimization uses Newton's direction with a line-search technique to exhibit superlinear convergence. We show a case in which a previously developed method cannot converge while our algorithm rapidly converges in a few iterations. We implement the algorithm both in a standalone flash code and in UTCOMP (Chang et al. 1990), a multiphase compositional simulator, and show that the algorithm is guaranteed to converge when a multiphase region exists as indicated by stability analysis.

Introduction

Mixtures of CO_2 and reservoir oil can exhibit complex phase behavior, especially at temperatures typically below 120°F where a third CO_2-rich liquid phase can coexist with the oleic and gaseous phases. Accurate numerical simulation of CO_2 flooding involving such complex phase behavior requires robust algorithms for multiphase equilibrium calculations. Nghiem and Li (1986) examined the importance of the third phase in multiphase compositional simulation and concluded that two-phase equilibrium approximations can be used with little loss of accuracy because the three-phase region exists only over a small part of the reservoir. However, their study was based on 1D slimtube simulations using only two recombined oils from the same field.

Khan et al. (1992) and Wang and Strycker (2000) later demonstrated displacements in which oil recoveries using only two-phase equilibrium calculations are significantly different from those made with three-phase calculations. Okuno (2009) and Okuno et al. (2009) demonstrated that the two-phase equilibrium approximation can result in nonconvergence of the simulation because of discontinuous changes in physical properties over a timestep. Therefore, numerical simulation of CO_2 flooding at low temperatures should be capable of multiphase equilibrium calculations.

Phase-behavior calculations in compositional simulation consist of stability analysis and flash calculations (see Appendix A). Stability analysis can detect instability of a phase mixture by searching for a negative value of the tangent-plane distance function (Michelsen 1982b). If unstable, the phase must split into two or more phases. A stability analysis that detects the presence of a third phase can also give the first initial estimates of the K-values for subsequent flash calculations.

Flash calculations initially use SS to provide better K-value estimates for final convergence of the equilibrium phase compositions using Newton's iterations. SS iterations and, depending on the choice of independent variables, Newton's iterations involve two loops—the outer iteration loop containing the inner iteration loop. K-values are updated on the basis of fugacity equations in the outer iteration loop. Compositions and mole fractions for phases are calculated to satisfy material-balance equations in the inner iteration loop. RR equations are commonly used for the inner loop as originally proposed by Rachford and Rice (1952). The goal of the RR iteration is to determine the phase mole fractions and phase compositions for a fixed overall composition and set of specified K-values. The solution of the RR equation is often referred to as a constant-K flash calculation. This paper focuses on the constant-K flash calculation (inner iteration) as part of a rigorous equation of state (EOS) flash (outer iteration) for compositional reservoir simulation.

There are two different possibilities for a constant-K flash calculation within the framework of compositional reservoir simulation. The first, a positive flash, is a N_P-phase flash where the overall composition lies within the N_P-phase region, resulting in positive phase mole fractions. The second, a negative flash, is a flash where the overall composition is not within the N_P-phase region, resulting in one or more negative phase mole fractions. In compositional reservoir simulation, the algorithm must be capable of performing both types of flash calculations successfully. Because a flash calculation is performed only when phase instability is detected on the basis of the tangent-plane criterion (Baker et al. 1982), phase mole fractions must be positive once the outer iteration correctly achieves convergence. However, equilibrium phase mole fractions calculated in a constant K-flash calculation can be negative before convergence, depending on the K-values provided by the most recent outer iteration.

Another type of negative flash that is not considered in this paper is one where the overall composition lies in negative composition space. This kind of negative flash can be used for minimum-miscibility-pressure calculations (Yuan and Johns 2005) but is not needed in compositional reservoir simulation.

The robustness of two-phase constant-K flash calculations is not an issue in compositional simulation because it is a root-finding problem for a univariate monotonic function that can be solved safely by a combination of bisection and Newton's method coupled with underrelaxation. The solution lies in the feasible region as identified previously by Whitson and Michelsen (1989). Bisection is performed to make the region narrower, while underrelaxation is used to avoid the Newton's iteration jumping out of the region of interest.

For multiphase cases, however, the increase in the number of independent variables to be solved makes the behavior of RR equations more implicit and complicated. There are several attempts to

This paper (SPE 117752) was accepted for presentation at the SPE Eastern Regional/AAPG Eastern Section Joint Meeting, Pittsburgh, Pennsylvania, USA, 11–15 October 2008, and revised for publication. Original manuscript received for review 16 October 2008. Revised manuscript received for review 11 August 2009. Paper peer approved 19 August 2009.

solve this multiphase K-value problem, which are generally of two types: root finding and minimization. The root-finding approach is more commonly used and solves a system of (N_P–1) RR equations with Newton's method (Nelson 1987; Bünz et al. 1991; Eubank 2006), none of which considers constraints for negative flash calculations. Leibovici and Neoschil (1995) extended the feasible region of Whitson and Michelsen to multiphase calculations and used Newton's method with underrelaxation. However, Leibovici and Nichita (2008) stated that the algorithm does not work for some situations, and we demonstrate in this paper a case where the method cannot converge. Haugen et al. (2007) proposed a 2D bisection algorithm to provide a good initial estimate for Newton's iterations for three-phase calculations. However, their procedure can take numerous iterations and likely does not always converge, especially when an overall composition is close to a critical endpoint. This is consistent with the statement by Dennis and Schnabel (1996) that the bisection method does not extend naturally to multiple dimensions.

Michelsen (1994) developed a novel approach to multiphase constant-K flash calculations by rewriting the RR-equation solution as the minimization of a convex function. Although no derivation of the function is given in his paper, the approach in our opinion is superior to the root-finding approach. Michelsen's algorithm, however, considered only positive flash calculations. Leibovici and Nichita (2008) attempted to overcome this problem by formulating the minimization using the feasible region of Leibovici and Neoschil (1995), resulting in a minimization problem with an open constraint set. Their feasible region is confined by poles where the function cannot be defined, so that the feasible region is not sufficient and convergence problems can occur. Also, Leibovici and Nichita (2008) did not develop an algorithm to solve their minimization problem. Instead, they used multipurpose minimization software. Robustness and efficiency of their minimization depend on the algorithm used because simple minimization algorithms cannot handle the open constraint set.

The importance of multiphase flash calculations for compositional simulation requires a robust and efficient algorithm for multiphase constant-K flash calculations that always converges. The main objective of this research is to develop an algorithm for which convergence can be analytically proved for any number of phases for both positive and negative constant-K flash calculations. First, we derive the convex function that we minimize to solve for the equilibrium phase mole fractions. This objective function is the same as Michelsen (1994), but we explicitly derive the function so that our algorithm can be clearly understood. The function is then analyzed in detail to develop a robust and efficient algorithm using Newton's method with a line-search technique. After that, we present example standalone calculations for three and five phases. Last, our algorithm is implemented in a multiphase compositional simulator and used in an example simulation of a low-temperature oil displacement by CO_2 involving three-hydrocarbon-phase flow.

Multiphase Equilibrium Calculations in Compositional Simulation

Multiphase equilibrium calculations in UTCOMP (Chang et al. 1990) consist of stability analysis and flash calculations. Appendix A presents a flow chart for the multiphase equilibrium calculations in UTCOMP. Although there is a wide variety of flash algorithms, a procedure for a multiphase flash calculation solves for the independent variables $\ln K_{ij}$ (i = 1, ..., N_C and j = 1, ..., N_P–1) using the fugacity equations,

$$\ln\left(x_{ij}\varphi_{ij}\right)-\ln\left(x_{iN_P}\varphi_{iN_P}\right)=0 \quad \text{for } i=1,...,N_C \;\; j=1,...,N_P-1 \quad \ldots\ldots\ldots (1)$$

with the material-balance equations

$$z_i=\sum_{j=1}^{N_P}\beta_j x_{ij} \quad \text{for } i=1,...,N_C, \quad \ldots\ldots\ldots (2)$$

$$1=\sum_{j=1}^{N_P}\beta_j, \quad \ldots\ldots\ldots (3)$$

and

$$\sum_{i=1}^{N_C}x_{ij}=1 \quad \text{for } j=1,...,N_P, \quad \ldots\ldots\ldots (4)$$

and definitions,

$$K_{ij}=x_{ij}/x_{iN_P} \quad \text{for } i=1,...,N_C \;\; j=1,...,N_P-1. \quad \ldots\ldots\ldots (5)$$

In Eqs. 1 through 5, the N_Pth phase is selected arbitrarily as the reference phase.

The multiphase RR equations are derived using Eqs. 2 through 5. Substitution of Eqs. 3 and 5 into Eq. 2 eliminates the dependent phase mole fraction and phase compositions resulting in

$$x_{iN_P}=z_i/t_i, \quad \ldots\ldots\ldots (6)$$

where $t_i=1-\left[\sum_{j=1}^{N_P-1}\left(1-K_{ij}\right)\beta_j\right]$ for $i=1,...,N_C$.
Using Eqs. 4 and 5, we obtain

$$\sum_{i=1}^{N_C}\left(1-K_{ij}\right)x_{iN_P}=0 \quad \text{for } j=1,...,N_P-1. \quad \ldots\ldots\ldots (7)$$

Substitution of Eq. 6 into Eq. 7 results in the final form of the multiphase RR equations,

$$f_j(\beta)=\sum_{i=1}^{N_C}\left(1-K_{ij}\right)z_i/t_i \quad \text{for } j=1,...,N_P-1, \quad \ldots\ldots\ldots (8)$$

where β is a vector with elements β_j (j = 1, …, N_P–1).

Flash calculations in UTCOMP initially use accelerated SS but are switched to minimization of the Gibbs free energy once accelerated SS satisfies a specified switching criterion (Perschke et al. 1989). In this research, we replaced accelerated SS with conventional SS to start the second-order convergence method robustly. Appendix B gives the SS algorithm where K-values are updated in the outer iteration loop using Eq. 1 and a selected cubic EOS and the RR equations are solved in the inner iteration loop to determine phase compositions and phase mole fractions. The minimization of the Gibbs free energy that follows the SS steps in UTCOMP does not use the RR equations because the independent variables $\beta_j x_{ij}$ (i = 1, ..., N_C and j = 1, …, N_P–1) can explicitly provide phase compositions and phase mole fractions (Michelsen 1982a).

Constant-K Flash Formulation and Algorithm

In this section, we derive a new algorithm for a N_P-phase constant-K flash calculation that is based on a minimization of a convex function with N_C linear constraints. The convex function is similar to the one presented by Michelsen (1994). Unlike previous research, however, we derive and analyze the behavior of the function to develop a practical and robust algorithm for compositional simulation. We also provide a very small region in which the solution to the RR equations must lie.

Convex Function. The convex function is derived by exploiting a special structure of the RR equations. We begin by examining the common root-finding technique to solve Eq. 8 by constructing the Jacobian matrix where the elements are given by

$$J_{jk}=\sum_{i=1}^{N_C}\left[\left(1-K_{ij}\right)\left(1-K_{ik}\right)z_i\right]/t_i^2,$$

where $j,k=1,...,N_P-1$. The Jacobian matrix is symmetric, indicating that there is a scalar function $F(\beta)$ for which the gradient vector consists of the RR equations. This is best explained by examining the relation between Newton's methods for root finding and those for minimization. That is, the iteration schemes for Newton's method are

$$\beta^{n+1}=\beta^n-\left[\nabla f\left(\beta^n\right)^T\right]^{-1}f\left(\beta^n\right) \text{ for root finding}$$

and

$$\beta^{n+1}=\beta^n-\left[\nabla^2 F\left(\beta^n\right)\right]^{-1}\nabla F\left(\beta^n\right) \text{ for minimization.}$$

Although the derivations of these schemes are different from each other, the latter can be viewed as a special case of the former. That is, f corresponds to the gradient of F if and only if the Jacobian matrix of f is symmetric (Bertsekas 1999).

The function F to be minimized is then easily found by performing an indefinite integral of f_j with respect to β_j to obtain

$$F(\beta)=\sum_{i=1}^{N_C}\left(-z_i \ln|t_i|\right). \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

In Eq. 9, the constant of integration is set to zero without loss of generality. Eq. 9 is identical to the one presented by Michelsen (1994) except that he retained the integration constant.

There is a significant advantage in using minimization to solve the multiphase constant-K flash problem. The minimization function F defined by Eq. 9 is easily proved to be convex because its Hessian matrix is positive semidefinite, as shown in Appendix C. A summary for Appendix C is that, for an $N_C\times(N_P-1)$ matrix $\{1-K_{ij}\}$ of full rank, the Hessian matrix of F is positive definite and the function is strictly convex. Otherwise, the Hessian matrix is only positive semidefinite. A semidefinite Hessian occurs for example at critical points, including critical endpoints [called bicritical points in Haugen et al. (2007)] where two of the three phases are critical and at equilibrium with the other noncritical phase.

Another important behavior of the function is that if the region $t_i > 0$ ($i = 1, \ldots, N_C$) is unbounded, then the function becomes monotonic. That is, the function does not have a minimum and there is no solution to the constant-K flash with N_P phases. The proof of this statement is straightforward. If the region is unbounded, there exists a vector d such that

$$\sum_{j=1}^{N_P-1}\left(K_{ij}-1\right)d_j \ge 0 \quad \text{for all } i.$$

Thus, the function is always nonincreasing along the vector d when

$$d^T\nabla F=\sum_{j=1}^{N_P-1}\sum_{i=1}^{N_C}-z_i\left(K_{ij}-1\right)d_j/t_i \le 0.$$

This result is in contrast to Leibovici and Nichita (2008) who incorrectly showed a case where the region is unbounded, but the function has a minimum (Fig. 4 with their explanation in section 4.2 "Reservoir oil/CO_2 mixture").

Constraints. Similarly to the traditional two-phase constant-K flash, the function has N_C poles defined when $t_i = 0$ ($i = 1, \ldots, N_C$). Positive values of t_i ($i = 1, \ldots, N_C$) define the most simplistic feasible region for the solution of a multiphase constant-K flash calculation. This is the same type of the feasible region used in the root-finding technique of Leibovici and Neoschil (1995) and the minimization approach of Leibovici and Nichita (2008). To formulate a minimization of F, Leibovici and Nichita (2008) used the constraint set $L'=\{\beta \mid t_i > 0,\ i = 1, \ldots, N_C\}$. Care must be taken in handling this open set L'. When one simplifies the open set to be a closed set $L = \{\beta \mid t_i \ge 0, i = 1, \ldots, N_C\}$, iterations can lie on a pole, where the function cannot be defined and the calculations will then cease. When an iteration point is near the pole, Newton's iteration makes little improvement in minimizing the objective function, if any, because of the nonquadratic behavior along the poles. The convergence problem is significant because it can stop the simulation from proceeding.

To increase the robustness of the iterations, we develop a new feasible region of smaller size that also contains no poles. Our feasible region is derived on the basis of nonnegativity of phase component mole fractions, $0 \le x_{ij} \le 1$ ($i = 1, \ldots, N_C$ and $j = 1, \ldots, N_P$). To allow for a negative flash, we consider that phase mole fractions can be negative. This is important because, as iterations proceed in EOS flash calculations, a few iterations may be outside of the multiphase region of interest but, with more iterations (if allowed), can move back into the multiphase region.

Using Eqs. 5 and 6, the nonnegativity of phase component mole fractions requires the following inequalities:

$$0 \le z_i \le t_i = 1-\left[\sum_{j=1}^{N_P-1}\left(1-K_{ij}\right)\beta_j\right]$$

$$\text{and } \ 0 \le K_{ij}z_i \le t_i = 1-\left[\sum_{j=1}^{N_P-1}\left(1-K_{ij}\right)\beta_j\right].$$

The final form of constraints is easily derived as

$$a_i^T\beta \le b_i, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

where $a_i=\{1-K_{ij}\}$, $\beta=\{\beta_j\}$, and $b_i=\min\{1-z_i, \min_j\{1-K_{ij}z_i\}\}$ for $i = 1, \ldots, N_C$, $j = 1, \ldots, N_P-1$. The constraint set $S = \{\beta \mid a_i^T\beta \le b_i,\ i = 1, \ldots, N_C\}$ leads to a smaller feasible region than that based on the set $L = \{\beta \mid t_i \ge 0,\ i = 1, \ldots, N_C\}$. The new set S does not contain the region where the minimization function exhibits poor convergence. The feasible region developed in this section is the primary but important difference between our algorithm and Michelsen (1994) because this feasible region leads to robustness for both positive and negative flash calculations.

Algorithm. In this subsection, we present a detailed algorithm that can be used to solve the multiphase constant-K flash calculations robustly. Our formulation for multiphase constant-K flash developed in the preceding subsection is as follows:

$$\text{Minimize } \ F(\beta)=\sum_{i=1}^{N_C}\left(-z_i \ln|t_i|\right) \ \text{ subject to } \ a_i^T\beta \le b_i.$$

We consider only cases where the minimization has a unique solution. That is, the minimization function is convex and nonmonotonic over the constraint set $S = \{\beta \mid a_i^T\beta \le b_i,\ i = 1, \ldots, N_C\}$. This is the case in compositional simulation because N_P-phase flash calculations are performed only when existence of a N_P-phase solution is ensured by stability tests (see Appendix A). Stability analysis can fail, but, when it does so, it fails to predict the existence of an additional phase, not the other way around. The minimization of the convex function is safely solved using search directions and step sizes based on Newton's method coupled with a line search along that direction.

In multiphase compositional simulation, such as with UTCOMP, a good initial estimate may be available from the previous timestep or from the previous iteration step in the flash calculation. Otherwise, we need to make an initial estimate of β to start the iterations. Haugen et al. (2007) proposed 2D bisection to initiate Newton's iterations for three-phase constant-K flash calculations. However, analysis of our minimization function explicitly generates good initial estimates for β by considering N_P constraints for phase mole fractions, $\beta_j \ge 0$ ($j = 1, \ldots, N_P$), in addition to Eq. 10. A feasible initial estimate is then determined by an equally weighted mean of the vertices of the intersection of the sets S and $P = \{\beta \mid \beta_j \ge 0,\ j = 1, \ldots, N_P\}$.

An algorithm to solve the minimization is as follows:

1. Generate an initial estimate for the independent variables β_j ($j = 1, \ldots, N_P-1$) based on the previous timestep, the previous iteration step, or an equally weighted mean of the vertices of the S and P intersection.

2. Calculate the gradient of the convex function F and set an iteration index n to unity.

3. Calculate the Hessian matrix and solve the system of equations to obtain the Newton's direction. That is, calculate $d^n=-\left(\nabla^2F^n\right)^{-1}\nabla F^n$.

4. Calculate the maximum feasible step size λ_{max} along the Newton's direction d^n, as shown below. If $\lambda_{max} \ge 1$, then set λ_{max} to 1.0 resulting in a unit Newton's step,

$$\lambda_{max}=\min_i\left\{\frac{b_i-a_i^T\beta^n}{a_i^Td^n}: a_i^Td^n>0\right\}.$$

5. Determine a step size λ^n using a line search technique (see Appendix D) applied over the range of $0 \le \lambda^n \le \lambda_{max}$.

6. Update the independent variables, $\beta_j^{n+1}=\beta_j^n+\lambda^n d^n$.

7. If the maximum norm of the gradient is less than a specified tolerance (e.g., 10^{-8}), stop. Otherwise, continue to Step 8.

8. Increase the iteration index ($n = n+1$) and go back to Step 3.

The Newton's direction is guaranteed to be a descent direction because the Hessian matrix is positive definite. In Step 3, the Hessian matrix can be inverted inexpensively for three-phase cases. Step 4 calculates the maximum feasible step size λ_{max} along the Newton's direction. The value of λ_{max} less than unity indicates

that the full Newton's step would go out of the feasible region, so it is not allowed. The line search used is based on the univariate Newton's method with a relatively large tolerance (e.g., 10^{-3}) for the maximum norm of the gradient. The procedure ensures that the Newton's iterations do not go out of the feasible region.

Comparisons in Standalone Calculations

In this section, we perform example calculations for three and five phases, although our algorithm can be applied with complete robustness for even more phases. Because five-phase calculations cannot be illustrated in a figure, three-phase calculations are discussed in more detail. For standalone calculations, there are no previous timesteps or iteration steps to use as an initial guess. We compare two algorithms to perform standalone calculations: the minimization with the constraint set S (MS) developed in this research and the conventional root-finding algorithm with the constraint set $L = \{\beta \mid t_i \geq 0, i = 1, \ldots, N_C\}$ (RL) developed by Leibovici and Neoschil (1995). For underrelaxation in the RL algorithm, we use 0.9 as the underrelaxation parameter.

To compare the two methods, we randomly generated overall compositions and K-values according to the following procedure:

1. Randomly generate N_P positive phase compositions. K-values are calculated on the basis of those compositions.
2. Randomly generate positive phase mole fractions, and calculate overall composition using Eq. 2.
3. Solve the generated problem using either MS or RL.
4. Go back to Step 1.

Thus, we are solving only flash calculations here where the phase mole fractions are positive. This is the most frequently encountered situation in compositional simulation.

Initial estimates for MS are generated as described in the preceding section. For RL, initial estimates for β_j ($j = 1, \ldots, N_P-1$) are set to $1/N_P$. This initial estimate is always feasible for RL because it corresponds to an equally weighted mean of vertices of the set $P = \{\beta \mid \beta_j \geq 0, j = 1, \ldots, N_P\}$, which is contained by the set L. Initial estimates generated for MS and RL are in general different from each other because they are based on the different feasible regions. The stopping criteria for MS and RL are 10^{-8} for the maximum norm of the gradient of F and for the maximum residual of the RR equations, respectively.

The results show that the average numbers of iterations per flash are almost the same: 3.5 for MS and 3.6 for RL for three phases and 3.8 for MS and 3.7 for RL for five phases. However, the number of iterations per problem is more problem-dependent for RL than for MS as shown in **Fig. 1.** The maximum number of iterations for the 1 million flash calculations when three phases are present is seven for MS, compared to 20 for RL. For five phases, MS requires only six iterations at most, compared to 20 for RL.

For example, consider the flash with overall composition and K-values shown as Example 1 in **Table 1.** As shown in **Fig. 2,** our algorithm converges in only four iterations compared to 20 for RL. **Fig. 3** shows the iteration path in β space for both methods. The shaded region in this figure is the feasible region, which is the intersection of the zones above the solid lines and those below the dashed lines. For RL, the first Newton's step does not go out of its feasible region, but goes very near one of the poles. Once an iteration approaches a pole, it takes many iterations for it to move away from the pole. This is best understood by considering the behavior of the function to be minimized instead of a system of equations actually solved by RL. As the pole is approached, the function exhibits more-planar shape, and the quadratic approximation, on which Newton's method is based, becomes less appropriate. Also, the nearly planar shape of the function results in a large condition number of the Hessian matrix. **Fig. 4** shows the condition numbers of the Hessian matrix for MS and the Jacobian matrix for RL during the iterations. Although the largest condition number for RL is 8.1 million for this case, the condition number can be much larger depending on how close the iteration point is to the pole. If the condition number is too large, the calculation cannot converge because of round-off errors.

Example 2 in Table 1 shows a case where RL cannot converge. As shown in **Fig. 5,** RL cannot decrease the residual of the RR equations, while MS converges in only four iterations. **Fig. 6** shows the iteration path in β space for each algorithm. For MS, a unit Newton's step for the first iteration moves out of its feasible region, and the line search is performed to determine the next iteration point. For RL, a unit Newton's step does not go out of its feasible region, but the resulting iteration point is located very near one of the poles. Because of the round-off errors caused by a significantly large condition number of the Jacobian matrix, the iterations cannot move away from the pole. **Fig. 7** shows the condition number as a function of a step size along the Newton's direction for the first iteration for each algorithm. A unit step size for each algorithm corresponds to a point on the boundary of its feasible region. For RL, the Jacobian matrix can be significantly ill-conditioned in the vicinity of a pole, resulting in poor convergence. The region of poor convergence is never encountered in our MS method.

Example 3 in Table 1 gives a case where thermodynamic conditions are close to a critical endpoint [called a bicritical point in Haugen et al. (2007)] (i.e., a point in P-T-x space at which two of the three phases merge in the presence of the other noncritical phase). The conditions near a critical endpoint are indicated by the two sets of K-values, which become equal to each other. The area of the feasible region is 25 times larger for RL than that for MS, as shown in **Fig. 8. Fig. 9** compares the convergence behaviors of MS and RL, where MS converges in four iterations compared to seven

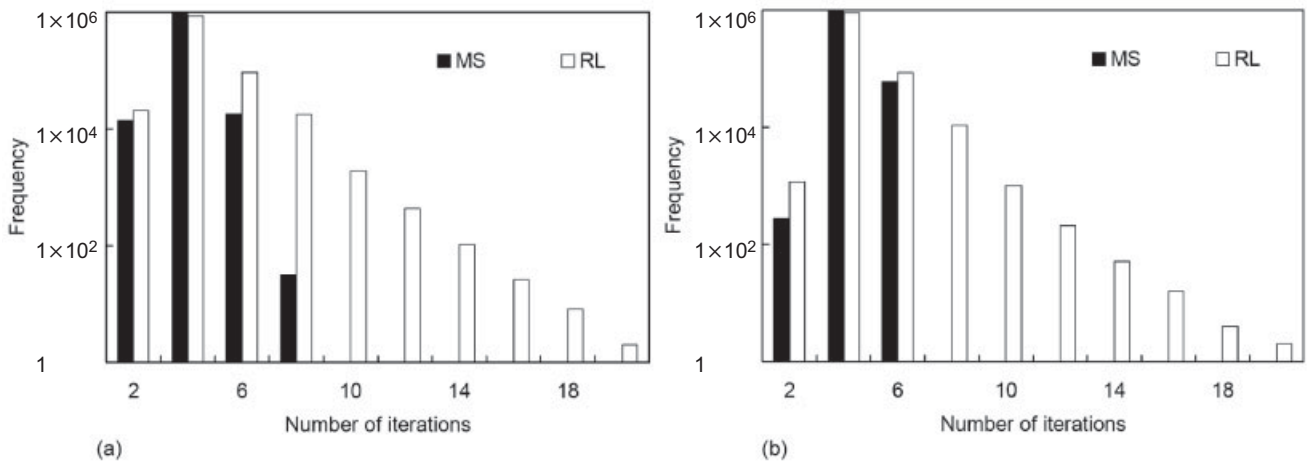

Fig. 1—Frequency distribution of the numbers of iterations required for convergence for 1 million flash calculations. (a) Three-phase calculations. (b) Five-phase calculations.

TABLE 1—EXAMPLE OVERALL COMPOSITIONS AND CONSTANT-*K* VALUES FOR FLASH CALCULATIONS

Example 1

Component	z_i	K_{i1}	K_{i2}
1	0.204322076984	1.23466988745	1.52713341421
2	0.070970999150	0.89727701141	0.02456487977
3	0.267194323384	2.29525708098	1.46348240453
4	0.296291964579	1.58954899888	1.16090546194
5	0.067046080882	0.23349348597	0.24166289908
6	0.062489248292	0.02038108640	0.14815282572
7	0.031685306730	1.40715641002	14.3128010831

Example 2

Component	z_i	K_{i1}	K_{i2}
1	0.132266176697	26.3059904941	66.7435876079
2	0.205357472415	1.91580344867	1.26478653025
3	0.170087543100	1.42153325608	0.94711004430
4	0.186151796211	3.21966622946	3.94954222664
5	0.111333894738	0.22093634359	0.35954341233
6	0.034955417168	0.01039336513	0.09327536295
7	0.159847699672	19.4239894458	12.0162990083

Example 3

Component	z_i	K_{i1}	K_{i2}
1	0.896646630194	1.64571122126	1.61947897153
2	0.046757914522	1.91627717926	2.65352105653
3	0.000021572890	0.71408616431	0.68719907526
4	0.000026632729	0.28582415424	0.18483049029
5	0.016499094171	0.04917567928	0.01228448216
6	0.025646758089	0.00326226927	0.00023212526
7	0.014401397406	0.00000570946	0.00000003964

Example 4

Component	z_i	K_{i1}	K_{i2}	x_{i1}	x_{i2}	x_{i3}
1	0.08860	0.112359551	1.011235955	0.100	0.900	0.890
2	0.81514	13.72549020	0.980392157	0.700	0.050	0.051
3	0.09626	3.389830508	0.847457627	0.200	0.050	0.059

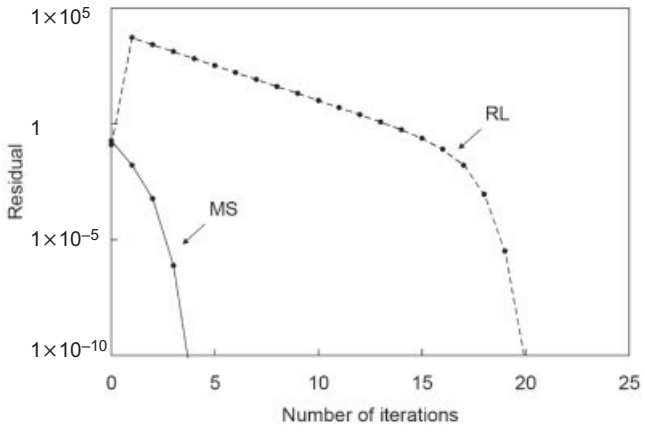

Fig. 2—Convergence behaviors of the MS and RL algorithms for Example 1 in Table 1. MS takes only four iterations, compared to 20 for RL.

for RL. As shown in **Fig. 10,** the smaller feasible region provides a better initial estimate for MS and also reduces the number of iterations required for convergence. The correct solution to which MS converges is $(\beta_1, \beta_2) = (0.87, 2.2\times10^{-6})$, which is very near a phase boundary. This example also shows that our algorithm has no convergence problem in the vicinity of phase boundaries.

Example 4 in Table 1 also provides conditions near a critical endpoint for three components. This example is given, however, to demonstrate that our algorithm is capable of robust negative flash calculations even for near-critical conditions. **Fig. 11** shows the tie-triangle on the ternary diagram. The elongated tie-triangle indicates that two of the three phases are near-critical and at equilibrium with the other highly immiscible phase. We consider the negative flash calculation for this near-critical mixture at an overall composition located outside the tie-triangle, as shown in Fig. 11. Fig. 11 shows that MS converges to the solution in five iterations, while RL does so in eight iterations. The iteration paths taken in β space are given in **Fig. 12.** Fig. 12 shows only part of the feasible region for clarity, but the feasible region is bounded as proved in the Convex Function subsection of this paper. The larger feasible

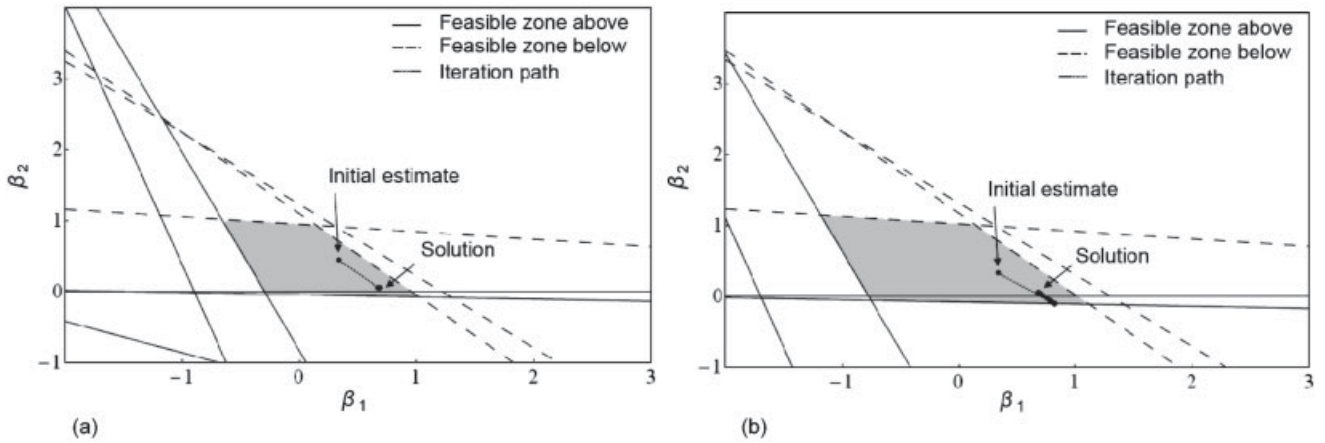

Fig. 3—Iteration path for Example 1 in Table 1. The shaded region is the feasible region, which is the intersection of the zones above the solid lines and those below the dashed lines. (a) The MS algorithm. (b) The RL algorithm.

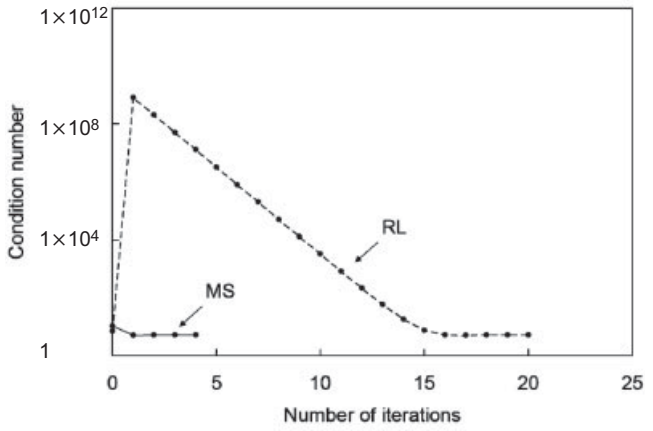

Fig. 4—Condition number during the iteration for Example 1 in Table 1. The condition number for RL becomes very large near the pole demonstrating the poor convergence properties of RL.

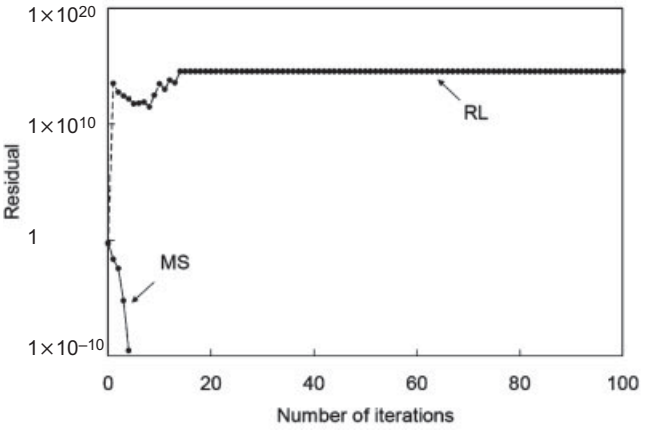

Fig. 5—Convergence behaviors of the MS and RL algorithms for Example 2 in Table 1. RL cannot converge, while MS converges in only four iterations.

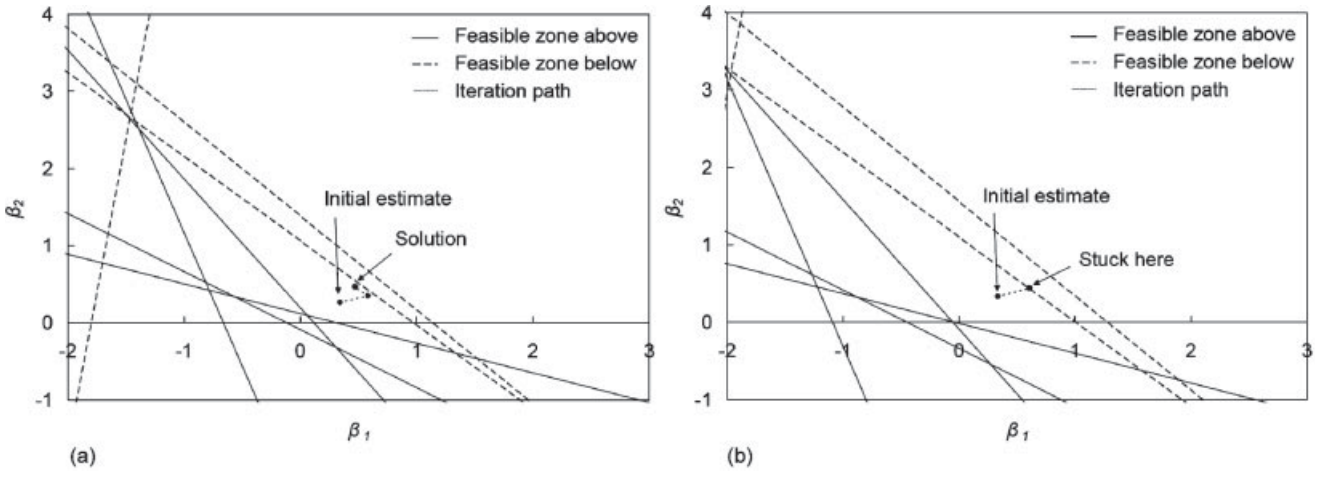

Fig. 6—Iteration path for Example 2 in Table 1. The feasible region is the intersection of the zones above the solid lines and those below the dashed lines. (a) The MS algorithm. (b) The RL algorithm. The iterations of RL become stuck very near the pole.

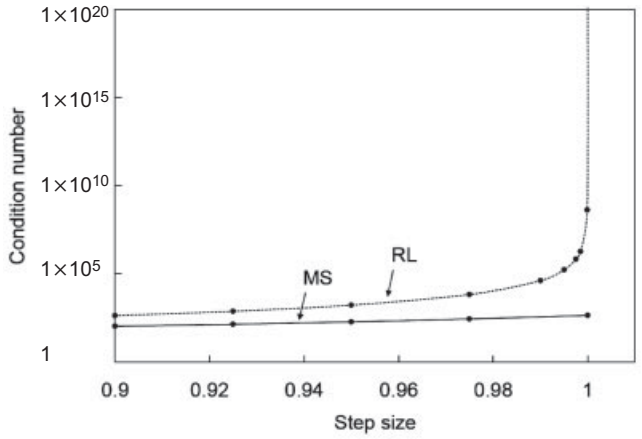

Fig. 7—Condition number along Newton's direction for the first iteration step for Example 2 in Table 1. The condition number for RL rapidly increases as the feasibility limit is approached, while that for MS increases only slightly.

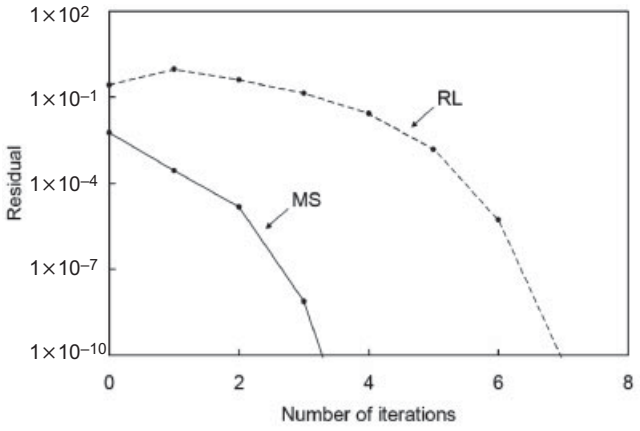

Fig. 9—Convergence behaviors of the MS and RL algorithms for Example 3 in Table 1.

region for RL results in a less accurate initial estimate and more iterations to convergence because the feasible region for RL is bigger and exhibits poor convergence near the poles.

MS occasionally performs several iterations for line search in the first iteration. In terms of computational effort, the line search is inexpensive, taking only 20% of the calculation time for one iteration of RL.

Simulation Case Study

We implemented and tested our new algorithm in various simulation runs with UTCOMP to confirm that it is applicable to multiphase compositional simulation. We emphasize here that our constant-K flash algorithm works as part of rigorous EOS flash calculations within UTCOMP. That is, we do not assume constant K-values for our simulations. This section presents one simulation example, where the computations are performed using a Pentium 4 at 3.0 GHz and 2.0 GB of RAM.

We consider gas injection (impure CO_2 injection) for one pore volume in a quarter of a staggered-line-drive pattern. A permeability field is stochastically generated for a 2D reservoir model, as shown in **Fig. 13.** The reservoir oil used is the BSB west Texas oil, for which the minimum miscibility pressure for pure CO_2 injection was experimentally measured to be approximately 1,200 psia at the reservoir temperature of 105°F (Khan et al. 1992). The reservoir and fluid properties are summarized in **Tables 2 and 3,** respectively. For the seven-component-fluid model, the Peng-Robinson EOS (Peng and Robinson 1976) is used with the van der Waals mixing rules. The parameters for the relative permeability model for the CO_2-rich liquid phase are assumed to be the same as those for the gaseous phase. The aqueous phase is at its residual saturation. The initial, injection, and production pressures are set to 1,100, 1,250, and 1,100 psia, respectively.

In the flash calculations, we use SS followed by minimization of the Gibbs free energy. Stability analysis uses the stationary point method (Michelsen 1982b) with SS followed by Newton's method. The switching criteria from SS to the second-order convergence methods are 10^{-3} both for flash calculations and stability analysis. The stopping criteria for flash calculations and stability analysis are 10^{-8}, while that for the constant-K flash calculations is 10^{-10}. Stability analysis for a single phase is performed only for well cells and cells adjacent to two- or three-phase cells, as described by Young and Stephenson (1983) and implemented in UTCOMP by Chang (1990).

The resulting oil recovery is shown in Fig. 13. **Fig. 14** shows the saturation distributions of the oleic, gaseous, and CO_2-rich liquid phases, and the distribution of the number of phases both at the breakthrough time of 0.45 pore volumes injected (PVI). The three-phase region exists in 23% of the total number of grid cells. Nghiem and Li (1986) observed that a three-phase region exists only in a small part of grid cells in their 1D simulations and concluded that the three-phase region can be ignored by using only two-phase flash calculations with little loss of accuracy. However, as shown here, the number of cells in a three-phase equilibrium state can be much higher when 2D simulations are considered.

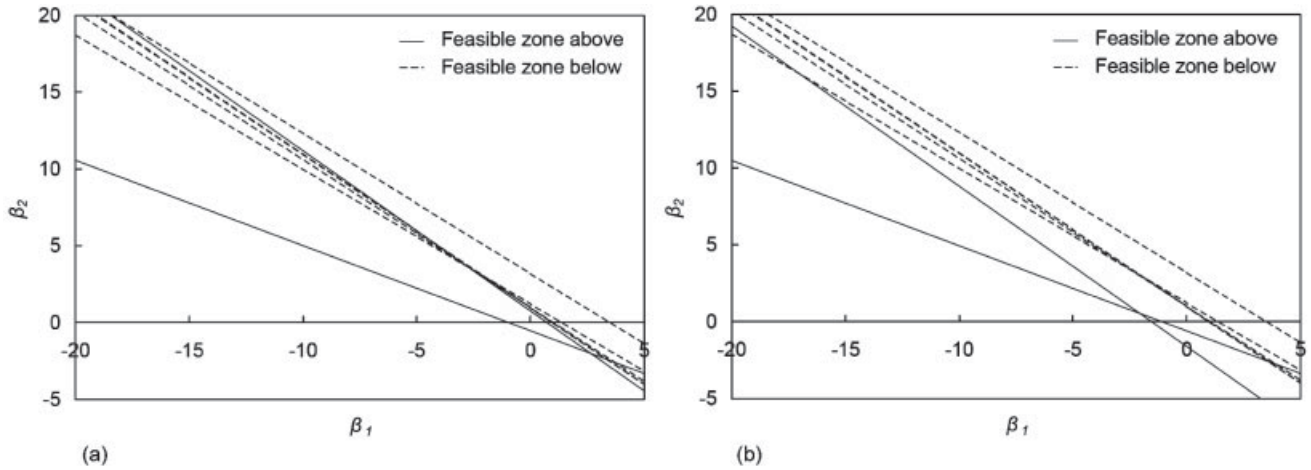

Fig. 8—Feasible region for Example 3 in Table 1 that is shown as the intersection of the zones above the solid lines and those below the dashed lines. (a) The MS algorithm. (b) The RL algorithm. The feasible region is 25 times larger for RL than that for MS.

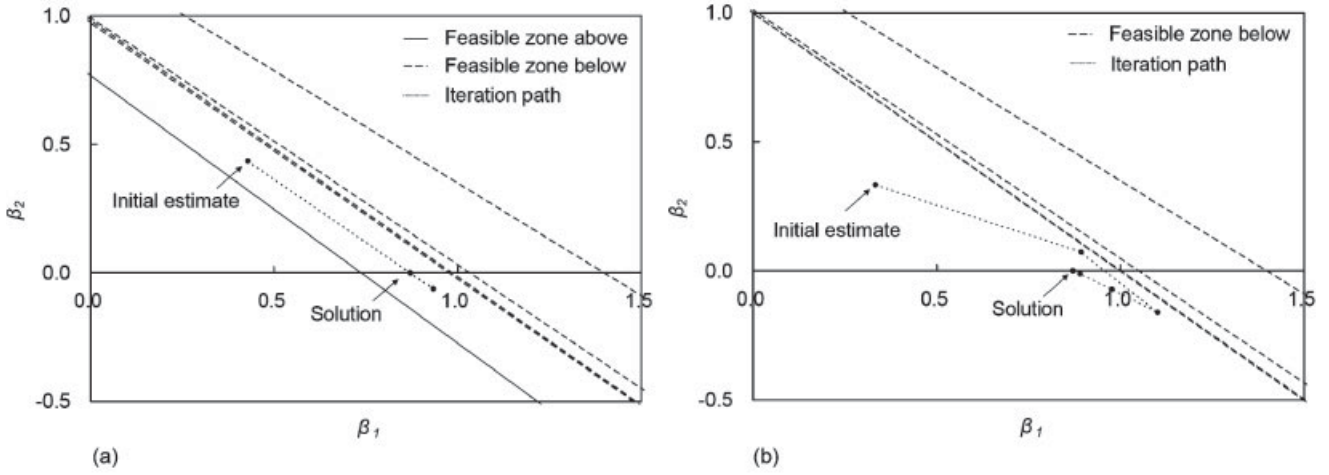

Fig. 10—Iteration path for Example 3 in Table 1. The feasible region is the intersection of the zones above the solid lines and those below the dashed lines. (a) The MS algorithm. (b) The RL algorithm.

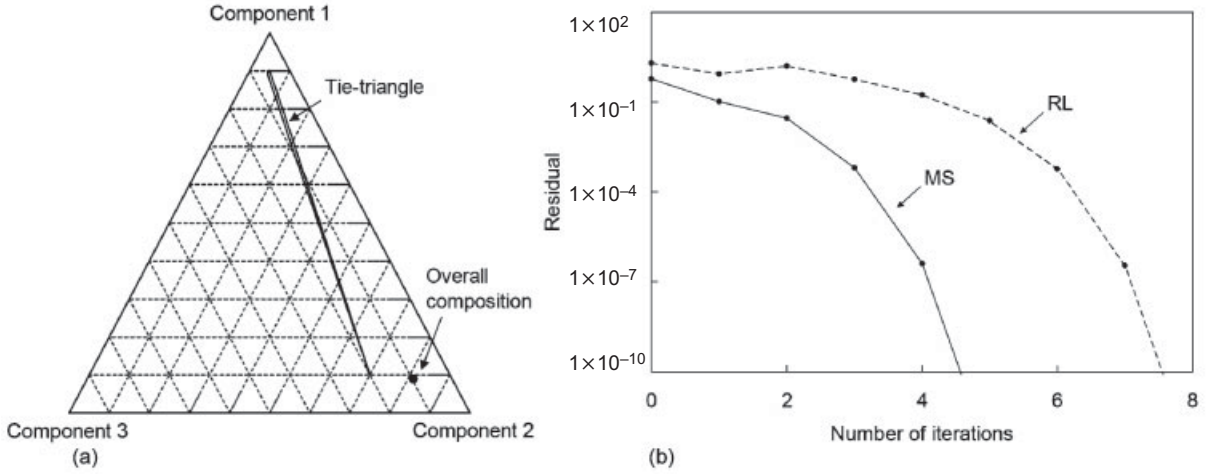

Fig. 11—(a) Ternary diagram showing the tie-triangle and overall composition for the negative flash Example 4 in Table 1. (b) Convergence behaviors of the MS and RL algorithms for Example 4 in Table 1.

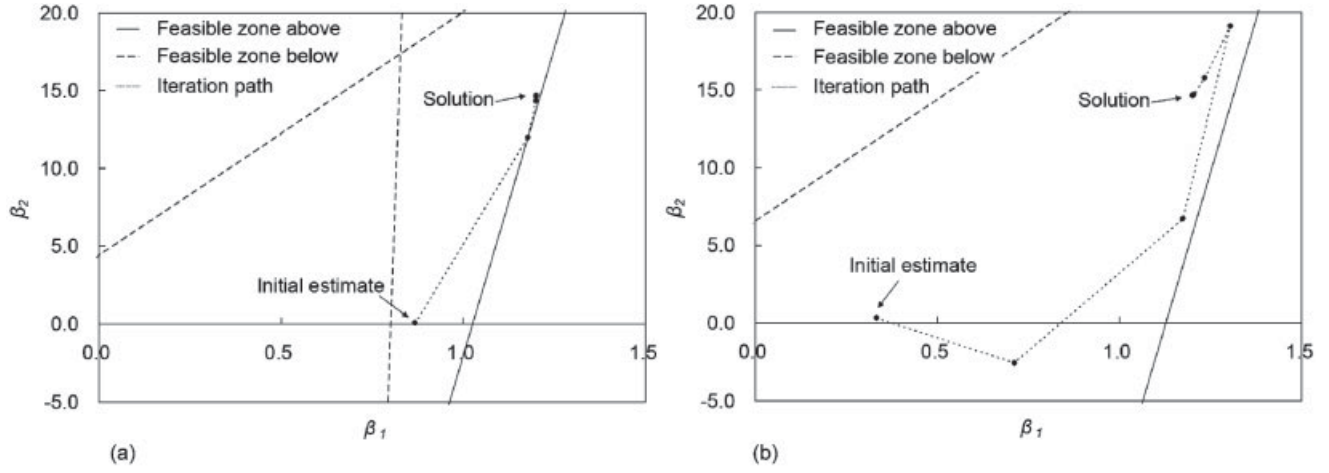

Fig. 12—Iteration path for Example 4 in Table 1. The feasible region is the intersection of the zones above the solid lines and those below the dashed lines. (a) The MS algorithm. (b) The RL algorithm.

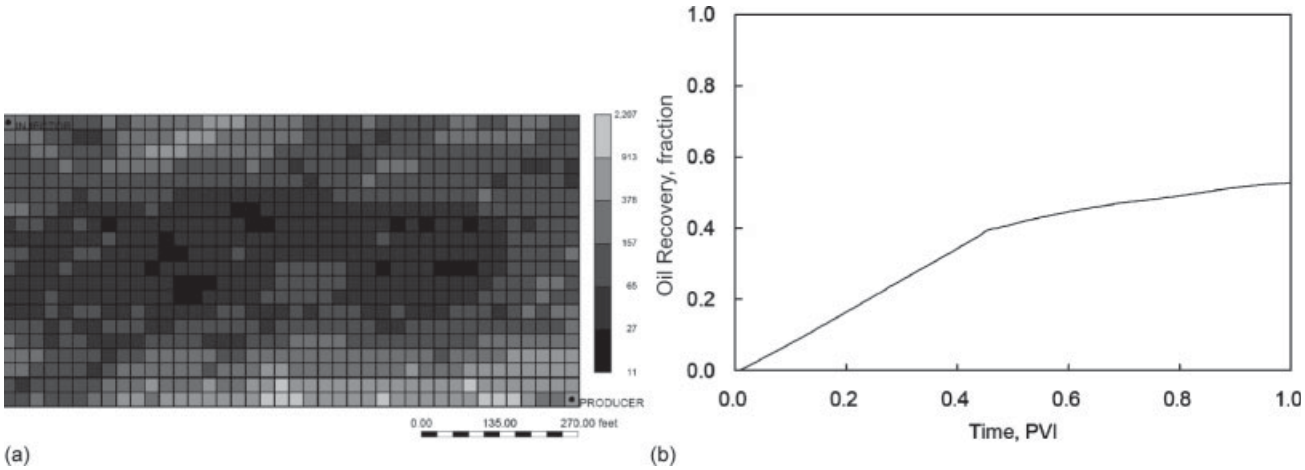

Fig. 13—(a) Randomly generated permeability field in millidarcies. (b) Oil recovery for the simulation case study.

TABLE 2—RESERVOIR PROPERTIES FOR SIMULATION CASE STUDY

Dimensions	1000×500×20 ft
Number of grid cells	40×20×1
Porosity	0.25
ln-mean permeability	102 md
Dykstra-Parsons coefficient	0.58
Correlation length	x = 600 ft, y = 300 ft
Reservoir temperature	105°F
Initial reservoir pressure	1,100 psia
Relative permeability model	Corey
Residual saturation ($W/L_1/G/L_2$*)	0.25/0.20/0.05/0.20
Endpoint relative permeability ($W/L_1/G/L_2$*)	0.21/0.70/0.35/0.35
Exponent ($W/L_1/G/L_2$*)	1.5/2.5/2.5/2.5
Initial saturation ($W/L_1/G/L_2$*)	0.25/0.75/0.0/0.0

* W: aqueous phase; L_1: oleic phase; G: gaseous phase; L_2: CO_2-rich liquid phase.

Three-phase constant-K flash calculations were performed approximately 0.7 million times. There were no failures in those calculations, confirming the robustness of our algorithm for this case. **Table 4** shows the breakdown in computational time for phase equilibrium calculations. Multiphase constant-K flash calculations, which are not shown in the table, take only 0.38% of the total simulation time. The phase equilibrium calculations take 70% of the total simulation time, where 14% is for stability analysis and 56% is for flash calculations. Table 4 also shows that when the method of Young and Stephenson (1983) is not used and stability analysis for a single phase is performed for all single-phase cells, the phase equilibrium calculations take 76% of the total simulation time. Thus, there is a 6% savings in time by not always performing stability calculations in this example. In this example, there is only a small difference in the simulation results using the two methods.

The total simulation time could be significantly increased because of the reduction in automatic timestep sizes that may result when nonconvergence occurs in phase equilibrium calculations (Okuno 2009; Okuno et al. 2009). Simulations where convergence problems exist may not even be possible to complete to required injection times. There are a variety of problems that can occur in compositional simulation, but our algorithm eliminates convergence problems associated with multiphase constant K-flash calculations.

Conclusions

We formulated the constant-K flash calculation with N_P phases as a minimization of a nonmonotonic convex function with N_C linear constraints. The behavior of the minimization function was investigated in detail to develop a robust and practical algorithm that is guaranteed to converge independently of the number of phases for both positive and negative flash calculations. The algorithm was implemented in a standalone flash code and in UTCOMP, a multiphase compositional simulator, where K-values change with composition. The conclusions are as follows:

- Our algorithm is guaranteed to converge because the minimization function is convex and because the small feasible region developed in this research does not contain zones near poles where the Hessian matrix is ill-conditioned.

TABLE 3—FLUID PROPERTIES USED IN SIMULATION CASE STUDY

	Oil (mol %)	Gas (mol %)	Molecular Weight	T_c (°F)	P_c (psia)	Acentric Factor	Binary Interaction Coefficient* CO_2
CO_2	3.37	95.0	44.01	87.90	1069.87	0.225	0
C_1	8.61	5.0	16.04	−116.59	667.20	0.008	0.055
C_{2-3}	15.03	0.0	37.20	159.90	652.56	0.131	0.055
C_{4-6}	16.71	0.0	69.50	374.13	493.07	0.240	0.055
C_{7-15}	33.04	0.0	140.96	630.68	315.44	0.618	0.105
C_{16-27}	16.11	0.0	280.99	892.16	239.90	0.957	0.105
C_{28+}	7.13	0.0	519.62	1236.79	238.12	1.268	0.105

*All others are 0.0.

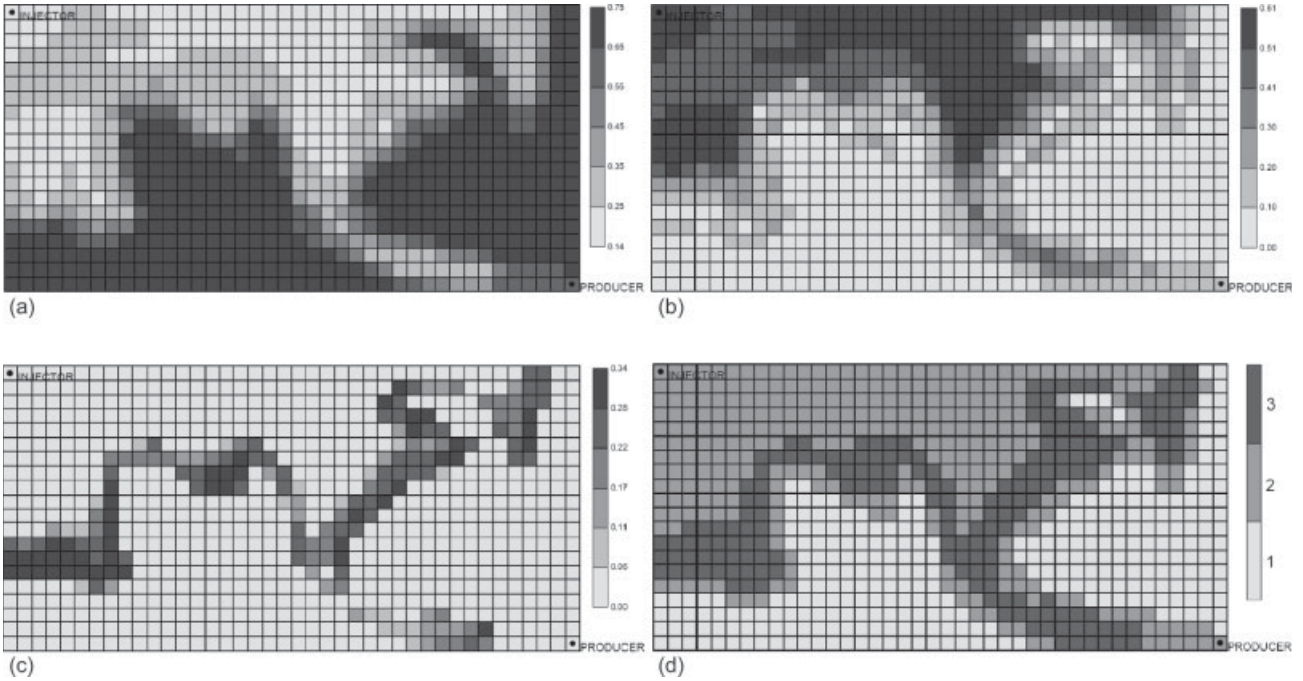

Fig. 14—(a) Oleic-phase distribution at 0.45 PVI. (b) Gaseous-phase distribution at 0.45 PVI. (c) CO_2-rich liquid-phase distribution at 0.45 PVI. (d) Distribution of the number of hydrocarbon phases at 0.45 PVI.

TABLE 4—COMPUTATIONAL TIME REQUIRED FOR SIMULATION CASE STUDY

		Limited Use of One-Phase Stability Analysis		One-Phase Stability Analysis Used for All One-Phase Cells	
		Time (seconds)	Percentage	Time (seconds)	Percentage
Overall simulation		1,829	100	2,209	100
Phase equilibrium		1,277	70	1,671	76
Stability analysis	1 phase	98	5	497	23
	2 phases	161	9	159	7
Flash calculations	2 phases	590	32	587	27
	3 phases	428	23	427	19

- Procedures for multiphase constant-K flash calculations developed before this research were shown not to converge for some situations or to converge slowly because of large feasible regions that are bounded by poles.
- Our algorithm can improve the reliability and efficiency of multiphase compositional simulation as part of rigorous EOS flash calculations. Convergence within our K-flash calculation is not dependent on the initial guess from a prior flash calculation because we ensure that the initial guess is in the feasible region.
- Fewer iterations are required for convergence using our algorithm because the feasible region is significantly smaller than that proposed in prior research. We demonstrated that, even for five equilibrium phases, our algorithm requires only six iterations at most in 1 million flash calculations with different K-values that were generated randomly.

Nomenclature

a = matrix with elements $(1-K_{ij})$ for $i = 1, \ldots, N_C, j = 1, \ldots, N_P-1$
b = vector with elements $\min\{1-z_i, \min_j\{1-K_{ij}z_i\}\}$ for $i = 1, \ldots, N_C, j = 1, \ldots, N_P-1$
d = vector representing a search direction to update independent variables in iterative solution
f_j = jth Rachford-Rice equation
f = vector with elements f_j
F = convex function to be minimized, defined by Eq. 9
H = Hessian matrix
J = Jacobian matrix
K_{ij} = K-value of component i in phase j
L = constraint set: $\{\beta \mid t_i \geq 0, i = 1, \ldots, N_C\}$
L' = constraint set: $\{\beta \mid t_i > 0, i = 1, \ldots, N_C\}$
N_C = number of components
N_P = number of phases
P = constraint set: $\{\beta \mid \beta_j \geq 0, j = 1, \ldots, N_P\}$
P_C = critical pressure
S = constraint set defined by Eq. 10 or phase saturation
T_C = critical temperature
t_i = term defined in Eq. 6
x_{ij} = component mole fraction of component i in phase j
z_i = component mole fraction of component i in a mixture
β_j = jth phase mole fraction
β = phase-mole-fraction vector with elements β_j
λ = step size to update independent variables in iterative solution
ϕ_{ij} = fugacity coefficient of component i in phase j

Superscripts

n = index for iteration steps
T = transpose
-1 = inverse

Subscripts

C = critical property or component
i = component index
j = phase index
k = phase index
max = maximum

Acknowledgments

We thank the Japan Petroleum Exploration Company and the Japan Oil, Gas, and Metals National Corporation for support of this research, along with the member companies of the gasflooding joint-industry project at The University of Texas at Austin. Russell T. Johns holds the B.J. Lancaster Professorship at The University of Texas.

References

Baker, L.E., Pierce, A.C., and Luks, K.D. 1982. Gibbs Energy Analysis of Phase Equilibria. *SPE J.* **22** (5): 731–742. SPE-9806-PA. doi: 10.2118/9806-PA.

Bertsekas, D.P. 1999. *Nonlinear Programming*, second edition. Nashua, New Hampshire: Athena Scientific.

Bünz, A.P., Dohrn, R., and Prausnitz, J.M. 1991. Three-Phase Flash Calculations for Multicomponent Systems. *Computers & Chemical Engineering* **15** (1): 47–51. doi: 10.1016/0098-1354(91)87005-T.

Chang, Y.-B. 1990. Development and Application of an Equation of State Compositional Simulator. PhD dissertation, University of Texas at Austin, Austin, Texas.

Chang, Y.-B., Pope, G.A., and Sepehrnoori, K. 1990. A Higher-Order Finite-Difference Compositional Simulator. *J. Pet. Sci. Eng.* **5** (1): 35–50. doi: 10.1016/0920-4105(90)90004-M.

Dennis, J.E. Jr. and Schnabel, R.B. 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, No. 16. Philadelphia, Pennsylvania: Classics in Applied Mathematics, SIAM.

Eubank, P.T. 2006. Equations and Procedures for VLLE Calculations. *Fluid Phase Equilibria* **241** (1–2): 81–85. doi: 10.1016/j.fluid.2005.11.011.

Haugen, K.B., Sun, L., and Firoozabadi, A. 2007. Three-Phase Equilibrium Calculations for Compositional Simulation. Paper SPE 106045 presented at the SPE Reservoir Simulation Symposium, Houston, 26–28 February. doi: 10.2118/106045-MS.

Khan, S.A., Pope, G.A., and Sepehrnoori, K. 1992. Fluid Characterization of Three-Phase CO_2/Oil Mixtures. Paper SPE 24130 presented at the SPE/DOE Enhanced Oil Recovery Symposium, Tulsa, 22–24 April. doi: 10.2118/24130-MS.

Leibovici, C.F. and Neoschil, J. 1995. A Solution of Rachford-Rice Equations for Multiphase Systems. *Fluid Phase Equilibria* **112** (2): 217–221. doi: 10.1016/0378-3812(95)02797-I.

Leibovici, C.F. and Nichita, D.V. 2008. A New Look at Multiphase Rachford-Rice Equations for Negative Flashes. *Fluid Phase Equilibria* **267** (2): 127–132. doi: 10.1016/j.fluid.2008.03.006.

Michelsen, M.L. 1982a. The Isothermal Flash Problem. Part II. Phase-Split Calculation. *Fluid Phase Equilibria* **9** (1): 21–40. doi: 10.1016/0378-3812(82)85002-4.

Michelsen, M.L. 1982b. The Isothermal Flash Problem. Part I. Stability. *Fluid Phase Equilibria* **9** (1): 1–19. doi: 10.1016/0378-3812(82)85001-2.

Michelsen, M.L. 1994. Calculation of Multiphase Equilibrium. *Computers & Chemical Engineering* **18** (7): 545–550. doi: 10.1016/0098-1354(93)E0017-4.

Michelsen, M.L. and Mollerup, J.M. 2004. *Thermodynamic Models: Fundamentals and Computational Aspects*. Holte, Denmark: Tie-Line Press.

Nelson, P.A. 1987. Rapid Phase Determination in Multiple-Phase Flash Calculations. *Computers & Chemical Engineering* **11** (6): 581–591. doi: 10.1016/0098-1354(87)87004-7.

Nghiem, L.X. and Li, Y.K. 1986. Effect of Phase Behavior on CO_2 Displacement Efficiency at Low Temperatures: Model Studies With an Equation of State. *SPE Res Eng* **1** (4): 414–422. SPE-13116-PA. doi: 10.2118/13116-PA.

Okuno, R, Johns, R.T., and Sepehrnoori, K. 2009. Three-Phase Flash in Compositional Simulation Using a Reduced Method. Accepted for *SPE J*. SPE-125226-PA.

Okuno, R. 2009. Modeling of Multiphase Behavior for Gas Flooding Simulation. PhD dissertation, University of Texas at Austin, Austin, Texas (August 2009).

Peng, D-Y. and Robinson, D.B. 1976. A New Two-Constant Equation of State. *Ind. Eng. Chem. Fund.* **15** (1): 59–64. doi: 10.1021/i160057a011.

Perschke, D.R., Chang, Y.-B., Pope, G.A., and Sepehrnoori, K. 1989. Comparison of Phase Behavior Algorithms for an Equation-of-State Compositional Simulator. Paper SPE 19443 available from SPE, Richardson, Texas.

Rachford, H.H. Jr. and Rice, J.D. 1952. Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium (Technical Note 136). *J. Pet Tech* **4** (10): 19; *Trans.,* AIME, **195:** 327–328. SPE-952327-G.

Wang, X. and Strycker, A. 2000. Evaluation of CO_2 Injection With Three Hydrocarbon Phases. Paper SPE 64723 presented at the International Oil and Gas Conference and Exhibition in China, Beijing, 7–10 November. doi: 10.2118/64723-MS.

Whitson, C.H. and Michelsen, M.L. 1989. The Negative Flash. *Fluid Phase Equilibria* **53** (December): 51–71. doi: 10.1016/0378-3812(89)80072-X.

Wilson, G.M. 1969. A Modified Redlich-Kwong Equation of State, Application to General Physical Data Calculations. Paper 15-C presented at the 65th National AIChE Meeting, Cleveland, Ohio, USA, 4–7 May.

Young, L.C. and Stephenson, R.E. 1983. A Generalized Compositional Approach for Reservoir Simulation. *SPE J.* **23** (4): 727–742. SPE-10516-PA. doi: 10.2118/10516-PA.

Yuan, H. and Johns, R.T. 2005. Simplified Method for Calculation of Minimum Miscibility Pressure or Enrichment. *SPE J.* **10** (4): 416–425. SPE-77381-PA. doi: 10.2118/77381-PA.

Appendix A

Fig. A-1 is a flow chart of multiphase equilibrium calculations in UTCOMP from Chang (1990).

Appendix B—Algorithm for SS

Initial estimates for each grid cell in compositional simulation are determined by either the previous timestep or stability analysis or correlations such as that of Wilson (1969). The steps for SS that we use in this paper are

1. Obtain initial estimates for (N_P-1) sets of N_C K-values.
2. Calculate phase mole fractions and phase compositions by solving (N_P-1) RR equations (see Eq. 8).
3. Calculate compressibility factors and fugacity coefficients for N_P phases using any cubic EOS. In this paper, we use the Peng-Robinson EOS (Peng and Robinson 1976).
4. Check for convergence based on the residuals of the fugacity equations, Eq. 1. If convergence is achieved, stop. Otherwise, continue to Step 5.
5. Update the (N_P-1) sets of K-values based on the fugacity equations using $\ln K_{ij} = \ln \phi_{iN_P} - \ln \phi_{ij}$, where ϕ_{ij} is the fugacity coefficient of component i in phase j.
6. Go to Step 2.

Appendix C—Convexity of the Function To Be Minimized

We give a proof for the convexity of our minimization function, which is similar to the one by Michelsen and Mollerup (2004) for their minimization function. We also provide details for the case when a constant-K flash problem has no solution.

A gradient vector of the function F consists of the RR equations, while the Hessian matrix consists of the derivatives of the RR equations with respect to the independent variables. That is,

$$\nabla F = f = \left\{ f_j \right\} \in \mathbb{R}^{(N_P-1)}$$

and

$$\nabla^2 F = \left\{ H_{kj} \right\} = \left\{ \partial f_j / \partial \beta_k \right\} \in \mathbb{R}^{(N_P-1)\times(N_P-1)},$$

where

$$\partial f_j / \partial \beta_k = \sum_{i=1}^{N_C} \left[(1-K_{ij})(1-K_{ik}) z_i \right] \Big/ t_i^2 = Y^T D Y$$

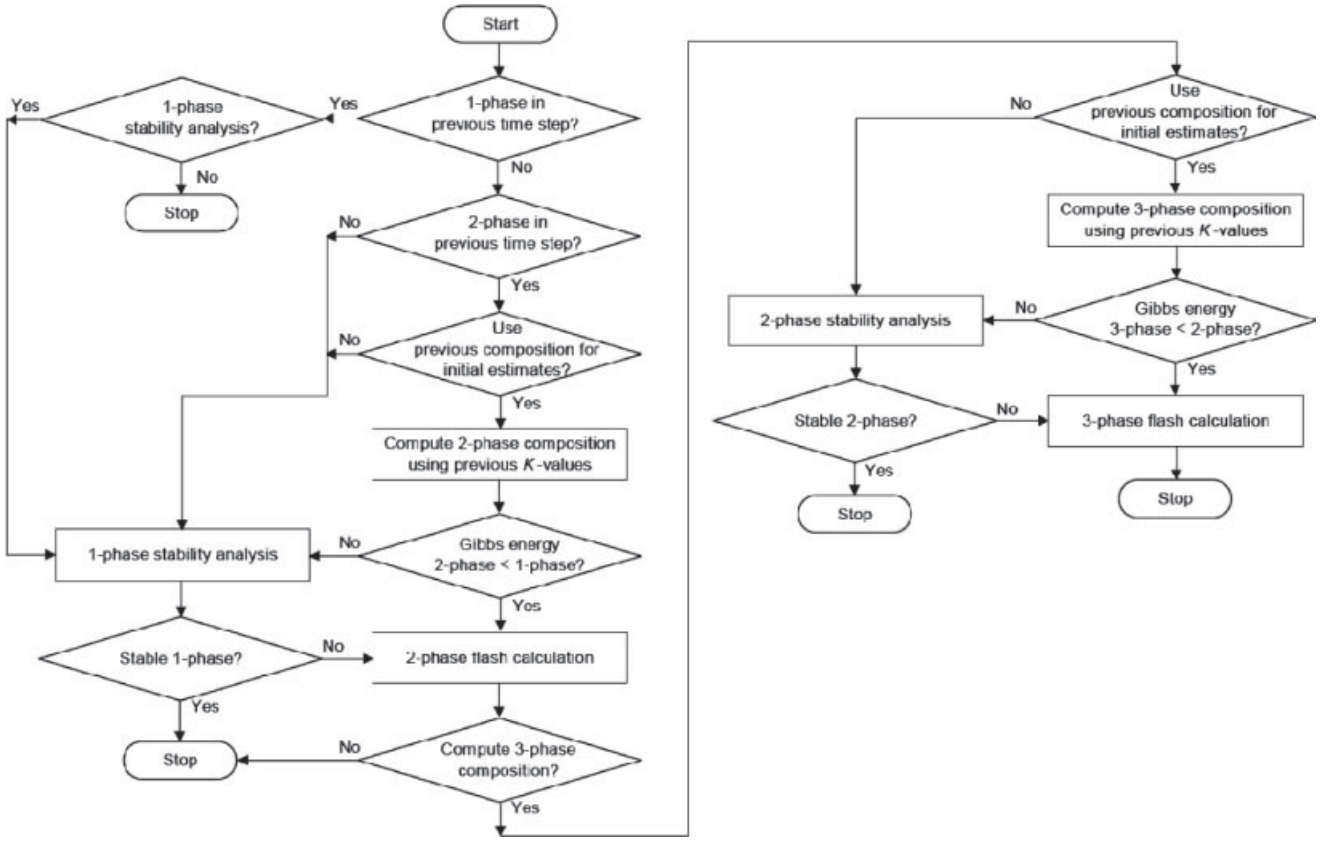

Fig. A-1—Flow chart of multiphase equilibrium calculations in UTCOMP.

and

$$D = \mathrm{diag}\left(z_1,...,z_{N_C}\right) \in \mathbb{R}^{N_C \times N_C}$$

$$Y = \left\{Y_{ij}\right\} = \left\{\left(1-K_{ij}\right)/t_i\right\} \in \mathbb{R}^{N_C \times (N_P-1)}.$$

The matrix D is positive definite because z_i (i = 1, …, N_C) are all positive.

Using the Gibbs phase rule, the degree of freedom in an isobaric- and isothermal-flash calculation is (N_C–N_P). Because the degree of freedom must be nonnegative, $N_P \le N_C$ for problems considered in this research. In addition, if the $N_C \times (N_P-1)$ matrix Y is of full rank, we have $Yy \neq 0$ for any vector $y \neq 0$.

Considering the positive definiteness of D, we can prove the positive definiteness of the Hessian matrix Y^TDY as follows:

$$y^T\left(Y^TDY\right)y = \left(Yy\right)^T D(Yy) > 0 \quad \text{for all vector } y.$$

The proof holds for both positive and negative flash calculations where one or more phase mole fractions are negative. However, it does not hold for another kind of negative flash where an overall composition lies in negative composition space (i.e., D is not positive definite). We do not consider this type of negative flash in this paper.

If Y is not of full rank, the Hessian matrix is only positive semidefinite and there exists a vector along which the function value does not change. To see this, suppose that the pth and qth columns of the matrix Y or a $N_C \times (N_P-1)$ matrix $\{1-K_{ij}\}$ are linearly dependent. Then, there exists a constant α such that

$$\left(1-K_{ip}\right) = \alpha\left(1-K_{iq}\right) \text{ for all } i.$$

Then, the pth and qth columns of the Hessian matrix become linearly dependent as shown below.

$$H_{kp} = \partial f_p/\partial\beta_k = \sum_{i=1}^{N_C}\left[\left(1-K_{ip}\right)\left(1-K_{ik}\right)z_i\right]/t_i^2$$
$$= \sum_{i=1}^{N_C}\left[\alpha\left(1-K_{iq}\right)\left(1-K_{ik}\right)z_i\right]/t_i^2 = \alpha H_{kq}.$$

The Hessian matrix, therefore, is only positive semidefinite for this case. This is also confirmed by the function values that are constant along a vector d with $(1-K_{1q})$ for pth, $-(1-K_{1p})$ for qth, and zeros for the other elements. That is,

$$d^t\nabla F = \left(1-K_{1q}\right)\left[\sum_{i=1}^{N_c}\left(1-K_{ip}\right)z_i/t_i\right]$$
$$-\left(1-K_{1p}\right)\left[\sum_{i=1}^{N_c}\left(1-K_{iq}\right)z_i/t_i\right]$$
$$= \sum_{i=1}^{N_C}\left(z_i/t_i\right)\left[\alpha\left(1-K_{1q}\right)\left(1-K_{iq}\right)-\alpha\left(1-K_{1q}\right)\left(1-K_{iq}\right)\right] = 0.$$

Appendix D—Line-Search Algorithm

The line search minimizes the following function over the region $0 \le s \le 1$:

$$g(s) = F\left(\beta + s\lambda_{\max}d\right).$$

Considering the convexity of the function, if the gradient is nonpositive at the initial point $s = 1$, we do not perform the line search taking $s = 1$ as a solution. The procedure is

1. Set an initial estimate for s to unity and an iteration index n to zero.

2. Calculate the first-order derivative of g as shown next. If the absolute value of the first-order derivative is less than a specified tolerance, then stop. Otherwise, continue to Step 3.

$$\left(dg/ds\right)^n = \lambda_{\max}\left[\nabla F\left(\beta + s^n\lambda_{\max}d\right)\right]^T d.$$

3. Calculate the second-order derivative of g.

$$\left(d^2g/ds^2\right)^n = \left(\lambda_{\max}\right)^2 d^T\left[\nabla^2 F\left(\beta + s^n\lambda_{\max}d\right)\right]d.$$

4. Update the iteration variable s.

$$s^{n+1} = s^n - \left(dg/ds\right)^n \Big/ \left(d^2g/ds^2\right)^n.$$

5. Let $n = n + 1$, and go back to Step 2.

Conversion Factors

ft × 3.048*	E–01 = m
°F × (°F–32)/1.8	= °C
psi × 6.894 757	E+00 = kPa

*Conversion factor is exact.

Ryosuke Okuno is a reservoir engineer with the Japan Petroleum Exploration Company in Tokyo. His research interests include modeling of gasflooding, compositional reservoir simulation, and phase-behavior modeling. Okuno holds BS and MS degrees in geosystem engineering from The University of Tokyo and a PhD degree in petroleum engineering from The University of Texas at Austin. **Russell T. Johns** is the B.L. Lancaster Professor in the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin. His research interests include theory of gas-injection processes, multiphase flow in porous media, and well testing. Johns holds a BS degree in electrical engineering from Northwestern University and MS and PhD degrees in petroleum engineering from Stanford University. **Kamy Sepehrnoori** is the Bank of America Professor in the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin. His research and teaching interests include computational methods, reservoir simulation, parallel computations, enhanced-oil-recovery modeling, and inverse modeling. Sepehrnoori holds a PhD degree from the University of Texas at Austin.