The Isothermal Flash Problem: New Methods for Phase Split Calculations

Various numerical approaches have been adopted to solve the single-stage isothermal flash problem. These approaches result in two classes of methods. The first includes equation-solving methods that try to solve a nonlinear equation system; the second is based on a minimization of the total Gibbs free energy. Most of these methods may fail to find a solution or may lead to erroneous solutions near critical conditions when an equation of state is applied to both the vapor and liquid phases.

New methods for solving the problem are proposed. Combining the simplicity in structure of the conventional successive-substitution method and the efficiency of some unconstrained minimization algorithms, they all ensure convergence to local minima of the Gibbs free energy. The new methods are compared from the standpoints of computer storage and calculational effort requirements. The performance of these methods is tested on four multicomponent systems taken from literature, and a comparison is made with other published methods.

M. N. Ammar and H. Renon
Centre Réacteurs & Processus
École Nationale Supérieure des Mines de Paris
75006 Paris, France

Introduction

The isothermal multiphase flash problem is one of the most important engineering problems encountered in modeling chemical and petroleum processes. The aim is to predict for a multicomponent system at fixed pressure and temperature, the number, identity, and composition of phases at equilibrium conditions.

Basically, this problem may be formulated as the search for the global minimum of the total Gibbs free energy of the system under the specified conditions. The commonly used approach for solving this problem consists first in fixing the number of phases, then in setting a thermodynamic model for the Gibbs free energy of each phase, and finally using either a minimization technique or an equation-solving method to find the solution. This generally leads, in case of convergence, to a local minimum rather than to the global one. In order to find the right solution, one needs a technique for adding or removing phases, and then one solves similar problems with a different phase configuration whenever a lower local minimum in the Gibbs-energy hypersurface may be located. Such a technique may be viewed as a stability analysis of the system.

As a first step, one is involved in solving the two-phase vapor-liquid flash. The computational problems that occur in systems at high pressures and when the same thermodynamic model (an equation of state) is applied to both phases, are mainly associated with the existence of critical points that may result in the possibility of convergence to trivial solutions while nontrivial ones exist, or in the slowness or failure of some algorithms to reach the solution. In some key respects, liquid-liquid equilibrium calculations at any pressure are subject to the same difficulties.

An efficient computational scheme is introduced in this paper. Based on a minimization of the Gibbs free energy, new methods are designed and convergence is always to a local minimum representing the physical solution. What follows is a sort of a unified approach taking advantage of the simplicity of structure in some equation-solving methods and the efficiency and robustness of some unconstrained minimization methods.

The successive-substitution method in the conventional scheme for solving the isothermal vapor-liquid flash problem has been identified as a gradient method of constant step length for minimizing the global Gibbs free energy of a multicomponent system. Also, other minimization methods have been tested and their performance compared. These include the steepest-descent method, the conjugate-gradients method, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method (Fletcher, 1980), and the second-order Newton method. Special care has been taken in initializing the different schemes to ensure efficiency and convergence and to avoid the trivial solution whenever a vapor-liquid solution exists, principally in the immediate vicinity of a critical point.

The performance of all these new methods on four test examples at different conditions of pressure and temperature is considered in terms of CPU execution time. A comparison is also made with other published methods, and the advantages over some of these of an optimal strategy for solving the problem have been demonstrated. Either the Mathias (1983) or the Soave (1972) Redlich-Kwong equation of state is used as a thermodynamic model for both the vapor and liquid phases, but one could apply as well any other thermodynamic model in conjunction with the algorithms considered.

Prior Work

Heidemann (1983) and Grens (1983) have reviewed most approaches adopted to solve the isothermal vapor-liquid flash problem. Although all approaches try to solve the same problem, they may differ either in the way of formulating it or in the numerical technique used to reach the solution. From the problem-formulation point of view, we can classify all the methods as one of two types: type A methods, or the equation-solving approach, and type B methods, or the Gibbs free energy minimization approach. It is worth noting that this split is in some way arbitrary as far as some methods of type A can be numerically identified as part of type B methods.

Methods of type A basically try to solve a set of nonlinear descriptive equations using an appropriate numerical technique. Two subclasses can be distinguished within this type. Methods of the first subclass adopt the global approach, which consists in solving simultaneously the whole set of descriptive equations in terms of all the problem unknowns. Hirose et al. (1978) and Asselineau et al. (1979) proposed the Newton-Raphson algorithm. Particular care has to be taken to prevent the case of singularity of the Jacobian matrix in both methods and to avoid the trivial solution with phases of equal properties, when an equation of state is used as in Asselineau et al. One also may notice the poor performance of the Newton-Raphson algorithm when starting from a poor initialization. Quadratic convergence is achieved only in a small neighborhood of the solution. Joulia et al. (1986) tested a hybrid method using a quasi-linearization procedure in a first step, and a quasi-Newton algorithm based on Schubert (1970) update formulae in a second step. This method conserves the sparsity form of the Jacobian matrix and leads at best to a superlinear rate of convergence. As noted by the authors, problems may be encountered close to the phase boundary in the critical region with this method, owing to the empirical switch test adopted.

Methods of the second-subclass consider a reduced set of equations and unknowns. The common feature of these methods is that the material balance equations are satisfied at each iteration. Most of the methods within this subclass consider the equations in terms of the equilibrium ratios. One of the earliest methods is the successive-substitution method, whose implementation has been discussed by several authors: Null (1970), Prausnitz et al. (1980), Anderson and Prausnitz (1980), Michelsen (1982b), Nghiem et al. (1983), Mehra et al. (1983). The successive-substitution method may be viewed as a search for a fixed point of an equilibrium ratios function. Thus, it has a linear rate of convergence, and convergence becomes slower as the specifications get closer to a critical point when an equation of state is applied to both phases (Michelsen, 1982b). In order to accelerate convergence, different methods have been applied. Boston and Britt (1978) as part of their "inside-out" algorithm used Broyden's (1965) method to update the equilibrium ratios between the successive iterations. Mehra et al. (1983) reported improvement in convergence speed by adjusting step length according to different quasi-Newton approximations. Nghiem and Heidemann (1982) have shown that this method is equivalent to one of Broyden's rank 1 algorithms for solving simultaneous equations. Michelsen (1982b) used the general dominant eigenvalue method of Crowe and Nishio (1975), one step each five iterations to accelerate convergence in the successive-substitution scheme. He finished occasionally by the Newton-Raphson method in difficult cases. The equation system considered by Michelsen (1982b) leads to savings in computer memory storage and calculational effort as far as only symmetric positive definitive matrices are involved.

Another implementation of the Newton-Raphson method has been considered by Fussell and Yanosik (1978) as part of their minimum variable Newton-Raphson (MVNR) algorithm. Although it reduces to the same number of independent variables, it is computationally rather uneconomical compared to Michelsen's implementation. Besides, Newton methods are known to perform well only in a small neighborhood of the solution. Nghiem et al. (1983) used Powell's (1970) hybrid method with different sets of independent variables after a certain progress has been made by the successive-substitution method. Empirical criteria have been set for the switch between the two algorithms. This method behaves at best like a quasi-Newton one near the solution. It is worthwhile to note, as Michelsen (1982b) established, that converged solutions from the successive-substitution method are local minima of the Gibbs free energy. Also, Mehra et al. (1983) have shown the relationship between the successive-substitution method and a method of steep descent in minimizing the Gibbs free energy. This is very important progress, for most of the other methods discussed try only to locate stationary points by solving the related equilibrium equations; whether these are minima, maxima, or saddle points is rarely considered.

Methods of type B try to solve the isothermal vapor-liquid flash problem in its basic thermodynamic form. One has to find the vapor and liquid molar compositions that correspond to the minimum of the Gibbs free energy. This formulation, based on a linearly constrained minimization, is rather recent. Gautam and Seider (1979) and Ohanomah and Thompson (1984) used the RAND method to solve the problem. This consists of developing a quadratic approximation of the Gibbs energy around a current point, setting the stationary point conditions for the quadratic problem with Lagrange multipliers for the equality constraints, then solving a system of linear equations to find increments for the whole set of variables. Any composition dependency of the fugacity coefficients is neglected in this scheme. It has been established by Ammar (1986) that convergence characteristics are the same as those of the successive-substitution method, although the RAND method performs more computations per iteration. Ohanomah and Thompson reported failure in using the RAND method with some flash problems and concluded that it was unreliable. Lucia et al. (1985) obtained very encouraging results and showed that symmetric and thermodynamically consistent quasi-Newton formulae, used in the context of a hybrid optimization algorithm, can compete favorably with the BFGS formulae and Newton's method. The optimization procedure considered by Lucia et al. operates on the full-space variables.

It is obvious that methods of type B, even if not as numerous as those of type A, have the advantage of considering a more rigorous basis to solve the isothermal flash problem.

Problem Formulation

The basic thermodynamic form of the two-phase vapor-liquid flash problem is to find the minimum of the total Gibbs free energy of the system at the given pressure and temperature:

$$\underset{\substack{v_i, l_i \\ i=1,\ldots,N}}{\text{Min}}\; G = \left(\sum_{i=1}^{N} v_i \mu_i^V + l_i \mu_i^L \right) \tag{1}$$

under the material balance constraints

$$l_i + v_i = z_i \quad i = 1, \ldots, N \tag{2}$$

and the inequality constraints

$$0 \le v_i \le z_i \quad i = 1, \ldots, N \tag{3}$$

where G is the molar Gibbs free energy of the system, z_i is the mole fraction of component i in the system, and v_i, l_i are mole numbers of component i in the vapor and liquid phases per mole of the feed. If the chemical potentials μ_i^L and μ_i^V are expressed in terms of fugacity coefficients and using $\underline{v}$ as independent variables, the problem reduces to the following constrained minimization:

$$\underset{\substack{v_i \\ i=1,\ldots,N}}{\text{Min}}\; \Delta g = \frac{(G - G^\circ)}{RT} = \sum_{i=1}^{N} v_i \ln\left(y_i \frac{P}{P^\circ} \Phi_i^V \right) + l_i \ln\left(x_i \frac{P}{P^\circ} \Phi_i^L \right) \tag{4}$$

with

$$0 \le v_i \le z_i \quad i = 1, \ldots, N \tag{5}$$

where

$$G^\circ = \sum_{i=1}^{N} z_i \mu_i^\circ$$

is the molar Gibbs free energy of the system at the standard state, μ_i° $i = 1, \ldots, N$ are standard chemical potentials, and l_i, x_i, y_i, Φ_i^L, and Φ_i^V are considered functions of $\underline{v}$ through Eq. 2.

When using a minimization algorithm, one may need to calculate the gradient vector and occasionally the Hessian matrix of Δg. These quantities could be evaluated from:

$$\underline{\nabla}(\Delta g) = \underline{g} = \left(\frac{\partial \Delta g}{\partial v_i} \right) = \ln\left(\frac{y_i \Phi_i^V}{x_i \Phi_i^L} \right) \quad i = 1, \ldots, N \tag{6}$$

$$\underline{\underline{\nabla}}^2 (\Delta g) = \underline{\underline{H}} = \left(\frac{\partial^2 \Delta g}{\partial v_i \, \partial v_j} \right) = \underline{\underline{A}} + \underline{\underline{Q}}$$

where

$$\underline{\underline{A}} = \frac{\partial \ln K_i}{\partial v_j} \quad i = 1, \ldots, N \quad j = 1, \ldots, N \tag{7}$$

$$\underline{\underline{Q}} = \frac{\partial \ln \Phi_i^V}{\partial v_j} + \frac{\partial \ln \Phi_i^L}{\partial l_j} \quad i = 1, \ldots, N \quad j = 1, \ldots, N \tag{8}$$

$K_i = y_i/x_i$, $i = 1, \ldots, N$ are the equilibrium ratios.

It is shown that

$$\underline{\underline{A}} = \left(\frac{\delta_{ij}}{v_i} + \frac{\delta_{ij}}{l_i} - \frac{1}{V} - \frac{1}{L} \right) = \frac{1}{VL} \left(-1 + \delta_{ij} \frac{z_i}{x_i y_i} \right) \quad i = 1, \ldots, N \quad j = 1, \ldots, N \tag{9}$$

is a $N \times N$ symmetric positive semidefinite matrix. It is definite for points where $\underline{x} \neq \underline{y}$ inside the two-phase region. In this case, the inverse $\underline{\underline{A}}^{-1}$ is known analytically (see the Appendix).

$$\underline{\underline{A}}^{-1} = VL \left[\frac{x_i y_i}{z_i} \left(\delta_{ij} + \frac{\dfrac{x_i y_i}{z_i}}{S} \right) \right] \quad i = 1, \ldots, N \quad j = 1, \ldots, N \tag{10}$$

where

$$S = 1 - \sum_{i=1}^{N} \frac{x_i y_i}{z_i} \quad V = \sum_{i=1}^{N} v_i \quad L = \sum_{i=1}^{N} l_i$$

Approach to Solving the Problem

In regard to the formulation of Eqs. 4 and 5, it seems very attractive to apply an unconstrained minimization algorithm by keeping the variables vector $\underline{v}$ inside the convex constraints domain, Eq. 5, during the search for the solution. But unless care is taken, some algorithms can lead to a violation of the constraints of Eq. 5 at the earliest iterations, when the initialization is far from the solution, even if the latter lies inside the vapor-liquid region. Setting variables at bounds, which acts as a projection on the constraints domain, can not be justified numerically with some algorithms and results in a singularity of matrix $\underline{\underline{A}}$ and in erratic behavior of the subsequent calculations.

To overcome these difficulties, we have adopted a hybrid approach starting with a robust algorithm to ensure a certain progress from poor initializations, and ending with a more efficient one to accelerate convergence. The switch to the second algorithm takes place once we are sure that the constraints of Eq. 5 could never be violated. The first algorithm, which is of a gradient type, allows projection in case of constraints violation, and acts as a stability analysis similar to that of Baker et al. (1982) and Michelsen (1982a). An accurate initialization is available after this first step, and convergence is easily achieved with a more efficient minimization algorithm.

Gradient-type Algorithms

Among the algorithms we can use to solve Eqs. 4 and 5, let us discuss first those of the gradient type. These algorithms perform badly when the problem is ill conditioned. Conditioning is related to the Hessian matrix and requires the definition of new variables in terms of which the new Hessian matrix is closer to the identity matrix $\underline{\underline{I}}_N$, Fletcher (1980). This can be formally done by using the Cholesky decomposition of the known matrix $\underline{\underline{A}}$,

$$\underline{\underline{A}} = \underline{\underline{U}}^T \underline{\underline{U}} \tag{11}$$

then defining the new variables $\underline{X}$ as

$$\underline{X} = \underline{\underline{U}}\,\underline{v} \tag{12}$$

where $\underline{\underline{U}}$ is an upper-triangular matrix with $U_{i,i} > 0$ $i = 1, \ldots, N$. One can now express the gradient vector and the Hessian matrix in terms of $\underline{X}$:

$$\underline{\nabla}_x(\Delta g) = \frac{\partial \Delta g}{\partial X_i} = \underline{\underline{U}}^{-T}\underline{g} \quad i = 1, \ldots, N \tag{13}$$

$$\underline{\underline{\nabla}}_x^2(\Delta g) = \frac{\partial^2 \Delta g}{\partial X_i\, \partial X_j}$$

$$= \underline{\underline{U}}^{-T}\underline{\underline{H}}\,\underline{\underline{U}}^{-1} = \underline{\underline{I}}_N + \underline{\underline{U}}^{-T}\underline{\underline{Q}}\,\underline{\underline{U}}^{-1} \quad \begin{matrix} i = 1, \ldots, N \\ j = 1, \ldots, N \end{matrix} \tag{14}$$

the iteration procedure in the preconditioned gradients scheme may be reset in terms of $\underline{v}$ as:

$$\underline{v}^{(k+1)} = \underline{v}^{(k)} - \lambda \underline{\underline{A}}^{-1}\underline{g} \tag{15}$$

where λ is the step length at the kth iteration.

When one neglects the composition dependence of the fugacity coefficients Φ_i^V and Φ_i^L ($\underline{\underline{Q}} \approx \underline{\underline{Q}}_N$), Eq. 15 acts as a Newton step with $\lambda = 1$. With a line search to fix λ, this method is similar to that of Gautam and Seider (1979) and Ohanomah and Thompson (1984) using the RAND algorithm. Although the latter is presented as a second-order method in minimizing G, it is rather of a gradient type.

Now, if the second-order terms in the Taylor expansion of $\underline{\ln K}$ in $\underline{v}$ are neglected, we can exploit the fact that

$$\Delta\, \underline{\ln K} \approx \underline{\underline{A}}\,\Delta \underline{v} \tag{16}$$

to define new schemes using $\underline{\ln K}$ as variables instead of $\underline{v}$. This leads to the conventional scheme:

1. Start with $\underline{\ln K}^{(k)}$'
2. Solve the equation

$$\sum_{i=1}^{N} \frac{z_i\,(K_i - 1)}{\beta\,(K_i - 1) + 1} = 0 \tag{17}$$

for the vapor fraction $\beta = \Sigma_{i=1}^{N} v_i$

3. Compute $\underline{x}$ and $\underline{y}$ from

$$x_i = \frac{z_i}{\beta\,(K_i - 1) + 1} \quad y_i = K_i x_i \quad i = 1, \ldots, N \tag{18}$$

4. Calculate $\Delta g^{(k)}$, $\underline{g}^{(k)}$ and test for convergence $\|\underline{g}^{(k)}\| < \epsilon$

5. Define

$$\underline{\ln K}^{(k+1)} = \underline{\ln K}^{(k)} - \lambda \underline{g}^{(k)} \tag{19}$$

fix λ and go to step 2.

The Rachford-Rice (1952) form of Eq. 17 gives a monotonic decreasing function of β. Consequently, there is a unique root in the range [0, 1] if the conditions

$$\left(\sum_{i=1}^{N} K_i z_i\right) - 1 > 0 \quad \text{and} \quad 1 - \left(\sum_{i=1}^{N} \frac{z_i}{K_i}\right) < 0 \tag{20}$$

are satisfied. In this case, Eq. 17 is solved by the Newton procedure and the initialization is done with the converged value of β from the previous iteration. When the conditions of Eq. 20 are not satisfied, β is set equal to zero or one depending on the sign of the real root of Eq. 17. This corresponds to a projection of the independent variables $\underline{v}$ on the convex constraints domain, Eq. 5.

The scheme of steps 1–5 reduces to the classical successive-substitution method (SSM) when $\lambda = 1$ in step 5. Hence, we have proved that the SSM algorithm is a gradient method in minimizing (Δg) of fixed step-length. Converged values from this algorithm correspond to a local minimum of Δg. Michelsen (1982b) and Mehra et al. (1983) gave a different proof of this result.

The SSM has a linear rate of convergence that is related to the condition number of the Hessian matrix. SSM convergence becomes slower as a critical point is approached, and many hundreds of iterations may be required to reach the solution in this region when an equation of state is applied for both the vapor and liquid phases. Indeed, the matrix $\underline{\underline{A}}$ tends to be singular and conditioning does not act any more close to a critical point.

Mehra et al. (1983) noted a convergence rate improvement when λ was fixed in (v) by a quasi-Newton approximation. This quasi-Newton successive-substitution (QNSS) scheme has the same convergence characteristics as a quasi-Newton algorithm with no matrix manipulation (Heidemann, 1983).

In spite of a certain convergence rate deficiency near the critical conditions, both the SSM and QNSS algorithms are stable in performing a good progress in the Gibbs free-energy hypersurface without violating the constraints of Eq. 5 when the solution lies inside the two-phase region. Hence, they are good starting algorithms that can provide a second minimization algorithm with an accurate initialization. The switch to the second algorithm occurs when the Gibbs free energy becomes lower than that of a hypothetical homogeneous phase having the same composition and at the same conditions as the global system. Any minimization method assuring the descent property will consequently converge to the two-phase solution, which is actually a local minimum in the Gibbs free-energy hypersurface.

New Methods for Flash Calculations

New methods for flash calculations can be defined, and they all start either with SSM or QNSS. These methods will differ only in the minimization algorithm used in the second step. Also, all the new methods will keep the same scheme structure as in steps 1–5 and only the update form of $\underline{\ln K}$ in (v) will differ from one method to another.

Newton methods

Two methods can be defined, namely FLASH1 and FLASH2, and will start respectively with the QNSS and the

SSM algorithms. The Newton step in the second algorithm:

$$\begin{aligned} &\underline{\underline{H}}^{(k)}\,\underline{s}^{(k)} = -\underline{g}^{(k)} \\ &\underline{p}^{(k)} = \underline{\underline{A}}\,\underline{s}^{(k)} \\ &\underline{\ln K}^{(k+1)} = \underline{\ln K}^{(k)} + \underline{p}^{(k)} \end{aligned} \tag{21}$$

is determined from a Cholesky decomposition of the Hessian matrix $\underline{\underline{H}}^{(k)}$ as far as only symmetric, positive definite matrices are involved close to a local minimum. This requires about N^2 operations per matrix inversion.

The implementation of the Newton algorithm is similar to that of Michelsen (1982b) and leads to a quadratic convergence rate near the solution. Also, the converged Hessian matrix can give the solution sensitivity to some parameters, such as the pressure or the temperature, so that computations can be initialized accurately near converged points. For example, at a fixed temperature T, when varying the pressure from P_1 to P_2, a K_i initialization of the problem at P_2 is available from the converged K_i at P_1 by:

$$\underline{\ln K}(P_2) = \underline{\ln K}(P_1) - \Delta P\, \underline{\underline{A}}^{(1)}\, [\underline{\underline{H}}^{(1)}]^{-1} \left[\frac{\partial \underline{g}^{(1)}}{\partial P}\right] \tag{22}$$

where

$$\Delta P = P_2 - P_1 \tag{23}$$

Quasi-Newton methods

One can terminate with the QNSS algorithm as far as it behaves like a quasi-Newton method at the end of calculations. But one can also use the BFGS minimization algorithm (Fletcher, 1980), starting with the SSM algorithm. This method is called FLASH3. The approximant $\underline{\underline{B}}$ to the inverse of the Hessian matrix is set equal to $\underline{\underline{A}}^{-1}$ at the end of the first algorithm (SSM) and is updated by BFGS formula during the subsequent iterations. All matrices $\underline{\underline{B}}$ are definite since $\underline{\underline{A}}^{-1}$ is positive definite. Hence, the descent property is ensured and convergence is always to local minima. The BFGS step can be written as:

$$\begin{aligned} &\underline{s}^{(k)} = -\underline{\underline{B}}^{(k)}\,\underline{g}^{(k)} \\ &\underline{p}^{(k)} = \underline{\underline{A}}\,\underline{s}^{(k)} \\ &\underline{\ln K}^{(k+1)} = \underline{\ln K}^{(k)} + \lambda \underline{p}^{(k)} \end{aligned} \tag{24}$$

where the step length λ can be set from a partial line-search algorithm. BFGS is insensitive to the accuracy of the line search, and the values of 0.01 and 0.9 for ρ and σ (Fletcher, 1980) are adopted, leading to a weak line search.

This method has a superlinear rate of convergence at the end. Also, the converged approximant to the inverse of the Hessian matrix is very close to the real one, so that FLASH3 has the same feature as FLASH1 and FLASH2 under the circumstances where the initiation procedure is performed near converged points.

Conjugate-gradients methods

These methods are known to be very sensitive to both conditioning or variables scaling and to the accuracy of the line search. Conditioning requires definition of the minimization problem in terms of $\underline{X}$ instead of $\underline{v}$ as in Eq. 11. A new method, FLASH4, can be defined starting with SSM and finishing with the conjugate-gradients algorithm whose step can be written as:

$$\begin{aligned} &\underline{p}^{(k)} = \underline{\underline{A}}\,\underline{s}^{(k)} \\ &\underline{\ln K}^{(k+1)} = \underline{\ln K}^{(k)} + \lambda \underline{p}^{(k)} \end{aligned} \tag{25}$$

where the step length λ is computed by an accurate line search with ρ and σ equal to 0.01 and 0.05, respectively (Fletcher, 1980). The step vector $\underline{p}^{(k)}$ is set equal to $-\underline{g}^{(k)}$ at the first iteration of the second algorithm and is updated by the conjugacy conditions on the descent directions $\underline{s}^{(k)}$:

$$\underline{p}^{(k+1)} = -\underline{g}^{(k+1)} + \nu^{(k)}\underline{p}^{(k)} \tag{26}$$

Either the Fletcher-Reeves (FR) or the Polak-Ribiere (PR) version of the conjugate-gradients (CG) method (Fletcher, 1980) can be used for $\nu^{(k)}$. In the former (CGFR) we use

$$\nu^{(k)} = \frac{\underline{g}^{(k+1)T}\,\underline{\underline{A}}^{-1}\,\underline{g}^{(k+1)}}{\underline{g}^{(k)T}\,\underline{\underline{A}}^{-1}\,\underline{g}^{(k)}} \tag{27}$$

while the formula

$$\nu^{(k)} = \frac{(\underline{g}^{(k+1)} - \underline{g}^{(k)})^T\,\underline{\underline{A}}^{-1}\,\underline{g}^{(k+1)}}{\underline{g}^{(k)T}\,\underline{\underline{A}}^{-1}\,\underline{g}^{(k)}} \tag{28}$$

is used in the latter (CGPR).

Steepest-descent method

This method, FLASH5, consists in using the gradient method as a second algorithm with a partial line search to fix the step length. It starts with SSM, and the step in the second algorithm can be expressed by:

$$\begin{aligned} &\underline{s}^{(k)} = -\underline{\underline{A}}^{-1}\underline{g}^{(k)} \\ &\underline{p}^{(k)} = \underline{\underline{A}}\underline{s}^{(k)} = \underline{g}^{(k)} \\ &\underline{\ln K}^{(k+1)} = \underline{\ln K}^{(k)} + \lambda \underline{p}^{(k)} \end{aligned} \tag{29}$$

The finishing algorithm is similar to the RAND method used by Gautam and Seider (1979) and Ohanomah and Thompson (1984).

The common feature of all these methods is that converged solutions are actual local minima in the Gibbs free-energy hypersurface. Some of them—FLASH3, FLASH4, and FLASH5—require a line search algorithm to compute the step length λ. This is done carefully, as discussed by Fletcher (1980), by a rigorous method. The purpose of the line search is to ensure a satisfactory decrease of the Gibbs free energy Δg. Two requirements have been set to achieve this aim:

$$\begin{aligned} &\Delta g^{(k)} - \Delta g^{(k+1)} > -\rho\lambda \underline{g}^{(k)T}\underline{s}^{(k)} \\ &|\underline{g}^{(k+1)T}\underline{s}^{(k)}| < -\sigma \underline{g}^{(k)T}\underline{s}^{(k)} \end{aligned} \tag{30}$$

Restricted, quadratic interpolations or cubic extrapolations and interpolations, depending on the tests in Eq. 30, are used to reach an acceptable value of λ. This line-search algorithm takes into account the effect of round-off errors and turns into an exact line search when the parameter σ tends to zero (Fletcher,

Table 1. System Composition and Parameters for Example 1

	Ethane C2	Propane C3	*n*-Butane nC4	*n*-Pentane nC5	*n*-Hexane nC6
z_i	0.39842	0.29313	0.20006	0.07143	0.03696
	$k_{1\text{-}2} = -0.0022$ $k_{2\text{-}3} = 0.0204$ $k_{3\text{-}4} = -0.01$ $k_{4\text{-}5} = 0.01$	$k_{1\text{-}3} = 0.0052$ $k_{2\text{-}4} = 0.0233$ $k_{3\text{-}5} = -0.0111$		$k_{1\text{-}4} = 0.0056$ $k_{2\text{-}5} = -0.0022$	$k_{1\text{-}5} = -0.043$

1980). This leads to more robustness and efficiency than the Fibonacci or the golden section line-search algorithms, which have been used respectively by Gautam and Seider and by Ohanomah and Thompson in their minimization procedure.

The computer storage requirements may vary from one method to another, but there is a great saving in storing $N \times N$ symmetric matrices over other published methods. Also, there is efficiency in using the Cholesky decomposition of symmetric positive definite matrices to determine Newton steps.

The function Δg (total molar Gibbs free energy), the gradient vector $\underline{g}$ (fugacity coefficients in both phases Φ_i^L and Φ_i^V $i = 1, \ldots, N$) and the Hessian matrix $\underline{\underline{H}}$ (second derivatives of Δg) are evaluated only if requested. Therefore, the number of evaluations of Δg, *N*GIBBS, of the fugacity coefficients, *N*FUG, and occasionally of the Hessian matrix, *N*HES, along with the number of iterations performed by the two algorithms used, give good information on the computational effort required by each of the methods cited.

Test Examples and Results

In the following, either the Soave-Redlich-Kwong equation of state (Soave, 1972) or the Mathias (1983) modification of it has been used as a thermodynamic model for both the vapor and liquid phases. The CPU execution time reported is for an IBM 4341. In comparison with other published results, three aspects should be taken into account: initialization; convergence criterion and tolerance; and computational effort.

Initialization. In all the new methods we used the correlation

$$K_i = \frac{P_c^i}{P} \exp\left[5.37(1 + \omega_i)\left(1 - \frac{T_c^i}{T}\right)\right] \quad i = 1, \ldots, N \tag{31}$$

to initialize the equilibrium ratios. This leads to a less accurate initialization than one from Raoult's law with an accurate correlation for pure-component vapor pressure.

Convergence Criterion and Tolerance. We used the gradient Euclidian norm as a convergence criterion:

$$\|\underline{g}\| = \left[\sum_{i=1}^{N} \ln^2\left(\frac{y_i\Phi_i^V}{x_i\Phi_i^L}\right)\right]^{1/2} \tag{32}$$

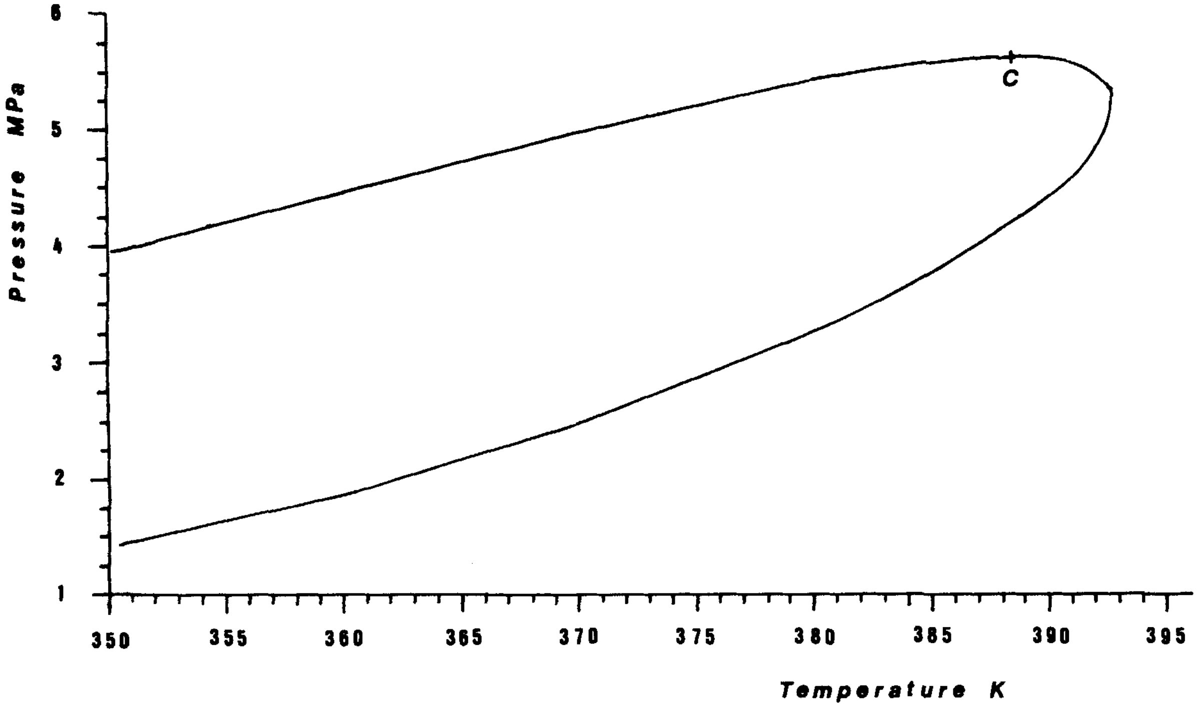

Figure 1. PT envelope of a five-component system, example 1.

Table 2. CPU Execution Time for Example 1 (43 points)

	Methods										
	FLASH1 QNSS + Newton	FLASH1*	FLASH2 SSM + BFGS	FLASH2*	FLASH3 SSM + BFGS	FLASH3*	FLASH4 SSM + CGFR	FLASH4 SSM + CGPR	FLASH5 SSM + Steep Desc.	QNSS	SSM
Time, s	6.34	3.84	9.15	4.32	11.48	5.74	10.94	13.15	11.13	7.27	23.57

*Calculations done with initialization procedure using previous converged points.

But one can also use the norm of the vector:

$$\underline{\Delta \ln K} = \underline{\ln K}^{(k)} - \underline{\ln K}^{(k-1)} \tag{33}$$

The relation between the two vectors is the following:

$$\underline{\Delta \ln K} = -(\underline{\underline{I}}_N + \underline{\underline{Q}}\,\underline{\underline{A}}^{-1})\underline{g} \tag{34}$$

so that the norms are of the same order of magnitude.

In global methods based on the Newton-Raphson or quasi-Newton techniques, the norm of the vector including all the equations is used. One can show that

$$\sum_{i=1}^{N} (y_i - K_i x_i)^2 \leq \sum_{i=1}^{N} \ln^2\left(\frac{y_i}{K_i x_i}\right) \tag{35}$$

Therefore, the convergence test based on Eq. 32 is more severe than the one used in the global approach in that the material balance equations are satisfied at each iteration in our approach. The tolerance in all the considered examples is set as $\epsilon = 10^{-5}$.

Computational Effort. All of our methods require function Δg, gradient vector g, and occasionally Hessian matrix $\underline{\underline{H}}$ computations. It is worth noting that Δg computational efforts are negligible compared to those of g and $\underline{\underline{H}}$, and using analytic derivatives from the equation of state in setting $\underline{\underline{H}}$ results in efforts not more than twice those required by the fugacity coefficient calculations (Michelsen, 1982b).

Example 1: A five-component system

The system is taken from Mehra et al. (1983); the pure-component parameters are from Prausnitz et al. (1977). The Redlich-Kwong equation of state as modified by Mathias (1983) has been used as a thermodynamic model for the vapor and liquid phases. The system composition and the binary interaction parameters are given in Table 1.

The temperature is fixed at $T = 390$ K (slightly above the critical temperature of the mixture, Figure 1, so that one can observe the retrograde condensation as the pressure varies. The calculations are done for 43 points ranking from the lowest to the highest saturation pressure.Table 2 shows the CPU execution time for the different methods; Table 3 presents results concerning some points.

Example 2: A seven-component system

This example is taken from Michelsen (1982b). A natural gas mixture has the following composition:

	C1	C2	C3	nC4	nC5	nC6	N2
z_i	0.943	0.027	0.0074	0.0049	0.0027	0.001	0.014

Table 3. Test 1, A Five-component System; T = 390 K

		FLASH1 ONSS + Newton					FLASH2 SSM + Newton					FLASH3 SSM + BFGS				FLASH4 SSM + CGFR				FLASH5 SSM + Steep Desc.		
Point	P MPa	NIT	NG	NF	NH	ER<	NIT	NG	NF	NH	ER<	NIT	NG	NF	NIT_2	NIT	NG	NF	NIT_2	NIT	NG	NF
A1	4.42	6	8	7	2	10^{-20}	7	9	8	2	10^{-12}	11	13	12	7	7	11	10	3	7	12	10
B1	4.71	5	7	6	2	10^{-12}	6	8	7	3	10^{-15}	9	11	10	7	6	10	9	4	6	10	9
C1	5.00	5	7	6	2	10^{-10}	6	8	7	3	10^{-11}	9	11	10	7	6	10	9	4	6	10	9
D1	5.29	6	8	7	3	10^{-11}	8	10	9	4	10^{-16}	12	14	13	9	7	16	11	4	8	14	13
E1	5.406	7	9	8	4	10^{-16}	9	11	10	4	10^{-13}	14	16	15	10	8	13	12	4	10	16	15
F1	5.435	7	9	8	3	10^{-12}	10	12	11	4	10^{-13}	15	17	16	10	9	14	13	4	11	16	15
G1	5.464	7	9	8	3	10^{-11}	11	13	12	4	10^{-13}	17	19	18	11	10	15	14	4	13	25	20
H1	5.493	7	9	8	3	10^{-10}	12	14	13	4	10^{-12}	18	20	19	11	11	16	15	4	14	26	21
I1	5.552	8	10	9	3	10^{-10}	15	17	16	4	10^{-12}	22	24	23	12	14	19	18	4	18	31	26
J1	5.551	10	12	11	4	10^{-14}	21	23	22	4	10^{-12}	29	31	30	13	20	25	24	4	24	38	33
K1	5.58	14	16	15	3	10^{-14}	21	25	24	4	10^{-11}	25	29	28	6	25	34	33	8	24	35	30
L1	5.5885	16	18	17	2	10^{-10}	21	26	25	3	10^{-13}	24	29	28	6	22	29	28	4	23	30	29

Vapor Fraction β

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1
0.9988	0.9473	0.8876	0.8179	0.7892	0.7829	0.7779	0.7753	0.7779	0.7947	0.8763	0.9693

NIT – no. iterations.
NIT_2 – no. iterations in step 2.
NG – NGIBBS; NF – NFUG; NH – NHES.
ER – $\|g\|^2$.

Table 4. Points for Example 2

	Point										
	A2	B2	C2	D2	E2	F2	G2	H2	I2	J2	K2
T, K	165.5	187.6	196.4	202.56	203.13	203	204	204.12	213	227.1	258.47
P, atm	20	40	50	57.11	58.1	57	58	59.32	70	80	25

Table 5. CPU Execution Time for Example 2 (11 points)

	Method							
	FLASH1 QNSS + Newton	FLASH2 SSM + Newton	FLASH3 SSM + BFGS	FLASH4 SSM + CGFR	FLASH4 SSM + CGPR	FLASH5 SSM + Steep Desc.	QNSS	SSM
Time, s	2.88	4.22	6.10	9.35	7.96	17.93	5.01	27.57

Table 6. Test 2, A Natural Gas Mixture

		FLASH1 ONSS + Newton					FLASH2 SSM + Newton					FLASH3 SSM + BFGS				FLASH4 SSM + CGFR				FLASH5 SSM + Steep Desc.		
Point	Michelsen* NIT	NIT	NG	NF	NH	ER<	NIT	NG	NF	NH	ER<	NIT	NG	NF	NIT_2	NIT	NG	NF	NIT_2	NIT	NF	NG
A2	1	4	6	5	0	10^{-13}	4	6	5	0	10^{-13}	4	6	5	0	4	6	5	0	4	6	5
B2	1	7	9	8	0	10^{-10}	8	10	9	0	10^{-13}	8	10	9	0	8	10	9	0	8	10	9
C2	1	8	10	9	3	10^{-15}	14	16	15	3	10^{-14}	18	20	19	6	15	21	20	5	16	20	19
D2	3	13	15	14	4	10^{-13}	18	22	21	7	10^{-15}	27	33	31	14	28	50	46	17	21	43	31
E2	4	13	15	14	4	10^{-13}	17	21	20	5	10^{-10}	30	35	33	16	71	98	96	69	86	175	174
F2	3	6	8	7	4	10^{-11}	7	9	8	5	10^{-18}	16	18	17	13	11	24	19	9	18	41	36
G2	3	6	8	7	4	10^{-14}	6	8	7	4	10^{-10}	15	17	16	12	10	16	15	8	31	66	63
H2	4	10	12	11	4	10^{-11}	17	21	20	4	10^{-15}	26	30	29	11	19	32	28	6	195	321	320
I2	2	12	14	13	5	10^{-15}	16	23	19	9	10^{-14}	30	38	35	20	20	43	36	12	41	79	71
J2	1	9	11	16	5	10^{-18}	8	10	9	4	10^{-13}	12	14	13	7	9	17	16	6	16	29	28
K2	1	4	6	5	2	10^{-18}	4	6	5	2	10^{-18}	4	6	5	2	4	6	5	2	4	6	5

Vapor Fraction β

A2	B2	C2	D2	E2	F2	G2	H2	I2	J2	K2
0	0	0.0343	0.4083	0.6122	0.7137	0.8040	0.8679	0.9958	0.9990	0.9999

NIT – no. iterations.
NIT_2 – no. iterations in step 2.
NG – NGIBBS; NF – NFUG; NH – NHES.
ER – $\|g\|^2$.
*The number of iterations reported by Michelsen concerns only the Newton part of his method.

Table 7. Points for Example 3

	Point														
	A3	B3	C3	D3	E3	F3	G3	H3	I3	J3	K3	L3	M3	N3	O3
T, C	−60	20	100	10	110	20	100	10	20	30	60	80	50	80	60
P, atm	40	40	40	100	100	150	190	190	190	190	190	190	200	200	205

Table 8. CPU Execution Time for Example 3 (15 points)

	Method							
	FLASH1 QNSS + Newton	FLASH2 SSM + Newton	FLASH3 SSM + BFGS	FLASH4 SSM + CGFR	FLASH4 SSM + CGPR	FLASH5 SSM + Steep Desc	QNSS	SSM
Time, s	3.66	4.39	6.15	6.16	7.04	10.42	4.22	10.66

Table 9. Test 3, A Natural Gas Mixture

Point	Asselineau et al.	FLASH1 ONSS + Newton					FLASH2 SSM + Newton					FLASH3 SSM + BFGS				FLASH4 SSM + CGFR				FLASH5 SSM + Steep Desc.		
	NIT	NIT	NG	NF	NH	ER<	NIT	NG	NF	NH	ER<	NIT	NG	NF	NIT_2	NIT	NG	NF	NIT_2	NIT	NG	NF
A3	5	3	5	4	3	10^{-18}	3	5	4	3	10^{-18}	5	7	6	2	5	8	7	2	6	15	10
B3	4	3	5	4	3	10^{-20}	3	5	4	3	10^{-20}	5	7	6	2	4	7	6	2	5	21	11
C3	5	3	5	4	3	10^{-18}	3	5	4	3	10^{-18}	5	7	6	2	5	8	7	2	6	19	12
D3	6	4	6	5	4	10^{-21}	4	6	5	4	10^{-21}	8	10	9	6	5	9	8	2	7	18	15
E3	5	3	5	4	2	10^{-18}	3	5	4	2	10^{-18}	8	10	9	6	5	9	8	2	3	5	4
F3	7	5	7	6	4	10^{-14}	5	7	6	4	10^{-14}	7	9	8	5	8	13	12	5	12	32	25
G3	6	5	7	6	4	10^{-20}	5	7	6	4	10^{-20}	9	11	10	6	6	10	9	3	7	14	13
H3	—	10	12	11	6	10^{-12}	15	18	17	8	10^{-15}	26	29	27	18	16	29	26	9	63	134	125
I3	—	7	9	8	4	10^{-11}	9	11	10	5	10^{-15}	16	18	17	12	15	24	23	11	33	75	70
J3	10	7	9	8	4	10^{-16}	7	9	8	4	10^{-10}	15	20	16	12	10	19	18	8	21	47	45
K3	8	6	8	7	4	10^{-7}	6	8	7	4	10^{-13}	12	14	13	9	7	13	12	4	9	29	16
L3	8	6	8	7	4	10^{-19}	6	8	7	4	10^{-14}	11	13	12	8	9	13	12	6	7	14	13
M3	9	7	9	8	4	10^{-18}	7	9	8	4	10^{-10}	13	15	14	10	8	14	13	5	12	33	22
N3	8	6	8	7	3	10^{-12}	7	9	8	4	10^{-14}	13	15	14	10	7	14	13	4	8	15	14
O3	11	7	9	8	2	10^{-17}	8	10	9	4	10^{-12}	13	15	14	9	10	15	14	6	14	33	30

Vapor Fraction β

A3	B3	C3	D3	E3	F3	G3	H3	I3	J3	K3	L3	M3	N3	O3
0.5892	0.8799	0.9552	0.7599	0.9416	0.7242	0.9240	0.5769	0.6712	0.7301	0.8438	0.8998	0.8209	0.9154	0.865

NIT – no. iterations.
NIT_2 – no. iterations in step 2.
NG – NGIBBS; NF – NFUG; NH – NHES.
ER – $\|g\|^2$.

Pure-component parameters are from Prausnitz et al. (1977), and all binary interaction parameters are set equal to zero in the Soave-Redlich-Kwong equation of state. The critical point of the mixture as reported by Michelsen corresponds to T_c = 203.125 K, P_c = 58.108 atm (5.888 Pa). Calculations are done for 11 points along the phase boundary of the PT diagram, Table 4; results are summarized in Tables 5 and 6.

Example 3: An eight-component system

This example is taken from Asselineau et al. (1979). A natural gas contains the following:

	N2	C1	C2	C3	nC4	nC5	nC6	nC9
z_i	0.045	0.70	0.1171	0.0497	0.0277	0.015	0.0098	0.0357

The Mathias-Redlich-Kwong equation of state is used here with binary interaction parameters set equal to zero. The calculations are done for 15 points inside the PT envelope of the mixture, Table 7. The critical point as reported by Asselineau et al. corresponds to T_c = 10°C, P_c = 190 atm (19.25M Pa). Results are summarized in Tables 8 and 9.

Example 4: A ten-component system

A volatile oil has the following composition (Nghiem et al., 1983):

	C1	C2	C3	nC4	nC5
z_i	0.6436	0.0752	0.0474	0.0412	0.0297

	nC6	nC7	nC8	nC9	nC10
z_i	0.0138	0.0303	0.0371	0.0415	0.0402

The Soave-Redlich-Kwong equation of state is used for thermodynamic properties with binary interaction parameters set equal to zero. The critical point as reported by Nghiem et al. (1983) was calculated by the method of Peng and Robinson (1977): T_c = 162.2°C, P_c = 19.4 MPa. Flash calculations were performed at this critical temperature for 12 pressures ranking from 2,700 to 2,810 psia (18.6 to 19.4 MPa).

We report in Table 10 the CPU execution time for each of the methods cited and in Table 11 the performance characteristics of some of them. When comparing with the results of Nghiem et

Table 10. CPU Execution Time for Example 4 (12 points)

	Method										
	FLASH1 QNSS + Newton	FLASH2 SSM + Newton	FLASH3 SSM + BFGS	FLASH1*	FLASH2*	FLASH3*	FLASH4 SSM + CGFR	FLASH4 SSM + CGPR	FLASH5 SSM + Steep Desc.	QNSS	SSM
Time, s	5.73	10.12	12.95	2.53	3.02	4.11	16.19	18.83	61.19	6.40	88.7

*Calculations done with initialization procedure using previous converged points.

Table 11. Test 4, A Ten-component System; T - 162.2°C

	Nghiem et al.	FLASH1					FLASH2					FLASH3				FLASH4				FLASH5		
P	SSM + Powell	SSM + Newton					SSM + Newton					SSM + BFGS				SSM + CGFR				SSM + Steep Desc.		
psia	NIT NIT	NIT	NG	NF	NH	ER<	NIT	NG	NF	NH	ER<	NIT	NG	NF	NIT_2	NIT	NG	NF	NIT_2	NIT	NF	NG
2,700	16 + 18	9	11	10	4	10^{-16}	15	17	16	5	10^{-16}	16	18	17	7	16	22	21	7	32	48	43
2,710	16 + 18	9	11	10	4	10^{-15}	16	18	17	5	10^{-16}	17	19	18	7	22	27	26	12	34	50	45
2,720	17 + 18	9	11	10	4	10^{-14}	17	19	18	5	10^{-16}	18	20	19	7	22	27	26	11	24	43	38
2,730	17 + 18	9	11	10	4	10^{-13}	19	21	20	5	10^{-16}	21	23	22	8	17	22	21	4	26	45	40
2,740	17 + 17	9	11	10	4	10^{-12}	18	21	20	5	10^{-10}	21	24	23	6	22	27	26	7	48	64	59
2,750	18 + 17	10	12	11	4	10^{-12}	19	22	21	3	10^{-14}	26	29	28	11	28	33	32	13	59	78	73
2,760	18 + 19	11	13	12	4	10^{-12}	19	23	22	4	10^{-14}	26	30	29	11	28	46	39	13	29	46	37
2,770	18 + 18	12	14	13	4	10^{-11}	18	22	21	4	10^{-10}	27	31	30	13	21	33	29	7	55	73	65
2,780	19 + 18	13	15	14	3	10^{-10}	19	23	22	5	10^{-12}	28	32	31	14	22	31	30	8	89	110	101
2,790	19 + 19	14	16	15	3	10^{-11}	18	21	20	3	10^{-10}	31	34	33	17	20	27	26	6	111	190	189
2,800	19 + 19	15	17	16	3	10^{-10}	19	22	21	4	10^{-12}	30	34	33	16	38	46	43	24	65	82	73
2,810	20 + 20	20	22	21	3	10^{-12}	21	26	25	4	10^{-10}	35	40	39	19	63	88	74	47	724	847	846

Vapor Fraction β

A4	B4	C4	D4	E4	F4	G4	H4	I4	J4	K4	L4
0.577	0.574	0.570	0.566	0.562	0.558	0.553	0.548	0.542	0.536	0.527	0.514

NIT - no. iterations.
NIT_2 - no. iterations in step 2.
NG - NGIBBS; NF - NFUG; NH - NHES.
ER - $\|g\|^2$.

al, it is worth noting that they used 10^{-8} as a convergence tolerance for $\Sigma_{i=1}^{N}(y_i\Phi_i^V/x_i\Phi_i^L - 1)^2$, which results in a less severe test compared to our convergence tolerance of 10^{-10} for $\|g\|^2$. Their method is also a hybrid one, starting with SSM and finishing with Powell's (1970) algorithm, but it is by contrast an equation-based method. Its performance should be compared to that of the FLASH3 method based on the BFGS quasi-Newton method.

Discussion

Among the new methods proposed, FLASH1, which is a combination of QNSS and Newton algorithms, is the best from the

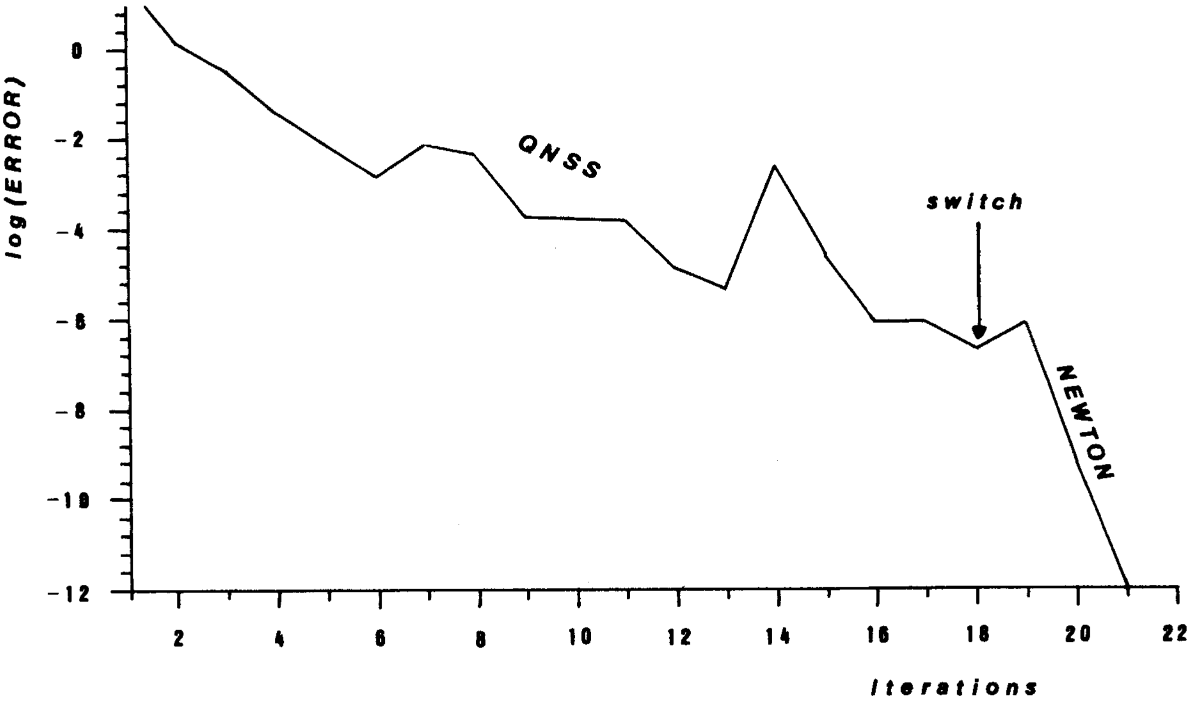

Figure 2. Log gradient-norm square vs. iteration number in point L4 calculations from FLASH1 method, example 4.

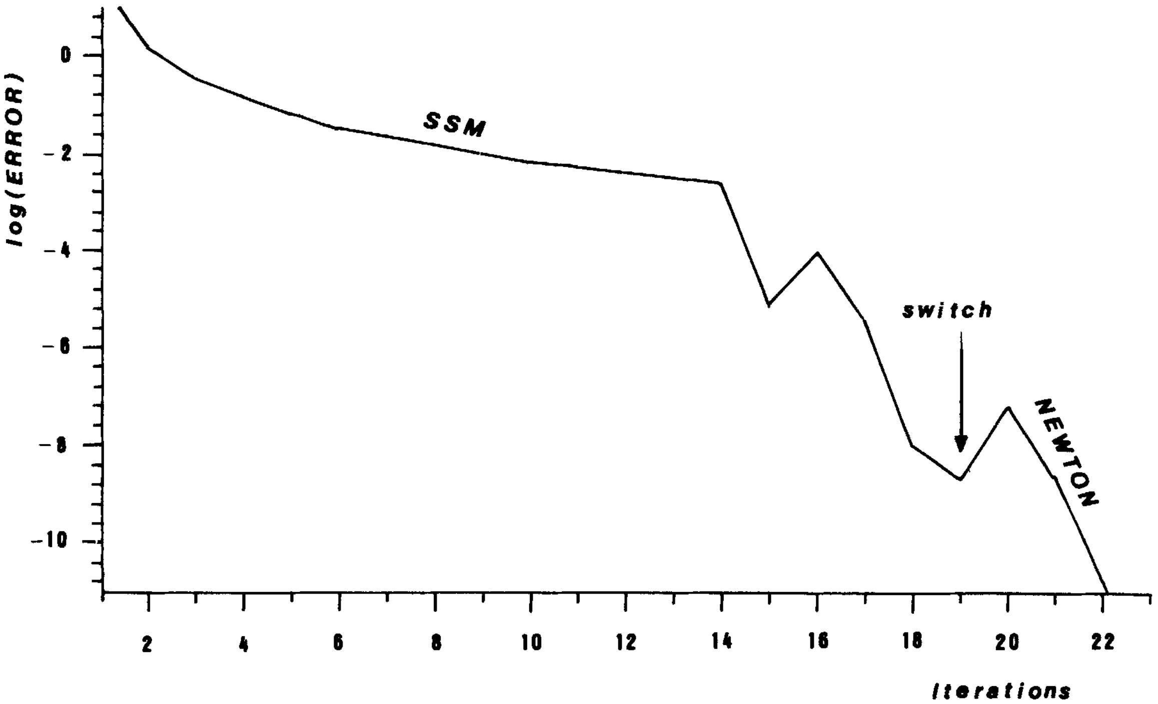

Figure 3. Log gradient-norm square vs. iteration number in point L4 calculations from FLASH2 method, example 4.

standpoints of computer storage and CPU time execution requirements.

FLASH2 performs less efficiently than FLASH1, as it uses SSM instead of QNSS, but quadratic convergence is achieved by both.

Both FLASH3 and QNSS are of the quasi-Newton type and require approximately the same number of iterations to converge. QNSS is faster in that it requires no matrix manipulation. The advantage of using FLASH3 over QNSS is that the converged matrix from BFGS is very close to the inverse of the actual Hessian matrix, so that it has the same features as Newton methods. FLASH3 is suitable when the derivatives of fugac-

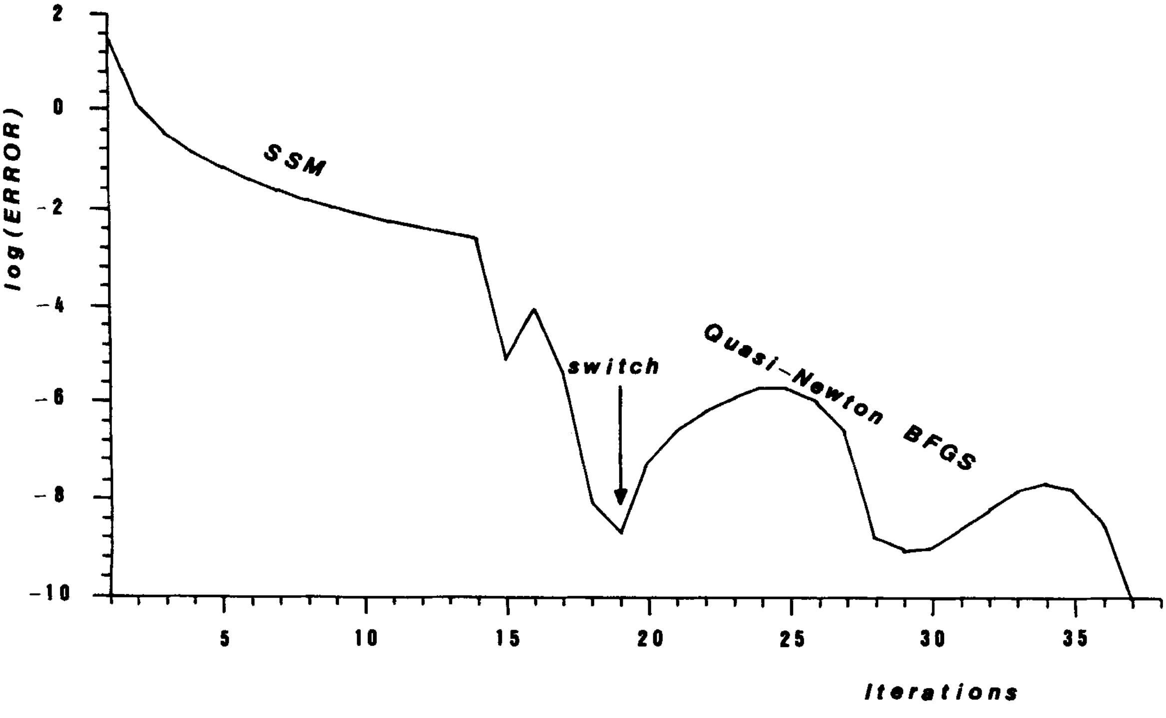

Figure 4. Log gradient-norm square vs. iteration number in point L4 calculations from FLASH3 method, example 4.

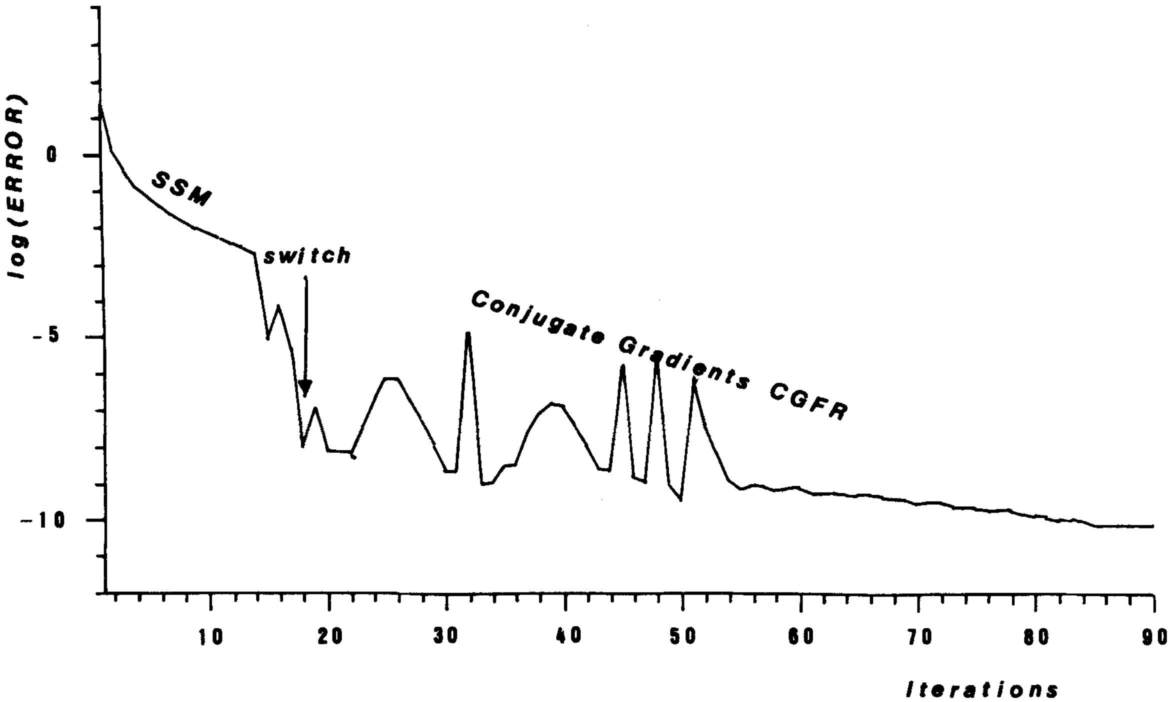

Figure 5. Log gradient-norm square vs. iteration number in point L4 calculations from FLASH4 method, example 4.

ity coefficients are not available from the thermodynamic model. Superlinear convergence is achieved by both FLASH3 and QNSS.

FLASH4, which is based on the conjugate-gradients algorithm, is less efficient than QNSS, as it requires the same amount of computer storage and computational effort per iteration. FLASH4 is indeed very sensitive to round-off errors.

One can notice that the convergence acceleration with FLASH1, FLASH2, FLASH3, and FLASH4 over SSM gets greater as the point gets closer to a critical point. This is illustrated by Figures 2 to 5, which respectively give the convergence profile with FLASH1, FLASH2, FLASH3, and FLASH4 for the closest point to the critical in example 4. SSM requires more than one thousand iterations for this point.

FLASH5, based on the steepest-descent algorithm, performs as badly as SSM near critical conditions and there is no significant improvement of convergence over SSM.

In comparison with other published methods, it is obvious that there is robustness in adopting a hybrid approach in all the cited methods, and efficiency in storing only $N \times N$ symmetric matrices and in using the Cholesky decomposition of symmetric positive definite matrices to determine the steps for Newton methods. Therefore, there is no doubt that the FLASH1 and FLASH2 methods perform better than that of Asselineau et al. based on a global Newton-Raphson algorithm. Also, our FLASH3 method, which is similar in structure to that of Nghiem et al. (based on Powell's algorithm), is more efficient than the latter from the standpoint of computer storage as well as the calculational effort requirements.

Conclusions

New methods of vapor-liquid flash calculation have been considered. All use a hybrid approach starting with a safe algorithm to make good progress from poor initializations. The common feature of these methods is that converged solutions correspond to actual minima of the Gibbs free energy of the system.

One can replace the first algorithm of all these methods by one of stability analysis, such as that of Baker et al. (1982) and Michelsen (1982a), trying to locate the point in the PT diagram before performing any flash calculations. This would certainly lead to greater efficiency for specifications close to a critical point, in the stable region outside the PT envelope.

All the methods cited may be modified to handle other types of vapor-liquid flash problems, such as the saturation points and the isenthalpic or isentropic flash calculations.

Although we used a cubic equation of state as a thermodynamic model for all tests, other models may be used as well, provided they can give the Gibbs free energy of a phase G^L (and G^V), the first derivatives of G^L (and G^V) (fugacity or activity coefficients), and the second derivatives of G^L (and G^V) for Newton methods.

Notation

$\underline{\underline{A}}$ - $N \times N$ symmetric matrix, Eq. 7
$\underline{\underline{B}}$ - approximant to inverse of Hessian matrix in BFGS
$\underline{f}$ - function, Appendix A
$\underline{g}$ - gradient vector $g_i = (\partial \Delta g/\partial v_i)$, $i = 1, \dots, N$
G^{o} - standard state Gibbs free energy of system
G^L - Gibbs free energy of liquid phase
G^V - Gibbs free energy of vapor phase
G - total Gibbs free energy of system
$\underline{\underline{H}}$ - $N \times N$ Hessian matrix $H_{i,j} = (\partial^2 \Delta g/\partial v_i \partial v_j)$, $i = 1, \dots, N$, $j = 1, \dots, N$
$\underline{\underline{I}}_N$ - $N \times N$ identity matrix
k_{i-j} - binary interaction parameter for equation of state
$\underline{K}$ - vector of equilibrium ratios, K_i, $i = 1, \dots, N$
$\underline{l}$ - vector of mole numbers l_i, $i = 1, \dots, N$, in liquid phase

$\bar{L}$ – total mole number in liquid phase
N – number of components
$\underline{O}_N$ – $N \times N$ null matrix
$\bar{\underline{p}}$ – step vector on variables $\ln K_i$, $i = 1, \ldots, N$
$P^{\bar{s}}$ – pressure at standard state of 1 atm (101.3 kPa)
P, P_1, P_2 – pressure of system
P_c – critical pressure of system
P_c^i – critical pressure of component i, $i = 1, \ldots, N$
$\underline{Q}$ – $N \times N$ symmetric matrix, Eq. 8
$\bar{R}$ – gas constant
$\underline{s}$ – descent direction in a minimization method
$\bar{S}$ – variable, Appendix
T – temperature of system
T_c – critical temperature of system
T_c^i – critical temperature of component i, $i = 1, \ldots, N$
$\underline{U}$ – $N \times N$ upper triangular matrix
$\bar{\underline{v}}$ – vector of mole numbers v_i, $i = 1, \ldots, N$, in vapor phase
$\bar{V}$ – total mole number in vapor phase
$\underline{x}$ – vector of molar fractions x_i, $i = 1, \ldots, N$, in liquid phase
$\bar{\underline{X}}$ – vector of variables, X_i, $i = 1, \ldots, N$, Eq. 12
$\bar{\underline{y}}$ – vector of molar fractions y_i, $i = 1, \ldots, N$, in vapor phase
$\bar{\underline{z}}$ – vector of molar fractions z_i, $i = 1, \ldots, N$, in system

Greek letters

β – molar vapor fraction
δ_{ij} – Kronecker delta $\delta_{ij} = 1$ if $i = j$, 0 if not $i = 1, \ldots, N$, $j = 1, \ldots, N$
ϵ – convergence tolerance for gradient norm
Φ_i^L – fugacity coefficient of component i in liquid phase
Φ_i^V – fugacity coefficient of component i in vapor phase
λ – step length
μ_i° – chemical potential of component i at standard state
μ_i^L – chemical potential of component i in liquid phase
μ_i^V – chemical potential of component i in vapor phase
ν – parameter in conjugate-gradients method
ω_i – acentric factor of component i
ρ – line-search parameter
ρ_i – variables, Appendix
σ – line-search parameter
ξ, ξ_i – eigenvalues of matrix A, Appendix
Δg – $(G - G^\circ)/RT$
Δ – increment
$\underline{\nabla}$ – first derivatives or gradient vector
$\underline{\underline{\nabla}}^2$ – second derivatives or Hessian matrix

Subscripts

i, j – component index
c – critical property

Superscripts

i – component index
(k) – iteration index
L – liquid phase
V – vapor phase
T – transpose of a matrix
$-T$ – inverse of the transpose of a matrix

Appendix

Properties of matrix $\underline{\underline{A}}$

The matrix $\underline{\underline{A}} = (\partial \ln K_i/\partial v_j)$ is symmetric positive definite in the domain of the vapor and liquid phases coexistence.

Proof

In the vapor-liquid region, we can express the matrix $\underline{\underline{A}}$ as:

$$\underline{\underline{A}} = \frac{\partial \ln K_i}{\partial v_j} = \delta_{ij}\left(\frac{1}{v_i} + \frac{1}{l_i}\right) - \frac{1}{V} - \frac{1}{L}$$

$$= \frac{1}{VL}\left(-1 + \delta_{ij}\frac{z_i}{x_i y_i}\right) \quad i = 1, \ldots, N \quad \text{(A1)}$$

where

$$x_i = \frac{l_i}{L} \qquad y_i = \frac{v_i}{V} \quad i=1, \ldots, N$$

$$L = \sum_{i=1}^{N} l_i \qquad V = \sum_{i=1}^{N} v_i \quad i = 1, \ldots, N$$

$$z_i = l_i + v_i$$

The eigenvalues of the symmetric matrix $[-1 + \delta_{ij}(z_i/x_i y_i)]$ are the roots of the algebraic equation:

$$f(\xi) = \left[\prod_{i=1}^{N}\left(\frac{z_i}{x_i y_i} + 1 - \xi\right)\right] \cdot \left[1 - \sum_{i=1}^{N}\left(\frac{1}{\dfrac{z_i}{x_i y_i} + 1 - \xi}\right)\right] = 0 \quad \text{(A2)}$$

Ordering $\rho_i = 1 + (z_i/x_i y_i)$, $i = 1, \ldots, N$, such that

$$1 < \rho_1 < \rho_2 < \cdots < \rho_{N-1} < \rho_N \quad \text{(A3)}$$

one can deduce that

$$f(\rho_i) = -\sum_{\substack{j=1 \\ j \neq i}}^{N} (\rho_j - \rho_i)$$

has the sign of $(-1)^i$, $i = 1, \ldots, N$. Also,

$$f(1) = \left(\prod_{i=1}^{N} \frac{z_i}{x_i y_i}\right)\left(1 - \sum_{i=1}^{N} \frac{x_i y_i}{z_i}\right)$$

is strictly positive since

$$S = 1 - \sum_{i=1}^{N} \frac{x_i y_i}{z_i}$$

is positive in the vapor-liquid region.

Since f is a continuous function, there exist N different eigenvalues of the matrix $[-1 + \delta_{ij}(z_i/x_i y_i]$, $\xi_1, \xi_2, \ldots, \xi_N$ that satisfy:

$$1 < \xi_1 < \rho_1 < \xi_2 < \rho_2 < \cdots < \xi_{N-1} < \rho_{N-1} < \xi_N < \rho_N \quad \text{(A4)}$$

Therefore, all the eigenvalues of matrix $\underline{\underline{A}}$ are greater than $1/LV$ and $\underline{\underline{A}}$ is positive definite.

Literature cited

Ammar, M. N., Ph.D. Thesis, Ecole des Mines de Paris (1986).

Anderson, T. F., and J. M. Prausnitz, "Computational Methods for High-pressure Phase Equilibria and Other Fluid Properties Using a Partition Function," *Ind. Eng. Chem. Process Des. Dev.*, **19**, 9 (1980).

Asselineau, L., G. Bogdanic, and J. Vidal, "A Versatile Algorithm for Calculating Vapor-Liquid Equilibria," *Fluid Phase Equil.*, **3**, 273 (1979).

Baker, L. E., A. C. Pierce, and K. D. Luks, "Gibbs Energy Analysis of Phase Equilibria," *Soc. Pet. Eng. J.*, **22**, 731 (1982).

Boston, J. F., and H. I. Britt, "A Radically Different Formulation and Solution of the Single-Stage Flash Problem," *Comput. Chem. Eng.*, **2**, 109 (1978).

Broyden, C. G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Math. Comp.*, **19**, 577 (1965).

Crowe, A. M., and M. Nishio, "Convergence Promotion in the Simulation of Chemical Processes. The General Dominant Eigenvalue Method," *AIChE J.*, **21**, 528 (1975).

Fletcher, R., *Practical Methods of Optimization*, Vol. 1 *Unconstrained Optimization*, Wiley-Interscience, New York (1980).

Fussell, D. D., and J. L. Yanosik, "An Iterative Sequence for Phase Equilibria Calculations Incorporating the Redlich-Kwong Equation of State," *Soc. Pet. Eng. J.*, **18**, 173 (1978).

Gautam, R., and W. D. Seider, "Computation of Phase and Chemical Equilibria," *AIChE J.*, **25**, 991 (1979).

Grens, E. A., "Efficient Use of Thermodynamic Models in Process Calculations," *Proc. 2nd Int. Conf. Foundations Computer-Aided Process Design*, CACHE, 249 (1983).

Heidemann, R. A., "Computation of High-pressure Phase Equilibria," *Fluid Phase Equil.*, **14**, 55 (1983).

Hirose, Y., Y. Kawase, and M. Kudoch, "General Flash Calculation by the Newton-Raphson Method," *J. Chem. Eng. Japan*, **11**(2), 150 (1978).

Joulia, X., P. Maggiochi, B. Koehret, H. Paradowski, and J. J. Bartuel, "Hybrid Method for Phase Equilibrium Calculations," *Fluid Phase Equil.*, **26**, 15 (1986).

Lucia, A., D. C. Miller, and A. Kumar, "Thermodynamically Consistent Quasi-Newton Formulae," *AIChE J.*, **31**, 1381 (1985).

Mathias, P. M., "A Versatile Phase Equilibrium Equation of State," *Ind. Eng. Chem. Process Des. Dev.*, **22**, 385 (1983).

Mehra, R. K., R. A. Heidemann, and K. Aziz, "An Accelerated Successive-Substitution Algorithm," *Can. J. Chem. Eng.*, **61**, 590 (1983).

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

———, "The Isothermal Flash Problem. II: Phase-Split Calculation," *Fluid Phase Equil.*, **9**, 21 (1982b).

Nghiem, L. X., and R. A. Heidemann, "General Acceleration Procedure for Multiphase Flash Calculation with Application to Oil-Gas-Water Systems," 2nd Eur. Symp. Enhanced Oil Recovery, Paris (1982).

Nghiem, L. X., K. Aziz, and Y. K. Li, "A Robust Iterative Method for Flash Calculations Using the Soave-Redlich-Kwong or the Peng-Robinson Equation of State," *Soc. Pet. Eng. J.*, **23**, 521 (1983).

Null, H. R., *Phase Equilibrium in Process Design*. Wiley-Interscience, New York (1970).

Ohanomah, M. O., and D. W. Thompson, "Computation of Multicomponent Phase Equilibria," *Comput. Chem. Eng.*, **8**, 147 (1984).

Peng, D. Y., and D. B. Robinson, "A Rigorous Method for Predicting the Critical Properties of Multicomponent Systems from an Equation of State," *AIChE J.*, **23**, 137 (1977).

Powell, M. J. D., *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach, London (1970).

Prausnitz, J. M., T. F. Anderson, E. A. Grens, C. A. Eckert, R. Hsieh, and J. P. O'Connell, *Computer Calculations for Multicomponent Vapor-Liquid and Liquid-Liquid Equilibria*, Prentice-Hall, Englewood Cliffs, NJ (1980).

Prausnitz, J. M., R. C. Reid, and T. K. Sherwood, *The Properties of Gases and Liquids*, McGraw-Hill, New York, 3rd ed. (1977).

Rachford, H. H., Jr., and J. D. Rice, *Petroleum Trans., AIME*, **195**, 327 (1952).

Schubert, L. K., "Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian," *Math. Comp.*, **24**, 27 (1970).

Soave, G., "Equilibrium Constants from a Modified Redlich-Kwong Equation of State," *Chem. Eng. Sci.*, **27**, 1197 (1972).

Manuscript received Apr. 10, 1986, and revision received Oct. 6, 1986.