

INFLUENCE OF THE THERMOPHYSICAL PROPERTIES EXTRAPOLATION, IN THE COURSE OF ITERATIVE CALCULATIONS, ON THE CONVERGENCE CHARACTERISTICS OF HIGHER LEVEL SIMULATION ALGORITHMS – A GENERAL ANALYSIS

Stefan G. TSVETKOV and Roumiana P. STATEVA

*Institute of Chemical Engineering, Bulgarian Academy of Sciences,
acad. G. Bontchev str., bl. 103, Sofia 1113, Bulgaria*

Received June 17, 1991

Accepted December 7, 1991

An extrapolating model provides values for thermophysical properties when infeasible specifications (either pressure or temperature) are recognized at a given iteration in a process simulator, applying an EOS technique. The model has a considerable effect on the convergence characteristics, accuracy and robustness of a higher level simulation algorithm. The present paper outlines a strategy for evaluating and comparing the abilities of different extrapolating techniques: to promote the convergence of higher level algorithms; to ensure for them a solution, which will be of accuracy, not significantly inferior, to the actual one. The latter could have been obtained using for example a different thermodynamic model.

Organizing thermophysical properties calculations in a process simulator is a responsible task as they are prone to consume around 80% of the computer time¹. In regard of this, additional difficulties, appearing in the case of an equation of state (EOS), applied to both phases (liquid and vapor), are highly undesirable. In EOS computations, it is the transition from regions of x , T , P , where there are three volume roots, to regions where there is only one volume root that can be viewed as responsible for convergence to a trivial solution². The trivial solution can interrupt the performance of higher level simulation algorithms and spoil their convergence in the two-phase region. Hence, the interest to developing efficient methods to circumvent such problems.

As shown by Jovanovic and Pounovic³ there are two approaches to ensure the proper solution of higher level simulation algorithms:

- the first is to make good initial estimates^{4,5} or to use initial estimates, provided by stability analysis⁶;
- the second is to force a satisfactory volume (density, compressibility) value to be generated by the EOS (refs⁷⁻¹²).

The second approach has proved to be more reliable. The strategies, suggested in the literature, apply different techniques to produce "pseudo"/"artificial" volume values,

once infeasible pressure specifications, for a given phase, are recognized at an EOS level in the process simulator. Then this volume value is used to estimate the "pseudofugacity" coefficients, and if required, the rest of the derived thermodynamic properties. Such procedure hereafter will be called an "extrapolating procedure".

As shown by Mathias et. al.⁹ the extrapolated thermoproperties should satisfy the following requirements: should follow the tendency of the real ones for the phase specified; should be continuous and differentiable; should improve the convergence characteristics of the higher level simulation algorithms. However, the papers published provide little knowledge of the strength and limitations of the suggested extrapolating strategies in the sense of the last requirement. Since the influence of an extrapolating model on the convergence characteristics of a higher level simulation algorithm is decisive for its robustness, a further examination in this direction is justifiable. To facilitate it, a general approach, designed to evaluate in an intelligible, quantitative manner this important feature of different extrapolating techniques, is presented in the manuscript. Its main characteristics are outlined on the example of the standard isothermal two-phase vapor-liquid flash problem, which could be treated as a simple higher level simulation algorithm. A brief discussion follows in a later section on the example of the bubble point temperature algorithm.

THEORETICAL

The approach is based on the classical fixed-point theory and its connection with the contraction-mapping principle since the latter has the potential to give insight into the convergence behavior of algorithms used for steady-state simulation. To make the manuscript more readable, firstly some general considerations about the connection between an algorithm and a mapping will be given.

Any computational algorithm at all, can be represented as a mapping:

$$\Phi : x \rightarrow y,$$

where $x \in \mathcal{X}^m$, $y \in \mathcal{X}^n$ are the vectors of the input and output values. The iterative calculations are represented as Φ^n , n must be equal to m .

A condition for convergence of a fixed-point iterative sequence, $x^{(k+1)} = \Phi(x)^{(k)}$, is given with the contraction-mapping theorem¹³. From it follows that: if $|\lambda_{\max}| < 1$, where λ_{\max} is the maximum modulus eigenvalue of the Jacobian matrix $M\Phi$, the mapping Φ is contractive. The contraction-mapping principle can be applied to any computational algorithm to establish whether the fixed-point iteration sequence converges. To elucidate the ideas the well-known isothermal vapor-liquid flash algorithm, based on the traditional single loop univariable method¹⁴, will be used as an example in the present study.

It is known that when the pressure-temperature specification is in two-phase region a physical ("real") solution of the flash problem exists. The representation of one iteration step of its algorithm (Chart 1-A), as a mapping Φ is given in Chart 1-B. Let one

CHART 1-A

1. Start with x_i and y_i .
2. Compute v^L , v^V , $\ln \varphi_i^L$, $\ln \varphi_i^V$.
3. Test for convergence $\sum (\ln (f_i^L / f_i^V)) < \Delta_{\text{err}}$?
 Yes – Stop
 No – Continue with Step 4.
4. Calculate $K_i = \exp (\ln \varphi_i^L - \ln \varphi_i^V)$.
5. Solve the equation: $f(\alpha, K_1, K_2, \dots, K_{Nc}) = 0$ for the vapor fraction α .
6. Compute x and y from:

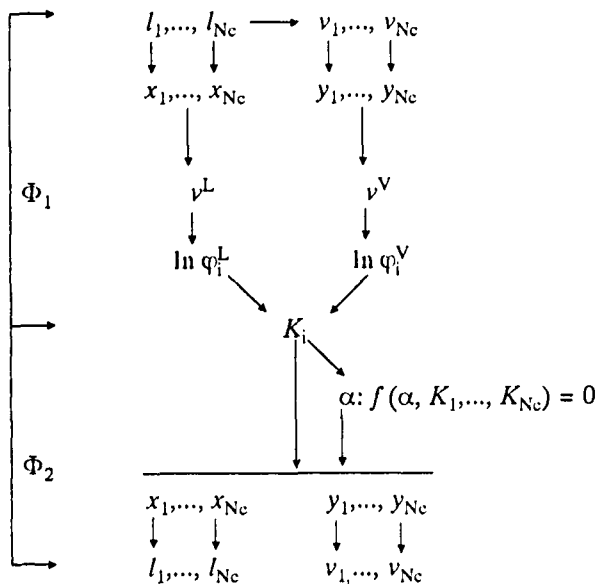
$$x_i = \frac{z_i}{\alpha(K_i - 1) + 1} \quad \text{and} \quad y = K_i x_i$$

and go back to Step 2.

The equation $f(\alpha, K_1, K_2, \dots, K_{Nc}) = 0$ might be any of the known formulations of the single loop univariable methods for solving the flash problem, as discussed by Ohanomah and Thompson¹⁴.

The accepted $\Delta_{\text{err}} = 10^{-12}$.

CHART 1-B



mole of the feed be flashed and let l_i be the independent variables. The assumption that $F = 1$ is just for simplification and does not in any way influence the generality of the following conclusions. In that case $L = 1 - \alpha$ and $V = \alpha$. The choice of the vapor-liquid (liquid-liquid) flash algorithm as an example is motivated by the fact that: firstly, it is used as a constituent part of higher level simulation algorithms; secondly, it contains the volume-finding routine. The latter is where an extrapolating procedure might be required to provide values for the volume and the derived properties (for the phase specified).

It is convenient, in our case, to represent the mapping Φ of the flash algorithm as a composition of two mappings (as shown on Chart 1-B) and its Jacobian matrix $M\Phi$ as a product of the Jacobian matrices of Φ_1 and Φ_2 mappings:

$$\Phi = \Phi_2 \circ \Phi_1 \quad \text{and} \quad M\Phi = M\Phi_2 \cdot M\Phi_1 \quad (1)$$

The operation "composition" is represented through the symbol \circ in the above expression.

$\Phi_1 : l \rightarrow K$ is the mapping, where extrapolations might occur; $\Phi_2 : K \rightarrow l$, where $l, K \in \mathcal{R}^n$.

The matrix of the first mapping is $M\Phi_1 = \partial K_i / \partial l_j$, where:

$$\frac{\partial K_i}{\partial l_j} = \frac{\partial(\exp(\ln \varphi_i^L - \ln \varphi_i^V))}{\partial l_j} = K_i \left(\frac{\partial \ln \varphi_i^L}{\partial l_j} + \frac{\partial \ln \varphi_i^V}{\partial v_j} \right) \quad (2)$$

The partial derivatives of the fugacity coefficients are well known and widely used analytical expressions¹⁵. The matrix of the second mapping is $M\Phi_2 = (\partial l_i / \partial K_j)$.

The elements of $M\Phi_2$, when the functional dependence $f(\alpha, K)$ is the widely-used Rachford-Rice formulation, are shown hereunder:

$$m_{ij} = \alpha(1 - \alpha) \frac{x_j y_i}{z_i} \frac{1}{K_i} \left(-\delta_{ij} + \frac{x_j y_j / z_j}{s_1} \right) \quad (3)$$

where

$$s_1 = (1 - \alpha) \sum \frac{z_i (K_i - 1)}{[1 + \alpha (K_i - 1)]^2}$$

Since the pressure-temperature specification is assumed to be in the two-phase region, the isothermal flash algorithm has a nontrivial solution and its mapping Φ must be contractive. The latter has to be proved.

Appendix demonstrates that $M\Phi$ and a matrix \mathbf{S} are equivalent. The matrix \mathbf{S} , discussed by Michelsen¹⁶ is:

$$s_{ij} = \frac{\partial \ln K_i^{(t+1)}}{\partial \ln K_j^{(t)}} = \frac{\partial(\ln \varphi_i^L - \ln \varphi_i^V)}{\partial \ln K_j^{(t)}} = \sum_{m=1}^N \left(\frac{\partial \ln \varphi_i^L}{\partial n_m} + \frac{\partial \ln \varphi_i^V}{\partial n_m} \right) \frac{\partial l_m}{\partial \ln K_j} \quad (4)$$

Michelsen¹⁶ has shown that the largest modulus eigenvalue of \mathbf{S} is less than one in the domain of the vapor-liquid phases coexistence. Since matrices \mathbf{S} and $M\Phi$ are equi-

valent, they share the same eigenvalues. Hence, the largest modulus eigenvalue of $M\Phi$ is less than one as well. From here follows directly that the mapping Φ is contractive.

The Flash Algorithm when an Extrapolating Procedure is Applied

In this section we demonstrate how the contraction-mapping principle, discussed above, can be used to analyse the convergence behavior of a higher level algorithm for steady-state calculations after an extrapolating procedure has been applied at some iteration. In particular to clarify whether an extrapolating model is: (i) suitable, i.e. promotes the convergence; (ii) reliable, i.e. ensures for a higher level simulation algorithm a solution, which will be of accuracy, not significantly inferior, to the "real" one. The "real" solution is the fixed point and it might be obtained either applying a mixed thermodynamic model (e.g. γ - φ method) or using different initial estimates.

The primary implication of the discussion in the previous paragraph is that since any extrapolating procedure can be represented as a mapping, it could be included as a constituent part in Φ_1 , instead of the original volume-finding routine and the standard methods for estimating fugacity coefficients. Thus, Φ_1 will be transformed to Φ_1^* . The latter manifests the situation when at a given iteration step k , for example for the current values of $f^{(k)}$, the volume-finding procedure fails to find (at the specified T and P) a root of the EOS, corresponding to the equilibrium liquid phase, and an extrapolating technique is applied. Values for the pseudovolume v^{L^*} and the pseudofugacity coefficients $\ln \varphi^{L^*}$ are obtained, which are used further to estimate $K^{(k)}$, α , $f^{(k+1)}$. The paradigm of the new composite mapping Φ^* is shown on Chart 1-C.

Mapping Φ^* will be contractive if the largest modulus eigenvalue of its Jacobian matrix $M\Phi^* = M\Phi_2 \cdot M\Phi_1^*$ is less than one. However, if the applied extrapolating technique is suitable, in the sense that it ensures and promotes convergence of the algorithm, Φ^* has to be a contraction mapping. To verify this, the following two-step analysis is suggested:

1. Form the Jacobian matrix $M\Phi^*$:

$$M\Phi^* = \left(\frac{\partial f_i^{(k+1)}}{\partial f_j^{(k)}} \right) = M\Phi_2 \cdot M\Phi_1^* = \left(\frac{\partial f_i^{(k+1)}}{\partial K_m} \right) \left(\frac{\partial K_m}{\partial f_j^{(k)}} \right),$$

where the elements of the matrix $M\Phi_2$ (Eq. (3)) are calculated after the extrapolating procedure has been introduced; the elements of the Jacobian matrix $M\Phi_1^*$ (Eq. (2)) are calculated, using $f^{(k)}/v^{(k)}$, the "pseudovolume" v^{L^*} for the liquid phase and the vapor phase volume v^v .

2. A standard procedure, calculating $M\Phi^*$ eigenvalues, is applied.

If the largest modulus eigenvalue of $M\Phi^*$ is greater than one, Φ^* is not contractive. This implies that the algorithm is not convergent and the iterates move away from the existing fixed point. Hence the extrapolating technique is not suitable since it

influences in an unacceptable way the convergence pattern of a previously convergent algorithm. It is recommended to reject it.

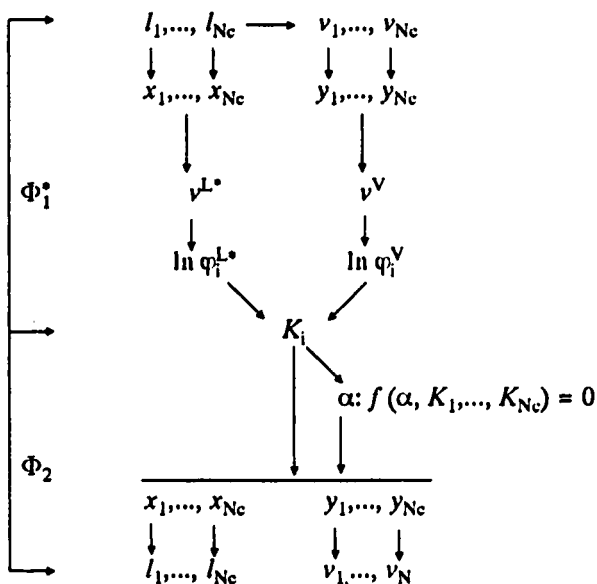
If the largest modulus eigenvalue of $M\Phi^*$ is less than one, the application of an extrapolating technique at a given iteration does not lead to a failure of the flash algorithm. However, it still could be either reliable or unreliable. A reliability guarantees that the solution of the higher level simulation algorithm, obtained after an extrapolating procedure has been triggered, is of accuracy, not significantly inferior to its solution, or is in the domain of the fixed point.

To establish whether an extrapolating model is reliable we propose as a criterion the quantity $\bar{\Delta}$:

$$\bar{\Delta} = \delta^k - \delta^{k^*} . \tag{5}$$

δ^{k^*} is the value of the iteration step, calculated after an extrapolating procedure has been used; δ^k represents the value of the "real" step, which could have been realized by the algorithm if a solution of the EOS for the volume of the phase specified had existed.

CHART 1-C



The value of δ^k could be estimated through a linear approximation, for which the following holds:

$$x^{(k+1)} = \Phi(x)^{(k)} = x^{(k)} + \frac{\partial \Phi(x)^{(k-1)}}{\partial x} (x^{(k)} - x^{(k-1)}) + r. \quad (6)$$

For the simple example of an isothermal flash algorithm, which on the k -th iteration requires an extrapolating procedure for the liquid phase, the iteration step on the independent variables l is given with:

$$\delta_l^{k*} = \Delta l = f^{(k+1)} - f^{(k)}. \quad (7)$$

The "real" step, applying Eq. (6) and neglecting the rest of the Taylor series "r", can be estimated as:

$$\delta_l^k = \Delta l_{\text{real}} = f_{\text{real}}^{(k+1)} - f^{(k)} = \frac{\partial \Phi(l)^{(k-1)}}{\partial l} (f^{(k)} - f^{(k-1)}), \quad (8)$$

where the partial derivative of the mapping Φ is taken before the extrapolating procedure is used.

The criterion then acquires the form:

$$\Delta = \delta_l^{k*} - \delta_l^k = f^{(k+1)} - f_{\text{real}}^{(k+1)}. \quad (9)$$

It is applied only once, right after an extrapolating procedure has been required by the volume-finding routine. Then if:

$$\| f^{(k+1)} - f_{\text{real}}^{(k+1)} \| < \varepsilon, \quad (9a)$$

where $\| l \| = \max (l_i), (1 < i < N_c)$, the extrapolating procedure studied is classified as reliable and is recommended to include it as a block in the structure of the higher level simulation algorithm. If

$$\| f^{(k+1)} - f_{\text{real}}^{(k+1)} \| > \varepsilon \quad (9b)$$

it is recommended to reject the procedure, since the deviation of the solution, which will be obtained at convergence, from the "real" one (fixed point), is not in the range of an acceptable tolerance.

The following relation for the upper bound ε is proposed:

$$\varepsilon = (1 - |\lambda_{\text{max}}^*|) \cdot \| (f^{(k)} - f^{(k-1)}) \|, \quad (10)$$

where $|\lambda_{\text{max}}^*|$ is the maximum modulus eigenvalue of $M\Phi^*$.

In the cases when the calculated $|\lambda_{\text{max}}^*|$ is considerably less than one, the tolerance between an iteration step, leading to the fixed point, and a step, obtained after extrapolations have been introduced and leading to an acceptable solution for the flash algo-

rithm, is given with a relatively big number. The algorithm, in such cases, is not that sensitive to volume extrapolations. In other words, even "rough" models can do the job.

To assert or dispel this conjecture, flash calculations for different multicomponent systems were tried. They were carried out at temperatures, below the pseudo-critical of the equilibrium liquid / vapor phase. It was expected that in such cases the convergence of the calculations would not be influenced in a drastic manner by the type of the extrapolating technique applied. A procedure, adopting the volume corresponding to the minimum/maximum of an isotherm, as suggested by Jovanovich et al.³, Gundersen⁵, Mills et al.¹⁰, to name a few, was tried. It is one of the most widely-used and the easiest possible to implement, since any second-order volume finding routine, at an infeasible pressure specification, stops at v_{\min}/v_{\max} . Furthermore, these values can be calculated directly from the EOS (ref.³). Here, for the sake of illustration only, the example of a flash calculation for an ethane-n-heptane mixture (mole fraction $c_2 = 0.85$) run at 422 K and $P = 50$ bar will be briefly presented.

Whenever an infeasible pressure specification is identified either for the liquid or vapor equilibrium phase, the corresponding volume value is required from the extrapolation model. It provides either $v^{L*} = v_{\min}$ or $v^{V*} = v_{\max}$. The evaluated $|\lambda_{\max}^*|$ in all cases did not exceed 0.237. As a result the flash calculations converge successfully with $\alpha = 0.9771$; $K_1 = 2.037$; $K_2 = 0.2421$; $v^L = 175.73$ cm³/mol; $v^V = 537.98$ cm³/mol.

On the other hand when $|\lambda_{\max}| \rightarrow 1$ (as in the vicinity of the true critical point, where the convergence of the algorithm is very slow), the higher level algorithm seems to be quite sensitive to the reliability of the extrapolating model applied. Flash calculations were run again, this time at temperatures, at which the equilibrium liquid phase's isotherms were not of the van der Waals type. The "rough" extrapolating strategy that was employed used the only real root that the EOS had and, in most of the cases, lead to the trivial solution. One such example is discussed in a later section.

The Bubble Point Temperature Algorithm

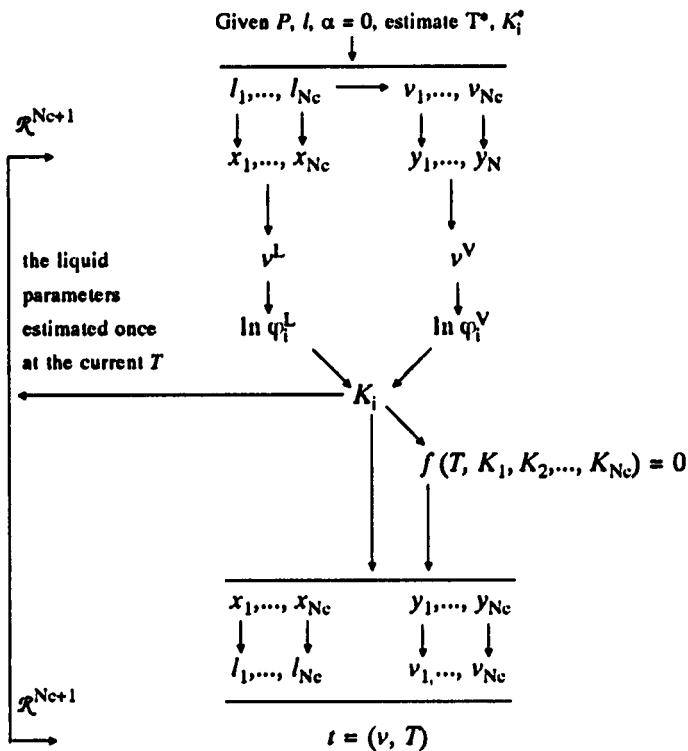
The analysis, outlined on the example of the isothermal flash calculations, can be successfully applied to test the suitability and reliability of any extrapolating technique, implemented in other higher level simulation algorithms, based on an EOS technique. Hereunder the bubble point temperature algorithm will be shortly discussed. The paradigm of the algorithm (the inner loop on the K_i is included, but at $T = \text{const}$) is given on Chart 2.

The Jacobian matrix of the mapping at the k -th iteration, where an extrapolating technique is required, is given:

$$M\Phi^* = \begin{bmatrix} \frac{\partial v_1^{(k+1)}}{\partial v_j^k} & \frac{\partial v_1^{(k+1)}}{\partial T^{(k)}} \\ \vdots & \vdots \\ \frac{\partial T^{(k+1)}}{\partial v_j^k} & \frac{\partial T^{(k+1)}}{\partial T^{(k)}} \end{bmatrix} P, \alpha = 0 \quad (11)$$

The required partial derivatives can be determined analytically: $\partial v_1^{(k+1)}/\partial v_j^k$, discussed in the first part of the paper in the form of its analogue $\partial A_1^{(k+1)}/\partial A_j^{(k)}$;

CHART 2



where t is the vector of the independent variables.

$$\frac{\partial v_i^{(k+1)}}{\partial T^{(k)}} = \left(\frac{\partial v_i}{\partial K_i} \right)_a \frac{\partial K_i}{\partial T}; \quad (11a)$$

$$\frac{\partial T}{\partial v_i} = \left(\frac{\partial T}{\partial K_i} \right) \left(\frac{\partial K_i}{\partial v_i} \right)_{a,P}; \quad (11b)$$

$$\frac{\partial T}{\partial K_i} = - \left(\frac{\partial \alpha}{\partial K_i} \right) / \left(\frac{\partial \alpha}{\partial T} \right) = \frac{z_i}{\sum z_i (\partial K_i / \partial T)}; \quad (11c)$$

$\partial K_i / \partial T$ is a well known expression;

$$\frac{\partial T^{(k+1)}}{\partial T^{(k)}} = \sum \left(\frac{\partial T}{\partial K_m} \right) \left(\frac{\partial K_m}{\partial T} \right). \quad (11d)$$

Once the partial derivatives (11a – 11d) are obtained, the influence of an extrapolating model on the convergence characteristics of the bubble point temperature calculations can be studied successfully, following the algorithm of the proposed general analysis.

RESULTS

Equilibrium calculations for a widely used typical natural gas mixture are run to demonstrate how an extrapolating model, incorporated in the isothermal vapor–liquid flash algorithm, is evaluated from the view point of the two important characteristics, discussed in the paper. The composition of the mixture (in mole fractions) is given in the corresponding tables.

The thermodynamic model used is the SRK CEOS. The binary interaction parameters are the recommended k_{ij} given in DECHEMA Chemistry Data Series¹⁷: $k_{12} = 0.0278$; $k_{13} = 0.0407$; $k_{14} = 0.0763$; $k_{15} = 0.08$; $k_{16} = 0.0878$; $k_{17} = 0.1496$; the rest k_{ij} are set equal to zero. The extrapolating model analysed is the one suggested by Stateva et al.¹². The following points are studied:

Point A. $T = 230$ K, $P = 70$ atm.

The point lies inside the two phase boundary. Infeasible pressure specifications are recognized at the 4-th iteration for the liquid phase and the block, containing the extrapolating model, is triggered. A “pseudovolume” v^{L^*} and the “pseudofugacity” coefficients $\ln \varphi^{L^*}$ are provided. Then the Jacobian matrix $M^{(l)^*}$ is formed and its eigenvalues are calculated. The following information is obtained:

- the extrapolating procedure is suitable, since $|\lambda_{\max}^*| = 0.482$;
- the extrapolating procedure is reliable, since the recommended value of ϵ , estimated according to relation (10), is $5 \cdot 10^{-4}$ and $\|f^{(k+1)} - f_{\text{real}}^{(k+1)}\|$ satisfies condition (9a), where l are given in mole numbers (Table I). The algorithm converges at the 9-th iteration.

Point B. $T = 204$ K, $P = 58$ atm.

This specification lies in the immediate vicinity of the true critical point of the mixture, reported by Michelsen¹⁶ to be $T_c = 203.125$ K, $P_c = 58.108$ atm. After the extrapolating routine has been executed once, the following results are obtained: $|\lambda_{\max}^*| = 0.992$; the recommended $\epsilon = 3 \cdot 10^{-6}$. The values of $f^{(k+1)}$ and $f_{\text{real}}^{(k+1)}$, at $k = 11$, and x and K at convergence, are summarised in Table II.

When, as a comparison, a simple extrapolating technique, adopting the only root of the EOS as the volume required, was applied, the flash calculations did not converge – the iterates moved away from the solution.

Point C. $T = 204$ K, $P = 57$ atm.

TABLE I

Convergence behavior of the flash algorithm at $T = 230$ K, $P = 70$ atm. $k = 4$, $\alpha_{\text{conv}} = 0.9896$, total number of iterations 9

Atom	z	$f^{(k+1)}$	$f_{\text{real}}^{(k+1)}$	x_{conv}	K_{conv}
N ₂	0.014	$2.9419 \cdot 10^{-5}$	$2.1573 \cdot 10^{-5}$	0.003125	4.5155
C ₁	0.943	$5.827671 \cdot 10^{-3}$	$5.629238 \cdot 10^{-3}$	0.600657	1.5759
C ₂	0.027	$7.55545 \cdot 10^{-4}$	$6.13921 \cdot 10^{-4}$	0.075738	0.3497
C ₃	0.0074	$5.79943 \cdot 10^{-4}$	$6.88113 \cdot 10^{-4}$	0.057284	0.1200
nC ₄	0.0049	$9.62287 \cdot 10^{-4}$	$9.18990 \cdot 10^{-4}$	0.093739	0.0423
nC ₅	0.0027	$1.109944 \cdot 10^{-3}$	$1.083512 \cdot 10^{-3}$	0.106995	0.0149
nC ₆	0.001	$6.50674 \cdot 10^{-4}$	$4.91180 \cdot 10^{-4}$	0.062462	0.0056

TABLE II

Convergence behavior of the flash algorithm at $T = 204$ K, $P = 58$ atm. $k = 9$, $\alpha_{\text{conv}} = 0.7972$, total number of iterations 38

Atom	z	$f^{(k+1)}$	$f_{\text{real}}^{(k+1)}$	x_{conv}	K_{conv}
N ₂	0.014	$2.16477 \cdot 10^{-4}$	$2.16452 \cdot 10^{-4}$	0.009470	1.5998
C ₁	0.943	$2.6261023 \cdot 10^{-2}$	$2.6261023 \cdot 10^{-2}$	0.903721	1.0545
C ₂	0.027	$1.786898 \cdot 10^{-3}$	$1.786852 \cdot 10^{-3}$	0.043426	0.5256
C ₃	0.0074	$8.77231 \cdot 10^{-4}$	$8.77252 \cdot 10^{-4}$	0.016371	0.3127
nC ₄	0.0049	$9.81613 \cdot 10^{-4}$	$9.81613 \cdot 10^{-4}$	0.013887	0.1883
nC ₅	0.0027	$8.60667 \cdot 10^{-4}$	$8.60593 \cdot 10^{-4}$	0.009236	0.1124
nC ₆	0.001	$4.55975 \cdot 10^{-4}$	$4.55991 \cdot 10^{-4}$	0.003889	0.0683

The flash algorithm, after the "extrapolating" block is triggered several times for the liquid phase, converges successfully. The results are shown in Table III. The same point was tried with different initial estimates and as a result the algorithm converged, no extrapolations required. The deviation between the solution, obtained after the "extrapolating procedure" has been applied and the one, obtained without such intervention, is in the range of the tolerance, required by the flash algorithm itself.

TABLE III
Convergence behavior of the flash algorithm at $T = 204$ K, $P = 57$ atm. $\alpha_{\text{convg}} = 0.8260$, total number of iterations 53

Atom	z	x_{convg}	K_{convg}
N ₂	0.014	0.008128	1.8741
C ₁	0.943	0.8845	1.08
C ₂	0.027	0.050859	0.4432
C ₃	0.0074	0.020884	0.2185
nC ₄	0.0049	0.018375	0.1123
nC ₅	0.0027	0.012216	0.0570
nC ₆	0.001	0.005038	0.0297

CONCLUSION

The paper has discussed and provides a direct way of estimating quantitatively the influence the "extrapolated values" of the thermophysical properties (required at infeasible pressure specifications, which are often occurring when solving engineering design problems) have on the convergence characteristics of higher level simulation algorithms.

The proposed strategies have been implemented in as a part of a general process simulator "EQUILIBRIA"¹⁸. The numerical performance which has been tested on a variety of systems and applications has shown that the computational requirements of the method are quite reasonable, since it includes standard numerical procedures (matrices and vector-matrix multiplication, eigenvalues estimation, etc.), providing a high degree of reliability for the conclusions obtained. The partial derivatives are given as analytical expressions and their estimation requires simple operations only.

The suggested general analysis can be successfully applied to other steady-state simulation/design calculations where the isothermal vapor-liquid flash, bubble point temperature and analogous algorithms (like the isenthalpic flash calculation, the bubble point pressure algorithm, etc.) are incorporated as a constituent part.

SYMBOLS

f_i	fugacity of the i -th component
F	total number of moles in the feed
k_{ij}	binary interaction coefficients for the RKS CEOS
K_i	equilibrium ratio of the i -th component
l_i	mole numbers of the i -th component in the liquid phase, $l_i = x_i(1 - \alpha)$
L	total number of moles in the liquid phase, $L = \sum l_i$
M	Jacobian matrix
N_C	total number of components
P	specified pressure
r	residual of the Taylor series
\mathcal{R}^n	n -dimensional space
T	temperature
v	molar volume of the liquid-vapor phase
v_i	mole numbers of the i -th component in the vapor phase, $v_i = y_i \alpha$
V	total number of moles in the vapor phase, $V = \sum v_i$
x_i	mole fraction of the i -th component in the liquid phase
y_i	mole fraction of the i -th component in the vapor phase
z_i	number of moles of the i -th component in the feed, $z_i = l_i + v_i$
α	vapor mole fraction, $\alpha = V/F$
γ	activity coefficient
Δ	change in appropriate variable
$\bar{\Delta}$	criterion, Eq. (5)
δ_{ij}	Kronecker delta
δ^k	correction to $f^{(k)}$ at iteration k
ϵ	recommended value, Eq. (10)
λ	eigenvalue
Φ	a mapping
φ_i	fugacity coefficient of the i -th component
Superscripts	
k	number of iterations
L	liquid
V	vapor
\circ	extrapolated
Subscripts	
convg	value of a parameter at convergence
max	maximal
min	minimal
real	true value (obtained without extrapolations)

REFERENCES

1. Westerberg A. W., Hutchinson H. P., Motard R. L., Winter P.: *Process Flowsheeting*. Cambridge University Press, Cambridge 1979.
2. Heidemann R. A.: *Fluid Phase Equilib.* 14, 55 (1983).
3. Jovanovic St., Pounovich R.: *Ind. Eng. Chem., Process Des. Dev.* 23, 801 (1984).
4. Ammar M. N., Rezon H.: *AIChE J.* 33, 926 (1987).

5. Michelsen M. L.: *Fluid Phase Equilib.* **4**, 1 (1980).
6. Michelsen M. L.: *Fluid Phase Equilib.* **9**, 1 (1982).
7. Huron M. J., Dufour G. N., Vidal J.: *Fluid Phase Equilib.* **1**, 247 (1978).
8. Gundersen T.: *Comput. Chem. Eng.* **6**, 245 (1982).
9. Mathias P. M., Boston J. F., Watanasiri S.: *AIChE J.* **30**, 182 (1984).
10. Mills M. B., Wills M. J., Bhirud V. L.: *AIChE J.* **26**, 902 (1980).
11. Polling B. E., Grens E. A., Prausnitz J. M.: *Ind. Eng. Chem., Process Des. Dev.* **20**, 127 (1981).
12. Stateva R. P., Tsvetkov S. G., Boyadjiev Ch. B.: *AIChE J.* **36**, 132 (1990).
13. Fletcher R.: *Practical Methods for Optimization*. Wiley Interscience, New York 1987.
14. Ohanomah M. O., Thompson D. W.: *Comput. Chem. Eng.* **8**, 147 (1984).
15. Michelsen M. L.: *SEP 8109*, Institutet for Kemiteknik (1981).
16. Michelsen M. L.: *Fluid Phase Equilib.* **9**, 21 (1982).
17. Knapp H., Oellrich L., Plocker U., Prausnitz J. M.: *DECHEMA Chemistry Data Series*, Vol. VI. DECHEMA, Frankfurt/Main 1982.
18. Bancheva I., Stateva R. P. in: *Proc. Mediterranean Congress on Chem. Eng.* **II**, 540. Barcelona 1987.

APPENDIX

The equivalence of the matrices S and $M\Phi$ is proved as follows: $S = \Phi D$, where:

$$\Phi_{ij} = \left(\frac{\partial \ln q_i^L}{\partial n_j} + \frac{\partial \ln q_i^V}{\partial n_j} \right)$$

$$d_{ij} = VL (x_i y_i / z_i) [\delta_{ij} + (x_j y_j / z_j) / s]$$

$$s = 1 - \sum (x_k y_k / z_k)$$

Analysis of $M\Phi = M\Phi_2 \cdot M\Phi_1$ shows that:

(i) The rows of $M\Phi_1$ (see Eq. (2)) are proportional to the rows of the matrix Φ , with a proportionality coefficient K_i ;

(ii) The columns of $M\Phi_2$ are proportional to the columns of matrix D with a proportionality coefficient $1/K_j$. Furthermore, the difference between the values of the elements of $M\Phi_2$ and the elements of the matrix D is given by the difference between s_1 and s .

When $f(\alpha, K)$ is the Rachford-Rice form, the expression for s can be rewritten as:

$$s = 1 - \sum \frac{z_i K_i}{[1 + \alpha (K_i - 1)]^2} = \sum z_i - \sum \frac{z_i K_i}{[1 + \alpha (K_i - 1)]^2} =$$

$$= \sum \frac{z_i (K_i - 1)}{[1 + \alpha (K_i - 1)]^2} [\alpha^2 (K_i - 1) + 2\alpha - 1] =$$

$$\begin{aligned}
&= \sum \frac{z_i (K_i - 1)}{[1 + \alpha (K_i - 1)]^2} \{ \alpha [\alpha (K_i - 1) + 1] - \alpha + 2\alpha - 1 \} = \\
&= \sum \frac{z_i (K_i - 1) \alpha}{[1 + \alpha (K_i - 1)]} - (1 - \alpha) \sum \frac{z_i (K_i - 1)}{[1 + \alpha (K_i - 1)]^2} = \\
&= - (1 - \alpha) \sum \frac{z_i (K_i - 1)}{[1 + \alpha (K_i - 1)]^2} \tag{A1}
\end{aligned}$$

Hence $s = -s_1$. The same result is obtained when $f(\alpha, K)$ is the Barnes–Flores form¹⁴. Taking into consideration the above derivations and performing the multiplication of $M\Phi_1 \cdot M\Phi_2$, proves the equivalence of matrices $M\Phi$ and S .