

# INFLUENCE OF THERMOPHYSICAL PROPERTIES EXTRAPOLATION, IN THE COURSE OF ITERATIVE CALCULATIONS, ON THE CONVERGENCE PATTERN OF A SIMULATION ALGORITHM—A GENERAL ANALYSIS

ROUMIANA P. STATEVA AND STEFAN G. TSVETKOV

*Institute of Chemical Engineering Bulgarian Academy of Sciences Acad.  
Georgi Bontchev str., b1.103 Sofia 1113 BULGARIA*

**Key Words:** Process Simulation, Algorithm Convergence, Thermophysical Properties, Contraction Mapping Principle

An extrapolating model provides values for the 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 general approach for evaluating and comparing the abilities of different extrapolation techniques: to promote the convergence of higher-level algorithms; to ensure for them a solution that is accurate and not significantly inferior to the actual one. The latter could have been obtained using for example a different thermodynamic model.

The approach is based on the classical fixed-point theory and its connection with the contraction-mapping principle.

## Introduction

High-level algorithms for steady-state simulation and design, based on an equation of state (EOS) technique, use a hierarchical structure of organization. The lower level includes the EOS model, which incorporates a very important routine. It solves the EOS for the volume corresponding to specified pressure and phase of interest. Then the EOS model returns a valid volume value (and the derived thermodynamic properties) to the higher-level algorithm. During phase-equilibrium calculations, however, combinations of pressure, temperature and composition may arise for which there is no valid volume root according to the EOS used. If care is not taken to circumvent this problem, it will lead to a trivial solution for the algorithm. The trivial solution interrupts the performance of simulation algorithms and spoils their convergence. Such situations can be dealt with either by higher-level algorithms or by the EOS model.

The second approach has proved to be more reliable. The EOS model is organized to allow for a specified phase: to recognize infeasible specifications (usually that of the pressure) and to supply values for the volume and the derived properties under all conditions. The latter is realized through a so-called "extrapolating procedure", interwoven in the struc-

ture of the EOS model.

Examples of extrapolating models are extensively discussed in the literature<sup>4,6,8,12,14,15</sup>, to name just a few references. A graphical representation of the performance of one, suggested by Stateva *et al.*<sup>15</sup>, at infeasible pressure specification for the liquid phase, is demonstrated in Fig. 1.

The extrapolated volume and derived thermophysical properties should satisfy the following requirements: to follow the tendency of the real values for the phase specified; to be continuous and differentiable; and to

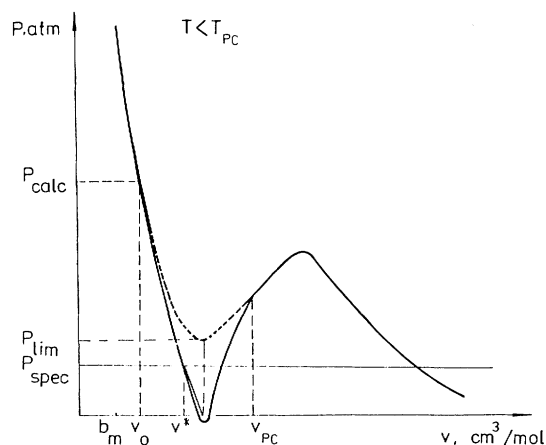


Fig. 1.  $P$ - $v$  isotherm (—) simulated by the extrapolating technique<sup>15</sup> for the liquid phase at  $T < T_{pc}$ . The infeasible pressure specification ( $P_{spec}$ ) is transformed to a feasible one for the EOS used

improve the convergence characteristics of the higher-level simulation algorithms<sup>8</sup>).

It is conceivable that satisfying this last requirement is decisive for the robustness of higher-level simulation algorithms. However, the papers that advocate new extrapolating techniques provide little knowledge of their strength and limitations in this respect. Hence, a detailed examination of the problem is justifiable and quite useful.

To facilitate it, a general approach, designed to predict in a quantitative manner the influence an extrapolating technique exercises on a higher-level algorithm, is presented in this paper. Its performance is demonstrated with an example of isothermal two-phase vapor-liquid flash, which can be treated as a simple higher-level simulation algorithm. It is followed by a brief discussion of a bubble-point temperature algorithm as an example.

The same analysis can be successfully applied to other steady-state simulation/design calculations where the above and analogous algorithms (like the isenthalpic flash calculation, the bubble-point pressure algorithm, etc.) are incorporated as constituent parts.

### Outline of the Approach

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 explanation more readable, firstly some general considerations will be given.

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

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

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

Then if a *fixed-point* iterative sequence

$$x^{(k+1)} = \Phi(x^{(k)}) \quad (1)$$

is constructed, its points of accumulation satisfy the equation

$$x^* = \Phi(x^*) \quad (1a)$$

where  $x^*$  are the *fixed points* of the mapping  $\Phi$ .

To show that the fixed-point iteration defined by Eq. 1 converges, the *contraction mapping principle*<sup>4</sup> is applied.

Let  $M\Phi$  be the Jacobian matrix of  $\Phi$  and  $\lambda_i$ —an eigenvalue of  $M\Phi$ . Then if the condition

$$\max_{1 \leq i \leq n} |\lambda_i(M\Phi)| < 1 \quad (1b)$$

is satisfied,  $\Phi$  is a *contraction mapping*<sup>3</sup>.

Table 1-A.

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

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

and go back to step 2.

The equation  $f(\alpha, K_1, K_2, \dots, K_{N_c}) = 0$  might be any of the known formulations of the single-loop univariable methods for solving the flash problem, as discussed by Ohanomah and Thompson<sup>13</sup>. The accepted  $\Delta_{err} = 10^{-12}$ .

Consequently, the iteration defined by Eq. 1 converges to a fixed point, as stated by the *contraction mapping theorem*<sup>3</sup>, which can be found in many other literature references.

The contraction-mapping principle can be applied to any computational algorithm to establish whether it converges. To elucidate the ideas, the well-known *isothermal vapor-liquid flash algorithm* will be used as an example of a simple higher-level simulation algorithm in the present study.

The choice of the vapor-liquid (liquid-liquid) flash algorithm is motivated by the following considerations. Firstly, it is not only frequently used in steady-state simulation, but is a constituent part of other important higher-level algorithms; secondly, it contains a volume-finding routine. As discussed previously, it is where an extrapolating procedure might be required.

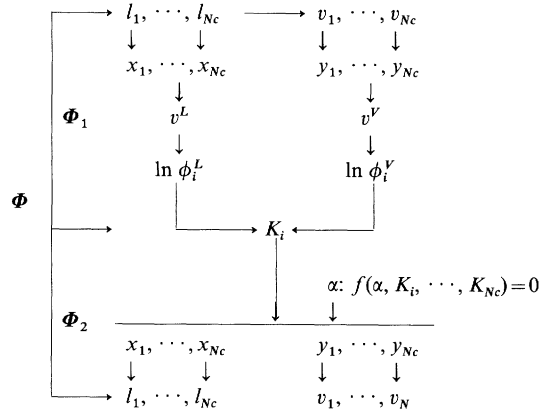
Furthermore, the method of solution and organization of the flash algorithm does not influence in any way the generality of the ideas presented further on. That is why the authors have accepted the univariable single-loop organization<sup>13</sup>, which has proved to be the most widely used.

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

Let one mole of the feed be flashed and let  $l_i$  be the independent variables. The assumption that  $F=1$  is for simplification only and does not influence the generality of the conclusions which follow. In that case  $L=1-\alpha$  and  $V=\alpha$ .

It is convenient to represent the mapping  $\Phi$  of the flash algorithm as a composition of two mappings (as shown on Table 1-B) and its Jacobian matrix  $M\Phi$  as

Table 1-B.



a product of the Jacobian matrices of  $\Phi_1$  and  $\Phi_2$  mappings:

$$\Phi = \Phi_2 \circ \Phi_1 \quad \text{and} \quad M\Phi = M\Phi_2 * M\Phi_1 \quad (2)$$

The operation "composition" is denoted by the symbols  $\circ$  in the above expression.

$\Phi_1: l \rightarrow K$  is the mapping, where extrapolations might occur;

$$\Phi_2: K \rightarrow l, \quad (l \in \mathbb{R}^n, K \in \mathbb{R}^n).$$

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

$$\frac{\partial K_i}{\partial l_j} = \frac{\partial (\exp(\ln \phi_i^L - \ln \phi_i^V))}{\partial l_j} = K_i \left( \frac{\partial \ln \phi_i^L}{\partial l_j} + \frac{\partial \ln \phi_i^V}{\partial v_j} \right) \quad (3)$$

The partial derivatives of the fugacity coefficients are well-known and widely discussed analytical expressions, e.g. in<sup>9</sup>.

The Jacobian matrix of the second mapping is  $M\Phi_2 = (\partial l_i / \partial K_j)$ .

Elements of  $M\Phi_2$  are given by:

$$m_{ij} = \alpha(1 - \alpha) \left[ \left( -\frac{x_i y_i}{z_i K_j} \right) \delta_{ij} + \frac{x_i y_i / z_i (\partial f / \partial K_j)}{\alpha(1 - \alpha) \partial f / \partial \alpha} \right] \quad (4)$$

Their derivation is demonstrated in Appendix I.

The elements of  $M\Phi_2$  take concrete form when the specific formulation of  $f(\alpha, K)$  is differentiated and introduced in Eq. 4 (Appendix II). If, for example,  $f(\alpha, K)$  is the Rachford-Rice formulation:

$$m_{il} = \alpha(1 - \alpha) \frac{x_i y_i}{z_i} \frac{1}{K_j} \left( -\delta_{ij} + \frac{x_j y_j / z_j}{s_1} \right) \quad (4a)$$

$$s_1 = (1 - \alpha) \sum \frac{z_i (K_i - 1)}{[1 + \alpha(K_i - 1)]^2} \quad (4b)$$

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  $\Phi$  must be *contractive*.

Appendix III demonstrates that, in the case of either the *Rachford-Rice* or *Barnes-Flores* formulations of the single-loop univariable method,  $M\Phi$  and a matrix  $S^{(11)}$  are equivalent.

The matrix  $S$  is:

$$\begin{aligned} S_{ij} &= \frac{\partial \ln K_i^{(t+1)}}{\partial \ln K_j^{(t)}} = \frac{\partial}{\partial \ln K_j^{(t)}} (\ln \phi_i^L - \ln \phi_i^V) \\ &= \sum_{m=1}^N \left( \frac{\partial \ln \phi_i^L}{\partial n_m} + \frac{\partial \ln \phi_i^V}{\partial n_m} \right) \frac{\partial l_m}{\partial \ln K_j} \end{aligned} \quad (5)$$

Michelsen<sup>11</sup> has shown that the largest modulus eigenvalue of  $S$  is less than one in the domain of vapor-liquid phase coexistence. Since matrices  $S$  and  $M\Phi$  are equivalent, they share the same eigenvalues. Hence, the largest modulus eigenvalue of  $M\Phi$  is less than one as well. Consequently, the mapping  $\Phi$  is contractive.

### Analysis of the Convergence Behavior of the Isothermal Flash Algorithm after an Extrapolating Procedure is Applied

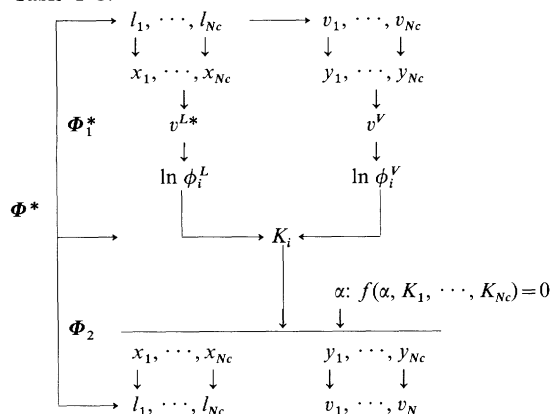
This section demonstrates how the *contraction-mapping principle* can be used to analyze the convergence behavior of a higher-level algorithm after an extrapolating procedure has been applied at some iteration. In particular, to clarify whether an extrapolating model is:

i) *suitable*, i.e. promotes convergence; ii) *reliable*, i.e. ensures for a higher-level simulation algorithm a solution, which will be accurate and not significantly inferior to the "real" one. The "real" solution of the algorithm is its fixed point. It might be obtained either applying a mixed thermodynamic model (e.g.  $\tau$ - $\phi$  method) or using a wisely chosen set of initial estimates (e.g. obtained from stability analysis of the system to be flashed).

The primary implication of the discussion in the previous section is that any extrapolating procedure can be represented as a mapping. Furthermore, it could be included as a constituent part in  $\Phi_1$ , instead of the volume-finding routine. Thus,  $\Phi_1$  will be transformed to  $\Phi_1^*$ . The paradigm of the new composite mapping  $\Phi^*$  is shown on Table 1-C.

In the language of the information flow organization of a higher-level algorithm, the above transformation reflects the following situation: At a given iteration step  $k$ , for example for the current values of  $x/l^{(k)}$ , the volume-finding procedure fails to find a root of the EOS, corresponding to the equilibrium liquid phase. Then the extrapolating technique is triggered. It provides values for the pseudovolume  $v^{L*}$  and the pseudofugacity coefficients  $\ln \phi^{L*}$ . They are then returned through the volume-finding routine to the higher-level algorithm. It uses them further on to estimate  $K^{(k)}$ ,  $\alpha$ ,  $l^{(k+1)}$ . However, it is not known

Table 1-C.



beforehand whether this will return the algorithm to its normal pattern of convergence, *i.e.* whether its mapping will be contractive.

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

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

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

where the elements of the matrix  $M\Phi_2^*$  (Eq. 4) are calculated *after* the extrapolating procedure has been introduced; —the elements of the Jacobian matrix  $M\Phi_1^*$  (Eq. 3) are calculated, using  $l^{(k)}$ ,  $v^{(k)}$ , the “pseudovolume”  $v^L$  for the liquid phase and the vapor-phase volume  $v^V$ .

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

If the largest modulus eigenvalue of  $M\Phi^*$  is greater than one,  $\Phi^*$  is not contractive. This means that the algorithm will not converge since the iterates move away from the existing fixed point. Consequently, the extrapolating technique applied is not suitable. It influences in an unacceptable way the convergence pattern of a *previously convergent algorithm* and it is recommended to reject it. A simplified graphical representation of the possible patterns of convergence of an algorithm, before and after extrapolations are applied, is given on Fig. 2.

A suitable extrapolating technique, however, could still be either reliable or unreliable. Reliability guarantees that the solution (obtained after extrapolations have been introduced) is accurate and not significantly inferior to the *real solution* of the

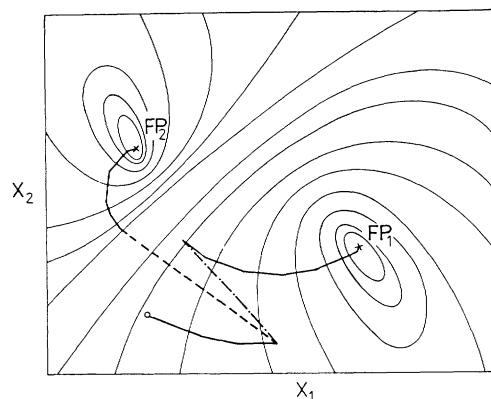


Fig. 2. A simplified graphical representation of the possible patterns of convergence of an algorithm, after an extrapolation is applied

$FP_1$ —the true solution;  $FP_2$ —the trivial solution or any other solution (fixed point) which does not satisfy the physical requirements of the problem solved; —, an iteration without extrapolation; — — —, a suitable and reliable extrapolation; ·····, an unacceptable extrapolation

algorithm, or is in the domain of its fixed point (Fig. 2).

To establish whether an extrapolating model is reliable the following criterion is used:

$$\tilde{\Delta} = \|\delta^k - \delta^{k*}\| < \varepsilon \quad (6)$$

$\delta^k$  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;  $\delta^{k*}$  is the value of the iteration step, calculated after an extrapolating procedure has been used.

The value of  $\delta^k$  can be estimated through a linear approximation, for which the following holds:

$$x^{(k+1)} = \Phi(x^{(k)}) = x^k + \frac{\partial \Phi(x^{(k-1)})}{\partial x} (x^{(k)} - x^{(k-1)}) + r \quad (7)$$

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 by:

$$\delta_1^{k*} = \Delta l = l^{(k+1)} - l^{(k)} \quad (8)$$

The “real” step, applying Eq. 7 and neglecting the rest of the Taylor series  $r$ , can be estimated as:

$$\delta_1^k = \Delta l_{\text{real}} = l_{\text{real}}^{(k+1)} - l^{(k)} = \frac{\partial \Phi(l^{(k-1)})}{\partial l} (l^{(k)} - l^{(k-1)}) \quad (9)$$

where the partial derivative of the mapping  $\Phi$  is taken before the extrapolating procedure is used.

The criterion then acquires the form:

$$\tilde{\Delta} = \|\delta_1^{k*} - \delta_1^k\| = \|l^{(k+1)} - l_{\text{real}}^{(k+1)}\| \quad (10)$$

where  $\|l\| = \max_{1 \leq i \leq N_c} (l_i)$ .

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

If:

$$\tilde{A} < \varepsilon \quad (10a)$$

the extrapolating procedure studied is classified as reliable. However, if:

$$\tilde{A} > \varepsilon \quad (10b)$$

it is recommended to reject the procedure, because the deviation of the solution (obtained at convergence) from the fixed point is not in the range of acceptable tolerance.

The upper bound  $\varepsilon$  is calculated accordingly:

$$\varepsilon = (1 - |\lambda_{\max}^*|) * \|(\mathbf{I}^{(k)} - \mathbf{I}^{(k-1)})\| \quad (11)$$

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

In the cases when the calculated  $|\lambda_{\max}^*|$  is considerably less than one it is expected that the higher-level simulation algorithm is not that sensitive to volume extrapolations. In other words, even "rough" models can do the job.

To confirm or to dismiss 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, where the  $P(v)$  relationship is of the van der Waals type. It was supposed 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 Gundersen<sup>4)</sup>, Jovanovich *et al.*<sup>6)</sup>, Mills *et al.*<sup>12)</sup>—to name a just a few, is considered. 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  $v_{\min}/v_{\max}$ <sup>15)</sup>. Furthermore, these values can be calculated directly from the EOS, as suggested in<sup>6)</sup>.

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.

Firstly, the calculations are run uninterruptedly till the algorithm converges. Whenever an infeasible pressure specification is identified at an EOS level (either for the liquid or vapor equilibrium phase) values for the volume and the fugacity coefficients are required from the extrapolation model. It provides either  $v^{*L} = v_{\min}$  or  $v^{*V} = v_{\max}$ . At convergence the following results are obtained:  $\alpha = 0.9771$ ;  $K_1 = 2.037$ ;  $K_2 = 0.2421$ ;  $v^L = 175.73 \text{ cm}^3/\text{mol}$ ;  $v^V = 537.98 \text{ cm}^3/\text{mol}$ .

Table 2.

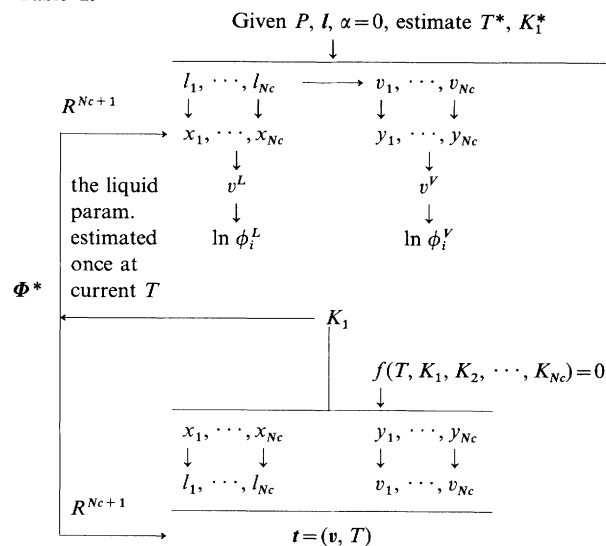


Table 3.

	$z$	$l^{(k+1)}$	$l_{\text{real}}^{(k+1)}$	$l_{\text{converged}}$	$x_{\text{convg}}$	$K_{\text{convg}}$
$N_2$	0.014	$2.9419 \cdot 10^{-5}$	$2.1573 \cdot 10^{-5}$	$3.25 \cdot 10^{-5}$	0.003125	4.5155
$C_1$	0.943	$5.827671 \cdot 10^{-3}$	$5.629238 \cdot 10^{-3}$	$6.2468 \cdot 10^{-3}$	0.600657	1.5759
$C_2$	0.027	$7.55545 \cdot 10^{-4}$	$6.13921 \cdot 10^{-4}$	$2.8767 \cdot 10^{-4}$	0.075738	0.3497
$C_3$	0.0074	$5.79943 \cdot 10^{-4}$	$6.88113 \cdot 10^{-4}$	$5.9575 \cdot 10^{-4}$	0.057284	0.1200
$nC_4$	0.0049	$9.62287 \cdot 10^{-4}$	$9.18990 \cdot 10^{-4}$	$9.7488 \cdot 10^{-4}$	0.093739	0.0423
$nC_5$	0.0027	$1.109944 \cdot 10^{-3}$	$1.083512 \cdot 10^{-3}$	$1.1127 \cdot 10^{-3}$	0.106995	0.0149
$nC_6$	0.001	$6.50674 \cdot 10^{-4}$	$4.91180 \cdot 10^{-4}$	$6.4960 \cdot 10^{-4}$	0.062462	$5.6 \cdot 10^{-3}$

$k=4$ ;  $\alpha_{\text{convg}}=0.9896$ ; total number of iterations=9.

$l$  value are given in mole numbers,  $x$ —in mole fractions.

Table 4.

	$z$	$l^{(K+1)}$	$l_{\text{real}}^{(K+1)}$	$x_{\text{convg}}$	$K_{\text{convg}}$
$N_2$	0.014	$2.16477 \cdot 10^{-7}$	$2.16452 \cdot 10^{-4}$	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

$k=9$ ;  $\alpha_{\text{convg}}=0.7972$ ; total number of iterations=38.

$$M\Phi^* = \begin{bmatrix} \frac{\partial v_i^{(k+1)}}{\partial v_j^{(k)}} & \frac{\partial v_i^{(k+1)}}{\partial T^{(k)}} \\ \frac{\partial T^{(k+1)}}{\partial v_j^{(k)}} & \frac{\partial T^{(k+1)}}{\partial T^{(k)}} \end{bmatrix}_{P, \alpha=0} \quad (12)$$

The partial derivatives can be determined analytically:  $\partial v_i^{(k+1)}/\partial v_j^{(k)}$ , discussed in the first part of the paper in the form of its analogue  $\partial l_i^{(k+1)}/\partial l_j^{(k)}$ ,

$$\frac{\partial v_i^{(k+1)}}{\partial T^{(k)}} = \left( \frac{\partial v_i}{\partial K_i} \right) \left( \frac{\partial K_i}{\partial T} \right); \quad (12a)$$

$$\frac{\partial T}{\partial v_i} = \left( \frac{\partial T}{\partial K_i} \right) \left( \frac{\partial K_i}{\partial v_i} \right)_{\alpha, P} \quad (12b)$$

$$\frac{\partial T}{\partial K_j} = - \left( \frac{\partial \alpha}{\partial K_j} \right) / \left( \frac{\partial \alpha}{\partial T} \right) = \frac{z_j}{\sum z_i (\partial K_i / \partial T)} \quad (12c)$$

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

$$\frac{\partial T^{(k+1)}}{\partial T^{(k)}} = \sum \left( \frac{\partial T}{\partial K_m} \right) \left( \frac{\partial K_m}{\partial T} \right) \quad (12d)$$

In the case where  $T=\text{const}$  in the inner loop on  $K_i$ , the relations applied are equivalent, but more simple (since  $\alpha=0$ ) than those mentioned above for the isothermal vapor-liquid flash.

Once the partial derivatives (12a–12d) are obtained, the steps of the proposed general analysis can easily be followed.

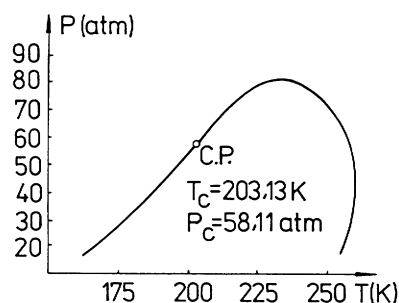


Fig. 3.  $P$ - $T$  diagram of a seven-component typical natural gas mixture

## Examples

Equilibrium calculations for the widely used example of a typical natural gas mixture<sup>1,10,11</sup> are run. The purpose is to demonstrate how an extrapolating model can be evaluated from the viewpoint of the two important characteristic discussed in the paper. The  $P$ - $T$  diagram of the mixture is shown on Fig. 3, and its composition (in mole fractions) in Tables 3, 4 and 5.

The thermodynamic model used is the Soave-Redlich-Kwong CEOS. The binary interaction parameters are the recommended  $k_{ij}$  given in DECHEMA Chemistry Data Series<sup>7</sup>:  $k_{12}=0.0278$ ;  $k_{13}=0.0407$ ;  $k_{14}=0.0763$ ;  $k_{15}=0.08$ ;  $k_{16}=0.0878$ ;  $k_{17}=0.1496$ ; the rest of the  $k_{ij}$  values are set equal to zero. The extrapolating model analysed is the one suggested by Stateva *et al.*<sup>15</sup>.

The following points are studied:

### Point A. $T=230\text{ K}$ , $P=70\text{ atm}$

The point lies inside the two-phase boundary. Infeasible pressure specifications are recognized at the 4th iteration for the liquid phase. The block, containing the extrapolating model, is triggered. It provides values for the “pseudovolume”  $v^{L*}$  and “pseudofugacity” coefficients  $\ln \phi^{L*}$ . The Jacobian matrix  $M\Phi^*$  is formed and its eigenvalues are calculated. The conclusion is that:

- the extrapolating procedure is suitable, since  $|\lambda_{\max}^*|=0.482$ ;
- the extrapolating procedure is reliable, since the recommended value of  $\varepsilon$ , estimated according to relation (11), is  $5.10^{-4}$  and  $\tilde{A}=\|I^{(k+1)}-I_{\text{real}}^{(k+1)}\|$  satisfies condition (10a).

The algorithm converges at the ninth iteration. Its performance characteristics, including the values of  $|\lambda_{\max}|$  at each iteration and the profile of the convergence criterion  $\Delta_{\text{err}}$ , are given in Figs. 4a, 4b. Table 3 summarizes some additional information for the calculations performed.

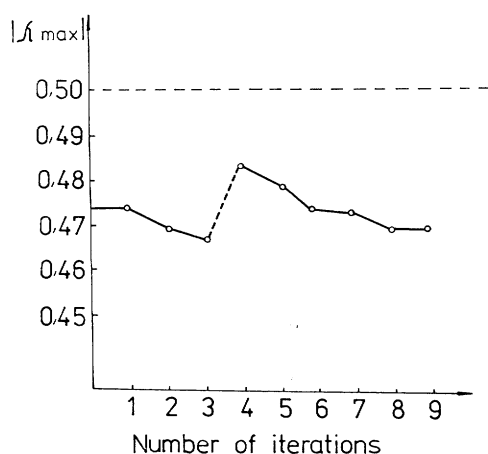
### Point B. $T=204\text{ K}$ , $P=58\text{ atm}$

This specification lies in the immediate vicinity of the true critical point of the mixture. After the extrapolating routine has been executed once, the

Table 5.

	$z$	$x_{\text{conv}}$	$K_{\text{conv}}$
$N_2$	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

$\alpha_{\text{conv}}=0.8260$ ; total number of iterations = 53.



following results are obtained:  $|\lambda_{\max}^*|=0.992$ ; the upper bound  $\varepsilon=3.10^{-6}$ . The values of  $I^{(k+1)}$  and  $I_{\text{real}}^{(k+1)}$  ( $k=11$ ), and  $x$  and  $K$  at convergence, are given in Table 4. The performance characteristics of the algorithm are presented in Figs. 5a, 5b.

For comparison, a simple extrapolating technique, adopting the only root of the EOS, was applied. The flash calculations did not converge—the iterates moved away from the solution (Figs. 5a, b).

### Point C. $T=204\text{ K}$ , $P=57\text{ atm}$

The flash algorithm, after the “extrapolating” block is triggered three times, converges successfully. The results are shown in Table 5, while the convergence history of the calculations—on Figs. 6a and 6b.

The same point was tried with different initial estimates and as a result the algorithm converged, with no extrapolations required. The deviation between the solution in Table 5 and that obtained without extrapolations is in the range of tolerance, required by the flash algorithm itself.

### Conclusion

This paper has discussed and provides a direct way of estimating quantitatively how “extrapolated values” of thermophysical properties (supplied by specifically designed extrapolating techniques) influence the convergence characteristics of higher-level simulation algorithms.

The proposed general analysis has been implemented as a part of a general process simulator “EQUILIBRIA”<sup>2)</sup>. The numerical performance, tested on a variety of systems and applications, has shown that the computational requirements of the method are quite reasonable. It includes standard numerical procedures (matrices and vector-matrix multiplication; eigenvalues estimation, etc.), providing a high degree of reliability for the conclusions

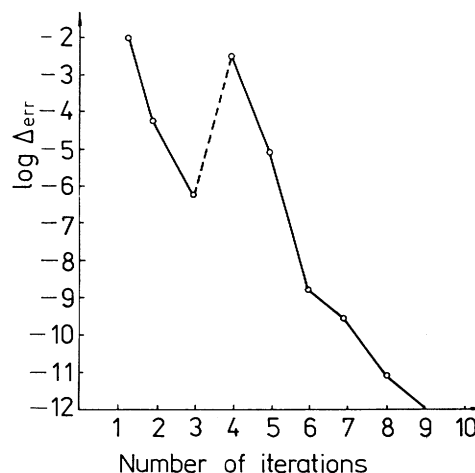
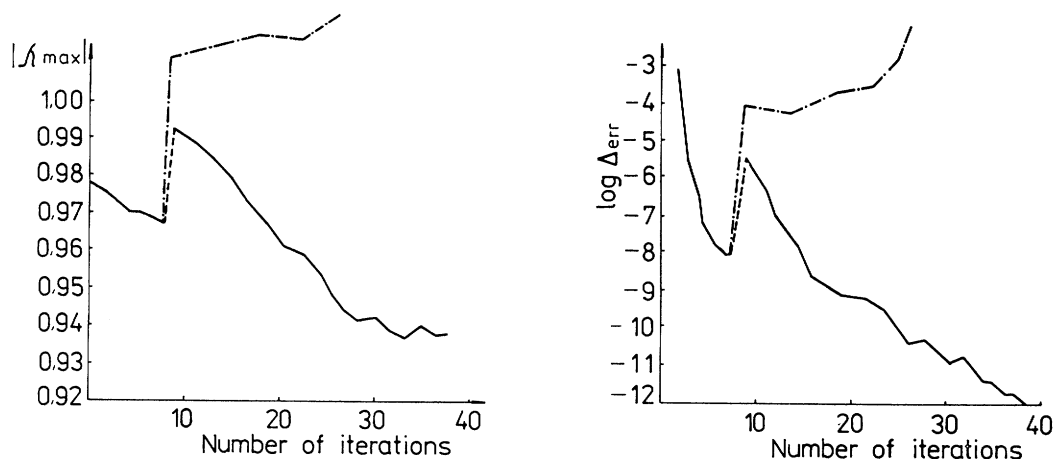
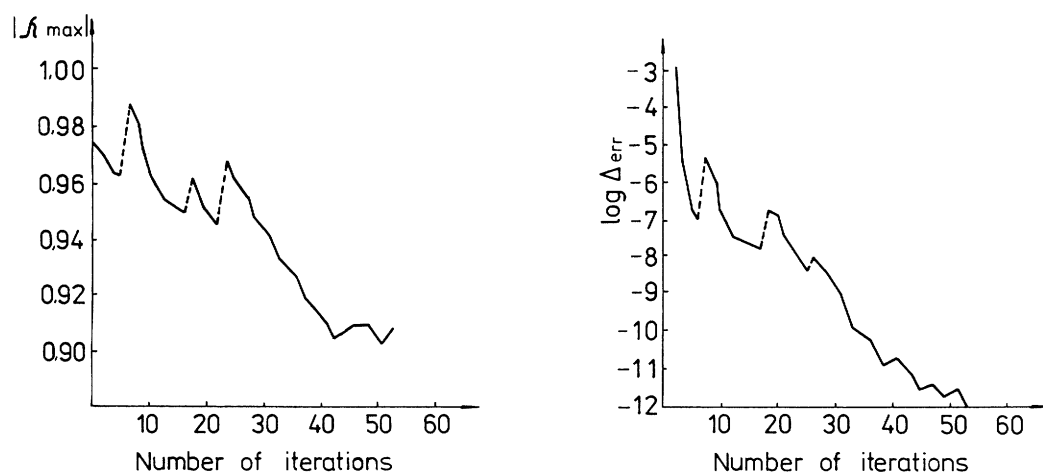


Fig. 4. Performance characteristics of the flash algorithm at  $T=230\text{ K}$  and  $P=70\text{ atm}$ : a) values of  $|\lambda_{\max}|$ , calculated at each iteration, b) profile of the convergence criterion  $\Delta_{\text{err}}$ ; —, an iteration without extrapolation; ---, performance of the extrapolation model<sup>1,5)</sup>



**Fig. 5.** Performance characteristics of the flash algorithm at  $T=204$  K and  $P=58$  atm: a) values of  $\lambda_{\max}$ , calculation at each iteration, b) profile of the convergence criterion  $\Delta_{\text{err}}$ ; —, an iteration without expolation; ---, performance of the extrapolation model<sup>15)</sup>; — · —, extrapolation leading to divergence of the calculations



**Fig. 6.** Performance characteristics of the flash algorithm at  $T=204$  K and  $P=57$  atm: a) values of  $|\lambda_{\max}|$ , calculated at each iteration, b) profile of the convergence criterion  $\Delta_{\text{err}}$ ; —, an iteration without extrapolation; ---, performance of the extrapolation model<sup>15)</sup>

obtained. The partial derivatives are given as analytical expressions and their estimation requires simple operations only.

#### Nomenclature

$b_m$	= van der Waal volume (Fig. 1)	$[\text{cm}^3 \text{mol}^{-1}]$
$f$	= fugacity	$[\text{atm}]$
$F$	= total number of moles in feed, $F = \sum z_i$	$[-]$
$k_{ij}$	= binary interaction parameter for the SRK CEOS	$[-]$
$K_i$	= equilibrium ratio of $i$ -th component	$[-]$
$l_i$	= mole number of liquid phase, component $i$	$[-]$
$L$	= total number of moles in liquid phase, $L = \sum l_i$	$[-]$
$M$	= Jacobian matrix	$[-]$
$N_c$	= total number of components	$[-]$
$P$	= specified pressure	$[\text{atm}]$
$R^n$	= $n$ -dimensional space	$[-]$
$T$	= temperature	$[\text{K}]$
$v$	= molar volume of liquid/vapor phase	$[\text{cm}^3 \text{mol}^{-1}]$
$v_i$	= mole number of vapor phase, component $i$	$[-]$

$V$	= total number of moles in vapor phase, $V = \sum v_i$	$[-]$
$v^*$	= "extrapolated" volume of liquid phase at infeasible pressure specifications, $P = P_{\text{spec}}$ (Fig. 1)	$[\text{cm}^3 \text{mol}^{-1}]$
$v_{pc}$	= pseudocritical volume at $T < T_{pc}$ of liquid phase (Fig. 1)	$[\text{cm}^3 \text{mol}^{-1}]$
$x$	= mole fraction of liquid phase	$[-]$
$y$	= mole fraction of vapor phase	$[-]$
$z_i$	= mole number of feed, component $i$	$[-]$
$\  * \ $	= norm of a vector	$[-]$
$\circ$	= composition $\Phi_2 \circ \Phi_1 : \Phi(x) = \Phi_2(\Phi_1(x))$	$[-]$
$\alpha$	= vapor mole fraction, $\alpha = V/F$	$[-]$
$\tau$	= activity coefficient	$[-]$
$\Delta$	= change in appropriate variable	$[-]$
$\bar{\Delta}$	= criterion, Eq. 6	$[-]$
$\delta^k$	= correction to $I^{(k)}$ at iteration $k$	$[-]$
$\delta_{ij}$	= Kroneker delta	$[-]$
$\varepsilon$	= recommended value, Eq. 11	$[-]$
$\lambda$	= eigenvalue	$[-]$
$\Phi$	= a mapping	$[-]$



$\phi$  = fugacity coefficient [—]

#### <Superscripts>

$k$  = number of iterations  
 $L$  = liquid  
 $V$  = vapor  
 $*$  = extrapolated

#### <Subscripts>

calc = calculated from the EOS  
convg = value of a parameter at convergence  
lim = limiting value  
max/min = maximal/minimal value  
pc = pseudocritical value  
real = true value (obtained without extrapolations)  
spec = specified

#### Literature Cited

- 1) Ammar, M. N. and H. Renon: *AIChE J.*, **33**, 926 (1987).
- 2) Bancheva, Iv. and R. P. Stateva: *Proc. Mediterranean Congress on Chem. Engng.*, **II**, 540 (1987).
- 3) Fletcher, R.: "Practical Methods for Optimization", 2nd ed., Wiley Interscience Publication, New York (1987).
- 4) Gundersen, T.: *Comp. Chem. Eng.*, **6**, 245 (1982).
- 5) Heidemann, R. A.: *Fluid Phase Equilibria*, **14**, 55 (1983).
- 6) Jovanovic, St. and R. Pounovich: *Ind. Eng. Chem. Process Des. Dev.*, **23**, 801 (1984).
- 7) Knapp, H., Doring, L. Oellrich, U. Plocker, J. M. Prausnitz: *DECHEMA Chemistry Data Series*, v. VI, DECHEMA, Frankfurt/M., West Germany (1982).
- 8) Mathias, P. M., J. F. Boston, J. F. and S. Watanasiri: *AIChE J.*, **30**, 182 (1984).
- 9) Michelsen, M.: *SEP 8109*, Instituttet for Kemiteknik (1981).
- 10) Michelsen, M.: *Fluid Phase Equilibria*, **9**, 1 (1982).
- 11) Michelsen, M.: *Fluid Phase Equilibria*, **9**, 21 (1982).
- 12) Mills, M. B., M. J. Wills and V. L. Bhirud: *AIChE J.*, **26**, 902 (1980).
- 13) Ohanomah, M. O. and D. W. Thompson: *Comp. Chem. Eng.*, **8**, 147 (1984).
- 14) Polling, B. E., E. A. Grens and J. M. Prausnitz: *Ind. Eng. Chem. Process Des. Dev.*, **20**, 127 (1981).
- 15) Stateva, R., St. Tsvetkov and Chr. Boyadjiev: *AIChE J.*, **36**, 132 (1990).

#### Appendix I

The following functional dependences are taken into consideration when deriving the elements of  $M\Phi_2$ :

$$\begin{aligned} l_i &= l_i(\alpha, K_i); \\ \alpha &= \alpha(K_1, K_2, \dots, K_{Nc}); \\ f &= f(\alpha, K_1, K_2, \dots, K_{Nc}); \\ K_i &= K_i(l, v); \\ v_i &= v_i(l_i). \end{aligned}$$

Then:

$$\frac{\partial l_i}{\partial K_j} = \left( \frac{\partial l_i}{\partial K_j} \right)_\alpha + \left( \frac{\partial l_i}{\partial \alpha} \right)_K \frac{\partial \alpha}{\partial K_j} = (1-\alpha) \left( \frac{\partial x_i}{\partial K_j} \right)_\alpha \delta_{ij} + \frac{\partial(1-\alpha)x_i}{\partial \alpha} \frac{\partial \alpha}{\partial K_j},$$

where:

$$\begin{aligned} \frac{\partial x_i}{\partial K_j} &= -\frac{z_i \alpha}{[1 + \alpha(K_i - 1)]^2} = -\alpha \frac{x_i y_i}{z_i K_i} \\ \frac{\partial(1-\alpha)x_i}{\partial \alpha} &= -\frac{z_i K_i}{[1 + \alpha(K_i - 1)]^2} = -\frac{x_i y_i}{z_i} \end{aligned}$$

$\partial \alpha / \partial K_j = -(\partial f / \partial K_j) / (\partial f / \partial \alpha)$  implicit differentiation from  $f(\alpha, K_1, \dots, K_{Nc}) = 0$ . The elements of  $M\Phi_2$  are given by:

$$m_{ij} = \alpha(1-\alpha) \left[ \left( -\frac{x_i y_i}{z_i K_j} \right) \delta_{ij} + \frac{x_i y_i / z_i (\partial f / \partial K_j)}{\alpha(1-\alpha) \partial f / \partial \alpha} \right] \quad (\text{I.1})$$

#### Appendix II

Appendix II demonstrates the differentiation of  $\partial f / \partial \alpha$  and  $\partial f / \partial K_j$  when  $f(\alpha, K)$  is either the *Rachford-Rice* or the *Barnes-Flores* formulation. These forms have proved to be most widely used.<sup>1,3)</sup>

A. The *Rachford-Rice* formulation:  $f(\alpha, K) = \sum (y_i - x_i)$

$$f(\alpha, K) = \sum \frac{z_i(K_i - 1)}{1 + \alpha(K_i - 1)} = 0 \quad (\text{II.1})$$

$$\begin{aligned} \alpha(1-\alpha) \frac{\partial f}{\partial \alpha} &= \alpha(1-\alpha) \frac{\partial \sum (y_i + x_i)}{\partial \alpha} = \alpha(1-\alpha) \left\{ -\sum \frac{z_i(K_i - 1)^2}{[1 + \alpha(K_i - 1)]^2} \right\} \\ &= -\sum \frac{z_i(K_i - 1)(1-\alpha)}{[1 + \alpha(K_i - 1)]} + \sum \frac{z_i(K_i - 1)(1-\alpha)}{[1 + \alpha(K_i - 1)]^2} \quad (\text{II.2}) \end{aligned}$$

Combining Eqs. (II.1) and (II.2), the following result is obtained:

$$\alpha(1-\alpha) \frac{\partial f}{\partial \alpha} = (1-\alpha) \sum \frac{z_i(K_i - 1)}{[1 + \alpha(K_i - 1)]^2} = s_1 \quad (\text{II.3})$$

$$\frac{\partial f}{\partial K_j} = \frac{\partial \sum (y_i - x_i)}{\partial K_j} = \frac{\partial}{\partial K_j} \left[ \sum \frac{z_i(K_i - 1)}{1 + \alpha(K_j - 1)} \right] = \frac{x_j y_j}{K_j z_j} \quad (\text{II.4})$$

Substituting Eqs. (II.3) and (II.4) into Eq. (I.1), the elements of  $M\Phi_2$ , when  $f(\alpha, K)$  is the *Rachford-Rice* form are obtained:

$$m_{ij} = \alpha(1-\alpha) \frac{x_i y_i}{z_i} \frac{1}{K_j} \left( -\delta_{ij} + \frac{x_j y_j / z_j}{s_1} \right) \quad (\text{II.5})$$

B. The *Barnes-Flores* formulation:  $f(\alpha, K) = \ln(\sum y_i) - \ln(\sum x_i)$

From  $f(\alpha, K) = 0$ , follows directly that  $\sum y_i = \sum x_i$  or the equivalent expression  $\sum (y_i - x_i) = 0$ . The latter is the *Rachford-Rice* formulation, Eq. (II.1).

Then:

$$\begin{aligned} \alpha(1-\alpha) \frac{\partial f}{\partial \alpha} &= \alpha(1-\alpha) \frac{\partial [\ln(\sum y_i) - \ln(\sum x_i)]}{\partial \alpha} \\ &= \frac{\alpha(1-\alpha) \frac{\partial (\sum y_i)}{\partial \alpha}}{\sum y_i} - \frac{\alpha(1-\alpha) \frac{\partial (\sum x_i)}{\partial \alpha}}{\sum x_i} \quad (\text{II.6}) \end{aligned}$$

However, since  $\sum y_i = \sum x_i$  the following relation holds:

$$\alpha(1-\alpha) \frac{\partial f}{\partial \alpha} = \frac{\alpha(1-\alpha) \frac{\partial (\sum (y_i - x_i))}{\partial \alpha}}{\sum y_i} \quad (\text{II.7})$$

The expression in the numerator on the right-hand side of Eq. (II.7) has been shown to be equal to  $s_1$  (see Eq. II.3). Hence:

$$\alpha(1-\alpha) \frac{\partial f}{\partial \alpha} = \frac{(1-\alpha) \sum \frac{z_i(K_i - 1)}{[1 + \alpha(K_i - 1)]^2}}{\sum y_i} = \frac{s_1}{\sum y_i} \quad (\text{II.8})$$

$$\begin{aligned} \frac{\partial f}{\partial K_j} &= \frac{\partial}{\partial K_j} [\ln(\sum y_i) - \ln(\sum x_i)] = \frac{\partial (\sum y_i)}{\sum y_i} - \frac{\partial (\sum x_i)}{\sum x_i} \\ &= \frac{\frac{\partial [\sum (y_i - x_i)]}{\partial K_j}}{\sum y_i} = \frac{x_j y_j}{\sum y_i} = \frac{x_j y_j}{\sum y_i} \quad (\text{II.9}) \end{aligned}$$

### Appendix III

$$\mathbf{S} = \Phi \mathbf{D} [11],$$

where:

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

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

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

Analysis of  $M\Phi = M\Phi_2 \cdot M\Phi_1$  shows that: i) The rows of  $M\Phi_1$  are proportional to the rows of the matrix  $\Phi$  (see Eq. 3), with a proportionality coefficient  $K_i$ ; ii) the columns of  $M\Phi_2$  are proportional to the columns of matrix  $\mathbf{D}$  with a proportionality coefficient  $1/K_j$ . Furthermore, the difference between the values of the elements of  $M\Phi_2$  are the elements of the matrix  $\mathbf{D}$  is given by the difference between  $s_1$  and  $s$ , where:

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

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

$$\begin{aligned} s &= 1 - \sum \frac{z_i K_i}{[1 + \alpha(K_i - 1)]^2} = \sum z_i - \sum \frac{z_i K_i}{[1 + \alpha(K_i - 1)]^2} \\ &= \sum \frac{z_i(K_i - 1)}{[1 + \alpha(K_i - 1)]^2} \{ [\alpha(\alpha(K_i - 1) + 1)] - \alpha + 2\alpha - 1 \} \\ &= \sum \frac{z_i(K_i - 1)\alpha}{[1 + \alpha(K_i - 1)]^2} - (1 - \alpha) \sum \frac{z_i(K_i - 1)}{[1 + \alpha(K_i - 1)]^2} \\ &= -(1 - \alpha) \sum \frac{z_i(K_i - 1)}{[1 + \alpha(K_i - 1)]^2} \end{aligned} \quad (\text{III.1})$$

Hence  $s = -s_1$ .

The same results is obtained when  $f(\alpha, \mathbf{K})$  is the Barnes-Flores form.

Taking into consideration the above derivation and performing the multiplication ( $M\Phi_1 \cdot M\Phi_2$ ) proves the equivalence of matrices  $M\Phi$  and  $\mathbf{S}$ .