# PHASE-SPLITTING PREDICTION IN ISOTHERMAL FLASH CALCULATIONS

By

Alejandro Antonio Romero

B.A.Sc. (Chemical Engineering) Universidad Nacional Autónoma de México

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

#### MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

CHEMICAL ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July 1989

© Alejandro Antonio Romero, 1989

10 -

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Chemical Engineering

The University of British Columbia Vancouver, Canada

Date July 31, 1989

#### Abstract

The development of processes that operate at conditions where multiple fluid-phase equilibria may occur demands fast and reliable simulation algorithms. The success of these algorithms depends on the correct prediction of the number and compositions of the phases present at a given temperature, pressure and overall fluid composition. The most commonly used routine for this purpose is the isothermal (single-stage) flash calculation.

A robust and efficient method to predict phase-splitting previous to performing an isothermal flash is implemented in this work. It is based on the thermodynamic stability analysis of the source phase using a Gibbs energy tangent-plane criterion. Depending on the outcome of the phase-split tests, the system may be declared stable as a single phase and thus no further calculations are needed, or unstable in which case a potential two- or three-phase solution can be obtained. Only in this last instance would the corresponding flash calculation follow, depending on the nature and quantity of the phases found: a vapour-liquid (VL) or a liquid-liquid (LL) flash if two phases are predicted, or a multiphase flash (VLL) when three phases are expected.

In addition to readily recognizing the number and type of phases present, the stability tests provide excellent initial composition estimates for the flash which assure fast convergence to the stable solution. Even though the algorithms developed can be used with any suitable model to calculate equilibrium properties, cubic equations of state are used throughout in this work as a single model to describe all fluid phases.

## **Table of Contents**

Al	bstra	ct	ii
Li	st of	Tables	vi
Li	st of	Figures	iii
$\mathbf{Li}$	st of	Symbols	xi
A	cknov	vledgement	v
1	Intr	oduction	1
	1.1	Purpose and scope	2
		1.1.1 The flash algorithms	3
		1.1.2 Thermodynamic models	3
	1.2	Structure of this study	4
2 Previous works on phase-splitting		vious works on phase-splitting	6
	2.1	Thermodynamic stability	6
	2.2	Phase-splitting	12
		2.2.1 Liquid-liquid split tests	12
		2.2.2 Gibbs energy minimization algorithms	18
		2.2.3 Mass balance approaches	30
3	Pro	posed algorithm	35
	3.1	General scheme	36

	3.2	Phase	search strategy
		3.2.1	Vapour phase search 49
		3.2.2	Liquid phases search
	3.3	Initiali	ization procedures
		3.3.1	Vapour phase
		3.3.2	Liquid phases
	3.4	Phase	removal tests
		3.4.1	Multiphase flash method
		3.4.2	Bubble- and dew-point equations method
		3.4.3	Coupled method
	<b>D</b>	14	A diamation 00
4	Res	ults an	id discussion 50
	4.1	System	ns tested
	4.2	Phase	search initialization
		4.2.1	Vapour phase
		4.2.2	Liquid phases
	4.3	Phase	search performance
		4.3.1	Vapour phase
		4.3.2	Liquid phases
		4.3.3	Wegstein's method 128
	4.4	Phase	removal tests
		4.4.1	Multiphase flash method
		4.4.2	Coupled method
		4.4.3	Bubble- and dew-point equations method
	4.5	Overa	ll phase-splitting prediction performance

5	Con	clusions and recommendations	152	
	5.1	General conclusions	152	
	5.2	Specific remarks and recommendations	153	
Bi	bliog	raphy	158	
Aŗ	open	dices	163	
A	Two	o- and three-phase flash algorithms	163	
	A.1	Isothermal VL or LL flash	163	
		A.1.1 Limit for the phase ratios	167	
		A.1.2 VL flash initialization	168	
		A.1.3 LL flash initialization	168	
	A.2	Isothermal VLL flash	170	
		A.2.1 Limit for the phase ratios	174	
		A.2.2 VLL flash initialization	175	
	A.3	Wegstein's method	175	
в	Act	ivity coefficient models in the phase-splitting algorithm	178	
	B.1	Activities and standard states	179	
	B.2	Vapour and liquid phases search	181	
	B.3	Liquid phase initialization	182	
Appendices 18				
С	C Description and listings of the computer programs			

-

v

## List of Tables

	Initial composition estimates for the vapour phase search using Raoult's	4.1
98	law and for the flash calculation using the incipient vapour found	
	Composition estimates for the incipient liquid phases obtained with the	4.2
	various initialization methods tried, composition of the incipient phases	
100	found and equilibrium values for two feed compositions in System 4	
	Comparison of the reliability and efficiency of the four methods tested to	4.3
103	provide initial compositions for the liquid phase search	
	% RMS deviation of the incipient phase estimates and of the equilibrium	4.4
	values from the actual incipient phases composition (mole fraction) in the	
105	prediction of LL immiscibility for System 1	
	Vapour and liquid phases search performance for System 2 from 90 to 320	4.5
	K indicating the stable solution, the number of iterations required for con-	
	vergence and the CPU time (in milliseconds) required for the initialization	
109	plus the phase search	
	Phase equilibria predictions comparison for System 5 when the phase-	4.6
	splitting algorithm is used with the optional additional phase search to find	
112	a missed vapour giving correct VL equilibria solutions.	
	Equilibria predictions when the phase-splitting algorithm is used with and	4.7
	without the optional additional phase search to find a missed liquid yielding	
120	correct LL equilibria solutions for System 4	

4.8	Performance of Wegstein's method in the vapour and liquid phase search	
	as a function of parameters w and $t_{max}$ for Systems 1, 2 and 3 compared	
	to the use of direct substitution.	130
4.9	$Performance\ of\ the\ phase\ removal\ tests\ and\ of\ the\ subsequent\ flash\ calcula-$	
	tion in the phase-splitting prediction for Systems 1, 2 and 3 near saturation	
	point conditions.	137
4.10	Comparison of the phase removal tests reliability with and without the	
	additional phase search versus the direct three-phase flash calculation for	
	Systems 1 and 6	140
4.11	Comparison of the prediction of the phase distribution for System 1 at	
	$different\ temperatures\ with\ the\ direct\ three-phase\ flash\ calculation\ and\ with$	
	the phase-splitting algorithm proposed followed by the corresponding type	
	of flash.	144
4.12	Predictions of the equilibrium conditions for System 2 at different tem-	
	peratures with the direct three-phase flash calculation and with the phase-	
	splitting algorithm proposed followed by the corresponding type of flash. $% \mathcal{L}^{(n)}_{\mathcal{L}}$ .	146
4.13	Number and type of phases predicted for System 3 at different pressures	
	with the direct three-phase flash calculation and with the phase-splitting	
	algorithm proposed followed by the corresponding flash. $\ldots$ $\ldots$ $\ldots$	148
4.14	Equilibria predictions for System 7 at three temperature and pressure con-	
	ditions with the direct three-phase flash calculation and with the phase-	
	splitting algorithm proposed followed by the corresponding flash. $\ldots$ .	150
4.15	Summary of the phase equilibrium predictions for the seven systems tested	
	obtained with the direct three-phase flash calculation and with the phase-	
	splitting algorithm proposed followed by the corresponding type of flash	
	when required	151

.

# List of Figures

2.1	Molar Gibbs energy of mixing at constant temperature and pressure for a	
	binary liquid system presenting a miscibility gap	9
2.2	Thermodynamic stability regions for a binary liquid mixture at constant	
	pressure showing the connodal and spinodal domes	10
2.3	Vapour-liquid thermodynamic stability regions in a Gibbs energy as a func-	
	tion of composition diagram for the methane- propane system at 278 K and	
	33.6 bar (modified from Radzyminski and Whiting)	11
2.4	Graphical representation of Michelsen's stability test for the case of a stable	
	binary system.	27
2.5	Michelsen's stability test and initial guess compositions for an unstable	
	binary system	28
3.6	Flow chart of the phase-splitting algorithm proposed in this work	40
3.7	Erroneous prediction of VL equilibrium due to bypass of a liquid incipient	
	phase when using Michelsen's method for the n-hexane-water system at	
	378 K and 4.08 atm (diagram modified from Heidemann).	45
3.8	Bypass of a vapour incipient phase when using the tangent plane criterion	
	leading to an incorrect LL equilibrium prediction in a hypothetical binary	
	mixture.	45
3.9	Additional phase search to find the n-hexane rich liquid incipient phase for	
	the n-hexane-water system (378 K and 4.08 atm) leading to the correct	
	prediction of LL equilibria	46

3.10	Detection of an incipient vapour phase by means of the additional search	
	proposed for the binary mixture of Figure 3.8 leading to the proper predic-	
	tion of a VL stable system.	47
3.11	Binary system stable as VL for which two incipient liquid phases are found	
	in the phase search, one being superfluous	80
4.12	Gibbs energy of mixing diagram for the n-hexane- water system at 378 K	
	and 5 atm (System 4) showing that both components are immiscible almost	
	throughout the entire composition range	101
4.13	Gibbs energy of mixing diagram for the hydrogen sulphide-methane system	
	at 190 K and 38 atm (System 5) for which the vapour phase is detected	
	with the additional phase search leading to the correct prediction of VL	
	equilibria	113
4.14	Gibbs energy of mixing as a function of composition for the methane-n-	
	butane-water system (System 6) at 311 K and 60 atm considered as a	
	vapour	123
4.15	Gibbs energy of mixing surface for System 6 considered as a liquid in the	
	entire composition range at 311 K and 60 atm.	123
4.16	Gibbs energy of mixing for the stable system (System 6 at 311 K and 60	
	atm) obtained by superimposing the vapour and liquid-like surfaces and	
	retaining the least Gibbs energy values at every composition	124
4.17	Gibbs energy of mixing as a function of composition for the methane-	
	carbon dioxide-hydrogen sulphide system (System 1) at 171 K and 20 atm	
	corresponding to the stable solution	126
4.18	Contour lines of the Gibbs energy of mixing surface for System 1 as a	
	stable system.	126

- 4.19 Type and number of phases present at equilibrium in the composition plane for System 1 at 171 K and 20 atm according to the location of the feed. 127

- 4.23 Convergence characteristics of Wegstein's method compared with direct substitution in the liquid phase search for System 1 at 180 K and 20 atm. 134

## List of Symbols

		• •
a	activ	nty/

- D determinant of  $H_{\Delta g}$
- F number of moles in the feed or stability criterion function
- f fugacity or function defined by equations (3.50) and (3.78)
- g molar Gibbs energy or dimensionless stability criterion function
- $g^*$  modified dimensionless stability criterion function
- $g_{i,j}''$  second partial derivative of  $\Delta g$  with respect to composition
- $\mathbf{H}_{\boldsymbol{\Delta}\mathbf{g}} \quad \text{Hessian matrix of } \Delta g$
- h parameter defined by equation (2.17)

K equilibrium constant

- k iteration number indicator
- $k_{i,j}$  binary interaction parameter
- L liquid or number of moles in the liquid
- m number of composition vectors for the liquid search
- n number of components in the mixture
- $n_i$  mole numbers of component i
- P pressure

P<sup>sat</sup> saturation pressure

- p phase initiator in liquid I
- $Q_1$  function defined by equation (2.27)
- $Q_2$  function defined by equation (2.28)
- q phase initiator in liquid II

- R universal gas constant
- r gradient function defined by equation (2.22)
- s variable defined by equation (3.43) or (3.70)
- T temperature
- t step size in Wegstein's method
- V vapour or number of moles in the vapour
- w Wegstein's method application frequency
- X mole numbers of the liquid incipient phase
- x liquid composition (mole fraction)
- Y mole numbers of the vapour incipient phase
- y vapour composition (mole fraction)
- z global or feed composition (mole fraction)
- **S** objective function

#### **Greek letters**

- $\alpha$  phase ratio V/F
- $\beta$  phase ratio  $L_i/F$  of function defined by equation (2.23)
- $\Gamma_L$  function defined by equation (3.116)
- $\Gamma_V$  function defined by equation (3.115)
- $\gamma$  activity coefficient
- $\Delta$  property of mixing
- $\varepsilon$  convergence tolerance
- $\eta$  control variable for the use of Wegstein's method
- $\mu$  chemical potential
- $\nu$  control variable in the phase removal tests
- $\pi$  number of phases

- $\phi$  fugacity coefficient
- $\psi_1$  phase ratio  $L_I/F$
- $\psi_2$  phase ratio  $L_{II}/F$
- $\omega$  Pitzer's acentric factor

superscripts

- *k* iteration number indicator
- <sup>L</sup> liquid
- v vapour
- I liquid phase I
- <sup>II</sup> liquid phase II
- $^{\infty}$  infinite dilution property
- ' incipient phase
- ° standard state
- property in the mixture

ideal state

subscripts

- B bubble-point
- c critical constant
- D dew-point
- init initial value
- i component index number
- $_{j}$  vector index number
- $_{p}$  index for the phase initiator in liquid I or indicator for a pseudo-parameter
- $_q$  index for the phase initiator in liquid II

- , index for the rest of the components  $(\neq p, q)$
- $_{3-\phi}$  three-phase
- $_I$  liquid phase I
- II liquid phase II

## Acknowledgement

Since it would be impossible to name them individually without making embarrassing omissions, I wish to express my gratitude to all whom directly or indirectly contributed to this work.

Specially I would like to thank my advisor, Dr. D. W. Thompson, for his guidance throughout my research.

The financial assistance provided by the University of British Columbia by means of a Graduate Fellowship is gratefully acknowledged.

#### Chapter 1

#### Introduction

The equipment design and simulation of processes that are conducted under conditions at which multicomponent mixtures may form two or three fluid phases require adequate prediction of phase equilibria using models that incorporate temperature, pressure and compositional effects. Failure to obtain correct estimates of phase equilibria can cause serious operation problems. For example, unexpected formation of two liquid phases on the trays of a distillation column can significantly reduce its capacity and efficiency. When the appearance of a second liquid phase is suspected, conditions should be determined to avoid it or the proper steps must be followed to take it into account [1].

Without doubt, the most commonly used routine for phase equilibria prediction is the isothermal isobaric flash, which according to Joulia et al. [2], is the most important routine in a process simulator. It is used not only to simulate an actual flash tank or other phase-contacting separation operations, but to define unknown conditions of inlet or effluent streams in process units as diverse as valves, heat exchangers, reactors, compressors, etc. When the behaviour of systems which may form three phases is described, the isothermal flash calculation must be able to determine whether a given mixture is a vapour (V), a liquid (L), a two-phase (VL or LL) or a three-phase (VLL) mixture for a wide range of temperature and pressure specifications and for a broad spectrum of components. Also, when two or three phases are present, the composition and amount of each phase should be obtained. Since flash calculations are performed hundreds or thousands of times in a moderately sized simulation, the algorithm must be fast as well as reliable.

#### 1.1 Purpose and scope

Two factors that determine the satisfactory outcome of isothermal flash calculations, other than the flash algorithm *per se*, are given primary importance and constitute the main objectives of this work:

- 1. The reliable prediction of the number and type of phases actually present before conducting the flash calculation, and
- 2. The generation of sound initial composition estimates when the flash is to be carried out.

When considering item 1., it has been recognized [3] that a large amount of computational time is wasted when a system which may have three phases is considered under conditions at which only one or two phases exist. For such cases it is desirable to detect the presence of only one or two phases as early as possible. Item 2. has been given considerable attention in the literature reviewed. It is well known that convergence either to the (most) stable solution (that with the least Gibbs energy at constant temperature and pressure) or to a spurious one (corresponding to a local minima) is very dependent on the initial composition estimates of the phases assumed to appear.

To achieve the desired objectives, prior to the isothermal flash a phase-splitting test based on a modified version of Michelsen's [4] thermodynamic stability analysis has been implemented. This analysis has the advantage that no initial estimates of the number of phases present at equilibrium are required and for unstable systems the composition of the new phases are provided and are used as starting values if the flash calculation is required. Thus, in the method to be proposed, if only one phase is found the flash is avoided. If two phases are predicted, the corresponding VL or LL two-phase flash is executed instead of the more time-consuming three-phase flash calculation.

#### 1.1.1 The flash algorithms

As mentioned earlier, the performance of the flash algorithms themselves is not the main concern in this work although they are used for comparative purposes and are incorporated in the computational scheme proposed. The flash routines used are those developed in [5] (with slight modifications) since they were considered to be dependable and were readily available. Two types of isothermal isobaric calculations are considered: a two- phase flash for the prediction of VL or LL equilibria, and a general three-phase flash for VLL equilibria computations. This last calculation can be reduced to either of the first kind if the existence of only two phases is detected. The flash algorithms are based on a mass-balance approach using an equation decoupling method which gives Rachford-Rice type check functions (for a general description of this computation method see Appendix A). A thorough discussion of the isothermal flash problem has recently been presented by Ammar and Renon [6].

#### 1.1.2 Thermodynamic models

The necessary and sufficient condition for thermodynamic equilibrium in a multi-phase multicomponent system at constant temperature and pressure, which requires the total Gibbs energy to have a minimum value, can be expressed by the condition of equality of fugacities for all the components in the phases in which they are present. Two general methods, which differ in the thermodynamic models used, are commonly utilized to obtain fugacities. Those that use an activity coefficient model for the liquid phase and a correlation or equation of state for the vapour (mixed model approach), and those where the same equation of state is used to represent both phases (single model approach). There are inherent advantages and disadvantages in each method, but they will not be addressed in this work in which only the single model approach is used.

Two well known cubic equations of state, which have gained popularity because of their simplicity and flexible application, are used as thermodynamic models. These are the Soave-Redlich-Kwong [7] and the Peng-Robinson [8] equations with the modification proposed by Mathias [9] to handle polar compounds and supercritical components. Also, to avoid the trivial solution in the density root calculation, the algorithm of Mathias et al. [10] is incorporated as implemented by Molina and Romero [5]. Although the use of these equations has been widespread in the gas and petroleum processing, it is not limited to these industries alone.

#### 1.2 Structure of this study

To give the reader a perspective of the approaches taken in the prediction of the stability of fluid mixtures, a literature review on the more relevant methods is given in Chapter 2. Then in the next chapter the proposed phase-splitting algorithm is presented. The global scheme is initially discussed followed by a more thorough description of each part of the algorithm. Because these parts are considered in detail in different sections, it may be helpful to occasionally make reference to the first section pertaining the global scheme. In Chapter 4 are given the results of the performance of each part of the algorithm according to their execution order. Also the overall results of the equilibria predictions are presented and compared against those obtained with the direct three-phase flash. The conclusions drawn from these calculations and the recommendations for the use of the proposed algorithm are given in the last chapter.

Appendix A gives a description of the flash algorithms (two- and three-phase) developed in reference [5], which are used in the final part of the phase-splitting algorithm. The direct three-phase flash is used also as a reference for the performance of the splitting method. Appendix B indicates the changes which would be needed in the proposed algorithm if a mixed model approach was to be employed to calculate equilibrium properties (using an activity coefficient model for the liquid phase and an equation of state for the vapour). Finally in Appendix C are presented the computer programs written (in Fortran 77) to perform the phase equilibria calculations. Complete listings and descriptions are given as well as the input and output files for an example calculation.

#### Chapter 2

#### Previous works on phase-splitting

The problem of determining whether or not a single-phase multicomponent mixture will split spontaneously and irreversibly into two or more distinct phases was first addressed by Gibbs in 1876 [11]. To date, a satisfactory solution to this problem has not been found except for the case of binary mixtures. The extension to ternary or higher systems is far from simple and despite several attempts made, there is still disagreement in the development of practical solutions <sup>1</sup>. The theoretical approach to this problem consists of doing a thermodynamic stability analysis on the system, which, as will be explained, is of limited significance to establish other more practical criteria.

#### 2.1 Thermodynamic stability

When considering the phase stability of homogeneous multicomponent systems at constant temperature, pressure and overall fixed composition, three equilibrium states can be defined: stable, unstable and metastable. A system is considered to be stable if after following a minor perturbation it reverts to its original state. Thermodynamically, the total Gibbs energy of a stable mixture is at a global minimum and thus can not be decreased relative to allowable changes that may take place within the system under the assumption of constant temperature and pressure. On the contrary, an unstable phase

<sup>&</sup>lt;sup>1</sup>Examples are the criticism of Van Dongen et al.'s work [11] by Michelsen in [12] and the corresponding response, ibid. p. 377, as well as the comments on Beegle et al.'s paper [13] by Heidemann in [14] and the subsequent reply, ibid., p. 826.

can not exist as such and will be permanently altered as a result of even an infinitesimal perturbation, decreasing its Gibbs energy by spontaneously changing to one or more stable phases.

Since the definition of the magnitude of a 'minor perturbation' is arbitrary, it can be ambiguous to define a metastable system which will not be altered by a small perturbation but will become unstable with respect to a larger perturbation. Thus, a phase change may or may not happen when a mixture is considered metastable. Classical thermodynamics textbooks consider metastable systems as stable, and only the limit between the unstable and the metastable region is discussed (e.g. in Modell and Reid [15], chapter 9). This limit is referred to as the limit of intrinsic stability, diffusional stability or material stability. Van Dongen et al. [11] follow this same criterion and show that the negative definiteness of the matrix of second partial derivatives of the molar Gibbs energy of mixing with respect to composition at constant temperature and pressure is the necessary and sufficient condition for instability. This square symmetric matrix, which is the Hessian of the molar Gibbs energy of mixing  $\Delta g$ , is:

$$\mathbf{H}_{\Delta g} = \begin{pmatrix} g_{1,1}'' & \dots & g_{1,n-1}'' \\ \vdots & \ddots & \vdots \\ g_{n-1,1}'' & \dots & g_{n-1,n-1}'' \end{pmatrix}$$
(2.1)

where n is the number of components in the mixture and  $g''_{i,j}$ , the second partial derivative of  $\Delta g$  with respect to composition (mole fraction z) at constant temperature and pressure, is

$$g_{i,j}^{\prime\prime} = \left(\frac{\partial^2 \Delta g}{\partial z_i \partial z_j}\right)_{T,P,z_{k\neq i,j}}$$
(2.2)

According to [11], the limit of intrinsic stability, at which  $H_{\Delta g}$  just ceases to be positive definite, occurs where the determinant D of the Hessian matrix itself becomes zero. This means that if D is negative, the mixture is unstable and will split in two or more phases, but if positive, in which case the system may be stable or metastable, the original phase is considered to remain as a homogeneous mixture. The locus of points where D = 0, called the spinodal curve, is relatively easy to calculate even for multicomponent systems as stated by Maurer and Prausnitz [16], however, the computational effort required is not negligible.

A major point overlooked when considering only the stability criterion to predict phase equilibria is that for practical engineering applications, metastable mixtures are unstable virtually in all situations, forming separate phases [3],[12]. Therefore, phase homogeneity can exist only in the stable region. Consequently, if the stability analysis indicates that the system is unstable (D < 0), a phase split will take place, but if D > 0 it is not possible to determine whether a mixture is stable and will remain as a single phase or metastable and hence form a new phase. In short, the result of doing a thermodynamic stability analysis on a homogeneous mixture (i.e. knowing the coordinates of the spinodal curve) is not always sufficient to predict phase-splitting. To overcome this problem, the limit between the metastable and the stable region, which defines the so-called connodal or equilibrium curve, must be known. Finding the coordinates of the connodal curve requires further calculations based on other phase-splitting tests, treated in section 2.2.

To illustrate the limitations of using phase stability criteria to predict phase equilibrium, let us consider the case of a binary mixture. Typically, a plot of the Gibbs energy of mixing as a function of composition is used to explain liquid-liquid equilibria when a miscibility gap exists. Such a diagram is depicted in Figure 2.1. The unstable region is bounded by the points C and D, which correspond to the inflection points on the



Figure 2.1: Molar Gibbs energy of mixing at constant temperature and pressure for a binary liquid system presenting a miscibility gap.

curve. At these points, D = 0 and in between them, where the curve is concave, D < 0. Regions BC and DE are metastable; the boundary between these and the stable regions is given by points B and E, defined by the common tangent represented by the dashed line. Clearly, a liquid in the region BE will form two distinct phases with equilibrium compositions  $x^{I}$  and  $x^{II}$  since by doing so will decrease its Gibbs energy to a minimum along the line BE. A thermodynamic stability analysis is only able to predict a miscibility gap in the region CD, but not in the metastable regions.

Figure 2.2 shows the connodal (sometimes called binodal) and spinodal curves for a hypothetical binary mixture at constant pressure. Note that the former curve is defined by the locus of points corresponding to the equilibrium compositions and the latter by the loci where D = 0. The metastable region is located between the connodal and the spinodal dome, and both meet at the critical point (corresponding in this case to an upper critical solution temperature).



Figure 2.2: Thermodynamic stability regions for a binary liquid mixture at constant pressure showing the connodal and spinodal domes.

For vapour-liquid equilibria in binary systems, the Gibbs energy of mixing curves are somewhat different from that represented in Figure 2.1. When cubic equations of state are used to obtain fluid properties, such as the Soave-Redlich-Kwong equation, Radzyminski and Whiting [17] stress that the Gibbs energy of a mixture is not a smooth, continuous function of composition. The actual curve does not exhibit a maximum, but rather it has two branches that meet at a cusp, one corresponding to the vapour and the other to the liquid density roots. They point out that misunderstanding of phase-splitting can lead to misuse of the algorithms for phase equilibrium calculation based on stability analyses. Although the problems associated with phase stability mentioned before are not specifically addressed, these can be inferred from the results presented by them. As an example, for the methane-propane system at 278 K and 33.6 bar their stability



Figure 2.3: Vapour-liquid thermodynamic stability regions in a Gibbs energy as a function of composition diagram for the methane- propane system at 278 K and 33.6 bar (modified from Radzyminski and Whiting).

analysis indicates that the unstable region, where a two-phase VL mixture is predicted, occurs in the composition range from about 0.466 to 0.504 methane mole fraction. At that temperature and pressure the bubble- and dew- point compositions are 0.205 and 0.780 respectively, so the actual VL region is much broader than predicted by the stability analysis as shown in Figure 2.3. The unstable region is reduced even more at higher temperatures, e.g. for the calculation at 311 K (and 33.6 bar) this region comprises the range 0.290 to 0.295, whereas the actual two-phase region extends from 0.131 to 0.518 methane mole fraction corresponding to the bubble- and dew-point compositions.

The fact that the spinodal dome is always contained within the equilibrium dome adds another feature that makes thermodynamic stability tests unattractive from a practical point of view. This feature is that the coordinates of the limit of intrinsic stability usually constitute poor initial estimates for the actual equilibrium compositions. The reason for this is that the initial composition estimates should preferably be outside the immiscibility region rather than inside it in order to avoid convergence of the equilibrium calculations to the trivial solution, in which the number of phases predicted is less than the number corresponding to the stable solution.

To eliminate ambiguity in some of the terms used from this point on, the region where D < 0 (section CD on the curve in Figure 2.1) will be referred to as 'intrinsically unstable' or 'thermodynamically unstable', whereas this and the metastable regions as a whole (portion BE of the curve) will be generically called 'unstable'. Also, when referring to the Hessian matrix criterion to determine phase stability, the term 'thermodynamic stability' will be employed as opposed to simply 'stability', which will be used in a broader sense to describe methods to find conditions in the unstable region.

#### 2.2 Phase-splitting

Alternative methods to thermodynamic stability analysis, whose goals are to predict the number and type of phases at equilibrium and/or to obtain accurate initial composition estimates for equilibrium calculations, are referred to as phase-splitting methods in this study. Some of them resort to stability tests of different kinds; the various approaches reviewed are discussed in this section.

#### 2.2.1 Liquid-liquid split tests

#### Maurer and Prausnitz's method

An approximate procedure to obtain the LL connodal curve in binary, ternary and higher systems is proposed by Maurer and Prausnitz in [16]. Their concern is not to calculate the equilibrium compositions but to determine whether or not at a given temperature, pressure and overall liquid composition immiscibility will occur. For multicomponent systems their method, which is developed for ternary mixtures and then extended to four or more components, starts by doing a thermodynamic stability analysis of the system. If intrinsic instability is detected, two liquid phases must be present and then it is suggested that the equilibrium compositions be obtained by a flash calculation. On the contrary, if the determinant D is positive, any binary pairs for which a miscibility gap exists at some composition in the range 0 < x < 1 are determined as well as the corresponding spinodal and connodal composition coordinates.

For the case where only one binary pair has a miscibility gap, a trial point t between the original feed composition z and the mid-point of the spinodal coordinates of the immiscible pair is tested for thermodynamic stability. If such a point is in the intrinsically unstable region, the original system is considered to be metastable, separating into two phases; otherwise it is assumed stable as a single phase. When two or more binaries are partially miscible the calculation is extended to all these binary systems. The usefulness of this method clearly depends on the magnitude of the distance between points z and t. Various criteria, based on geometrical considerations, are given by these authors to select this length parameter.

Although the approximate calculations presented are claimed to require about one order of magnitude less computing time than "the exact calculations" (presumably an equilibrium flash), the method has some inherent features that restrict its use. It is intended only for LL equilibria prediction where at least one of the binary pairs has a miscibility gap, the immiscible region tends to be overestimated, and is based on geometrical relationships rather than on thermodynamic grounds. Also, since each possible binary pair in the mixture must be tested for stability in the whole composition range, the extension to systems with a large number of components would significantly reduce its computational efficiency<sup>2</sup>.

#### Shah's method

The method presented by Shah [1] is intended to determine whether a second liquid phase may exist in any tray of a multicomponent distillation column, but the heuristic algorithm developed to test for the stability of the existing liquid is flexible enough to be used in other equilibrium calculations. For example, Fournier and Boston [18] use it in their VLL flash algorithm which is based on the same strategy as Shah's tray test: beginning with the assumption that only one liquid is present, the VL equilibrium and mass balance equations are solved. Then the liquid resulting from the two-phase calculation is examined for stability. If it is stable, there is no need to search for a second liquid; however, if found unstable, the two-phase solution is not valid and calculations are carried out to find a three-phase solution. This strategy as a whole defeats the purpose of the present study since two or at least one flash calculation is required to determine the configuration of the system at equilibrium. Nevertheless, the heuristic stability test, which accounts for phase separation in the metastable region, is of interest. It is based on the following features: a) the stability of the VL system is determined by examining only the liquid phase; b) instability is indicated by splitting the liquid phase into a pair of liquids, not necessarily in equilibrium but subject to the mass balance constraints, having a lower Gibbs energy than the original liquid; c) a splitting algorithm generates an initial guess of the liquid split which allows a region of lower Gibbs energy to be found quickly and prevents convergence to a trivial solution when the original liquid is unstable.

The assumption taken in a) is somewhat dubious since it can be conceived that the vapour phase found from the VL calculation can itself be unstable and split into different

<sup>&</sup>lt;sup>2</sup>The number of binary pairs that can be formed out of an *n* component system is  $n!/[(n-2)! \cdot 2!]$  or n(n-1)/2. E.g. for a 3 component system there are 3 possible binary pairs; for 10 components the number of binaries is 45.

vapour and liquid phases. His argument stating that "The validity of determining the stability of the whole system just by testing for the stability of the liquid phase results from the fact that the vapour and the liquid phases are in equilibrium with each other before testing for the presence of the second liquid phase" seems inconclusive. Looked at from a Gibbs energy analysis perspective, his justification can be questioned as follows: if the VL solution corresponds to a local minimum in Gibbs energy and a global minimum can be reached by obtaining an alternative V'L'L'' solution, the vapour V' may or may not correspond to the same V vapour. If V' is in the same amount and is of the same composition as the original V, it can be concluded that V was stable and that the unstable L split into liquids L' and L". However, if V' is of a different nature than V and L is similar to either L' or L", it is clear that V split into V' and L" or into V' and L', respectively, and the assertion that V was indeed unstable can be made. An example in which this last situation prevails is given by Fournier and Boston [18] when flashing at 377.59 K and 40.83 atm a mixture consisting of propylene, diisopropyl ether, isopropanol and water. The vapour fraction obtained for a VL flash is 0.8725 whereas for the VLL flash is 0.2414. It is mentioned that the second liquid phase was formed primarily from the vapour phase of the two-phase flash solution and that the original liquid phase of the VL flash was "nearly stable". Even though the three-phase solution was correctly predicted using Shah's stability test in this example, the first assumption on which the stability test is based may make it prone to failure.

For this reason and since it requires a previous VL flash calculation, this stability test algorithm is not adequate for the purpose of this work. Of more importance is the initial split algorithm proposed by Shah (which is the second step of the stability test) and thus will be considered next. This algorithm generates composition estimates for two new liquid phases if they were to form from an unstable liquid phase. It is based on selecting the component most likely to initiate a new separate phase from the original liquid (source phase). Once this new phase (designated phase I) is initiated, the component with the lowest solubility in it is selected as the initiator of the second new phase (phase II). Then the remaining components are split between the two new phases as dictated by the mass balance and equilibrium equations.

Initially, the component with the largest pseudo infinite-dilution activity in the original mixture is selected as the initiator of phase I and is indicated by subscript p. The pseudo infinite- dilution activity for component i is obtained as the product of the global composition and the activity coefficient at infinite dilution for that component, i.e.  $x_i \gamma_i^{\infty}$ . Shah asserts that the component with the maximum pseudo infinite- dilution activity in a given solution will have the maximum tendency to initiate a separate liquid phase since the mole fraction of a component in a solution is a measure of its availability and the inverse of the infinite-dilution activity coefficient is an indirect measure of its solubility in the rest of the solution. Phase II is initiated by selecting the component which has the largest pseudo infinite-dilution activity in the pure p component and is designated with subscript q. Then components p and q are split between phases I and II assuming no other components are present. Letting  $n_p^I$ ,  $n_p^I$ ,  $n_q^I$  and  $n_q^{II}$  denote the mole numbers of p and q in phases I and II, the phase equilibrium and mass balance requirements dictate that:

$$\gamma_p^{II^{\infty}} n_p^{II} / (n_p^{II} + n_q^{II}) = n_p^I / (n_p^I + n_q^I)$$
(2.3)

$$\gamma_q^{I^{\infty}} n_q^{I} / (n_p^{I} + n_q^{I}) = n_q^{II} / (n_p^{II} + n_q^{II})$$
(2.4)

where it has been considered that the activity coefficients of p and q in phases II and I, respectively, are constant and each equal to its corresponding infinite dilution value, and in phases I and II are equal to the pure component values (i.e. equal to 1 considering symmetric normalization of the activity coefficients). Also, considering 1 mole of the original mixture,

$$n_p^I + n_p^{II} = x_p \tag{2.5}$$

and

$$n_q^I + n_q^{II} = x_q \tag{2.6}$$

Equations (2.3) through (2.6) can be arranged to give a system of two simultaneous non-linear equations which may be solved for the moles of p and q in both phases. Once these variables are determined, the split for the remaining components between phases I and II is determined again assuming that only p and q are present. Thus, for the rest of the components (designated by subscript r) taken one at a time, the equilibrium and mass balance equations yield:

$$\gamma_r^{I^{\infty}} n_r^I / (n_p^I + n_q^I + n_r^I) = \gamma_r^{II^{\infty}} n_r^{II} / (n_p^{II} + n_q^{II} + n_r^{II})$$
(2.7)

and given the condition that for 1 mole of total mixture

$$n_r^I + n_r^{II} = x_r \tag{2.8}$$

equation (2.7) is a simple non-linear equation in one unknown (either  $n_r^I$  or  $n_r^{II}$ ) which needs to be solved for the remaining n-2 components. Once all the  $n^I$  and  $n^{II}$  vectors are obtained, normalization gives the two initial estimates  $x^{I}$  and  $x^{II}$  assuming phasesplitting of the original liquid phase of composition x.

### 2.2.2 Gibbs energy minimization algorithms

### Gautam and Seider's method

In the work presented by Gautam and Seider [19], an algorithm that permits the split of either the vapour or the liquid phases is introduced. This algorithm is coupled with a particular Gibbs energy minimization method (the Rand method, a Newton-based algorithm), but any other minimization method can be used. Their strategy allows for the assumption of one or more phases and the addition of vapour or liquid phases as necessary during the equilibrium calculations so that the system's Gibbs energy is minimized. Given guesses for phase distribution and compositions, the calculation is started using the minimization algorithm. Along the search trajectory an additional phase is postulated by splitting the source phase into two or by splitting each phase if two are assumed. In this last case, the source phase is selected as the phase which leads to the lowest Gibbs energy after being split into two. If a decrease in Gibbs energy occurs by splitting the source phase, the trial phases replace the source phase and the minimization algorithm is continued. However, since the two trial phases may coalesce after several iterations if the new phase is present in trace quantity (in which case the decrease in Gibbs energy is not significant), the phase present in trace amount is dropped and the split test is repeated for the trial phase having the next lowest Gibbs energy.

If an increase in Gibbs energy is observed when the first phase split is attempted, a few iterations of the minimization algorithm are carried out to permit a decrease. If a decrease is still not achieved, the calculation is repeated for the next trial phase with the lowest Gibbs energy. If after examining all combinations of source and trial phases the phase-split is unsuccessful, the source phase(s) is(are) considered stable. To decide when to attempt the addition of a phase during the search trajectory, a criterion based on a measure of the degree of convergence is given to compute the number of iterations between splitting checks.

Gautam and Seider assert that their algorithm works well in the metastable region and that the global minimum in Gibbs energy was obtained in every case tested. They also point out a potential pitfall which could be avoided, but the efficiency of the method would be considerably reduced. The liability arises from the decision to accept the first trial phases that lead to a Gibbs energy reduction in order to save computer time, rather than examine all combinations of source and trial phases. This gives rise to the possibility that the same two trial phases are accepted repeatedly with one of them rejected after a certain number of iterations and/or that other trial phases which might lead to a more stable solution (with the least Gibbs energy) are prevented from being considered. For example, starting with a liquid, in the phase-splitting test  $L \rightarrow VL \rightarrow VL'L'' \rightarrow \cdots$ the sequence delimited by the bracket could be repeated preventing the examination of the trial phases L'L''.

Other restrictions to the algorithm are that it is designed to work coupled with a minimization algorithm to obtain equilibrium compositions, but not as a stand-alone method to predict phase- splitting prior to the actual equilibrium calculation. Also, the approach is based on the assumption that if a solution with  $\pi$  phases exists, it has a lower Gibbs energy than a  $\pi - 1$  phase solution which Baker et al. [21] have claimed to be a heuristic rule with possible exceptions. Finally, but not of least importance, is the restriction that the systems tested must contain at least one partially miscible pair for the algorithm to apply.

Despite these drawbacks the phase-splitting part of the algorithm, used to generate initial composition estimates for the postulated new phases, has been frequently alluded to in many phase-splitting methods. For instance, in a recent paper Walraven and Van Rompay [20] use a modified Gautam and Seider's splitting algorithm to determine the stability of the liquid phase and if unstable propose a short-cut liquid-liquid calculation to estimate the equilibrium compositions. Since it also serves as a basis for one of the initialization schemes proposed in Chapter 3, the phase-splitting method of Gautam and Seider will be reviewed here. According to these authors the magnitude of the activity of a component in a mixture is a measure of its tendency to form two phases, so the first step of their splitting method is to locate the component with the highest activity in the original mixture (identified in this study as component p). Then a second component (labeled here q) is selected as the one with the highest binary activity with p, considering a composition proportional to that of the source phase. When forming a new liquid phase, if a component is present in large amounts in the source phase and this phase is a liquid, such a component is by passed in the selection of p and q. This is done to assure that the liquid trial phases are not similar in composition to any already existing liquid. With components p and q identified, if the two trial phases are liquid, their compositions  $x^{I}$  and  $x^{II}$  are determined by solving the equations

$$\gamma_p^I x_p^I = \gamma_p^{II} x_p^{II} \tag{2.9}$$

$$\gamma_q^I x_q^I = \gamma_q^{II} x_q^{II} \tag{2.10}$$

where  $x_p^I = n_p^I/(n_p^I + n_q^I)$ ,  $x_q^I = n_q^I/(n_p^I + n_q^I)$  and similarly for  $x_p^{II}$  and  $x_q^{II}$ ,  $\gamma$  being the activity coefficient and  $n_p$ ,  $n_q$  the mole numbers. Equations (2.9) and (2.10) constitute a system of two simultaneous non-linear equations considering that the total number of moles of p and q in the mixture are known:  $n_p = n_p^I + n_p^{II}$  and  $n_q = n_q^I + n_q^{II}$ . A
Newton-Raphson procedure is suggested by Gautam and Seider to solve these equations with the physical properties assumed constant over an iteration.

If the trial phases are a vapour and a liquid, the corresponding equations to obtain compositions y and x are

$$y_p \phi_p P = x_p \gamma_p f_p^o \tag{2.11}$$

$$y_q \phi_q P = x_q \gamma_q f_q^o \tag{2.12}$$

which are solved making analogous considerations as those described above for equations (2.9) and (2.10). In this case,  $\phi$  is the fugacity coefficient, P the pressure and  $f^o$  the standard state fugacity.

The remaining n-2 chemicals in the mixture are ordered according to their decreasing binary activity with component p. Each of these components, identified by subscript r, is distributed in turn according to

$$\gamma_r^I x_r^I = \gamma_r^{II} x_r^{II} \tag{2.13}$$

for two liquid trial phases, where  $x_r^I = n_r^I / (n_p^I + n_q^I + n_r^I)$  and similarly for  $x_r^{II}$ , or according to

$$y_r \phi_r P = x_r \gamma_r f_r^o \tag{2.14}$$

for a vapour and liquid trial phases with  $y_r = n_r^V/(n_p^V + n_q^V + n_r^V)$  and  $x_r$  obtained accordingly. Since the total number of moles of each r component is known  $(n_r = n_r^I + n_r^{II})$ 

or  $n_r = n_r^V + n_r^L$ ) and those of p and q in each trial phase can be solved for, equation (2.13) or (2.14) has a single unknown. Again in this case, when forming a new liquid phase from a liquid source phase, any chemical present in large amounts in the original liquid is distributed after the remaining components. Because each component r is distributed leaving the amounts of the previous components fixed, these authors [19] stress that the compositions given by either of equations (2.13) or (2.14) do not correspond to equilibrium states. This means that initially the two trial phases may not have a lower Gibbs energy than the source phase, but this is accounted for in the minimization algorithm.

# Soares et al.'s method

Among the many flash algorithms based on Gibbs energy minimization, the work done by Soares et al. [22] pays special attention to the selection of initial guess compositions and flow rates for partially miscible systems. This initialization scheme, rather than the minimization algorithm, is of particular interest because the total number of phases expected in the solution need not be specified in advance and because of the distinctive approach to initialize the vapour phase. The method begins by estimating the flow rate and composition of the vapour phase based on calculating the two-phase bubble and dew temperatures  $(T_B \text{ and } T_D)$  of the feed at the specified pressure. Knowing that for  $T = T_B$ the vapour flow rate is equal to zero and that for  $T = T_D$  it is equal to the feed flow rate, a linear interpolation is carried out to estimate the vapour flow rate at the specified temperature. Also, since for  $T = T_B$  the vapour composition is the one obtained from the bubble-point calculation and for  $T = T_D$  it is that of the feed, the composition of the vapour can also be readily obtained by interpolation. Although not specified by these authors, it is clear that by doing these saturation calculations the condition where the mixture is all vapour  $(T > T_D)$  or where no vapour is present  $(T < T_B)$  can be detected at this point.

With the estimates of the vapour flow rate and composition, a mass balance leads to the overall liquid flow rate and composition guesses. This liquid may then be split into two new liquid phases using a LL equilibrium calculation. In order to avoid a trivial solution in this calculation, the initial composition guesses overestimate the separation to be obtained. This is done by assuming a number of liquid phases equal to the number of components, considering each phase as being almost a pure component with small specified amounts of the other constituents. The number of liquid phases is eventually reduced as phases with equal compositions or zero flow rates are eliminated.

The above method is claimed to allow the determination of initial composition estimates for the different phases which are close to the solution; nevertheless it has some disadvantages. Since three rigorous phase equilibrium calculations are required (two saturation point calculations and a LL equilibrium computation) the method lacks the desired efficiency. In addition, the saturation point calculations themselves require composition estimates for the incipient phases as well as initial guesses for the saturation temperatures. Methods to provide these estimates are not included as part of the algorithm discussed.

#### Michelsen's method

Stability tests which do not require estimates of the number of phases at equilibrium and that provide compositions of the new phases for unstable systems are presented by Michelsen in [4] as a preliminary step in isothermal flash calculations. These tests are based on the tangent plane criterion of Gibbs as put in theorem form by Baker et al. [21], who demonstrate that the necessary and sufficient condition for a system to be stable at constant temperature and pressure is that the tangent plane to the Gibbs energy surface at the global composition z should at no other point intersect (lie above) the Gibbs energy surface <sup>3</sup>. The resulting corollary would express that at a given composition a system is unstable if the tangent plane to the Gibbs energy surface at that point intersects (lies above) the Gibbs energy surface at some other point in the overall composition range.

Baker et al. [21] indicate that mathematically the solution of the phase equilibria problem can be obtained by finding a plane tangent to the Gibbs energy surface at two or more points which leads to the least value in Gibbs energy. Such points of tangency correspond to the compositions of the predicted equilibrium phases, being required by the material balance restrictions that the global composition of the system lie within the region bounded by these points. Since the slope of the tangent plane corresponds to the components' chemical potentials, this tangent plane criterion is equivalent to that requiring equality of chemical potentials, preservation of the material balance and a state of lowest possible Gibbs energy as the conditions for equilibrium at the specified temperature and pressure. With this in mind, Michelsen's stability test can be more easily understood. This test has its foundation on the fact that if a decrease in Gibbs energy can not be achieved when a mixture is divided into two phases (formed by removing an infinitesimal amount from the original mixture), then the mixture is stable. This criterion for stability can be expressed as

$$F(x) = \sum_{i=1}^{n} x_i [\mu_i(x) - \mu_i(z)] \ge 0$$
(2.15)

for all trial compositions x, where  $\mu_i(z)$  and  $\mu_i(x)$  are the chemical potentials of component i at the feed composition and at any other composition and n is the total number of components in the mixture.

<sup>&</sup>lt;sup>3</sup>For binary systems the Gibbs energy surface is a curve and the tangent plane is a straight line; for multicomponent systems the Gibbs energy surface is a hypersurface and the tangent plane corresponds to a hyperplane. Note also that the Gibbs energy of mixing surface can be equivalently used for this analysis.

Michelsen asserts that F(x) will be non-negative if it is non-negative at all stationary points, that is, points where the derivatives of F(x) with respect to all independent variables equal zero. In terms of fugacity coefficients  $\phi_i$ , the stability criterion can be written in the form

$$g(x) = F(x)/RT = \sum_{i=1}^{n} x_i [\ln x_i + \ln \phi_i(x) - h_i] \ge 0$$
 (2.16)

where

$$h_i = \ln z_i + \ln \phi_i(z) \tag{2.17}$$

for i = 1, 2, ..., n. The corresponding stationarity criterion is

$$\ln x_i + \ln \phi_i(x) - h_i = k \tag{2.18}$$

Introducing new variables  $X_i = \exp(-k)x_i$ , this last equation becomes

$$\ln X_i + \ln \phi_i(x) - h_i = 0 \tag{2.19}$$

The independent variables  $X_i$  can be interpreted as mole numbers with corresponding mole fractions  $x_i = X_i / \sum_{i=1}^n X_i$ . Michelsen shows that stationary points are found by solving equation (2.19) and that stability is verified provided that at all stationary points  $g(X) \ge 0$ , corresponding to  $\sum_{i=1}^n X_i \le 1$ . Conversely, a phase is unstable if stationary points where g(X) < 0 or  $\sum_{i=1}^n X_i > 1$  can be located.

#### Chapter 2. Previous works on phase-splitting

An equivalent stability criterion to that of equation (2.16), based on variables  $X_i$ , is formulated as

$$g^*(X) = 1 + \sum_{i=1}^n X_i [\ln X_i + \ln \phi_i(x) - h_i - 1] \ge 0$$
(2.20)

where no constraints on  $X_i$  except that  $X_i > 0$  are required. Michelsen also shows that the stationary points of  $g^*(X)$  correspond to those of g(X) and are given by solving equation (2.19). Moreover, since  $g^*(X)$  is negative at all points where g(X) is negative, a negative  $g^*(X)$  indicates that the system is unstable.

Thus, solving for equation (2.19) and obtaining  $\sum_{i=1}^{n} X_i$  are all the necessary calculations required to determine the stability of the system of composition z at constant temperature and pressure. The geometrical interpretation of the stability test corresponds to finding the maximum (in absolute value) vertical distances F from the tangent plane (at the feed composition z) to the Gibbs energy surface at any other composition. If any of these distances are found to be negative then the system is unstable, but if all F's are non-negative the system is stable. Therefore, new phases will be found whenever negative minima in F are located and the corresponding points will also provide initial composition estimates for these new phases. These concepts are shown in Figures 2.4 and 2.5 for a hypothetical binary mixture, where the tangent line to the Gibbs energy is represented by the dotted line. For the case represented in Figure 2.4 the system is stable at composition z since only non-negative values of F can be obtained. The tangent line intersects the Gibbs energy of mixing curve only at the feed composition and never lies above it. In Figure 2.5 two minima in F are found; the tangent line lies above the Gibbs energy of mixing curve and intersects it at points other than z. The composition of the two incipient phases corresponding to the minima in F are  $x'^{I}$  and  $x'^{II}$  whereas



Figure 2.4: Graphical representation of Michelsen's stability test for the case of a stable binary system.

the equilibrium values, given by the equilibrium tangent line (dashed line), are  $x^{I}$  and  $x^{II}$ . (Note that in this case the composition of the incipient phases are very close to the actual equilibrium compositions).

The methods proposed in [4] for solving the stationarity condition (equation (2.19)) are direct substitution or accelerated direct substitution, where subsequent iterates are determined from

$$\ln X_i^{(k+1)} = h_i - \ln \phi_i^{(k)}(x) \tag{2.21}$$

It is stated that direct substitution has linear convergence and that if the composition dependence of the fugacity coefficients is weak, convergence will be rapid. Acceleration methods such as the General Dominant Eigenvalue Method (Crowe and Nishio [23]) or



Figure 2.5: Michelsen's stability test and initial guess compositions for an unstable binary system.

Broyden's method (Dennis and Moré [24]) are recommended but not tested. A minimization method (Murray [25]) applied to the stability function (equation (2.20)) is also suggested. When direct substitution is used, iterations can be terminated early in cases where the trivial solution X = z is approached. After each iteration a convergence variable r is calculated as

$$r = 2g^*(X)/\beta \tag{2.22}$$

where

$$\beta = \sum_{i=1}^{n} (X_i - z_i) \frac{\partial g^*(X)}{\partial X_i} = \sum_{i=1}^{n} (X_i - z_i) [\ln X_i + \ln \phi_i(x) - h_i]$$
(2.23)

The value of r will approach 1 as X approaches the trivial solution, so the search is abandoned when |r - 1| < 0.2 and  $g^*(X) < 10^{-3}$ . Michelsen recognizes that except for

the case where the trivial solution X = z is the only minimum in the modified distance function  $g^*(X)$ , the minimization problem has multiple solutions and thus multiple initial estimates are necessary to assure that negative minima are detected. For VL equilibria problems he claims that the initial estimate is not crucial and essentially it is only required that the two initial composition vectors be located on opposite sides of the phase composition z. For hydrocarbon systems the following two sets of initial estimates are recommended:

$$X_i = z_i K_i \tag{2.24}$$

$$X_i = z_i / K_i \tag{2.25}$$

where the equilibrium K factors are approximated by

$$K_{i} = \frac{P_{c_{i}}}{P} \exp[5.42(1 - \frac{T_{c_{i}}}{T})]$$
(2.26)

in which subscript c indicates the corresponding critical constant for component i. When the stability of multiphase systems is investigated (or for the case of LL equilibria), the search for an additional phase is required and a different initiation procedure is suggested. The number of initial estimates chosen is equal to the number of components and the compositions of the phases are selected as the pure components. It is recommended that instead of converging each initial estimate in turn, calculations be done in parallel for all estimates. For each trial phase the value of r is evaluated and iterations are discontinued when r is increasing or when the trivial solution is approached. If no negative values of  $g^*(X)$  are encountered after four iterations, only the X vector with the smallest value of r is converged; if this leads to the trivial solution it is assumed that the system is stable.

In spite of having strong thermodynamic foundations and being claimed very reliable at a reasonable computational cost, the method proposed by Michelsen has some weaknesses. For example, it is shown that direct substitution converges to a local minimum of  $g^*(X)$  and also the initialization procedure does not guarantee the detection of all negative minimum. Also, initializing with the pure components when searching for a multiphase solution may lead to conditions where a pure liquid does not exist in which case additional procedures must be followed (such as evaluation of fugacity coefficients at lower temperatures or higher pressures to ensure liquid-like properties). Nonetheless, these drawbacks mainly concern the initialization step and can be partially if not totally eliminated by undertaking more rigorous starting procedures, in which case the method has advantages that offset these limitations. For this reason a modified version of this method is incorporated into the phase-splitting algorithm proposed in this thesis.

### 2.2.3 Mass balance approaches

# Wu and Bishnoi's method

Wu and Bishnoi [26] developed a procedure to determine which phases are present in three-phase equilibrium calculations and to generate the appropriate initialization parameters. They affirm that it allows the solution of the equilibrium and mass balance equations (using a Newton-Raphson method) without the difficulties caused by disappearing phases. For isobaric-isothermal flash calculations the following strategy is suggested to establish whether three phases exist and to generate initial values for the unknowns. Initially, a three-phase bubble-point temperature calculation for the feed mixture is executed. If the specified temperature is less than the bubble-point temperature  $(T < T_{B3\phi})$ 

the existence of a vapour phase is ruled out. In such a case the liquid stability test developed by Shah [1] is performed. If only one liquid is found, the rest of the calculations are skipped. Otherwise, if the liquid mixture is unstable, the necessary initial estimates are available and a liquid-liquid calculation is carried out. In the case that  $T > T_{B3\phi}$ , a two-phase dew-point temperature calculation is performed. If the specified temperature is larger than the dew- point temperature  $(T > T_D)$  only a vapour phase can exist and the calculation ends. On the contrary, if  $T < T_D$ , a VL flash calculation is proposed to generate new vapour and liquid phases V' and L'. The stability of the vapour V' is tested by doing an isothermal flash calculation for the V' mixture. If no liquid is generated, V' is stable. Otherwise the vapour V' is unstable and new vapour V" and liquid L''phases are generated. Then a LL equilibrium calculation is done for the liquid mixture L''' obtained by combining L' and L''. If the liquid L''' is stable (claimed possible only when L'' = 0 and L' is stable), the solution is VL equilibrium and the flash calculation is completed. If L''' is unstable, the amounts and compositions of the vapour and the two liquid phases are used as initial values for the three-phase flash. In their calculations, these authors find that both or either of the vapour V' and the liquid L' can be unstable and hence the necessity for the stability tests on both of these phases.

The procedure described above, although straight-forward, can be very inefficient since it constitutes a 'brute-force' method to determine the number and type of phases expected at equilibrium. For instance, when the stable solution is VLL equilibrium, a total of five rigorous equilibrium calculations (two saturation-point determinations, two VL flashes and a LL equilibrium calculation) are required prior to the actual three-phase flash. For a VL stable solution these same five calculations could be needed to obtain the equilibrium conditions. Clearly, the practical use of such an algorithm is limited since it would be too time consuming to be used in a process simulator on a routine basis.

### Nelson's method

The multiphase flash algorithm presented by Nelson [3] is based on the solution of the mass balance and equilibrium equations using a Newton-Raphson technique and has two features that make it attractive to this work. First, particular attention is given to the arrangement of the system of non-linear equations to be solved in order to make them more suitable for solution by the Newton-Raphson method. The resulting equations are of the Rachford-Rice type (for a more thorough treatment on this particular subject see for example Ohanomah's [27] work) and are very similar to those developed in independent form for the flash algorithms used in this thesis (see Appendix A). Secondly, the method initially assumes the presence of three phases (two liquids and a vapour) but the determination of the actual number of phases present, based on the so-called bubble-and dew-point criteria, is incorporated into the flash algorithm. Therefore the presence of only one or two phases is quickly recognized.

Defining the VL equilibrium constants as  $K_{i1} = y_i/x_i^I$  and  $K_{i2} = y_i/x_i^{II}$ , and the phase ratios as  $\psi_1 = L_I/F$  and  $\psi_2 = L_{II}/F$  where  $z, x^I, x^{II}, F, L_I$  and  $L_{II}$  are the compositions and total moles of feed, of liquid phase I and of liquid phase II respectively, and y is the composition of the vapour, the working equations for three-phase calculations derived by Nelson from the mass balance and equilibrium conditions are

$$Q_1(\psi_1,\psi_2) = \sum_{i=1}^n \frac{z_i K_{i2}(1-K_{i1})}{K_{i1}K_{i2} + \psi_1 K_{i2}(1-K_{i1}) + \psi_2 K_{i1}(1-K_{i2})} = 0$$
(2.27)

$$Q_2(\psi_1,\psi_2) = \sum_{i=1}^n \frac{z_i K_{i1}(1-K_{i2})}{K_{i1}K_{i2} + \psi_1 K_{i2}(1-K_{i1}) + \psi_2 K_{i1}(1-K_{i2})} = 0$$
(2.28)

which are well suited for numerical solution to obtain  $\psi_1$  and  $\psi_2$  since they contain no superfluous roots in the domain of interest  $(0 \le \psi_1 \le 1 \text{ and } 0 \le \psi_2 \le 1)$ . These

equations are solved holding  $K_{i1}$  and  $K_{i2}$  fixed using a Newton-Raphson procedure with the required partial derivatives evaluated analytically.

It is shown in [3] that criteria to determine the number of phases at equilibrium can be derived based on the bubble- and dew-point type of equations obtained by examining the behaviour of equations (2.27) and (2.28) at the limiting cases where only one or two phases exist. These criteria apply under circumstances that Nelson defines as "normal", that is, where the functions  $Q_1$ ,  $Q_2$  and  $Q_1 - Q_2$  behave in such a way that in the three- phase region the following relations are satisfied:  $\sum_{i=1}^{n} z_i K_{i1} > 1$ ,  $\sum_{i=1}^{n} z_i/K_{i1} > 1$ ,  $\sum_{i=1}^{n} z_i K_{i2} > 1$ ,  $\sum_{i=1}^{n} z_i/K_{i2} > 1$ ,  $\sum_{i=1}^{n} z_i K_{i1}/K_{i2} > 1$  and  $\sum_{i=1}^{n} z_i K_{i2}/K_{i1} > 1$ .

The criteria for the existence of two phases are summarized as follows:

- 1. Only vapour and liquid phase I exist if  $\sum_{i=1}^{n} z_i K_{i1} > 1$ ,  $\sum_{i=1}^{n} z_i / K_{i1} > 1$  and  $Q_2(\psi_1, 0) < 0$  at the root of  $Q_1(\psi_1, 0) = 0$ ;
- 2. Only vapour and liquid phase II exist if  $\sum_{i=1}^{n} z_i K_{i2} > 1$ ,  $\sum_{i=1}^{n} z_i / K_{i2} > 1$  and  $Q_1(0, \psi_2) < 0$  at the root of  $Q_2(0, \psi_2) = 0$ ;
- 3. Only liquid I and liquid II exist if  $\sum_{i=1}^{n} z_i K_{i1}/K_{i2} > 1$ ,  $\sum_{i=1}^{n} z_i K_{i2}/K_{i1} > 1$  and  $Q_1(\psi_1, 1 \psi_1) > 0$  or  $Q_2(\psi_1, 1 \psi_1) > 0$  at the root of  $Q_1(\psi_1, 1 \psi_1) Q_2(\psi_1, 1 \psi_1) = 0$ .

Nelson demonstrates that no more than one of the above two-phase tests can be satisfied at one time.

The criteria for the existence of only one phase are:

- 1. Only the vapour phase exists if  $\sum_{i=1}^{n} z_i/K_{i1} < 1$  and  $\sum_{i=1}^{n} z_i/K_{i2} < 1$ ;
- 2. Only liquid phase I exists if  $\sum_{i=1}^{n} z_i K_{i1} < 1$  and  $\sum_{i=1}^{n} z_i / K_{i2} < 1$ ;
- 3. Only liquid phase II exists if  $\sum_{i=1}^{n} z_i K_{i2} < 1$  and  $\sum_{i=1}^{n} z_i / K_{i1} < 1$ .

Since the tests for the existence of two phases require a 1- dimensional iterative search, it is recommended to conduct them after the one-phase tests are completed. Nelson stresses that the compositions for all three phases should be calculated even when only one or two phases are present. The reason given for this is that "the calculation of a K-value ordinarily requires the composition of both liquid and vapour phases so these must be provided even if one or both of these phases do not exist". Thus, when convergence is achieved in the inner Newton-Raphson iteration loop (to solve for  $\psi_1$  and  $\psi_2$ ) with a given set of K-values, these are reevaluated in an outer loop at the new-found compositions and the three-phase flash is continued. This procedure is repeated until compositions and K-values converge to a pre- specified degree of accuracy.

Although Nelson recognizes the importance of detecting the presence of only one or two phases in a potential three-phase system, the method described above suggests that non-existing phases should not be eliminated but carried along the computation. This is justified for the case where it is determined that only two phases are present by stating that the phase ratios are already at hand from the 1-D iterative search procedure (in the two-phase existence test) and that only the phase compositions remain to be calculated. Nonetheless, in this scheme the recognition of the correct number of phases seems of small significance since a three-phase calculation is performed irrespective of the phases predicted by the proposed criteria. Another limitation of the method is that the criteria to test for the existence of one or two phases are subject to the assumption of three phases being present under the "normal" conditions described above. No actions are suggested when these conditions are violated and the circumstances leading to their violation are not specified.

# Chapter 3

# Proposed algorithm

After reviewing the main phase-splitting methods presented in Chapter 2, it can be concluded that most of them seem inadequate to fulfill the objectives of this work, namely provide a phase- splitting algorithm which can accurately and efficiently:

- 1. Predict the number and type of phases formed at equilibrium by a multicomponent system at a specified temperature, pressure and overall composition without performing a flash calculation.
- 2. Provide composition estimates for the phases present in order to initiate the flash calculation when required.

Without doubt, Michelsen's stability analysis [4] is the method that most closely appears to satisfy these objectives. Because of this and since it has other desirable features, it was decided to incorporate it into the algorithm proposed in this thesis. In spite of appearing very robust at first instance, when Michelsen's method was implemented in the algorithm to be discussed, drawbacks other than the ones mentioned in Section 2.2.2 became evident. This lead to several proposed modifications of the original Michelsen method which exploit its advantages and attempt to eliminate its pitfalls. Basically the flaws found are that under certain circumstances the tangent plane criterion is unable to detect an incipient phase which is present at equilibrium and in other cases a superfluous trial phase not corresponding to the equilibrium solution may be found. These problems and the solutions proposed to avoid them are discussed in more detail in the following sections.

### 3.1 General scheme

The following scheme is used in the phase-splitting algorithm developed in this work (a more detailed description of the procedure in step by step form is given next). Initially, it is established whether the multicomponent mixture considered as a monophasic system is liquid-like or vapour-like at the specified temperature, pressure and overall composition, depending on which state has a lower Gibbs energy. Then, a phase search is conducted based on an improved version of Michelsen's stability analysis, that is, incipient phases are sought in order to minimize the system's Gibbs energy by splitting it into two or three phases. If no incipient phases are detected the calculation terminates and the system is declared stable as a single phase (either as a vapour or as a liquid depending on whether the original system was found to be vapour-like or liquid-like). When only one incipient phase is found, it is assumed that the stable system is in the two phase region (with phases corresponding to the incipient phase and the monophasic system) and then the appropriate flash calculation can be conducted. If two incipient phases are detected, then depending on the nature of these and of the initial monophasic system, a two or threephase solution is postulated as described below. For the case that three incipient phases are found (possible only if the original system is liquid-like as will be seen) a three-phase solution is assumed. In all instances where a VLL (three-phase) solution is feasible, a phase removal test is conducted to assure that three phases are present at equilibrium. This test, which allows dropping any superfluous phase found in the search, can be done based on the criteria to reduce the number of phases in flash calculations developed in [5], on Nelson's bubble- and dew-point equations approach [3] or using a coupled method suggested in this work.

# Phase-splitting algorithm

The phase-splitting algorithm developed consists of the following steps:

- At the given temperature, pressure and global composition, the Gibbs energy of mixing of the system is calculated considering it as a homogeneous vapour and as a homogeneous liquid. The original system (source phase) is considered either vapour- like or liquid-like depending on which of the hypothetical homogeneous systems has a lower Gibbs energy of mixing. If the system is vapour-like, Step 8 is executed, otherwise Step 2 follows.
- 2. The system is liquid-like. Using the phase-search algorithm described in Section 3.2.1 (modified Michelsen method), a vapour trial phase is searched. If the search fails, a stable vapour phase may not exist, however it could have been missed because of the limitations in Michelsen's algorithm (this is illustrated and discussed in Section 3.2). Irrespective of the outcome of this search, the next step is followed.
- 3. One or more incipient liquid phases (originated from the liquid-like system) are sought using the phase-search procedure described in Section 3.2.2. If no vapour phase was found in Step 2, Step 6 is executed.
- 4. If no liquid trial phases were found in the previous step, then the system can be assumed stable as a VL mixture (since an incipient vapour has been found). However, because an incipient liquid may have been overlooked in the original search (modified Michelsen's method), an additional search can be conducted (this optional search is also described in Section 3.2.2). If still a new liquid is not found, it is considered that the system is stable as VL. At this point the phase-splitting calculation ends and a VL flash can be carried out with initial compositions of the vapour being y = y' and of the liquid x = z (where the prime indicates the

composition of the incipient phase found). If an incipient liquid phase is detected in the additional search, then the system is potentially a three-phase mixture (with initial compositions y = y',  $x^{I} = x'$  and  $x^{II} = z$ ) and a phase removal test is conducted as indicated in Section 3.4 to determine whether the system is stable as VL, LL or VLL. Then the corresponding flash calculation can be performed.

- 5. If one or two incipient liquid phases are detected using the modified Michelsen method (Step 3), the system can form three phases (with initial compositions  $x^{I} = x'$ ,  $x^{II} = z$  or  $x^{I} = x'^{I}$ ,  $x^{II} = x'^{II}$  and y = y') and any of the phase elimination tests proposed can be conducted to determine the configuration of the system. The phase-splitting calculation terminates here and again the corresponding flash can be executed to determine the equilibrium compositions and phase ratios.
- 6. No incipient vapour phase exists for the liquid-like system. If no liquid trial phases could be found in Step 3, then the original system is stable as a liquid and no further calculations are required.
- 7. If one or two liquid trial phases were found (in Step 3) the system can be assumed stable as LL. However, the possibility of existence of a vapour which may have been missed in the search can be checked (optional) by carrying out an additional search (see Section 3.3.1). If no incipient vapour is found after this more thorough search, the stable solution is considered to be a LL mixture with starting compositions  $x^{I} = x'$  and  $x^{II} = z$  or  $x^{I} = x'^{I}$  and  $x^{II} = x'^{II}$ . At this point the calculation ends and the two-phase flash can be performed, except if in the additional search a vapour phase is detected. In this case a phase removal test is again executed to determine the stable system and the corresponding flash can then be conducted.

- 8. The system is vapour-like. One or more liquid trial phases are looked for using the modified Michelsen phase search. If incipient liquid phases are not found, the system is stable as a vapour and the calculation ends.
- 9. If only one liquid is found in the previous step, the system can be assumed stable as a VL mixture. In this case a second incipient liquid may have been missed in the initial search, thus an additional search may also be conducted. If a new liquid trial phase is not found, the system is considered stable as VL having initial compositions y = z, x = x' and the calculation ends. A two-phase flash may then follow to obtain the equilibrium compositions and phase fraction. However, if a new liquid trial phase is obtained in the additional search, the stable solution among the possible VL, LL or VLL solutions is determined with the phase number reduction method chosen (Section 3.4) starting with y = z,  $x^{I} = x'^{I}$  and  $x^{II} = x'^{II}$ and afterwards the flash calculation can be conducted.
- 10. If two incipient liquid phases were found in Step 8, again it is possible to have either a two or three-phase solution at equilibrium. The phase removal test is then performed (with the same starting values as above) to determine the configuration of the system followed by the specific type of flash calculation if required.

The flow chart representing the phase-splitting algorithm just described is depicted in Figure 3.6.

Some important aspects considered when implementing the algorithm just seen, other than the phase search strategy and the phase removal tests, are described next. Throughout this work, the molar Gibbs energy of mixing rather than just the molar Gibbs energy is used in the calculations since the former property gives a measure of the deviation of the system's condition from ideality. This can be seen when considering the definition of the molar Gibbs energy of mixing for a homogeneous multicomponent phase at fixed



Figure 3.6: Flow chart of the phase-splitting algorithm proposed in this work.

temperature and pressure, which is the mixture's molar Gibbs energy g less the mole fraction weighted sum of the pure components Gibbs energies  $g_i$ :

$$\Delta g = g - \sum_{i=1}^{n} z_i g_i = RT \sum_{i=1}^{n} z_i \ln \hat{a}_i$$
(3.29)

where  $z_i$  represents the global composition of each component and  $\hat{a}_i$  its activity in the mixture. When an equation of state is used to represent the properties of both the vapour and liquid phases, the activity can be expressed for either phase as

$$\hat{a}_i = \frac{\hat{f}_i}{f_i^o} = \frac{z_i \hat{\phi}_i P}{\phi_i^o P^o}$$
(3.30)

where  $z_i$  is the composition of the corresponding phase (either a vapour or a liquid);  $f_i$ and  $\phi_i$  are the fugacity and fugacity coefficients of component *i* at the given *T*, *P* and composition. Variables with a hat "^" indicate the property in the mixture and superscript "o" indicates the property in the standard state. As Walas [28] points out, in order to take advantage of the requirement of equality of fugacities of individual components in all phases at equilibrium, the standard states of condensed and vapour phases must be consistent. Equally, when a Gibbs energy analysis is done to determine equilibrium states, consistency in the standard states must be observed. Since equation (3.29) is obtained for isothermal conditions, the standard state temperature must be the same as that of the mixture, but the pressure and composition of the standard state are arbitrary. Because cubic equations of state can represent continuity of states, the same standard state can be chosen for the vapour and liquid phases. The most convenient one for the fugacity is the fugacity of the pure component in the vapour phase at a pressure of one atmosphere. This gives the advantage of consistency and ease of evaluation since the values of  $\phi_i^V$  are generally unity at 1 atm. Considering that the reference state is arbitrary, even if  $\phi_i^V$  is different from one (for example for components that associate in the vapour phase), the value of  $f_i^o$  is set to 1 atm. Taking this into account, for the vapour phase equation (3.30) reduces to

$$\hat{a}_i^V = \frac{y_i \hat{\phi}_i^V P}{1 \ atm} \tag{3.31}$$

and for the liquid phase it can be expressed as

$$\hat{a}_i^L = \frac{x_i \hat{\phi}_i^L P}{1 \ atm} \tag{3.32}$$

In short, the dimensionless molar Gibbs energy of mixing is calculated as

$$\Delta g^{V}/RT = \sum_{i=1}^{n} y_{i} \ln[y_{i} \hat{\phi}_{i}^{V}(T, P, y) P(atm)]$$
(3.33)

for the vapour phase and as

$$\Delta g^L/RT = \sum_{i=1}^n x_i \ln[x_i \hat{\phi}_i^L(T, P, x) P(atm)]$$
(3.34)

for the condensed phases.

To obtain the fugacity coefficients, either the Soave-Redlich-Kwong or the Peng-Robinson equations of state are used in this work. If three real density roots are found when solving for the cubic equation, the evaluation of the fugacity coefficients for the vapour and liquid phases is done using respectively the vapour-like and the liquid-like roots. If at the specified conditions only one real density root exists, this root is used for the corresponding phase and an extrapolated value of density is used for the phase that is infeasible as indicated in [5] (also described in [10]). This extrapolation algorithm has been used very successfully in this work to avoid the so called 'trivial solution' in the calculation of the density roots. It should be mentioned that in these cubic equations the classical mixing rules (allowing the use of binary interaction coefficients) are used with the modifications proposed by Mathias [9] to include polar compounds and supercritical components.

### 3.2 Phase search strategy

Before going into the details of the search strategy for the vapour and liquid phases, flaws in the detection of incipient phases when using Michelsen's method under certain circumstances will be addressed. Specifically, failure to correctly predict the phases at equilibrium may occur when three potential phases exist and the tangent plane criterion is unable to detect one of the incipient phases for a range of feed compositions. These flaws, which are not foreseen by Michelsen, are shown here graphically for the case of binary mixtures but are likely to happen in multicomponent systems.

Three cases where Michelsen's method is prone to failure are expected when incorporated into the phase-splitting algorithm proposed. These are (refer to the general scheme described earlier):

- i) when the system is vapour-like and one liquid is found, another incipient liquid could have been bypassed;
- ii) when the system is liquid-like and one or two incipient liquid phases are found but no vapour is detected, a vapour phase may have been missed, and
- iii) when the system is liquid-like, no incipient liquids are detected and a trial vapour

is found, a liquid trial phase may have been bypassed.

In order to eliminate these failures, an optional additional search is suggested in this work to detect possibly missed phases.

Figure 3.7 shows the Gibbs energy of mixing for a binary mixture of water and nhexane as obtained by Heidemann [29] which can be used to exemplify case i). At the feed composition the system is vapour-like and the tangent line to the Gibbs energy curve at this point (dotted line) indicates the presence of one incipient liquid phase (corresponding to a negative minimum F). Hence, a VL solution would be predicted using the tangent plane criterion since only one incipient liquid is found from the vapour-like source phase (the n-hexane rich liquid phase is bypassed). However, the stable solution is that of a LL mixture as indicated by the lower dashed line since in this state a minimum in Gibbs energy is achieved. It should be noted that failure to detect the n-hexane rich phase can occur for a wide range of feed compositions, not only in the vicinity of the point zconsidered.

In Figure 3.8 case ii) is shown for a hypothetical mixture which is liquid-like at the feed composition z. Again in this case Michelsen's method indicates the presence of one incipient liquid phase. Due to the slope of the tangent line (dotted line) to the Gibbs energy curve at z, a vapour incipient phase is missed in the phase search. This would lead to the prediction of a LL solution instead of the more stable VL solution which has a lower Gibbs energy. In this case the vapour incipient phase is bypassed only for feed compositions in the vicinity of point z.

Although no graphical representation for case iii) could be depicted for a binary system, evidence of failure under the conditions corresponding to this case was found in calculations performed for multicomponent systems.

The additional phase search method proposed is intended to find phases which may



Figure 3.7: Erroneous prediction of VL equilibrium due to bypass of a liquid incipient phase when using Michelsen's method for the n-hexane-water system at 378 K and 4.08 atm (diagram modified from Heidemann).



Figure 3.8: Bypass of a vapour incipient phase when using the tangent plane criterion leading to an incorrect LL equilibrium prediction in a hypothetical binary mixture.



Figure 3.9: Additional phase search to find the n-hexane rich liquid incipient phase for the n-hexane-water system (378 K and 4.08 atm) leading to the correct prediction of LL equilibria.

have been bypassed in the initial search. Because a more thorough search in not always required or may not be desired by the user, it is left as an option in the overall algorithm. The rationale behind the method is that after the modified Michelsen search is completed, whenever cases i) to iii) occur, a pseudo- tangent plane is constructed and used with the tangent plane criterion in the same fashion as the actual tangent plane but only to search for a specific phase. For binary systems this additional search can again be illustrated graphically for cases i) and ii) described above. In the former case, the liquid trial phase found in the initial search and the feed point are used to obtain the slope of the line connecting them which characterizes the pseudo- tangent line used for the additional liquid phase search as illustrated in Figure 3.9. This figure shows how a value of F < 0 is obtained when considering the pseudo-tangent line which allows the n-hexane rich phase missed in Figure 3.7 to be found, leading to the correct prediction of LL equilibria.

For case ii) when in the initial search only one liquid incipient phase is obtained this and

stable system.



Figure 3.10: Detection of an incipient vapour phase by means of the additional search proposed for the binary mixture of Figure 3.8 leading to the proper prediction of a VL

the liquid-like feed point are used to construct the pseudo-tangent line. This is shown in Figure 3.10 in which the vapour phase missed with Michelsen's method (see Figure 3.8) can be found thanks to the additional search strategy giving the correct prediction of a VL stable system. When two incipient liquids are found in the initial search for case ii) these points are used to construct the pseudo-tangent line needed to look for the possibly missed vapour phase (although not illustrated this case is very similar to that shown in Figure 3.10)

A similar approach is used for case iii) but in this instance the points required for the pseudo-tangent line are the liquid-like feed and the vapour trial phase already found in order to search for an incipient liquid.

Although the pseudo-tangent method was found to work quite well for the additional phase search in binary mixtures, a direct upgrading for systems with three or more components could not be made since the multi-dimensionality of the problem did not allow a successful analogous approach. For binary systems the feed point and the point corresponding to the incipient phase found (cases i) to iii)) or the two incipient liquids (case ii)) define the pseudo-tangent line required in the additional phase search method proposed. For multicomponent mixtures  $(n \ge 3)$  the orientation of the pseudo- tangent plane or hyperplane can not be defined with only two points since an infinite number of planes can pass through them. To overcome this problem, a variation in the two component method was devised in which a pseudo-tangent plane orthogonal to the Gibbs energy of mixing axis is always used in the additional search. This variation, which can be visualized for ternary systems as having a pseudo-tangent plane parallel to the composition plane in a 3-D  $\Delta g$  versus mole fraction diagram, worked satisfactorily in many cases but in others it did not aid decisively to find new phases. The implementation of the additional phase search (for binary and higher systems) is described in Section 3.2.1 when searching for a vapour (case ii)) and in Section 3.2.2 when searching for a liquid (cases i) and iii)).

Another aspect regarding the phase search strategy which should be mentioned here is the numerical method used (in the modified Michelsen method) to solve for the mole numbers of the incipient phases from the stationarity condition (equation 2.19). As Michelsen [4] suggests, direct substitution or accelerated direct substitution are the obvious candidates to solve for this equation since it is well suited for these methods when arranged in the form of equation 2.21. According to him, when using direct substitution the approach to the solution will be rapid if the dependence of the fugacity coefficients on composition is weak since the method has a linear rate of convergence. Because the calculations performed when LL or VLL equilibrium is involved imply highly nonideal systems (in which fugacity coefficients are strongly dependent on composition), it is expected that convergence will be slow if direct substitution is used. This calls for acceleration methods with faster convergence rates that do not require excessive computational effort. In their evaluation of Michelsen's method in LL calculations, Swank and Mullins [30] find that the most efficient method to accelerate the direct iteration calculation is the Newton-Raphson method when compared to the General Dominant Eigenvalue method (GDE) [23] and to Broyden's method [24]. They recommend a direct iteration scheme accelerated with the GDE method and the Newton-Raphson method for a maximum of 15 iterations. For final convergence the use of a Gibbs energy minimization with Murray's method [25] is suggested after 15 iterations. Murray's minimization of the Gibbs energy is recommended for every iteration if greater reliability is required but at the sacrifice of speed.

An efficient method for accelerated direct substitution not considered in [30], which does not require gradient information, is Wegstein's method [31]. It has been shown in [5] that this method has good convergence characteristics when used to solve for equilibrium compositions in flash and saturation point calculations with a minimal added computational load. Given its desirable features to accelerate compositions as independent variables and since the objective function (equation 2.21) is well suited for its use (i.e. is of the form x = f(x)), Wegstein's method is the numerical method chosen in the modified Michelsen's phase search.

#### 3.2.1 Vapour phase search

The objective of the vapour phase search method proposed as part of the phase-splitting algorithm described earlier is to find the vapour composition y that satisfies the stationarity condition (analogous to equation (2.19)):

$$\ln Y_i + \ln \hat{\phi}_i^V(y) - h_i^L = 0 \tag{3.35}$$

for i = 1, 2, ..., n, where n is the number of components in the mixture. The steps of the proposed search method are:

# Modified Michelsen's method. Vapour phase search

1. Given composition estimates for the incipient vapour to be searched, the initial values of the mole number variables  $Y_i$  (independent variables) are set to

$$Y_i^{(0)} = y_i^{(0)} = y_{i \ init} \tag{3.36}$$

and parameters  $\exp(h_i^L)$  are obtained as

$$\exp(h_i^L) = z_i \hat{\phi}_i^L(z) \tag{3.37}$$

where  $\hat{\phi}_i^L(z)$  is the fugacity coefficient of *i* for the system as a liquid at the feed composition (already obtained for the liquid-like system by taking  $x_i = z_i$  when evaluating  $\Delta g^L/RT$  from equation (3.34)).

2. From the chosen equation of state the fugacity coefficients  $\phi_i^{(0)} = \hat{\phi}_i^{V(0)}(T, P, y^{(0)})$ are obtained and variables  $Y_i^{(1)}$  are calculated as (refer to equation (2.21))

$$Y_i^{(1)} = \exp(h_i^L - \ln \phi_i^{(0)}) = \exp(h_i^L) / \phi_i^{(0)} = f(Y_i^{(0)})$$
(3.38)

3. The corresponding vapour compositions are calculated by normalization of the mole numbers:

$$y_i^{(1)} = \frac{Y_i^{(1)}}{\sum_{i=1}^n Y_i^{(1)}}$$
(3.39)

and the values of  $\phi_i^{(1)}$  are obtained with these compositions from the equation of state. By direct substitution the updated mole numbers are calculated as

$$Y_i^{(2)} = \exp(h_i^L) / \phi_i^{(1)} = f(Y_i^{(1)})$$
(3.40)

4. Start of the iterative cycle. If conditions are met to use Wegstein's method, new values of  $Y_i^{(k+1)}$  are obtained from

$$Y_i^{(k+1)} = (1 - t_i)Y_i^{(k)} + t_i f(Y_i^{(k)})$$
(3.41)

where k indicates the iteration number  $(k=1,2,\ldots,k_{max})$  and  $t_i$ , the step size parameter, is calculated as

$$t_{i} = \frac{1}{1 - s_{i}} \tag{3.42}$$

with

$$s_{i} = \frac{f(Y_{i}^{(k)}) - f(Y_{i}^{(k-1)})}{Y_{i}^{(k)} - Y_{i}^{(k-1)}}$$
(3.43)

The values of  $t_i$  obtained are limited in such a way that  $t_i = t_{max}$  when  $t_i > t_{max}$ and  $t_i = -t_{max}$  if  $t_i < -t_{max}$ .

If direct substitution is used (i.e.  $t_i$  is set to 1 for every component) then

$$Y_i^{(k+1)} = f(Y_i^{(k)}) \tag{3.44}$$

# 5. The new compositions are obtained as

$$y_i^{(k+1)} = \frac{Y_i^{(k+1)}}{\sum_{i=1}^n Y_i^{(k+1)}}$$
(3.45)

and the corresponding fugacity coefficients are calculated from the equation of state:  $\phi_i^{(k+1)} = \hat{\phi}_i^{V(k+1)}(T, P, y^{(k+1)}).$ 

6. The objective function is then evaluated

$$\Im(Y_i) = \sum_{i=1}^n |Y_i^{(k+1)} - Y_i^{(k)}|$$
(3.46)

as well as parameters  $g^*, \beta$  and r:

$$g^* = 1 + \sum_{i=1}^{n} Y_i^{(k+1)} (\ln Y_i^{(k+1)} + \ln \phi_i^{(k+1)} - h_i^L - 1)$$
(3.47)

$$\beta = \sum_{i=1}^{n} (Y_i^{(k+1)} - z_i) (\ln Y_i^{(k+1)} + \ln \phi_i^{(k+1)} - h_i^L)$$
(3.48)

$$r = 2g^*/\beta \tag{3.49}$$

- 7. To monitor the unsuccessful convergence behaviour of the method, the following checks are made:
  - (a) Check for convergence of the current composition vector to the feed composition, i.e. check if  $y^{(k+1)}$  is approaching z.
  - (b) Check if the value of r is approaching 1 in consecutive iterations.
  - (c) Check if  $\sum_{i=1}^{n} Y_i^{(k+1)}$ , when < 1, is decreasing in consecutive iterations.
  - (d) Check if the objective function  $\Im(Y_i)$  is increasing in consecutive iterations.
  - (e) Check if  $g^*$ , when < 0, is not changing significantly in consecutive iterations.
  - (f) Check if the objective function increases after using Wegstein's method. If so, the acceleration method is used less frequently until the objective function decreases again after applying it.
- 8. Test for convergence either to a minimum in Gibbs energy or to the trivial solution: -Convergence to a minimum in Gibbs energy. If  $\Im(Y_i) < \varepsilon$  or if check (e) is satisfied, convergence has been achieved and Step 10 is executed.

-Convergence to the trivial solution. If |r - 1| < 0.2 and  $g^* < 10^{-3}$  or if check (a) is satisfied or if any of checks (b) through (d) are met in two consecutive iterations, Step 11 is carried out.

If none of the above tests are satisfied, the method continues in the next step.

9. The functions

$$f(Y_i^{(k+1)}) = \exp(h_i^L) / \phi_i^{(k+1)}$$
(3.50)

are calculated and the variables for the next iterative loop are assigned by setting k = k + 1. If  $k < k_{max}$  Step 4 is executed, otherwise convergence has failed and Step 12 is performed.

10. If  $\sum_{i=1}^{n} Y_i^{(k+1)} > 1$  (or equivalently  $g^* < 0$ ) a stationary point corresponding to an incipient vapour with composition  $y^{(k+1)}$  has been found. This means that the original system (liquid-like) is unstable and the appearance of the trial vapour phase decreases the Gibbs energy of the global system.

If  $\sum_{i=1}^{n} Y_i^{(k+1)} \leq 1$  (corresponding to  $g^* \geq 0$ ) no vapour phase has been found in the search and the original system is stable with respect to the formation of an incipient vapour.

- 11. The trivial solution in which y = z has been found or no solution to equation (3.35) could be obtained. In either case it is considered that the search is unsuccessful, i.e. no incipient vapour could be found and a split of the original system into a VL system does not occur.
- 12. The maximum number of iterations allowed has been reached forcing the calculation to end. In spite of not having reached convergence, if the current composition

values for the vapour trial phase are such that  $\sum_{i=1}^{n} Y_{i}^{(k+1)} > 1$  it is considered that an incipient vapour has been found. If this condition is not met, the search is designated unsuccessful.

# Additional vapour phase search

When the system is liquid-like and one or two liquid phases are detected in the phasesplitting algorithm but no vapour is found when applying the phase search method just seen, an additional search may be conducted to check for the presence of an incipient vapour (case ii)).

For binary systems, the slope of the pseudo-tangent line needed for the proposed additional search (see Figure 3.10) is obtained according to:

$$\frac{\partial \Delta g}{\partial z_i} = \frac{\Delta g^L(z) - \Delta g^L(x')}{z_i - x'_i}$$
(3.51)

when this line passes through the feed point and the one incipient liquid found or from

$$\frac{\partial \Delta g}{\partial z_i} = \frac{\Delta g^L(x'^{II}) - \Delta g^L(x'^{I})}{x_i^{II} - x_i^{I}}$$
(3.52)

when two incipient liquid phases with compositions  $x'^{I}$  and  $x'^{II}$  have been found. This slope can be used to calculate a pseudo-chemical potential of mixing  $\Delta \hat{\mu}_{p_{i}}^{L}$  from the following thermodynamic relation (from Walas [20], equations (2.56), (2.65) and (7.4)):

$$\Delta \hat{\mu}_{p_i}^L = \Delta g^L(z) - \sum_{k \neq i}^n z_k \left(\frac{\partial \Delta g}{\partial z_k}\right)_{z_{j \neq i,k}}$$
(3.53)

which in turn is needed to obtain a pseudo-fugacity coefficient at the feed composition

defined as

$$\ln \hat{\phi}_{p_{i}}^{L}(z) = \frac{\Delta \hat{\mu}_{p_{i}}^{L}}{RT} - \ln(z_{i}P)$$
(3.54)

With this variable a pseudo-parameter  $h_{p_i}^L$  analogous to  $h_i$  (see equation (2.17)) can be obtained from

$$h_{p_{i}}^{L} = \ln z_{i} + \ln \hat{\phi}_{p_{i}}^{L}(z)$$
(3.55)

By combining equations (3.54) and (3.55)  $h_{p_i}^L$  can be expressed as

$$h_{p_{i}}^{L} = \frac{\Delta \hat{\mu}_{p_{i}}^{L}}{RT} - \ln P$$
(3.56)

Once  $h_{p_i}^L$  has been calculated, the additional vapour phase search can be conducted by using the same search method described above but substituting  $\exp(h_i^L)$  for  $\exp(h_{p_i}^L)$  in all instances.

For systems having 3 or more components the slope of the pseudo- tangent plane in all directions is set to zero:

$$\frac{\partial \Delta g}{\partial z_i} = 0 \tag{3.57}$$

and because of this condition equation (3.53) reduces to

$$\Delta \hat{\mu}_{p_i}^L = \Delta g^L(z) \tag{3.58}$$
Consequently,  $h_{p_i}^L$  can be easily obtained as

$$h_{p_i}^L = \frac{\Delta g^L(z)}{RT} - \ln P \tag{3.59}$$

for all components. Again  $\exp(h_{p_i}^L)$  is used instead of  $\exp(h_i^L)$  to conduct the additional search.

# 3.2.2 Liquid phases search

When searching for one or two liquid phases with the phase- splitting algorithm proposed, it is the liquid compositions x that satisfy the following stationarity conditions (equivalent to equation (2.19)) that are sought:

$$\ln X_i + \ln \hat{\phi}_i^L(x) - h_i^L = 0 \tag{3.60}$$

when the original system is liquid-like and

$$\ln X_i + \ln \hat{\phi}_i^L(x) - h_i^V = 0 \tag{3.61}$$

if it is vapour-like. The liquid phase search algorithm is similar to the one already described for the vapour. One of the major differences is that because in the liquid search at least two distinct liquid phases are sought simultaneously, more than one composition vector is required in the calculations. Thus, instead of having a single vector of n components, an array of m initial composition vectors is used in the search. As the calculation proceeds the number of vectors can be reduced if required such that a maximum of two vectors are considered for final convergence. The steps of the proposed search method are:

# Modified Michelsen's method. Liquid phases search

1. Having the array of liquid composition estimates, the starting values of the mole number variables  $X_{j,i}$  are assigned as

$$X_{j,i}^{(0)} = x_{j,i}^{(0)} = x_{j,i \text{ init}}$$
(3.62)

where subscript j indicates the vector number (j = 1, 2, ..., m) and subscript i the component index (i = 1, 2, ..., n). Then the parameters  $\exp(h_i^L)$  or  $\exp(h_i^V)$  are obtained as

$$\exp(h_i^L) = z_i \hat{\phi}_i^L(z) \tag{3.63}$$

or as

$$\exp(h_i^V) = z_i \hat{\phi}_i^V(z) \tag{3.64}$$

depending on whether the system is liquid-like or vapour-like (for simplicity, in either case these parameters will be written as  $\exp(h_i)$ ). The values of  $\hat{\phi}_i^L(z)$  and  $\hat{\phi}_i^V(z)$  are at hand since they are required to evaluate equations (3.33) and (3.34) at the feed composition (Step 1 in the phase-splitting algorithm). 2. The fugacity coefficients  $\phi_{j,i}^{(0)} = \hat{\phi}_{j,i}^{L(0)}(T, P, x_j^{(0)})$  are evaluated from the equation of state and the mole numbers  $X_{j,i}^{(1)}$  are calculated according to:

$$X_{j,i}^{(1)} = \exp(h_i - \ln \phi_{j,i}^{(0)}) = \exp(h_i) / \phi_{j,i}^{(0)} = f(X_{j,i}^{(0)})$$
(3.65)

3. The new liquid compositions are calculated by normalization of the mole numbers:

$$x_{j,i}^{(1)} = \frac{X_{j,i}^{(1)}}{\sum_{i=1}^{n} X_{j,i}^{(1)}}$$
(3.66)

and the corresponding fugacity coefficients  $\phi_{j,i}^{(1)}$  are evaluated. Using direct substitution, the updated mole numbers are calculated as

$$X_{j,i}^{(2)} = \exp(h_i) / \phi_{j,i}^{(1)} = f(X_{j,i}^{(1)})$$
(3.67)

4. Start of the iterative cycle. If the necessary conditions are met to use Wegstein's method, accelerated compositions  $X_{j,i}^{(k+1)}$  are obtained as

$$X_{j,i}^{(k+1)} = (1 - t_{j,i})X_{j,i}^{(k)} + t_{j,i}f(X_{j,i}^{(k)})$$
(3.68)

where k indicates the iteration number  $(k = 1, 2, ..., k_{max})$  and parameters  $t_{j,i}$  are calculated as

$$t_{j,i} = \frac{1}{1 - s_{j,i}} \tag{3.69}$$

with

$$s_{j,i} = \frac{f(X_{j,i}^{(k)}) - f(X_{j,i}^{(k-1)})}{X_{j,i}^{(k)} - X_{j,i}^{(k-1)}}$$
(3.70)

The values of  $t_{j,i}$  are limited in such a way that  $t_{j,i} = t_{max}$  if  $t_{j,i} > t_{max}$  and  $t_{j,i} = -t_{max}$  if  $t_{j,i} < -t_{max}$ .

When direct substitution is used then the updated mole numbers are obtained as

$$X_{j,i}^{(k+1)} = f(X_{j,i}^{(k)}) \tag{3.71}$$

5. The new mole fraction compositions are obtained according to

$$x_{j,i}^{(k+1)} = \frac{X_{j,i}^{(k+1)}}{\sum_{i=1}^{n} X_{j,i}^{(k+1)}}$$
(3.72)

and the corresponding fugacity coefficients are obtained from the equation of state:  $\phi_{j,i}^{(k+1)} = \hat{\phi}_{j,i}^{L(k+1)}(T, P, x_j^{(k+1)}).$ 

6. The objective functions are evaluated for each composition vector

$$\Im_{j}(X_{j,i}) = \sum_{i=1}^{n} |X_{j,i}^{(k+1)} - X_{j,i}^{(k)}|$$
(3.73)

as well as parameters  $g_j^*$ ,  $\beta_j$  and  $r_j$ :

÷

$$g_{j}^{*} = 1 + \sum_{i=1}^{n} X_{j,i}^{(k+1)} (\ln X_{j,i}^{(k+1)} + \ln \phi_{j,i}^{(k+1)} - h_{i} - 1)$$
(3.74)

$$\beta_j = \sum_{i=1}^n (X_{j,i}^{(k+1)} - z_i) (\ln X_{j,i}^{(k+1)} + \ln \phi_{j,i}^{(k+1)} - h_i)$$
(3.75)

$$r_j = 2g_j^*/\beta_j \tag{3.76}$$

- 7. The unsuccessful convergence behaviour of the method is monitored by doing the following checks on every composition vector j:
  - (a) Check for convergence to the feed composition, i.e. check if  $x_j^{(k+1)}$  is approaching z.
  - (b) Check if the value of  $r_j$  is approaching 1 in consecutive iterations.
  - (c) Check if  $\sum_{i=1}^{n} X_{j,i}^{(k+1)}$ , when < 1, is decreasing in consecutive iterations.
  - (d) Check if the objective function  $\Im_j(X_{j,i})$  is increasing in consecutive iterations.
  - (e) Check if  $g_j^*$ , when < 0, is not changing significantly in three consecutive iterations.
  - (f) Check if the objective function increases for any composition vector after using Wegstein's method. If so, the acceleration method is used less frequently until the objective function decreases again after applying it.

- (g) If all values of  $g_j^*$  are > 0, the composition vector with the largest value of  $|r_j 1|$  is marked.
- 8. Test for convergence of the composition vectors either to a minimum in Gibbs energy or to the trivial solution:

-Convergence to a minimum in Gibbs energy. If  $\Im_j(X_{j,i}) < \varepsilon$  or if check (e) is satisfied and in either case  $\sum_{i=1}^n X_{j,i}^{(k+1)} > 1$ , a liquid trial phase has been found. Its composition and fugacity coefficients are saved.

-Convergence to the trivial solution. If |r - 1| < 0.2 and  $g^* < 10^{-3}$  or if check (a) is satisfied or if any of checks (b) through (d) are met in two consecutive iterations, a trivial solution has been found for the corresponding vector.

9. Each pair of composition vectors are tested to be significantly different from one another to avoid convergence to a common vector. It is considered that two vectors are approaching the same value if

$$\sum_{i=1}^{n} |X_{j,i} - X_{j',i}| \le \varepsilon \tag{3.77}$$

10. The following composition vectors are eliminated from the search:

-Vectors that have converged to a minimum in Gibbs energy or to the trivial solution.

-Vectors that are not likely to converge, namely those that are not marked when check (g) is satisfied.

-Vectors that have converged to a common value, except the one with the smallest objective function.

Whenever a composition vector is eliminated, the value of m (the number of vectors in the array) is set to m - 1. If all the composition vectors have been eliminated, the calculation terminates and Step 12 is executed.

11. The functions

$$f(X_{j,i}^{(k+1)}) = \exp(h_i) / \phi_i^{(k+1)}$$
(3.78)

are calculated and the variables for the next iterative loop are assigned by setting k = k + 1. If  $k < k_{max}$  Step 4 is executed, otherwise the next step is followed.

12. If at least two trial phases are found, again a test for the possible convergence to a common value is performed again as indicated in Step 9. All common vectors but one are eliminated.

If two incipient liquid phases are found, it is considered that the presence of two liquids leads to a more stable solution than the presence of a single liquid. (In the odd case that three different composition vectors are found, the two with the least Gibbs energy of mixing are considered stable).

If one incipient liquid phase is found the original system is unstable and the appearance of the new phase will decrease it's Gibbs energy.

If no incipient liquid phases are found, the search has failed and the original system is stable with respect to the formation of a new liquid phase.

# Additional liquid phase search

The attempt to find a liquid phase which may have been bypassed in the initial search of the phase-splitting algorithm can be conducted when cases i) and iii) (described at the beginning of Section 3.2) occur. In similar form as for the additional vapour search in case ii), the additional search for a liquid phase consists on finding a pseudo-parameter  $h_{p_i}^V$  or  $h_{p_i}^L$  (for cases i) or iii) respectively) and conducting the modified Michelsen method with them. These pseudo-parameters define the pseudo-tangent plane necessary to apply the tangent plane criterion for the additional search.

When the system has 2 components,  $h_{p_i}^V$  is calculated as

$$h_{p_i}^V = \frac{\Delta \hat{\mu}_{p_i}^V}{RT} - \ln P \tag{3.79}$$

and  $h_{p_i}^L$  according to equation (3.56). The pseudo- chemical potential of mixing for the system as a vapour,  $\Delta \hat{\mu}_{p_i}^V$ , is obtained similarly to  $\Delta \hat{\mu}_{p_i}^L$  from equation (3.53) but using the vapour properties. The slope of the pseudo-tangent line is obtained for case i) considering the vapour-like feed point and the one liquid found:

$$\frac{\partial \Delta g}{\partial z_i} = \frac{\Delta g^V(z) - \Delta g^L(x')}{z_i - x'_i}$$
(3.80)

and for case iii) from the liquid-like feed and the incipient vapour phase:

$$\frac{\partial \Delta g}{\partial z_i} = \frac{\Delta g^L(z) - \Delta g^V(y')}{z_i - y'_i}$$
(3.81)

For multicomponent systems  $(n \ge 3)$ , because it is considered that the pseudo-tangent plane is orthogonal to the  $\Delta g$  axis at the feed composition, i.e.  $\partial \Delta g / \partial z_i = 0$ , the value of  $h_{p_i}^V$  is obtained as

$$h_{p_i}^V = \frac{\Delta g^V(z)}{RT} - \ln P$$
 (3.82)

and  $h_{p_i}^L$  is calculated from equation (3.59).

With  $h_{p_i}^V$  or  $h_{p_i}^L$  evaluated, the additional search can be performed by using either  $\exp(h_{p_i}^V)$  or  $\exp(h_{p_i}^L)$  in the modified liquid search method described above.

Once the algorithms to search for the vapour and liquid incipient phases have been presented, some relevant aspects of them can be addressed. The procedures followed to obtain the composition estimates required to initiate the search are treated in Section 3.3. With respect to the solution of the stationarity condition (equations (3.35), (3.60) and (3.61)), if desired direct substitution can be used throughout the calculation, i.e. the use of Wegstein's method is optional. When Wegstein's method is employed, the best results are obtained if the acceleration is started in the fourth iteration (if convergence has not been reached) allowing four direct iteration steps, which is also what Swank and Mullins [30] advise. To prevent overshooting when Wegstein's formula is applied, the value of the step size parameter  $t_i$  is limited. The typical value of  $t_{max}$  recommended in the literature is about 10, but as will be seen in the next chapter, a smaller value gave better results for the cases tested. The necessity to limit this parameter arises from the different conditions that develop depending on its value, namely:

value of $t_i$	value of $s_i$	condition
$t_i > 1$	$0 < s_i < 1$	acceleration (extrapolation)
$t_i = 1$	$s_i = 0$	direct substitution
$0 < t_i < 1$	$s_i < 0$	damping (interpolation)
$t_i = 0$	$s_i = \pm \infty$	stagnation (no progress)
$t_i < 0$	$s_i > 1$	deceleration (extrapolation)
$t_i = \pm \infty$	$s_i = 1$	no solution or $\infty$ solutions

Another important consideration in the use of Wegstein's method is its frequency of application. If used sporadically, the convergence rate may be similar to that of direct substitution and the advantage of the method is lost. On the other hand, if used every iteration it can lead to undesirable oscillatory behaviour. In the search method proposed, the frequency of application for acceleration can be pre-specified allowing its use every iteration or every 2nd, 3rd,..., etc. iteration. However, to avoid oscillations in the calculation and even divergence in some cases, whenever the value of the objective function increases after applying Wegstein's method its frequency of use is reduced. In such cases it is employed only every  $\eta * w$  iterations (where  $\eta$  is a control variable which is increased by one when the objective function increases after accelerating and w is the selected frequency of application) until the objective function decreases after accelerating, and only then it is set back to its original frequency w.

To avoid convergence to the trivial solution, Michelsen proposes to stop the search when the value of r is close to 1 and  $g^*$  is small  $(|r - 1| < 0.2 \text{ and } g^* < 10^{-3})$  or when r is greater than 0.8 and increases in consecutive iterations  $(r > 0.8 \text{ and } r^k > r^{k-1})$ . From the observations made for the systems tested, especially for the liquid phase search, the first criterion  $(r \sim 1 \text{ and } g^* \text{ small})$  was found adequate but not the second since in several cases it lead to a wrong prediction in which a trivial solution was indicated when an incipient phase corresponding to the correct solution was being approached. To overcome this weakness, the second criterion was substituted by checks (b) through (d) in the phase search (Step 7) for both the vapour and the liquid phases. Because in some cases none of these criteria were satisfied when a trivial solution was approached, check (a) was introduced. Check (e) was established to speed up the detection of an unstable solution in the odd cases found (in particular for the liquid phase) where convergence near the solution was very slow. These checks and the rest of the suggested modifications made to Michelsen's scheme to obtain more reliable results in the search are discussed in detail in Chapter 4.

## 3.3 Initialization procedures

In order to initiate the vapour and liquid search methods described before, estimates of the composition of the incipient phases are required. For the vapour phase an approach similar to that proposed by Michelsen is used, but for the liquid phase(s) various alternatives have been studied in this work. Since one of the objectives of the phase-splitting method proposed here is to provide estimates for the equilibrium compositions, the methods to estimate the composition of the incipient phases (not necessarily at equilibrium) must be simple (to not cause a significant additional load to the overall algorithm) and dependable (to provide estimates in the vicinity of the stationary points).

## 3.3.1 Vapour phase

The simplest way to initialize vapour compositions is to assume that Raoult's law applies in which case the equilibrium constants can be obtained as

$$K_i^\circ = y_i / x_i = P_i^{sat} / P \tag{3.83}$$

where superscript '°' indicates ideal conditions. For hydrocarbon systems an approximation for the saturation pressure  $P_i^{sat}$  is given by (reference [5])

$$P_i^{sat} = P_{c_i} \exp[5.3727(1+\omega_i)(1-T_{c_i}/T)]$$
(3.84)

where variables with subscript c are the critical constants and  $\omega$  is Pitzer's acentric factor. This approximation was satisfactory for the cases tested and there was no need to utilize more general and/or accurate vapour pressure correlations. Once estimates for the equilibrium ratios are available, the vapour composition can be obtained from

$$y_{i\ init}^{\diamond} = \frac{z_i K_i^{\diamond}}{\sum_{i=1}^n z_i K_i^{\diamond}} \tag{3.85}$$

Although the assumption of ideal behaviour ceases to be valid as conditions deviate from those in which the vapour can be treated as an ideal gas and the liquid as an ideal solution, the results obtained for the systems studied indicate that the estimates obtained by means of equation (3.83) are adequate. Even for highly non-ideal conditions the estimates obtained with equation (3.85) were not far from the stationary points in the cases tested and lead to convergence of the vapour phase search method in a few iterations.

# 3.3.2 Liquid phases

Since the liquid search is for one or two phases, multiple initial estimates must be made (at least two different estimates are needed). One obvious choice is to consider ideal solution behaviour and use the values of  $K_i^{\circ}$  already obtained from equation (3.83). The composition estimates for an incipient liquid are obtained from

$$x_{i\ init}^{\diamond} = \frac{z_i/K_i^{\diamond}}{\sum_{i=1}^n (z_i/K_i^{\diamond})}$$
(3.86)

However, these values may not be very accurate and to assure that an incipient liquid is not missed other initial estimates are needed.

In the review by Swank and Mullins [30] of LL phase-splitting algorithms, three methods for generating initial guesses for immiscible phases are tested for reliability and efficiency: that of Gautam and Seider [19], Shah [1] and Michelsen [4], which have also been discussed in Chapter 2. The first two methods are based on selecting two components as phase initiators (for systems of 3 or more components) and then assigning compositions for each phase according to the equilibrium and mass balance equations. Thus two initial composition vectors are obtained in this form. The method of Michelsen uses as many initial guesses as there are components, starting each phase with the pure components. The results obtained in [30] indicate that the method of Gautam and Seider and that of Shah are less reliable than Michelsen's when the same final convergence algorithm is used. Nevertheless, the three methods are implemented in this work with some modifications to make them more efficient and a fourth method is proposed based also on the selection of phase initiators.

#### Modified Michelsen's method

According to Michelsen, using as many initial composition vectors as the number of components in the mixture and starting with the pure trial phases has the advantage that liquid immiscibility in highly non-ideal systems is promptly detected and component fugacity coefficients are evaluated cheaply. When dealing with hydrocarbon systems, he considers only the lightest and the heaviest components in the mixture plus two additional estimates: one corresponding to a hypothetical ideal gas and another considering the arithmetic mean of the phase compositions already present. For VL equilibria involving hydrocarbons, two sets of compositions are recommended by him according to equations (2.24) and (2.25). As mentioned in Section 2.2.2, starting with the pure phases as composition estimates has the disadvantage that for supercritical components the system's temperature and/or pressure must be altered temporarily to a point where liquid properties can be evaluated for such components. In addition, the advantage of the easy evaluation of pure component fugacities is small since after the initialization properties must be obtained for each component in the mixture. For hydrocarbon systems, the selection of the pure phases based only on the lightest and heaviest components is not justified since their relative amounts in the mixture are not taken into account. Also, for the initialization of liquid trial phases the ideal gas estimate is unnecessary.

The modified initialization scheme proposed consists of using as many composition estimates as components in the mixture but instead of taking each phase as a pure component, for each phase a different component is considered predominant and the rest are evenly distributed among that phase. It was found that assigning a composition of 0.98 in mole fraction for the most abundant component and 0.02/(n-1) for the remaining ones gave good results. Since the composition estimates obtained with equation (3.86) are easily evaluated, this composition vector is also used giving a total of m = n + 1initial estimates which are:

$$x_{j,i \ init} = \begin{cases} 0.98 & \text{if } j = i \\ & i = 1, \dots, n \\ 0.02/(n-1) & \text{if } j \neq i \\ & j = 1, \dots, m \\ x_{i \ init}^{\circ} & \text{if } j = n+1 \end{cases}$$
(3.87)

These initial estimates have the advantage that by considering the starting phases as a mixture (rather than a pure component), extrapolated values of density from the equation of state can be used to evaluate fugacity coefficients if at the specified conditions a liquid does not exist. Thus the system's temperature and pressure need not be altered in the initialization step of the search method. Also, by considering the phases to be almost the pure components the advantage of overestimating the phase-split to quickly detect liquid immiscibility is kept.

#### Shah's method

To estimate the composition of immiscible liquid phases by selecting components p and q as phase initiators of phases I and II, the following scheme is implemented using Shah's method [1] seen in Section 2.2.1. The component with the largest pseudo-infinite dilution activity  $\hat{a}_{p_i}^{L^{\infty}}$  is designated phase initiator p. The activity is calculated as

$$\hat{a}_{p_i}^{L^{\infty}} = z_i \hat{\phi}_i^{L^{\infty}} P \tag{3.88}$$

where  $\hat{\phi}_i^{L^{\infty}}$  is the fugacity coefficient for component *i* at infinite dilution. This fugacity coefficient is obtained considering that component *i* is present in infinitesimal quantity  $(10^{-6} \text{ in mole fraction})$  and that the compositions of the rest of the components are

proportional to their compositions in the feed. Then the second phase initiator q is selected as that component with the largest pseudo-infinite dilution activity in pure component p according to equation (3.88) but with  $\hat{\phi}_i^{L^{\infty}}$  calculated considering only a binary mixture of almost pure p with each component i in turn ( $i \neq p$ ) present in very small amount (mole fraction of  $10^{-6}$ ).

Once components p and q have been obtained, their composition can be calculated from the mass balance and equilibrium conditions by solving equations analogous to (2.3) and (2.4), but in terms of fugacity coefficients:

$$\hat{\phi}_{p}^{L_{II}^{\infty}} n_{p}^{II} / (n_{p}^{II} + n_{q}^{II}) = \phi_{p}^{L} n_{p}^{I} / (n_{p}^{I} + n_{q}^{I})$$
(3.89)

$$\hat{\phi}_{q}^{L_{I}^{\infty}} n_{q}^{I} / (n_{p}^{I} + n_{q}^{I}) = \phi_{q}^{L} n_{q}^{II} / (n_{p}^{II} + n_{q}^{II})$$
(3.90)

where it has been assumed that the fugacity coefficients of components p and q in phases I and II are those of the pure components ( $\phi_p^L$  and  $\phi_q^L$ ) and in phases II and I are those at infinite dilution in the binary mixture ( $\hat{\phi}_p^{L_{T}^{\infty}}$  and  $\hat{\phi}_q^{L_{T}^{\infty}}$ ). Considering 1 mole of overall mixture, the mass balance restrictions dictate that

$$n_{p}^{I} + n_{p}^{II} = z_{p} \tag{3.91}$$

$$n_q^I + n_q^{II} = z_q \tag{3.92}$$

which are similar relations to equations (2.5) and (2.6).

The solution of equations (3.89) to (3.92) reduces to solving a quadratic equation in any of the mole numbers  $n_p^I$ ,  $n_q^I$ ,  $n_p^{II}$  or  $n_q^{II}$ . Taking  $n_q^I$  as the independent variable, the value of  $n_p^I$  can be obtained by solving

.

$$a(n_p^I)^2 + bn_p^I + c = 0 ag{3.93}$$

and choosing the root that satisfies the condition  $0 < n_p^I < z_p$ , where

$$a = ABD - ACD - A^2E + A^2F \tag{3.94}$$

$$b = B^2 D - BCD - ABE - ACE + 2ACF$$
(3.95)

$$c = C^2 F - BCE \tag{3.96}$$

with

$$A = \phi_p^L - \hat{\phi}_p^{L^{\Omega}_I} \tag{3.97}$$

-

$$B = z_p (\hat{\phi}_p^{L^{\infty}} - 2\phi_p^L) \tag{3.98}$$

$$C = z_p \hat{\phi}_p^{L_{II}^{\infty}} \tag{3.99}$$

$$D = \phi_q^L - \hat{\phi}_q^{L_I^\infty} \tag{3.100}$$

$$E = 2z_p \hat{\phi}_q^{L_T^{\infty}} - z_q \phi_q^L \tag{3.101}$$

$$F = -z_q \phi_q^L \tag{3.102}$$

Then  $n_q^I$  is obtained as

$$n_{q}^{I} = -\frac{A(n_{p}^{I})^{2} + Bn_{p}^{I}}{An_{p}^{I} + C}$$
(3.103)

which must satisfy the condition  $0 < n_q^I < z_q$ . If both solutions to equation (3.93) meet the mass balance condition, the one selected is that which makes the condition for  $n_q^I$ hold. If both  $n_p^I$  roots do, it is irrelevant which one is selected. The values of  $n_p^{II}$  and  $n_q^{II}$ are then obtained as  $n_p^{II} = z_p - n_p^I$  and  $n_q^{II} = z_q - n_q^I$ .

Once components p and q are chosen and their mole numbers in both phases are known, the remaining r components  $(r = 1, ..., n - 2, r \neq p, r \neq q)$  are distributed among phases I and II one at the time considering that p and q are the only other components present. The corresponding mole numbers  $n_r^I$  and  $n_r^{II}$  are obtained from the mass balance and equilibrium relations giving:

$$\hat{\phi}_{r}^{L_{I}^{\infty}} n_{r}^{I} / (n_{r}^{I} + n_{p}^{I} + n_{q}^{I}) = \hat{\phi}_{r}^{L_{II}^{\infty}} n_{r}^{II} / (n_{r}^{II} + n_{p}^{II} + n_{q}^{II})$$
(3.104)

which is equivalent to equation (2.7). Taking into account that for 1 mole of the original mixture

$$n_r^I + n_r^{II} = z_r (3.105)$$

equation (3.104) can also be reduced to solving a quadratic for  $n_r^I$ :

$$d(n_r^I)^2 + en_r^I + f = 0 (3.106)$$

where

$$d = \hat{\phi}_{\boldsymbol{r}}^{L^{\infty}} - \hat{\phi}_{\boldsymbol{r}}^{L^{\infty}} \tag{3.107}$$

$$e = (z_r - n_p^I - n_q^I)\hat{\phi}_r^{L_{II}^{\infty}} - (z_r + n_p^{II} + n_q^{II})\hat{\phi}_r^{L_{I}^{\infty}}$$
(3.108)

$$f = z_r (n_p^I + n_q^I) \hat{\phi}_r^{L_{II}^{\infty}} \tag{3.109}$$

The value of  $n_r^I$  should satisfy the condition  $0 < n_r^I < z_r$ ; then  $n_r^{II}$  is obtained as  $n_r^{II} = z_r - n_r^I$ . In case any of the mole numbers calculated is outside the range determined by the mass balance conditions, its value is set close to either the upper limit  $z_i$  or the lower limit 0, whichever is exceeded (i.e.  $n_i = 0.9z_i$  if  $n_i \ge z_i$  or  $n_i = 0.1z_i$  if  $n_i \le 0$ ). With the mole numbers for all components in phases I and II available, the normalized initial compositions (mole fraction) are calculated as

$$x_{1,p} = \frac{n_p^I}{n_p^I + n_q^I + \sum_{r \neq p,q}^{n-2} n_r^I}$$
(3.110)

$$x_{2,p} = \frac{n_p^{II}}{n_p^{II} + n_q^{II} + \sum_{r \neq p,q}^{n-2} n_r^{II}}$$
(3.111)

and similarly for  $x_{1,q}$ ,  $x_{2,q}$ ,  $x_{1,r}$  and  $x_{2,r}$ . This gives 2 composition vectors  $x_{1,i}$  and  $x_{2,i}$ 

(i = 1, ..., n) as initial estimates for the liquid phase, plus the one given by equation (3.86), yields a total of 3 initial composition estimates:

$$x_{j,i \ init} = \begin{cases} x_{1,i} = x_{1,p}, \ x_{1,q}, \ x_{1,r} \\ r = 1, \dots, n-2, \ r \neq p, q \\ x_{2,i} = x_{2,p}, \ x_{2,q}, \ x_{2,r} \\ i = 1, \dots, n \\ x_{3,i} = x_{i \ init}^{\circ} \end{cases}$$
(3.112)

#### Modified Gautam and Seider method

The method of Gautam and Seider (Section 2.2.2) to generate initial compositions for LL phase-splitting introduced the idea of selecting two key components as phase initiators. Because the calculation of the mole numbers of the components present in phases Iand II is rather cumbersome in their method, a simplified approach is suggested. What makes this part of the method inefficient is that it requires solving two simultaneous non-linear equations (equations (2.9) and (2.10)) plus n - 2 single non-linear equations (corresponding to equation (2.13)) in which the activity coefficients  $\gamma_i$  are strongly dependent on composition. The solution of these equations would increase the computational load considerably <sup>1</sup> and would also require that initial estimates be supplied.

In the proposed modification to Gautam and Seider's method component p is chosen as that with the highest activity in the mixture (considered as a liquid). In this way the selection of this component is straightforward since the activities of each component in

<sup>&</sup>lt;sup>1</sup>For example, solving equations (2.9) and (2.10) for a binary system is equivalent to obtaining the actual equilibrium compositions, whereas what in wanted are estimates of the incipient phases to be used as initial compositions for the search method.

the mixture have already been evaluated at the beginning of the phase-splitting algorithm (from equation 3.32). Then component q is selected as the component with the highest activity with component p in a binary liquid consisting of only these two components, considering their composition proportional to that of the mixture. With components p and q available, to simplify the solution of equations (2.9) and (2.10) (but in terms of fugacity coefficients) in order to obtain their compositions, the assumption taken in Shah's method is also made. That is, the fugacity coefficients of p in phase I and of qin phase II are calculated considering them as almost the pure components (since each phase initiator is assumed the most abundant component in the phase where it acts as 'solvent') and their fugacity coefficients in phases II and I respectively are obtained as the infinite-dilution values (by assuming that components p and q are present in small quantities in phases II and I). Hence, the calculation of the mole numbers for p and q simply requires the solution of equations (3.89) and (3.90) as described above. Then the compositions for the remaining components (if  $n \ge 3$ ) in the immiscible phases are also obtained as in Shah's method with 3 composition vectors being generated (as in equation (3.112)). Clearly, if components p and q assigned in this form correspond to those obtained with Shah's method, the same initial composition vectors will be obtained.

# **Proposed method**

Another approach proposed in this work is based on Shah's method and it only differs from it in the way component p is selected. Since the activities of all components in the mixture are available and because the activity was seen to give a relative measure of the tendency for a component to form a new phase, following Gautam and Seider component p is chosen as that with the highest activity in the original liquid-like system. Then component q is selected as explained for Shah's method (i.e. as the component with the largest pseudo-infinite dilution activity in almost pure p), which is then followed throughout to obtain the 3 initial composition vectors. In this form the advantages of the two previous methods seen are combined giving an efficient and dependable way to obtain composition estimates for the phase search in the splitting algorithm.

## 3.4 Phase removal tests

At the beginning of this chapter it was mentioned that when Michelsen's method was incorporated into the phase-splitting algorithm proposed, two main flaws were found. The first, which concerns the bypass of an incipient phase corresponding to a phase present at equilibrium, has been dealt with and to overcome it an additional phase search is recommended. The second fault involves finding an incipient phase which does not correspond to the equilibrium solution. The elimination of such phase is not attempted in the modified search method since it was found that a superfluous phase can also be obtained when the additional phase search is conducted.

The principle behind the phase removal tests used in this work to eliminate a superfluous phase and thus reduce a potential three- phase system to a VL or LL solution at equilibrium can be understood by looking at the source of the problem. The conditions that a phase equilibrium solution must satisfy are that the system should have the lowest possible Gibbs energy (at the specified T, P and composition), the chemical potentials (or equivalently the fugacities) for each component must be equal in all phases and the mass balance must be preserved. The first two conditions are taken care of in the search method since they are implied in the tangent plane criterion used. However, because the phases searched with the modified Michelsen method originate from a single phase system as incipient phases, their amount is considered infinitesimal and the mass balance restriction may not hold. Hence, phases for which the tangent plane criterion is satisfied but which do not correspond to the most stable solution may be found and if so must be eliminated to correctly predict the configuration of the system at equilibrium.

For binary systems the presence of a superfluous incipient phase can be illustrated in a diagram of the Gibbs energy of mixing as a function of composition such as Figure 3.11. This diagram depicts the case of a vapour-like system at the feed composition zwhere two incipient liquid phases are found in the search. Only the vapour and one of the liquids (liquid I) pertain to the equilibrium solution (shown by the  $VL_I$  dashed line). The superfluous incipient liquid II should be ruled out even if a lower Gibbs energy seems to be obtained with the LL solution  $(L_I L_{II} \text{ dashed line})$  since at equilibrium the mass balance restriction would not be satisfied for such solution. Graphically, the mass balance restriction is satisfied if the feed composition is in between the compositions of the phases at equilibrium as dictated by the lever rule  $^2$ . This condition is not met for the  $L_I L_{II}$  solution shown in Figure 3.11, but is satisfied for the  $VL_I$  solution, i.e. the equilibrium compositions y and  $x^{I}$  lie at both sides of the feed z. Thus for systems with two components a test based on the lever rule could be used to eliminate superfluous incipient phases combined with the conditions of equality of fugacities and a minimum in Gibbs energy at equilibrium (this is necessary for example in the case of Figure 3.11 in which a  $VL_{II}$  solution -not shown- would satisfy the mass balance restriction and equality of fugacities but would not lead to a minimum in Gibbs energy for the system).

Because for multicomponent systems there is no analog to the lever rule in the context used here, the detection of a superfluous phase can not be done by knowing the composition of the incipient phases. Tests must then be made as the equilibrium compositions are approached by monitoring the violation of the mass balance restriction due

<sup>&</sup>lt;sup>2</sup>For binary systems the lever rule can be applied to establish the ratio of the amount of one phase to that of the feed. For example, the vapour fraction in a VL system can be obtained as  $V/F = (z_i - x_i)/(y_i - x_i)$ .



Figure 3.11: Binary system stable as VL for which two incipient liquid phases are found in the phase search, one being superfluous.

to the presence of a superfluous phase. The methods described next provide such tests and have been incorporated into the phase-splitting algorithm to eliminate a superfluous phase whenever it is detected in a potential VLL system. They are based on using a three-phase flash calculation scheme to converge the compositions of the phases found towards the equilibrium values by solving for the mole and phase fractions that satisfy the mass balance and equilibrium equations. Since an iterative procedure is required to accomplish this, a limited number of iterations are carried out and during the calculation the tests for the fulfillment of the material balance restrictions are applied to the updated compositions to detect any extraneous phase. Depending on the outcome of these tests it is decided if at equilibrium the system is a three-phase mixture or if a two-phase solution is more stable.

#### 3.4.1 Multiphase flash method

Starting with the compositions of the phases found in the search, a maximum of five iterations are conducted using a three-phase flash calculation scheme based on the method described in Appendix A (but without the acceleration option). With the new-found compositions at each iteration, the phase removal tests proposed in [5] are performed to detect any superfluous phase. If in five iterations no phase elimination has been done, it is considered that the mixture is stable as a VLL system and the current composition values are used as initial estimates for the eventual flash calculation. If any of the phases is found to be extraneous, it is eliminated and the composition of the phases found in the phase search for the corresponding two-phase flash (not the current composition values) are used as initial estimates if the flash calculation is to be conducted. The steps of this phase removal method are:

## Phase removal test. Multiphase flash method

1. With the initial composition vectors obtained from the phase search  $(y = y', x^{I} = x', x^{II} = z; y = y', x^{I} = x'^{I}, x^{II} = x'^{II}$  or  $y = z, x^{I} = x'^{I}, x^{II} = x'^{II}$ ) the corresponding fugacity coefficients are obtained from the equation of state and the equilibrium ratios, defined as  $K_{i_{V}} = y_{i}/x_{i}^{II}$  and  $K_{i_{L}} = x_{i}^{I}/x_{i}^{II}$ , are calculated according to

$$K_{i_V} = \hat{\phi}_i^{L_{II}} / \hat{\phi}_i^V \tag{3.113}$$

$$K_{i_L} = \hat{\phi}_i^{L_{II}} / \hat{\phi}_i^{L_I} \tag{3.114}$$

- 2. Start of the iterative cycle. If the maximum number of iterations is reached (set to five) the calculation terminates, the system is considered stable as a threephase mixture and the current composition values are used as initial estimates for the VLL flash if needed. Otherwise, when all the  $K_{i_L}$  values are near unity, then both liquid phases are very similar and the one for which the sum of the activities is greater is eliminated (since it is the phase with higher tendency to disappear). In such case the system is considered stable as VL and the calculation ends. The initial compositions for the flash are the compositions corresponding to the phases found in the previous search (for the vapour and the liquid not eliminated).
- The phase ratios α and β, defined as α = V/F and β = L<sub>I</sub>/F (note that 1 α β = L<sub>II</sub>/F) are solved for from the Rachford-Rice type arrangement of the mass balance and equilibrium equations using a Newton-Raphson procedure

$$\Gamma_V(\alpha,\beta) = \sum_{i=1}^n \frac{z_i(1-K_{i_V})}{1+\alpha(K_{i_V}-1)+\beta(K_{i_L}-1)}$$
(3.115)

$$\Gamma_L(\alpha,\beta) = \sum_{i=1}^n \frac{z_i(1-K_{i_L})}{1+\alpha(K_{i_V}-1)+\beta(K_{i_L}-1)}$$
(3.116)

where V,  $L_I$ ,  $L_{II}$  and F are the total mole numbers of the vapour, liquid I, liquid II and the feed.

4. Once  $\alpha$  and  $\beta$  are obtained, they are used as phase indicators to find any phase which should be eliminated in the calculation. The following checks are done and the corresponding control variables  $\nu_V$ ,  $\nu_{L_I}$  and  $\nu_{L_{II}}$  (all being zero initially) are set accordingly:

(a) If 
$$\alpha < 0$$
 and  $\beta < 0$  or  $\alpha < 0$  and  $\beta > 1$  then  $\nu_V = \nu_V + 1$  and  $\nu_{L_I} = \nu_{L_I} - 1$ .

- (b) If  $\alpha < 0$  and  $1 \alpha \beta < 0$  or  $\alpha < 0$  and  $1 \alpha \beta > 1$  then  $\nu_V = \nu_V + 1$  and  $\nu_{L_{II}} = \nu_{L_{II}} - 1.$
- (c) If  $\alpha > 1$  and  $\beta < 0$  and  $1 \alpha \beta < 0$  then if for liquid I the sum of the activities is higher than for liquid II  $\nu_{L_{II}} = \nu_{L_{II}} 1$ , otherwise  $\nu_{L_I} = \nu_{L_I} 1$  when the sum of the activities for liquid II is greater than for liquid I.
- 5. When outside the interval [0,1] the calculated values of  $\alpha$  and  $\beta$  are limited (as indicated in Appendix A) to avoid the calculation of negative mole fractions from equation (3.117). After these parameters have been limited, additional checks are done:
  - (a) If  $\alpha$  is still outside the interval [0, 1], then  $\nu_V = \nu_V + 1$ , otherwise  $\nu_V = 0$ .
  - (b) If  $\beta$  is not in the interval [0, 1] then  $\nu_{L_I} = \nu_{L_I} + 1$ , otherwise  $\nu_{L_I} = 0$ .
  - (c) If  $1 \alpha \beta$  is not in the interval [0, 1] then  $\nu_{L_{II}} = \nu_{L_{II}} + 1$ , otherwise  $\nu_{L_{II}} = 0$ .
- 6. The elimination of any of the phases is analyzed depending on the value of the control variables:

- If  $\alpha < 0$  and  $\nu_V \ge 3$  then the vapour phase is eliminated and the system is considered stable as LL.

- If  $\beta < 0$  and  $\nu_{L_I} \ge 3$  liquid I is eliminated; if  $1 - \alpha - \beta < 0$  and  $\nu_{L_{II}} \ge 3$  then liquid II is eliminated. In either case the system is declared stable as VL.

When any of the phases is eliminated the calculation terminates and the appropriate solution is the result of the phase-splitting prediction. The composition of the corresponding phases found in the search are used to initialize the two-phase flash if required.

7. The new compositions of the three phases are calculated according to

$$x_{i}^{*II} = \frac{z_{i}}{1 + \alpha(K_{i_{V}} - 1) + \beta(K_{i_{L}} - 1)}$$
(3.117)

$$x_{i}^{\star I} = x^{\star II} K_{iL} \tag{3.118}$$

$$y_{i}^{\star} = x^{\star II} K_{iv} \tag{3.119}$$

which are normalized so that the sum of each of the mole fractions equals unity:

$$x_i^{II} = \frac{x_i^{*II}}{\sum_{i=1}^n x_i^{*II}}$$
(3.120)

$$x_i^I = \frac{x_i^{\star I}}{\sum_{i=1}^n x_i^{\star I}}$$
(3.121)

$$y_{i} = \frac{y_{i}^{\star}}{\sum_{i=1}^{n} y_{i}^{\star}} \tag{3.122}$$

The fugacity coefficients are calculated from the equation of state with the updated composition values and the equilibrium ratios are obtained from equations (3.113) and (3.114). A new iteration is started by executing Step 2.

# 3.4.2 Bubble- and dew-point equations method

Following the method proposed by Nelson [3] (reviewed in Section 2.2.3) a scheme based on analyzing the behaviour of the mass balance and equilibrium equations at the limit where only two or one phase exist is adopted. In this way tests to determine the number of phases at equilibrium can be performed based on the so- called bubble- and dew-point equations derived. Instead of the phase and equilibrium ratios defined in [3], parameters  $\alpha = V/F$ ,  $\beta = L_I/F$ ,  $K_{i_V} = y_i/x_i^{II}$  and  $K_{i_L} = y_i/x_i^{II}$  as defined in the previous section are used for consistency in the relationships developed. This leads to slightly different but equivalent criteria for the phase removal tests based on the following equations (in terms of these parameters)

$$Q_1(\psi_1,\psi_2) = \sum_{i=1}^n \frac{z_i(K_{i_L} - K_{i_V})}{1 + \alpha(K_{i_V} - 1) + \beta(K_{i_L} - 1)}$$
(3.123)

$$Q_2(\psi_1, \psi_2) = \sum_{i=1}^n \frac{z_i(1 - K_{i_V})}{1 + \alpha(K_{i_V} - 1) + \beta(K_{i_L} - 1)}$$
(3.124)

versus equations (2.27) and (2.28) where the phase and equilibrium ratios used are  $\psi_1 = L_1/F$ ,  $\psi_2 = L_2/F$ ,  $K_{i1} = y_i/x_i^I$  and  $K_{i2} = y_i/x_i^{II}$ . Also, in contrast to Nelson's method in which the phase removal tests are executed throughout using a three-phase calculation (even if fewer than three phases have been already detected) until convergence is achieved, in the proposed method the configuration of the system is determined when any of the tests is satisfied for three consecutive iterations and at that point the calculation ends. This alternative method, which is also incorporated into a flash-like calculation, consists of the following steps:

#### Phase removal test. Bubble- and dew-point method

1. With the initial compositions obtained from the phase search the equilibrium ratios are obtained according to equations (3.113) and (3.114).

- 2. Start of the iterative cycle. The phase ratios  $\alpha$  and  $\beta$  are calculated from equations (3.115) and (3.116) holding  $K_{i_V}$  and  $K_{i_L}$  fixed using a Newton-Raphson technique.
- 3. If  $\alpha$  and/or  $\beta$  are outside the interval [0, 1] they are limited to avoid the calculation of negative mole fractions (as described in Appendix A). Then the new compositions of the three phases are calculated as indicated by equations (3.117) to (3.122).
- 4. With the updated compositions the fugacity coefficients are evaluated from the equation of state and new equilibrium ratios are calculated.
- 5. Test for the existence of a single phase. The following checks are made:
  - (a) If  $\sum_{i=1}^{n} z_i/K_{i_V} < 1$  and  $\sum_{i=1}^{n} z_iK_{i_L}/K_{i_V} < 1$  then the control variable  $\nu_V$  is set to  $\nu_V + 1$ , otherwise  $\nu_V = 0$ . If  $\nu_V \ge 3$  the system is considered stable as a vapour and the calculation is terminated.
  - (b) If  $\sum_{i=1}^{n} z_i/K_{i_L} < 1$  and  $\sum_{i=1}^{n} z_iK_{i_V}/K_{i_L} < 1$  or  $\sum_{i=1}^{n} z_iK_{i_V} < 1$  and  $\sum_{i=1}^{n} z_iK_{i_L} < 1$  then  $\nu_L = \nu_L + 1$ , otherwise  $\nu_L$  is set to 0. If  $\nu_L \ge 3$  the system is considered stable as a liquid and the calculation ends.
- 6. Test for the existence of two phases. To determine if the system is stable as a two-phase mixture the criteria used are:
  - (a) If  $\sum_{i=1}^{n} z_i K_{i_V} / K_{i_L} > 1$  and  $\sum_{i=1}^{n} z_i K_{i_L} / K_{i_V} > 1$  and  $Q_2(\psi_1, 0) < 0$  at the root of  $Q_1(\psi_1, 0) = 0$  then  $\nu_{VL_I} = \nu_{VL_I} + 1$  and control variable  $\nu_{VLL}$  is set to 0, otherwise  $\nu_{VL_I} = 0$ , where

$$Q_1(\psi_1, 0) = \sum_{i=1}^n \frac{z_i (K_{i_L} - K_{i_V})}{K_{i_V} + \beta (K_{i_L} - K_{i_V})}$$
(3.125)

$$Q_2(\psi_1, 0) = \sum_{i=1}^n \frac{z_i(1 - K_{i_V})}{K_{i_V} + \beta(K_{i_L} - K_{i_V})}$$
(3.126)

The system is considered stable as a VL mixture  $(VL_I)$  if  $\nu_{VL_I} \ge 3$  and the calculation ends at this point.

(b) If  $\sum_{i=1}^{n} z_i K_{i_V} > 1$  and  $\sum_{i=1}^{n} z_i / K_{i_V} > 1$  and  $Q_1(0, \psi_2) < 0$  at the root of  $Q_2(0, \psi_2) = 0$  then  $\nu_{VL_{II}} = \nu_{VL_{II}} + 1$  and  $\nu_{VLL} = 0$ , otherwise  $\nu_{VL_{II}} = 0$ , where

$$Q_1(0,\psi_2) = \sum_{i=1}^n \frac{z_i(K_{i_L} - K_{i_V})}{1 - \alpha(1 - K_{i_V})}$$
(3.127)

$$Q_2(0,\psi_2) = \sum_{i=1}^n \frac{z_i(1-K_{i_V})}{1-\alpha(1-K_{i_V})}$$
(3.128)

The system is considered stable as a VL mixture  $(VL_{II})$  if  $\nu_{VL_{II}} \ge 3$  and if so the calculation terminates.

(c) If  $\sum_{i=1}^{n} z_i K_{i_L} > 1$  and  $\sum_{i=1}^{n} z_i / K_{i_L} > 1$  and  $Q_2(\psi_1, 1 - \psi_1) > 0$  at the root of  $Q_1(\psi_1, 1 - \psi_1) - Q_2(\psi_1, 1 - \psi_1) = 0$  then  $\nu_{LL} = \nu_{LL} + 1$  and  $\nu_{VLL} = 0$ , otherwise  $\nu_{LL} = 0$ , where

$$Q_1(\psi_1, 1 - \psi_1) - Q_2(\psi_1, 1 - \psi_1) = \sum_{i=1}^n \frac{z_i(1 - K_{i_L})}{\beta(1 - K_{i_L}) - 1}$$
(3.129)

$$Q_2(\psi_1, 1 - \psi_1) = \sum_{i=1}^n \frac{z_i(1 - K_{i_V})}{1 - \beta(1 - K_{i_L})}$$
(3.130)

The system is considered stable as a LL mixture  $(L_I L_{II})$  if  $\nu_{LL} \ge 3$  and in such case the calculation ends.

7. If none of the tests described in Steps 5 and 6 are satisfied then  $\nu_{VLL} = \nu_{VLL} + 1$ . The system is considered stable as a VLL mixture if  $\nu_{VLL} \ge 3$  in which case the calculation ends. Otherwise a new iteration is started by going back to Step 2.

As with the multiphase flash reduction tests, whenever a two-phase solution is obtained the initial estimates for the flash are the compositions of the corresponding phases found in the search and not the current composition values which are used only if a VLL solution is predicted.

## 3.4.3 Coupled method

As shown in the next chapter, better results were obtained in the phase removal tests when the two methods just described were used in a coupled fashion. This approach is suggested because when the additional phase search was conducted, the multiphase flash method failed in some specific cases in which the bubble- and dew-point equations method was successful and vice-versa. Specifically, the multiphase flash method is used in every case except when the option for the additional phase search is selected and the system is liquid-like, a vapour is found and no liquids are detected (in the initial search) until the additional phase search is performed, in which case the bubble- and dew-point method is used as phase removal test. This strategy can be summarized as follows:

# Phase removal test. Coupled method

1. If the additional phase search is conducted and cases i) or ii) of the search (described in Section 3.2) occur, then the multiphase flash method is used. It is also used when

.

case iii) arises only if an additional liquid is not found in the search.

.

2. If in the additional phase search a liquid is found when case iii) develops, then the bubble and dew-point method is used as phase removal test.

## Chapter 4

## **Results and discussion**

The evaluation of the performance of any phase-splitting algorithm is not a simple task since a number of criteria can be considered for this purpose. For example, its efficiency can be matched against that of other methods by determining the computational load required to perform a calculation, but unless the comparison is based on the same grounds the results can be misleading. Therefore the same or equivalent conditions should be used if any comparison is attempted (such as initialization parameters, convergence tolerance and objective function, correctness of the prediction and even CPU time evaluation at the same time of day). The same applies when comparing for reliability since in cases where a given method may fail another may perform flawlessly or vice versa.

To assess the phase-splitting method proposed, the ultimate purpose sought in achieving the objectives of this work must be taken into account. This is to reliably and efficiently determine the equilibrium configuration of a system in order to avoid the time- consuming three-phase flash calculation (or the more commonly used two-phase flash) when unnecessary. For this reason the equilibrium conditions predicted with the algorithm presented here are compared with the results obtained with the more crude three-phase flash calculation, which itself can be reduced to a VL or LL flash when a twoor single-phase solution prevails at equilibrium. In this way if with the phase-splitting method a system is found stable as a single phase then the flash calculation is avoided; if the presence of two phases is recognized then only the corresponding two-phase flash is necessary, and if three phases are found then and only then a three-phase flash is executed. When a flash is required after the phase-splitting prediction is conducted, the initial estimates generated in the search (as described in Chapter 3) are used; however, for the direct three-phase flash (i.e. when not using the phase-splitting algorithm) the initial estimates as proposed in [5] are employed. This initialization scheme and the flash algorithms are also described in Appendix A. (Note: all the calculations presented here where done on an Amdhal 5860 and the CPU times given are in microseconds.)

## 4.1 Systems tested

The performance of the different parts of the algorithm proposed is assessed by testing a total of seven multicomponent systems at conditions where one-, two- or three-phase equilibria exists according to Soave-Redlich-Kwong (SRK) or Peng-Robinson (PR) thermodynamics. These systems have been taken from the literature reviewed except for the last one which corresponds to the actual operating conditions for a compressor gas mixture. It should be understood at this point that most of these systems are representative examples of different types of equilibria which can be described by the aforementioned equations of state. Thus, for many of these systems experimental data to compare the results obtained are not available and it is the qualitative description of the equilibria behaviour rather than the quantitative predictions what is of interest. This means that the ability of the equations of state to reproduce experimental data is beyond the scope of this work and that the results obtained are as accurate (quantitatively) as the thermodynamic models used. When available, interaction parameters for each binary pair in the systems tested  $(k_{i,j}$  values) are used as reported from the source (unless otherwise specified) and if not given, they are taken from Reid et al. [32] for the SRK equation. These systems and the temperature, pressure and composition conditions under which they were studied as well as the binary interaction parameters used are:

System 1: methane-carbon dioxide-hydrogen sulphide

 $z_1 = 0.5, z_2 = 0.1, z_3 = 0.4$  $T=120 \rightarrow 280 \text{ K}$ P=20 atm *T*=171 & 200 K P=20 atm66 feed points z for each T $k_{i,j}$ comp. no. component  $k_{1,2} = 0.12$ CH₄ 1  $CO_2$  $k_{1,3} = 0.08$ 2  $H_2S$   $k_{2,3} = 0.12$ 3

Note that  $k_{i,j} = k_{j,i}$  and that  $k_{i,i} = k_{j,j} = 0.0$ 

This ternary mixture, which is capable of splitting into three phases, is taken from Michelsen [4] and has a very narrow temperature range in which three phases coexist at the specified pressure for the composition given (the upper three-phase critical point is at about 218 K and 62 atm). The SRK equation of state is used to describe this system and since binary interaction parameters are not reported in [4], the values given in [32] are used.

System 2: methane-ethane-propane-n-butane-n-pentane-nitrogen

 $T=90 \rightarrow 320 \text{ K}$  P=40 atm

comp. no.	component	$z_i$	$k_{i,6}$
1	$\mathrm{CH}_4$	0.5479	0.02
2	$C_2H_6$	0.0708	0.06
3	$C_3H_8$	0.0367	0.08
4	$C_4H_{10}$	0.0208	0.08
5	$\mathrm{C_5H_{12}}$	0.0198	0.08
6	$N_2$	0.3040	0.0
All other 
$$k_{i,j} = 0$$

Michelsen [33] uses this nitrogen-rich light-hydrocarbon system as an example of a mixture which exhibits unexpectedly complicated behaviour since according to SRK thermodynamics it has two separate three-phase regions and a total of six critical points. At 40 atm the upper three-phase region is encountered around 151 K and as in System 1 the temperature range for which three-phase equilibria is found is of less than two degrees ( $k_{i,j}$  values from Reid et al. [32], not reported by Michelsen [33]).

System 3: propane-n-butane-n-pentane-n-hexane-n-octane- -water

T=430 K  $P=5 \rightarrow 100 \text{ atm}$ 

comp. no.	component	$z_i$	$k_{i,6}$
1	$C_3H_8$	0.16667	0.48
2	$nC_4H_{10}$	0.16667	0.48
3	$nC_5H_{12}$	0.20000	0.48
4	$nC_6H_{14}$	0.06667	0.48
5	$nC_8H_{18}$	0.13333	0.48
6	$H_2O$	0.26667	0.0

This hypothetical system is used by Peng and Robinson in [34] to make VLL calculations where the capacity of the model to describe all three phases at equilibrium with the equation of state proposed by them is discussed. The  $k_{i,j}$  values reported in their work are used here with the PR equation to describe this water-hydrocarbon mixture; in addition a value of 0.1277 is used in this study as Mathias' [9] polar parameter for water. In this case the calculations are done at constant temperature and various pressures. Because the two liquid phases found are highly immiscible the three phase region is very broad (at 430 K the three-phase region extends from about 25 to 35 atm).

System 4: n-hexane-water

T=378 K P=5 atm 21 feed points z comp. no. component  $k_{1,2}$ 1 nC<sub>6</sub>H<sub>14</sub> 0.48 2 H<sub>2</sub>O

Heidemann [29] uses this mixture as one of the systems to exemplify the prediction of VLL equilibria with the Wilson modification of the Redlich-Kwong equation of state. With the binary interaction parameter set to 0.5 a semi-quantitative prediction of the solubility of water in the hydrocarbon rich phase is possible. In the present work the SRK equation is used considering a value of  $k_{1,2} = 0.48$  to correspond to the value used for System 3. For this binary the calculations are done considering the feed in the entire composition range, i.e. from 0 to 1 in n-hexane mole fraction.

System 5: hydrogen sulphide-methane

T=190 K P=38 atm 21 feed points z comp. no. component  $k_{1,2}$ 1 H<sub>2</sub>S 0.08 2 CH<sub>4</sub> This binary system is considered by Michelsen in [4] because at the specified temperature and pressure the methane-rich phase is nearly supercritical, which makes this liquid phase difficult to detect. The SRK equation of state is used with the binary interaction parameter taken from Reid et al. [32]. The composition range tested includes the interval from 0 to 1 in hydrogen sulphide mole fraction.

System 6: methane-n-butane-water

<i>T</i> =311 K	$P{=}60 \text{ atm}$	66 feed points $z$		
	comp. no.	component	$k_{i,3}$	
	1	$\mathrm{CH}_4$	0.50	
	2	$nC_4H_{10}$	0.48	
	3	H <sub>2</sub> O	0.0	

This mixture is studied by Heidemann in [29] using also the Wilson modification of the Redlich-Kwong equation of state which predicts three-phase equilibria at the given temperature and pressure. The results obtained by him are claimed to be in good agreement with the experimental data of McKetta and Katz [35] especially to match the water mole fraction in both the hydrocarbon-rich liquid and in the vapour phase at 311 K. In the present work the SRK equation of state is used with interaction parameters set to 0.0 for the methane-n-butane pair, to 0.48 for n-butane-water and to 0.5 for methane-water, this last value as used in [29].

System 7: hydrogen-carbon dioxide-hydrogen sulphide- methane-ethane-ethylene-propanepropylene-n-butane-isobutane--1-butene-n-pentane-isopentane

T=283 & 327 K P=5 & 15 atm

comp. no.	component	$z_i$	$k_{1,j}$	$k_{2,j}$	$k_{3,j}$
1	$H_2$	0.1561	0.0	0.0	0.0
2	$CO_2$	0.0038	0.0	0.0	0.1020
3	$H_2S$	0.0151	0.0	0.1020	0.0
4	$C_1$	0.0608	-0.0205	0.0936	0.0823
5	$C_2$	0.0341	-0.0009	0.1340	0.0852
6	$C_2^=$	0.0176	0.0325	0.0575	$0.07^{\ddagger}$
7	$C_3$	0.1177	0.1077	0.1280	0.0798
8	$C_3^=$	0.3169	0.1344	0.0681	$0.07^{\ddagger}$
9	$nC_4$	0.0259	0.3628	0.1376	0.0588
10	iC4	0.0862	$0.37^{\S}$	$0.15^\ddagger$	0.0595
11	$1 - C_4^=$	0.1570	$0.4^{\S}$	0.0629	0.06 <sup>§</sup>
12	$nC_5$	0.0059	$0.5^{\$}$	0.1407	0.0681
13	iC5	0.0029	$0.5^{-5}$	0.1341	0.06 <sup>‡</sup>

All other  $k_{i,j} = 0$ ; also  $k_{i,j} = k_{j,i}$  and  $k_{i,i} = k_{j,j} = 0$ 

 $k_{i,j}$  values taken from Elliot and Daubert [36]; <sup>‡</sup> from Reid et al. [32];

 $\S$  estimated values from the homologous series for which values were available

This mixture corresponds to a wet gas compression operation for which it was desired to determine if condensation would occur in the second stage of compression (283 K and 5 atm at the suction and 327 K and 15 atm at the discharge). The mole fractions of the stream correspond to conditions where failure in the compressor is reported presumably due to liquid accumulation when there is a high flow rate of feed gas during cold weather. The SRK equation was used to predict the equilibrium properties with updated  $k_{i,j}$  values available from the literature.

### 4.2 Phase search initialization

The start of the phase search methods described in Chapter 3 requires initial composition estimates of the trial phases to be sought. A successful outcome in the phase-splitting prediction depends among other factors on this first step, particularly in the search for liquid phases as will be seen. The importance of this part of the algorithm becomes clear when it is recognized that Michelsen's method and also the modification of it proposed here do not guarantee convergence to a global minimum in Gibbs energy. Estimates in the vicinity of the stable solution are necessary to detect a solution corresponding to a global minimum rather than to just a local one.

## 4.2.1 Vapour phase

Under the conditions at which the different systems were tested it was found that the generation of composition estimates for the vapour search  $(y_{i\,init}^{\circ})$  according to equation (3.85), in which Raoult's law is assumed, gave satisfactory results. Even at high pressures where the ideal gas laws cease to be valid the values of  $y_{i\,init}^{\circ}$  followed the correct trend in mole number distribution and temperature dependence observed for the incipient vapour composition (also for the actual equilibrium values) and allowed the incipient phase to be found in a few iterations. This can be seen in the examples presented in Table 4.1 in which representative results are given for three systems (5, 1 and 2) at conditions where they are liquid- like and a vapour phase is found giving a correct phase-split prediction in every case. Setting the convergence tolerance  $\varepsilon$  to  $10^{-5}$  for the objective function (equation (3.46)) in all the points tested, 5 iterations were required at the most to find the incipient vapour except for System 2 at 169.9 K and 60.5 atm (not shown in Table 4.1) for which 6 iterations were required.

			Vapou	r phase sea	urch initi	alization	
	Systen	1 5: H <sub>2</sub> S	5(1)– CH	$I_4(2)$ $T$	=190 K	P=38  atm	L
feed zi	$Raoult y_1^\circ$	t's law y <sup>°</sup> 2	incipie: $y'_1$	$\begin{array}{c} nt \ vapour \\ y_2' \end{array}$	equilibr y <sub>1</sub>	ium comp. Y2	no. of iterations to find incip. vapour
$0.1 \\ 0.3 \\ 0.5 \\ 0.7$	.0007 .0028 .0066 .0152	.9993 .9972 .9934 .9848	.0197 .0221 .0169 .0136	.9804 .9779 .9831 .9864	.0178 .0178 .0178 .0178	.9822 .9822 .9822 .9822 .9822	3 3 3 3
	Systen	n 1: CH	4(1)- C0	$D_2(2)-H_2S$	$(3)  z_1 =$	$= 0.5  z_2 =$	0.1 P = 20  atm
T(K)	$egin{array}{c} Raoul \ y_1^\diamond \end{array}$	t's law y <sub>2</sub> °	$incipie y'_1$	$nt \ vapour \\ y_2'$	equilibr Yı	ium comp. Y2	no. of iterations to find incip. vapour
180 200	.9916 .9850	.0047 .0079	.9794 .9559	.0125 .0241	.9617 .8952	.0291 .0707	2 2
System 2: hydrocarbons(1-5)– $N_2(6)$ P=40 atm							
T(K)	$egin{array}{c} Raoul \ y_1^\diamond \end{array}$	t's law $y_2^\diamond$	$incipie y'_1$	$\begin{array}{c} nt \ vapour \\ y_2' \end{array}$	equilibr Yı	rium comp. Y2	no. of iterations to find incip. vapour
160 180	.2102 .2503	.7894 .7489	.2890 .3564	.7083 .6374	.3875 .5581	.6087 .4288	5 4

Table 4.1: Initial composition estimates for the vapour phase search using Raoult's law and for the flash calculation using the incipient vapour found.

Although the proposed method to estimate the incipient vapour composition could be inappropriate under more drastic conditions such as near critical points or in retrograde behaviour regions, calculations to see if the search would fail due to poor estimates were not done under such conditions. For the cases studied it was found that very accurate composition estimates were not crucial for the correct prediction of a vapour phase (probably because the fugacity coefficients were not highly composition dependent) and since the simple initialization procedure used worked flawlessly and efficiently no alternative methods were tried.

## 4.2.2 Liquid phases

The initialization for the liquid phase search constitutes a more severe problem than the initialization for the vapour not only because at least two distinct phases are sought but because the composition dependence of the fugacity coefficients is much stronger for condensed phases. This means that good initial estimates must be provided to assure a correct and efficient prediction of the incipient liquids. It has been recognized by several authors (Michelsen [4], Maurer and Prausnitz [16], Walraven and Van Rompay [20] among others) that in order to reliably predict LL equilibria the initial composition estimates of the liquid phases should overestimate the immiscibility region (i.e. lie outside of it). Although this condition is a desirable feature for the initialization of the flash calculations, it is not imperative and certainly an excessive overestimation in which the guess compositions are far from the equilibrium values is no better than values inside the two-phase region when close to the binodal curve. This was also observed for the initialization of the incipient phases in the search methods. These concepts can be exemplified with the mixture n-hexane(1)- water(2) (System 4) for which Table 4.2 gives the composition estimates of the incipient phases  $(x_p^I, x_q^{II} \text{ and } x_1^{\circ})$ , the composition of the incipient phases found  $(x_1^{II}, x_1^{II})$  and the actual equilibrium values  $(x_1^{II}, x_1^{II})$  at two

Liquid phase search initialization								
System 4: $nC_6H_{14}(1)-H_2O(2)$ T=378 K P=5 atm								
	mod Mich	lified elsen	mod. Seider	Gautam & r (p=2,q=1)	Sh ( $p=1$ )	aah	propo (p	product set set method = 2,q=1)
zi	0.90	0.95	0.90	0.95	0.90	0.95	0.90	0.95
$x_p^I$	.9800	.9800	1.000	.3425*	.9890	.9948	1.000	.3425*
$x_q^{II}$	.9800	.9800	.9988	.9994	1.000	1.000	.9988	.9994
$x_1^\diamond$	.8087	.8992	.8087	.8992	.8087	.8992	.8087	.8992
$x_1^{\prime I}$	<.0001	<.0001	<.0001	no L found +	<.0001	<.0001	<.0001	no L found +
$x_1^{\prime II}$	.9000	.9500	.9000	.9500	.9000	.9500	.9000	.9500
$x_1^I$	<.0001	<.0001	<.0001	wrong VL	<.0001	<.0001	<.0001	wrong VL
$x_1^{II}$	.9894	.9894	.9894	prediction	.9894	.9894	.9894	prediction
* Compositions for the water-rich incipient phase $I$ that underestimate the water content. + As a consequence phase $I$ can not be found leading to an incorrect phase-splitting prediction.								

Table 4.2: Composition estimates for the incipient liquid phases obtained with the various initialization methods tried, composition of the incipient phases found and equilibrium values for two feed compositions in System 4.



Figure 4.12: Gibbs energy of mixing diagram for the n-hexane-water system at 378 K and 5 atm (System 4) showing that both components are immiscible almost throughout the entire composition range.

feed compositions of a n-hexane rich mixture.

At both  $z_1 = 0.90$  and  $z_1 = 0.95$  only one incipient phase was found in the search (phase I, rich in water) and thus the other incipient phase (n-hexane rich phase II) is taken as the feed in order to initialize the flash calculation. Because of this, the initial compositions for the flash slightly underestimated the amount of n-hexane in phase II but this did not cause an erroneous outcome in the calculation. However, the composition estimates for the phase search obtained with the modified Gautam and Seider method and with the proposed one at  $z_1 = 0.95$  (marked with an asterisk in Table 4.2) clearly underestimated the water composition (phase initiator p = 2) in phase I preventing the prediction of a water-rich incipient phase (marked with a cross). This lead to the incorrect prediction of a VL system at this composition. It was found that to avoid this problem the initial water composition in phase I should be greater than 0.75 in mole fraction ( $x_p^I > 0.75$ ,

p = 2); the reason for this can be seen in Figure 4.12, which shows the Gibbs energy of mixing as a function of composition. To reach the minimum in the  $\Delta g/RT$  curve for the liquid corresponding to the water-rich phase the initial composition must be to the left of the maximum located approximately at a composition of 0.25 in n-hexane (or 0.75 in water), otherwise the phase search algorithm goes in the 'downhill' direction towards the other minimum (corresponding to the n-hexane rich phase to the right of the maximum).

As mentioned in Section 3.3.2, Swank and Mullins [30] conclude that Gautam and Seider's and Shah's methods are less reliable than Michelsen's as initial algorithms for calculating the equilibrium compositions to predict LL phase-splitting. The evaluation of these methods as implemented in the present work to initialize the liquid search, as well as the alternative scheme proposed (by combining Gautam and Seider's with Shah's criteria to select the phase initiators), indicate that Shah's is perhaps the most reliable method followed closely by the one proposed. Then the modified Gautam and Seider and Michelsen methods are next in that order. These conclusions are based on the frequency with which any of the initialization algorithms lead to an incorrect prediction of the incipient phases due to faulty initial composition values. In Table 4.3 the performance of these methods is shown for some of the systems tested under the same conditions ( $\varepsilon = 10^{-5}$  and using Wegstein's acceleration method); the table gives the number of iterations needed to find the incipient liquids, the CPU time required to generate the initial compositions and to reach convergence in the phase search (in milliseconds), and the number of times in which an erroneous solution was obtained. Considering then the results shown in this table and in Table 4.2, where the proposed method as well as the modified Gautam and Seider's failed once, it can be concluded that only a slight difference in dependability was found among the four methods.

With regard to the efficiency of these methods (assessed by the CPU time required for the initialization plus the phase search), that requiring the least computational effort

Liquid phase search initialization								
system no.	no. of data points	no. of iters./( modified Michelsen	CPU msecs. initizn modified Gautam & Seider	. + search)/wm Shah	ong prediction proposed method			
1	11	65/(99)/0	63/(96)/0	54/(107)/0	54/(86)/0			
2	16	99/(444)/2	121/(394)/0	121/(480)/0	121/(381)/0			
3	11	37/(260)/0	39/(184)/0	39/(239)/0	39/(177)/0			
7	3	6/(438)/0	6/(137)/0	6/(259)/0	6/(137)/0			
Total	41	207/(1241)/2	229/(811)/0	220/(1085)/0	220/(781)/0			

Table 4.3: Comparison of the reliability and efficiency of the four methods tested to provide initial compositions for the liquid phase search.

is the proposed method as can be observed from the results presented in Table 4.3. When the number of components in the mixture is large (about 6 or more), the modified Michelsen method is the most time consuming option. This becomes more patent as the number of components increases since initially as many composition vectors are used in the phase search as there are components in the system, which eventually are reduced to two or one. For mixtures with a reduced number of components this and Shah's methods were of similar efficiency. Thus, when reliability is of primary importance at the sacrifice of speed Shah's method should be used, otherwise as a short-cut method the scheme proposed becomes advantageous.

To give an idea of the proximity of the initial composition estimates to the incipient phase compositions and of these to the equilibrium values, the percent root mean square deviation (% RMS dev.) from the incipient liquid compositions are given in Table 4.4 for System 1 at conditions where two liquid phases were found <sup>1</sup>. At 120 and 140 K the modified Michelsen method gave the closest estimates to the actual incipient phases but at the higher temperature (160 K) Shah's and the proposed method gave better results, whereas the modified Gautam and Seider method gave poor estimates in all cases. Since the compositions given by the modified Michelsen method are fixed, they are not dependent on the composition or temperature of the feed in contrast to the other three methods. For all methods except the modified Michelsen, better estimates for the incipient phases were obtained as the temperature increased; also the incipient phases became better estimates of the equilibrium compositions (i.e., as T increased the composition of the incipient phases approached the equilibrium values.)

Contrary to what could be expected, it was found that not always (although in most cases) having better composition estimates lead to convergence in less number

<sup>&</sup>lt;sup>1</sup>The composition vector obtained from Raoult's law,  $x_{i\,init}^{\diamond}$ , is not considered in the calculation of the % RMS deviation.

Liquid phase search initialization											
	System 1: $CH_4(1)-CO_2(2)-H_2S(3)$ $P=20$ atm										
T(K)	$L_I$	modi Miche L <sub>II</sub>	fied elsen no. its.	mo Seid L <sub>I</sub>	d. Gau ler (p L <sub>II</sub>	utam & p=1,q=2) no. its.	$She met L_I$	nh & hod (p L <sub>II</sub>	proposed p=1,q=3) no. its.	equil comp L <sub>I</sub>	ibrium position L <sub>II</sub>
120	0.8	1.3	5	24.0	20.0	7	4.6	7.6	5	9.4	4.8
140	2.8	3.7	5	20.1	15.2	8	3.9	5.9	6	8.0	4.4
160	5.9	6.7	13	16.8	10.8	8	3.3	4.8	6	6.7	3.9
	% RMS dev. = $[\sum_{i=1}^{n} (comp{i} - incip. comp{i})^{2}/n]^{0.5} \times 100$										

Table 4.4: % RMS deviation of the incipient phase estimates and of the equilibrium values from the actual incipient phases composition (mole fraction) in the prediction of LL immiscibility for System 1.

of iterations as was the case at 160 K in Table 4.4. A larger number of iterations were required to reach convergence when using the modified Michelsen method than when the modified Gautam and Seider scheme was employed, in spite of the former method giving estimates much closer to the solution.

With respect to the initial liquid composition vector obtained with Raoult's law, it was found that in some cases it was a good approximation for one of the incipient liquids when two were found or for the only liquid when just one phase was detected. In other cases it could be considered only fair. Since for the methods in which phase initiators are selected one of the vector estimates usually was similar or at least resembled the composition vector obtained with the ideal solution assumption, it was considered at one point that this vector was unnecessary. With the idea to increase the efficiency of the search algorithm by reducing the number of initial vectors, the elimination of vector  $x^{\circ}_{i\,init}$ was tested. This was also done when using the modified Michelsen initialization scheme to reduce the number of initial composition vectors from n+1 to n. The results obtained showed that indeed the search was faster when reducing by one the number of initial vectors (since the more initial vectors the more calculations are required). However, for System 2 failure in the liquid search occurred by removing the ideal behaviour estimates when using the modified Gautam and Seider, Shah's and the proposed methods which otherwise gave correct results for all the cases tested. For Systems 1 and 3 no failures were detected because of the removal of this vector from any of the initialization methods. From these results it was decided to include the composition estimates  $x_{i,init}^{\diamond}$  as one of the initial vectors for the sake of reliability even if an additional computational effort was required. For example, of 13 points tested with System 2 the CPU time saved by not considering this vector was of about 19 % when initializing with the proposed algorithm, but at the expense of 3 incorrect predictions; for Systems 1 and 3 a reduction of 12 % and 11 % was achieved when testing 11 points for each system.

The generation of adequate composition estimates for the methods in which phase initiators are selected for each liquid phase (modified Gautam and Seider, Shah's and proposed methods) depends to a great extent on the correct allocation of such components. For binary systems the selection is almost trivial <sup>2</sup> but not so when 3 or more components are present. From the results obtained it was observed that in the majority of the cases the same components were selected when using Shah's and the proposed method, and that when the modified Gautam and Seider method lead to convergence in a larger number of iterations, different components than those obtained with aforementioned methods had been selected. For example, for System 1 (see Table 4.3) more iterations are required in the search when using the modified Gautam and Seider method which designates as phase initiators p and q methane and carbon dioxide respectively, than with Shah's or the proposed method which select methane and hydrogen sulphide.

#### 4.3 Phase search performance

As seen in the previous section, the performance of the search method for condensed phases depends to some extent on the initialization routine employed. In addition, there are other factors that can be determinant in the outcome of the detection of these phases such as the value of the parameters selected when Wegstein's method is applied or the criteria used to eliminate unnecessary composition vectors along the search. For the vapour phase, the search is less complex since only one composition vector is used and it is also less sensitive to the initial compositions or the acceleration parameters chosen. For either the vapour or liquid phases the additional search for possibly missed phases can avoid erroneous predictions in some cases. In this section the influence of these factors

<sup>&</sup>lt;sup>2</sup>Although the decision of which components act as phase initiators is immaterial, the assignment of which is phase initiator p and which is phase initiator q can make a difference in the outcome of the results; e.g., the erroneous predictions shown in Table 4.2 are due to the incorrect allocation (swapped) of the two components.

on the search algorithms is discussed.

To illustrate the outcome of the search for a vapour and liquid phases the results for System 2 in the temperature interval from 90 K to 320 K at fixed pressure are given in Table 4.5. In this case Wegstein's method was used every three iterations (starting in the fourth) with a maximum step size  $t_{max}=3$  and setting the convergence tolerance to  $10^{-5}$ . From low to high temperatures, in the range specified the system can exist as a LL mixture, as a single liquid, then again as a biphasic liquid mixture (i.e. there is a lower and upper consolute temperature), as a three-phase system (between 150 and 152 K, not shown in Table 4.5), as a VL mixture and finally as a vapour. This complex behaviour was adequately predicted with the phase-splitting algorithm proposed (without using the additional vapour or liquid phase searches) except for two cases when the modified Michelsen initialization scheme was used. From 90 to 180 K the system is liquid-like and thus the vapour search method was applied (refer to the phase- splitting scheme in Section 3.1) leading to correct predictions except at 150 K in which a superfluous vapour was found (i.e. a vapour is not present in the stable solution, however this phase appears at 150.2 K corresponding to the three-phase bubble point temperature at 40 atm) but was later eliminated with the phase removal tests implemented. Note that only a few iterations were required to achieve convergence in the vapour search and that the computational effort required is minimal when compared to that needed for the search of the liquid phases. From 200 K upward the system is vapour-like and thus only the liquid phase search was conducted. Since the system is liquid-like from 90 to 180 K, in those cases in which none or only one incipient liquid was found an additional incipient liquid is considered present having a composition equal to the feed. Thus, finding only one incipient liquid when the system is stable as a LL or VLL system or none when it is stable as a liquid or as a VL mixture does not constitute an erroneous prediction as long as the system is liquid-like because the feed is considered as an incipient phase. From

	Vapour and liquid phases search								
	System 2: hydrocarbons(1-5)- $N_2(6)$ $P=40$ atm								
	stable	vapour search		liquids s	earch				
(K)	system	Raoult's	modified	mod. Gautam	Shah	proposed			
		law	Michelsen	& Seider		method			
00	тт	$n \in V \mathcal{D}(A)$	OT 16/40)	OT 13(40)	OT 12/15)	OT 12(28)			
90	TT	10 V 2(4)	21110(49) 11 * 22(46)	2113(40)	2113(43)	2113(30) 2124(65)			
110	LL T	10 V 2(4)	11.22(40)	2124(00)	2124(11)	2124(03)			
110	L T	no V $Z(4)$	$\frac{10 \text{ L} 3(22)}{1 \text{ L} 3(22)}$	10 L 4(17) L 2(16)	10 L 4(22)	10 L 4(17)			
120	L T	no V $2(4)$	no L $3(22)$	no L $3(10)$	no L $\mathcal{J}(21)$	no L $3(14)$			
130	L -	no V $3(5)$	no L $3(22)$	no L $3(16)$	no L $3(22)$	no L $3(16)$			
140	$\mathbf{L}$	no V 4(6)	no L 3(22)	no L 3(16)	no L 3(21)	no L 3(16)			
150	$\operatorname{LL}$	V 5(6)	no L <sup>+</sup> 4(25)	$1L\ 25(47)$	$1L \ 25(51)$	$1L \ 25(46)$			
160	VL	V 5(7)	1L 13(36)	$1L \ 13(29)$	1L 13(33)	1L 13(28)			
180	$\mathbf{VL}$	V 4(6)	1L 7(30)	1L 7(23)	1L 7(26)	1L 7(21)			
200	$\mathbf{VL}$	vapour-like	1L 3(24)	1L 3(17)	1L 3 (22)	1L 3(16)			
220	$\mathbf{VL}$	system	1L 3(23)	1L 3(17)	1L 3(23)	1L 3(16)			
240	$\mathbf{VL}$	Ļ	1L 3(23)	1L 3(16)	1L 3(24)	1L 3(17)			
260	VL		1L 4(25)	1L 4(18)	1L 4(24)	1L 4(17)			
280	$\mathbf{VL}$		1L4(27)	1L 5(22)	1L 5(28)	$1L\ 5(21)$			
300	V		no L $\dot{4}(24)$	no L 4(17)	no L $\dot{4}(24)$	no L 4(17)			
320	V		no L 4(24)	no L 4(17)	no L 4(23)	no L 4(16)			

\* Prediction leads to LL solution corresponding to a local minimum, not to the most stable solution. + With the modified Michelsen method a liquid is not found in the interval 148-151 K leading to an incorrect VL prediction.

Table 4.5: Vapour and liquid phases search performance for System 2 from 90 to 320 K indicating the stable solution, the number of iterations required for convergence and the CPU time (in milliseconds) required for the initialization plus the phase search.

110 to 140 K where neither liquid or vapour incipient phases were found the system was correctly declared stable as a single liquid, avoiding the need of a flash calculation. The flash was also avoided at 300 K and above where no liquid was found and the system was stated stable as a vapour. It is also noteworthy to mention that for this system whenever conditions were close to where a new stable phase appeared (i.e. near bubble or dew points), the number of iterations required in the search of the corresponding phase increased noticeably, particularly for the liquid phase.

### 4.3.1 Vapour phase

Initially, the search algorithm for the vapour phase was implemented in similar fashion as described by Michelsen in [4] but using Wegstein's method to accelerate convergence. Then most of the modifications leading to the final form of the algorithm were derived from changes made to the liquid search method (described in the next section). Although in many instances the need for these modifications was not observed for the vapour phase search, they were incorporated into the algorithm for consistency with the analogous liquid search method and as 'safety' features to assure reliability. With respect to the use of Wegstein's method, treated in more detail in Section 4.3.3, it was rarely required for the vapour search since it was applied after the third iteration and in most cases convergence was achieved in three iterations or less. However, for the cases where convergence had not been reached in the fourth iteration, it aided in achieving it at a faster rate than when direct substitution was used.

In practically all the cases tested, the search method proposed worked flawlessly either finding a vapour phase when present at equilibrium or indicating its absence when not. Only at conditions very close to a bubble point (but not quite where a vapour has appeared) a vapour corresponding to a superfluous phase was occasionally detected in the search. This was the case for System 2 at 150 K described above (see Table 4.5) in which a vapour was found just below the three-phase bubble point temperature. Although this could lead to an erroneous prediction, it is taken into account since such a phase is considered only as a potential phase (this is always done when three phases are detected as explained in the phase search strategy) whose presence is examined by the subsequent phase removal tests. For instance, for the example referred, the superfluous vapour was eliminated with the multiphase flash test giving a correct LL prediction. Thus finding a superfluous vapour does not necessarily constitute a major fault.

Of more serious consequences can be the situation in which a vapour that exists at equilibrium is bypassed in the search. Since phases can not be added after the search has been conducted, the omission of the vapour would lead to a spurious prediction. With the additional search proposed in Section 3.2 the possibility for the occurrence of this problem is reduced. For example, when using the original Michelsen method or the proposed one without the additional phase search, an erroneous LL solution was predicted for the hydrogen sulphide(1)-methane(2) system (System 5) in the feed composition range of about 0.15 to 0.25 in hydrogen sulphide mole fraction. For the compositions within this interval a vapour phase was missed in the initial search of the phase-splitting method giving faulty LL predictions. When the additional search was carried out, the vapour phase was successfully detected with the pseudo-tangent plane criterion proposed giving correct VL solutions in all cases. Table 4.6 gives the results obtained in the entire composition range when the configuration of the system was predicted with a VLL flash calculation and with the phase-splitting algorithm with and without the option to search for additional phases (the additional search for a liquid phase was not conducted since conditions for it are not met). When the direct flash calculation was used to predict the equilibrium conditions incorrect results are obtained in two cases.

The Gibbs energy of mixing for System 5 as a function of composition is given in

	System 5: $H_2S(1)-0$	$CH_4(2)$	T = 190  K $P =$	38 atm
<i>z</i> 1	stable solution	VLL flash	n proposed no additional	algorithm additional
0 00	V	V	V	V V
0.00	VI.	VL.	VI.	VI.
).10	VL	VL	VL VL	VL VL
).15	$\mathbf{VL}$	V*	$LL^+$	$\mathbf{VL}$
).20	$\mathbf{VL}$	VL	LL+	$\mathbf{VL}$
).25	$\mathbf{VL}$	$\mathbf{VL}$	$LL^+$	$\mathbf{VL}$
).30	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$
).35	VL	$LL^*$	$\mathbf{VL}$	$\mathbf{VL}$
.40	VL	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$
.45	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$
.50	VL	$\mathbf{VL}$	$\mathbf{VL}$	VL
.55	VL	VL	$\mathbf{VL}$	$\mathbf{VL}$
.60	· VL	$\mathrm{VL}$	VL	$\mathrm{VL}$
.65	$\mathrm{VL}$	$\mathrm{VL}$	$\mathbf{VL}$	${ m VL}$
.70	$\mathrm{VL}$	$\mathrm{VL}$	$\mathbf{VL}$	$\mathrm{VL}$
.75	$\mathrm{VL}$	VL	$\mathrm{VL}$	$\mathbf{VL}$
.80	VL	VL	VL	VL
.85	VL	VL	VL	VL
.90		L	L	L
0.95	L T	L T	L T	
.00	Ŀ	L	L	L
Wre	ong prediction(s)	2	3	0
fotal	CPU time (msecs.)	276	278	280

Table 4.6: Phase equilibria predictions comparison for System 5 when the phase-splitting algorithm is used with the optional additional phase search to find a missed vapour giving correct VL equilibria solutions.



Figure 4.13: Gibbs energy of mixing diagram for the hydrogen sulphide-methane system at 190 K and 38 atm (System 5) for which the vapour phase is detected with the additional phase search leading to the correct prediction of VL equilibria.

Figure 4.13. As can be seen, the system is liquid-like almost throughout the whole composition range except in the hydrogen sulphide dilute region, where it is vapour-like. This section of the Gibbs energy curve is the one that accounts for the presence of the vapour phase in the VL solution. Although it may seem that a LL system is also feasible, the stable two-phase solution corresponds to a vapour rich in methane and a liquid rich in hydrogen sulphide. For example, at the feed compositions  $z_1=0.15$ , 0.20 and 0.25 the Gibbs energies of mixing are respectively 985.25, 897.36 and 809.48 cal/gmole for the LL solution (2.61, 2.38 and 2.14 in dimensionless terms) against 980.42, 892.87 and 805.31 cal/gmole for the stable VL solution (2.60, 2.37 and 2.13 in dimensionless form).

In Table 4.6 are also given the total CPU times (in milliseconds) required for the calculations presented in which the modified Michelsen method was used to initialize the liquid search (for 2 components it was more efficient than the other methods) and parameters  $\varepsilon$ , w and  $t_{max}$  were set to  $10^{-5}$ , 3 and 3 respectively. The direct flash algorithm was faster in this case than the proposed phase-splitting method but the difference is negligible. Even when the additional phase search was conducted, the difference in CPU time was insignificant. Considering the improved reliability obtained when this last option was chosen, the small additional computational effort is of no concern.

When the additional search was not executed, evidence of a vapour phase being bypassed in the search was also observed for Systems 1 and 6. For System 1 ( $CH_4(1)$ - $CO_2(2)-H_2S(3)$ ) in 2 out of the 66 points tested throughout the entire composition range (at 171 K and 20 atm) a vapour was not found when it should have been. At the feed composition  $z_1=0.7$ ,  $z_2=0.1$  and  $z_3=0.2$  a vapour was not detected in the initial search leading the phase-splitting algorithm to a LL equilibrium prediction (with  $\Delta g = 409.03$ cal/gmole) whereas the stable system is a three- phase mixture ( $\Delta g = 408.56 \text{ cal/gmole}$ ). Similarly, for the point  $z_1=0.7$ ,  $z_2=0.3$  and  $z_3=10^{-8}$  a LL solution was predicted when the system is in fact stable as a VL mixture ( $\Delta q = 553.45$  cal/gmole for the LL system versus  $\Delta g = 548.02$  cal/gmole for the VL solution). Again in this case the incorrect prediction was caused by skipping a vapour phase in the search. When selecting the option to carry out the additional search, for the first point  $(z_1=0.7, z_2=0.1, z_3=0.2)$  a vapour phase was still not found and thus the correct solution could not be obtained. For the second point  $(z_1=0.7, z_2=0.3, z_3=10^{-8})$  a vapour phase was found  $(y_1=0.1591, z_2=0.3, z_3=10^{-8})$  $y_2=0.3493, y_3=0.4916$ ) but of composition very different from the vapour found at equilibrium  $(y_1=0.9725, y_2=0.0275, y_3 < 10^{-5})$ , which caused its removal when the phase reduction tests were conducted. For the methane(1)-n-butane(2)-water(3) system (System 6) only in 1 of the 66 points tested at different compositions (T=311 K and P=60atm) a vapour was not found in the initial phase search (at the feed composition  $z_1=0.2$ ,  $z_2=0.1$  and  $z_3=0.7$ ). As a consequence a wrong LL solution was predicted since the system is stable as a VLL mixture (the Gibbs energy of the system decreases in 24.50 cal/gmole with the appearance of the vapour phase). Even when the additional phase search was carried out, the vapour phase was still bypassed. Therefore, in these cases the additional search for a vapour was not successful. Since no other systems were found for which a vapour was missed with the initial search, the additional search strategy implemented could not be tested more extensively.

# 4.3.2 Liquid phases

The search method for liquid phases as implemented in the original form proposed by Michelsen[4] was found to function acceptably in some cases but frequently incorrect predictions were obtained with it. To improve its performance several modifications were devised, mainly with regard to the detection of convergence of any composition vector to the trivial solution (where the incipient phase searched has the same composition as the feed). These modifications were incorporated also for the vapour phase search, unless otherwise specified. The only difference is that if unsuccessful convergence is detected for the vapor the calculation terminates whereas for the liquid the corresponding vector is eliminated and the calculation continues, except if all the other vectors have been eliminated in which case the calculation also ends.

As mentioned in Section 3.2, Michelsen indicates that a trivial solution is found when |r-1| < 0.2 and  $g^* < 10^{-3}$  (parameter r is defined in equation (2.22) and  $g^*$  in (2.20)) or when r > 0.8 and  $r^k > r^{k-1}$ . The first criterion was found adequate (|r-1| < 0.2 and  $g^* < 10^{-3}$ ), but to detect as early as possible the approach to a trivial solution only the first condition seemed necessary. The reason for this is that in many cases a few more iterations were required for the condition  $g^* < 10^{-3}$  to be satisfied, but the outcome of the result was not changed if this condition was eliminated. The second criterion  $(r > 0.8 \text{ and } r^k > r^{k-1})$  was abolished as a test to detect a trivial solution since in many occasions as a correct incipient phase was being approached, the value of r would increase

(r being > 0.8) from one iteration to the other. Thus, instead of the criteria used by Michelsen, in the modified method the value of r is tested to see if it is approaching 1 in consecutive iterations since such pattern indicated an approach to a trivial solution or to a stable system (this gives check (b) in Step 7 of the search methods, Sections 3.2.1 and 3.2.2). It was also found that early termination of the search should be made when stability is detected before convergence is reached, i.e., when an incipient phase can not be formed from the original phase <sup>3</sup>. Based on this fact, a criterion to stop the search when  $\sum_{i=1}^{n} X_i < 1$  and this sum is decreasing in consecutive iterations is established (corresponding to check (c) in the search methods). In the cases where the value of the objective function (equation (3.46) or (3.73)) increased in consecutive iterations, termination of the search for the corresponding vector was needed to avoid divergence in the calculation since in such case an incipient phase was not being approached (this situation is accounted for in check (d)).

These checks worked well in situations where Michelsen's criteria failed. However, still for a few of the points tested, in particular for Systems 1 and 6, monitoring convergence with checks (b) through (d) was not sufficient to detect a trivial solution. Because of this, it was necessary to first test if the actual composition of the incipient phase being searched was approaching the feed composition. This is done with check (a) by testing if  $\sum_{i=1}^{n} |x_i - z_i| < 2n \times 10^{-4}$ , where  $x_i$  is the current composition vector (in mole fraction), in which case a trivial solution is found. Finally, an additional test was required in the modified Michelsen method for the cases where convergence to an unstable solution was slow (especially for the liquid phase). In this instance, when negative values of  $g^*$  (which indicate an unstable system and thus the formation of a new phase) are not changing

<sup>&</sup>lt;sup>3</sup>According to the stability criteria developed by Michelsen, a phase is stable if at a stationary point  $\sum_{i=1}^{n} X_i < 1$  (where  $X_i$  are the mole numbers of the phase being searched). Thus, whenever a stationary point corresponding to a stable phase is approached an incipient phase can not be obtained.

significantly in 3 consecutive iterations <sup>4</sup>, it is considered that a new phase has been found even if convergence has not been reached and the current value of the composition vector is taken as a converged solution. It was observed that in this case if convergence was allowed (taking as many iterations as necessary) the composition values of the converged solution were practically the same as those obtained when the above criterion was used. This test corresponds to check (e) in the search methods for the vapour and liquid phases.

For the search of liquid phases, Michelsen recommends that if no negative values of  $g^*$  are found after 4 iterations the composition vector with the smallest value of rbe converged, eliminating the rest. This criterion was found faulty in cases where a composition vector had large r values since it would be eliminated from the search when such vector corresponded to an unstable trial phase. To avoid this, an analogous criterion was established which consists on eliminating after the fourth iteration all the composition vectors (provided that for all  $g^*$  was > 0) except the one with the largest value of |r - 1|, that is, the one furthest from the trivial solution. This alternative criterion, used only for the liquid phase search, performed adequately in the cases tested (corresponding to check (g) in the liquid search method).

Since during the liquid phase search all the composition vectors are converged simultaneously, it is important to keep track of the vectors that have reached convergence and of those that are converging to a common point. In the first case, once a vector satisfies one of the convergence tests of Step 8 (Section 3.2.2) it is saved along with the corresponding fugacity coefficients and then eliminated from the search. In many cases two or more vectors would be converging to a common point, and eventually became the same. To avoid excess calculations, a criterion was set to eliminate all alike vectors but one. The test for convergence to a common point is done in Step 9 and the elimination takes place in Step 10. In this last step the vectors that have converged to a minimum in

<sup>&</sup>lt;sup>4</sup>When stating that the values of  $g^*$  did not change significantly a tolerance of  $10^{-6}$  was allowed.

Gibbs energy or to the trivial solution and those not likely to converge are also eliminated, namely those not marked in check (g) of Step 7. When all vectors have been eliminated or when the maximum number of iterations allowed is reached, the calculation ends. If no liquid trial phases are found, the system is stable with respect to the formation of a new liquid phase. If one incipient liquid is found, it is arbitrarily designated as liquid Iand if the system is liquid-like the composition of the feed is assigned to liquid II. If two or more liquid phases are found, again a test for convergence to a common vector is done in case the test made in Step 9 failed. This can happen if a vector quickly converges to a solution and another one, initially being distinctively different from the first, eventually converges to the same point. In case three different liquid trial phases are found (which is unlikely but nevertheless may happen), the two with the lowest Gibbs energy of mixing are considered as the incipient phases.

With the proposed modifications the liquid phase search algorithm performed soundly for the systems tested. On the rare occasions where an incorrect prediction was obtained with a particular initialization scheme, a correct result could always be obtained when using an alternative method. In these cases the liquid search method appears to be sensitive to the initial compositions used and failure can be attributed at least partially to the initialization method used. For most of the systems studied, one or two incipient liquids were found when at equilibrium this number of liquid phases were present, except on certain occasions when the system was liquid-like where none or only one incipient liquid was found respectively. For example, for System 2 (Table 4.5) points near the twoand three-phase dew points, as well as the other points tested, were always correctly predicted (except for two cases when initializing with the modified Michelsen method) in spite of finding in some cases one less incipient phase than the number present at equilibrium.

Superfluous liquid phases were found occasionally, mainly near dew point conditions

but in other cases too. For instance, for System 1 at 160 K and 20 atm two incipient liquid phases were found (no incipient vapour detected) giving the correct prediction of a LL stable system. At 180 K the system is stable as a VL mixture but still two incipient liquids were found in the search (as well as an incipient vapour). As for the vapour phase, the presence of superfluous phases is contemplated since the phase removal tests are always applied when three potential phases are found. What must be avoided is the situation in which a phase that exists at equilibrium is bypassed in the search. For this reason the additional search presented in Section 3.2 is suggested after the initial search is conducted. As an example of the usefulness of the additional phase search, in Table 4.7 are given the equilibrium predictions for the binary mixture n-hexane(1)water(2) (System 4) at 378 K and 5 atm throughout the entire composition range. With the VLL flash calculation incorrect predictions were obtained in 4 occasions and in 11 with the proposed algorithm (initializing with the modified Michelsen method) when the additional phase search was not conducted. In contrast, when the additional search was carried out no erroneous predictions were obtained. In all cases it was an additional liquid phase that was found giving a correct LL prediction instead of the wrong VL solution. As can be seen in Figure 4.12, the system is vapour-like in all cases where the liquid was missed in the initial search. In the composition range from 0.10 to 0.55 in n-hexane mole fraction it was the n- hexane rich liquid that was bypassed in the original search and then found with the additional search, whereas for the point corresponding to a feed composition of 0.80 (also in n-hexane mole fraction) it was the water rich phase that was missed and then found. Similarly to the results shown in Table 4.6, the extra CPU time required for the additional search is not very significant particularly when considering the improvement in reliability (parameters  $\varepsilon$ , w and  $t_{max}$  were also set to  $10^{-5}$ , 3 and 3 to generate the results given in Table 4.7).

Contrary to what was observed for the vapour phase, the additional phase search was

S	System 4: nC <sub>6</sub> H <sub>14</sub> (1)	<i>T</i> =378 К <i>Р</i>	=5  atm			
$z_1$	stable	VLL flash	proposed	algorithm		
	solution		no additional	additional		
0.00	т	т	pnase searcn	pnase search		
0.00		L VI *				
0.05	TT TT					
0.10	ц т ц т					
0.15	11 1	LL 1/*		1) 1 1		
0.20	ЦЦ Т Т	V* 1/*				
0.25	ТТ					
0.30	ТТ					
0.35						
0.40	11					
0.45						
0.50	ТТ Ц	1.1.	VL' VI+	ЪЪ Т Т		
0.00	ЦЦ ТТ					
0.00						
0.65	TT I I I I		11 11			
0.70	17		11 11			
0.75	ЦЦ Т Т	ЪЪ Т Т	ЪL VI +			
0.80	ЦЦ ТТ					
0.80	ТТ					
0.90						
0.95				LL T		
1.00	L	<b>با</b> ان ر	L	L		
Wro	ng prediction(s)	4	11	0		
Total (	CPU time (msecs.)	143	152	192		
<ul> <li>* Incorrect prediction.</li> <li>+ Wrong prediction due to the bypass of a liquid phase.</li> </ul>						

Table 4.7: Equilibria predictions when the phase-splitting algorithm is used with and without the optional additional phase search to find a missed liquid yielding correct LL equilibria solutions for System 4.

very effective for System 6 in finding an incipient liquid. Out of the 66 points tested at various compositions (with starting values obtained with the proposed initialization method), in 6 of them incorrect VL equilibria predictions were obtained because a nbutane rich liquid trial phase was missed in the initial search (these points are centered in a well defined region around the feed point  $z_1=0.6$ ,  $z_2=0.2$ ,  $z_3=0.2$ ). However, when using the additional search, in all 6 cases a new incipient liquid (rich in n-butane) was detected giving correct VLL predictions. The additional liquid search strategy was also tested for System 2 at 150 K in which an incorrect VL system was predicted when using the modified Michelsen method to obtain initial liquid composition estimates. In this case, although a new incipient phase was found with the additional search, the correct LL solution was not obtained but a spurious single-liquid prediction. The reason for this is that the incipient phase found has a composition very far from the equilibrium value (an n-pentane rich liquid was found instead of a liquid with composition similar to the feed) which causes the consequent LL flash to predict the presence of only one liquid phase. A liquid phase was bypassed by the initial search only for the systems described above, and therefore the results obtained when the additional search was used are discussed just for these mixtures.

In short, as for the vapour phase, the additional liquid search strategy worked very well for binary systems, increasing markedly the reliability of the equilibrium predictions. For mixtures with 3 or more components, in which a simplified criterion derived from the pseudo-tangent plane criterion used for binary systems was implemented, the proposed additional search aided in finding bypassed liquid phases in many (but not all) cases. However, no successful results were obtained in finding a vapour phase when missed in the initial search. In an attempt to better understand the tangent plane criterion in order to improve the additional phase search strategy for multicomponent systems (n > 3), the Gibbs energy of mixing surfaces for the two three-component systems studied (Systems 1 and 3) were plotted as a function of composition considering the systems as vapour, as liquid and then as the stable system (this last one constructed from the surface portions having a lower Gibbs energy from either of the vapour or liquid surfaces). Although the 3-D plots obtained helped in visualizing these surfaces, the information gained from them was limited in spite of having the Gibbs energy contour lines that indicate the presence of points such as local or global minima.

Such surfaces are shown for the methane(1)-n-butane(2)-water(3) system (System 6) at 311 K and 60 atm in Figures 4.14, 4.15 and 4.16. In the first figure the Gibbs energy surface is obtained by considering that the system is a vapour in the entire composition range and in Figure 4.15 by considering it as a liquid throughout. To do so, even at conditions where neither one of the phases can exist as a single phase, extrapolated values of density are used as obtained with the equation of state in order to calculate the necessary thermodynamic properties (fugacity coefficients). Figure 4.16 represents the Gibbs energy that the stable system would have, which is obtained by superimposing the surfaces for the vapour-like and the liquid-like systems and eliminating the highest values of Gibbs energy at each composition, i.e., only the portions corresponding to the lower Gibbs energy are shown. Because the Gibbs energy values for the vapour-like system are higher than for the liquid-like system at almost every composition, the combined surface (for the stable system) looks very similar to that for the liquid-like system. Only in the methane-rich portion of the surface (farthest corner from the observers view-point) does the vapour have a lower Gibbs energy than the liquid and thus for the stable system the vapour curve prevails in this region although this is barely noticeable in the surface represented in Figure 4.16.



Figure 4.14: Gibbs energy of mixing as a function of composition for the methane-n-butane-water system (System 6) at 311 K and 60 atm considered as a vapour.



Figure 4.15: Gibbs energy of mixing surface for System 6 considered as a liquid in the entire composition range at 311 K and 60 atm.



Figure 4.16: Gibbs energy of mixing for the stable system (System 6 at 311 K and 60 atm) obtained by superimposing the vapour and liquid-like surfaces and retaining the least Gibbs energy values at every composition.

For System 1 similar 3-D diagrams were obtained. In them the surface corresponding to the vapour-like system was also above that obtained when considering the system as a liquid, except for the methane-rich portion of the diagram which indicates that a vapour composed mainly of methane is present as a stable phase for some feed compositions. The Gibbs energy of mixing surface as a function of composition for the stable system is given in Figure 4.17. By inspecting this diagram it is difficult to foresee that the system can split in two or three phases at equilibrium, depending on the feed composition. Looking in more detail at the contour lines given in Figure 4.18, there is still no clear indication that phase splitting will occur. However, when performing the equilibrium calculations (either with a direct flash or with the phase- splitting algorithm proposed), a significant decrease in Gibbs energy was obtained when a phase-split occurred. For example, at the feed composition of  $z_1=0.1$ ,  $z_2=0.8$ ,  $z_3=0.1$  the Gibbs energy of mixing for the system as a vapour is 1425.19 cal/gmole, as a liquid is 923.97 cal/gmole and as the stable VL solution 725.79 cal/gmole; at  $z_1=0.3$ ,  $z_2=0.6$ ,  $z_3=0.1$  the values of  $\Delta g$  for the system as a single vapour, as a liquid, as a VL mixture and as the stable VLL solution are 1491.25, 1210.02, 1207.51 and 1013.81 cal/gmole respectively. In Figure 4.19 the stability of the feed is given as a function of composition indicating the type and number of phases in which the system will separate in order to reach a stable state. Figure 4.20 shows the composition of each of the phases found at equilibrium. As can be seen, for the twophase solutions the composition of the vapour and of liquid I remain in well defined areas whereas that of liquid II changes considerably depending on the feed point (when a VL solution is obtained it is liquid II that is found). When the system separates in three phases, the compositions of each phase remain fixed.





Figure 4.19: Type and number of phases present at equilibrium in the composition plane for System 1 at 171 K and 20 atm according to the location of the feed.



Figure 4.20: Composition of the phases formed at equilibrium when System 1 splits as two- or three-phase stable mixtures at different feed locations.

The fact that the vapour-like Gibbs energy surface was above that for the liquid-like system at almost every composition is most certainly the reason why the additional vapour phase search failed for the multicomponent systems tested. As mentioned in Section 3.2, the pseudo-tangent plane criterion used (for mixtures with 3 or more components) considers a pseudo-tangent plane orthogonal to the Gibbs energy axis (parallel to the composition plane for ternary systems). Thus, whenever the vapour Gibbs energy of mixing surface lies above this pseudo-plane, an incipient vapour will not be found. Also, a missed liquid could be more easily found with the additional search since the pseudotangent plane is more likely to intersect the Gibbs energy surface for the liquid-like system. This can be anticipated to happen for any system for which the vapour surface lies mainly above that for the system considered as a liquid. In the opposite case, when the surface for the liquid lies chiefly above that for the vapour (which did not occur for the systems studied), it could be expected that the additional search for a missed vapour would be more successful, with a corresponding decreased likelihood of locating any bypassed liquid phase.

### 4.3.3 Wegstein's method

To solve for the mole numbers (independent variables) from the stationarity condition written in the form of equations (3.35), (3.60) and (3.61), direct substitution can be used but Wegstein's acceleration method was successfully applied to increase the convergence rate in the search algorithms. As mentioned in Section 3.2, the method is started in the fourth iteration if convergence has not been obtained after four direct substitution steps. Although only two direct substitution steps are required to initiate the method, two more are conducted to provide better starting points and to assure proper convergence behaviour. The two important convergence parameters that can be specified by the user when opting for this method are the frequency of application w and the maximum step
size  $t_{max}$ . The outcome of the search depends to some extent on the values assigned to these parameters, more so for the liquid than for the vapour phase.

In order to select the optimum values of w and  $t_{max}$  which minimize the number of iterations required to achieve convergence, different combinations of these variables were tested for Systems 1, 2 and 3. The values of w tried were such that the method was applied on every iteration, or every second or third iterations (i.e., w=1, 2 or 3, respectively). When the values of w were higher than 3, the method performed very similarly to direct substitution and its advantage over it was lost. The results obtained are summarized in Table 4.8, which gives the number of iterations required to obtain convergence in the vapour and liquid phase searches (in this last case using starting compositions given by the proposed initialization method) setting the convergence tolerance to  $10^{-5}$ . For each of the systems tested the temperature and pressure conditions are given as well as the number of data points tested, indicated in parenthesis. Also the results obtained with direct substitution (by fixing w to 0) are given in this table.

From the results presented it is evident that no single combination of w and  $t_{max}$  can be considered optimum. However, it is clear that the acceleration method performed better than direct substitution except for the vapour search for System 1 for which similar results where obtained with both methods. For System 2 the use of Wegstein's method in every iteration lead to faulty convergence irrespective of the value of  $t_{max}$  when searching for liquid phases. This happened also for w=2 and w=3 when the value chosen for  $t_{max}$  was small ( $t_{max}=1.5$ ) and similarly when direct substitution was used. The results obtained indicate that the use of Wegstein's method every iteration is not advisable since oscillatory behavior in the convergence was discovered in a few cases and even divergence may occur. By alternating with one or more direct substitution steps this problem was eliminated. The best results were obtained when the acceleration is carried every two iterations and assigning a value of  $t_{max}$  between 3 and 6. Higher values for this last

		Wegstein's acc	celeration me	thod	
		no. of	iterations va	pour/liquid	search
Wegstein's	method	System (	no. of data p	oints vapou	r/liquid)
		1 (5/6)	2 (5/5)	(1/1)	3 (1/4)
frequency	$t_{max}$	120→220 K	90→170 K	169.9 K	430 K
w		20 atm	40 atm	60.5 atm	$20 \rightarrow 50 atm$
	1.5	10/39	18/55**	7/14	9/17
1	3.0	10/35	18/61*	7/12	8/17
-	4.5	10/35	18/54*	7/12	6/17
	6.0	10/35	18/54*	7'/12	11/17
	1.5	10/40	18/82*	7/17	9/17
2	3.0	10/35	18/83	7/11	9/17
	4.5	10/35	18/68	7/11	6/17
	6.0	10/35	18/65	7/11	10/17
	1.5	10/40	18/82*	8/17	10/17
3	3.0	$\frac{10}{36}$	18/84	6/13	9/17
·	4.5	10/35	18/79	6/13	11/17
	6.0	10/35	18/73	6/13	11/17
direct substitution 10/43 1			19/89*	10/19	13/19

\*\* Erroneous prediction obtained at two points because of faulty convergence.

Table 4.8: Performance of Wegstein's method in the vapour and liquid phase search as a function of parameters w and  $t_{max}$  for Systems 1, 2 and 3 compared to the use of direct substitution.

parameter did not improve the outcome of the calculations and can be detrimental if much larger than 10; values too close to 1 reduce the method to direct substitution. A second choice would be to accelerate every third iteration with the maximum step size parameter also in the interval from 3 to 6. Because the optimum convergence parameters may vary from one system to another, the values given above (obtained from the results of Systems 1, 2 and 3) served only as guidelines which were used for the other systems tested. It was found that convergence in the vapour search was relatively insensitive to parameters w and  $t_{max}$  except at high pressures, where direct substitution became slow and the use of Wegstein's method improved significantly the convergence rate as for the liquid search.

Although for many of the points tested the objective function  $\Im(Y_i)$  or  $\Im_j(X_{j,i})$  (defined by equations (3.46) and (3.73) decreased right after the acceleration method was applied, this was not always the case. For some systems (in particular 1 and 2) it was frequent that when compositions were accelerated with Wegstein's method, the objective function evaluated at that iteration increased with respect to the value obtained in the previous one. Then in the following iteration a significant decrease would be achieved. This behaviour can be explained by the way in which  $\Im$  is defined, i.e., as the sum of the absolute value of the difference in compositions obtained in the present and in the previous iterations. Thus, when a notable variation in the compositions was obtained after accelerating, the objective function augmented accordingly. Then in the following iteration, with the compositions closer to the converged values, 3 would decrease markedly. On a few occasions it was also found that when the objective function 3 increased after using Wegstein's method, oscillations in the calculation and even divergence would occur if the method was applied continuously with the pre-specified frequency. For this reason whenever the value of  $\Im$  increased after accelerating, the frequency of the method was reduced to break the undesirable pattern. In such cases it is employed only every  $\eta * w$  iterations, where  $\eta$  is a control variable which is increased by one (initially  $\eta=1$ ) when the objective function increases. When  $\Im$  decreases after accelerating, the frequency of use is set to its original value by fixing  $\eta$  to 1. This strategy worked well in every case tested allowing the use of Wegstein's method in a reliable manner.

Examples of the matters treated above are given in Figures 4.21, 4.22 and 4.23 where the performance of Wegstein's method is compared with that of direct substitution. Figure 4.21 concerns the vapour phase search for System 2 at 169.9 K and 60.5 atm; at this high pressure convergence became slow when direct substitution was used, requiring 10 iterations to achieve a tolerance of  $10^{-5}$ . Using Wegstein's method with a frequency w=3 and with  $t_{max}=3$  convergence was reached in only 6 iterations (the same result was obtained with  $t_{max}$  set to 4.5 or 6 as indicated in Table 4.8). In this case, when compositions were accelerated in the fourth iteration the objective function decreases with respect to the value of the previous iteration, but increased when compared to the result obtained when using direct substitution. Then in the next iteration a considerable reduction in S was obtained leading to convergence in one more iteration. Since the frequency of application was set to 3 in this case, the acceleration would have been conducted in the seventh iteration if convergence had not been reached in the sixth. When w was fixed to 2 (not shown in the figure) acceleration took place in the sixth iteration but the objective function did not decrease enough to achieve convergence until the seventh iteration.

In Figure 4.22 the convergence behaviour for the liquid phase search is shown again for System 2 but at 160 K and 40 atm. Although the convergence criterion was not met ( $\varepsilon = 10^{-5}$ ) because the calculation was terminated early (check (e) in Step 7 of the search method was satisfied), using Wegstein's method improved the convergence rate considerably. In this case w was set to 2 with  $t_{max}$ =4.5. After accelerating in iteration 4, the objective function increased in relation to the previous iteration and thus parameter



Figure 4.21: Performance of direct substitution and Wegstein's method in the vapour phase search for System 2 at 169.9 K and 60.5 atm



Figure 4.22: Convergence behaviour when using direct substitution and Wegstein's method in the liquid phase search for System 2 at 160 K and 40 atm.



Figure 4.23: Convergence characteristics of Wegstein's method compared with direct substitution in the liquid phase search for System 1 at 180 K and 20 atm.

 $\eta$  was set to 2. Then the frequency of application of Wegstein's method was changed to every 4 iterations ( $\eta * w = 4$ ); therefore the next time compositions were accelerated was in iteration 8. If the calculation had not been terminated early and convergence had not been reached, since in iteration 8 the objective function increased, the method would have been applied again in iteration 14 (since  $\eta$  was set to 3 acceleration would be conducted every 6 iterations). On both occasions that Wegstein's method was used  $\Im$  increased in that iteration but decreased notably in the next. Only the convergence behaviour of one composition vector is shown in Figure 4.22 since in this case only one phase was found in the search.

For System 1 at 180 K and 20 atm two composition vectors were converged because two liquid phases were found in the search. The results are shown in Figure 4.23. Applying initially Wegstein's method every 3 iterations with  $t_{max}$  fixed to 3, the convergence tolerance ( $\varepsilon = 10^{-5}$ ) was reached for liquid I (the slowest to converge) in 11 iterations, whereas with direct substitution it took 19. For this phase the objective function increased after accelerating the first time (iteration 4) and also the next occasion in iteration 10 (the acceleration frequency was changed from 3 to 6). In both instances a decrease in  $\Im$  was obtained in the following iteration. For the composition vector corresponding to liquid II convergence was achieved in 8 iterations whether Wegstein's method was used or not, but when used the objective function  $\Im$  acquired a much lower value. After accelerating the first time  $\Im$  was reduced and the method was applied again in the seventh iteration, where it increased (with respect to the value obtained in iteration 6) but then dropped in the next iteration, reaching convergence.

### 4.4 Phase removal tests

The strategies developed for the phase removal tests to eliminate superfluous phases were described in detail in Section 3.4. In the previous Sections 4.3.1 and 4.3.2 it was mentioned that whenever three potential phases were found in the phase search, the need for such tests arose mainly in calculations near saturation point conditions. Examples already seen are the superfluous vapour found for System 2 at 150 K and 40 atm near the three-phase bubble-point (located at 150.2 K at 40 atm) and the calculation for System 1 at 180 K and 20 atm where a superfluous liquid was found close to the three-phase dew-point conditions (171.1 K at 20 atm).

## 4.4.1 Multiphase flash method

In the iterative scheme proposed for the multiphase flash method a maximum of five iterations are allowed to determine if any of the phases found should be eliminated. This limit was set to gain efficiency by declaring the system stable as VLL if at that point no phase had been eliminated, since each iteration corresponds to a full three-phase flash calculation. Also, by setting the limit of the control variables  $\nu_V$ ,  $\nu_{L_I}$  and  $\nu_{L_{II}}$ to 3 (as opposed to the value of 5 indicated in [5]) a good compromise was obtained between speed and dependability. In Table 4.9 are given the results obtained with this method (also with the bubble- and dew-point equations method) and with the subsequent flash calculation for three systems tested near saturation point conditions. In all cases the splitting algorithm was carried out without the additional phase search option. For System 1 at the feed composition  $z_1=0.5$ ,  $z_2=0.1$ ,  $z_3=0.4$  and 20 atm, correct results were obtained with the phase removal method at conditions near the three-phase bubble- and dew-points (located at 170.1 K and 171.1 K respectively). In all cases the outcome of the test indicated the presence of the phases corresponding to the stable system. These predictions were the same as obtained when the subsequent flash calculation followed (either two- or three-phase, depending on the results of the phase removal test) to obtain the exact phase distribution and compositions.

In four of the six points tested for System 2 near saturation points (Table 4.9), no phase elimination took place when the mixture was actually stable as a two-phase system (the three-phase bubble- and dew-point temperatures are 150.2 and 151.3 K at 40 atm). However, when the eventual three-phase flash was conducted to obtain quantitative results, it was properly reduced to a two- phase flash giving correct predictions. In such cases the phase- splitting algorithm anticipated a higher number of phases present at equilibrium. This problem could be avoided by increasing the limit in the number of iterations in the phase removal test, but since usually a flash calculation follows the phase-splitting prediction (except when only the phase distribution is pursued), this was not considered necessary. As shown in Table 4.9, near the three-phase bubble-point correct predictions were obtained for System 3 with the phase removal test and with the subsequent flash.

		Phase	e removal tests	5		
T(K)	stable system	multiphase test result	flash method flash result	bubble & de test result	ew point method flash result	
	System 1: CH <sub>4</sub> (	1)- CO <sub>2</sub> (2)-	$-{\rm H}_2{ m S}(3)$ $z_1 =$	$0.5  z_2 = 0.1$	$P{=}20 \text{ atm}$	
167	$\mathbf{L}\mathbf{L}$	$\mathbf{L}\mathbf{L}$	$\mathbf{L}\mathbf{L}$	VL*	VL	
170	$\mathbf{L}\mathbf{L}$	$\mathbf{L}\mathbf{L}$	$\mathbf{L}\mathbf{L}$	VLL+	$VL^{\times}$	
170.6	$\operatorname{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	
171	$\operatorname{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	
172	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$	VL	VL	
180	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{VL}$	$\mathbf{L}^{*}$	$\mathbf{L}$	
200	VL	VL	VL	VLL+	VL	
4	System 2: hydro	carbons(1-5	)- N <sub>2</sub> (6) $P=$	=20 atm		
150	$\mathbf{L}\mathbf{L}$	VLL+	$\mathbf{L}\mathbf{L}$	VLL+	VL	
150.9	VLL	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	$\mathbf{VLL}$	
151.7	$\mathbf{VL}$	$VLL^+$	$\mathbf{VL}$	VLL+	VL	
152	$\mathbf{VL}$	VLL+	$\mathbf{VL}$	$VLL^+$	VL	
160	$\mathbf{VL}$	VLL+	$\mathbf{VL}$	VLL+	$\mathbf{VL}$	
180	VL	VL	VL	VLL	$LL^{\times}$	
P(atm)	System 3: hydro	carbons(1-5	$- H_2O(6) $	<b>F=430</b> K		
25	VLL	VLL	VLL	VLL	VLL	
30	VLL	VLL	VLL	VLL	VLL	
35	VLL	VLL	VLL	VLL	VLL	
40	LL	LL	LL	LL		
<ul> <li>* Erroneous result in the phase removal test leading to incorrect phase-splitting prediction.</li> <li>+ No phase elimination with the phase removal tests but done in the subsequent 3-phase flash.</li> <li>× Incorrect result in the 3-phase flash.</li> </ul>						

Table 4.9: Performance of the phase removal tests and of the subsequent flash calculation in the phase-splitting prediction for Systems 1, 2 and 3 near saturation point conditions.

#### 4.4.2 Coupled method

Earlier in Section 3.4 it was mentioned that whenever the option to carry out the additional phase search is selected in the phase- splitting algorithm, superfluous phases (either vapour or liquid) may be found. These phases should then be eliminated with the phase removal tests proposed. Unfortunately, it was discovered that when conducting the additional phase search, the outcome of the phase reduction test based on the multiphase flash method was incorrect in some cases due to very poor composition estimates of the additional liquid phase found which would throw-off the calculation. This did not happen when the test based on the bubble- and dew-point method of Nelson was applied. The results of these tests are presented in Table 4.10 for Systems 1 and 6 at the conditions specified when tested in the entire composition range (considering 66 different feed compositions). For the former system at 171 K and 20 atm, two incorrect results were obtained with the multiphase flash removal test when the additional phase search was not conducted; however, when it was executed, incorrect predictions were obtained in 5 more occasions (nevertheless more incorrect results were obtained with the bubbleand dew-point equations test and with the direct three-phase flash). In the additional 5 cases the system was found to be liquid-like, initially a vapour was found in the search but no liquid phases <sup>5</sup>, and with the additional phase search a liquid was recognized. With Nelson's method (bubble- and dew-point) these points were correctly predicted. Poorer results were obtained with the multiphase flash method at 200 K and 20 atm when conducting the additional phase search since the incorrect predictions hiked from 2 to 13, a number much higher than with the other options. Again in these cases the system was liquid-like, in the search a vapour was detected but no liquids, except in the additional search, and Nelson's method performed fine.

<sup>&</sup>lt;sup>5</sup>This corresponds to case iii) described in the phase search strategy at the beginning of Section 3.2

Because of these observations it was decided to implement the improved empirical phase removal strategy described in Section 3.4.3 (coupled method), which can be selected as an option when the type of phase removal test is specified. The results obtained with this strategy are given in Table 4.10. For System 1 the incorrect predictions were reduced to 2 at both temperatures, making the method the most reliable and efficient at 171 K, and as reliable as the direct three-phase flash but slightly slower at 200 K.

In contrast to System 1, for System 6 the additional phase search helped in reducing the incorrect predictions from 9 to 5 when the multiphase flash reduction test was used. Nevertheless, in two cases incorrect predictions were obtained again because poor estimates of the incipient liquid found. With the coupled method devised these wrong predictions were avoided, reducing to 3 the cases where faulty results were obtained as compared to 4 when Nelson's method was employed and to 13 when the direct three-phase flash was conducted.

## 4.4.3 Bubble- and dew-point equations method

In order to increase the efficiency of the method suggested by Nelson [3], one or two of the phases found in the search are subject to be eliminated when any of the corresponding tests is satisfied in three consecutive iterations instead of waiting until convergence is obtained (see Section 3.4.2). This is justified by the observation made in practically all the cases tested: once three consecutive identical predictions were obtained, the same result would be obtained when convergence was reached. The results given in Table 4.9 (which were obtained without the additional phase search) indicate that this method was less reliable than the multiphase flash approach for Systems 1 and 2 but performed equally well for System 3.

In two occasions incorrect results were obtained for System 1 because of the phase removal test (at 167 K and 180 K); for the point at 180 K the removal of two phases

Pł	nase removal tests	5		
	no. of incorrect predictions (CPU time, msecs.)			
	Syst 171 K, 20 atm	em 1 200 K, 20 atm	System 6 311 K, 60 atm	
multiphase flash method without the additional search	2 (1150)	2 (917)	9 (1066)	
multiphase flash method with the additional phase search	7 (1238)	13 (1088)	5 (1295)	
bubble & dew point method without the additional search	9 (1214)	0 (925)	10 (1138)	
bubble & dew point method with the additional search	9 (1329)	0 (1228)	4 (1401)	
coupled method with the additional search	2 (1237)	2 (1210)	3 (1297)	
direct three-phase flash	8 (1276)	2 (1024)	13 (906)	
Note: 66 points where tested for each system in the entire composition range.				

Table 4.10: Comparison of the phase removal tests reliability with and without the additional phase search versus the direct three-phase flash calculation for Systems 1 and 6. was incorrectly executed, giving an erroneous L solution. For the points tested at 170 K and 200 K no phase removal was done in spite of a two-phase solution being stable. At 170 K, three phases were anticipated and when the subsequent flash was executed, an erroneous VL prediction was obtained instead of the stable LL solution. At 200 K the three- phase flash was reduced to a VL flash giving a correct final result. For System 2, VLL solutions were determined in four cases whereas the final calculation conducted with the flash (three- phase) indicated a VL stable mixture. Thus, as with the multiphase flash based method, more phases than those present at equilibrium were predicted in most cases for this mixture. At 180 K again three phases were declared stable and then the eventual flash gave an erroneous LL solution. Finally, the results obtained for System 3 gave correct predictions for all the points tested.

When considering the results given in Table 4.10, it is evident that performing the additional phase search did not set back the outcome of the phase removal test based on the bubble- and dew- point method, and for System 6 it aided in six occasions to obtain correct predictions. Although for System 1 at 200 K correct results were obtained in every case with this test, when considering the points tested at 171 K and for System 6 as well as those shown in Table 4.9, it seems that Nelson's approach is more prone to failure than the method based on the multiphase flash. An exception to this is the particular situation described above when the additional phase search is conducted (i.e. a liquid is found when the system is liquid-like, a vapour is found in the initial search but no liquids), where the use of Nelson's method avoids the incorrect predictions. The coupled method suggested takes advantage of this and constituted the most robust option for the systems mentioned above as well as for the other systems tested.

# 4.5 Overall phase-splitting prediction performance

As indicated at the beginning of this chapter, the equilibria conditions predicted with the algorithm proposed are compared with the results of the direct three-phase flash calculation. To obtain the equilibrium compositions and phase ratios after the number and type of phases have been determined with the phase-splitting algorithm, the corresponding flash calculations (if necessary) are also executed. As starting values for these calculations the compositions found in the phase search are used as suggested in the general scheme described in Section 3.1. For the direct three- phase flash method, the initial estimates of the compositions and the phase ratios are obtained according to the initialization schemes described in Appendix A. In order to compare the computational effort required to generate the initial estimates in this form and with the phase-splitting algorithm, as well as that needed to conduct the flash <sup>6</sup>, detailed results indicating the CPU time (milliseconds) spent for these computations are given for Systems 1, 2, 3 and 7 in Tables 4.11 to 4.14. In these tables are also shown the temperatures and pressures specified, the stable systems at equilibrium, the predictions with the VLL flash and with the phase- splitting algorithm, and the type of flash required according to this last algorithm.

Finally, in Table 4.15 is given a summary of the results obtained for the seven systems tested. In this table are indicated the number of data points tested for each system, the number of cases in which incorrect predictions were determined with the direct three-phase flash and with the corresponding flash following the splitting algorithm (if necessary), and the CPU time (msecs.) required for the entire calculations (comprising either the flash initialization or the phase-splitting step, and the flash). The parameters and

<sup>&</sup>lt;sup>6</sup>When the flash calculations are performed, compositions are accelerated with Wegstein's method applied every 2 iterations with damping since this gives the best results according to [5] (see also Appendix A).

options selected for the phase-splitting algorithm are those considered as default values, based on the overall best performance for the systems tested. These are: initialization of the liquid phase search with the proposed algorithm (except for binary mixtures in which the modified Michelsen method is used), Wegstein's method applied every two iterations (w=2) with  $t_{max} = 2$ , use of the additional phase search, convergence tolerance  $\varepsilon = 10^{-3}$ in the search and phase removal test with the coupled method. To make the comparisons equitable, the same convergence tolerance  $\varepsilon$  was specified for the flash calculations in the direct VLL flash and the flash conducted after the phase-splitting prediction (set to  $10^{-5}$ ).

For System 1 (Table 4.11) correct predictions were obtained with both methods for all the points in the temperature range specified. Considering that the three-phase bubble and dew temperatures are 170.1 and 171.1 K respectively and that the two-phase dew-point occurs at 260.8 K, the predictions obtained with the phase-splitting algorithm indicate that the phase search and phase removal tests worked equally well near these saturation points as within the one-, two- and three-phase regions. Although the initialization for the direct three-phase flash as suggested in [5] was much quicker than with the phase-splitting algorithm for this three component system, the flash calculations required less computational effort when the phase-splitting results were employed. The reason for this is that the values provided in this form are much closer to the equilibrium solutions. Also, the flash calculation was very efficient because a considerable amount of time was saved by conducting only a two-phase flash when only two phases were present and since the flash was avoided at conditions where only one phase exists. However, the overall CPU time consumed with both methods was practically the same: 251 msecs. for the direct three-phase flash against 250 for the phase- splitting-flash method initializing with the proposed algorithm, using Wegstein's method with w = 3,  $t_{max} = 3$ , without the additional phase search, using the multiphase flash removal test and setting  $\varepsilon = 10^{-3}$ 

Phase-splitting prediction performance							
System	System 1: CH <sub>4</sub> (1)-CO <sub>2</sub> (2)- H <sub>2</sub> S(3) $z_1 = 0.5 \ z_2 = 0.1 \ P=20 \text{ atm}$						
	prediction (initlzn. CPU time/flash CPU time)						
T(K)	stable system	VLL flash	phase- splitting	flash type			
120	$\mathbf{L}\mathbf{L}$	LL (2/8)	LL (9/3)	$2$ - $\phi$			
140	$\mathbf{L}\mathbf{L}$	LL(2/9)	LL $(12/4)$	$2-\phi$			
160	$\mathbf{L}\mathbf{L}$	LL $(2/10)$	LL(9/5)	$2$ - $\phi$			
166	$\mathbf{L}\mathbf{L}$	LL $(3/11)$	LL $(10/4)$	$2-\phi$			
167	$\mathbf{L}\mathbf{L}$	LL $(3/11)$	LL $(10/4)$	$2-\phi$			
170	$\mathbf{L}\mathbf{L}$	LL(3/9)	LL $(14/4)$	$2$ - $\phi$			
170.6	VLL	VLL (3/10)	VLL (18/8)	$3-\phi$			
171	VLL	VLL $(3/11)$	VLL (18/8)	$3-\phi$			
172	$\mathbf{VL}$	VL(3/14)	VL (15/4)	$2-\phi$			
180	$\mathbf{VL}$	VL(2/15)	VL(14/4)	$2$ - $\phi$			
200	$\mathbf{VL}$	VL(2/14)	VL(12/3)	$2 - \phi$			
220	$\mathbf{VL}$	VL(2/13)	VL(7/3)	$2$ - $\phi$			
240	$\mathbf{VL}$	VL(2/11)	VL(7/3)	$2$ - $\phi$			
260	VL	VL(3/11)	VL(7/2)	$2-\phi$			
261	V	V(3/11)	V(7/0)	not needed			
280	V	V(3/12)	V(7/0)	not needed			
300	$\mathbf{V}$ .	V(2/13)	V (8/0)	not needed			
320	V	V (3/12)	V (7/0)	not needed			
Total (46/205) (191/59)							
Initizn. for the liquid search with the proposed algorithm, $w = 3$ , $t_{max} = 3$ , no additional phase search, multiphase flash phase removal test.							

Table 4.11: Comparison of the prediction of the phase distribution for System 1 at different temperatures with the direct three-phase flash calculation and with the phase-splitting algorithm proposed followed by the corresponding type of flash. (268 msecs. were required when fixing  $\varepsilon$  to  $10^{-5}$  in the phase search, but practically identical results were obtained). Therefore, the efficiency and reliability of both approaches is comparable for this system at the conditions specified to obtain the results given in Table 4.11 and no advantage is gained using a particular method.

The results pertaining to the calculations at 171 K and 200 K (both at 20 atm) when the feed was varied along the entire composition plane are also summarized in Table 4.15. When the specified temperature is 171 K, the phase-splitting algorithm was more dependable and slightly more efficient than the direct flash calculation. At 200 K both methods gave 2 erroneous predictions, with the former algorithm requiring also less global CPU time to obtain the equilibrium compositions and phase distribution. By reducing the convergence tolerance from  $10^{-3}$  to  $10^{-5}$  in the phase search, the CPU times increased to 1264 msecs. at 171 K and 1194 at 200 K (not shown in Table 4.15), but this did not alter the outcome of the calculations and still two erroneous predictions were obtained in each case.

In the calculations done for System 2, the direct three-phase flash gave incorrect results in 5 of the 19 points tested. Although when using the phase-splitting algorithm a three-phase solution was predicted in 4 occasions, the stable system being two-phase, correct predictions were obtained after executing the flash as shown in Table 4.12 (this has already been discussed in Section 3.4 and presented in Table 4.9). For the rest of the points where only two phases exist at equilibrium, the use of the specific two-phase flash (either LL or VL) reduced considerably the computational load required, and when only the existence of one phase was predicted the evasion of the flash was definitely time-saving. For this six component mixture the generation of initial estimates for the direct three-phase flash was faster on a global basis (but not at every point) than with the phase-splitting counterpart when not considering the following flash calculation (492 msecs. against 608). However, the difference was reduced notably when the results

Г

	Phase	e-splitting predi	iction performance		
	System 2: 1	nydrocarbons(1	$-5)-N_2(6)$ $P=40$	0 atm	
	predic	tion (initlzn. C	CPU time/flash CP	U time)	
T(K)	stable system	VLL flash	phase- splitting	flash type	
90	$\mathbf{L}\mathbf{L}$	L* (25/32)	LL (32/8)	$2$ - $\phi$	
100	$\mathbf{L}\mathbf{L}$	$L^{*}(26/33)$	LL(54/7)	$2$ - $\phi$	
110	$\mathbf L$	L(26/32)	L(22/0)	not needed	
120	$\mathbf L$	L (26/31)	L(22/0)	not needed	
<b>13</b> 0	$\mathbf L$	L (26/29)	L(23/0)	not needed	
140	$\mathbf{L}$	L (26/35)	L(22/0)	not needed	
150	$\mathbf{L}\mathbf{L}$	VL* (25/47)	LL (58/113)	$3-\phi^+$	
150.9	$\mathbf{VLL}$	VL* (26/35)	VLL (58/13)	$3$ - $\phi$	
151.7	$\mathbf{VL}$	VL (26/35)	VL (57/27)	$3-\phi^+$	
152	$\mathbf{VL}$	VL(26/35)	VL (57/32)	$3-\phi^+$	
160	$\mathbf{VL}$	VL(25/37)	VL (46/31)	$3-\phi^+$	
180	$\mathbf{VL}$	VL (26/79)	VL (33/11)	$2$ - $\phi$	
200	$\mathbf{VL}$	VL (26/28)	VL (17/12)	$2$ - $\phi$	
220	$\mathbf{VL}$	VL (26/29)	VL (17/8)	$2$ - $\phi$	
240	$\mathbf{VL}$	VL (26/31)	VL (17/8)	$2$ - $\phi$	
260	$\mathbf{VL}$	VL(26/32)	VL (17/8)	$2$ - $\phi$	
280	$\mathbf{VL}$	V* (27/34)	VL (21/8)	$2$ - $\phi$	
300	V	V (26/33)	V (17/0)	not needed	
320	V	V (26/33)	V (18/0)	not needed	
	Total	(492/680)	(608/286)		
* Incor	rect prediction.				
<sup>+</sup> No phase elimination in the phase removal test but done in the subsequent flash.					
Initlzn.	for the liquid sea	rch with the prop	osed algorithm, $w = 2$	$t_{max} = 5, no$	
additio	nal phase search, i	nultiphase flash p	bhase removal test.		

Table 4.12: Predictions of the equilibrium conditions for System 2 at different temperatures with the direct three-phase flash calculation and with the phase-splitting algorithm proposed followed by the corresponding type of flash.

are compared to the three component System 1 (Table 4.11). This trend, i.e., the phasesplitting approach being much more efficient as the number of components in the mixture increases, was observed in all cases and becomes very patent for System 7 (having 13 constituents) as will be addressed later. Table 4.15 indicates that for the points tested, the phase-splitting algorithm is not only more reliable on a global basis when the flash is used subsequently, but also more efficient.

The phase-splitting-flash method proposed was particularly advantageous over the direct three-phase flash under the conditions tested for System 3. Both methods performed without fault in all instances, but the gain in efficiency when the equilibrium conditions were determined with the phase-splitting method is evident as shown in Table 4.13 and particularly with the default parameters according to Table 4.15. In most cases the initialization part of the methods was faster with the phase- splitting approach, but the benefit of this method came mainly in the final flash calculation (when necessary) where convergence was obtained readily. Only a few iterations were required in the flash because the equilibrium composition estimates generated with the phase-splitting method (by means of the phase search strategy) are very close to the solution. Although only three points were tested for the 13 component mixture tested (System 7), it is evident that when a large number of species are present the computational load required in the phase equilibria predictions can be greatly decreased if the phase- splitting method is employed. This can be observed in the results presented in Table 4.14. Not only the total CPU time was brought down when compared to the direct three-phase flash because the flash is avoided or only the two-phase calculation is performed, but also the initialization of these calculations was much faster with the phase-splitting approach.

For System 4, which is one of the binary mixtures tested in the whole composition range, the phase-splitting-flash method was slightly more time-consuming than the direct three-phase flash when calculations were done with the default values. Nevertheless, -

. . .

L	System 5. Hydro	carbons(1-3)=1	12O(0) 1 - 430	aim		
	predictio	on (initlzn. CP)	U time/flash CPU	time)		
P(atm)	stable system	VLL flash	phase- splitting	flash type		
5	v	V (27/25)	V (15/0)	not needed		
10	V	V (25/25)	V (15/0)	not needed		
15	v	V (27/22)	V (15/0)	not needed		
20	$\mathbf{VL}$	VL(27/24)	VL(18/4)	$2$ - $\phi$		
25	$\mathbf{VLL}$	VLL (27/16)	VLL(34/3)	$3-\phi$		
30	$\mathbf{VLL}$	VLL (27/16)	VLL (34/3)	$3-\phi$		
35	$\mathbf{VLL}$	VLL (26/22)	VLL (36/3)	$3-\phi$		
40	$\mathbf{L}\mathbf{L}$	LL $(26/41)$	LL $(22/4)$	$2-\phi$		
60	$\mathbf{L}\mathbf{L}$	LL (26/22)	LL $(21/8)$	$2 - \phi$		
80	$\mathbf{L}\mathbf{L}$	LL (26/18)	LL (19/8)	$2-\phi$		
100	$\mathbf{L}\mathbf{L}$	LL (25/18)	LL (19/8)	$2$ - $\phi$		
Total (289/249) (248/41)						
Initizn, for the liquid search with the proposed algorithm, $w = 3$ , $t_{max} = 3$ .						

Table 4.13: Number and type of phases predicted for System 3 at different pressures with the direct three-phase flash calculation and with the phase-splitting algorithm proposed followed by the corresponding flash.

· .

considering that at all points correct predictions were obtained as indicated in Table 4.15, this method is more appealing that the direct flash counterpart when dependability becomes an important factor. For this last method (direct three-phase flash), erroneous predictions were obtained in 4 occasions. For the other binary mixture tested, System 5, the phase-splitting algorithm was more reliable than the direct flash (no erroneous predictions versus two with the flash) and also more efficient as indicated in Table 4.15.

The results obtained for System 6 (ternary mixture), given also in Table 4.15, indicate again that the direct flash was somewhat more efficient than the proposed method, but reliability was sacrificed. In this case 13 incorrect results were predicted with the direct flash (out of 66), whereas the phase-splitting approach failed to give correct predictions only at 3 feed points. By setting  $\varepsilon$  to  $10^{-5}$  reliability was not improved and the CPU time increased to 1297 msecs., which indicates that a more strict convergence tolerance in the phase search was not beneficial.

م الم

149

Phase-splitting prediction performance								
	System 7: compressor gas mixture							
		predictio	n (initlzn. CPI	U time/flash CPU	J time)			
T(K)	T(K) $P(atm)$ stable system VLL flash phase-splitting flash type							
283	5	V	V (495/68)	V (52/0)	not needed			
283	15	$\mathbf{VL}$	VL (504/84)	VL (53/27)	$2$ - $\phi$			
327	15	V	V (504/71)	V (54/0)	not needed			
Total (1503/223) (159/27)								
Initizn. for the liquid search with the proposed algorithm, $w = 3$ , $t_{max} = 3$ , no								
additional phase search, multiphase flash phase removal test.								

Table 4.14: Equilibria predictions for System 7 at three temperature and pressure conditions with the direct three-phase flash calculation and with the phase-splitting algorithm proposed followed by the corresponding flash.

Summary of the results for the systems tested using the default parameters for the phase-splitting predictions								
n	no. of wrong predictions (total CPU time, msecs.)							
system no.	no. data points	VLL flash	phase- splitting	T(K), P(atm)				
1	18	0 (251)	0 (278)	$120 \rightarrow 320,20$ fixed z				
1	66	8 (1276)	2 (1239)	171,20 all z				
1	66	2 (1204)	2 (1137)	200,20 all <i>z</i>				
2	19	5 (1172)	0 (961)	$90 \rightarrow 320,40$ fixed z				
3	11	0 (538)	0 (267)	$430,5 \rightarrow 100$ fixed z				
4	21	4 (143)	0 (184)	378,5 all $z$				
5	21	2 (276)	0 (256)	190,38 all <i>z</i>				
6	66	13 (906)	3 (1239)	311,60 all <i>z</i>				
7	3	0 (1726)	0 (186)	283,5&15 327,15 all <i>z</i>				
Total	281	34 (7492)	7 (5747)					
Default parameters: Initlzn. for the liquid search with the proposed algorithm, Wegstein's method with $w = 3$ , $t_{max} = 3$ , additional phase search, phase removal test with the coupled method.								

Table 4.15: Summary of the phase equilibrium predictions for the seven systems tested obtained with the direct three-phase flash calculation and with the phase-splitting algorithm proposed followed by the corresponding type of flash when required.

## Chapter 5

## Conclusions and recommendations

### 5.1 General conclusions

Based on the evaluation of the results presented in the previous chapter, it can be stated that the algorithm developed, in most instances, is able to accurately and efficiently:

- 1. Establish a priori the least number of phases present at equilibrium in order to determine the need for an isothermal flash calculation.
- Predict the configuration of the system when more than one phase is found, i.e. predict the type of phases and the corresponding flash calculation to be executed: VL, LL or VLL.
- 3. Concurrently generate composition estimates of the phases present to be used in the eventual flash which are close to the equilibrium values, thus avoiding convergence to a trivial solution.

This means that the objectives proposed at the beginning of this work have been reached and that the phase-splitting algorithm suggested offers a better alternative than the direct three-phase flash calculation when the equilibrium conditions for multicomponent mixtures capable of separating into three fluid phases are to be determined. The major advantage of using the proposed algorithm is its dependability since incorrect predictions were obtained in only 2.5 % of the cases tested against 12.1 % when the direct flash was employed (which represents a reduction of almost 80 %). Also a very desirable characteristic of the splitting algorithm is it's efficiency when compared to the direct flash approach, although the reduction in CPU time for the equilibrium predictions was a not so overwhelming 23 %<sup>1</sup>. Nevertheless this is a significant improvement when considering that refinements in reliability are almost always done at the sacrifice of speed, and here a good compromise between both characteristics was achieved.

# 5.2 Specific remarks and recommendations

In spite of being very dependable, incorrect predictions were still obtained with the proposed algorithm for some of the points tested (see Table 4.15). The source of the failures was the inability to detect a vapour phase even when the additional phase search was conducted (in two occasions for System 1 at 171 K and 20 atm and once for System 6 at 311 K and 60 atm) and the incorrect outcome of the phase removal strategy (twice for System 1 at 200 K and 20 atm a liquid was not eliminated when it should have been and in two other cases a vapour was not removed for System 6 at 311 K and 60 atm). Therefore, these parts of the algorithm are the ones that would need improvement in order to have a more robust method. With respect to the efficiency of the calculations, it was observed that changing the convergence tolerance  $\varepsilon$  in the phase search from  $10^{-5}$  to  $10^{-3}$  reduced the CPU time required in the search (and also that of the total calculation) without affecting the reliability of the predictions. However, it diminished the accuracy of the compositions of the incipient phases used to initialize the flash calculations. This was noted because in some cases more iterations were required in the flash when the less strict convergence tolerance was set (and consequently more CPU time). Thus a more relaxed value of  $\varepsilon$  can have a negative effect on the efficiency of the method and

<sup>&</sup>lt;sup>1</sup>Clearly these figures are only gross indicators of the performance of both methods, but serve as guidelines for comparison when considering that only a limited number of systems and conditions could be tested.

ultimately in its reliability.

The particular conclusions that can be drawn from each part of the algorithm and some recommendations are:

- 1. The assumption of ideal gas behaviour to provide initial estimates for the vapour phase search was acceptable for the conditions tested, which were away from critical and retrograde behaviour regions. As Michelsen [4] suggests, in the critical region the compositions corresponding to the ideal liquid could also be used since by definition the vapour and liquid phases become indistinguishable.
- 2. For the liquid search initialization Shah's method was the most reliable. The new scheme proposed here was the next most reliable, and it was also the least time-consuming especially for mixtures with a large number of components. For binary systems the modified Michelsen method gave the fastest results and was very dependable. In the comparison for the efficiency of the initializations, the CPU time required to generate the composition estimates plus the time needed to reach convergence in the search gives a measure of how fast the initialization methods are and of how close to the incipient phase composition are the values provided (considering that normally the better the initial estimates the less CPU time required for convergence).
- 3. Convergence in the search strategy for a vapour phase was obtained in few iterations (three or less) for most cases except at high pressures. In these conditions the use of Wegstein's method accelerated the convergence rate compared to direct substitution, but the outcome of the search was not very sensitive to the acceleration parameters chosen. Nevertheless it is suggested to set the frequency of acceleration to each 2 or 3 iterations with a maximum step size between 3 and 6. Although the phase search performed without flaw in most cases, a superfluous vapour was

found occasionally near bubble-point conditions which had to be removed with the phase elimination tests. The additional phase search was very helpful in finding a vapour missed in the initial search for binary systems, but for the systems tested with 3 or more components it was not successful (the core of the problem was the pseudo-tangent plane criterion used, not the composition estimates generated since these were fairly close to the equilibrium values in all cases in which a vapour was not found).

4. With the modifications proposed to Michelsen's method the reliability of the liquid phase search as well as its efficiency improved considerably. The liquid search was found sensitive to the initialization scheme selected; when an incorrect prediction was obtained with one particular method, correct results were always obtained when using an alternative one. The use of Wegstein's method to accelerate compositions allowed convergence to be achieved much faster than with direct substitution. The selection of the parameters for acceleration frequency and maximum step size had a significant influence in the efficiency of the method. The best results were obtained accelerating every two iterations (starting in the fourth) with a maximum step size between 3 and 6 (specifically a value of 5 seemed to be the optimum); a second choice is to accelerate every three iterations with the maximum step size in the same interval (between 3 and 6). Acceleration every iteration is not recommended since it can disrupt the steady convergence to the solution. The additional phase search proposed was very useful when a liquid was missed in the initial search, particularly for binary systems. For the multicomponent systems tested it aided in obtaining a correct prediction in most instances, increasing the dependability of the phase-splitting predictions. Therefore, the additional phase search strategy should always be used for binary systems and is also recommended for multicomponent systems as long as the improved phase removal strategy is selected to assure the best possible performance (see next item). When superfluous liquid phases were found in the search, they were mainly in the vicinity of two- and three-phase dew-points. Away from these conditions the liquid search functioned quite well.

- 5. Concerning the removal of a superfluous vapour with the phase removal tests implemented, the one based on the multiphase flash method performed better than the bubble- and dew-point equations method, although it was not flawless. Similar results were obtained for the elimination of a superfluous liquid, except when the additional phase search was conducted (in systems with 3 or more components) for case iii) <sup>2</sup> where the bubble- and dew-point equations method was more reliable. The coupled method suggested takes this situation into account and uses the multiphase flash removal test in all cases but the one described. Thus it is recommended when choosing the phase removal test option.
- 6. The convergence tolerance specified for the flash calculations was set to  $10^{-5}$  in all cases. This value was strict enough to achieve accurate results, in particular to compare the Gibbs energies when two possible solutions were obtained in order to determine the stable one. A more strict convergence tolerance contributed only to an increase in the number of iterations to reach convergence, which were unnecessary since the outcome of the results was practically unchanged. For the phase search method a convergence tolerance of  $10^{-3}$  gave the best compromise between reliability and efficiency.

 $<sup>^{2}</sup>$ Corresponding to the case in which the system is liquid-like, a vapour is found in the initial search but not a liquid.

Based on these remarks, the default values for the parameters and options that are recommended for the phase-splitting algorithm are:

- Initialization of the liquid phase search: with the algorithm proposed here, except for binary systems in which the modified Michelsen method is recommended.
- Convergence tolerance for the phase search:  $\varepsilon = 10^{-3}$ .
- Convergence tolerance for the flash calculations:  $\varepsilon = 10^{-5}$ .
- Acceleration of compositions with Wegstein's method applied every two iterations (w = 2) with  $t_{max} = 5$ .
- Phase search using the additional phase search strategy.
- Phase removal test with the coupled method.

## **Bibliography**

- V. B. Shah, Multicomponent Distillation with Two Liquid Phases, Ph.D. Thesis, University of Toledo, Ohio (1980).
- [2] X. Joulia, R. Maggiochi and B. Koehret, 'Hybrid Methods for Phase Equilibrium Flash Calculations', *Fluid Phase Equilib.* 29, 15 (1986).
- [3] P. A. Nelson, 'Rapid Phase Determination in Multiple-Phase Flash Calculations', Comput. Chem. Engng. 11, 581 (1987).
- [4] M. L. Michelsen, 'The Isothermal Flash Problem. Part I. Stability', Fluid Phase Equilib. 9, 1 (1982).
- [5] F. Molina and A. A. Romero, Cálculo de Equilibrio Líquido-Líquido-Vapor para Sistemas Multicomponentes Utilizando Ecuaciones de Estado Cúbicas, B.A.Sc. Thesis, Universidad Nacional Autónoma de México, México (1987).
- [6] M. N. Ammar and H. Renon, 'The Isothermal Flash Problem: New Methods for Phase Split Calculations', AIChE J. 33, 926 (1987).
- [7] G. Soave, 'Equilibrium Calculations from a Modified Redlich-Kwong Equation of State', Chem. Eng. Sci. 27, 1197 (1972).
- [8] D. Y. Peng and D. B. Robinson, 'A New Two-Constant Equation of State', Ind. Eng. Chem. Funadm. 15, 59 (1976).
- [9] P. M. Mathias, 'A Versatile Phase Equilibrium Equation of State', Ind. Eng. Chem. Process Des. Dev. 22, 385 (1983).

- [10] P. M. Mathias, J. F. Boston and S. Watanasiri, 'Effective Utilization of Cubic Equations of State for Thermodynamic Properties is Process Simulation', AIChE J. 30, 182 (1984).
- [11] D. B. Van Dongen, M. F. Doherty and J. R. Haight, 'Material Stability of Multicomponent Mixtures and the Multiplicity of Solutions to Phase-Equilibrium Equations.
  1. Nonreacting Mixtures', Ind. Eng. Chem. Fundam. 22, 472 (1983).
- [12] M. L. Michelsen, "Comments on 'Material Stability of Multicomponent Mixtures and the Multiplicity of Solutions to Phase-Equilibrium Equations. 1. Nonreacting Mixtures'", Ind. Eng. Chem. Funadm. 23, 373 (1984).
- [13] B. L. Beegle, M. Modell and R. C. Reid, 'Thermodynamic Stability Criterion for Pure Substances and Mixtures', AIChE J. 20, 1200 (1974).
- [14] R. A. Heidemann, 'The Criteria for Thermodynamic Stability', AIChE J. 21, 824 (1975).
- [15] M. Modell and R.C. Reid, Thermodynamics and its Applications, 2nd. ed. Prentice-Hall, N.J. (1983).
- [16] G. Maurer and J.M. Prausnitz, 'Thermodynamics of Multicomponent Liquid-Liquid-Vapour Equilibria for Distillation Column Design', Inst. Chem. Eng. 3, 1.3/41 (1979).
- [17] I. F. Radzyminski and W. B. Whiting, 'Fluid Phase Stability and Equations of State', Fluid Phase Equilib. 34, 101 (1987).
- [18] R. L. Fournier and J. F. Boston, 'A Quasi-Newton Algorithm for Solving Multiphase Equilibrium Flash Problems', Chem. Eng. Commun. 8, 305 (1981).

- [19] R. Gautam and W. D. Seider, 'Computation of Phase and Chemical Equilibrium. Part II. Phase-Splitting', AIChE J. 25, 999 (1979).
- [20] F. F. Y. Walraven and P. V. Van Rompay, 'An Improved Phase-Splitting Algorithm', Comput. Chem. Engng. 12, 777 (1988).
- [21] L. E. Baker, A. C. Pierce and K. D. Luks, 'Gibbs Energy Analysis of Phase Equilibria', Soc. Pet. Eng. J., October, 731 (1982).
- [22] M. E. Soares, A. G. Medina, C. McDermott and N. Ashton, 'Three Phase Flash Calculation Using Free Energy Minimisation', Chem. Eng. Sci. 37, 521 (1982).
- [23] A. M. Crowe and M. Nishio, 'Convergence Promotion in the Simulation of Chemical Processes-The General Dominant Eigenvalue Method', AIChE J. 21, 528 (1975).
- [24] J. E. Dennis and J. J. Moré, 'Quasi-Newton Methods, Motivation and Theory', Siam Rev. 19, 46 (1977).
- [25] W. Murray, 'Second Derivative Methods', in W. Murray (Ed.) Methods for Unconstrained Optimization. Academic Press, London (1972).
- [26] J. S. Wu and P. R. Bishnoi, 'An Algorithm for Three- Phase Equilibrium Calculations', Comput. Chem Engng. 10, 269 (1986).
- [27] M. O. Ohanomah, Computational Algorithms for Multicomponent Phase Equilibria and Distillation, Ph.D. Thesis, University of British Columbia, Vancouver, B.C. (1981).
- [28] S. M. Walas, Phase Equilibria in Chemical Engineering. Butterworth Publishers, Boston (1985).

- [29] R. A. Heidemann, 'Three-Phase Equilibria using Equations of State', AIChE J. 20, 847 (1974).
- [30] D. J. Swank and J. C. Mullins, 'Evaluation of Methods for Calculating Liquid-Liquid Phase-Splitting', *Fluid Phase Equilib. 30*, 101 (1986).
- [31] J. H. Wegstein, 'Accelerating Convergence of Iteration Processes', Comm. ACM 1, 9 (1958).
- [32] R. C. Reid, J. M. Prausnitz and T. K. Sherwood, The Properties of Gases and Liquids. 3rd ed. McGraw-Hill, New York (1977).
- [33] M. L. Michelsen, 'The Isothermal Flash Problem. Part II. Phase-Split Calculation', Fluid Phase Equilib. 9, 21 (1982).
- [34] D. Y. Peng and D. B. Robinson, 'Two and Three Phase Equilibrium Calculations for Systems Containing Water', Can. J. Chem. Eng. 6, 245 (1982).
- [35] J. J. McKetta and D. L. Katz, 'Methane-n-Butane- Water System in Two- and Three-Phase Regions', Ind. Eng. Chem. 40, 853 (1948).
- [36] J. R. Elliot Jr. and T. E. Daubert, 'Revised Procedures for Phase Equilibrium Calculations with the Soave Equation of State', Ind. Eng. Chem. Process Des. Dev. 24, 743 (1985).
- [37] M. O. Ohanomah and D. W. Thompson, 'Computation of Multicomponent Phase Equilibria—Part I Vapour-Liquid Equilibria', Comp. Chem. Engng. 8, 147 (1984).
- [38] J. M. Prausnitz, T. F. Anderson, E. A. Grens, C. A. Eckert, R. Hsieh and J. P. O'Connell, Computer Calculations for Multicomponent Vapour-Liquid and Liquid-Liquid Equilibria. Prentice Hall, New Jersey (1980).

•

[39] M. O. Ohanomah and D. W. Thompson, 'Computation of Multicomponent Phase Equilibria—Part II Liquid-Liquid and Solid- Liquid Equilibria', Comp. Chem. Engng. 8, 157 (1984).

# Appendix A

### Two- and three-phase flash algorithms

The isothermal flash algorithms used in this thesis have been taken from the work of Molina and Romero [5] and are described in this section. The two-phase flash is capable of performing either a VL or LL calculation and the three-phase flash, which is intended for VLL computations, can be reduced to either of the first two types of equilibrium. The original algorithms have been slightly modified to adapt them to the scheme used here when any of the flash calculations is necessary after the phase-splitting prediction. Besides the initialization procedures for the three calculations (VL, LL and VLL), the phase ratio limiting strategies are also given, of which the one used for the VLL flash serves as a basis for the phase removal test described in Section 3.4.1. In addition, a brief description of Wegstein's method as applied to accelerate the compositions of the phases is included.

#### A.1 Isothermal VL or LL flash

In this calculation a feed stream of composition z is separated, at constant temperature and pressure, into two streams which can be a vapour and a liquid or two immiscible liquids in equilibrium. The problem consists in finding the compositions and phase ratios (on a molar basis) of the two new phases or in determining if the system is stable as a single phase. The data, unknowns and equations necessary to solve the flash are:

Data: T, P, z

Unknowns:

$$lpha = V/F, y_i, x_i$$
 (VL)  
 $eta = L_I/F, x_i^I, x_i^{II}$  (LL)

number of unknowns=2n + 1

# **Equations:**

- a) Mass balance
- $\alpha y_i + (1 \alpha) x_i = z_i \quad (VL)$

$$\beta x_i^I + (1 - \beta) x_i^{II} = z_i \quad \text{(LL)}$$

b) Equilibrium relations

$$f_i^L = f_i^V \quad (VL)$$
$$f_i^{L_I} = f_i^{L_{II}} \quad (LL)$$

c) Additional relations

$$\sum_{i=1}^{n} x_{i} - \sum_{i=1}^{n} y_{i} = 0 \quad (VL)$$
$$\sum_{i=1}^{n} x_{i}^{II} - \sum_{i=1}^{n} x_{i}^{I} = 0 \quad (LL)$$

number of equations = 2n + 1

where V,  $L_I$  and F are the mole numbers of the vapour, liquid I and the feed;  $y_i$ ,  $x_i$  (or  $x_i^I$ ,  $x_i^{II}$ ) and  $z_i$  are the corresponding mole fractions and f the fugacities in the indicated phases. The solution of this non-linear system of equations is given next for the case of a LL flash, and for VL calculations the method can be equally applied by substituting  $\alpha$ ,  $y_i$ ,  $x_i$  and  $K_{i_V}$  for  $\beta$ ,  $x_i^I$ ,  $x_i^{II}$  and  $K_{i_L}$  respectively. The equilibrium constants (as defined in equations (3.113) and (3.114)) are calculated initially in the starting routine. The steps followed in the calculation are:
## Two-phase flash algorithm

1. Start of the iterative cycle. Calculate  $\beta$  from

$$\Gamma_L(0,\beta) = \sum_{i=1}^n \frac{z_i(1-K_{i_L})}{1+\beta(K_{i_L}-1)} = 0$$
(A.131)

using a Newton-Raphson technique (inner iterative cycle) where the derivative is obtained analytically. If after 15 iterations the value of  $\beta$  does not converge to the solution, the calculation continues in the next step with the value generated in the final iteration.

- 2. If the calculated value of  $\beta$  is > 1 or < 0, it is limited to avoid the calculation of negative mole fractions in the next step. If the value of  $\beta$  is < 0 or > 1 in 5 consecutive global iterations, the calculation terminates indicating that at the specified conditions only one phase exists.
- 3. The new compositions of the two phases are obtained as:

$$x_i^{\star II} = \frac{z_i}{1 + \beta(K_{i_L} - 1)} \tag{A.132}$$

$$x_i^{\star I} = x_i^{\star II} K_{i_L} \tag{A.133}$$

which are then normalized

$$x_i^{II} = \frac{x_i^{*II}}{\sum_{i=1}^n x_i^{*II}}$$
(A.134)

$$x_{i}^{I} = \frac{x_{i}^{\star I}}{\sum_{i=1}^{n} x_{i}^{\star I}}$$
(A.135)

- 4. If applicable, the compositions of the two phases are accelerated using Wegstein's method.
- 5. The fugacity coefficients and the equilibrium constants are evaluated with the new compositions.
- 6. The objective function is calculated for the LL flash as:

$$\Im = \sum_{i=1}^{n} |\hat{\phi}_{i}^{L_{I}} x_{i}^{I} - \hat{\phi}_{i}^{L_{II}} x_{i}^{II}|$$
(A.136)

and for the VL calculation as:

$$\Im = \sum_{i=1}^{n} |\hat{\phi}_{i}^{V} y_{i} - \hat{\phi}_{i}^{L} x_{i}|$$
(A.137)

If  $\Im < \varepsilon$ , the calculation ends as long as the compositions have not been accelerated in the present iteration; if they have, the calculation continues.

- 7. In case it is applicable, the damping factor to be used in Wegstein's method is calculated.
- 8. If the maximum number of iterations allowed is reached, the calculation is terminated. Otherwise the cycle is repeated by executing Step 1 again.

#### A.1.1 Limit for the phase ratios

During the LL (or VL) flash calculation the phase ratio  $\beta$  (or  $\alpha$ ) calculated in the inner cycle (Newton-Raphson method) can be outside the interval [0,1]. Depending on the values of  $K_{i_L}$  and  $\beta$  (or of  $K_{i_V}$  and  $\alpha$  for the VL case), it is possible that the compositions calculated with equation (A.132) be negative, which should be avoided. A way to overcome this problem could be to set to zero any negative composition, but this can lead to divergence in the calculation. The proposed method consists in equating the denominator of equation (A.132) to zero, detect for which components the values of  $\beta$ are greater or smaller and as a function of these values limit  $\beta$  such that all compositions calculated are positive. This procedure is only applied in the cases where the phase ratio is < 0 or > 1. For the LL calculation the strategy is as follows:

1. If  $\beta < 0$ , the greatest value of  $K_{i_L}$  is selected and with it  $\beta_{min}$  is obtained from

$$\beta_{min} = 1/(1 - K_{i_L max})$$
(A.138)

This is the minimum value for  $\beta$  as long as it is negative. If  $\beta$  was less than  $\beta_{min}$ , then at least for one component negative mole fractions would be calculated, therefore the value of  $\beta$  is taken as  $\beta_{min}/2$ . On the contrary, if  $\beta \ge 0$  (which occurs if all  $K_{i_L}$  are < 1), then there is no lower limit for  $\beta$ , but anyhow it is bounded so it does not take values less than  $-\beta_{min}$ .

2. If  $\beta > 1$ , the smallest value of  $K_{i_L}$  is chosen and  $\beta_{max}$  is calculated as

$$\beta_{max} = 1/(1 - K_{i_L min})$$
(A.139)

As long as it is positive, this is the maximum value for  $\beta$ . If all  $K_{i_L}$  are greater than one, then no upper limit for  $\beta$  exists and it is bounded to  $1 - \beta_{max}$ . If  $\beta$  was greater than  $\beta_{max}$ , also negative mole fractions would be obtained, thus  $\beta$  is set to  $(1 + \beta_{max})/2$ .

The value of  $\beta$  obtained in this form is used then to calculate the new compositions. The procedure for the VL case is identical, except that  $\beta$  is substituted by  $\alpha$  and  $K_{i_L}$  by  $K_{i_V}$ .

#### A.1.2 VL flash initialization

At the temperature and pressure of the system, the compositions of the vapour and liquid are obtained according to equations (3.85) and (3.86). With these compositions the equilibrium constants are calculated with the equation of state and the initial vapour fraction  $\alpha$  is computed from

$$\Gamma_V(\alpha, 0) = \sum_{i=1}^n \frac{z_i(1 - K_{i_V})}{1 + \alpha(K_{i_V} - 1)} = 0$$
(A.140)

using a Newton-Raphson procedure with the starting value of  $\alpha$  estimated in a similar fashion as proposed by Ohanomah and Thompson [37]. If the calculated value of  $\alpha$  is greater than 0.95 it is limited to this value and if less than 0.05 it is fixed to this limit.

## A.1.3 LL flash initialization

The initialization of compositions for the two immiscible liquids is based on the selection of two key components as described by Prausnitz et al. [38]. Several modifications are suggested in [5] to make this method more flexible and efficient. The algorithm consists on determining the two components that will act as 'solvents' in each phase. A composition of 0.98 (in mole fraction) is assigned to the key component of phase I in this phase and of 0.02 in phase II and similarly for the key component of phase II. If both key components are already known, only the compositions are assigned. If one of the key components is known (arbitrarily assigned to liquid II), the algorithm finds the key component for the other phase (liquid I) and then the compositions are assigned.

When one key component is known, the other is selected by considering the rest of the components with feed composition > 0.1 as possible key components, one at a time. Each time the equilibrium constants are calculated considering compositions as described above. When the key components for both phases must be determined a similar procedure is applied, but the equilibrium constants are calculated considering all the binary pairs of components with feed composition > 0.1 (contrary to the method described by Prausnitz et al. [38], the order of the binary pairs is considered unimportant, which saves the calculation of n-1 equilibrium constants). The key component selected for a given phase is the one that gives the largest equilibrium constant or its inverse. This means that its fugacity coefficients in both phases are very different and thus has a tendency to pass from the phase with large fugacity to that in which the fugacity is small, acting as solvent in the later phase. Another modification made to the original algorithm of Prausnitz et al. is that when neither key component is known, the one for phase II can be selected with what has been called by Molina and Romero [5] the 'composition criterion'. It consists on assigning as key component for this phase the most abundant component in the feed, since very likely it will act as solvent for one of the phases (when two or more components have the same composition and greater than that of the rest of the components this criterion is not valid). The advantage of this criterion is that the selection of the key component is made dependent on the feed composition, which permits the description of systems with more than one immiscible pair. This also saves a considerable amount of CPU time since only one key components needs to be

determined, particularly when the number of components in the system is large. Also, contrary to the method suggested by Prausnitz et al., which does not consider the case where only one component has a feed composition > 0.1, in the proposed method [5] such component is taken as key component and the other is determined among the rest in the fashion described above, but without the feed composition restriction.

The estimation of the initial value of  $\beta$  is done by means of the mass balance as suggested by Ohanomah and Thompson in [39], considering that phase I has  $z_{k2}K_{k2_L}$ moles of key component 2 and phase II has  $x_{k1}/K_{k1_L}$  moles of key component 1 (where subscripts k1 and k2 indicate the index number of key components 1 and 2, respectively). Thus, the initial value of  $\beta$  is obtained as the mole numbers of key components 1 and 2 in phase I divided by the total moles of these key components in the feed, that is

$$\beta_{init} = \frac{x_{k1}^I + x_{k2}^I}{x_{k1}^I + x_{k1}^{II} + x_{k2}^I + x_{k2}^{II}} \tag{A.141}$$

which can be expressed as

$$\beta_{init} = \frac{z_{k1}(1 - 1/K_{k1_L}) + z_{k2}K_{k2_L}}{z_{k1} + z_{k2}}$$
(A.142)

## A.2 Isothermal VLL flash

In an isothermal VLL flash a feed stream of composition z is separated at constant temperature and pressure into three phases in equilibrium (two liquids and a vapour) or only in two (a vapour and a liquid or two liquids) depending on the nature of the components, the value of z, and the temperature and pressure. The flash problem consists of finding the number and type of phases as well as their amount and composition. The data, unknowns and equations necessary to solve the flash are:

Data: T, P, z

Unknowns:

 $\alpha = V/F, \ \beta = L_I/F, \ y_i, \ x_i^I, \ x_i^{II}$ 

number of unknowns=3n + 2

**Equations:** 

a) Mass balance

 $\alpha y_i + \beta x_i^I + (1 - \alpha - \beta) x_i^{II} = z_i$ 

b) Equilibrium relations

$$f_i^{L_{II}} = f_i^V$$
$$f_i^{L_{II}} = f_i^{L_I}$$

c) Additional relations

$$\sum_{i=1}^n x_i^{II} - \sum_{i=1}^n y_i = 0$$

$$\sum_{i=1}^{n} x_{i}^{II} - \sum_{i=1}^{n} x_{i}^{I} = 0$$

number of equations = 3n + 2

where V,  $L_I$ ,  $L_{II}$  and F are the mole numbers of the vapour, liquid I, liquid II and the feed;  $y_i$ ,  $x_i^I$ ,  $x_i^{II}$  and  $z_I$  are the corresponding mole fractions and f the fugacities in the indicated phases.

Expressing the additional mass balance relations as a function of the feed composition, of the equilibrium constants and of the phase ratios, equations (3.115) and (3.116) can be

obtained. Since the flash algorithm is designed in such way that phases can be eliminated, it can be reduced to a VL or LL calculation. This algorithm consists of the following steps:

#### Three-phase flash algorithm

- 1. With the estimated values of the compositions of the three phases the initial values of the equilibrium constants are calculated.
- 2. Start of the iterative cycle. All the LL equilibrium constants  $(K_{i_L})$  are tested to see if they are close to 1. If so, the compositions of the two liquids are very similar and the calculation is reduced to a VL separation (this is done to avoid convergence problems in the calculation of  $\alpha$  and  $\beta$  in the next step).
- 3. The phase ratios  $\alpha$  and  $\beta$  are obtained solving simultaneously equations (3.115) and (3.116) with a Newton method (inner iterative cycle) for non-linear equations. The partial derivatives necessary are calculated analytically. If during the inner iterative cycle the determinant of the system (for the derived linear system of equations) is close to zero, the calculation continues outside of the inner loop with the values of  $\alpha$  and  $\beta$  obtained with Newton's method at that point. This problem can occur when the initial composition values are very far from the solution. Also, if  $\alpha$  and  $\beta$ do not converge in 15 iterations in the inner loop, the calculation continues in the next step with the values in this last iteration.
- 4. If after the third global iteration  $\alpha > 1$ ,  $\beta < 0$  and  $1 \alpha \beta < 0$  (which indicates that at least one of the liquid phases must be eliminated), it is determined which liquid phase has the greatest tendency to disappear. This phase is the one for which the sum of fugacities for all components is larger.

- 5. If any of the phase ratios is < 0 or > 1, its value is limited to avoid the calculation of negative compositions. If a phase has been eliminated, the corresponding phase ratio is limited as explained for the two-phase flash. In the case of a three-phase calculation  $\alpha$  is limited as in a VL separation. With the value of  $\alpha$  obtained, the limiting value of  $\beta$  is calculated and with it the final value of  $\beta$  is determined such that all mole fractions are positive.
- 6. Test for the elimination of any of the phases. This is done when any of the phase ratios ( $\alpha$  or  $\beta$ ) or  $1 \alpha \beta$  is < 0 in five consecutive iterations, in which case the corresponding phase is removed (when one of the liquid phases is eliminated the other is labeled as liquid II). Also, if any of the variables just mentioned is > 1 in five consecutive iterations, the calculation terminates indicating that only one phase exists.
- 7. The new compositions are calculated according to equations (3.117) to (3.119) and are then normalized as indicated in equations (3.120) to (3.122). If any of the phases has been eliminated, its composition does not need to be obtained.
- 8. If applicable, Wegstein's method is used to accelerate the composition of the phases present.
- 9. The fugacity coefficients of all components in the existing phases and the corresponding equilibrium constants are obtained.
- 10. The objective function is calculated from

$$\Im = \sum_{i=1}^{n} |\hat{\phi}_{i}^{L_{I}} x_{i}^{I} - \hat{\phi}_{i}^{L_{II}} x_{i}^{II}| + \sum_{i=1}^{n} |\hat{\phi}_{i}^{V} y_{i} - \hat{\phi}_{i}^{L_{II}} x_{i}^{II}|$$
(A.143)

If the calculation has been reduced to a LL or VL flash the objective function is obtained from equations (A.136) or (A.137). In the case that  $\Im < \varepsilon$ , the computation terminates as long as compositions have not been accelerated in the present iteration. Otherwise the method continues.

- 11. If required, the damping factor to be used in Wegstein's method is calculated.
- 12. The sequence is repeated by going back to Step 2. If the maximum number of iterations are reached without convergence, the computation is terminated.

#### A.2.1 Limit for the phase ratios

For the case of three phases,  $\alpha$  is limited as for the VL flash. With  $\alpha$  the limiting value of  $\beta$  is obtained and with it a new value of  $\beta$  is calculated such that the mole fractions derived are positive, according to the following scheme:

1. Calculate the limiting value of  $\beta$  for all components as

$$\beta_{i\,lim} = \frac{-\alpha(K_{iv} - 1) - 1}{K_{iv} - 1} \tag{A.144}$$

- 2. If  $\beta$  (calculated in Newton's method) is  $\geq 0$ , the smallest value of  $\beta_{i\,lim}$  which is also greater than  $\beta$  is selected.
- 3. If  $\beta < 0$ , the greatest negative value of  $\beta_{i \, lim}$  which is also greater than  $\beta$  is selected.
- 4. The new value of  $\beta$  to be used to obtain the new compositions is calculated as follows:

If the calculation has been reduced to a LL or VL flash the objective function is obtained from equations (A.136) or (A.137). In the case that  $\Im < \varepsilon$ , the computation terminates as long as compositions have not been accelerated in the present iteration. Otherwise the method continues.

- 11. If required, the damping factor to be used in Wegstein's method is calculated.
- 12. The sequence is repeated by going back to Step 2. If the maximum number of iterations are reached without convergence, the computation is terminated.

#### A.2.1 Limit for the phase ratios

For the case of three phases,  $\alpha$  is limited as for the VL flash. With  $\alpha$  the limiting value of  $\beta$  is obtained and with it a new value of  $\beta$  is calculated such that the mole fractions derived are positive, according to the following scheme:

1. Calculate the limiting value of  $\beta$  for all components as

$$\beta_{i\,lim} = \frac{-\alpha(K_{iv} - 1) - 1}{K_{iv} - 1} \tag{A.144}$$

- 2. If  $\beta$  (calculated in Newton's method) is  $\geq 0$ , the smallest value of  $\beta_{i\,lim}$  which is also greater than  $\beta$  is selected.
- 3. If  $\beta < 0$ , the greatest negative value of  $\beta_{i \, lim}$  which is also greater then  $\beta$  is selected.
- 4. The new value of  $\beta$  to be used to obtain the new compositions is calculated as follows:

If 
$$\beta > 0$$
 and  $1 \le \beta_{lim} < \beta$  then  $\beta = 1 + (\beta_{lim} - 1)/2$   
If  $\beta > 0$  and  $1 > \beta_{lim} < \beta$  then  $\beta = \beta_{lim} - 0.005$   
If  $\beta < 0$  and  $\beta < \beta_{lim}$  then  $\beta = \beta_{lim}/2$ 

## A.2.2 VLL flash initialization

The initial compositions of the two liquids are obtained with the key component criterion described for the LL flash initialization. The vapour composition is estimated considering it as an ideal gas using equation (3.85). With respect to the phase rations  $\alpha$  and  $\beta$ , the value of  $\beta_{init}$  is fixed to 0.5 and with this value  $\alpha_{init}$  is calculated considering the mass balance for key component 1, that is,

$$\alpha_{init} = \frac{z_{k1} - \beta_{init} x_{k1}^{I} - (1 - \beta_{init}) x_{k1}^{II}}{y_{k1} - \beta_{init} x_{k1}^{I} - (1 - \beta_{init}) x_{k1}^{II}}$$
(A.145)

and then a better estimate of  $\beta_{init}$  is obtained as

$$\beta_{init} = 0.5(1 - \alpha_{init}) \tag{A.146}$$

## A.3 Wegstein's method

When the equilibrium constants are strongly dependent on the phase compositions, the equilibrium calculations can be extremely slow. In such cases it is convenient to accelerate these variables. Wegstein's method, applied for this purpose, consists on extrapolating linearly the compositions of each component using the two latest composition values available. The extrapolation is done in such a way as to make the objective function zero. The new composition variables z (which in this case can be either the liquid or vapour compositions, not the feed) are obtained as

$$z_{i\,new}^{(k)} = z_i^{(k)} + c(z_i^{(k)} - z_i^{(k-1)})$$
(A.147)

where k indicates the iteration number and c is the acceleration or damping factor according to its value:

c > 0	acceleration		
c = 0	direct substitution		
-1 < c < 0	$\mathbf{damping}$		
$c \leq -1$	deceleration		

In the method proposed by Prausnitz et al. [38] c is obtained as

$$c = \frac{\Im^{(k-1)}}{\Im^{(k-2)} - \Im^{(k-1)}}$$
(A.148)

as long as  $\Im^{(k-2)} > \Im^{(k-1)}$ , in such form that c is always > 0 and thus there is only possibility to accelerate. If  $\Im^{(k-2)} < \Im^{(k-1)}$  these authors use direct substitution. In [5] it was found that for LL calculations near the plait point the objective function increased when accelerating compositions with c calculated with equation (A.148) and convergence was not achieved. With direct substitution convergence was reached in these cases but in a large number of iterations. However, with values of c < 0 the number of iterations was reduced significantly if c was fixed between 0 and -0.6. In these cases of difficult convergence a strategy that allowed to dampen the calculation as required was developed in [5] and was incorporated into Wegstein's method. The sequence of this scheme is:

- 1. Calculation of the new compositions in iteration k and of the phase ratios  $\alpha$  and/or  $\beta$ .
- When conditions are satisfied for Wegstein's method as described by Prausnitz et al. [38], the new compositions are calculated with equation (A.147) where c is obtained from equation (A.148).
- 3. The fugacity coefficients and equilibrium constants are calculated with the compositions obtained in Step 1 or in Step 2 when applied.
- 4. Calculation of the objective function for the corresponding type of equilibrium and convergence test.
- 5. If  $\mathfrak{T}^{(k)} > \mathfrak{T}^{(k-1)}$  then c is calculated as

$$c = \frac{\Im^{(k-1)} - \Im^{(k)}}{\Im^{(k)}}$$
(A.149)

If c is < -0.6, it is limited to this value to avoid a slow progress. If > -0.2 it is considered that the increase in the objective function is not large enough to require damping. When c is between these limits new compositions are obtained according to equation (A.147) and the updated value of the objective function in iteration k is calculated. Once damping has been applied, the method keeps damping with the specified frequency of application of Wegstein's method until  $\Im$  becomes less than a pre-specified value to avoid divergence in the calculation. After this Wegstein's method is applied in the normal way and a new iteration is started by executing Step 1.

# **Appendix B**

## Activity coefficient models in the phase-splitting algorithm

When the equilibrium calculations involve polar compounds at low or moderate pressures, the liquid phase interactions can be better described with an activity coefficient model than with cubic equations of state. In such cases the vapour phase properties can be adequately obtained with the virial equation of state (usually truncated after the second term). At the present level of understanding of fluid phase molecular behaviour, the best practical pathway to obtain activity coefficients is to employ semi-empirical correlations that relate these variables to the temperature and the liquid phase composition through interaction parameters determined from equilibrium data. The predictive correlations for the activity coefficient  $\gamma_i$  include the Wohl-type equations (e.g. the Margules, van Laar and Scatchard- Hamer equations), the Redlich-Kister equation and the models based on the concept of local composition, such as Wilson, NRTL, UNIQUAC and UNIFAC.

This approach is useful for multicomponent mixtures because in many cases, to a good approximation, extensions from binary to higher systems can be easily made. Nevertheless, since the activity coefficients dependence on pressure is usually not taken into account, the range of application of these models is limited to low or moderate pressures, far from critical regions. Also, since a standard state fugacity is needed for each component, care must be exercised because this standard state is defined in a different form depending on whether such component is condensable or non- condensable at the specified temperature and pressure (where the boundary between condensable and non-condensable is itself arbitrary). Although activity coefficient models are not used in this work, some considerations with regard to their use in the phase-splitting algorithm proposed are discussed here.

## **B.1** Activities and standard states

When an activity coefficient model is used for the liquid phase, the activity is defined as

$$\hat{a}_{i}^{L} = \frac{\hat{f}_{i}^{L}}{f_{i}^{o}} = \frac{x_{i}\gamma_{i}f_{i}^{o}(P_{r})}{f_{i}^{o}}$$
(B.150)

where  $\gamma_i$  is the activity coefficient of component *i* at the system's temperature *T* and pressure *P*,  $f_i^o(P_r)$  the standard state fugacity at *T* and a reference pressure  $P_r$  and  $f_i^o$ the standard state fugacity at *T* and *P*. According to Prausnitz et al. [38],  $\gamma_i$  can be obtained as

$$\gamma_i = \gamma_i(P_r) \exp \int_{P_r}^{P} \frac{\bar{v}_i^L}{RT} dP$$
(B.151)

where  $\gamma_i(P_r)$  is the activity coefficient at the reference pressure and  $\bar{v}_i^L$  is the partial molar liquid volume. This last equation is introduced to evaluate all isothermal activity coefficients at the same reference pressure in order to use the isothermal-isobaric form of the Gibbs-Duhem equation to correlate experimental activity coefficient data. This has the indirect effect of making the activity coefficients 'independent' of the pressure. Thus, substituting equation (B.151) in (B.150), the activity in the liquid phase can be expressed as

$$\hat{a}_{i}^{L} = \frac{x_{i}\gamma_{i}(P_{r})\{\exp[\bar{v}_{i}^{L}(P - P_{r})/RT]f_{i}^{o}(P_{r})\}}{f_{i}^{o}}$$
(B.152)

 $\gamma < \epsilon$ 

Considering that the term in curly brackets corresponds to  $f_i^o$  for a condensable component, this equation becomes

$$\hat{a}_i^L = x_i \gamma_i(P_r) \tag{B.153}$$

Then  $\gamma_i(P_r)$  can be obtained from any activity coefficient model and is pressure independent. This expression is analogous to equation (3.32) obtained in Section 3.1. Thus, when an activity coefficient model is used, the reference state fugacity  $f_i^o$  is taken as the fugacity of the pure liquid *i* at the temperature and pressure of the mixture.

To be consistent with this reference state, when using an equation of state for the vapour phase the reference state must be the pure component i at T and P in the same state as the mixture, that is, as a vapour. The activity of i in the vapour can then be obtained, according to equation (3.30), as

$$\hat{a}_i^V = \frac{y_i \hat{\phi}_i^V}{\phi_i^V} \tag{B.154}$$

where  $\hat{\phi}_i^V$  is the fugacity coefficient of *i* in the mixture and  $\phi_i^V$  the fugacity coefficient of pure *i* (both evaluated at *T* and *P*). This is the analogous to equation (3.31).

After making these considerations, the dimensionless Gibbs energies of mixing for the vapour and liquid phases, corresponding to equations (3.33) and (3.34), can be expressed respectively as:

$$\Delta g^{V}/RT = \sum_{i=1}^{n} y_{i} \ln[\frac{y_{i}\hat{\phi}_{i}^{V}(T, P, y)}{\phi_{i}^{V}(T, P)}]$$
(B.155)

$$\Delta g^{L}/RT = \sum_{i=1}^{n} x_{i} \ln[x_{i} \gamma_{i}(P_{r})] = g^{E}/RT + \sum_{i=1}^{n} x_{i} \ln(x_{i})$$
(B.156)

where  $g^E$  is the excess Gibbs energy given by the activity coefficient model.

# **B.2** Vapour and liquid phases search

When a split model is used (i.e. using an equation of state to obtain the vapour fugacity and an activity coefficient model for the liquid), the search for an incipient vapour is conducted in the same way as described in Section 3.2.1, except that parameters  $\exp(h_i^L)$ should be obtained from

$$\exp(h_i^L) = \frac{z_i \gamma_i(z) f_i^o}{P} \tag{B.157}$$

instead of using equation (3.37). For the additional vapour phase search also an analogous procedure can be followed as indicated in Section 3.2.1, but instead of using equation (3.55) to obtain the pseudo-parameter  $h_{p_i}^L$ , the following expression is used

$$h_{p_i}^L = \ln z_i + \ln \gamma_{p_i}(z)$$
 (B.158)

where the pseudo-activity coefficient at the feed composition  $\gamma_{p_i}(z)$  can be obtained as

$$\ln \gamma_{p_i}(z) = \frac{\Delta \hat{\mu}_{p_i}^L}{RT} - \ln z_i \tag{B.159}$$

Similarly, the liquid phase search can be conducted as indicated in Section 3.2.2, but the stationarity conditions are defined as

$$\ln X_i + \ln \gamma_i(x) - h_i^L = 0$$
 (B.160)

and

$$\ln X_i + \ln \gamma_i(x) - h_i^V = 0$$
 (B.161)

which replace equations (3.60) and (3.61). Also, the following equation is used instead of (3.63)

$$\exp(h_i^L) = z_i \gamma_i(z) \tag{B.162}$$

In addition, instead of using fugacity coefficients  $\hat{\phi}_{j,i}^{L(k)}$  evaluated from an equation of state, activity coefficients  $\gamma_{j,i}^{(k)}$  evaluated with an activity coefficient model are employed throughout the calculation (i.e.  $\gamma_{j,i}^{(k)}$  replace  $\hat{\phi}_{j,i}^{L(k)}$  in all instances). For the additional liquid phase search, the value of  $\exp(h_i^L)$  used is that defined with equation (B.162), otherwise the method is identical as the one described in Section 3.2.2.

## **B.3** Liquid phase initialization

The necessary changes required to conduct the initialization of the liquid compositions with Shah's method, as well as with the modified Gautam and Seider and the proposed method are also straightforward. The calculation of the pseudo-infinite dilution activity  $\hat{a}_{p_i}^{L^{\infty}}$  required to select phase initiator p is done according to

$$\hat{a}_{p_i}^{L^{\infty}} = z_i \gamma_i^{\infty} \tag{B.163}$$

where  $\gamma_i^{\infty}$  is the infinite-dilution activity coefficient for component *i*. This equation replaces equation (3.88). The selection of phase initiator *q* is done as described for Shah's method but the activity coefficient at infinite dilution for the binary mixture is used instead of the corresponding fugacity coefficient. Then equations (2.3) and (2.4) are used as explained in Section 2.2.1 (which replace equations (3.89) and (3.90)) to obtain the compositions of components *p* and *q*. The solution to these equations can then be obtained by substituting equations (3.97) to (3.102) by equations (B.164) to (B.169) respectively:

$$A = 1 - \gamma_p^{II^{\infty}} \tag{B.164}$$

$$B = z_p (\gamma_p^{II^{\infty}} - 2) \tag{B.165}$$

$$C = z_p \gamma_p^{II^{\infty}} \tag{B.166}$$

$$D = 1 - \gamma_a^{I^{\infty}} \tag{B.167}$$

$$E = 2z_p \gamma_q^{I^{\infty}} - z_q \tag{B.168}$$

$$F = -z_q \tag{B.169}$$

To solve for the composition of the rest of the components (if n > 2), equation (2.7) replaces (3.104) and (3.107) to (3.109) are replaced by (B.170) to (B.172), which are:

$$d = \gamma_r^{I^{\infty}} - \gamma_r^{II^{\infty}} \tag{B.170}$$

$$e = \gamma_r^{II^{\infty}}(z_r - n_p^I - n_q^I) - \gamma_r^{I^{\infty}}(z_r + n_p^{II} + n_q^{II})$$
(B.171)

$$f = z_r \gamma_r^{II^{\infty}} (n_p^I + n_q^I) \tag{B.172}$$

The rest of the calculation can then be followed as indicated in Section 3.2.2 and similarly for the modified Gautam and Seider and the proposed methods.

# **Appendix C**

## Description and listings of the computer programs

The description and listings of all the programs and subroutines written for the phasesplitting and flash calculations are presented in this appendix (in Fortran 77 language). For each program an explanation of its purpose and main variables is given; also comments have been written throughout the listings to make them as self-explanatory as possible.

The main program for the phase-splitting predictions, program PHASPLIT, reads all necessary input parameters from a data file (in free format) with the following structure:

```
'SYSTEM NO.: A(1)-B(2)- ... N(n)'

IFLG(0),N,ITMAX,EPS,IWEG,TIMAX

IFTP,T,P,QIN,QFN,DELQ

INITL,IFLG(15),INOLC,ISTA,ISRCH

EPSF,IFLG(7),IWEGF,IFLG(8)

Z(1),TC(1),PC(1),W(1),PP(1)

...

Z(n),TC(N),PC(n),W(n),PP(n)

KIJ(1,2)

KIJ(1,3)

...

KIJ(1,n)

KIJ(2,n)

...

KIJ(2,n)
```

where all the variables are described in the program listing.

Similarly, for the direct flash calculations (either two- or three-phase), the necessary input data for program PHASEQ should be given according to the following format:

As an example of an input file for program PHASPLIT, the following data file corresponds to the equilibrium calculation for System 3 at 430 K and 30 atm using the defalut values for the different options and parameters for the phase-splitting algorithm as recommend in Chapter 5.

```
'SYSTEM 3: C3H8(1)-nC4H10(2)-nC5H12(3)-nC6H14(4)-nC8H18(5)-H2O(6)'
                    5.0D0
1
  6 50
         1.D-3 2
                    5.D0
   430.0D0
            30.DO
                            40.D0
0
                                    5.D0
4
  0
      0 2
            1
0.00001D0
              2
           1
                 1
           369.8D0
0.16667D0
                      41.9D0
                              0.152D0
                                       0.D0
                              0.193D0
                                       0.D0
0.16667D0
           425.2D0
                      37.5DO
0.2000D0
           469.6D0
                      33.3D0
                              0.251D0
                                       0.D0
0.06667D0
           507.4D0
                      29.3D0
                                       0.D0
                              0.296D0
0.13333D0
           568.8D0
                      24.5D0
                              0.394D0
                                       O.DO
0.26667D0
           647.3D0
                    217.6D0
                              0.344D0
                                       0.1277D0
0.D0
O.DO
O.DO
0.D0
0.48D0
0.D0
0.D0
0.D0
0.48D0
0.D0
0.D0
0.48D0
```

186

```
0.D0
0.48D0
0.48D0
```

The file for the equilibrium calculation with the direct three-phase flash, read by program PHASEQ, is:

```
'SYSTEM 3: C3H8(1)-nC4H10(2)-nC5H12(3)-nC6H14(4)-nC8H18(5)-H2O(6)'
  6 50 0.00001D0 1 2 1
1
  0 430.0D0
              30.DO
                      5.D0
                             40.D0
                                     5.D0
3
1 0 0 0 0 0.D0 0.D0
0.16667D0 369.8D0
                    41.9D0
                            0.152D0
                                    0.D0
                    37.5D0
0.16667D0
          425.2D0
                            0.193D0
                                    0.D0
0.2000D0
          469.6D0
                    33.3D0
                            0.251D0
                                     0.D0
0.06667D0
          507.4D0
                    29.3D0
                            0.296D0
                                     0.D0
0.13333D0 568.8D0
                    24.5D0
                            0.394D0
                                    0.D0
0.26667D0 647.3D0
                   217.6D0 0.344D0 0.1277D0
0.D0
0.D0
0.D0
0.D0
0.48D0
0.D0
O.DO
O.DO
0.48D0
0.D0
0.D0
0.48D0
0.D0
0.48D0
0.48D0
```

The corresponding output from the phase-splitting prediction obtained with program PHASPLIT is

```
SYSTEM 3: C3H8(1)-nC4H10(2)-nC5H12(3)-nC6H14(4)-nC8H18(5)-H2O(6)

PENG-ROBINSON EOS

Z(1)= 0.16667

Z(2)= 0.16667

Z(3)= 0.20000

Z(4)= 0.06667

Z(5)= 0.13333

Z(6)= 0.26667

MAX. NO. ITERATIONS= 50 CONVERGENCE= 0.10000E-02

WEGSTEIN METHOD EACH 2 ITER TIMAX= 5.000
```

T= 430.000 K P= 30.000 atm

```
DEL G VAP= 1163.4519 cal/gmol
                                       Z VAP= 0.6388
                                      Z VAF- ...
Z LIQ= 0.3000
    DEL G LIQ= 1444.4997 cal/gmol
                                   *** EXTRAPOLATED Z LIQ ***
THE SYSTEM IS VAPOUR-LIKE
 CPU TIME=
            2 msecs
SUBROUTINE INLIG RESULTS
  IP= 6 IQ= 5
    0.00054 0.00013 0.00002 0.00000 0.00000 0.99930
    0.20393 0.20402 0.24486 0.08162 0.16324 0.10233
    0.01257 0.03109 0.08380 0.05987 0.45872 0.35395
CPU TIME=
            4 msecs
SUBROUTINE LIQSER RESULTS
  UNSTABLE SYSTEM. TWO LIQUID PHASES FOUND
     \begin{array}{c} \textbf{x1(1)=0.000000} \quad \textbf{x2(1)=0.072379} \\ \textbf{x1(2)=0.000000} \quad \textbf{x2(2)=0.115358} \\ \textbf{x1(3)=0.000000} \quad \textbf{x2(3)=0.210365} \\ \end{array} 
    x1(4) = 0.000000 x2(4) = 0.105158
  x1(5)= 0.000000 x2(5)= 0.443579
x1(6)= 1.000000 x2(6)= 0.053161
NO. OF ITERATIONS= 4
CPU TIME= 14 msecs
SUBROUTINE TESTA RESULTS
 +++ STABLE SYSTEM AS VLL +++
CPU TIME= 16 msecs
VLL FLASH INITIALIZATION T= 430.000 K P= 30.000 atm
                    V/F= 0.35542 L1/F= 0.16964
           Z
                     Y
     Т
                            X1
                                          12
       0.16667 0.26991 0.00000 0.14894
0.16667 0.21653 0.00000 0.18889
     1
     2
     3 0.20000 0.20334 0.00000 0.26893
       0.06667 0.05128 0.00000 0.10200
     5
        0.13333 0.05652 0.00000 0.23843
     6 0.26667 0.20243 1.00000 0.05281
CPU TIME= 36 msecs
WEGSTEINS METHOD EACH 2 ITERATIONS WITH DAMPING
MAXIMUM NO. OF ITERATIONS= 50 CONVERGENCE= 0.00001
           VLL FLASH
                        CALCULATION RESULTS
T= 430.000 K P= 30.000 atm NUM. ITER.= 2 OBJ. FUNC.= 0.42037E-05
V/F= 0.35536 L1/F= 0.16963 L2/F= 0.47501 DEL G=
ZVAP= 0.70478 ZLIQ1= 0.02052 ZLIQ2= 0.14416
                                                        1099.786 cal/gmol
                                               0.14416
KEY COMP.#1= 6 KEY COMP.#2= 5
                    Y
                             X1
                                                KV=Y/X2
    Ι
          Ζ
                                       X2
                                                              KL = X1/X2
      0.16667 0.26992 0.00000 0.14895
                                               1.812
                                                             0.2032E-05
    1
       0.16667
                                                1.146
                                                             0.1008E-06
    2
    3
                                               0.7560
                                                             0.3328E-08
    4 0.06667 0.05127 0.00000 0.10200
                                               0.5026
                                                             0.5728E-10
    5 0.13333 0.05651 0.00000 0.23841
                                               0.2370
                                                            0.3333E-13
    6 0.26667 0.20247 1.00000 0.05282 3.833
                                                              18.93
```

```
CPU TIME= 6 msecs
TOT CPU TIME= 42 msecs
GLOBAL CPU TIME= 42 msecs
```

The output from program PHASEQ for the same conditions is:

```
SYSTEM 3: C3H8(1)-nC4H10(2)-nC5H12(3)-nC6H14(4)-nC8H18(5)-H2D(6)
  PENG-ROBINSON EOS
                               VLL FLASH
  WEGSTEINS METHOD EACH 2 ITERATIONS WITH DAMPING
  MAXIMUM NO. OF ITERATIONS= 50 CONVERGENCE= 0.00001
  AUTO-INITIALIZATION
      K(1,6)=0.4800
K(2,6)=0.4800
K(3,6)=0.4800
K(4,6)=0.4800
      K(5, 6) = 0.4800
  VLL FLASH INITIALIZATION T= 430.000 K P= 30.000 atm
                              V/F= 0.74823 L1/F= 0.12589
                                 X1
       Τ
              7.
                       Y
                                           X2
         0.16667 0.56751 0.00000 0.00000
       1
          0.16667 0.22938 0.00000 0.00000
       2
          0.20000 0.12253
                              0.00000 0.00000
       3
                   0.01906 0.00000 0.00000
          0.06667
       4
         0.13333 0.00995 0.98000 0.02000
       Б
          0.26667 0.05158 0.02000 0.98000
       6
  CPU TIME= 27 msecs
              VLL FLASH CALCULATION RESULTS
  T= 430.000 K P= 30.000 atm NUM. ITER.= 6 OBJ. FUNC.= 0.18390E-06 V/F= 0.35536 L1/F= 0.47502 L2/F= 0.16963 DEL G= 1099.786 cal/gmol
  ZVAP= 0.70479 ZLIQ1= 0.14416 ZLIQ2=
                                                 0.02052
  KEY COMP. #1= 5 KEY COMP. #2= 6
                                         X2
                                                  KV=Y/X2
                                                                KL = X1/X2
                      Y
                                X1
      1 0.16667 0.26992 0.14895 0.00000
                                                 0.8918E+06
                                                               0.4921E+06
```

	2	0.16667	0.21652	0.18889	0.00000	0.1137E+08	0.9919E+07
	3	0.20000	0.20332	0.26893	0.00000	0.2271E+09	0.3005E+09
	4	0.06667	0.05127	0.10200	0.00000	0.8774E+10	0.1746E+11
	5	0.13333	0.05651	0.23841	0.00000	0.7111E+13	0.3000E+14
	6	0.26667	0.20247	0.05282	1.00000	0.2025	0.5282E-01
CPU	TI	ME= 22 m	secs				

TOT CPU TIME= 49 msecs

GLOBAL CPU TIME= 49 msec

```
******
             PROGRAM PHASPLIT (PHASE-SPLITTING PREDICTION)
**********
* Purpose:
 Main program to determine the stability of a global homogeneous
* mixture of n components at a given temperature, pressure and
* composition. Calculations can be done at a fixed T, P and comp.;
* at fixed T, comp. and different pressures; at fixed P, comp. and
* different temperatures or at fixed T, P and different compositions *
 for a binary or ternary mixture
*
 Main variables:
*
  -Alphanumeric
           Indicates if damping is used in Wegstein's method when
*
   DAMP
           applied in flash calculations
*
   EOS
           Equation of state used: Soave-Redlich-Kwong or Peng-
           Robinson
   TYPEQ Type of equilibrium: VL, LL or VLL
SYSTEM System's components and index number
*
*
   ZV,ZL1, Indicate if the compressibility factor of a given phase
   ZL2
           has been extrapolated in a flash calculation
  -Numeric
*
   ALPHA
*
           Molar phase ratio V/F
   ALSL
           Vector of the logarithm of the activity of each component
*
*
           in the system as a liquid
   APUR
*
           Vector of pure component dimensionless attraction
*
           parameters in the EOS
   BETA
           Molar phase ratio L1/F
*
   BPUR
*
           Vector of pure component dimensionless repulsion
*
           parameters in the EOS
           Gibbs energy of mixing of the system as a liquid
*
   DELGL
*
   DELGV
           Convergence tolerance in the phase search subroutines
*
   EPS
*
   EPSF
                                            flash subroutines
*
   FUCL1
           Fug. coeff. vector of liq. 1 in the phase search
   FUCL2
*
                                        2
   FUCV
            **
                  ...
                           ..
                                " the vapour in the phase search
*
   FUCD
           Dummy fug. coeff. vector
*
   FUCSL
           Fug. coeff. vector of the system as a liquid
*
   FUCSV
  FUCL1N
           Updated fug. coeff. vector of liq. 1
*
                н
Н
                       ´ ++ ++
++ ++
*
   FUCL2N
                                        •• ••
              ...
                                        " the vapour
*
   FUCVN
*
   HT.
           See subroutine LIQSER
*
   HP
                           PHASER
            ....
                    ....
   HV
                           VAPSER
*
   IFLG
           Vector of calculation's control flags
*
   IFTP
           Indicator of the T, P and comp. calculation conditions:
           O=fixed T, P and comp.; 1=fixed T & comp., loop for
different P; 2=fixed P & comp., loop for different T; 3=
fixed T & P, loop for different comp. for a binary or
*
*
*
           ternary mixture
*
   INITL
           Indicator of the type of initialization subroutine for
           the liquid search: 1=INMICH; 2=INGASE; 3=INSHAH; 4=INLIQ
Indicator for monitoring the complexity of the calctn.:
   INOLC
           O=full calcn. with VAPSER, LIQSER (& PHASER if specified) *
           followed by the corresponding flash; 1=results of all the *
           liq. init. routines without doing any further calcns.; 2= *
```

*		calc using VAPSER & LIGSER without the flash (PHASER not *
*		verd)
÷.	TOTA	Indicator of the 3 nhage stability subrouting used:
Ŧ	TOTA	A-TECTA, 1-VEISTA, 2-TECTA used except then the system is t
Ξ.		U-IESIA; I-RELSIA; Z-IESIA used except when the system is -
Ξ.		Duce Advantage and Advantage and Advantage Advanta
Ŧ.		PRASER detects an additional liquid in which case MELSIA +
*		
*	ISRC	H Indicator for the type of phase search: 0=search without *
*		the additional search; 1=thorough search using subroutine *
*		PHASER *
*	ITMA	X Maximum number of iterations allowed *
*	ITNU	M Iteration number in which convergence is reached *
*	IWEG	Frequency of application of Wegstein's method in VAPSER *
*		or LIQSER subroutines; if zero, Wegstein's method not *
*		used and the solution is obtained by direct substitution *
*	IWEG	F Frequency of application of Wegstein's method in the *
*		flash calculation routines *
*	KEO	Dummy equilibrium constants vector *
*	KV	VL equilibrium constants vector *
*	KL.	
*	KIJ	NxN array of binary interaction parameters in the EOS *
*	MX	Number of initial liquid composition vectors *
*	N	Number of components in the system *
*	P	System's pressure (atm) *
*	PC	Vector of critical pressures *
*	PD	Vector of Withing plasmaters *
*	DVAD	Vaccor of Machinas poral parameters Vanour pressure estimation for each component
*	R	Universal gas constant (1 987 cal/gmole K)
*	T	System's temperature (K)
*	ŤĊ	Vactor of critical temperatures
*	TTMA	Vector of critical temperatures *
*	TTUR	used ing subroutines VAPSER or LINSER
*	u	Votor of Ditaria scontric factors
*		" " initial liquid phase composition using *
*	A . H	Raoult's law *
*	¥Т	Nature & area of initial liquid compositions then routing t
Ŧ	AT.	TWICH is used and 2 K array in all other initializing t
*		cubroutines
*	۳D	Dummy vector of liquid composition
Ŧ	T T	Vector of light composition in the phase search
Ŧ	X.J	
*	AZ VV	Indated vector of single liquid phase composition
-	71W	Updated vector of light 1 composition *
*	YON	
*	ΥT	Vector of initial wanour composition using Raoult's law #
*	v	" " vanour composition the phase search *
*	, vv	lindated vector of vanour composition *
*	7	Vector of global mixture composition (mole fraction) *
*	רוג)	vectors are of dimension N except when otherwise specified) *
*	(844)	*
*	Sveta	m's flags ±
*	i	TFIC(i)
*	ō	
*	1	v-oun doo, I-ru doo T 1=extranolated dengity for the yanour +
*	2	1-extrap density for the lig (lig 1)
*	2	1-extrap. density for the lig. (lig. 2)
Ŧ +	3	-extraption DAD has not been executed. (-it) - her executed to
+	т с	V-SUDIOULING FAD HAS NOT DEEN EXECUTED; 1=11'S DEEN EXECUTED *
¥ 	0	U-vapour trial phase not found; 1=vapour trial phase found *
¥ 	o	U=no 11q. trial phase(s) found; 1=one 11q. trial phase found; *
*	7	Z=TWO 11q. Trial phases Iound *
<b>#</b>	(	U=wegstein's method not used in the Hash Calculations;
*	•	
Ŧ	0	v-no damping used in wegstein's method in the Hash calchs.; *

```
1=damping is used
   9
*
         0-user supplied initial estimates for the flash calculation
*
         1=use of initialization routines (as in ref. [5])
*
   10
        O=sequential initialization not used in the flash;
                                   ...
*
         1=
                                              is
        O=stable system as V; 1=stab. sys. as L; 2=stab. sys. as VL;
*
   11
*
         3=stab. syst. as LL; 4=stab. syst. as VLL
        O=no equilibrium calc. has been done; 1=a calc. has been done
O=calc. of VL or LL eq. cts. when subroutine K is called;
1= " " fug. coeffs. when subroutine K is called
O=initialization for 2 phase eq. calc.; 1=init. for 3 phase
*
   12
*
   13
*
   14
         eq. calc. (as in ref [5])
*
*
   15
        O=no printing of intermediate calcs.; 1=printing of some
        calcs.; 2=full printing of intermediate calcs.
1=extrapolation control for liq. 2 (VLL flash)
0=the original system is vapour-like 1=is liquid-like
*
   16
*
    17
        1=calculation using PHASPLIT to obtain initial values for
*
   18
         the 2 or 3 phase flash unknowns
         O=no additional liquid found by PHASER when the system is
liquid-like, a vapour has been found and no liquids found
   19
         1=additional liquid found by PHASER in the case described
*
  Subroutines called:
*
   K
   VAPSER
   INMICH
   INGASE
    INSHAH
*
   INLIQ
*
   LIQSER
   PHASER
   NELSTA
*
   TESTA
   FLASH2
   FLASH3
   GTBBS
                                                                                            *
                                                                                            *
********
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 KEQ(20), KV(20), KL(20), KIJ(20,20)
       CHARACTER DAMP*16, EOS*25, SYSTEM*72, TYPEQ*10, ZV*8, ZL1*9, ZL2*9
       DIMENSION ALSL(20), APUR(20), BPUR(20), C1(20), C2(20), FUCL1(20),
                     FUCL2(20), FUCV(20), FUCD(20), FUCSL(20), FUCSV(20),
                     FUCL1N(20), FUCL2N(20), FUCL1P(2), FUCL2P(20), FUCVN(20),
      +
                    HL(20), HP(20), HV(20), IFLG(0:20), OBJL(2), OBJLP(2),
PC(20), PP(20), TC(20), W(20), XI(21,20), XIN(20), XN(20),
XD(20), X1(20), X2(20), X1N(20), X2N(20), X1P(20), X2P(20),
      +
                     Y(20), YI(20), YN(20), Z(20), ZED(3)
       ITOTIM=0
        R=1.987D0
       DO 10 I=0,20
          IFLG(I)=0
    10 CONTINUE
        IFLG(18)=1
       DO 30 I=1,20
          DO 20 J=1,20
             KIJ(I,J)=0.D0
          CONTINUE
    20
    30 CONTINUE
```

\* The calculation conditions and variables are read from a data file

```
READ(5,*) SYSTEM, IFLG(0), N, ITMAX, EPS, IWEG, TIMAX,
                    IFTP, T, P, QIN, QFN, QDEL,
      +
      + INITL, IFLG(15), INOLC, ISTA, ISRCH,
+ EPSF, IFLG(7), IWEGF, IFLG(8)
READ(5,*) (Z(I), TC(I), PC(I), W(I), PP(I), I=1, N)
      +
      +
       DO 50 I=1,N
         DO 40 J=I+1,N
            READ(5,*) KIJ(I,J)
            KIJ(J,I) = KIJ(I,J)
         CONTINUE
   40
   50 CONTINUE
* Printing of the calculation conditions
       WRITE(6,600) SYSTEM
  600 FORMAT (//2X,A60)
       IF (IFLG(0) .EQ. 0) THEN
         EOS='SOAVE-REDLICH-KWONG EOS'
       ELSE
         EOS='PENG-ROBINSON EOS'
       END IF
  WRITE(6,610) EOS
610 FORMAT(/4X,A25)
       IF (IFTP .EQ. 3) THEN
         WRITE(6,615) T.P
         FORMAT(/4X,'T=',F8.3,' K',4X,'P=',F8.3,' atm'/)
  615
       ELSE
         WRITE(6,620) (I,Z(I), I=1,N)
FORMAT(8X,'Z(',I2,')=',F8.5)
  620
       END IF
       IF (IFLG(15) .NE. 2) GO TO 80
DO 70 I=1,N
         DO 60 J=I+1,N
            IF (KIJ(I,J) .EQ. 0.D0) GO TO 60
WRITE(6,630) I,J,KIJ(I,J)
FORMAT(8X,'K(',I2,',',I2,')=',F7.4)
  630
         CONTINUE
   60
   70 CONTINUE
   80 WRITE(6,640) ITMAX,EPS
  640 FORMAT (4X, 'MAX. NO. ITERATIONS=', I3, 2X, 'CONVERGENCE=', G12.5)
IF (IWEG .EQ. 0) THEN
          WRITE(6,650)
         FORMAT(4X, 'DIRECT SUBSTITUTION'/)
  650
       ELSE
         WRITE(6,660) IWEG,TIMAX
  660
         FORMAT(4X, 'WEGSTEIN METHOD EACH', I3,' ITER', 2X, 'TIMAX=', F7.3/)
       END IF
       IF (IFTP .EQ. 0) GO TO 97
IF (IFTP .EQ. 3) GO TO 92
* Loop for calculations at fixed T and various P or at fixed P at
* various T
       DO 590 Q=QIN,QFN,QDEL
          IF (IFTP .EQ. 1) THEN
            P=Q
          ELSE
            T=Q
          END IF
          GO TO 97
```

· 7

```
* Loop for calculations at fixed T and P at different compositions
  92 IF (N .GT. 3) THEN
        WRITE(6,665)
  665
        FORMAT(/4X, 'WARNING: CALC. AT FIXED T & P AT DIFFERENT'
                /4X, 'COMPOSITIONS ALLOWED ONLY FOR BINARY OR '
     +
     +
                /4X, 'TERNARY MIXTURES')
        GO TO 590
      END IF
      IF (N .EQ. 2) THEN
        DZ=0.05D0
      ELSE
        DZ=0.1D0
      END IF
      IF (N .EQ. 2) GO TO 95
      DO 580 ZF1=1.D-8,1.DO+DZ,DZ
      DO 570 ZF2=1.D-8,1.D0+DZ,DZ
        Z(1)=ZF1
        Z(2)=ZF2
        Z(3)=1.D0-ZF1-ZF2
        IF (Z(1).GT. 1.D0) Z(1)=1.D0-1.D-8
IF (Z(2).GT. 1.D0) Z(2)=1.D0-1.D-8
IF (Z(3).LE. 0.D0) Z(3)=1.D-8
        IF (ZF1+ZF2 .GT. 1.D0+DZ) GO TO 570
        ZTOT = Z(1) + Z(2) + Z(3)
        Z(1)=Z(1)/ZTOT
        Z(2)=Z(2)/ZTOT
Z(3)=Z(3)/ZTOT
        GO TO 97
   95 DO 565 ZA1=1.D-8,1.D0+DZ,DZ
        Z(1)=ZA1
        Z(2) = 1.D0 - ZA1
        IF (Z(2) .LE. 0.D0) Z(2)=1.D-8
* Calculation of the fugacity coefficients for the vapour-like
* and liquid-like system
   97
        CALL TIME(0,0,ITDUM)
        IFLG(1)=0
        IFLG(2)=0
        IFLG(13)=1
        ITIMT=0
        ITIMC=0
        IP=0
        MX=3
        CALL K(XD,Z,TC,PC,W,PP,P,T,N,1,1,APUR,BPUR,FUCSV,FUCD,
              Z1,ZD,KIJ,KEQ,IFLG)
     +
        CALL K(XD,Z,TC,PC,W,PP,P,T,N,2,1,APUR,BPUR,FUCSL,FUCD,
     +
              Z2,ZD,KIJ,KEQ,IFLG)
* System's Gibbs energy of mixing calculation and specification of
* the type of system the original phase is: vapour-like or liquid-
* like
        SUM1=0.DO
        SUM2=0.D0
        PLOG=DLOG(P)
        DO 100 I=1,N
           HV(I)=DLOG(Z(I)*FUCSV(I))
           HL(I)=DLOG(Z(I)*FUCSL(I))
           ALSL(I)=HL(I)+PLOG
           SUM1=SUM1+Z(I)*(HV(I)+PLOG)
```

. .

```
SUM2=SUM2+Z(I)*ALSL(I)
  100
          CONTINUE
         DELGV=R+T+SUM1
         DELGL=R*T*SUM2
          IF (DELGV .LT. DELGL) THEN
            IFLG(17)=0
          ELSE
            IFLG(17)=1
          END IF
* Calculation of the initial vapour and liquid compositions using
* Raoult's law
          SUM1=0.DO
          SUM2=0.D0
          DO 110 I=1,N
            PVAP=PC(I)*DEXP(5.3727D0*(1.D0+W(I))*(1.D0-TC(I)/T))/P
            C1(I)=Z(I)+PVAP
            C2(I)=Z(I)/PVAP
            SUM1=SUM1+C1(I)
            SUM2=SUM2+C2(I)
          CONTINUE
  110
         DO 120 I=1,N
            YI(I)=C1(I)/SUM1
            XIN(I)=C2(I)/SUM2
  120
          CONTINUE
          CALL TIME(1,0,ITIMI)
          ITIMT=ITIMT+ITIMI
* Printing of the calculation conditions
          IF (IFTP .EQ. 3) THEN
            WRITE(6,620) (I,Z(I), I=1,N)
          ELSE
            WRITE(6,615) T,P
          END IF
         WRITE(6,670) DELGV,Z1,DELGL,Z2
FORMAT(/8X,'DEL G VAP=',F10.4,' cal/gmole',4X,'Z VAP=',F7.4/
8X,'DEL G LIQ=',F10.4,' cal/gmole',4X,'Z LIQ=',
  670
                   F7.4)
      +
          IF (IFLG(1) .EQ. 1) WRITE(6,680)
FORMAT(40X, '*** EXTRAPOLATED Z VAP ***')
  680
          IF (IFLG(2) .EQ. 1) WRITE(6,690)
FORMAT(40X,'*** EXTRAPOLATED Z LIQ ***')
  690
          IF (IFLG(17) .EQ. 0) WRITE(6,700)
  700
          FORMAT(/4X,'THE SYSTEM IS VAPOUR-LIKE')
           IF (IFLG(17) .EQ. 1) WRITE(6,710)
         FORMAT(/4X,'THE SYSTEM IS LIQUID-LIKE')
IF (IFLG(15) .EQ. 2) WRITE(6,720) (I,HV(I),I,HL(I), I=1,N)
FORMAT(6X,'hv(',I2,')=',G11.4,2X,'hl(',I2,')=',G11.4)
  710
  720
          WRITE(6,780) ITIMI
          IF (IFLG(17) .EQ. 0) GO TO 300
IF (INOLC .EQ. 1) GO TO 150
* The system is liquid like. A vapour phase is searched
          WRITE(6,730)
FORMAT(/4X,'SUBROUTINE VAPSER RESULTS')
  730
          CALL TIME(0,0,ITDUM)
          CALL VAPSER(EPS, ITMAX, IWEG, TIMAX, N, P, T, Z, YI, HL, TC, PC, W, PP,
                         APUR, BPUR, KIJ, ITNUM, OBJV, FUCV, Y, IFLG)
      +
          CALL TIME(1,0,ITIMI)
          ITIMT=ITIMT+ITIMI
```

```
IF (IFLG(15) .NE. 0) GO TO 130
        IF (IFLG(5) .EQ. 1) THEN
           WRITE(6,740)
  740
          FORMAT(6X, 'UNSTABLE SYSTEM. VAPOUR PHASE FOUND')
        ELSE
           WRITE(6,750)
  750
          FORMAT(6X,'STABLE SYSTEM. VAPOUR PHASE NOT FOUND')
        END IF
        WRITE(6,760) (I,Y(I), I=1,N)
FORMAT(8X,'y(',I2,')=',F9.6)
  130
  760
        WRITE(6,770) ITNUM
        FORMAT(6X,'NO. OF ITERATIONS=',13)
  770
        WRITE(6,780) ITIMI
  780
        FORMAT(4X, 'CPU TIME=', I4, ' msecs')
* The system is liquid-like. A liquid phase is searched
         IF (INITL .NE. 1) GO TO 150
        DO 140 I=1,N
          XI(N+1,I)=XIN(I)
  140
        CONTINUE
        CALL TIME(0,0,ITDUM)
        CALL INMICH(N,XI)
        CALL TIME(1,0,ITIMI)
        ITIMT=ITIMT+ITIMI
        WRITE(6,790)
  790
        FORMAT(/4X,'SUBROUTINE INMICH')
        IF (IFLG(15) .EQ. 2) WRITE(6,800) ((XI(J,I), I=1,N), J=1,N+1)
  800
        FORMAT(6X,6F9.5)
        WRITE(6,780) ITIMI
        WRITE(6,810)
        FORMAT(/4X,'SUBROUTINE LIQSER RESULTS')
  810
        CALL TIME(0,0,ITDUM)
        CALL LIQSER(EPS, ITMAX, IWEG, TIMAX, N+1, N, P, T, Z, XI, HL, TC, PC, W, PP,
                      APUR, BPUR, KIJ, ÍTNUM, ÓBJL, FÚCLÍ, FÚCLÍ, XÍ, X2, IFLG)
        CALL TIME(1,0,ITIMI)
        ITIMT=ITIMT+ITIMI
        GO TO 210
        DO 160 I=1,N
  150
          XI(3,I)=XIN(I)
  160
        CONTINUE
        IF (INOLC .EQ. 1) GO TO 170
GO TO (170,180,190) INITL-1
        CALL TIME(0,0, ITDUM)
  170
        CALL INGASE(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,ALSL,
                      IP,IQ,XI)
     +
        CALL TIME(1,0, ITIMI)
        ITIMT=ITIMT+ITIMI
        WRITE(6,820) IP,IQ
        FORMAT(/4X,'SUBROUTINE INGASE RESULTS'/6X,'IP=',I2,2X,
  820
                'IQ=',I2)
        WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
  830
        FORMAT(6X,6F9.5)
        WRITE(6,780) ITIMI
        IF (INOLC .EQ. 0) GO TO 200
        CALL TIME(0,0,ITDUM)
CALL INSHAH(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,IP,IQ,
  180
     +
                      XI)
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
        WRITE(6,840) IP,IQ
```

```
FORMAT(/4X, 'SUBROUTINE INSHAH RESULTS'/6X, 'IP=', I2, 2X,
  840
         'IQ=',I2)
WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
WRITE(6,780) ITIMI
         IF (INOLC .EQ. 0) GO TO 200
  190
         CALL TIME(0,0, ITDUM)
         CALL INLIQ(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,ALSL,
      +
                      IP,IQ,XI)
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
         WRITE(6,850) IP,IQ
         FORMAT(/4X, 'SUBROUTINE INLIQ RESULTS'/6X, 'IP=', 12, 2X,
  850
         'IQ=',I2)
WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
WRITE(6,780) ITIMI
         IF (INOLC .EQ. 1) GO TO 570
  200
         WRITE(6,810)
         CALL TIME(0,0,ITDUM)
         CALL LIQSER (EPS, ITMAX, IWEG, TIMAX, MX, N, P, T, Z, XI, HL, TC, PC, W, PP,
                       APUR, BPUR, KIJ, ITNUM, OBJL, FUCL1, FUCL2, X1, X2, IFLG)
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
  210
         IF (IFLG(15) .NE. 0) GO TO 220
         IF (IFLG(6) .EQ. 1) THEN
           WRITE(6,860)
  860
           FORMAT(6X, 'UNSTABLE SYSTEM. ONE LIQUID PHASE FOUND')
         ELSE IF (IFLG(6) .EQ. 2) THEN
           WRITE(6,870)
  870
           FORMAT(6X, 'UNSTABLE SYSTEM. TWO LIQUID PHASES FOUND')
         ELSE
           WRITE(6,880)
  880
           FORMAT(6X, 'STABLE SYSTEM. NO LIQUID PHASE(S) FOUND')
         END IF
         WRITE(6,890) (I,X1(I),I,X2(I), I=1,N)
FORMAT(8X,'x1(',I2,')=',F9.6,2X,'x2(',I2,')=',F9.6)
  220
  890
         WRITE(6,770) ITNUM
WRITE(6,780) ITIMI
         IF (INOLC .EQ. 2) GO TO 1135
IF (ISRCH .EQ. 0) GO TO 226
* If a vapour is found and no liquid(s) found, a liquid phase which
* might have been missed is searched using the additional search
* (subroutine PHASER)
         IF (IFLG(5) .EQ. 1 .AND. IFLG(6) .EQ. 0) THEN CALL TIME(0,0,ITDUM)
           CALL PHASER(Z,Y,XD,P,T,N,DELGL,FUCV,FUCD,IFLG,HP)
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
           IF (INITL .EQ. 1) THEN
             MX=N+1
           ELSE
             MX=3
           END IF
           WRITE(6,892)
  892
           FORMAT(/4X, 'SUBROUTINE LIQSER RESULTS AFTER CALLING PHASER')
           WRITE(6,780) ITIMI
           CALL TIME(0,0,ITDUM)
           CALL LIQSER(EPS, ITMAX, IWEG, TIMAX, MX, N, P, T, Z, XI, HP, TC, PC, W,
                         PP, APUR, BPUR, KIJ, ITNUM, OBJL, FUCL1, FUCL2, X1, X2,
     +
                         IFLG)
           IF (IFLG(15) .NE. 0) GO TO 222
```

```
IF (IFLG(6) .EQ. 1) THEN
             IFLG(19)=1
             WRITE(6,860)
           ELSE IF (IFLG(6) .EQ. 2) THEN
             WRITE(6,870)
           ELSE
             WRITE(6,880)
           END IF
          WRITE(6,890) (I,X1(I),I,X2(I), I=1,N)
WRITE(6,770) ITNUM
  222
           WRITE(6,780) ITIMI
         END IF
* If no vapour is found and one or two liquid phases are found, a
* vapour which might have been missed is looked for with the proposed
* additional search
         IF (IFLG(5) .EQ. 0 .AND. IFLG(6) .NE. 0) THEN CALL TIME(0,0,ITDUM)
           IF (IFLG(6) .EQ. 1) THEN
             CALL PHASER(Z,X1,XD,P,T,N,DELGL,FUCL1,FUCD,IFLG,HP)
           ELSE
             CALL PHASER(Z,X1,X2,P,T,N,DELGL,FUCL1,FUCL2,IFLG,HP)
           END IF
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
          WRITE(6,895)
FORMAT(/4X,'SUBROUTINE VAPSER RESULTS AFTER CALLING PHASER')
  895
           CALL TIME(0,0,ITDUM)
           CALL VAPSER(EPS, ITMAX, IWEG, TIMAX, N, P, T, Z, YI, HP, TC, PC, W, PP,
                        APUR, BPUR, KIJ, ITNUM, OBJ, FUCV, Y, IFLG)
     +
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
           IF (IFLG(15) .NE. 0) GO TO 224
           IF (IFLG(5) .EQ. 1) THEN
             WRITE(6,740)
           ELSE
             WRITE(6,750)
           END IF
          WRITE(6,760) (I,Y(I), I=1,N)
WRITE(6,770) ITNUM
WRITE(6,780) ITIMI
  224
         END IF
* If no vapour is found and no liquid(s) either, the system is
* stable as a liquid
         IF (IFLG(5) .EQ. 0 .AND. IFLG(6) .EQ. 0) THEN
  226
           IFLG(11)=1
           WRITE(6,900)
  900
           FORMAT(/4X, '+++ STABLE SYSTEM AS L +++')
           GO TO 440
         END IF
* If no vapour is found and one or two liquid phases are found,
* the system is stable as LL
         IF (IFLG(5) .EQ. O .AND. IFLG(6) .NE. O) THEN
           IFLG(11)=3
           WRITE(6,910)
           FORMAT(/4X,'+++ STABLE SYSTEM AS LL +++')
IF (IFLG(6) .EQ. 1) THEN
  910
```

```
DO 230 I=1,N
                KL(I)=FUCSL(I)/FUCL1(I)
                X2(I)=Z(I)
  230
             CONTINUE
           ELSE
             DO 240 I=1,N
               KL(I)=FUCL2(I)/FUCL1(I)
  240
             CONTINUE
           END IF
           GO TO 440
         END IF
* If a vapour is found and no liquid phase(s) found, the system is
* stable as VL
         IF (IFLG(5) .EQ. 1 .AND. IFLG(6) .EQ. 0) THEN
           IFLG(11)=2
           WRITE(6,920)
FORMAT(/4X,'+++ STABLE SYSTEM AS VL +++')
  920
           DO 250 I=1,N
             KV(I)=FUCSL(I)/FUCV(I)
             X2N(I)=Z(I)
  250
           CONTINUE
           GO TO 440
         END IF
* If a vapour is found and one or two liquid phases are found, the
* most stable system is determined using Nelson's or the multiphase
* stability test (subroutines NELSTA or TESTA)
         IF (IFLG(6) .EQ. 1) THEN
DO 260 I=1,N
             KV(I)=FUCSL(I)/FUCV(I)
             KL(I)=FUCSL(I)/FUCL1(I)
              X_2(I) = Z(I)
  260
           CONTINUE
         ELSE
           DO 270 I=1,N
             KV(I)=FUCL2(I)/FUCV(I)
             KL(I)=FUCL2(I)/FUCL1(I)
  270
           CONTINUE
         END IF
         IF (INITL .EQ.1) THEN
           IP=1
           X1MAX=X1(1)
           DO 280 I=2,N
             IF (X1(I) .GT. X1MAX) THEN
                IP=I
                X1MAX=X1(I)
              END IF
           CONTINUE
  280
         END IF
         IF (ISTA .EQ. 1 .OR. ISTA .EQ. 2 .AND. IFLG(19) .EQ. 1) THEN
           IFLG(19)=0
           WRITE(6,930)
FORMAT(/4X,'SUBROUTINE NELSTA RESULTS')
  930
           CALL TIME(0,0,ITDUM)
           CALL NELSTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
KV,KL,IFLG,YN,X1N,X2N,FUCVN,FUCL1N,FUCL2N,
     +
                         ALPHA, BETA)
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
         ELSE
```
```
WRITE(6,940)
          FORMAT(/4X, 'SUBROUTINE TESTA RESULTS')
CALL TIME(0,0,ITDUM)
CALL TESTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
  940
                       KV, KL, IFLG, YN, X1N, X2N, FUCVN, FUCL1N, FUCL2N,
     +
                       ALPHA, BETA)
     +
        CALL TIME(1,0,ITIMI)
         ITINT=ITINT+ITINI
        END IF
        WRITE(6,950) ALPHA, BETA
        950
  960
           WRITE(6,970)
        FORMAT(/4X,'+++ SYSTEM STABLE AS V +++')
ELSE IF (IFLG(11) .EQ. 1) THEN
  970
           WRITE(6,900)
        ELSE IF (IFLG(11) .EQ. 2) THEN
           WRITE(6,920)
           IF (IFLG(14) .EQ.1) THEN
             DO 290 I=1,N
               KV(I)=FUCL1(I)/FUCV(I)
  290
             CONTINUE
          END IF
        ELSE IF (IFLG(11) .EQ. 3) THEN
WRITE(6,910)
         ELSE
           WRITE(6,980)
  980
          FORMAT(/4X, '+++ STABLE SYSTEM AS VLL +++')
         END IF
         WRITE(6,780) ITIMI
         GO TO 440
* The system is vapour-like. A liquid phase is searched
. 300
       IF (INOLC .EQ. 1) GO TO 320
        IF (INITL .NE. 1) GO TO 320
        DO 310 I=1,N
           XI(N+1,I)=XIN(I)
  310
         CONTINUE
         CALL TIME(0,0,ITDUM)
        CALL INMICH(N,XI)
CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
         WRITE(6,790)
         IF (IFLG(15) .EQ. 2) WRITE(6,800) ((XI(J,I), I=1,N), J=1,N+1)
         WRITE(6,780) ITIMI
         WRITE(6,810)
         CALL TIME(0,0,ITDUM)
         CALL LIQSER(EPS, ITMAX, IWEG, TIMAX, N+1, N, P, T, Z, XI, HV, TC, PC, W, PP,
                      APUR, BPUR, KIJ, ITNUM, OBJL, FUCL1, FUCL2, X1, X2, IFLG)
     +
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
         GO TO 380
  320
         DO 330 I=1,N
           XI(3,I)=XIN(I)
  330
         CONTINUE
         IF (INOLC .EQ. 1) GO TO 340
         GO TO (340,350,360) INITL-1
  340
         CALL TIME(0,0,ITDUM)
```

```
CALL INGASE(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,ALSL,
     +
                       IP,IQ,XÌ)
         CALL TIME(1,0, ITIMI)
         ITIMT=ITIMT+ITIMI
         WRITE(6,820) IP,IQ
WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
WRITE(6,780) ITIMI
         IF (INOLC .EQ. 0) GO TO 370
CALL TIME(0,0,ITDUM)
  350
         CALL INSHAH(Z, TC, PC, W, PP, P, T, N, APUR, BPUR, KIJ, IFLG, IP, IQ,
     +
                       XI)
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
         WRITE(6,840) IP,IQ
         WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
         WRITE(6,780) ITIMI
         IF (INOLC .EQ. 0) GO TO 370
CALL TIME(0,0,ITDUM)
  360
         CALL INLIQ(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,ALSL,
                      IP,IQ,XI)
         CALL TIME(1,0, ITIMI)
         ITIMT=ITIMT+ITIMI
         WRITE(6,850) IP,IQ
         WRITE(6,830) ((XI(J,I), I=1,N), J=1,3)
         WRITE(6,780) ITIMI
         IF (INOLC .EQ. 1) GO TO 570
         WRITE(6,810)
  370
         CALL TIME(0,0,ITDUM)
         CALL LIQSER(EPS, ITMAX, IWEG, TIMAX, MX, N, P, T, Z, XI, HV, TC, PC, W, PP,
                       APUR, BPUR, KIJ, ITNUM, OBJL, FUCL1, FUCL2, X1, X2, IFLG)
         CALL TIME(1,0,ITIMI)
         ITIMT=ITIMT+ITIMI
         IF (IFLG(15) .NE. 0) GO TO 390
IF (IFLG(6) .EQ. 1) THEN
  380
           WRITE(6,860)
         ELSE IF (IFLG(6) .EQ. 2) THEN
           WRITE(6,870)
         ELSE
           WRITE(6,880)
         END IF
         WRITE(6,890) (I,X1(I),I,X2(I), I=1,N)
WRITE(6,770) ITNUM
  390
         WRITE(6,780) ITIMI
         IF (INOLC .EQ. 2) GO TO 1135
IF (ISRCH .EQ. 0) GO TO 395
* If one liquid phase is found, another liquid phase which might
* have been missed is searched using the proposed additional search
         IF (IFLG(6) .EQ. 1) THEN
CALL TIME(0,0,ITDUM)
           CALL PHASER(Z,X1,XD,P,T,N,DELGV,FUCL1,FUCD,IFLG,HP)
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
           IF (INITL .EQ. 1) THEN
             MX=N+1
           ELSE
             MX=3
           END IF
           WRITE(6,892)
           WRITE(6,780) ITIMI
           CALL TIME(0,0,ITDUM)
```

```
CALL LIQSER(EPS, ITMAX, IWEG, TIMAX, MX, N, P, T, Z, XI, HP, TC, PC, W,
                        PP, APUR, BPUR, KIJ, ITNUMP, OBJLP, FUCL1P, FUCL2P,
     +
                        X1P,X2P,IFLG)
     4
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
           IF (IFLG(6) .EQ. 2) THEN
OBJL(1)=OBJLP(1)
             OBJL(2)=OBJLP(2)
             DO 392 I=1,N
               FUCL1(I)=FUCL1P(I)
               FUCL2(I)=FUCL2P(I)
               X1(I)=X1P(I)
               X2(I) = X2P(I)
  392
             CONTINUE
           END IF
           IF (IFLG(15) .NE. 0) GO TO 393
IF (IFLG(6) .EQ. 1) THEN
             WRITE(6,860)
             WRITE(6,890) (I,X1P(I),I,X2P(I), I=1,N)
             GO TO 394
           ELSE IF (IFLG(6) .EQ. 2) THEN
             WRITE(6,870)
           ELSE
             IFLG(6)=1
             WRITE(6,880)
          END IF
           WRITE(6,890) (I,X1(I),I,X2(I), I=1,N)
WRITE(6,770) ITNUMP
  393
  394
           WRITE(6,780) ITIMI
        END IF
* If no liquid phase(s) are found, the system is stable as vapour
  395
         IF (IFLG(6) .EQ. 0) THEN
           IFLG(11)=0
           WRITE(6,970)
           GO TO 440
         END IF
* If one liquid phase is found, the system is stable as VL
         IF (IFLG(6) .EQ. 1) THEN
           IFLG(11)=2
           WRITE(6,920)
           DO 400 I=1,N
             KV(I)=FUCL1(I)/FUCSV(I)
             Y(I)=Z(I)
             X2N(I)=X1(I)
  400
           CONTINUE
           GO TO 440
        END IF
* If two liquid phases are found, the most stable system is
* determined using subroutines NELSTA or TESTA
        DO 410 I=1,N
           KV(I)=FUCL2(I)/FUCSV(I)
           KL(I)=FUCL2(I)/FUCL1(I)
           Y(I)=Z(I)
  410
        CONTINUE
         IF (INITL .EQ. 1) THEN
           IP=1
           X1MAX=X1(1)
```

```
DO 420 I=2,N
             IF (X1(I) .GT. X1MAX) THEN
               IP=I
               X1MAX=X1(I)
             END IF
  420
           CONTINUE
        END IF
        IF (ISTA .EQ. 1) THEN
           WRITE(6,930)
           CALL TIME(0,0,ITDUM)
          CALL NELSTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
                        KV, KL, IFLG, YN, XIN, X2N, FUCVN, FUCL1N, FUCL2N,
                        ALPHA, BETA)
     +
           CALL TIME(1,0,ITIMI)
           ITIMT=ITIMT+ITIMI
        ELSE
          WRITE(6,940)
           CALL TIME(0,0,ITDUM)
           CALL TESTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
                       KV,KL,IFLG,YN,XIN,X2N,FUCVN,FUCL1N,FUCL2N,
     +
     +
                       ALPHA, BETA)
           CALL TIME(1,0,ITÍMI)
           ITIMT=ITIMT+ITIMI
        END IF
        WRITE(6,950) ALPHA,BETA
WRITE(6,960) (I,YN(I),I,X1N(I),I,X2N(I), I=1,N)
IF (IFLG(11) .EQ. 0) THEN
           WRITE(6,970)
        ELSE IF (IFLG(11) .EQ. 1) THEN
           WRITE(6,900)
        ELSE IF (IFLG(11) .EQ. 2) THEN
WRITE(6,920)
           IF (IFLG(14) .EQ. 1) THEN
             DO 430 I=1,N
               KV(I)=FUCL1(I)/FUCSV(I)
  430
             CONTINUE
           END IF
        ELSE IF (IFLG(11) .EQ. 3) THEN
           WRITE(6,910)
        ELSE
          WRITE(6,980)
        END IF
        WRITE(6,780) ITIMI
* If the system is in the one phase region the calculation ends
 440
       IF (IFLG(11) .EQ. 0 .OR. IFLG(11) .EQ. 1) THEN
          WRITE(6,985) ITIMT
FORMAT(/4X,'TOT CPU TIME=',I4,' msecs')
 985
           GO TO 1135
        ELSE IF(IFLG(11) .EQ. 2) THEN
* Calculation of the initial value of V/F for the VL flash
           SUM1=0.D0
           SUM2=0.D0
          DO 450 I=1,N
             SUM1=SUM1+KV(I)*Z(I)
             SUM2=SUM2+(1.DO-KV(I))*Z(I)/(1.DO+KV(I))
  450
           CONTINUE
           SUM1=1.DO-SUM1
           SUM2=2.DO*SUM2
          F=SUM2/(SUM2-SUM1)
```

```
IF (F .GT. O.DO) THEN
              ALPHA=0.5D0*(1.D0-F)
              GO TO 470
            END IF
            SUM3=0.D0
           DO 460 I=1,N
              SUM3=SUM3+Z(I)/KV(I)
  460
            CONTINUE
            SUM3=SUM3-1.DO
            ALPHA=(SUM2-0.5D0*SUM3)/(SUM2-SUM3)
           IF (ALPHA .LT. 0.05D0) ALPHA=0.05D0
IF (ALPHA :GT. 0.95D0) ALPHA=0.95D0
  470
            DO 480 I=1,N
              YN(I)=Y(I)
              XN(I)=X2N(I)
  480
            CONTINUE
* Printing of the initial estimates for the VL flash
            WRITE(6,990) T,P,ALPHA
           FORMAT(/4X, 'VL FLASH INITIALIZATION',2X, 'T=',F8.3,' K',
2X, 'P=',F8.3,' atm',2X, 'V/F=',F8.5/9X,'I',5X,'Z',
9X,'Y',8X,'X')
WRITE(6,1000) (I,Z(I),YN(I),XN(I), I=1,N)
  990
      +
      +
 1000
           FORMAT(8X,12,1X,F8.5,1X,F8.5,1X,F8.5)
            WRITE(6,780) ITIMT
* VL flash calculation
           ITEQ=1
            TYPEQ='VL FLASH'
            CALL TIME(0,0, ITDUM)
            CALL FLASH2(ITEQ, EPSF, ITMAX, ITNUM, IWEGF, OBJ, N, P, T, ALPHA, XN,
                          YN,Z,TC,PC,W,PP,KIJ,FUCVN,FUCL1N,Z1,Z2,KV,IFLG)
      +
            CALL TIME(1,0, ITIMC)
            BETA=ALPHA
            CALL GIBBS(ITEQ, N, P, T, ALPHA, BETA, YN, XN, X1N, X2N, Z, TC, PC, W,
                         PP, KIJ, FUCVN, FUCL1N, FUCL2N, DG, IFLG)
      +
           BETA=0.DO
            GO TO 530
         ELSE IF (IFLG(11) .EQ. 3) THEN
* Calculation of the initial value of L1/F for the LL flash
            IF (INITL .EQ. 1) THEN
              IF (IP .EQ. O) THEN
                IP=1
                X1MAX=X1(1)
                DO 490 I=2,N
                   IF (X1(I) .GT. X1MAX) THEN
                     IP=I
                     X1MAX=X1(I)
                   END IF
  490
                CONTINUE
              END IF
              IQ=1
              \overline{X2}MAX=X2(1)
              DO 500 I=2,N
                IF (X2(I) .GT. X2MAX) THEN
                   IQ=I
                   X2MAX=X2(I)
                END IF
  500
              CONTINUE
```

```
END IF
           BETA=(Z(IP)*(1.DO-1.DO/KL(IP))+KL(IQ)*Z(IQ))/
      +
                  (Z(IP)+Z(IQ))
           IF (BETA .LT. 0.05D0) BETA=0.05D0
IF (BETA .GT. 0.95D0) BETA=0.95D0
           DO 510 I=1,N
              X1N(I) = XI(I)
              X2N(I)=X2(I)
  510
           CONTINUE
* Printing of the initial estimates for the LL flash
           WRITE(6,1010) T,P,BETA
           FORMAT(/4X,'LL FLASH INITIALIZATION',2X,'T=',F8.3,' K',
 1010
           2X,'P=',F8.3,' atm',2X,'L1/F=',F8.5/9X,'I',5X,
'Z',8X,'X1',8X,'X2')
WRITE(6,1000) (I,Z(I),X1N(I),X2N(I), I=1,N)
      +
           WRITE(6,780) ITIMT
* LL flash calculation
           ITEQ=2
           TYPEQ='LL FLASH'
           CALL TIME(0,0, ITDUM)
           CALL FLASH2(ITEQ, EPSF, ITMAX, ITNUM, IWEGF, OBJ, N, P, T, BETA, X2N, X1N, Z, TC, PC, W, PP, KIJ, FUCVN, FUCL1N, Z1, Z2, KL, IFLG)
     +
           CALL TIME(1,0,ITIMC)
           CALL GIBBS(ITEQ, N, P, T, ALPHA, BETA, X1N, X2N, X1, X2, Z, TC, PC, W,
                        PP,KIJ,FUCVN,FUCL1N,FUCL2N,DG,IFLG)
      +
           DO 520 I=1,N
              YN(I)=0.DO
              X1N(I)=X1(I)
              X2N(I) = X2(I)
  520
           CONTINUE
           GO TO 530
         ELSE
* Printing of the initial estimates for the VLL flash
           WRITE(6,1020) T,P,ALPHA,BETA
           1020
      +
      +
 1030
           WRITE(6,780) ITIMT
* VLL flash calculation
           ITEO=3
           TYPEQ='VLL FLASH'
           CALL TIME(0,0,ITDUM)
           CALL FLASH3(EPSF, ITMAX, ITNUM, IWEGF, OBJ, N, P, T, ALPHA, BETA, YN,
                         X1N, X2N, Z, TC, PC, W, PP, KIJ, ZED, FUCVN, FUCL1N,
      +
                         FUCL2N, IFLG)
      +
           CALL TIME(1,0,ITIMC)
         END IF
         CALL GIBBS(ITEQ, N, P, T, ALPHA, BETA, YN, XN, X1N, X2N, Z, TC, PC, W, PP,
                      KIJ, FUCVN, FUCL1N, FUCL2N, DG, IFLG)
* Printing of the calculation results
```

530 ONMAB=1.DO-ALPHA-BETA

```
IF (ITEQ .EQ. 2) THEN
           ZED(1)=0.D0
           ZED(2)=Z1
           ZED(3)=Z2
        ELSE IF (ITEQ .EQ. 1) THEN
           ZED(1)=Z1
           ZED(2)=0.D0
           ZED(3)=Z2
        END IF
        IF (IFLG(7) .EQ. 0) THEN
           WRITE(6,1040)
FORMAT(/4X,'WITHOUT WEGSTEINS METHOD')
1040
           GO TO 540
        END IF
        IF (IFLG(8) .EQ. 0) THEN
           DAMP=' WITHOUT DAMPING'
        ELSE
           DAMP=' WITH DAMPING'
        END IF
      WRITE(6,1050) IWEGF, DAMP
1050 FORMAT(/4X, 'WEGSTEINS METHOD EACH', I3, ' ITERATIONS', A16)
540 WRITE(6,1060) ITMAX, EPSF
1060 FORMAT(4X, 'MAXIMUM NO. OF ITERATIONS=', I3,2X,
               'CONVERGENCE=',F8.5)
        WRITE(6,1070) TYPEQ

FORMAT (/15X,A10,2X,'CALCULATION RESULTS')

WRITE(6,1080) T,P,ITNUM,OBJ,ALPHA,BETA,ONMAB,DG

FORMAT(4X,'T=',F8.3,'K',2X,'P=',F8.3,'atm',2X,

'NUM. ITER.=',I3,2X,'OBJ. FUNC.=',E12.5/

4X,'V/F=',F8.5,2X,'L1/F=',F8.5,2X,'L2/F=',
1070
1080
    +
                 F8.5,2X,'DEL G=',F11.3,' cal/gmole')
     +
        DO 560 I=1,N
           IF (ITEQ .EQ. 2) THEN
             KV(I)=1.DO
           ELSE
              KV(I)=FUCL2N(I)/FUCVN(I)
           END IF
           IF (ITEQ .EQ. 1) THEN
              KL(I)=1.D0
           ELSE
              KL(I)=FUCL2N(I)/FUCL1N(I)
           END IF
 560
        CONTINUE
        IF (IFLG(3) .EQ. 1) THEN
           ZL2='ZLIQ2ext='
        ELSE
           ZL2='ZLIQ2='
        END IF
        IF (IFLG(2) .EQ. 1) THEN
           ZL1='ZLIQ1ext='
        ELSE
           ZL1='ZLIQ1='
        ENDIF
        IF (IFLG(1) .EQ. 1) THEN
           ZV='ZVAPext='
        ELSE
           ZV='ZVAP='
        END IF
        WRITE(6,1090) ZV,ZED(1),ZL1,ZED(2),ZL2,ZED(3)
1090
        FORMAT(4X, A8, F7.5, 2X, A9, F7.5, 2X, A9, F7.5)
        IF (ITEQ .NE. 1) WRITE(6,1100) IP, IQ
1100
        FORMAT(4X, 'KEY COMP.#1=', I2, 2X, 'KEY COMP.#2=', I2)
        WRITE(6,1110)
```

```
1110
                     I=1,N)
       FORMAT(7X,12,1X,F8.5,1X,F8.5,1X,F8.5,1X,F8.5,2X,
G11.4,2X,G11.4)
WRITE(6,1130) ITIMC
1120
     +
       FORMAT(4X, 'CPU TIME=', I4, ' msecs')
 1130
       WRITE(6,985) ITIMT+ITIMC
       WRITE(6,1140)
FORMAT(/'======:*/)
 1135
1140
       ITOTIM=ITOTIM+ITIMT+ITIMC
* End of loop for calculation at different compositions at fixed T & P
 565 CONTINUE
 570 CONTINUE
 580 CONTINUE
* End of loop for calculations at fixed comp. & T at various P or at
* fixed comp. & P at different T
 590 CONTINUE
     WRITE(6,1150) ITOTIM
 1150 FORMAT(/4X, 'GLOBAL CPU TIME=', I5, ' msecs'/)
     STOP
     END
```

```
END
                                                                                                                                                                                   RETURN
                                                                                                                                                                            30 CONTINUE
                                                                                                                                                                      CONTINUE
                                                                                                                                                                                                        ٥T
                                                                                                                                          END IE
XI(1'I)=XINE1
XI(1'1)=XINE1
                                                                                                                                                                             ELSE
                                                                                                                                 XI(1'1)=XIE1
IF (1 .EQ. 1) THEN
                                                                                                                                                             DO IO 1=I'M
                                                                                                                                                                   DO 50 I=1'M
                                                                                                                               (1-N)/(1-1)/(N-1)
                                                                                                                                                                   XIE1=0.98D0
                                                                                                                                                                                               * Vector
           * components in their corresponding vector and the 0.02 difference
* is distributed equally among the rest of the components in that
     A comparison of the section is a series of the termination of terminatio of termination of termination of termination of ter
                                                                                                                    DIWENZION XI(S1'SO)
                                                                                                                               SUBROUTINE INMICH(N, XI)
                                                      ******************
***
*
*
                                                                                                                                                                                      TIJ92AH9
                                                                                                                                                                                                                     *
                                                                                                                                                                                    * Called by:
*
*
                                                                                                                                                                                                   anoN
*
                                                                                                                                                                                                                     *
                                                                                                                                                       * Subroutines called:
*
*
                                                       N x N array of initial liquid compositions
+
                                                                                                                                                                                                        IX
                                                                                                                                                                                            auqau0-
                                                                                                                                                                                                                      *
*
                                                                                                                           Rumber of components
                                                                                                                                                                                                            N
*
                                                                                                                                                                                                                     *
                                                                                                                                                                                                andu<u>I</u>-
                                                                                                                                                                                                                     *
                                                                                                                                                                                Parameters:
                                                                                                                                                                                                                    *
*
*
                                                                                                                                                                                      вөтьшітвө
                                                                                                                                                                                                                     *
                     to Michelsen's scheme of using the pure components as initial
                                                                                                                                                                                                                     *
    W composition vectors of 'almost' pure components in analogous way
                                                                                                                                                                                                                     *
*
        Suitabization ASCOL entrout tor zation for subtrouting Light considering
                                                                                                                                                                                                                      *
                                                                                                                                                                                         * Purpose:
                 ******************
                                                                                                                                                                                                                      *
      SUBROUTINE INMICH (INIT. OF LIQ. COMPS. AS IN MICHELSEN'S METHOD)
                                                                                                                                                                                                                     *
***
             ***********************
```

Appendix C. Description and listings of the computer programs

```
SUBROUTINE INSHAH (INIT. OF LIQ. COMPS. BY SHAH'S METHOD)
* Purpose:
  Initialize liquid compositions to be used in subroutine LIQSER
* by obtaining key components p and q according to Shah's method.
* These components will be the phase initiators for the two liquid
* phases searched (liq 1 and 2).
* Parameters:
*
 -Input
 Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG as in program PHASPLIT
*
*
*
 -Output
*
  IP
          Index of key component p in phase 1
                                  q "
*
  IQ
                           ...
                                         44
                                             2
*
  XI
          2 x N array of initial liquid compositions (mole frac.)
*
 Main variables:
*
  APT
          Vector of pure component dimensionless attraction param.
*
          in the EOS for the trial components
*
  ACI
          Pseudo-infinite dilution activity of component i in the
*
          original mixture
  ACT
          Pseudo-infinite dilution activity of trial components for
*
          phase 2 in comp. p
*
*
  BPT
          Vector of pure component dimensionless repulsion param.
          in the EOS for the trial components
*
          Vector of dummy fug. coeffs.
""" infinite dilution fug. coeffs. in the mixture
*
  FUCD
*
  FUCLI
           ..
                 ...
*
  FUCLT
                                                   for the trial
*
          components
         Fug. coeff. of pure p with trace of comp. q
""" p at infinite dilution in q
""" q""" p
*
  FUCP1
*
  FUCP2
                                                 "Р
*
  FUCQ1
                      " pure q with trace of comp. p
" i at infinite dilution in li
           ...
                 ....
*
  FUCQ2
                 ....
           .....
                      " i at infinite dilution in liq. 1
*
  FUCR1
           *
  FUCR2
          Vector of critical pressures for the trial components
" " Mathias polar param." " " "
  PCT
*
*
  PPT
          Number of moles of comp. p in phase 1
*
  PN1
                          .....
*
  PN2
                                               2
                              •1
                                    q "
                          ••
            ...
                 .....
                      ...
  QN1
*
                                               1
                         ...
                 ....
                             ...
                                          ..
            ...
                      ....
*
  QN2
                                               2
                      **
            ..
                 н
                          ..
*
  RN1
                            the remaining i comps. in phase 1
            ....
                 ....
                      **
                          ....
*
  RN2
                                                               2
*
  TCT
          Vector of critical temperatures for the trial comps.
*
  TOTN1
          Total mole number in phase 1
*
  TOTN2
*
  XINF
          Vector of infinite dilution composition
          Comp. vector for the trial components in the second phase
*
  XS
                                                    the liq.
*
  XT
          Remaining variables defined as in program PHASPLIT
*
  Subroutines called:
*
  K
* Called by:
* PHASPLIT
```

```
**********
      SUBROUTINE INSHAH(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,
                          IP,IQ,XI)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 KEQ(3),KIJ(20,20)
DIMENSION APT(3),APUR(20),BPT(3),BPUR(20),FUCD(3),FUCLI(20),
+ FUCLT(3),IFLG(0:20),PC(20),PCT(3),PP(20),PPT(3),
     +
                 RN1(18), RN2(18), TC(20), TCT(3), W(20), WT(3), XD(3),
     +
      XI(21,20),XINF(20),XS(3),XT(3),Z(20)
IF (IFLG(15) .EQ. 2) WRITE(6,500)
     +
  500 FORMAT(/4X, 'SUBROUTINE INSHAH TRACING')
IF (IFLG(15) .EQ. 2) WRITE(6,510)
  510 FORMAT(/4X, FUG COEFF OF I IN THE LIQ MIXTURE AT INF DILUTION'
              /4X,'I',10X,'FUG COEFF')
     +
* Selection of phase initiator p as the component with the largest
* pseudo-infinite dilution activity in the overall mixture
      HAI=0.DO
      ZI=1.D-6
      DO 30 I=1,N
         SUM=0.DO
        DO 10 J=1,N
           IF (I .EQ. J) GO TO 10
           SUM=SUM+Z(J)
   10
         CONTINUE
        SUM=SUM+ZI
         XINF(I)=ZI/SUM
        DO 20 J=1,N
IF (I .EQ. J) GO TO 20
                                                         . -
           XINF(J)=Z(J)/SUM
   20
         CONTINUE
         CALL K(XD,XINF,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCLI,FUCD,
        ZI,ZD,KÍJ,KEQ,IFLG)
ACI=Z(I)*FUCLI(I)*P
     +
      IF (IFLG(15) .EQ. 2) WRITE(6,520) I,FUCLI(I)
  520 FORMAT(3X,12,9X,F10.6)
         IF (HAI .LT. ACI) THEN
           HAI=ACI
           IP=I
         END IF
   30 CONTINUE
* Selection of phase initiator q as the component with the largest
* pseudo-infinite dilution activity in pure component p
      HAT=0.DO
      TCT(1)=TC(IP)
      PCT(1)=PC(IP)
      WT(1)=W(IP)
      PPT(1)=PP(IP)
      APT(1)=APUR(IP)
      BPT(1)=BPUR(IP)
      XT(2)=1.D-6
      XT(1)=1.DO-XT(2)
      IF (IFLG(15) .EQ. 2) WRITE(6,530)
  530 FORMAT(/4X,'i
+ /4X,'
                         FUG COEFF OF PURE p
WITH TRACE OF COMP i
                                                      FUC COEFF OF i AT'
                                                     INF DILUTION IN p')
      DO 40 I=1,N
         IF (I .EQ. IP) GO TO 40
         TCT(2)=TC(I)
```

```
PCT(2)=PC(I)
        WT(2)=W(I)
        PPT(2)=PP(I)
        APT(2)=APUR(I)
        BPT(2)=BPUR(I)
        CALL K(XD,XT,TCT,PCT,WT,PPT,P,T,2,2,3,APT,BPT,FUCLT,FUCD,
ZT,ZD,KIJ,KEQ,IFLG)
      ACT=Z(I)*FUCLT(2)*P
IF (IFLG(15) .EQ. 2) WRITE(6,540) I,FUCLT(1),FUCLT(2)
  540 FORMAT(3X, 12, 8X, F10.6, 15X, F10.6)
        IF (HAT .LE. ACT) THEN
          HAT=ACT
          IQ=I
* Assignment of fugacity coefficients of 'pure' p in phase 1 and
* of q at infinite-dilution in phase 1
          FUCP1=FUCLT(1)
          FUCQ1=FUCLT(2)
        END IF
   40 CONTINUE
      IF (IFLG(15) .EQ. 2) WRITE(6,550) FUCP1,FUCQ1
  550 FORMAT(4X,'
                    FUCP1=',F10.6,'
                                               FUCQ1=',F10.6)
      XT(1)=1.D-6
      XT(2)=1.DO-XT(1)
      TCT(2)=TC(IQ)
      PCT(2)=PC(IQ)
      WT(2)=W(IQ)
      PPT(2) = PP(IQ)
      APT(2)=APUR(IQ)
      BPT(2)=BPUR(IQ)
      CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
ZT, ZD, KIJ, KEQ, IFLG)
     +
      IF (IFLG(15) .EQ. 2) WRITE(6,560)
  560 FORMAT(/4X,'
                                                  FUC COEFF OF PURE Q'
                       FUG COEFF OF P AT
              /4X,'
                       INF DILUTION IN Q
                                                  WITH TRACE OF COMP P')
     +
      IF (IFLG(15) .EQ. 2) WRITE(6,570) FUCLT(1), FUCLT(2)
                    FUCP2=',F10.6,'
                                               FUCQ2=',F10.6)
  570 FORMAT(4X,'
* Assignment of fugacity coefficients of p at infinite-dilution in
* phase 2 and of 'pure' q in phase 2
      FUCP2=FUCLT(1)
      FUCQ2=FUCLT(2)
* Calculation of the mole numbers of p and q from the equilibrium
* and mass balance equations assuming no other components are
* present
      A0=FUCP1-FUCP2
      B0=Z(IP)*(FUCP2-2.D0*FUCP1)
      CO=Z(IP)*FUCP2
      D0=FUCQ2-FUCQ1
      E0=2.D0+Z(IP)+FUCQ1-Z(IQ)+FUCQ2
      FO = -Z(IQ) * FUCQ2
      A=A0*(D0*(B0-C0)-A0*(E0-F0))
      B=B0*D0*(B0-C0)-A0*(E0*(C0+B0)-2.D0*C0*F0)
      C=CO*(CO*FO-BO*EO)
      IF (A .EQ. O.DO) THEN
        PN1=-C/B
        GO TO 50
      ELSE IF (C .EQ. O.DO) THEN
        PN1 = -B/A
```

```
GO TO 50
       ELSE
          DISC=B*B-4.DO*A*C
          IF (DISC .LT. 0.DO) THEN
            WRITE(6,580)
            FORMAT(/6X,'*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
/9X,'TO SOLVE THE QUADRATIC EQUATION FOR mpI IN'
/9X,'SUBROUTINE INSHAH')
PN1=Z(IP)/2.DO
  580
      +
            GO TO 60
          ELSE IF (DISC .EQ. 0.DO) THEN
            PN1=-B/(2.DO*A)
            GO TO 50
          END IF
       END IF
       TWA=2.DO*A
       RTDISC=DSQRT(DISC)
       PN1=(-B+RTDISC)/TWA
       PNO=(-B-RTDISC)/TWA
       IF (PN1 .GT. O.DO .AND. PN1 .LT. Z(IP)) THEN
          QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
          IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) GO TO 80
       ELSE IF (PNO .GT. O.DO .AND. PNO .LT. Z(IP)) THEN
          PN1=PNO
          QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
          IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) THEN
            GO TO 80
          ELSE
            GO TO 70
          END IF
   END IF
50 IF (PN1 .LE. 0.DO) PN1=0.1D0*Z(IP)
       IF (PN1 .GE. Z(IP)) PN1=0.9D0*Z(IP)
   60 QN1=-PN1*(A0*PN1+B0)/(A0*PN1+C0)
   70 IF (QN1 .LE. 0.DO) QN1=0.1D0*Z(IQ)
IF (QN1 .GE. Z(IQ)) QN1=0.9D0*Z(IQ)
80 PN2=Z(IP)-PN1
       QN2=Z(IQ)-QN1
       SMRN1=0.DO
       SMRN2=0.DO
       DEN1=PN1+QN1
       DEN2=PN2+QN2
       IF (IFLG(15) .EQ. 2) WRITE(6,590)A,B,C,DISC,Z(IP),Z(IQ),
  + PN1,QN1,PN2,QN2
590 FORMAT(/4X,'QUAD EQN FOR npl'
                2X, 'A=', E12.5,1X, 'B=', E12.5,1X, 'C=', E12.5,1X, 'DISC=',
E12.5/4X, 'Z(IP)=', F6.4,2X, 'Z(IQ)=', F6.4/4X, 'PN1=',
F7.5,2X, 'QN1=', F7.5,2X, 'PN2=', F7.5,2X, 'QN2=', F7.5)
      +
      +
       IF (N .EQ. 2) GO TO 110
* The mole numbers of the remaining components (one at the time)
* are obtained again from the equilibrium and mass balance equations
* assuming only p and q are present
       XT(1)=PN1/DEN1
       XT(2)=QN1/DEN1
       XT(3)=1.D-6
       XS(1)=PN2/DEN2
       XS(2) = QN2/DEN2
       XS(3)=1.D-6
       IF (IFLG(15) .EQ. 2) WRITE(6,600)
  600 FORMAT(/4X,'i FUG COEFF OF i AT INF
+ /4X,' DILUTION IN LIQUID 1
                                                            FUC COEFF OF i AT INF'
                                                            DILUTION IN LIQUID 2')
```

```
DO 110 I=1,N
         IF (I .EQ. IP .OR. I .EQ. IQ) GO TO 110
TCT(3)=TC(I)
         PCT(3)=PC(I)
         WT(3)=W(I)
         PPT(3)=PP(I)
         APT(3)=APUR(I)
         BPT(3)=BPUR(I)
         CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 3, 2, 3, APT, BPT, FUCLT, FUCD,
                 ZT,ZD,KIJ,KEQ,IFLG)
     +
         FUCR1=FUCLT(3)
         CALL K(XD,XS,TCT,PCT,WT,PPT,P,T,3,2,3,APT,BPT,FUCLT,FUCD,
                 ZT,ZD,KIJ,KEQ,IFLG)
         FUCR2=FUCLT(3)
        IF (IFLG(15) .EQ. 2) WRITE(6,610) I,FUCR1,FUCR2
FORMAT(3X,I2,' FUCR1=',F10.6,' FUCR2=',I
  610
                                                      FUCR2=',F10.6)
         A=FUCR1-FUCR2
         B=FUCR2*(Z(I)-DEN1)-FUCR1*(Z(I)+DEN2)
         C=FUCR2*Z(I)*DEN1
         IF (A .EQ. 0.DO) THEN
RN1(I)=-C/B
           GO TO 90
         ELSE IF (C .EQ. 0.DO) THEN
RN1(I)=-B/A
           GO TO 90
         ELSE
           DISC=B*B-4.DO*A*C
           IF (DISC .LT. O.DO) THEN
             WRITE(6,620)
FORMAT(/6X,'*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
  620
             /9X,'TO SOLVE THE QUADRATIC EQUATION FOR nrI IN'
/9X,'SUBROUTINE INSHAH')
RN1(I)=Z(I)/2.DO
     .
             GO TO 100
           ELSE IF (DISC .EQ. O.DO) THEN
             RN1(I) = -B/(2.DO*A)
             GO TO 90
           END IF
         END IF
         TWA=2.DO*A
         RTDISC=DSQRT(DISC)
         RN1(I)=(-B+RTDISC)/TWA
         IF (RN1(I) .GT. 0.DO .AND. RN1(I) .LT. Z(I)) GO TO 100
         RN1(I) = (-B - RTDISC)/TWA
        IF (RN1(I) .GT. 0.D0 .AND. RN1(I) .LT. Z(I)) GO TO 100
IF (RN1(I) .LE. 0.D0) RN1(I)=0.1D0*Z(I)
IF (RN1(I) .GE. Z(I)) RN1(I)=0.9D0*Z(I)
   90
         RN2(I)=Z(I)-RN1(I)
  100
         SMRN1=SMRN1+RN1(I)
         SMRN2=SMRN2+RN2(I)
        630
  110 CONTINUE
* Calculation of the initial composition estimates
```

TOTN1=DEN1+SMRN1 TOTN2=DEN2+SMRN2 XI(1,IP)=PN1/TOTN1 XI(1,IQ)=QN1/TOTN1 XI(2,IP)=PN2/TOTN2

```
XI(2,IQ)=QN2/TOTN2
IF (IFLG(15) .EQ. 2) WRITE(6,640) DEN1,DEN2,SMRN1,SMRN2,
+ TOTN1,TOTN2
640 FORMAT(/4X,'DEN1=',F8.4,1X,'DEN2=',F8.4
+ /4X,'SMRN1=',F8.4,1X,'SMRN2=',F8.4
+ /4X,'TOTN1=',F8.4,1X,'TOTN2=',F8.4
DO 120 I=1,N
IF (I .EQ. IP .OR. I .EQ. IQ) GO TO 120
XI(1,I)=RN1(I)/TOTN1
XI(2,I)=RN2(I)/TOTN2
120 CONTINUE
RETURN
END
```

```
SUBROUTINE INGASE (INIT. OF LIQ. COMPS. BY A MODIFIED GAUTAM
*
                       & SEIDER METHOD)
                                                                  *
*
***********
* Purpose:
 Initialize liquid compositions to be used in subroutine LIQSER by *
* a simplified version of the method proposed by Gautam & Seider to
* obtain the phase initiator components p and q in liquids 1 and 2
*
 respectively.
* Parameters:
 -Input
 Z, TC, PC, W, PP, P, T, N, APUR, BPUR, KIJ, IFLG, ALSL as in program PHASPLIT
* -Output
 IP, IQ, XI as in subroutine INSHAH
* Main variables:
         Activity if trial component i in a binary mixture with
*
  ACT
  component p
APT,BPT,FUCP1,FUCP2,FUCQ1,FUCQ2,FUCR1,FUCR2,PCT,PPT,PN1,PN2,QN1,
  QN2,RN1,RN2,TOTN1,TOTN2,WT,XS,XT as in subroutine INSHAH
         Remaining variables defined as in program PHASPLIT
*
*
 Subroutines called:
×
  K
*
 Called by:
  PHASPLIT
*******
     SUBROUTINE INGASE(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,
                       ALSL, IP, IQ, XI)
    +
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL*8 KEQ(3),KIJ(20,20)
     DIMENSION ALSL(20), APT(3), APUR(20), BPT(3), BPUR(20), FUCD(3),
               FUCLT(3), IFLG(0:20), PC(20), PCT(3), PP(20), PPT(3),
               RN1(18), RN2(18), TC(20), TCT(3), W(20), WT(3), XD(3),
     - XI(21,20),XS(3),XT(3),Z(20)
IF (IFLG(15) .EQ. 2) WRITE(6,500)
  500 FORMAT(/4X,'SUBROUTINE INGASE TRACING')
     IF (IFLG(15) .EQ. 2) WRITE(6,510)
 510 FORMAT(/4X,'ACTIVITY OF COMP. i IN THE ORIGINAL MIXTURE'
+ /4X,'i',10X,'ACTIVITY')
* Selection of phase initiator p as the component with the highest
* activity in the original mixture
     HAL=ALSL(1)
     IP=1
     DO 10 I=2,N
       IF (HAL .LT. ALSL(I)) THEN
         HAL=ALSL(I)
         IP=I
       END IF
   10 CONTINUE
      IF (IFLG(15) .EQ. 2) WRITE(6,520) (I,ALSL(I), I=1,N)
  520 FORMAT(3X,12,9X,F10.6)
```

```
* Selection of phase initiator q as the component with the highest
* binary activity with component p, considering a composition
* proportional to that of the original mixture
      HAT=0.DO
      TCT(1)=TC(IP)
      PCT(1)=PC(IP)
      WT(1)=W(IP)
      PPT(1)=PP(IP)
       APT(1)=APUR(IP)
      BPT(1)=BPUR(IP)
      IF (IFLG(15) .EQ. 2) WRITE(6,530)
FORMAT(/4X,'i FUG COEFF OF i IN'
  530 FORMAT(/4X,'i
+ /4X,'
                         BINARY MIXTURE WITH p')
      DO 20 I=1,N
         IF (I .EQ. IP) GO TO 20
XT(1)=Z(IP)/(Z(I)+Z(IP))
         XT(2)=1.DO-XT(1)
         TCT(2)=TC(1)
         PCT(2)=PC(I)
WT(2)=W(I)
         PPT(2)=PP(I)
         APT(2) = APUR(I)
         BPT(2)=BPUR(I)
         CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
                 ZT,ZD,KIJ,KEQ,IFLG)
     +
         ACT=XT(2)*FUCLT(2)*P
         IF (IFLG(15) .EQ. 2) WRITE(6,540) I,FUCLT(2)
         FORMAT(3X,12,8X,F10.6)
IF (HAT .LE. ACT) THEN
  540
           HAT=ACT
           IQ=I
         END IF
   20 CONTINUE
* Calculation of the pure and infinite-dilution fug. coeffs. for
* components p and q in phases 1 and 2
       XT(2)=1.D-6
XT(1)=1.D0-XT(2)
       TCT(2)=TC(IQ)
       PCT(2)=PC(IQ)
       WT(2)=W(IQ)
      PPT(2) = PP(IQ)
       APT(2) = APUR(IQ)
       BPT(2)=BPUR(IQ)
      CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
     +
               ZT, ZD, KIJ, KEQ, IFLG)
* Assignment of fug. coeffs. of 'pure' p in phase 1 and of q at
* infinite-dilution in phase 1
       FUCP1=FUCLT(1)
       FUCQ1=FUCLT(2)
       XT(1)=1.D-6
       XT(2)=1.DO-XT(1)
       CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
               ZT,ZD,KIJ,KEQ,IFLG)
* Assignment of fug. coeffs. of p at infinite-dilution in phase 2
* and of 'pure' q in phase 2
```

```
FUCP2=FUCLT(1)
      FUCQ2=FUCLT(2)
* Calculation of the mole numbers of p and q from the equilibrium
* and mass balance equations assuming no other components are
* present
      A0=FUCP1-FUCP2
      B0=Z(IP)*(FUCP2-2.D0*FUCP1)
      CO=Z(IP)*FUCP2
      D0=FUC02-FUC01
      E0=2.D0+Z(IP)+FUCQ1-Z(IQ)+FUCQ2
      FO = -Z(IQ) * FUCQ2
       A=AO*(DO*(BO-CO)-AO*(EO-FO))
      B=B0*D0*(B0-C0)-A0*(E0*(C0+B0)-2.D0*C0*F0)
       C=CO*(CO*FO-BO*EO)
      IF (A .EQ. O.DO) THEN
PN1=-C/B
         GO TO 30
      ELSE IF (C .EQ. O.DO) THEN
         PN1=-B/A
         GO TO 30
      ELSE
         DISC=B*B-4.DO*A*C
         IF (DISC .LT. 0.DO) THEN
           WRITE(6,550)
  550
           FORMAT(/6X, '*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
                   /9X,'TO SOLVE THE QUADRATIC EQUATION FOR npI IN'
/9X,'SUBROUTINE INGASE')
     +
           PN1=Z(IP)/2.DO
           GO TO 40
         ELSE IF (DISC .EQ. 0.DO) THEN
           PN1=-B/(2.D0*A)
           GO TO 30
         END IF
      END IF
       TWA=2.DO*A
      RTDISC=DSQRT(DISC)
      PN1=(-B+RTDISC)/TWA
      PNO=(-B-RTDISC)/TWA
       IF (PN1 .GT. 0.DO .AND. PN1 .LT. Z(IP)) THEN
         QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
      IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) GO TO 60
ELSE IF (PNO .GT. O.DO .AND. PNO .LT. Z(IP)) THEN
         PN1=PNO
         QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
         IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) THEN
           GO TO 60
         ELSE
           GO TO 50
         END IF
       END IF
   30 IF (PN1 .LE. 0.D0) PN1=0.1D0*Z(IP)
IF (PN1 .GE. Z(IP)) PN1=0.9D0*Z(IP)
   40 QN1=-PN1*(A0*PN1+B0)/(A0*PN1+C0)
   50 IF (QN1 .LE. 0.D0) QN1=0.1D0+Z(IQ)
   IF (QN1 .GE. Z(IQ)) QN1=0.9D0*Z(IQ)
60 PN2=Z(IP)-PN1
       QN2=Z(IQ)-QN1
       SMRN1=0.DO
       SMRN2=0.DO
      DEN1=PN1+QN1
      DEN2=PN2+QN2
```

```
IF (IFLG(15) .EQ. 2) WRITE(6,560)A,B,C,DISC,Z(IP),Z(IQ),
  + PN1,QN1,PN2,QN2
560 FORMAT(/4X,'QUAD EQN FOR npI'
                2X, 'A=', E12.5, 1X, 'B=', E12.5, 1X, 'C=', E12.5, 1X, 'DISC=',
E12.5/4X, 'Z(IP)=', F6.4, 2X, 'Z(IQ)=', F6.4/4X, 'PN1=',
      +
      +
                F7.5,2X,'QN1=',F7.5,2X,'PN2=',F7.5,2X,'QN2=',F7.5)
      +
       IF (N .EQ. 2) GO TO 90
* The mole numbers of the remaining components (one at the time)
* are obtained again from the equilibrium and mass balance equations
* assuming only p and q are present
       XT(1)=PN1/DEN1
       XT(2)=QN1/DEN1
       XT(3)=1.D-6
       XS(1)=PN2/DEN2
       XS(2)=QN2/DEN2
       XS(3)=1.D-6
       IF (IFLG(15) .EQ. 2) WRITE(6,570)
  570 FORMAT(/4X,'i
+ /4X,'
                         FUG COEFF OF i AT INF
                                                         FUC COEFF OF i AT INF'
                          DILUTION IN LIQUID 1
                                                         DILUTION IN LIQUID 2')
       DO 90 I=1,N
          IF (I .ÉQ. IP .OR. I .EQ. IQ) GO TO 90
TCT(3)=TC(I)
          PCT(3)=PC(I)
          WT(3)=W(I)
          PPT(3)=PP(I)
          APT(3)=APUR(I)
          BPT(3)=BPUR(I)
          CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 3, 2, 3, APT, BPT, FUCLT, FUCD,
         ZT,ZD,KIJ,KEQ,IFLG)
FUCR1=FUCLT(3)
          CALL K(XD,XS,TCT,PCT,WT,PPT,P,T,3,2,3,APT,BPT,FUCLT,FUCD,
                  ZT, ZD, KIJ, KEQ, IFLG)
          FUCR2=FUCLT(3)
         IF (IFLG(15) .EQ. 2) WRITE(6,580) I,FUCR1,FUCR2
FORMAT(3X,I2,' FUCR1=',F10.6,' FUCR2=',I
                             FUCR1=',F10.6,'
  580
                                                         FUCR2=', F10.6)
          A=FUCR1-FUCR2
          B=FUCR2*(Z(I)-DEN1)-FUCR1*(Z(I)+DEN2)
          C=FUCR2*Z(I)*DEN1
IF (A .EQ. 0.D0) THEN
            RN1(I) = -C/B
            GO TO 70
          ELSE IF (C .EQ. O.DO) THEN
            RN1(I) = -B/A
            GO TO 70
          ELSE
            DISC=B*B-4.DO*A*C
            IF (DISC .LT. 0.DO) THEN
               WRITE(6,590)
  590
               FORMAT(/6X,'*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
                       /9X,'TO SOLVE THE QUADRATIC EQUATION FOR nrI IN'
/9X,'SUBROUTINE INGASE')
      +
      +
               RN1(I)=Z(I)/2.DO
               GO TO 80
            ELSE IF (DISC .EQ. 0.DO) THEN
               RN1(I) = -B/(2.DO*A)
               GO TO 70
            END IF
          END IF
          TWA=2.DO*A
          RTDISC=DSQRT(DISC)
          RN1(I)=(-B+RTDISC)/TWA
```

```
IF (RN1(I) .GT. 0.DO .AND. RN1(I) .LT. Z(I)) GO TO 80
          RN1(I) = (-B - RTDISC)/TWA
         IF (RN1(I) .GT. 0.D0 .AND. RN1(I) .LT. Z(I)) GO TO 80
IF (RN1(I) .LE. 0.D0) RN1(I)=0.1D0*Z(I)
IF (RN1(I) .GE. Z(I)) RN1(I)=0.9D0*Z(I)
RN2(I)=Z(I)-RN1(I)
   70
   80
          SMRN1=SMRN1+RN1(I)
          SMRN2=SMRN2+RN2(I)
         600
      +
   90 CONTINUE
* Calculation of the initial composition estimates
       TOTN1=DEN1+SMRN1
       TOTN2=DEN2+SMRN2
       XI(1,IP)=PN1/TOTN1
XI(1,IQ)=QN1/TOTN1
       XI(2, IP)=PN2/TOTN2
       XI(2,IQ)=QN2/TOTN2
       IF (IFLG(15) .EQ. 2) WRITE(6,610) DEN1, DEN2, SMRN1, SMRN2,
  + TOTN1, TOTN2
610 FORMAT(/4X, 'DEN1=', F8.4, 1X, 'DEN2=', F8.4
+ /4X, 'SMRN1=', F8.4, 1X, 'SMRN2=', F8.4
+ /4X, 'TOTN1=', F8.4, 1X, 'TOTN2=', F8.4)
       DO 100 I=1,N
          IF (I .EQ. IP .OR. I .EQ. IQ) GO TO 100
XI(1,I)=RN1(I)/TOTN1
          XI(2,I)=RN2(I)/TOTN2
  100 CONTINUE
       RETURN
       END
```

```
SUBROUTINE INLIQ (INITIALIZATION OF LIQ. COMPOSITIONS BY THE
*
                         PROPOSED METHOD)
                   ******
* Purpose:
*
   Initialize liquid compositions to be used in subroutine LIQSER.
* The proposed method selects the first phase initiator p as the
* component with the highest activity in the original system as in
* the Gautam & Seider method. This is the key comp. for phase 1. The *
* second phase initiator Q (key comp. for phase 2) is that one with *
* the largest pseudo-infinite dilution activity in pure comp. p *
* obtained in the form proposed by Shah. *
* Parameters:
*
  -Input
*
 Z, TC, PC, W, PP, P, T, N, APUR, BPUR, KIJ, IFLG, ALSL as in program PHASPLIT
*
*
  -Output
*
  IP,ÎQ,XI as in subroutine INSHAH
*
  Main variables:
  ACT, APT, BPT, FUCP1, FUCP2, FUCQ1, FUCQ2, FUCR1, FUCR2, PCT, PPT, PN1, PN2,
   QN1, QN2, RN1, RN2, TOTN1, TOTN2, WT, XS, XT as in subroutine INSHAH
           Remaining variables defined as in program PHASPLIT
*
  Subroutines called:
*
*
   K
*
  Called by:
  PHASPLIŤ
******
      SUBROUTINE INLIQ(Z,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG,
                          ALSL, IP, IQ, XI)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KEQ(2), KIJ(20,20)
      DIMENSION ALSL(20), APT(3), APUR(20), BPT(3), BPUR(20), FUCD(3),
FUCLT(3), IFLG(0:20), PC(20), PCT(3), PP(20), PPT(3),
     +
                 RN1(18), RN2(18), TC(20), TCT(3), W(20), WT(3), XD(3),
     +
                 XI(21,20), XS(3), XT(3), Z(20)
     +
  IF (IFLG(15) .EQ. 2) WRITE(6,500)
500 FORMAT(/4X,'SUBROUTINE INLIQ TRACING')
      IF (IFLG(15) .EQ. 2) WRITE(6,510)
  510 FORMAT(/4X,'ACTIVITY OF COMP. I IN THE ORIGINAL MIXTURE'
              /4X, 'I', 10X, 'ACTIVITY')
* Selection of phase initiator p as the component with the highest
* activity in the original mixture
      HAL=ALSL(1)
      IP=1
      DO 10 I=2,N
         IF (HAL .LT. ALSL(I)) THEN
           HAL=ALSL(I)
           IP=I
         END IF
   10 CONTINUE
      IF (IFLG(15) .EQ. 2) WRITE(6,520) (I,ALSL(I), I=1,N)
```

520 FORMAT(3X,12,9X,F10.6)

```
* Selection of phase initiator q as the component with the largest
* pseudo-infinite dilution activity in pure component p
      HAT=0.DO
      TCT(1)=TC(IP)
      PCT(1)=PC(IP)
      WT(1)=W(IP)
      PPT(1)=PP(IP)
      APT(1)=APUR(IP)
      BPT(1)=BPUR(IP)
      XT(2)=1.D-6
      XT(1)=1.DO-XT(2)
  IF (IFLG(15) .EQ. 2) WRITE(6,530)
530 FORMAT(/4X,'i FUG COEFF OF PURE p
+ /4X,' WITH TRACE OF COMP
                                                       FUC COEFF OF i AT'
                          WITH TRACE OF COMP i
                                                      INF DILUTION IN p')
      DO 20 I=1,N
         IF (I .EQ. IP) GO TO 20
TCT(2)=TC(I)
         PCT(2)=PC(I)
         WT(2)=W(I)
         PPT(2)=PP(I)
         APT(2)=APUR(I)
         BPT(2)=BPUR(1)
         CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
     +
                 ZT,ZD,KIJ,KEQ,IFLG)
         ACT=Z(I)*FUCLT(2)*P
         IF (IFLG(15) .EQ. 2) WRITE(6,540) I,FUCLT(1),FUCLT(2)
         FORMAT(3X,12,8X,F10.6,15X,F10.6)
  540
         IF (HAT .LE. ACT) THEN
           HAT=ACT
           IQ=I
* Assignment of fugacity coefficients of 'pure' p in phase 1 and
* of q at infinite-dilution in phase 1
           FUCP1=FUCLT(1)
           FUCQ1=FUCLT(2)
         END IF
   20 CONTINUE
       IF (IFLG(15) .EQ. 2) WRITE(6,550) FUCP1,FUCQ1
FORMAT(4X,' FUCP1=',F10.6,' FUCQ1='
  550 FORMAT(4X,'
                                                  FUCQ1=',F10.6)
       XT(1)=1.D-6
       XT(2)=1.DO-XT(1)
       TCT(2)=TC(IQ)
PCT(2)=PC(IQ)
       WT(2)=W(IQ)
       PPT(2)=PP(IQ)
       APT(2)=APUR(IQ)
       BPT(2)=BPUR(IQ)
       CALL K(XD, XT, TCT, PCT, WT, PPT, P, T, 2, 2, 3, APT, BPT, FUCLT, FUCD,
       ZT,ZD,KIJ,KEQ,IFLG)
IF (IFLG(15) .EQ. 2) WRITE(6,560)
                         FUG COEFF OF P AT
INF DILUTION IN q
                                                      FUC COEFF OF PURE q'
  560 FORMAT(/4X,'
               /4X,'
                                                      WITH TRACE OF COMP p')
      +
       IF (IFLG(15) .EQ. 2) WRITE(6,570) FUCLT(1), FUCLT(2)
                      FUCP2=',F10.6,'
  570 FORMAT(4X,'
                                                   FUCQ2=',F10.6)
* Assignment of fugacity coefficients of p at infinite-dilution in
* phase 2 and of 'pure' q in phase 2
       FUCP2=FUCLT(1)
```

```
FUC02=FUCLT(2)
* Calculation of the mole numbers of p and q from the equilibrium
* and mass balance equations assuming no other components are
* present
      A0=FUCP1-FUCP2
      B0=Z(IP)*(FUCP2-2.D0*FUCP1)
      CO=Z(IP)*FUCP2
      D0=FUCQ2-FUCQ1
      E0=2.D0*Z(IP)*FUCQ1-Z(IQ)*FUCQ2
      FO = -Z(IQ) * FUCQ2
      A=A0*(D0*(B0-C0)-A0*(E0-F0))
      B=B0*D0*(B0-C0)-A0*(E0*(C0+B0)-2.D0*C0*F0)
      C=CO*(CO*FO-BO*EO)
      IF (A .EQ. O.DO) THEN
        PN1=-C/B
        GO TO 30
      ELSE IF (C .EQ. O.DO) THEN
        PN1 = -B/A
        GO TO 30
      ELSE
        DISC=B*B-4.DO*A*C
        IF (DISC .LT. O.DO) THEN
           WRITE(6,580)
  580
          FORMAT(/6X, '*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
                  /9X,'TO SOLVE THE QUADRATIC EQUATION FOR npI IN'
                  /9X, 'SUBROUTINE INLIQ')
     +
          PN1=Z(IP)/2.D0
           GO TO 40
        ELSE IF (DISC .EQ. 0.DO) THEN
           PN1=-B/(2.DO*A)
           GO TO 30
        END IF
      END IF
      TWA=2.DO+A
      RTDISC=DSQRT(DISC)
      PN1=(-B+RTDISC)/TWA
PNO=(-B-RTDISC)/TWA
      IF (PN1 .GT. O.DO .AND. PN1 .LT. Z(IP)) THEN
        QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
         IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) GO TO 60
      ELSE IF (PNO .GT. O.DO .AND. PNO .LT. Z(IP)) THEN
        PN1=PN0
        QN1 = -PN1 * (AO * PN1 + BO) / (AO * PN1 + CO)
         IF (QN1 .GT. O.DO .AND. QN1 .LT. Z(IQ)) THEN
           GO TO 60
        ELSE
           GO TO 50
        END IF
      END IF
   30 IF (PN1 .LE. 0.D0) PN1=0.1D0*Z(IP)
IF (PN1 .GE. Z(IP)) PN1=0.9D0*Z(IP)
   40 QN1=-PN1*(A0*PN1+B0)/(A0*PN1+C0)
   50 IF (QN1 .LE. 0.D0) QN1=0.1D0*Z(IQ)
      IF (QN1 .GE. Z(IQ)) QN1=0.9D0*Z(IQ)
   60 PN2=Z(IP)-PN1
      QN2=Z(IQ)-QN1
      SMRN1=0.DO
      SMRN2=0.DO
      DEN1=PN1+QN1
      DEN2=PN2+QN2
      IF (IFLG(15) .EQ. 2) WRITE(6,590)A,B,C,DISC,Z(IP),Z(IQ),
```

```
PN1,QN1,PN2,QN2
      +
  590 FORMAT(/4X, 'QUAD EQN FOR npI',

+ 2X, 'A=', E12.5, 1X, 'B=', E12.5, 1X, 'C=', E12.5, 1X, 'DISC=',

+ E12.5/4X, 'Z(IP)=', F6.4, 2X, 'Z(IQ)=', F6.4/4X, 'PN1=',

+ F7.5, 2X, 'QN1=', F7.5, 2X, 'PN2=', F7.5, 2X, 'QN2=', F7.5)
       IF (N .EQ. 2) GO TO 90
* The mole numbers of the remaining components (one at the time)
* are obtained again from the equilibrium and mass balance equations
* assuming only p and q are present
       XT(1)=PN1/DEN1
       XT(2)=QN1/DEN1
       XT(3)=1.D-6
       XS(1)=PN2/DEN2
       XS(2)=QN2/DEN2
       XS(3)=1.D-6
       IF (IFLG(15) .EQ. 2)WRITE(6,600)
  600 FORMAT(/4X,'i
+ /4X,'
                            FUG COEFF OF i AT INF
                                                            FUC COEFF OF i AT INF'
                            DILUTION IN LIQUID 1
                                                            DILUTION IN LIQUID 2')
       DO 90 I=1,N
          IF (I .EQ. IP .OR. I .EQ. IQ) GO TO 90
TCT(3)=TC(I)
          PCT(3)=PC(I)
          WT(3)=W(I)
          PPT(3)=PP(I)
          APT(3)=APUR(I)
          BPT(3)=BPUR(I)
          CALL K(XD,XT,TCT,PCT,WT,PPT,P,T,3,2,3,APT,BPT,FUCLT,FUCD,
                   ZT,ZD,KIJ,KEQ,IFLG)
      +
          FUCR1=FUCLT(3)
          CALL K(XD,XS,TCT,PCT,WT,PPT,P,T,3,2,3,APT,BPT,FUCLT,FUCD,
      +
                   ZT,ZD,KIJ,KEQ,IFLG)
          FUCR2=FUCLT(3)
          IF (IFLG(15) .EQ. 2) WRITE(6,610) I,FUCR1,FUCR2
FORMAT(3X,I2,' FUCR1=',F10.6,' FUCR2=',I
  610
                                                            FUCR2=',F10.6)
          A=FUCR1-FUCR2
          B=FUCR2*(Z(I)-DEN1)-FUCR1*(Z(I)+DEN2)
          C=FUCR2*Z(I)*DEN1
          IF (A .EQ. O.DO) THEN
             RN1(I) = -C/B
          GO TO 70
ELSE IF (C .EQ. 0.DO) THEN
             RN1(I) = -B/A
             GO TO 70
          ELSE
             DISC=B*B-4.DO*A*C
             IF (DISC .LT. 0.DO) THEN
               WRITE(6,620)
               FORMAT(/6X, '*** WARNING: NO REAL ROOTS FOUND WHEN TRYING'
  620
               /9X,'TO SOLVE THE QUADRATIC EQUATION FOR nrI IN'
/9X,'SUBROUTINE INLIQ')
RN1(I)=Z(I)/2.DO
      +
                GO TO 80
             ELSE IF (DISC .EQ. 0.DO) THEN
                RN1(I) = -B/(2.DO*A)
                GO TO 70
             END IF
          END IF
          TWA=2.DO*A
          RTDISC=DSQRT(DISC)
          RN1(I)=(-B+RTDISC)/TWA
          IF (RN1(I) .GT. 0.DO .AND. RN1(I) .LT. Z(I)) GO TO 80
```

```
RN1(I)=(-B-RTDISC)/TWA
          IF (RN1(I) .GT. 0.D0 .AND. RN1(I) .LT. Z(I)) GO TO 80
IF (RN1(I) .LE. 0.D0) RN1(I)=0.1D0*Z(I)
IF (RN1(I) .GE. Z(I)) RN1(I)=0.9D0*Z(I)
RN2(I)=Z(I)-RN1(I)
   70
   80
          SMRN1=SMRN1+RN1(I)
          SMRN2=SMRN2+RN2(I)
          630
      +
   90 CONTINUE
* Calculation of the initial composition estimates
       TOTN1=DEN1+SMRN1
       TOTN2=DEN2+SMRN2
       XI(1, IP)=PN1/TOTN1
       XI(1,IQ)=QN1/TOTN1
XI(2,IP)=PN2/TOTN2
       XI(2,IQ)=QN2/TOTN2
       IF (IFLG(15) .EQ. 2) WRITE(6,640) DEN1, DEN2, SMRN1, SMRN2, TOTN1,
             TOTN2
      +
  640 FORMAT(/4X, 'DEN1=', F8.4, 1X, 'DEN2=', F8.4
+ /4X, 'SMRN1=', F8.4, 1X, 'SMRN2=', F8.4
+ /4X, 'TOTN1=', F8.4, 1X, 'TOTN2=', F8.4)
       DO 100 I=1,N
          IF (I .EQ. IP .OR. I .EQ. IQ) GO TO 100
XI(1,I)=RN1(I)/TOTN1
          XI(2,I)=RN2(I)/TOTN2
  100 CONTINUE
       RETURN
       END
```

```
**********
             SUBROUTINE VAPSER (VAPOUR PHASE SEARCH)
*:
   * Purpose:
 Search for an incipient vapour phase forming from the original system when it is liquid-like. The method used is based on the
*
 stability analysis proposed by Michelsen as modified in this work *
*
 Parameters:
*
 -Input
  EPS, ITMAX, IWEG, TIMAX, N, P, T, Z, YI, TC, PC, W, PP, APUR, BPUR, KIJ, IFLG as
  defined in program PHASPLIT
          Vector of the logarithm of the product of the feed comp.
*
   HL
          and the fug. coeff. for the original liquid-like mixture
*
 -Output:
   ITNUM Number of iterations required in the calculation
*
          Value of the objective function, defined as the sum of
*
   OBJ
          absolute value of the composition difference in two
          consecutive iterations
   FUCV
*
          Vector of fugacity coeffs. of the incipient vapour phase
                     compositions
   Y
*
 Main variables:
*
   BET
          Value of the gradient defined by Michelsen
          Vector of comp. vars. big Y (equivalent to mole numbers)
  BY
*
*
  BYN
          Vector of updated values of comp. vars. big Y
  BYO
*
                     old
            11
                 ....
*
   EXHL
                     values of the exponential of HL
            **
                 " the new values of big Y obtained by direct
*
   FBY
          substitution
*
   GSTR
          Value of g^*, which constitutes the stability criterion;
if g^{*>=0} the system is stable; if g^{*<0} it's unstable
*
          Value of GSTR in the previous iteration
Control variable for the application of Wegstein's method
*
   GSTRO
*
   ITCRL
          with a frequency determined by the variable IWEG
*
   JCF
*
          Control variable to check for convergence of the comp.
*
          vector to the feed comp.
   JCOB
*
          Control variable for the application of Wegstein's method
          in case the method causes the objective function to
*
          increase
*
   JDGST Control variable to check if g*, when negative, is not
*
          changing significantly in consecutive iterations
*
   JRS
          Control variable to detect if RSM is approaching 1 in
*
          consecutive iters.
*
   JIOB
          Control variable to check if the objective function is
*
          increasing in consecutive iters.
   JSMBY Control variable to spot if SMBYN, when less than 1, is
*
          decreasing in consecutive iters.
   OBJO
*
          Value of the objective function in the previous iteration
*
   RSM
          Gradient ratio r, defined by Michelsen, which is used to
*
          detect the approach to a trivial solution
          Value of RSM in the previous iteration
   RSMO
*
  SMBYN Sum of the values of BYN
*
*
   SMBYNO Value of SMBYN in the previous iteration
*
   SMSTAT Sum of the stationarity criterion for all components;
          approaches zero when a stationary point is reached
*
   STAT
          Value of the stationarity criterion for each component as *
          defined by Michelsen
```

```
TI
           Step size parameter in Wegstein's method. If TI=1, the
*
                                                                           *
          method reduces to direct substitution
*
                                                                           *
*
*
                                                                           *
           Remaining variables defined as in program PHASPLIT
*
                                                                           *
                                                                           *
*
 Subroutines called:
                                                                           *
*
  K
                                                                           *
*
 Called by:
                                                                           ×
*
   PHASPLIT
                                                                           *
SUBROUTINE VAPSER(EPS, ITMAX, IWEG, TIMAX, N, P, T, Z, YI, HL, TC, PC,
                          W, PP, APUR, BPUR, KIJ, ITNUM, OBJ, FUCV, Y, IFLG)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KEQ(20),KIJ(20,20)
      DIMENSION APUR(20), BPUR(20), BY(20), BYN(20), BYO(20), EXHL(20)
                 FBY(20),FUCD(20),FUCV(20),HL(20),IFLG(0:20),PC(20),
PP(20),TC(20),W(20),XD(20),Y(20),YI(20),Z(20)
     +
     +
      ITNUM=0
      ITCRL=0
      JCOB=0
* The initial values of BY are set and parameter EXP(Hiliq) is
* obtained
      DO 10 I=1,N
        BYO(I) = \dot{Y}I(I)
        EXHL(I)=DEXP(HL(I))
   10 CONTINUE
      IF (IFLG(15) .NE. 2) GO TO 20
  WRITE(6,500)
WRITE(6,500)
500 FORMAT(/4X,'INITIAL y COMPOSITIONS USING RADULTS LAW')
WRITE(6,510) (I,YI(I), I=1,N)
510 FORMAT(8X,'y(',I2,')=',F8.5)
* The fugacity coefficient for the vapour corresponding to the
* initial composition is evaluated
   20 CALL K(XD,YI,TC,PC,W,PP,P,T,N,1,3,APUR,BPUR,FUCV,FUCD,Z1,ZD,
              KIJ,KEQ,IFLG)
* Calculation of a new value of BY using direct substitution
      SUM=0
      DO 30 I=1,N
        BY(I)=EXHL(I)/FUCV(I)
         SUM=SUM+BY(I)
   30 CONTINUE
* Computation of the corresponding vapour composition (mole fraction)
      DO 40 I=1,N
        Y(I) = BY(I) / SUM
   40 CONTINUE
      IF (IFLG(15) .NE. 2) GO TO 50
      WRITE(6,520)
  520 FORMAT(/4X, 'y COMPOSITIONS WITH DIRECT SUBSTITUTION')
      WRITE(6,510) (I,Y(I), I=1,N)
* Evaluation of the fugacity coefficients corresponding to this
* composition
```

```
50 CALL K(XD,Y,TC,PC,W,PP,P,T,N,1,3,APUR,BPUR,FUCV,FUCD,Z1,ZD,
              KIJ,KEQ,IFLG)
* The value of F(BY) to be used in Wegstein's method is obtained
      DO 60 I=1,N
        FBY(I)=EXHL(I)/FUCV(I)
   60 CONTINUE
* Start of the iterative cycle
   70 ITNUM=ITNUM+1
      SMBYN=0.DO
      IF (IFLG(15) .NE. 0) WRITE(6,530) ITNUM
  530 FORMAT(/4X, '*** ITERATION NO.= ', 12/)
* Wegstein's method is used to obtain updated BY values
      IF (ITNUM .LT. 4 .OR. ITNUM .LT. ITCRL+IWEG .OR.
IWEG .EQ. 0) THEN
        TI=1.DO
      ELSE
        IF (IFLG(15) .NE. 0) WRITE (6,540)
        FORMAT(4X, 'WEGSTEIN METHOD USED')
  540
        TI=0.D0
      END IF
      DO 100 I=1,N
        IF (TI .EQ. 1.DO) GO TO 90
SI=(FBY(I)-BY(I))/(BY(I)-BYO(I))
        TI=1.D0/(1.D0-SI)
* Limitation of the value of parameter TI to avoid overshooting
         IF (TI .GT. TIMAX) TI=TIMAX
         IF (TI .LT. -TIMAX) TI=-TIMAX
         IF (IFLG(15) .NE. 2) GO TO 80
        IF (TI .EQ. TIMAX) THEN
WRITE(6,550) I
  550
           FORMAT(6X, 'TI EQUALS TIMAX FOR COMPONENT ',12)
         ELSE IF (TI .EQ. -TIMAX) THEN
           WRITE(6,560) I
  560
           FORMAT(6X,'TI EQUALS -TIMAX FOR COMPONENT ',12)
        END IF
   80
         ITCRL=ITNUM
         BYN(I)=(1.DO-TI)*BY(I)+TI*FBY(I)
   90
         IF (BYN(I) .LT. 0.D0) BYN(I)=1.D-6
         SMBYN=SMBYN+BYN(I)
  100 CONTINUE
* The new vapour composition is calculated
      DO 120 I=1,N
         Y(I)=BYN(I)/SMBYN
  120 CONTINUE
      IF (IFLG(15) .NE. 2) GO TO 130
      WRITE(6,570)
  570 FORMAT(4X,'I',5X,'y',9X,'BYO',9X,'BY',9X,'FBY',9X,'BYN')
WRITE(6,580) (I,Y(I),BYO(I),BY(I),FBY(I),BYN(I), I=1,N)
  580 FORMAT(3X,12,2X,F8.5,3X,F8.5,4X,F8.5,4X,F8.5,4X,F8.5)
```

```
* Evaluation of the corresponding fugacity coefficients
  130 CALL K(XD,Y,TC,PC,W,PP,P,T,N,1,3,APUR,BPUR,FUCV,FUCD,Z1,ZD,
              KIJ, KEQ, IFLG)
* Calculation of the objective function and of parameter r
      SUM=0.D0
      BET=0.DO
      OBJ=0.DO
      SMSTAT=0.D0
      DO 140 I=1,N
        OBJ=OBJ+DABS(BYN(I)-BY(I))
        STAT=DLOG(BYN(I)*FUCV(I))-HL(I)
        SMSTAT=SMSTAT+STAT
        SUM=SUM+BYN(I)*(STAT-1.DO)
        BET=BET+(BYN(I)-Z(I))*STAT
  140 CONTINUE
      GSTR=1.DO+SUM
      IF (DABS(BET) .LT. 1.D-10) THEN
IF (BET .GE. 0.D0) BET=1.D-10
IF (BET .LT. 0.D0) BET=-1.D-10
      END IF
      RSM=2*GSTR/BET
* Check for the convergence of the composition vector to the feed
* composition
      SUM=0.D0
      DO 145 I=1,N
        SUM=SUM+DABS(Y(I)-Z(I))
  145 CONTINUE
      IF (SUM .LT. N*2.D-4) THEN
        JCF=1
      ELSE
        JCF=0
      END IF
* Check for the value of r approaching 1 in consecutive iterations
      IF (DABS(RSM-1.DO) .LT. DABS(RSMO-1.DO) .AND. ITNUM .GT.
          1) THEN
        JRS=JRS+1
      ELSE
        JRS=0
      END IF
* Check if SMBYN when less than 1 is decreasing in consecutive
* iterations
      IF (SMBYN .LT. 1.DO .AND. SMBYN .LT. SMBYNO .AND. ITNUM
           .GT. 1) THEN
        JSMBY=JSMBY+1
      ELSE
        JSMBY=0
      END IF
* Check for the objective function increasing in consecutive
* iterations
      IF (OBJ .GT. OBJO .AND. ITNUM .GT. 1) THEN
        JIOB=JIOB+1
```

```
ELSE
         JIOB=0
      END IF
* Check if g*, when negative, is not changing significantly in
* consecutive iterations
      IF (GSTR .LT. 0.DO .AND. DABS(GSTR-GSTRO) .LT. 1.D-6 .AND.
ITNUM .GT. 1) THEN
     +
         JDGST=JDGST+1
      ELSE
         JDGST=0
      END IF
* Check for the objective function increasing after using Wegstein's
* method. If so, the method is used only every JCOB*IWEG iterations
* until the objective function decreases again after applying the
* acceleration method
      IF (TI .NE. 1.DO .AND. OBJ .GT. OBJO) THEN
         JCOB=JCOB+1
         ITCRL=ITNUM+JCOB*IWEG
      ELSE IF (TI .NE. 1.DO) THEN
         JCOB=0
         ITCRL=ITNUM
      END IF
  IF (IFLG(15) .NE. 0) WRITE(6,590) OBJ,SMBYN,GSTR,SMSTAT,BET,RSM
590 FORMAT(/4X,'OBJ. FUNCN.=',F9.6,2X,'SUM Y=',F9.6,2X,'g*=',F9.6
+ /4X,'SUM STAT CR=',F9.6,2X,'beta=',G12.5,2X,'r=',G12.5)
* Test for convergence either to a minimum in Gibbs energy or to
* the trivial solution
      IF (OBJ .LE. EPS .AND. ITNUM .GT. 1) GO TO 160
      IF (JDGST .EQ. 3) GO TO 170
      IF (DABS(RSM-1.DO) .LT. 0.2DO .AND. GSTR .LT. 1.D-3 .OR. JRS
           .EQ. 2 .OR. JIOB .EQ. 2 .OR. JSMBY .EQ. 2 .OR. JFC .EQ. 1)
     +
     4
          GO TO 180
* Assignment of the variables for the next iterative loop
      DO 150 I=1.N
         BYO(I) = BYO(I)
         BY(I)=BYN(I)
         FBY(I)=EXHL(I)/FUCV(I)
  150 CONTINUE
         RSMO=RSM
         OBJO=OBJ
         SMBYNO=SMBYN
* If the maximum number of iterations is reached, the calculation
* ends
       IF (ITNUM .LT. ITMAX) THEN
         GO TO 70
       ELSE
         GO TO 190
      END IF
* Convergence has been reached. Check to see if the stationary point
* corresponds to an unstable system or to a stable one
```

```
IFLG(5)=1
         IF (IFLG(15) .NE. 0) WRITE(6, 600)
  600
        FORMAT(/4X,'CONV. REACHED. UNSTABLE SYSTEM. VAPOUR FOUND')
      ELSE
         IFLG(5)=0
         IF (IFLG(15) .NE. 0) WRITE(6, 610)
  610
       FORMAT(/4X, 'CONV. REACHED. STABLE SYSTEM. NO VAPOUR FOUND')
      END IF
      GO TO 200
* Convergence not reached but a stationary minimum has been found
  170 IFLG(5)=1
  IF (IFLG(15) .NE. 0) WRITE(6, 620)
620 FORMAT(/4X,'CONV. NOT REACHED. UNSTABLE SYSTEM. VAPOUR FOUND')
      GO TO 200
* A trivial solution is found
  180 IFLG(5)=0
  IF (IFLG(15) .NE. 0) WRITE(6, 630)
630 FORMAT(/4X,'TRIVIAL SOLUTION FOUND')
      GO TO 200
* Maximum number of iterations reached
  190 IF (SMBYN .GT. 1.DO) THEN
         IFLG(5)=1
         IF (IFLG(15) .NE. 0) WRITE(6,620)
      ELSE
         IFLG(5)=0
       IF (IFLG(15) .NE. 0) WRITE(6,640)
FORMAT(/4X,'CONV. NOT REACHED. STABLE SYSTEM. VAPOUR NOT FOUND
  640
                 ")
     +
      END IF
  200 RETURN
      END
```

• •

.

```
SUBROUTINE LIQSER (LIQUID PHASE SEARCH)
                                                                          *
           *********
 Purpose:
   Search for two or one incipient liquid phase(s) forming from the
  original system, whether it's vapour-like or liquid-like. The
  method used is based on the modified Michelsen's stability
  analysis
*
  Parameters:
  -Input
  EPS, ITMAX, IWEG, TIMAX, N, P, T, Z, XI, TC, PC, W, PP, APUR, BPUR, KIJ, IFLG as
   defined in program PHASPLIT
            Number of initial liquid composition vectors. M=N+1 when
subroutine INMICH is used, and M=3 for any of the other
            initializing routines
*
   HL
            Vector of the logarithm of the product of the feed comp.
            and the fug. coeff. for the original system as a liquid
            if it is liquid-like or as a vapour if it is vapour-like
  -Output
   ITNŪM
            Number of iterations required for the calculation
   OBJL
            Two element vector with the value of the objective func.
            for liqs 1 and 2. Defined as the sum of the absolute
*
            value of the comp. difference (for each liq) in two
            consecutive iterations
   FUCL1
*
            Vector of fug. coeffs. for the incipient liq. phase 1
*
   FUCL2
                                                                     2
               ..
                   " compositions
                                      ...
                                          ...
                                                  ...
                                                          ...
                                                                ...
   X1
*
                                                                     1
               ...
                                      ..
                                                  ..
                                                          ...
                                                                ..
   X2
                                                                     2
*
  Main variables:
            Value of the gradient defined by Michelsen
*
   BET
            M x N array of values of the composition variables big
   BX
            X (equivalent to mole numbers) in the current iteration
   BXN
            Array of updated values of comp. variables BX
*
   BXO
                       old
   DGRT
            Value of the Gibbs energy of mixing for the comp. vector
*
            when more than two different liq. phases are found
*
   EXHL
*
            Vector of values of the exponential of HL
*
   FBX
            Array (M x N) of the new values of big X obtained by
            direct substitution
   FUCLJC
            Array of values of FUCL for the comp. vectors for which
            a liq. trial phase has been found
           Vector of fug. coeffs. for each liq. being searched
M x N array of fug. coeffs. for the M liq. comp. vectors
Vector (M dim.) with the values of g^*, the stability
   FUCLO
*
*
   FUCLQS
   GSTR
            criterion. If g*>=0 for a given comp. vector, the system
            is stable; otherwise it's unstable
*
   GSTRJC
            Vector of values of GSTR for the comp. vectors for which
            a liq. trial phase has been found
*
            Value of GSTR in the previous iteration
   GSTRO
*
   ITCRL
            Vector of control variables for the application of
            Wegstein's method with a frequency determined by the
            variable IWEG
   JC
            Control variable for the index and number of comp.
            vectors for which a liq. trial phase has been found
   JCF
            Vector (M dim.) of control variables to check for
            convergence of any comp. vector the feed comp.
```

```
JCOB
*
            Vector of control variables for the application of
            Wegstein's method in case the method causes the obj.
*
*
            func. to increase
   JDGST
            Vector (M dim.) of control variables to check if g* for
*
            a given comp. vector, when negative, is not changing
*
            significantly in consecutive iters.
*
   JE
            Control variable for the index and number of comp.
            vectors that have been eliminated from the search
Flag control variables to register the status of the
*
   JFLG
*
            comp. vectors. If JFLG=0, the vector remains in the
*
*
            search; if JFLG=1, a liq. trial phase is found; if
*
            JFLG#0 the vector is eliminated
*
   JGSTR
            Control variable to account for the comp. vectors for
            which g* is greater than zero
*
            Control variable which indicates the comp. vector index
*
   JGRS
*
            with the largest value of /r-1/ when all g* values are
*
            greater than zero
Vector (M dim.) of control variables to check if the
   JIOB
*
*
            objective function for any comp. vector is increasing
*
            in consecutive iterations
*
   JR
            Control variable for the index and number of comp.
            vectors for which different (distinct) liq. phases have
*
*
            been found
            Vector (M dim.) of control variables to detect if the
*
   JRS
*
            value of r for a given comp. vector is approaching one
*
            in consecutive iters.
*
   JSMBX
            M dim. vector of control variables to check if SMBXN,
*
            when less than 1, is decreasing in consec. iterations
            Array of control flags to identify the comp. vectors
*
   LJFLG
            which have converged to the same liquid phase
*
*
   M
            Number of liquid com. vectors for which a liq. phase is
*
            being searched
   OBJJC
            Vector of values of the obj. function of the comp.
*
            vectors for which a liq. trial phase has been found
Vector of values of OBJ in the previous iteration
*
   OBJO
*
            Vector of the gradient ratio r defined by Michelsen;
used to detect the approach to the trivial solution
*
   RSM
*
            Values if RSM in the previous iteration
   RSMO
*
   SMBXN
            M dim. vector of the sum of the values of BXN for each
            liq. searched
*
   SMBXNO
            Vector of values of SMBXN in the previous iteration
            Sum of the stationarity criterion for all comps. of a
   SMSTAT
            given comp. vector. Approaches zero when a stationary
            point is reached
   STAT
            Value of the stationarity criterion for each comp. as
*
            defined by Michelsen
   TI
            M dim. vector of the step size parameter in Wegstein's
            method. If TI=1, the method reduces to direct substn.
            M x N array of coms. vectors of the liq. phases being
*
   X
            searched (mole fraction)
Array of values of X for the comp. vectors for which a
   XJC
*
            liq. phase has been found
                                                                             *
   XJR
            Array of values of X for the comp. vectors for which
            different (distinct) liq. phases have been found
Vector of comps. for each liq. phase being searched
*
   XL
*
            Remaining variables defined as in program PHASPLIT
  Subroutines called:
   Κ
 Called by:
  PHASPLIT
```

```
SUBROUTINE LIQSER(EPS, ITMAX, IWEG, TIMAX, M, N, P, T, Z, XI, HL, TC,
                              PC, W, PP, APUR, BPUR, KIJ, ITNUM, OBJL, FUCL1,
                              FUCL2, X1, X2, IFLG)
      +
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 KEQ(20), KIJ(20,20)
       DIMENSION APUR(20), BPUR(20), BX(21,20), BXN(21,20), BXO(21,20),
                    DGRT(3), EXHL(20), FBX(21,20), FUCD(20), FUCL1(20),
FUCL2(20), FUCLJC(21,20), FUCLQ(20), FUCLQS(21,20),
GSTR(21), GSTRO(21), GSTRJC(21), HL(20), IFLG(0:20),
      +
      t
      +
                    ITCRL(21), JCF(20), JCOB(21), JDGST(21), JFLG(21),
JIOB(21), JRS(21), JSMBX(21), LJFLG(21, 21), OBJ(21),
OBJJC(21), OBJL(2), OBJO(21), PC(20), PP(20), RSM(21),
RSM0(21), SMBXN0(21), SUMJ(21), TC(20), TI(21),
W(20), X(21, 20), X1(20), X2(20), XD(20), XI(21, 20),
VIC(21, 20), VIP(21, 20), VI(20), 7(20)
      ÷
      ÷
      +
      +
      +
                    XJC(21,20),XJR(21,20),XL(20),Z(20)
      4
       ITNUM=0
       JE=0
       JC=0
* The initial values of BX are set, parameter EXP(Hiliq) or
* EXP(Hivap) is obtained and the fugacity coefficients for the
* liquids corresponding to the initial compositions are evaluated
       DO 30 J=1,M
          ITCRL(J)=0
          JCOB(J)=0
          SUMJ(J)=0.D0
          DO 10 I=1,N
BXO(J,I)=XI(J,I)
            XL(I)=XI(J,I)
IF (J .NE. 1) GO TO 10
            EXHL(I)=DEXP(HL(I))
   10
          CONTINUE
          CALL K(XD,XL,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCLQ,FUCD,
      +
                   Z2,ZD,KIJ,KEQ,IFLG)
          DO 20 I=1,N
            FUCLQS(J,I)=FUCLQ(I)
          CONTINUE
   20
   30 CONTINUE
       IF (IFLG(15) .NE. 2) GO TO 40
       WRITE(6,700)
  700 FORMAT(/4X,'INITIAL x COMPOSITIONS')
       WRITE(6,710) ((J,I,XI(J,I), I=1,N), J=1,M)
  710 FORMAT(8X,'x(',I2,I2,')=',F8.5)
* Calculation of the new values of BX using direct substitution
   40 DO 60 J=1,M
          DO 50 I=1,N
            BX(J,I)=EXHL(I)/FUCLQS(J,I)
             SUMJ(J)=SUMJ(J)+BX(J,I)
          CONTINUE
   50
   60 CONTINUE
* The corresponding liquid composition (mole fraction) are obtained
       DO 80 J=1,M
          DO 70 I=1,N
            X(J,I)=BX(J,I)/SUMJ(J)
   70
          CONTINUE
```

```
80 CONTINUE
      IF (IFLG(15) .NE. 2) GO TO 90
      WRITE(6,720)
  720 FORMAT(/4X,'x COMPOSITIONS WITH DIRECT SUBSTITUTION')
      WRITE(6,710) ((J,I,X(J,I), I=1,N), J=1,M)
* Calculation of the fugacity coefficients corresponding to these
* compositions
   90 DO 120 J=1,M
        DO 100 I=1,N
          XL(I)=X(J,I)
  100
        CONTINUE
        CALL K(XD,XL,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCLQ,FUCD,
                Z2,ZD,KIJ,KEQ,IFLG)
     +
        DO 110 I=1,N
          FUCLQS(J,I)=FUCLQ(I)
  110
        CONTINUE
  120 CONTINUE
* The values of F(BX) to be used in Wegstein's method are calculated
      DO 140 J=1,M
        DO 130 I=1,N
          FBX(J,I)=EXHL(I)/FUCLQS(J,I)
        CONTINUE
  130
  140 CONTINUE
* Start of the iterative cycle
  150 ITNUM=ITNUM+1
      DO 160 J=1,M
        SMBXN(J)=0.DO
  160 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,730) ITNUM
  730 FORMAT(/4X, '*** ITERATION NO.=', 12)
* Use of Wegstein's method to obtain updated BX values
      DO 200 J=1,M
        IF (ITNUM .LT. 4 .OR. ITNUM .LT. ITCRL(J)+IWEG
             .OR. IWEG .EQ. O) THEN
     +
           TI(J)=1.D0
        ELSE
          IF (IFLG(15) .NE. 0) WRITE(6,740) J
  740
          FORMAT(4X, 'WEGSTEINS METHOD USED FOR VECTOR', 13)
           TI(J)=0.D0
        END IF
        DO 190 I=1,N
           IF (TI(J) .EQ. 1.DO) GO TO 180
           SI=(FBX(J,I)-BX(J,I))/(BX(J,I)-BXO(J,I))
           TI(J)=1.DO/(1.DO-SI)
* The value of parameter t is limited to avoid overshooting
          IF (TI(J) .GT. TIMAX) TI(J)=TIMAX
IF (TI(J) .LT. -TIMAX) TI(J)=-TIMAX
IF (IFLG(15) .NE. 2) GO TO 170
           IF (TI(J) .EQ. TIMAX) THEN
WRITE(6,750) I,J
          FORMAT(6X,'TI = TIMAX FOR COMP. ',12,' IN VECTOR ',12)
ELSE IF (TI(J) .EQ. -TIMAX) THEN
  750
```

```
WRITE(6,760) I,J
            FORMAT(6X,'TI = -TIMAX FOR COMP. ', I2,' IN VECTOR ', I2)
  760
          END IF
  170
          ITCRL(J)=ITNUM
          BXN(J,I)=(1.DO-TI(J))*BX(J,I)+TI(J)*FBX(J,I)
  180
          IF (BXN(J,I) .LT. 0.DO) BXN(J,I)=1.D-6
          SMBXN(J)=SMBXN(J)+BXN(J,I)
        CONTINUE
  190
  200 CONTINUE
* The new liquid compositions are obtained
      DO 220 J=1,M
        DO 210 I=1,N
          X(J,I)=BXN(J,I)/SMBXN(J)
        CONTINUE
  210
  220 CONTINUE
      IF (IFLG(15) .NE. 2) GO TO 240
      DO 230 J=1,M
        WRITE(6,770) J
        FORMAT(/4X,'J=',12,5X,'I',6X,'x',9X,'BXO',9X,'BX',9X,'FBX',
  770
               9X,'BXN')
        WRITE(6,780) (I,X(J,I),BXO(J,I),BX(J,I),FBX(J,I),BXN(J,I),
                       I=1,N)
  780
        FORMAT(12X, I2, 2X, F8.5, 3X, F8.5, 4X, F8.5, 4X, F8.5, 4X, F8.5)
  230 CONTINUE
* Calculation of the corresponding fugacity coefficients
  240 DO 270 J=1,M
        DO 250 I=1,N
          XL(I)=X(J,I)
  250
        CONTINUE
        CALL K(XD,XL,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCLQ,FUCD,
               Z2,ZD,KIJ,KEQ,IFLG)
     +
        DO 260 I=1,N
          FUCLQS(J,I)=FUCLQ(I)
        CONTINUE
  260
  270 CONTINUE
* Evaluation of the objective functions and parameters r
      JGSTR=0
      DO 290 J=1,M
        SUM=0.D0
        BET=0.DO
        OBJ(J)=0.D0
        SMSTAT=0.D0
        DO 280 I=1,N
          OBJ(J)=OBJ(J)+DABS(BXN(J,I)-BX(J,I))
          STAT=DLOG(BXN(J,I)*FUCLQS(J,I))-HL(I)
          SMSTAT=SMSTAT+STAT
          SUM=SUM+BXN(J,I)*(STAT-1.DO)
          BET=BET+(BXN(J,I)-Z(I))*STAT
  280
        CONTINUE
        GSTR(J)=1.DO+SUM
        IF (DABS(BET) .LE. 1.D-10) THEN
          IF (BET .GE. 0) BET=1.D-10
IF (BET .LT. 0) BET=-1.D-10
        END IF
        RSM(J)=2*GSTR(J)/BET
        IF (GSTR(J) .GT. 0.DO) JGSTR=JGSTR+1
```
```
* Check for the convergence of any composition vector to the
* feed composition
        SUM=0.D0
        DO 285 I=1,N
SUM=SUM+DABS(X(J,I)-Z(I))
        CONTINUE
  285
        IF (SUM .LT. N*2.D-4) THEN
          JCF(J)=1
        ELSE
          JCF(J)=0
        END IF
* Check if the values of r are approaching 1 in consecutive
 iterations
        IF (DABS(RSM(J)-1.DO) .LT. DABS(RSMO(J)-1.DO) .AND. ITNUM
     +
             .GT. 1) THEN
          JRS(J)=JRS(J)+1
        ELSE
          JRS(J)=0
        END IF
* Check if the values of SMBXN when less than 1 are decreasing in
* consecutive iterations
        IF (SMBXN(J).LT. 1.DO .AND. SMBXN(J).LT. SMBXNO(J) .AND.
ITNUM .GT. 1) THEN
     +
          JSMBX(J)=JSMBX(J)+1
        ELSE
          JSMBX(J)=0
        END IF
* Check for the objective function increasing in consecutive
* iterations
        IF (OBJ(J) .GT. OBJO(J) .AND. ITNUM .GT. 1) THEN
          JIOB(J)=JIOB(J)+1
        ELSE
          JIOB(J)=0
        END IF
* Check if the values of g* when negative are not changing
* significantly consecutive iterations
        IF (GSTR(J) .LT. 0.DO .AND. DABS(GSTR(J)-GSTRO(J)) .LT.
1.D-6 .AND. ITNUM .GT. 1) THEN
     +
           JDGST(J)=JDGST(J)+1
        ELSE
          JDGST(J)=0
        END IF
* Check for the objective function increasing after using Wegstein's
* method. If so, the method is used only every JCOB(J)*IWEG iterations
* until the objective function decreases again after using the
* acceleration method
        IF (TI(J) .NE. 1.DO .AND. OBJ(J) .GT. OBJO(J)) THEN
           JCOB(J)=JCOB(J)+1
          ITCRL(J)=ITNUM+JCOB(J)*IWEG
        ELSE IF (TI(J) .NE. 1.DO) THEN
           JCOB(J)=0
```

```
ITCRL(J)=ITNUM
         END IF
         IF (IFLG(15) .NE. 0) WRITE(6,790) J,OBJ(J),SMBXN(J),GSTR(J),
                        SMSTAT, BET, RSM(J)
  790
         FORMAT(/4X,'J=',I2,
               4X, 'OBJ. FUNCN.=', F9.6, 2X, 'SUM X=', F9.6, 2X, 'g*=', F9.6/
      +
               12X, 'SUM STAT CR=', G13.6, 2X, 'beta=', G13.6, 2X, 'r=', G13.6)
      +
  290 CONTINUE
* If all values of g* are greater than zero, the composition vector
* with the largest value of /r-1/ is identified
       IF (JGSTR .EQ. M) THEN
         GRSM=DABS(RSM(1)-1.D0)
         JGRS=1
         DO 300 J=2,M
           IF (DABS(RSM(J)-1.DO) .GT. GRSM) THEN
              GRSM=DABS(RSM(J)-1.D0)
              JGRS=J
           END IF
         CONTINUE
  300
      END IF
* Test for convergence either to a minimum in Gibbs energy or to
* the trivial solution
      DO 320 J=1,M
         IF (JDGST(J) .EQ. 3) THEN
         IF (SMBXN(J), GE, 1, DO) JFLG(J)=1
IF (SMBXN(J), LT, 1, DO) JFLG(J)=2
ELSE IF (OBJ(J), LE, EPS, AND, ITNUM, GT, 1) THEN
IF (SMBXN(J), GE, 1, DO) JFLG(J)=1
IF (SMBXN(J), GE, 1, DO) JFLG(J)=2
         IF (SMBXN(J) .LT. 1.DO) JFLG(J)=2
ELSE IF (DABS(RSM(J)-1.DO) .LT. 0.2DO .AND. GSTR(J) .LT.
                    1.D-3) THEN
     +
            JFLG(J)=3
         ELSE IF (GSTR(J) .GT. O.DO .AND. ITNUM .GE. 4 .OR. JRS(J)
                    .EQ. 2 .OR. JIOB(J) .EQ. 2 .OR. JSMBX(J) .EQ. 2
.OR. JCF(J) .EQ. 1) THEN
      +
      +
            JFLG(J)=4
         ELSE IF (JGSTR .EQ. M .AND. J .NE. JGRS) THEN
           JFLG(J)=5
         ELSE
           JFLG(J)=0
         END IF
* If a liquid trial phase is found, it's composition and fugacity
* coefficients are saved
         IF (JFLG(J) .EQ. 1) THEN
            JC=JC+1
            GSTRJC(JC)=GSTR(J)
           OBJJC(JC)=OBJ(J)
           DO 310 I=1,N
              XJC(JC,I)=X(J,I)
              FUCLJC(JC,I)=FUCLQS(J,I)
  310
            CONTINUE
            IF (IFLG(15) .NE. 0) WRITE(6,800) JC,J
  800
            FORMAT(/4X,12,' LIQUID TRIAL PHASE(S) FOUND'
                    /4X, 'CORRESPONDING COMP. VECTOR #=', I2)
         END IF
  320 CONTINUE
```

```
* Test for the convergence of any composition vector(s) to a common
* vector. The vector with the largest objective function of the two
* common vectors being compared is selected for elimination
      DO 350 L=1,M
        DO 340 J=L+1.M
          SUM=0.D0
          DO 330 I=1,N
            SUM=SUM+DABS(X(L,I)-X(J,I))
  330
          CONTINUE
          IF (SUM .LE. N*2.D-2) THEN
            IF (OBJ(J) .GT. OBJ(L)) THEN
              JFLG(J)=6
            ELSE
              JFLG(L)=6
            END IF
          END IF
        CONTINUE
  340
  350 CONTINUE
* Elimination of the composition vector(s) that have converged,
* that are not likely to converge or that have converged to a
* common vector. Assignment of the variables for the next iteration
      DO 370 J=1,M
        IF (JFLG(J) .NE. 0) THEN
          JE=JE+1
          IF (IFLG(15) .NE. 0) WRITE(6,810) JE,ITNUM,J,JFLG(J)
          FORMAT(/4X,12,' COMP.VECTOR(S) ELIMINATED IN ITNUM=',12
  810
                  /4X, 'COMP. VECTOR ELIMINATED #= ',12,2X,'JFLG= ',12)
          GO TO 370
        END IF
        DO 360 I=1,N
          BXO(J-JE,I)=BX(J,I)
BX(J-JE,I)=BXN(J,I)
          FBX(J-JE,I)=EXHL(I)/FUCLQS(J,I)
  360
        CONTINUE
        RSMO(J-JE)=RSM(J)
        OBJO(J-JE)=OBJ(J)
        GSTRO(J-JE)=GSTR(J)
        SMBXNO(J-JE)=SMBXN(J)
        JRS(J-JE)=JRS(J)
        JIOB(J-JE)=JIOB(J)
        JSMBX(J-JE)=JSMBX(J)
        JCOB(J-JE)=JCOB(J)
        ITCRL(J-JE)=ITCRL(J)
        TI(J-JE)=TI(J)
        JDGST(J-JE)=JDGST(J)
  370 CONTINUE
      M=M-JE
      JE=0
* The calculation terminates if all the composition vectors have
* been eliminated
      IF (M .EQ. 0) GD TO 380
* If the maximum number of iterations is reached, terminate the
* calculation
      IF (ITNUM .LT. ITMAX) GO TO 130
      IF (IFLG(15) .NE. 0) WRITE(6,820)
```

```
820 FORMAT(/4X,'MAX. NO. OF ITERATIONS REACHED')
* Convergence has been obtained
  380 IF (JC .EQ. 0) THEN
* No liquid trial phase(s) found (liquids 1 and 2 do not correspond
* to a solution, but the variables of current composition vectors
 1 and 2 are assigned to them)
        IF (IFLG(15) .NE. 0) WRITE(6,830)
        FORMAT(/4X,'STABLE SYSTEM. NO LIQUID PHASE(S) FOUND')
  830
        DO 390 I=1,N
           X1(I)=X(1,I)
           X2(I)=X(2,I)
           FUCL1(I)=FUCLQS(1,I)
           FUCL2(I)=FUCLQS(2,I)
  390
        CONTINUE
        OBJL(1)=OBJ(1)
        OBJL(2)=OBJ(2)
        IFLG(6)=0
        GO TO 590
      END IF
IF (IFLG(15) .NE. 0) WRITE(6,840) JC
  840 FORMAT(/4X,'UNSTABLE SYSTEM.', I2,' LIQUID PHASE(S) FOUND')
IF (IFLG(15) .EQ. 2) WRITE(6,710)((J,I,XJC(J,I), I=1,N), J=1,JC)
      IF (JC .EQ. 1) THEN
* One liquid trial phase found, designated as liquid 1. Compositions
* and fugacity coefficients are assigned (liquid 2 does not correspond
* to a solution, but the variables of current vector 1 are assigned
* to it)
  400
        DO 410 I=1,N
           X1(I)=XJC(1,I)
           X2(I)=X(1,I)
           FUCL1(I)=FUCLJC(1,I)
           FUCL2(I)=FUCLQS(1,I)
  410
         CONTINUE
         OBJL(1)=OBJJC(1)
        OBJL(2)=OBJ(1)
         IFLG(6)=1
        GO TO 590
      END IF
* At least two liquid phases found. Test for the convergence of any
* composition vector(s) to a common vector
       JR=JC
      DO 440 L=1,JC
         DO 430 J=L+1.JC
           SUM=0.DO
           DO 420 I=1,N
             SUM=SUM+DABS(XJC(L,I)-XJC(J,I))
           CONTINUE
  420
           IF (SUM .LE. N*2.D-2) THEN
             JR=JR-1
             LJFLG(L, J)=1
           ELSE
             LJFLG(L,J)=0
           END IF
         CONTINUE
  430
  440 CONTINUE
```

```
IF (IFLG(15) .NE. 0) WRITE(6,850) JR
 850 FORMAT(/4X,'NUMBER OF DIFFERENT LIQUID PHASES=',12)
IF (JR .EQ. JC) GO TO 520
* Composition vectors have converged to a common vector. All common
* vectors but one are eliminated
      LJR=0
      DO 450 I=1,N
        XJR(1,I)=XJC(1,I)
 450 CONTINUE
     DO 480 L=1,JR
        DO 470 J=L+1, JC
          IF (LJFLG(L,J) .EQ. 1) LJR=LJR+1
          IF (J+LJR .GT. N) GO TO 490
          DO 460 I=1,N
            XJR(J,I)=XJC(J+LJR,I)
 460
          CONTINUE
       CONTINUE
 470
 480 CONTINUE
* Assignment of the composition for the different liquid phase(s)
 490 DO 510 J=1,JR
        DO 500 I=1.N
          XJC(J,I)=XJR(J,I)
 500
        CONTINUE
 510 CONTINUE
* If only one liquid phase is found, the corresponding variables are
* assigned
      IF (JR .EQ. 1) GO TO 400
* Two or more liquid phases found
 520 IFLG(6)=2
      IF (JR .EQ. 2) THEN
* When two liquid trial phases have been found, the compositions and
* fugacity coefficients are assigned
        DO 530 I=1,N
          X1(I)=XJC(1,I)
          X2(I) = XJC(2,I)
          FUCL1(I)=FUCLJC(1,I)
          FUCL2(I)=FUCLJC(2,I)
 530
        CONTINUE
        OBJL(1)=OBJJC(1)
        OBJL(2)=OBJJC(2)
        GO TO 590
      ELSE
* If more than two different liquid phases have been found (with a
* maximum of 3), the two with the least dimensionless Gibbs energy
* of mixing (DEL g/RT) are selected as the trial phases and the
* corresponding variables are assigned
        DO 550 J=1,3
          DGRT(J) = 0.D0
          DO 540 I=1,N
```

```
DGRT(J)=DGRT(J)+XJC(J,I)*DLOG(XJC(J,I)*FUCLJC(J,I))
         CONTINUE
540
550
       CONTINUE
       IF (DGRT(3) .GT. DGRT(1) .AND. DGRT(3) .GT. DGRT(2)) THEN
         DO 560 I=1,N
           X1(I)=XJC(1,I)
           X2(1) = XJC(2,1)
           FUCL1(I)=FUCLJC(1,I)
           FUCL2(I)=FUCLJC(2,I)
560
         CONTINUE
         OBJL(1)=OBJJC(1)
OBJL(2)=OBJJC(2)
       ELSE IF (DGRT(2) .GT. DGRT(1) .AND. DGRT(2) .GT. DGRT(3)) THEN
         DO 570 I=1.N
           X1(I)=XJC(1,I)
X2(I)=XJC(3,I)
FUCL1(I)=FUCLJC(1,I)
           FUCL2(I)=FUCLJC(3,I)
570
         CONTINUE
         OBJL(1)=OBJJC(1)
         OBJL(2)=OBJJC(3)
      ELSE
         DO 580 I=1,N
           X1(I)=XJC(2,I)
           X2(I)=XJC(3,I)
           FUCL1(I)=FUCLJC(2,I)
FUCL2(I)=FUCLJC(3,I)
580
         CONTINUE
         OBJL(1)=OBJJC(2)
         OBJL(2)=OBJJC(3)
      END IF
IF (IFLG(15) .NE. 0) WRITE(6,860)
      FORMAT(/4X, 'VECTORS WITH LEAST GIBBS ENERGY OF MIXING')
860
    END IF
590 RETURN
    END
```

```
************
                  SUBROUTINE PHASER (PHASE SEARCH)
***********
* Purpose:
 Calculate the value of Hp to be used in the optional additional
* search in order to look for a possibly missed phase by the
* modified Michelsen method (initial search)
* Parameters:
*
 -Input:
  Z,P,T,N,IFLG as defined in program PHASPLIT
*
          Composition vector of the phase found in the initial
  XΡ
          phase search
                         ....
                             " a second liquid if found;
  XQ
           Composition
          otherwise a dummy variable
*
*
  DELGS
          Gibbs energy of mixing for the system as a stable phase
           (either of the liquid-like or vapour-like system)
          Fugacity coeff. vector of the phase found with PHASPLIT
""" second liquid if found;
  FUCP
*
*
  FUCQ
          otherwise a dummy variable
*
 -Output:
          Vector of the pseudo-values of HL when the system is
*
  HP
          liquid-like and of HV when it is vapour-like (see
           subroutines LIQSER and VAPSER)
*
 Main variables:
  DELGP
          Gibbs energy of mixing of the phase found in the initial
          phase search
  DELGQ
          Gibbs energy of mixing of a second liquid if found
          Difference in Gibbs energy of mixing between the phase
found and the system as a single phase or between the
  DIFDG
*
          two liquids found
  PDGDZ
          Pseudo-slope of the Gibbs energy of mixing a the feed
*
          composition
  PMU
           Vector of values of the pseudo-chemical potential of
          mixing
***************
      SUBROUTINE PHASER(Z, XP, XQ, P, T, N, DELGS, FUCP, FUCQ, IFLG, HP)
      IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION FUCP(20), FUCQ(20), HP(20), IFLG(0:20), PDGDZ(20),
PMU(20), XD(20), XP(20), XQ(20), Z(20)
      PLOG=DLOG(P)
      RT=1.987D0*T
      IF (IFLG(15) .NE. 0) WRITE (6,500)
  500 FORMAT(/4X, 'SUBROUTINE PHASER RESULTS')
* Calculation of Hp for a binary mixture
      IF (N .GT. 2) GO TO 90
* Calculation of the Gibbs energy of mixing for the phase(s) found
* with the modified Michelsen method
     SUM=0.D0
      DO 10 I=1,N
        SUM=SUM+XP(I)*(DLOG(XP(I)*FUCP(I))+PLOG)
```

```
10 CONTINUE
      DELGP=RT*SUM
      IF (IFLG(5) .EQ. O .AND. IFLG(6) .EQ. 2) THEN
        SUM=0.DO
        DO 20 I=1,N
           SUM=SUM+XQ(I)*(DLOG(XQ(I)*FUCQ(I))+PLOG)
   20
        CONTINUE
        DELGQ=RT*SUM
        DIFDG=DELGQ-DELGP
        GO TO 40
      END IF
                                                . -21
      DIFDG=DELGS-DELGP
* The pseudo-slope of the Gibbs energy of mixing at the feed,
* the pseudo-chemical potential of mixing and the value of Hp
* are evaluated
      DO 30 I=1,N
        PDGDZ(I) = DIFDG/(Z(I) - XP(I))
   30 CONTINUE
      IF (.NOT. (IFLG(5) .EQ. 0 .AND. IFLG(6) .EQ. 2)) GO TO 60
   40 DO 50 I=1.N
        PDGDZ(I)=DIFDG/(XQ(I)-XP(I))
   50 CONTINUE
   60 DO 80 I=1,N
        SUMPD=0.D0
        DO 70 J=1,N
           IF (J .EQ. I) GO TO 70
           SUMPD=SUMPD+Z(J)*PDGDZ(J)
   70
        CONTINUE
        PMU(I)=DELGS-SUMPD
        HP(I)=PMU(I)/RT-PLOG
   80 CONTINUE
      IF (IFLG(15) .NE. 0) THEN
WRITE(6,510) (I,PDGDZ(I),I,PMU(I), I=1,N)
        FORMAT(4X, 'PDGDZ(', I2,')=',G11.4,2X, 'PMU(',I2,')=',G11.4)
WRITE(6,520) (I,HP(I), I=1,N)
FORMAT (6X,'hp(',I2,')=',G11.4)
  510
  520
      END IF
      RETURN
* Calculation of Hp for a multicomponent mixture (N>2)
   90 DO 100 I=1,N
        HP(I)=DELGS/RT-PLOG
  100 CONTINUE
      IF (IFLG(15) .NE. 0) THEN
        WRITE(6,520) (I,HP(I), I=1,N)
      END IF
      RETURN
      END
```

```
SUBROUTINE TESTA (TEST FOR THE STABILITY OF 3 PHASES)
           ********
* Purpose:
  Using the multiphase flash reduction tests, determine the number
*
 and type of phases corresponding to the most stable system from
 the results of the phase search when a potential 3 phase solution
 is found
*
 Parameters:
 -Input
*
  Z, Y, X1, X2, IP, TC, PC, W, PP, P, T, N, APUR, BPUR, KIJ, IFLG as defined in
  program PHASPLIT
  K1
           Vector of VL equilibrium constants corresponding to the
           comps. obtained from the phase search
*
  K2
           Vector of LL equilibrium constants corresponding to the
           comps. obtained from the phase search
*
  -Output
  YN,X1N,X2N,FUCVN,FUCL1N,FUCL2N,ALPHA,BETA as defined in program
*
  PHASPLIT
*
 Main variables:
*
  BL1
           Variable to limit the value of BETA
*
  BL2
  D
*
           Determinant of the mass balance and equilibrium system of
           equations solved by Newton-Raphson's method
*
  E1
           Partial derivative of the VL mass balance and equil. eqn.
*
           with respect to ALPHA
           Same as £1 but derivative with respect to BETA
*
  E2
  E3
           Partial derivative of the LL mass balance and equil. eqn.
*
           with respect to ALPHA
*
  E4
           Same as E3 but derivative with respect to BETA
           Value of the VL mass balance and equil. eqn.
*
  G1
  G2
*
   JL1
           Control variable to count the number of successive iters.
*
           in which BETA<0 or BETA>1
   JL2
*
           Control variable to count the number of successive iters.
*
           in which 1-ALPHA-BETA<0 or 1-ALPHA-BETA>1
           Control variable to count the number of successive iters.
   JV
           in which ALPHA<0 or ALPHA>1
  OMKL
*
          Vector of values of one minus the LL equil. constants
   OMKV
                               ...
                                    ...
                                         .....
                                            VI.
           Remaining variables as defined in program PHASPLIT
*
 Subroutines called:
*
  LIMAB
  K
 Called by:
*
  PHASPLIT
                  ******
     SUBROUTINE TESTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
                       K1, K2, IFLG, YN, X1N, X2N, FUCVN, FUCL1N, FUCL2N,
     +
     +
                       ALPHA, BETA)
```

```
IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 K1(20), K2(20), KEQ(20), KIJ(20,20), KV(20), KL(20)
DIMENSION APUR(20), BPUR(20), FUCD(20), FUCL1N(20), FUCL2N(20),
FUCVN(20), IFLG(0:20), OMKL(20), OMKV(20), PC(20),
      +
                    PP(20), TC(20), W(20), X1(20), X2(20), X1N(20), X2N(20),
      +
      +
                    Y(20), YN(20), Z(20)
       IFLG(13)=1
       ITMAX=5
       ITNUM=0
        JV=0
        JL1=0
        JL2=0
* Calculation of initial estimates of phase indicators V/F and L1/F
       BETA=0.5D0
       ALPHA=(Z(IP)-BETA*X1(IP)-X2(IP)*(1.DO-BETA))/
(Y(IP)-BETA*X1(IP)-X2(IP)*(1.DO-BETA))
      +
       BETA=BETA*(1.DO-ALPHA)
       IF (ALPHA .LT. 0.05D0) ALPHA=0.05D0
       IF (ALPHA .GT. 0.95D0) ALPHA=0.95D0
IF (BETA .LT. 0.05D0) BETA=0.05D0
IF (BETA .GT. 0.95D0) BETA=0.95D0
       IF (IFLG(15) .NE. 0) WRITE(6,500) ALPHA, BETA
  500 FORMAT(4X, 'V/F INIT=', F9.4, 2X, 'L1/F INIT=', F9.4)
       DO 10 I=1,N
          KV(I) = KI(I)
          KL(I) = K2(I)
          OMKV(I)=1.DO-KV(I)
          OMKL(I)=1.DO-KL(I)
          YN(I)=Y(I)
          X1N(I)=X1(I)
          X2N(I) = X2(I)
    10 CONTINUE
* Start of the iterative cycle
    20 ITNUM=ITNUM+1
        IF (IFLG(15) .NE. 0) WRITE(6,510) ITNUM
  510 FORMAT(4X, '*** ITNUM=',I3)
IF (ITNUM .GT. ITMAX) THEN
          IFLG(11)=4
          GO TO 200
        END IF
* If the KL values are near unity, the liquid phase with greater
* tendency to disappear is eliminated
        SUM=0.D0
       DO 30 I=1,N
          SUM=SUM+DABS(OMKL(I))
    30 CONTINUE
        IF (SUM .LT. 0.001D0) THEN
          SMFL1=0.D0
          SMFL2=0.D0
          DO 40 I=1,N
             SMFL1=SMFL1+X1N(I)*FUCL1N(I)
             SMFL2=SMFL2+X2N(I)*FUCL2N(I)
    40
          CONTINUE
          IF (SMFL1 .GT. SMFL2) THEN
            BETA=0.DO
          ELSE
             BETA=1.DO-ALPHA
```

```
END IF
        IF (IFLG(15) .NE. 0) WRITE(6,520)
        FORMAT(/4X,'STABLE SYSTEM AS V-L (KL=1)'/)
 520
        IFLG(11)=2
        GO TO 160
      END IF
* Calculation of V/F and/or L1/F with VL and LL K's fixed using
* Newton-Raphson's method
      DO 60 J=1,15
G1=0.D0
        G2=0.D0
        E1=0.D0
        E2=0.D0
        E3=0.D0
        E4=0.D0
        DO 50 I=1,N
          E5=(BETA*KL(I)+1.DO-BETA-ALPHA+ALPHA*KV(I))**2
          IF (E5 .LT. 1.D-30) E5=1.D-30
          G1=G1+Z(I) *OMKV(I)/DSQRT(E5)
          E1=E1+Z(I)*OMKV(I)**2/E5
          E2=E2+Z(I)*OMKV(I)*OMKL(I)/E5
          G2=G2+Z(I)*OMKL(I)/DSQRT(E5)
          E3=E2
          E4=E4+Z(I)*OMKL(I)**2/E5
        CONTINUE
   50
        IF (DABS(G1)+DABS(G2) .LT. 1.D-7) GO TO 70
        D=E1*E4-E2*E3
        IF (DABS(D) .LT. 1.D-15) GO TO 70
A0=ALPHA+(G2*E2-G1*E4)/D
        BO=BETA+(G1*E3-G2*E1)/D
        ALPHA=AO
        BETA=BO
        IF (DABS(D) .LT. 1.D-9) GO TO 70
   60 CONTINUE
   70 IF (IFLG(15) .NE. 0) WRITE(6,530) ALPHA, BETA
  530 FORMAT(4X, 'V/F=', F11.4, 2X, 'L1/F=', F11.4)
* If more than one phase indicator is outside the interval [0,1],
*
 the phase with greater tendency to disappear is found
      IF (ALPHA .LT. O.DO .AND. (BETA .LT. O.DO .OR. BETA .GT.
          1.DO)) THEN
     +
        JL1=JL1-1
        JV=JV+1
      END IF
      ONMAB=1.DO-ALPHA-BETA
      IF (ALPHA .LT. O.DO .AND. (DNMAB .LT. O.DO .OR. ONMAB .GT.
+ 1.DO)) THEN
        JL2=JL2-1
        JV=JV+1
      END IF
      IF (ALPHA .LE. 1.DO .OR. BETA .GE. O.DO .OR. ONMAB .GE.
• O.DO) GO TO 90
      SMFL1=0.D0
      SMFL2=0.D0
      DO 80 I=1,N
        SMFL1=SMFL1+X1N(I)*FUCL1N(I)
        SMFL2=SMFL2+X2N(I)*FUCL2N(I)
   80 CONTINUE
```

```
IF (SMFL1 .GT. SMFL2) THEN
         JL2=JL2-1
       ELSE
         JL1=JL1-1
       END IF
* Limitation of the calculated values of V/F and/or L1/F when
* outside the interval [0,1] to avoid calculation of negative
* mole fractions
   90 IF (ALPHA .GT. 1.DO .OR. ALPHA .LT. O.DO) CALL LIMAB(N,
      + KV.ALPHA)
       BL2=0.D0
       DO 100 I=1,N
IF (OMKL(I) .EQ. 0.DO) GO TO 105
         BL1=(ALPHA+OMKV(I)-1.DO)/(-OMKL(I))
         IF (BETA .GT. O.DO .AND. BL1 .GT. O.DO .AND. BETA .GT.
         BL1 .AND. (BL1 .LT. BL2 .OR. BL2 .EQ. 0.DO)) BL2=BL1
IF (BETA .LT. 0.DO .AND. BL1 .LT. 0.DO .AND. BETA .LT.
BL1 .AND. (BL1 .GT. BL2 .OR. BL2 .EQ. 0.DO)) BL2=BL1
     +
  100 CONTINUE
  105 IF (BL2 .EQ. 0.D0) GO TO 110
IF (BETA .GT. 0.D0 .AND. BETA .GT. BL2 .AND. BL2 .GE. 1.D0)
         BETA=1.D0+(BL2-1.D0)/2.D0
       IF (BETA .GT. O.DO .AND. BETA .GT. BL2 .AND. BL2 .LT. 1.DO)
        BETA=BL2-0.005D0
       IF (BETA .LT. O.DO .AND. BETA .LT. BL2) BETA=BL2/2.DO
  110 IF (ALPHA .LE. O.DO .OR. ALPHA .GE. 1.DO) THEN
         JV=JV+1
       ELSE
         JV=0
       END IF
       IF (BETA .LE. O.DO .OR. BETA .GE. 1.DO) THEN
         JL1=JL1+1
       ELSE
         JL1=0
       END IF
       ONMAB=1.DO-ALPHA-BETA
       IF (ONMAB .LE. O.DO .OR. ONMAB .GE. 1.DO) THEN
         JL2=JL2+1
       ELSE
         JL2=0
       END IF
       IF (IFLG(15) .NE. 0) WRITE(6,540) ALPHA, BETA
  540 FORMAT(4X, 'V/F LIM=', F11.4, 2X, 'L1/F LIM=', F11.4)
* If the value of V/F, L1/F or L2/F has been outside the interval
* [0,1] for three consecutive iterations, the corresponding phase
* is eliminated
  IF (IFLG(15) .NE. 0) WRITE(6,550) JV,JL1,JL2
550 FORMAT(4X,'IL1L1=',I3,2X,'IVL2=',I3,2X,'IVL1=',I3)
IF (.NOT. (ALPHA .LE. 0.D0 .AND. JV .GE. 3)) GO TO 120
* Elimination of the vapour phase
       ALPHA=0
       IF (IFLG(15) .NE. 0) WRITE(6,560)
  560 FORMAT(/4X, 'STABLE SYSTEM AS L-L'/)
       IFLG(11)=3
       GO TO 160
```

```
* Elimination of liquid 1
  120 IF (JL1 .GE. 3 .AND. BETA .LE. O.DO) THEN
        BETA=0.DO
        IF (IFLG(15) .NE. 0) WRITE(6,570)
        FORMAT(/4X,'STABLE SYSTEM AS V-L2'/)
  570
        IFLG(11)=2
        GO TO 160
      END IF
* Elimination of liquid 2
      IF (JL2 .GE. 3 .AND. 1.DO-ALPHA-BETA .LE. 0.DO) THEN
        BETA=1.DO-ALPHA
        IF (IFLG(15) .NE. 0) WRITE(6,580)
        FORMAT(/4X, 'STABLE SYSTEM AS V-L1'/)
  580
        IFLG(11)=2
        GO TO 160
      END IF
* Calculation of the new compositions of the 3 phases
      S0=0.D0
      S1=0.D0
      S2=0.D0
      DO 130 I=1,N
        X2N(I)=Z(I)/(BETA*KL(I)+1.DO-ALPHA-BETA+ALPHA*KV(I))
        IF (X2N(I) .LT. 0.D0) X2N(I)=0.D0
        S2=S2+X2N(I)
        X1N(I) = X2N(I) * KL(I)
        S1=S1+X1N(I)
        YN(I) = X2N(I) * KV(I)
        SO=SO+YN(I)
  130 CONTINUE
      DO 140 I=1,N
        X2N(I)=X2N(I)/S2
        X1N(I)=X1N(I)/S1
        YN(I)=YN(I)/SO
  140 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,590) (I,YN(I);I,X1N(I),I,X2N(I),
     +
          I=1,N)
  590 FORMAT(6X,'YN(',12,')=',F9.6,2X,'X1N(',I2,')=',F9.6,2X,'X2N(',
+ I2,')=',F9.6)
* Calculation of the new values of K's
      IFLG(1)=0
      IFLG(2)=0
IFLG(3)=0
      IFLG(16)=0
      CALL K(XD, YN, TC, PC, W, PP, P, T, N, 1, 3, APUR, BPUR, FUCVN, FUCD,
             Z1,Z0,KIJ,KEQ,IFLG)
     +
      CALL K(XD,X1N,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCL1N,FUCD,
             Z2,Z0,KIJ,KEQ,IFLG)
     +
      IF (IFLG(2) .EQ. 1) THEN
        IFLG(2)=0
        IFLG(16)=1
      END IF
      CALL K(XD, X2N, TC, PC, W, PP, P, T, N, 2, 3, APUR, BPUR, FUCL2N, FUCD,
     +
             Z3,Z0,KIJ,KEQ,IFLG)
      IF (IFLG(2) .EQ. 1) THEN
        IFLG(2)=0
        IFLG(3)=1
```

```
END IF
       IF (IFLG(16) .EQ. 1) IFLG(2)=1
       DO 150 I=1,N
          KV(I)=FUCL2N(I)/FUCVN(I)
          KL(I)=FUCL2N(I)/FUCL1N(I)
          OMKV(I)=1.DO-KV(I)
OMKL(I)=1.DO-KL(I)
  150 CONTINUE
  IF (IFLG(15) .NE. 0) WRITE(6,600) (I,KV(I),I,KL(I), I=1,N)
600 FORMAT(6X,'KV(',I2,')=',F10.4,2X,'KL(',I2,')=',F10.4)
IF (IFLG(1) .EQ. 1 .AND. IFLG(15) .NE. 0) WRITE(6,610)
610 FORMAT(4X,'EXTRAPOLATED ZVAP')
       IF (IFLG(2) .EQ. 1 .AND. IFLG(15) .NE. 0) WRITE(6,620)
  620 FORMAT(4X, 'EXTRAPOLATED ZLIQ1')
  IF (IFLG(3) .EQ. 1 .AND. IFLG(15) .NE. 0) WRITE(6,630)
630 FORMAT(4X,'EXTRAPOLATED ZLIQ2')
* End of iterative cycle
       GO TO 20
* Results of calculation
  160 IF (ALPHA .EQ. O.DO) THEN
          DO 170 I=1,N
            YN(I)=0.DO
            X1N(I)=X1(I)
            X2N(I)=X2(I)
  170
          CONTINUE
       GO TO 200
ELSE IF (BETA .EQ. 0.DO) THEN
          DO 180 I=1,N
            YN(I)=Y(I)
            X1N(I)=0.D0
X2N(I)=X2(I)
  180
          CONTINUE
          IFLG(14)=0
       GO TO 200
ELSE IF (BETA .EQ. 1.DO-ALPHA) THEN
          DO 190 I=1,N
            YN(I)=Y(I)
            X1N(I)=0.D0
            X2N(I)=X1(I)
  190
          CONTINUE
          IFLG(14)=1
       END IF
  200 RETURN
       END
* SUBROUTINE FOR LIMITING V/F OR L1/F AS IN A 2-PHASE CALCULATION
       SUBROUTINE LIMAB(N,RK,C)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION RK(20)
       RKM=RK(1)
       DO 20 I=2,N
IF (C .LT. 0.DO) GO TO 10
          IF (RK(I) .LT. RKM) RKM=RK(I)
          GO TO 20
    10 IF (RK(I) .GT. RKM) RKM=RK(I)
    20 CONTINUE
       CO=1.DO/(1.DO-RKM)
```

IF (C .LT. 0.D0 .AND. C0 .GT. 0.D0) THEN C=-C0 GO TO 30 ELSE IF (C .GT. 0.D0 .AND. C0 .LT. 0.D0) THEN C=1.D0-C0 GO TO 30 END IF IF (C .LT. 0.D0 .AND. C .LT. C0) C=C0/2.D0 IF (C .GT. 1.D0 .AND. C .GT. C0) C=(1.D0+C0)/2.D0 30 RETURN END

•

```
*******
       SUBROUTINE NELSTA (NELSON'S STABILITY TEST FOR 3 PHASES)
k sk
*
 Purpose:
 Using the modified Nelson's phase reduction tests, determine the
* number and type of phases corresponding to the most stable system
* from the results of the phase search when a potential 3 phase
* solution is found
* Parameters:
*
 -Input
* Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,IFLG as defined in
*
  program PHASPLIT
*
  K1, K2 as defined is subroutine TESTA
*
*
 -Output
  YN, X1N, X2N, FUCVN, FUCL1N, FUCL2N, ALPHA, BETA as defined in program
*
  PHÁSPLIT
*
* Main variables:
*
  BL1,BL2,D,E1,E2,E3,E4,G1,G2,OMKL,OMKV as defined in subroutine
*
  TESTA
*
  DKV
          Vector of the values of the difference between the LL
*
           and VL equil. constants
*
  DGAMLL
          Partial derivative of GAMLL with respect to BETA
                       ...
*
  DGMVL1
             ...
                             " GAMVL1 "
                                                   11
                                                      ....
                             н
                                              ..
                                                   " ALPHA
             ..
                       ...
                                GAMVL2 "
*
  DGMVL2
          Mass balance and equil. eqn. to test for the system LL as
  GAMLL
*
*
          the stable solution
                                       ...
                                           ..
                                                ..
  GAMVL1 Mass balance and equil.
                                                              VL1
*
*
           as the stable solution
  GAMVL2 Mass balance and equil.
                                   ..
                                       ...
                                           ..
                                                14
                                                    41
                                                         ...
                                                              VL2
*
*
           as the stable solution
*
  IL
           Control variable for a liq phase as the stable solution
                            " the system VL1 as the stable soln.
" " " VL2 " " " "
*
  IVL1
             ....
                      ....
                      ...
             11
*
  IVL2
                                          LL "
             **
                      ...
                            ...
                                ....
                                     ...
                                                  ...
                                                       ...
                                                              ..
  ILL
*
                                          VLL "
             ...
                      ...
                            ...
                                **
                                     11
                                                  ....
                                                        ...
                                                              ••
*
  IVLL
          Limit for the number of consecutive iterations to
*
  ILIM
*
           determine the stable solution
*
  S1 - S6 Value of the sum of the various feed comp. and equil.
           constants defined by Nelson as criteria for the
*
           existence of one, two or three phases
           Remaining variables as defined in program PHASPLIT and
           subroutine TESTA
*
 Subroutines called:
*
  LIMAB
*
  K
 Called by:
*
  PHASPLIT
*
    *********
     SUBROUTINE NELSTA(Z,Y,X1,X2,IP,TC,PC,W,PP,P,T,N,APUR,BPUR,KIJ,
                       K1,K2,IFLG,YN,X1N,X2N,FUCVN,FUCL1N,FUCL2N,
```

```
ALPHA, BETA)
      +
       IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 K1(20), K2(20), KEQ(20), KIJ(20, 20), KV(20), KL(20)
      DIMENSION APUR(20), BPUR(20), DKV(20), FUCD(20), FUCL1N(20)
                   FUCL2N(20),FUCVN(20),IFLG(0:20),OMKL(20),OMKV(20),
                   PC(20), PP(20), TC(20), W(20), X1(20), X2(20), X1N(20), X2N(20), Y(20), YN(20), Z(20)
      +
      +
       ILIM=3
       IV=0
       IL=0
       IVL1=0
       IVL2=0
       ILL=0
       IVLL=0
* Calculation of initial estimates of phase indicators V/F and L1/F
       BETA=0.5D0
       ALPHA=(Z(IP)-BETA*X1(IP)-X2(IP)*(1.DO-BETA))/
              (Y(IP)-BETA*X1(IP)-X2(IP)*(1.DO-BETA))
      +
       BETA=BETA*(1.DO-ALPHA)
       IF (ALPHA .LT. 0.05D0) ALPHA=0.05D0
IF (ALPHA .GT. 0.95D0) ALPHA=0.95D0
       IF (BETA .LT. 0.05D0) BETA=0.05D0
       IF (BETA .GT. 0.95D0) BETA=0.95D0
  IF (IFLG(15) .NE. 0) WRITE(6,500) ALPHA,BETA
500 FORMAT(4X,'V/F INIT=',F9.4,2X,'L1/F INIT=',F9.4)
       DO 10 I=1,N
         KV(I) = KI(I)
         KL(I)=K2(I)
         OMKV(I)=1.DO-KV(I)
         OMKL(I)=1.DO-KL(I)
   10 CONTINUE
  IF (IFLG(15) .NE. 0) WRITE(6,510) (I,KV(I),I,KL(I), I=1,N)
510 FORMAT(6X,'KV(',I2,')=',F10.4,2X,'KL(',I2,')=',F10.4)
* Start of the iterative cycle
* Calculation of V/F and L1/F with VL and LL K's fixed using Newton-
* Raphson's method
   20 DO 40 J=1,15
         G1=0.D0
         G2=0.D0
         E1=0.D0
         E2=0.D0
         E3=0.D0
         E4=0.D0
         DO 30 I=1,N
            E5=(BETA*KL(I)+1.DO-BETA-ALPHA+ALPHA*KV(I))**2
            IF (E5 .LT. 1.D-30) E5=1.D-30
            G1=G1+Z(I) *OMKV(I)/DSQRT(E5)
            E1=E1+Z(I)*OMKV(I)**2/E5
E2=E2+Z(I)*OMKV(I)*OMKL(I)/E5
            G2=G2+Z(I)*OMKL(I)/DSQRT(E5)
            E3=E2
            E4=E4+Z(I)*OMKL(I)**2/E5
   30
         CONTINUE
         IF (DABS(G1)+DABS(G2) .LT. 1.D-7) GO TO 50
         D=E1*E4-E2*E3
         IF (DABS(D) .LT. 1.D-15) GO TO 50
         AO=ALPHA+(G2*E2-G1*E4)/D
         BO=BETA+(G1*E3-G2*E1)/D
          ALPHA=AO
```

```
BETA=BO
        IF (DABS(D) .LT. 1.D-9) GO TO 50
   40 CONTINUE
   50 IF (IFLG(15) .NE. 0) WRITE(6,520) ALPHA, BETA
  520 FORMAT(4X, 'V/F=', G11.4, 2X, 'L1/F=', G11.4)
* Limitation of the calculated values of V/F and/or L1/F when
* outside the interval [0,1] to avoid calculation of negative
* mole fractions
      IF (ALPHA .GT. 1.DO .OR. ALPHA .LT. O.DO) CALL LIMAB(N,
     + KV, ALPHA)
      BL2=0.D0
      DO 60 I=1,N
        IF (KL(I) .EQ. 1.DO) GO TO 65
        BL1=(ALPHA*(1.DO-KV(I))-1.DO)/(KL(I)-1.DO)
        IF (BETA .GT. O.DO .AND. BL1 .GT. O.DO .AND. BETA .GT.
            BL1 .AND. (BL1 .LT. BL2 .OR. BL2 .EQ. 0.D0)) BL2=BL1
     +
        IF (BETA .LT. O.DO .AND. BL1 .LT. O.DO .AND. BETA .LT.
            BL1 .AND. (BL1 .GT. BL2 .OR. BL2 .EQ. 0.D0)) BL2=BL1
     +
   60 CONTINUE
   65 IF (BL2 .EQ. 0.D0) GO TO 70
      IF (BETA .GT. O.DO .AND. BETA .GT. BL2 .AND. BL2 .GE. 1.DO)
       BETA=1.DO+(BL2-1.DO)/2.DO
      IF (BETA .GT. O.DO .AND. BETA .GT. BL2 .AND. BL2 .LT. 1.DO)
       BETA=BL2-0.005D0
     +
  IF (BETA .LT. O.DO .AND. BETA .LT. BL2) BETA=BL2/2.DO
70 IF (IFLG(15) .NE. 0) WRITE(6,530) ALPHA,BETA
  530 FORMAT(4X, 'V/F LIM=',G11.4,2X, 'L1/F LIM=',G11.4)
* Calculation of the new compositions in the 3 phases
      S0=0.D0
      S1=0.D0
      S2=0.D0
      DO 80 I=1,N
        X2N(I)=\dot{Z}(I)/(BETA*KL(I)+1.DO-ALPHA-BETA+ALPHA*KV(I))
        IF (X2N(I) .LT. 0.DO) X2N(I)=0.DO
S2=S2+X2N(I)
        X1N(I)=X2N(I)*KL(I)
        S1=S1+X1N(I)
        YN(I)=X2N(I)*KV(I)
        SO=SO+YN(I)
   80 CONTINUE
      DO 90 I=1,N
        X2N(I) = X2N(I)/S2
        X1N(I)=X1N(I)/S1
        YN(I)=YN(I)/SO
   90 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,540) (I,YN(I),I,X1N(I),I,X2N(I),
          I=1,N)
  540 FORMAT(6X,'YN(',I2,')=',F9.6,2X,'X1N(',I2,')=',F9.6,2X,'X2N(',
+ I2,')=',F9.6)
* Calculation of KV and KL with the new composition values
      IFLG(1)=0
      IFLG(2)=0
      IFLG(3)=0
      IFLG(16)=0
      CALL K(XD, YN, TC, PC, W, PP, P, T, N, 1, 3, APUR, BPUR, FUCVN, FUCD, ZV, ZD,
              KIJ, KEQ, IFLG)
      CALL K(XD,X1N,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUCL1N,FUCD,ZL1,ZD,
```

```
KIJ, KEQ, IFLG)
       IF (IFLG(2) .EQ. 1) THEN
         IFLG(2)=0
         IFLG(16)=1
       END IF
       CALL K(XD, X2N, TC, PC, W, PP, P, T, N, 2, 3, APUR, BPUR, FUCL2N, FUCD, ZL2, ZD,
                KIJ,KEQ,IFLG)
      +
       IF (IFLG(2) .EQ. 1) THEN
          IFLG(2)=0
          IFLG(3)=1
       END IF
IF (IFLG(16) .EQ. 1) IFLG(2)=1
DO 100 I=1,N
         KV(I)=FUCL2N(I)/FUCVN(I)
         KL(I)=FUCL2N(I)/FUCL1N(I)
         OMKV(I)=1.DO-KV(I)
OMKL(I)=1.DO-KL(I)
         DKV(I)=KL(I)-KV(I)
 100 CONTINUE
       IF (IFLG(15) .NE. 0) WRITE(6,510) (I,KV(I),I,KL(I), I=1,N)
  IF (IFLG(1) .EQ. 1 .AND. IFLG(15) .NÉ. 0) WRÍTE(6,550)
550 FORMAT(4X,'EXTRAPOLATED ZVAP')
       IF (IFLG(2) .EQ. 1 .AND. IFLG(15) .NE. 0) WRITE(6,560)
  560 FORMAT(4X, 'EXTRAPOLATED ZLIQ1')
       IF (IFLG(3) .EQ. 1 .AND. IFLG(15) .NE. 0) WRITE(6,570)
  570 FORMAT(4X, 'EXTRAPOLATED ZLIQ2')
* Test for the existence of a single phase
       S1=0.D0
       S2=0.D0
       S3=0.D0
       S4=0.D0
       S5=0.D0
       S6=0.D0
       DO 110 I=1,N
         S1=S1+Z(I)/KV(I)
          S2=S2+Z(I)*KL(I)/KV(I)
          S3=S3+Z(I)*KV(I)/KL(I)
          S4=S4+Z(I)/KL(I)
         S5=S5+Z(I)*KV(I)
          S6=S6+Z(I)*KL(I)
  110 CONTINUE
  IF (IFLG(15) .NE. 0) WRITE(6,580)S1,S2,S3,S4,S5,S6
580 FORMAT(6X,'S1=',F9.5,2X,'S2=',F9.5,2X,'S3=',F9.5/
+ 6X,'S4=',F9.5,2X,'S5=',F9.5,2X,'S6=',F9.5)
IF (S1 .LT. 1.D0 .AND. S2 .LT. 1.D0) THEN
         IV=IV+1
          IF (IFLG(15) .NE. 0) WRITE(6,590)
         FORMAT(/4X,'STABLE SYSTEM AS VAPOUR'/)
IF (IV .GE. ILIM) THEN
  590
            IFLG(11)=0
            ALPHA=1.DO
            BETA=0.DO
            DO 120 I=1,N
               YN(I)=Z(I)
               X1N(I)=0.D0
               X2N(I)=0.D0
  120
            CONTINUE
            GO TO 330
          ELSE
            GO TO 20
```

```
END IF
      ELSE
        IV=0
      END IF
      IF (S3 .LT. 1.D0 .AND. S4 .LT. 1.D0 .OR.
S5 .LT. 1.D0 .AND. S6 .LT. 1.D0) THEN
     +
        IL=IL+1
        IF (IFLG(15) .NE. 0) WRITE(6,600)
        FORMAT(/4X,'STABLE SYSTEM AS LIQUID'/)
IF (IL .GE. ILIM) THEN
IFLG(11)=1
  600
           ALPHA=0.DO
           BETA=1.DO
           DO 130 I=1,N
             YN(I)=0.D0
             X1N(I)=Z(I)
             X2N(I)=0.D0
  130
           CONTINUE
           GO TO 330
        ELSE
           GO TO 20
        END IF
      ELSE
        IL=0
      END IF
      IF (S2 .LE. 1.D0 .OR. S3 .LE. 1.D0) GO TO 180
* Test for the existence of vapour and liquid 1
      BET=BETA
      DO 150 J=1,15
        GAMVL1=0.DO
        DGMVL1=0.D0
        DO 140 I=1,N
           DEN=BET*DKV(I)+KV(I)
           GAMVL1=GAMVL1+Z(I)*DKV(I)/DEN
           DGMVL1=DGMVL1-Z(I)*(DKV(I)/DEN)**2
  140
         CONTINUE
         IF (DABS(GAMVL1) .LT. 1.D-5) GO TO 160
         BO=BET-GAMVL1/DGMVL1
         BET=BO
  150 CONTINUE
  160 SO=0.DO
      DO 170 I=1,N
         SO=SO+Z(I)*OMKV(I)/(BET*DKV(I)+KV(I))
  170 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,610) ALP,S0
  610 FORMAT(4X, 'TEST FOR VL1'/
+ 4X, 'ROOT BET OF GAMVL1=',G11.4,2X, 'GAMVL2=',F9.5)
      IF (SO .LT. O.DO) THEN
         IVL1=IVL1+1
         IVLL=0
         IF (IFLG(15) .NE. 0) WRITE(6,620)
         FORMAT(/4X, 'STABLE SYSTEM AS VL1'/)
  620
         IF (IVL1 .GE. ILIM) THEN
           BETA=1.DO-ALPHA
           IFLG(11)=2
           GO TO 290
         ELSE
           GO TO 20
         END IF
      ELSE
```

```
IVL1=0
  END IF
180 IF (S1 .LE. 1.D0 .OR. S5 .LE. 1.D0) GO TO 230
* Test for the existence of vapour and liquid 2
      ALP=ALPHA
      DO 200 J=1,15
        GAMVL2=0.DO
        DGMVL2=0.D0
        DO 190 I=1,N
          DEN=1.DO-ALP*OMKV(I)
          GAMVL2=GAMVL2+Z(I)+OMKV(I)/DEN
          DGMVL2=DGMVL2+Z(I)*(OMKV(I)/DEN)**2
  190
        CONTINUE
        IF (DABS(GAMVL2) .LT. 1.D-5) GO TO 210
        AO=ALP-GAMVL2/DGMVL2
        ALP=AO
  200 CONTINUE
  210 S0=0.D0
      DO 220 I=1,N
        SO=SO+Z(I)*DKV(I)/(1.DO-ALP*OMKV(I))
  220 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,630) ALP,S0
 630 FORMAT(4X,'TEST FOR VL2'/
+ 4X,'ROOT ALP OF GAMVL2=',G11.4,2X,'GAMLL=',F9.5)
IF (S0 .LT. 0.D0) THEN
        IVL2=IVL2+1
        IVLL=0
        IF (IFLG(15) .NE. 0) WRITE(6,640)
        FORMAT(/4X, 'STABLE SYSTEM AS VL2'/)
  640
        IF (IVL2 .GE. ILIM) THEN
          BETA=0.DO
          IFLG(11)=2
          GO TO 290
        ELSE
          GO TO 20
        END IF
      ELSE
        IVL2=0
      END IF
  230 IF (S4 .LE. 1.D0 .OR. S6 .LE. 1.D0) GO TO 280
* Test for the existence of liquid 1 and liquid 2
      BET=BETA
      DO 250 J=1,15
        GAMLL=0.DO
        DGAMLL=0.DO
        DO 240 I=1,N
          DEN=BET*OMKL(I)-1.DO
          GAMLL=GAMLL+Z(I)*OMKL(I)/DEN
          DGAMLL=DGAMLL-Z(I)*(OMKL(I)/DEN)**2
  240
        CONTINUE
        IF (DABS(GAMLL) .LT. 1.D-5) GO TO 260
        BO=BET-GAMLL/DGAMLL
        BET=BO
  250 CONTINUE
  260 S0=0.D0
      DO 270 I=1,N
        SO=SO+Z(I)*OMKV(I)/(1.DO-BET*OMKL(I))
  270 CONTINUE
      IF (IFLG(15) .NE. 0) WRITE(6,650) BET,SO
```

```
650 FORMAT(4X,'TEST FOR L1-L2'/
+ 4X,'ROOT BET OF GAMLL=',G11.4,2X,'GAMVL2=',F9.5)
IF (S0 .GT. 0.D0) THEN
         ILL=ILL+1
         IVLL=0
         IF (IFLG(15) .NE. 0) WRITE(6,660)
FORMAT(/4X,'STABLE SYSTEM AS LL'/)
IF (ILL .GE. ILIM) THEN
  660
            ALPHA=0.DO
            IFLG(11)=3
            GO TO 290
                                                                 . .
         ELSE
            GO TO 20
         END IF
       ELSE
         ILL=0
       END IF
  280 IVLL=IVLL+1
       IF (IFLG(15) .NE. 0) WRITE(6,670)
  670 FORMAT(/4X,'STABLE SYSTEM AS VLL'/)
       IF (IVLL .GE. ILIM) THEN
IFLG(11)=4
         GO TO 330
       ELSE
         GO TO 20
       END IF
* Results of the calculation
  290 IF (ALPHA .EQ. 0) THEN
         DO 300 I=1,N
           YN(I)=0.DO
            X1N(I)=X1(I)
            X2N(I) = X2(I)
  300
         CONTINUE
         GO TO 330
       ELSE IF (BETA .EQ. O.DO) THEN
         DO 310 I=1,N
            YN(I)=Y(I)
            X1N(I)=0.D0
            X2N(I)=X2(I)
  310
         CONTINUE
         IFLG(14)=0
         GO TO 330
       ELSE IF (BETA .EQ. 1.DO-ALPHA) THEN
         DO 320 I=1.N
            YN(I)=Y(I)
           X1N(I)=0.DO
X2N(I)=X1(I)
         CONTINUE
  320
         IFLG(14)=1
      END IF
  330 RETURN
       END
```

\* SUBROUTINE FOR LIMITING V/F OR L1/F AS IN A 2-PHASE CALCULATION

SUBROUTINE LIMAB(N,RK,C) IMPLICIT REAL\*8 (A-H,O-Z) DIMENSION RK(20) RKM=RK(1) DO 20 I=2,N

```
IF (C .LT. 0.D0) GO TO 10
IF (RK(I) .LT. RKM) RKM=RK(I)
GO TO 20
10 IF (RK(I) .GT. RKM) RKM=RK(I)
20 CONTINUE
CO=1.D0/(1.D0-RKM)
IF (C .LT. 0.D0 .AND. CO .GT. 0.D0) THEN
C=-C0
GO TO 30
ELSE IF (C .GT. 0.D0 .AND. CO .LT. 0.D0) THEN
C=1.D0-C0
GO TO 30
END IF
IF (C .LT. 0.D0 .AND. C .LT. C0) C=C0/2.D0
IF (C .GT. 1.D0 .AND. C .GT. C0) C=(1.D0+C0)/2.D0
30 RETURN
END
```

.

```
*
                     PROGRAM PHASEQ (PHASE EQUILIBRIA)
* Purpose:
  Main program to calculate a VL, LL or VLL isothermal flash using
 the Soave-Redlich-Kwong or the Peng-Robinson EOS as described in
 Appendix B
* Main variables
  -Alphanumeric
  DAMP, EOS, TYPEQ, SYSTEM, ZV, ZL1, ZL2 as defined in program PHASPLIT
           Indicates if the initial estimates are supplied by the
  INIT
           user or if calculated with the initialization routines
*
  -Numeric
  ALPHA, BETA, IFLG, IFTP, ITMAX, KIJ, N, P, PC, PP, T, TC, W, Z as defined in
  program PHASPLIT
  DG
           Dimensionless Gibbs energy of mixing
  EPS
*
           Convergence tolerance
  FUGC1
           Vector of fug. coeffs. for the vapour (VL or VLL calc.)
           or for liq. 1 (LL calc.)
*
  FUGC2
           Vector of fug. coeffs. for the liq. (VL calc.), for liq.
           1 (VLL calc.) or for liq. 2 (LL calc.)
  FUGC3
*
           Vector of fug. coeffs. for liq. 2 (VLL calc.)
           Control variable to apply the composition criterion in
*
  10
           selection of key components; 0=not used; 1=used
           Key component for liq. 1; if 0 it is calculated
  I1
*
   12
*
   ITEQ
           Indicator of the type of calculation: 1=VL; 2=LL; 3=VLL
           Number of iterations required to reach convergence
   ITNUM
  IWEG
           Frequency of application of Wegstein's method
*
  K1
           Equilibrium constants yi/x2i at equilibrium
                            ii.
*
  K2
                                  x1i/x2i at
           Vector of VL or LL equilibrium constants
*
  KEQ
*
           Vector of mole fractions for the only liquid (VL calc.)
  X
*
           or for liq. 2 (LL calc.)
  X1
           Vector of mole fractions for liq. 1 (VLL calc.)
*
*
  X2
                                               2
             ...
                  **
                              ...
                                      ...
  Y
                       ..
                                         the vapour (VL or VLL calc.)
           or for liq. 1 (LL calc.)
  ZED
           3 element vector with the compressibility factors for the
           VLL calc.; the first corresponds to the vapour, the second to liq. 1 and the last to liq. 2
* System's flags:
  0,4,7,8,9,10,12,13,14,16,18 as described in the program PHASPLIT
*
*
  5,6,11,15,17 not used
                       IFLG(i)
  i
*
  1
       1=extrapolated density for the vapour (VL or VLL calc.) or
       for liq. 1 (LL calc.)
       1=extrap. density for the liquid (VL calc.), for liq. 1 (VLL calc.) or liq. 2 (LL calc.)
  2
*
  3
       1=extrap. density for liq. 2 (VLL calc.)
* Subroutines called:
  IFLL
  IFVL
  IFVLL
```

```
FLASH2
*
  FLASH3
   GIBBS
IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KEQ(20), KIJ(20,20), K1(20), K2(20)
      CHARACTER SYSTEM*72, EOS*25, TYPEQ*10, DAMP*16, INIT*20,
     +
                 ZV*8,ZL1*9,ZL2*9
      DIMENSION FUGC1(20), FUGC2(20), FUGC3(20), IFLG(0:20),
PC(20), PP(20), TC(20), W(20),
     +
                 X(20), X1(20), X2(20), Y(20), Z(20), ZED(3)
     +
      ITOTIM=0
      DO 10 I=0,20
        IFLG(I)=0
     CONTINUE
  10
      DO 30 I=1,20
        DO 20 J=1,20
          KIJ(I,J)=0.D0
  20
        CONTINUE
  30
      CONTINUE
      READ(5,*) SYSTEM, IFLG(0), N, ITMAX, EPS, IFLG(7), IWEG, IFLG(8),
                ITEQ, IFTP, T, P, QIN, QFN, DELQ,
IFLG(9), IFLG(10), I0, I1, I2, ALPHA, BETA
     +
     +
      DO 40 I=1,N
        READ(5,*) Z(I),TC(I),PC(I),W(I),PP(I)
  40
     CONTINUE
      DO 60 I=1,N
        DO 50 J=1,N
           IF (I .GE. J) GO TO 50
           READ(5,*) KIJ(I,J)
           KIJ(J,I)=KIJ(I,J)
        CONTINUE
  50
  60 CONTINUE
* Printing of calculation conditions
      WRITE(6,500) SYSTEM
     FORMAT (//2X,A60)
 500
      IF (IFLG(0) .EQ. 0) THEN
        EOS='SOAVE-REDLICH-KWONG EOS'
      ELSE
        EOS='PENG-ROBINSON EOS'
      END IF
      IF (ITEQ .EQ.1) THEN
        TYPEQ='VL FLASH'
      ELSE IF (ITEQ. EQ. 2) THEN
TYPEQ='LL FLASH'
      ELSE
        TYPEQ='VLL FLASH'
      END IF
      WRITE(6,510) EOS, TYPEQ
 510
      FORMAT(/4X,A25,2X,A10)
      IF (IFLG(7) .EQ. 0) THEN
        WRITE(6,520)
 520
        FORMAT(4X, 'WITHOUT WEGSTEINS METHOD')
        GO TO 70
      END IF
      IF (IFLG(8) .EQ. 0) THEN
        DAMP=' WITHOUT DAMPING'
      ELSE
        DAMP=' WITH DAMPING'
```

\* \*

\*

```
END IF
       WRITE(6,530) IWEG, DAMP
530 FORMAT(4X, 'WEGSTEINS METHOD EACH', 13, ' ITERATIONS', A16)
 70 WRITE(6,540) ITMAX, EPS
540 FORMAT(4X, 'MAXIMUM NO. OF ITERATIONS=', I3, 2X,
               'CONVERGENCE=',F8.5)
      IF (IFLG(10) .EQ. 1) THEN
         WRITE(6,550)
         FORMAT(4X, 'SEQUENTIAL CALCULATION')
550
       END IF
       IF (IFLG(9) .EQ. 1) THEN
         INIT='AUTO-INITIALIZATION'
       ELSE
         INIT='USER INITIALIZATION'
      END IF
       WRITE(6,560) INIT
 560 FORMAT(4X,A20/)
      DO 90 I=1,N
         DO 80 J=1,N
           IF (I .GE. J .OR. KIJ(I,J) .EQ. 0.D0) GO TO 80
WRITE(6,570) I,J,KIJ(I,J)
FORMAT(8X,'K(',I2,',',I2,')=',F7.4)
 570
         CONTINUE
  80
      CONTINUE
  90
      IF (IFTP .EQ. 0) GO TO 115
IF (IFTP .EQ. 3) THEN
         WRITE(6,575) T,P
         FORMAT(/4X,'T=',F8.3,' K',4X,'P=',F8.3,' atm'/)
 575
         GO TO 100
       END IF
* Loop for calculations at fixed T and various P or at fixed P at
* various T
       DO 310 Q=QIN,QFN,DELQ
         IF (IFTP .EQ.1) THEN
           P=Q
         ELSE
           T=Q
      END IF
       GO TO 115
* Loop for calculations at fixed T and P ar different compositions
 100 IF (N .GT. 3) THEN
        WRITE(6,577)
FORMAT(/4X,'WARNING: CALC. AT FIXED T & P AT DIFFERENT'
/4X,'COMPOSITIONS ALLOWED ONLY FOR BINARY OR'
/4X,'TERNARY MIXTURES')
577
     +
         GO TO 310
      END IF
       IF (N .EQ. 2) THEN
         DZ=0.05D0
       ELSE
         DZ=0.1D0
      END IF
      IF (N .EQ. 2) GO TO 110
DO 305 ZF1=1.D-8,1.D0+DZ,DZ
      DO 300 ZF2=1.D-8,1.D0+DZ,DZ
         Z(1)=ZF1
         Z(2)=ZF2
         Z(3)=1.D0-ZF1-ZF2
         IF (Z(1) .GT. 1.DO) Z(1)=1.DO-1.D-8
```

IF (Z(2) .GT. 1.D0) Z(2)=1.D0-1.D-8 IF (Z(3) .LT. 0.D0) Z(3)=1.D-8 IF (ZF1+ZF2 .GT. 1.D0+DZ) G0 T0 300 ZT0T=Z(1)+Z(2)+Z(3) Z(1)=Z(1)/ZTOTZ(2)=Z(2)/ZTOT Z(3)=Z(3)/ZTOT GO TO 115 110 DO 295 ZA1=1.D-8,1.D0+DZ,DZ Z(1)=ZA1Z(2)=1.DO-ZA1 IF (Z(2) .LE. 0.D0) Z(2)=1.D-8 IF ((IFLG(9) .EQ. 0) .OR. (IFLG(10) .EQ. 1 .AND. IFLG(12) .EQ. 1)) GO TO 150 115 \* Calculation of the unknown variables using the initialization \* routines I1=0 I2=0 GO TO (120,130,140) ITEQ CALL TIME(0,0, ITDUM) 120 CALL IFVL(BETA,P,T,N,X,Y,Z,TC,PC,W,PP,KIJ,KEQ,IFLG) CALL TIME(1,0, ITIMI) GO TO 200 130 CALL TIME(0,0,ITDUM) CALL IFLL(IO, I1, I2, BETA, P, T, N, X, Y, Z, TC, PC, W, PP, KIJ, KEQ,IFLG) CALL TIME(1,0, ITIMI) GO TO 200 CALL TIME(0,0,ITDUM) CALL IFLLV(I0,I1,I2,BETA,ALPHA,P,T,N,Y,X1,X2,Z,TC,PC, 140 W, PP, KIJ, KEQ, IFLG) CALL TIME(1,0,ITIMI) GO TO 200 \* The initial composition estimates are read from the data file \* when user supplied estimates are given IF (IFLG(10) .EQ. 1 .AND. IFLG(12) .EQ. 1) GO TO 200 DO 190 I=1,N 150 GD TO (160,170,180) ITEQ 160 READ(5,\*) Y(I), X(I)GO TO 190 READ(5,\*) Y(I),X(I) 170 GO TO 190 180 READ(5,\*) Y(I),X1(I),X2(I) CONTINUE 190 IF (ITEQ .EQ. 1 .AND. IFLG(12) .EQ. 1 .AND. IFLG(10) .EQ. 1) BETA=ALPHA 200 IF (ITEQ .EQ. 1 .AND. IFLG(12) .EQ. 0 .AND. IFLG(9) + .EQ. O) BETA=ALPHA IF (ITEQ .EQ. 2 .AND. IFLG(12) .EQ. 1 .AND. IFLG(10) .EQ. 1) THEN +DO 205 I=1,N Y(I) = X1(I)205 CONTINUE END IF \* Printing of the initial estimates GO TO (210,220,230) ITEQ 210 WRITE(6,580) T,P,BETA

```
580
          FORMAT(/4X,'VL FLASH INITIALIZATION',2X,'T=',F8.3,' K'
         2X,'P=',F8.3,' atm',2X,'V/F=',F8.5/9X,'I',5X,'Z',
9X,'Y',8X,'X')
WRITE(6,590) (I,Z(I),Y(I),X(I), I=1,N)
          GD TO 240
         GU 10 240

WRITE(6,600) T,P,BETA

FORMAT(/4X,'LL FLASH INITIALIZATION',2X,'T=',F8.3,' K',

2X,'P=',F8.3,' atm',2X,'L1/F=',F8.5/9X,'I',5X,

'Z',8X,'X1',8X,'X2')

WRITE(6,590) (I,Z(I),Y(I),X(I), I=1,N)

TO F 17 F0 F 17 F0 F 17 F0 5)
 220
 600
      +
 590
          FORMAT(8X,12,1X,F8.5,1X,F8.5,1X,F8.5)
          GO TO 240
 230
          WRITE(6,610) T, P, ALPHA, BETA
         610
      +
          FORMAT(8X, 12, 1X, F8.5, 1X, F8.5, 1X, F8.5, 1X, F8.5)
 620
 240
          IFLG(12)=1
          WRITE(6,622) ITIMI
 622
          FORMAT(4X,'CPU TIME=',I4,' msecs')
* Isothermal flash calculation routines
          GO TO (250,250,260) ITEQ
CALL TIME(0,0,ITDUM)
 250
          CALL FLASH2(ITEQ, EPS, ITMAX, ITNUM, IWEG, OBJ, N, P, T, BETA, X, Y, Z,
                         TC, PC, W, PP, KIJ, FUGC1, FUGC2, Z1, Z2, KEQ, IFLG)
      +
          CALL TIME(1,0,ITIMC)
          GO TO 270
 260
          CALL TIME(0,0, ITDUM)
          CALL FLASH3 (EPS, ITMAX, ITNUM, IWEG, OBJ, N, P, T, ALPHA, BETA, Y, X1,
                         X2,Z,TC,PC,W,PP,KIJ,ZED,FUGC1,FUGC2,FUGC3,IFLG)
          CALL TIME(1,0,ITIMC)
* System's Gibbs energy of mixing calculation
 270
          CALL GIBBS(ITEQ, N, P, T, ALPHA, BETA, Y, X, X1, X2, Z, TC, PC, W, PP, KIJ,
                        FUGC1, FUGC2, FUGC3, DG, IFLG)
* Printing of results of calculation
          ONMAB=1.DO-ALPHA-BETA
          IF (ITEQ .EQ. 2) THEN ZED(1)=0.D0
             ZED(2)=Z1
             ZED(3)=Z2
          ELSE IF (ITEQ .EQ. 1) THEN
             ZED(1)=Z1
             ZED(2)=0.D0
             ZED(3)=Z2
          ENDIF
          WRITE(6,625) TYPEQ
          FORMAT (//15X,A10,2X,'CALCULATION RESULTS')
 625
          WRITE(6,630) T,P,ITNUM,OBJ,ALPHA,BETA,ONMAB,DG
         FORMAT(/4X, 'T=',F8.3,' K',2X,'P=',F8.3,' atm',2X,
'NUM. ITER.=',I3,2X,'OBJ. FUNC.=',E12.5/
4X,'V/F=',F8.5,2X,'L1/F=',F8.5,2X,'L2/F=',
 630
      +
                   F8.5,2X,'DEL G=',F11.3,' cal/gmole')
      +
          DO 280 I=1,N
             IF (ITEQ .EQ. 2) THEN
               K1(I)=1.D0
             ELSE
```

```
K1(I)=FUGC3(I)/FUGC1(I)
           END IF
            IF (ITEQ .EQ. 1) THEN
              K2(I)=1.D0
            ELSE
              K2(I)=FUGC3(I)/FUGC2(I)
           ENDIF
 280
         CONTINUE
         IF (IFLG(3) .EQ. 1) THEN
            ZL2='ZLIQ2ext='
         ELSE
            ZL2='ZLIQ2='
         END IF
         IF (IFLG(2) .EQ. 1) THEN
           ZL1='ZLIQ1ext='
         ELSE
           ZL1='ZLIQ1='
         ENDIF
         IF (IFLG(1) .EQ. 1) THEN
            ZV='ZVAPext='
         ELSE
           ZV='ZVAP='
         END IF
         WRITE(6,640) ZV,ZED(1),ZL1,ZED(2),ZL2,ZED(3)
  640
         FORMAT(4X, A8, F7.5, 2X, A9, F7.5, 2X, A9, F7.5)
         IF (IFLG(9) .EQ. 0) GO TO 290
IF (ITEQ .NE. 1 .AND. IO .EQ.1) WRITE(6,650) I1,I2
FORMAT(4X,'KEY COMP.#1=',I2,2X,'KEY COMP.#2=',I2,
  650
         3X, 'COMP. CRITERION')
IF (ITEQ .NE. 1) WRITE(6,660) I1,I2
FORMAT(4X, 'KEY COMP.#1=',I2,2X, 'KEY COMP.#2=',I2)
      +
  660
  290
         WRITE(6,670)
         FORMAT(/8X,'I',5X,'Z',8X,'Y',8X,'X1',7X,'X2',5X,
'KV=Y/X2',2X,'KL=X1/X2')
WRITE(6,680) (I,Z(I),Y(I),X1(I),X2(I),K1(I),K2(I),
  670
     +
      +
                          I=1,N)
  680
         FORMAT(7X,I2,1X,F8.5,1X,F8.5,1X,F8.5,1X,F8.5,2X,
                  G11.4,2X,G11.4)
     +
         WRITE(6,690) ITIMC
         FORMAT(4X, 'CPU TIME=', I4, ' msecs')
  690
         WRITE(6,700) ITIMI+ITIMC
  700
         FORMAT(/4X,'TOT CPU TIME=',I4,' msecs')
         WRITE(6,710)
FORMAT(/'=======:')
  710
         ITOTIM=ITOTIM+ITIMI+ITIMC
* End of loop for calculation at different compositions at fixed T
  and P
*
  295 CONTINUE
  300 CONTINUE
  305 CONTINUE
* End of loop for calculation at fixed T and various P or at fixed
* P and different T
  310 CONTINUE
       WRITE(6,720) ITOTIM
  720 FORMAT(/4X,'GLOBAL CPU TIME=', I5, ' msec'/)
       STOP
       END
```

```
TIJ92AH9
                                                                  PHASEQ
                                                               Called by:
                                                                     OTMA
                                                                      MEG
                                                                        Х
                                                     Subroutines called:
                                                 past iteration
      Vector of mole fracs. for the liquid or liq. 2 in the
                                                                       ₽X
                                                 noitereti taaq
      Vector of mole fracs. for the vapour or liq. 1 in the
                                                                       X3
         Value of the object function in the past iteration
                                                                     OBJL
                          " biupil
                                                                       ZS
                     ...
                                    ...
                                         ...
             Sum of mole fracs. for the vapour or for lid. 1
                                                                       IS
        Equilibrium constants vector for the past iteration
                                                                       ĽХ
                            for which V/F or L1/F is <0 or >1
  Control variable to count the number of successive iters.
*
                                                                       13
                                        Tespect to V/F or L1/F
*
      Temporary value of the ratio V/F or L1/F
Rachford-Rice type object function
Partial derivative of the Rachford-Rice function with
                                                                       ΕO
                                                                        ь
                                                                       BO
*
                                                          Rain variables:
                        biupil
                                                                       ZZ
 Compressibility factor for the vapour (VL) or liq. 1 (LL)
                                                                       τz
                                                 or liq. 2 (LL)
    Vector of calculated mole fractions for the liquid (LL)
                                                                       X7
                                                                            *
                                                 or 114. 1 (LL)
    Vector of calculated mole fractions for the vapour (VL)
                                                                       τx
                                      the phases where present
    value of the fugacity differences for each component in
       Value of the objective function: sum of the absolute
                                                                      081
                      Molar phase ratio V/F (VL) of LI/F (LL)
                                                                     BETA
                  ITNUM, FUGC1, FUGC2, KEQ as defined in program PHASEQ
                                                                   JuqJu0-
                                              liq. 2 (LL calc.)
  Mole fraction estimates for the liquid (VL calc.) or for
                                                                       X2
                                              liq. 1 (LL calc.)
  Mole fraction estimates for the vapour (VL calc.) or for
                                                                       IX
        TIMAX, M, P, T, Z, TC, PC, W, PP, KIJ as defined in program PHASPLIT
BETA Initial value of the phase ratio V/F or L1/F
                     ITEQ, EPS, IWEG, IFLG as defined in program PHASEQ
                                                                    indui-
                                                              Parameters:
                      calculation can be done for VL or LL equilibria
 composition z is flashed at constant temperature and pressure. The
                                                                            *
        phases originated when a mixture of n components with global
                                                                            *
 Calculate the equilibrium compositions and the phase ratio of the
                                                                  Purpose:
                                                                            *
                 ************
            SUBROUTINE FLASH2 (2-PHASE ISOTHERMAL FLASH)
```

Appendix C. Description and listings of the computer programs

```
SUBROUTINE FLASH2(ITEQ, EPS, ITMAX, ITNUM, IWEG, OBJ, N, P, T,
BETA, X2, X1, Z, TC, PC, W, PP, KIJ, FUGC1,
      +
                              FUGC2, Z1, Z2, KEQ, IFLG)
      +
      Implicit Real+8 (A-H, 0-Z)
REAL+8 KEQ(20),KIJ(20,20),K1(20)
DIMENSION FUGC1(20),FUGC2(20),IFLG(0:20),PC(20),PP(20),
TC(20),W(20),X1(20),X2(20),X3(20),X4(20),Z(20),
TC(20), BDUP(20)
      +
                    APUR(20), BPUR(20)
      +
       ITNUM=0
       IFLG(4)=0
IFLG(13)=0
       IC1=0
       IC2=0
       IC3=0
       IC4=0
       IC5=0
       J3=0
       W1=0.D0
* Calculation of the initial K values when not available
       IF (IFLG(9) .EQ. 1 .AND. IFLG(10) .EQ. 0 .OR. IFLG(18) .EQ. 1)
      +
           GO TO 5
       CALL K(X2,X1,TC,PC,W,PP,P,T,N,ITEQ,3,APUR,BPUR,FUGC1,
FUGC2,Z1,Z2,KIJ,KEQ,IFLG)
      +
* Start of the iterative cycle
    5 ITNUM=ITNUM+1
       L2=0
       IF (ITNUM .GT. ITMAX) GO TO 170
* Calculation of beta with fixed K's using Newton's method
       DO 20 J=1,15
          F=0.D0
          F0=0.D0
          DO 10 I=1,N
RK4=BETA*(KEQ(I)-1.D0)+1.D0
            F=F+Z(I)*(KEQ(I)-1.DO)/RK4
            F0=F0-Z(I)*((KEQ(I)-1.D0)/RK4)**2
          CONTINUE
    10
          IF (DABS(F) .LT. 1.D-5) GO TO 30
          BO=BETA-F/FO
          BETA=BO
   20 CONTINUE
* Calculation of the LIMITING value of beta when not in the
* interval [0,1]
   30 IF (BETA .GE. O.DO .AND. BETA .LE. 1.DO) GO TO 60
       RK5=KEQ(1)
       DO 50 I=2,N
          IF (BETA .LT. 0.D0) GO TO 40
IF (KEQ(I) .LT. RK5) RK5=KEQ(I)
          GO TO 50
IF (KEQ(I) .GT. RK5) RK5=KEQ(I)
    40
    50 CONTINUE
       B3=1.D0/(1.D0-RK5)
       IF (BETA .LT. O.DO .AND. B3 .GT. O.DO) THEN
          BETA=-B3
          GO TO 60
       END IF
```

```
IF (BETA .GT. O.DO .AND. B3 .LT. O.DO)THEN
         BETA=1.DO-B3
         GO TO 60
       END IF
      IF (BETA .LT. O.DO .AND. BETA .LT. B3) BETA=B3/2.DO
IF (BETA .GT. 1.DO .AND. BETA .GT. B3) BETA=(1.DO+B3)/
           2.D0
   60 IF (BETA .LT. O.DO .OR. BETA .GT. 1.DO) THEN
         J3=J3+1
      ELSE
         J3=0
       END IF
* If the value of beta has been outside the interval [0,1] for
* five consecutive iterations, the calculation ends indicating
* the presence of only one phase
       IF (J3 .GE. 5) GO TO 180
* Calculation of the new compositions in both phases
   70 SUM1=0.D0
       SUM2=0.D0
       DO 80 I=1,N
         X2(I)=Z(I)/(BETA*(KEQ(I)-1.D0)+1.D0)
         IF (X2(I) .LT. 0.D0) X2(I)=0.D0
         X1(I) = KEQ(I) * X2(I)
         SUM1=SUM1+X1(I)
         SUM2=SUM2+X2(I)
   80 CONTINUE
       DO 90 I=1,N
         X1(I)=X1(I)/SUM1
         X2(I)=X2(I)/SUM2
   90 CONTINUE
* Starting in the third iteration Wegstein's method is used (optional)
      IF (IFLG(7) .EQ. 0 .OR. ITNUM .LT. 3) GO TO 110
IF (ITNUM .LE. 5 .AND. (BETA .LT. 0.DO .OR. BETA .GT.
1.DO)) GO TO 110
     +
      IF (ITNUM .LT. IC1+IWEG) GO TO 110
      IF (W1 .LE. -0.2D0) GO TO 100
IF (DBJ .GE. OBJL) GO TO 110
IF (DBJ .GT. 0.2D0) GO TO 110
  100 CALL WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X1, X3)
       CALL WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X2, X4)
* Calculation of the new K values
  110 CALL K(X2,X1,TC,PC,W,PP,P,T,N,ITEQ,3,APUR,BPUR,FUGC1,
               FUGC2, Z1, Z2, KIJ, KEQ, IFLG)
     +
* Calculation of the objective function
       OBJL=OBJ
       OBJ=0.DO
       DO 120 I=1,N
         OBJ=OBJ+DABS(FUGC1(I)*X1(I)-FUGC2(I)*X2(I))
  120 CONTINUE
* Test for convergence
       IF (OBJ .LT. EPS) GO TO 160
```

```
* Calculation of the damping factor for Wegstein's method (optional)
      IF (IFLG(7) .EQ. 0 .OR. IFLG(8) .EQ. 0) GO TO 140
      IF (ITNUM .LT. 3) GO TO 140
      IF (L2 .EQ. 1) GO TO 140
CALL AMTO(OBJ,OBJL,L2,W1,IC2,IC3,IC4,IC5)
IF (IC5 .EQ. 0) GO TO 140
      IC5=0
      IC1=IC1-IWEG
      DO 130 I=1,N
         KEQ(I)=K1(I)
  130 CONTINUE
      GO TO 70
* Storage of the last composition vector to be used in Wegstein's
* method ·
  140 DO 150 I=1.N
          K1(I) = KEQ(I)
          X3(I)=X1(I)
X4(I)=X2(I)
  150 CONTINUE
      GO TO 5
* Convergence is not allowed if Wegstein's method has been applied in
* the present iteration
  160 IF (IC1 .EQ. ITNUM) GO TO 140
* Convergence: test to see if two phases exist
  170 IF (BETA .LT. 0.DO .OR. BETA .GT. 1.DO) GO TO 180
      GO TO 200
* Feed composition outside the two phase region
  180 DO 190 I=1,N
         X1(I)=Z(I)
X2(I)=Z(I)
  190 CONTINUE
       IFLG(12)=0
      IF (BETA .LT. 0.DO) BETA=0.DO
IF (BETA .GT. 1.DO) BETA=1.DO
  200 RETURN
      END
```

## Appendix C. Description and listings of the computer programs

```
**********
          SUBROUTINE IFVL (INITIALIZATION FOR THE VL FLASH)
                                                                  *
*
   ************
*
 Purpose:
  Provide initial estimates of the vapour and liquid compositions
 as well as for the phase fraction V/F to initialize the VL flash
 calculation
*
*
 Parameters:
 -Input
*
  P,T,N,Z,TC,PC,W,PP,KIJ as defined in program PHASEQ
 -Output
*
*
  ALPĤA
          Estimated value of the ratio V/F
                   mole fractions for the liquid
*
  X
              ..
                            ...
                                  ...
                                      " vapour
*
  Y
          Vector of initial equilibrium constants
*
  KEO
*
 Main variables:
*
  C1
          Vector of non-normalized vapour compositions
            54
                 ...
                        ....
  C2
                                  liquid
  PVAP
          Ratio of the vapour pressure to the total pressure
*
          Sum of the vapour compositions
""" liquid "
  S1
*
  S2
*
*
 Subroutines called:
*
*
  ĸ
* Called by:
  PHASEQ
***********
     SUBROUTINE IFVL(ALPHA, P, T, N, X, Y, Z, TC, PC, W, PP, KIJ, KEQ,
    +
                     IFLG)
     IMPLICIT REAL*8 (A-H, D-Z)
REAL*8 KEQ(20), KIJ(20, 20)
     DIMENSION FUGC1(20), FUGC2(20), IFLG(0:20), PC(20), PP(20),
               TC(20), W(20), X(20), Y(20), Z(20), C1(20), C2(20),
               APUR(20), BPUR(20)
    +
     IFLG(4)=0
     IFLG(13)=0
* Calculation of liquid and vapour compositions using Raoult's law
     SUM1=0.DO
     SUM2=0.D0
     DO 10 I=1,N
       PVAP=PC(I)*DEXP(5.3727D0*(1.D0+W(I))*(1.D0-TC(I)/T))/P
       C1(I)=Z(I)*PVAP
       C2(I)=Z(I)/PVAP
       SUM1=SUM1+C1(I)
       SUM2=SUM2+C2(I)
   10 CONTINUE
* Normalization of the calculated compositions
     DO 20 I=1,N
       Y(I)=C1(I)/SUM1
```

.

```
X(I)=C2(I)/SUM2
   20 CONTINUE
* Calculation of the initial VL equilibrium constants
      CALL K(X,Y,TC,PC,W,PP,P,T,N,1,3,APUR,BPUR,FUGC1,FUGC2,Z1,
Z2,KIJ,KEQ,IFLG)
      +
* Calculation of the estimated value of alpha
      SUM1=0.DO
      SUM2=0.D0
      DO 30 I=1,N
SUM1=SUM1+KEQ(I)*Z(I)
         SUM2=SUM2+(1.DO-KEQ(I))*Z(I)/(1.DO+KEQ(I))
   30 CONTINUE
      SUM1=1.DO-SUM1
      SUM2=2.DO*SUM2
      F=SUM2/(SUM2-SUM1)
      IF (F .GT. 0.DO) THEN
         ALPHA=0.5D0*(1.D0-F)
         GO TO 50
      END IF
      SUM3=0.D0
      DO 40 I=1,N
         SUM3=SUM3+Z(I)/KEQ(I)
   40 CONTINUE
      SUM3=SUM3-1.DO
      ALPHA=(SUM2-0.5D0*SUM3)/(SUM2-SUM3)
   50 IF (ALPHA .LT. 0.05D0) ALPHA=0.05D0
IF (ALPHA .GT. 0.95D0) ALPHA=0.95D0
      RETURN
      END
```

```
*******
          SUBROUTINE IFLL (INITIALIZATION FOR THE LL FLASH)
                                                                   *
********
* Purpose:
  Select the key components for liquid phases 1 and 2 and/or obtain *
*
* the initial composition values as well as the phase fraction L1/F
* to start the LL flash calculation
* Parameters:
*
 -Input
 IO, I1, I2, P, T, N, Z, TC, PC, W, PP, KIJ as defined in program PHASEQ
*
 -Output
*
*
  I1
          Key component for liq. 1
                            ...
*
  12
                                2
*
  BETA
          Initial estimate for the fraction L1/F
*
  X1
          Mole fraction estimates for liq. 1
*
  X2
                           ....
                                                                   *
*
  KEQ
          LL equilibrium constants
*
 Main variables:
          Temporary key component for liq. 1
*
  13
*
  14
          Matrix of the equilibrium cts. for all the binary pairs
  K2
*
*
  U1
          Temporary value of the equilibrium ct. for the probable
*
          key component for liq. 1
                                             ...
                                                 ...
                                                     ..
                                                           н
                                      ...
*
  U2
          Temporary value of the
*
          key component for liq. 2
                                                                   *
*
 Subroutines called:
*
  K
*
* Called by:
  PHASEQ
*
  IFVLL
*
*********
     SUBROUTINE IFLL(I0, I1, I2, BETA, P, T, N, X2, X1, Z, TC, PC, W, PP,
                     KIJ, KEQ, IFLG)
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL+8 KEQ(20), KIJ(20,20), K2(20,20)
DIMENSION FUGC1(20), FUGC2(20), IFLG(0:20), PC(20), PP(20),
               TC(20), W(20), X1(20), X2(20), Z(20), APUR(20),
     +
               BPUR(20)
     IFLG(4)=0
     IFLG(13)=0
     IF (I1 .NE. O .AND. I2 .NE. O) GO TO 190
* Test to see if there is more than one component with mole fraction
* greater than 0.1
     IC=0
     DO 10 I=1,N
       IF (Z(I) .GT. 0.1D0) THEN
         J=I
         IC=IC+1
       END IF
   10 CONTINUE
```
```
IF (IC .EQ. 1) THEN
         IP2=1
         I2=J
         GO TO 80
       END IF
       IF (I1 .NE. 0 .OR. I2 .NE. 0) GO TO 70
* Selection of key component for liquid 2 with the higher composition
* criterion
       IF (IO .EQ. 0) GO TO 30
       IC=0
       C2=Z(1)
       I2=1
      DO 20 I=2,N
IF (Z(I) .LT. C2) GO TO 20
IF (C2 .EQ. Z(I)) THEN
           IC=IC+1
            GO TO 20
         END IF
         C2=Z(I)
         I2=I
         IC=0
   20 CONTINUE
       IF (IC .GT. 0) GO TO 30
GO TO 70
* Calculation of both key components
   30 U1=1.D0
       U2=1.D0
       GO TO 150
   40 CONTINUE
       J1=1
       DO 60 J=2,N
         J1=J-1
         IF (Z(J) .LE. 0.1D0) GO TO 60
         DO 50 I=1,J1
IF (Z(I) .LE. 0.1D0) GO TO 50
IF (K2(J,I) .GT. U1) THEN
U1=K2(J,I)
              I3=I
            END IF
            IF (1.DO/K2(J,I) .GT. U1) THEN
U1=1.DO/K2(J,I)
              I3=I
            END IF
            IF (1.DO/K2(I,J) .GT. U2) THEN
              U_{2=1.D0/K_{2}(1, j)}
              I4=J
            END IF
            IF (K2(I,J) .GT. U2) THEN
              U_{2}=K_{2}(I,J)
              I4=J
            END IF
         CONTINUE
   50
   60 CONTINUE
       IF (U1 .GT. U2) THEN
          I1=I3
          I2=I4
       ELSE
         I1=I4
          I2=I3
```

```
END IF
       GO TO 190
* Calculation of one key component when the other is known
   70 IP2=0
   80 DO 90 I=1,N
          X1(I)=0.D0
          X2(I)=0.D0
   90 CONTINUE
       IF (I2 .NE. 0) I4=I2
IF (I1 .NE. 0) I4=I1
       U1=1.D0
       X1(I4)=0.02D0
       X2(I4)=0.98D0
       DO 140 J=1,N
         IF (J .EQ. 14) GO TO 140
X2(J)=0.02D0
          X1(J)=0.98D0
         IF (N .EQ. 2) I3=J
IF (J .EQ. I4+1) GO TO 130
IF (J .GT. 1) THEN
            X2(J-1)=0.D0
            X1(J-1)=0.D0
         END IF
IF (IP2 .EQ. 1) GO TO 110
IF (Z(J) .LE. 0.1D0) GO TO 140
  100
         CALL K(X2,X1,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUGC1,
FUGC2,Z1,Z2,KIJ,KEQ,IFLG)
IF (KEQ(J) .LT. U1) GO TO 120
  110
      +
          U1=KEQ(J)
          I3=J
          GO TO 140
          IF (1.DO/KEQ(J) .LT. U1) GO TO 140
U1=1.DO/KEQ(J)
  120
          I3=J
          GO TO 140
  130
          IF (J .GT. 2) THEN
            X2(J-2)=0.D0
            X1(J-2)=0.D0
          END IF
          GO TO 100
  140 CONTINUE
       I2=I4
       I1=I3
       GO TO 190
* Calculation of the LL equilibrium constants for all the binary
* pairs
  150 DO 160 I=1,N
          X1(I)=0.D0
          X2(I)=0.D0
  160 CONTINUE
       J1=1
       DO 180 J=2,N
          X2(J1)=0.D0
          X1(J1)=0.D0
          J1=J-1
          X2(J)=0.98D0
          X1(J)=0.02D0
          X2(J-1)=0.D0
          X1(J-1)=0.D0
```

```
DO 170 I=1,J1
           X1(I)=0.98D0
           X2(I)=0.02D0
           IF (I .GT. 1) THEN
             X1(I-1)=0.D0
             X2(I-1)=0.D0
           END IF
           CALL K(X2,X1,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUGC1,
FUGC2,Z1,Z2,KIJ,KEQ,IFLG)
     +
           K2(I,J)=KEQ(J)
           K2(J,I)=KEQ(I)
  170
         CONTINUE
  180 CONTINUE
      GO TO 40
* Allocation of the mole fractions for both liquids according to
* the calculated key components
  190 DO 200 I=1,N
        X1(I)=0.D0
         X2(I)=0.D0
  200 CONTINUE
      X1(I1)=0.98D0
      X2(I1)=0.02D0
      X1(I2)=0.02D0
      X2(I2)=0.98D0
      IF (IFLG(14) .EQ. 1) THEN
         IFLG(14)=0
         BETA=0.5DO
         GO TO 210
      END IF
* Calculation of the initial values of the LL flash equilibrium
* constants
      CALL K(X2,X1,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUGC1,
FUGC2,Z1,Z2,KIJ,KEQ,IFLG)
     +
* Calculation of the initial value of beta
      BETA=(Z(I1)*(1.D0-1.D0/KEQ(I1))+KEQ(I2)*Z(I2))/
           (Z(I1)+Z(I2))
     +
      IF (BETA .LT. 0.05D0) BETA=0.05D0
IF (BETA .GT. 0.95D0) BETA=0.95D0
  210 RETURN
      END
```

```
*************
                SUBROUTINE FLASH3 (3-PHASE ISOTHERMAL FLASH)
     ********
*
 Purpose:
  Calculate the equilibrium compositions and the phase ratios for
* the phases originated when a mixture of n components with global
 composition z is flashed at constant temperature and pressure. It
* is considered that a maximum of two liquid phases and a vapour
 exist (VLL calc.), but the calculation can be reduced to VL or LL
 equilibrium
 Parameters:
 -Input
  ITEQ, EPS, IWEG, IFLG as defined in program PHASEQ
ITMAX, N, P, T, Z, TC, PC, W, PP, KIJ as defined in program PHASPLIT
           Initial value of the phase ratio V/F
   ALPHA
   BETA
                                             L1/F
           Mole fraction estimates for the vapour
  X1
                                      .....
            ...
                    ...
                               ...
                                          " liq. 1
*
   X2
            ...
                    ...
                               ..
                                      ...
                                          14
   ΧЗ
*
*
 -Output
   ITNUM, FUGC1, FUGC2, FUGC3, ZED as defined in program PHASEQ
           Molar phase ratio V/F
   ALPHA
                   ....
                          " L1/F
   BETA
*
           Value of the objective function: sum of the absolute
   OBJ
           value of the fugacity differences for each component in
           the phases where present
*
  X1
           Vector of calculated mole fractions for the vapour
               44
                   ....
                          ....
                                   ....
                                          44
   X2
                                                     liq. 1
               ••
                   ..
                          ..
                                   ...
                                          ..
                                                   ...
                                                           2
   Х3
*
 Main variables:
           Temporary value of V/F
" " L1/F
   AO
   BO
           Determinant for the Rachford-Rice derived system of
   D
           equations used in the Newton-Raphson solution
           Partial derivatives of G1 and G2 with respect to V/F
*
  E1,E3
                         11
                                 44 . 44
                                        14
                                           - 11
                                                ...
                                                               L1/F
*
   E2.E4
           Rachford-Rice type function (eqn. 3.115)
   G1
   G2
                                                3.116)
                                          (
*
   I1
           Key component for liq. 1
*
   12
                                    2
   JЗ
           Counter for the number of successive iterations in which
*
           V/F<O or V/F>1
                                                                   ..
   J4
           Counter for the
*
           L1/F<0 or L1/F>0
                                                              ..
                                                                   н
                                н
                                    н
                                          11
                                                      ..
   J5
           Counter for the
           1-V/F-L1/F<0 or 1-V/F-L1/F>0
*
   K1
           Vector of the VL2 equilibrium constants
                         L1L2
*
   K2
              ••
                   ..
                       " VL2
                                   н
                                                ..
   KЗ
                                                     in the past itern.
                      " L1L2
              =
                   ...
                                   ....
                                                ••
   K4
*
   OBJL
           Value of the objective function in the past iteration
*
  Subroutines called:
   ĸ
   WEG
   AMTO
```

```
*
*
  Called by:
   PHASEO
                                                                                                       *
                                                                                                       *
   PHASPLIT
*
                                                                                                       *
  *************
        SUBROUTINE FLASH3(EPS,ITMAX,ITNUM,IWEG,OBJ,N,P,T,ALPHA,
BETA,X1,X2,X3,Z,TC,PC,W,PP,KIJ,ZED,
FUGC1,FUGC2,FUGC3,IFLG)
       +
       +
        IMPLICIT REAL*8 (A-H,O-Z)

      INFLICIT REAL+8 (A=1,0-2)

      REAL+8 KEQ(20), KIJ(20,20), K1(20), K2(20), K3(20), K4(20)

      DIMENSION ZED(3), FUGC1(20), FUGC2(20), FUGC3(20), FUGC4(20),

      +
      IFLG(0:20), PC(20), PP(20), TC(20), W(20), X0(20),

      +
      X1(20), X2(20), X3(20), X4(20), X5(20), X6(20),

      +
      X1(20), X2(20), X3(20), X4(20), X5(20), X6(20),

       +
       +
       ÷
                       Z(20), APUR(20), BPUR(20)
        ITNUM=0
        ITEQ=3
        IFLG(4)=0
        IFLG(13)=1
        IC1=0
        IC2=0
        IC3=0
        IC4=0
        IC5=0
         J3=0
         J4=0
         J5=0
        J6=0
        W1=0.D0
* Assignment of key components for the two liquids from their initial
* compositions
         I1=1
         I2=1
        H1=X2(1)
        H2=X3(1)
        DO 10 I=2,N
         IF (X3(I) .GT. H2) THEN
           I2=I
           H2=X3(I)
         ELSE IF (X2(I) .GT. H1) THEN
           I1=I
           H1=X2(I)
         END IF
    10 CONTINUE
* Calculation of the initial VL and LL K values
         IF (IFLG(18) .EQ. 1) GO TO 305
         GO TO 300
* Start of the iterative cycle
    20 ITNUM=ITNUM+1
         L2=0
         IF (ITNUM .GT. ITMAX) GO TO 410
         SUM=0.D0
* Storage of the last composition vector to be used in Wegstein's
* method
```

\*

\_

```
DO 30 I=1,N
         K3(I) = K1(I)
         K4(I) = K2(I)
         X4(I) = X1(I)
         X5(I)=X2(I)
         X6(I)=X3(I)
         SUM=SUM+DABS(K2(I)-1.D0)
   30 CONTINUE
* If the LL K values are near unity, one of the liquid phases is
* eliminated
       IF (SUM .LT. N*0.001D0 .AND. ITEQ .EQ. 3) THEN
         BETA=0.DO
         ITEQ=1
         J3=0
         J4=0
         J5=0
         IFLG(13)=0
       END IF
       AO=ALPHA
       BO=BETA
* Calculation of V/F and/or L1/F with VL and LL K'S fixed using
* a Newton-Raphson method
       DO 80 J=1,15
         G1=0.D0
         G2=0.D0
         E1=0.D0
         E2=0.D0
         E3=0.D0
         E4=0.D0
         IF (ITEQ .EQ. 1) THEN
            E4=1.D0
            E3=0.D0
         ELSE IF (ITEQ .EQ. 2) THEN
            E1=1.D0
            E2=0.D0
         END IF
IF (J6 .EQ. 1) BETA=1.DO-ALPHA
         DO 50 I=1,N
            E5=(BETA*K2(I)+1.DO-BETA-ALPHA+ALPHA*K1(I))**2
           IF (E5 .EQ. 0.D0) GO TO 50
IF (ITEQ .EQ. 2) GO TO 40
G1=G1+Z(I)*(1.D0-K1(I))/DSQRT(E5)
            E1=E1+Z(I)*(1.D0-K1(I))**2/E5
            IF (ITEQ .NE. 3) GO TO 40
E2=E2+Z(I)*(1.DO-K1(I))*(1.DO-K2(I))/E5
            IF (ITEQ .EQ. 1) GO TO 50
G2=G2+Z(I)*(1.DO-K2(I))/DSQRT(E5)
   40
            E4=E4+Z(I)*(1.D0-K2(I))**2/E5
            IF (ITEQ .NE. 3) GO TO 50
E3=E2
   50
         CONTINUE
         IF (DABS(G1)+DABS(G2) .LT. 1.D-7) GO TO 90
         D=E1*E4-E2*E3
         IF (DABS(D) .EQ. 0.D0) GO TO 90
IF (ITEQ .EQ. 2) GO TO 60
         AO=ALPHA+(G2*E2-G1*E4)/D
         IF (ITEQ .EQ. 1) GO TO 70
BO=BETA+(G1*E3-G2*E1)/D
   60
   70
         ALPHA=AO
```

```
BETA=BO
        IF (DABS(D) .LT. 1.D-9) GO TO 90
   80 CONTINUE
* If more than one phase indicator is outside the interval [0,1], the
 phase with greater tendency to disappear is found
   90 IF (ITEQ .NE. 3) GO TO 130
      IF (ALPHA .LT. O.DO .AND. (BETA .LT. O.DO .OR. BETA .GT.
          1.DO)) THEN
     +
        J4=J4-1
        J3=J3+1
      END IF
      ONMAB=1.DO-ALPHA-BETA
      IF (ALPHA .LT. O.DO .AND. (ONMAB .LT. O.DO .OR. ONMAB .GT.
          1.DO)) THEN
     +
        J5=J5-1
        J3=J3+1
      END IF
      IF (ALPHA .LE. 1.DO .OR. BETA .GE. O.DO .OR. ONMAB .GE.
          0.DO) GO TO 110
      S7=0.D0
      S8=0.D0
      DO 100 I=1,N
        S7=S7+X2(I)*FUGC2(I)
        S8=S8+X3(I)*FUGC3(I)
  100 CONTINUE
      IF (S7 .GT. S8) THEN
        J5=J5-1
      ELSE
        J4=J4-1
      END IF
* Limitation of the calculated values of V/F and/or L1/F when
* outside the interval [0,1] to avoid calculation of negative
* mole fractions
  110 IF (ITEQ .NE. 3) GO TO 130
      IF (ALPHA .GT. 1.DO .OR. ALPHA .LT. O.DO) CALL LIMAB(N,
     + K1, ALPHA)
      B5=0.D0
      DO 120 I=1,N
        IF (K2(I) .EQ. 1.DO) GO TO 125
        B3=(ALPHA*(1.DO-K1(I))-1.DO)/(K2(I)-1.DO)
        IF (BETA .GT. 0.DO .AND. B3 .GT. 0.DO .AND. BETA .GT.
B3 .AND. (B3 .LT. B5 .OR. B5 .EQ. 0.DO)) B5=B3
     +
        IF (BETA .LT. O.DO .AND. B3 .LT. O.DO .AND. BETA .LT.
             B3 .AND. (B3 .GT. B5 .OR. B5 .EQ. 0.D0)) B5=B3
  120 CONTINUE
  125 IF (B5 .EQ. 0.D0) GO TO 160
IF (BETA .GT. 0.D0 .AND. BETA .GT. B5 .AND. B5 .GE. 1.D0)
       BETA=1.D0+(B5-1.D0)/2.D0
      IF (BETA .GT. 0.DO .AND. BETA .GT. B5 .AND. B5 .LT. 1.DO)
        BETA=B5-0.005D0
      IF (BETA .LT. 0.DO .AND. BETA .LT. B5) BETA=B5/2.DO
      GO TO 160
  130 IF (ITEQ .EQ. 2) GO TO 140
IF (ALPHA .GT. 0.DO .AND. ALPHA .LT. 1.DO) THEN
        J3=0
        GO TO 140
      ELSE
        J3=J3+1
      END IF
```

```
CALL LIMAB(N,K1,ALPHA)
 140 IF (ITEQ .EQ. 1) GO TO 170
IF (BETA .GT. 0.DO .AND. BETA .LT. 1.DO) THEN
        J4=0
        GO TO 150
      ELSE
        J4=J4+1
      END IF
      CALL LIMAB(N,K2,BETA)
 150 IF (ITEQ .EQ. 1) GO TO 170
      ONMAB=1.DO-ALPHA-BETA
      IF (ONMAB .LE. O.DO .OR. ONMAB .GE. 1.DO) THEN
        J5=J5+1
      ELSE
        J5=0
      END IF
      GO TO 170
 160 IF (ALPHA .LE. O.DO .OR. ALPHA .GE. 1.DO) THEN
        J3=J3+1
      ELSE
        J3=0
      END IF
      IF (BETA .LE. O.DO .OR. BETA .GE. 1.DO) THEN
        J4=J4+1
      ELSE
        J4=0
      END IF
      ONMAB=1.DO-ALPHA-BETA
      IF (ONMAB .LE. O.DO .OR. ONMAB .GE. 1.DO) THEN
        J5=J5+1
      ELSE
        J5=0
      END IF
* If the value of V/F, L1/F or L2/F has been outside the interval
* [0,1] for five consecutive iterations, the corresponding phase
* is eliminated
  170 IF (ALPHA .LE. O.DO .AND. J3 .GE. 5 .AND. ITEQ .EQ. 1)
     + GO TO 390
      IF (.NOT. (ALPHA .LE. O.DO .AND. J3 .GE. 5 .AND. ITEQ .NE.
          1)) GO TO 190
     +
* Elimination of the vapour phase
      ALPHA=0
      J3=0
      J4=0
      J5=0
      ITEQ=2
      L2=1
      IFLG(13)=0
      IC1=ITNUM-1
      ₩1=0.D0
      IF (BETA .GT. 0.DO .AND. BETA .LT. 1.DO) GD TO 240
     DO 180 I=1,N
        X3(I)=0.D0
        X2(I)=0.D0
  180 CONTINUE
      X2(I1)=0.98D0
      X2(I2)=0.02D0
      X3(I2)=0.98D0
```

```
X3(I1)=0.02D0
      BETA=0.5D0
      GD TD 300
* Elimination of liquid 1
  190 IF (J4 .GE. 5 .AND. BETA .LE. O.DO) THEN
        BETA=0.DO
        ITEQ=1
        J3=0
        J4=0
        J5=0
        IFLG(13)=0
        GO TO 220
      END IF
* Elimination of liquid 2
IF (J5 .GE. 5 .AND. 1.DO-ALPHA-BETA .LE. 0.DO) THEN
        J6=1
        BETA=1.DO-ALPHA
        ITEQ=1
        J3=0
        J4=0
        J5=0
        IFLG(13)=0
        GO TO 200
      END IF
* Feed composition in the one phase region. The calculation ends
      IF (J3 .GE. 5 .AND. ALPHA .GE. 1.DO) THEN
        ALPHA=1.DO
        GO TO 390
      ELSE IF (J4 .GE. 5 .AND. BETA .GE. 1.DO) THEN
        BETA=1.DO
        GO TO 390
      ELSE IF (J5 .GE. 5 .AND. 1.DO-ALPHA-BETA .GE. 1.DO) THEN
        GO TO 390
      END IF
  GD TD 240
200 DD 210 I=1,N
        X3(I)=X2(I)
        K1(I) = K1(I) / K2(I)
        K3(I)=K3(I)/K4(I)
        X6(I)=X5(I)
  210 CONTINUE
  220 IC1=ITNUM-1
      L2=1
      W1=0
      IF (ALPHA .GT. 1.DO) ALPHA=0.98 DO DO 230 I=1,N
        K2(I)=1.D0
  230 CONTINUE
* Calculation of the new compositions in the 3 phases
  240 S0=0.D0
      S1=0.D0
      S2=0.D0
      DO 260 I=1,N
        X3(I)=Z(I)/(BETA*K2(I)+1.DO-ALPHA-BETA+ALPHA*K1(I))
        IF (X3(I) .LT. 0.D0) X3(I)=0.D0
        S2=S2+X3(I)
```

```
IF (ITEQ .EQ. 1) GO TO 250
         X2(I) = X3(I) * K2(I)
         S1=S1+X2(I)
         IF (ITEQ .EQ. 2) GO TO 260
X1(I)=X3(I)*K1(I)
  250
         S0=S0+X1(I)
  260 CONTINUE
      DO 280 I=1,N
         X3(I) = X3(I)/S2
         IF (ITEQ .EQ. 1) GO TO 270
         X2(I)=X2(I)/S1
         IF (ITEQ .EQ. 2) GO TO 280
X1(I)=X1(I)/S0
  270
  280 CONTINUE
* Starting in the third iteration Wegstein's method is used (optional)
       IF (IFLG(7) .EQ. 0 .OR. ITNUM .LT. 3) GO TO 300
       ONMAB=1.DO-ALPHA-BETA
       IF (BETA .LT. O.DO .OR. BETA .GT. 1.DO .OR. ALPHA .LT.
      O.DO .OR. ALPHA .GT. 1.DO .OR. ONMAB .LT. O.DO .OR.
ONMAB .GT. 1.DO) GO TO 300
IF (ITNUM .LT. IC1+IWEG) GO TO 300
      +
      +
       IF (W1 .LE. -0.2D0) GO TO 290
       IF (OBJ .GE. OBJL) GO TO 300
IF (OBJ .GT. 0.8D0) GO TO 300
  290 CALL WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X3, X6)
IF (ITEQ .NE. 1) CALL WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X2, X5)
       IF (ITEQ .NE. 2) CALL WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X1, X4)
  300 IF (ITEQ .NE. 3) GO TO 320
* Calculation of the new values of K's
      IFLG(1)=0
IFLG(2)=0
IFLG(3)=0
       IFLG(16)=0
       CALL K(XO,X1,TC,PC,W,PP,P,T,N,1,3,APUR,BPUR,FUGC1,FUGC4,
               Z1,Z0,KIJ,KEQ,IFLG)
      +
      CALL K(X0,X2,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUGC2,FUGC4,
Z2,Z0,KIJ,KEQ,IFLG)
      +
       IF (IFLG(2) .EQ. 1) THEN
         IFLG(2)=0
         IFLG(16)=1
      END IF
       CALL K(XO,X3,TC,PC,W,PP,P,T,N,2,3,APUR,BPUR,FUGC3,FUGC4,
               Z3,Z0,KIJ,KEQ,IFLG)
      +
       IF (IFLG(2) .EQ. 1) THEN
         IFLG(2)=0
         IFLG(3)=1
       END IF
       IF (IFLG(16) .EQ. 1) IFLG(2)=1
  305 DO 310 I=1,N
         K1(I)=FUGC3(I)/FUGC1(I)
         K2(I)=FUGC3(I)/FUGC2(I)
  310 CONTINUE
       GO TO 340
  320 IF (ITEQ .EQ. 2) GO TO 330
CALL K(X3,X1,TC,PC,W,PP,P,T,N,ITEQ,3,APUR,BPUR,FUGC1,
               FUGC3,Z1,Z3,KIJ,K1,IFLG)
      +
       GO TO 340
  330 IF (ITEQ .EQ. 1) GO TO 340
       CALL K(X3,X2,TC,PC,W,PP,P,T,N,ITEQ,3,APUR,BPUR,FUGC2,
```

```
FUGC3,Z2,Z3,KIJ,K2,IFLG)
  340 IF (ITNUM .EQ. 0) GO TO 20
* Calculation of the objective function
      OBJL=OBJ
      OBJ=0.DO
      S5=0.D0
      S6=0.D0
      DO 360 I=1,N
        IF (ITEQ .EQ. 2) GO TO 350
        S5=S5+DABS(FUGC1(I)*X1(I)-FUGC3(I)*X3(I))
        IF (ITEQ .EQ. 1) GO TO 360
        S6=S6+DABS(FUGC2(I)*X2(I)-FUGC3(I)*X3(I))
  350
  360 CONTINUE
        OBJ=S5+S6
* Convergence test
      IF (OBJ .LT. EPS) GO TO 380
* Calculation of the damping factor for Wegstein's method (optional)
      IF (IFLG(7) .EQ. 0 .OR. IFLG(8) .EQ. 0) GO TO 20
      IF (ITNUM .LT. 3) GO TO 20
      ONMAB=1.DO-ALPHA-BETA
      IF (BETA .LT. O.DO .OR. BETA .GT. 1.DO .OR. ALPHA .LT.
          O.DO .OR. ALPHA .GT. 1.DO .OR. ONMAB .LT. O.DO .OR.
          ONMAB .GT. 1.DO) GO TO 20
      IF (L2 .EQ. 1) GO TO 20
      CALL AMTO(OBJ,OBJL,L2,W1,IC2,IC3,IC4,IC5)
      IF (IC5 .EQ. 0) GO TO 20
      IC5=0
      IC1=IC1-IWEG
      DO 370 I=1,N
        K1(I) = K3(I)
        IF (ITEQ .NE. 1) K_2(I)=K_4(I)
  370 CONTINUE
      GO TO 240
  380 IF (ITEQ .EQ. 3 .AND. (BETA .LE. 0.DO .OR. 1.DO-ALPHA-BETA
+ .LE. 0.DO)) GO TO 20
* Convergence is not allowed if Wegstein's method has been applied
      IF (IC1 .EQ. ITNUM) GO TO 20
* Convergence: test for the existence of 3 phases
  390 IF (ALPHA .GE. 1.DO) THEN
        ALPHA=1.DO
        BETA=0.DO
      ELSE IF (ALPHA .LE. O.DO) THEN
        ALPHA=0.DO
      END IF
IF (BETA .GE. 1.DO) THEN
        BETA=1.DO
        ALPHA=0.DO
      ELSE IF (BETA .LE. O.DO) THEN
        BETA=0.DO
      END IF
      ONMAB=1.DO-ALPHA-BETA
      IF (ONMAB .LE. O.DO) BETA=1.DO-ALPHA
      IF (BETA .LE. O.DO .OR. BETA .GE. 1.DO .OR. ALPHA .LE.
```

```
O.DO .OR. ALPHA .GE. 1.DO .OR. ONMAB .LE. O.DO .OR.
ONMAB .GE. 1.DO) IFLG(12)=0
      +
      +
       DO 400 I=1,N
         IF (ALPHA .EQ. 1.DO) THEN
X1(I)=Z(I)
           X2(I)=0.D0
           X3(I)=0.D0
         ELSE IF (BETA .EQ. 1.DO .OR. ONMAB .EQ. 1.DO) THEN
           X1(I)=0.D0
           X2(I)=0.D0
           X3(I)=Z(I)
         END IF
                                    2.1.2
         IF (ALPHA .EQ. 0.D0) X1(I)=0.D0
IF (BETA .EQ. 0.D0 .OR. ONMAB .EQ. 0.D0) X2(I)=0.D0
  400 CONTINUE
  410 IFLG(13)=0
       ZED(1)=Z1
       ZED(2)=Z2
       ZED(3)=Z3
       RETURN
       END
* SUBROUTINE FOR LIMITING V/F OR L1/F AS IN A 2 PHASE CALCULATION
       SUBROUTINE LIMAB(N,RK3,C)
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION RK3(20)
       RK7=RK3(1)
       DO 20 I=2,N
         IF (C .LT. 0.DO) GO TO 10
IF (RK3(I) .LT. RK7) RK7=RK3(I)
         GO TO 20
   10 IF (RK3(I) .GT. RK7) RK7=RK3(I)
   20 CONTINUE
       C3=1.D0/(1.D0-RK7)
       IF (C .LT. O.DO .AND. C3 .GT. O.DO) THEN
         C=-C3
         GO TO 30
       ELSE IF (C .GT. O.DO .AND. C3 .LT. O.DO) THEN
         C=1.D0-C3
         GO TO 30
       END IF
       IF (C .LT. 0.D0 .AND. C .LT. C3) C=C3/2.D0
IF (C .GT. 1.D0 .AND. C .GT. C3) C=(1.D0+C3)/2.D0
   30 RETURN
       END
```

```
*
         SUBROUTINE IFVLL (INITIALIZATION FOR THE VLL FLASH)
                                                                       *
*
                                                                       *
***********
                                                                      ***
* Purpose:
*
 Obtain the key components for the liquid phases and/or provide
 composition estimates for the vapour and both liquids as well as estimates for the phase fractions V/F and L1/F to initialize the
*
*
 VLL flash calculation
* Parameters:
*
 -Input
  IO, I1, I2, P, T, N, Z, TC, PC, W, PP, KIJ as defined in program PHASEQ
* -Output
  ALPĤA
*
           Estimated value of the ratio V/F
*
  BETA
                            44
                                ...
                                         L1/F
  11
           Key component for liq. 1
*
  12
*
  Y
           Mole fraction estimates for the vapour
                                                                       *
                                     ..
*
  X1
                                      liq. 1
             ..
                   ..
                             н
                                     11
                                             2
*
  X2
*
*
 Main variables:
*
  S1
           Sum of the mole fractions for the vapour
*
  PVAP
           Ratio of the vapour pressure to the total pressure
 Subroutines called:
  IFLL
* Called by:
  PHASEQ
*
*********
      SUBROUTINE IFVLL(IO, I1, I2, BETA, ALPHA, P, T, N, Y, X1, X2, Z,
                       TC, PC, W, PP, KIJ, IFLG)
     +
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KEQ(20), KIJ(20,20)
     DIMENSION IFLG(0:20),PC(20),PP(20),TC(20),W(20),X1(20),

+ X2(20),Y(20),Z(20),C1(20)

IF (I1 .NE. 0 .AND. I2 .NE. 0) GO TO 10
      IFLG(14)=1
* Assignment of key components for both liquids when not known
      CALL IFLL(IO, I1, I2, BETA, P, T, N, X2, X1, Z, TC, PC, W, PP, KIJ, KEQ,
     +
                IFLG)
      GO TO 30
 Allocation of mole fractions for both liquids with the key component
 criterion
   10 DO 20 I=1,N
        X1(I)=0.D0
        X2(I)=0.D0
  20 CONTINUE
      X1(I1)=0.98D0
      X2(I1)=0.02D0
      X1(I2)=0.02D0
```

```
X2(I2)=0.98D0
    30 BETA=0.5D0
* Calculation of the vapour initial composition
        SUM1=0.D0
        DO 40 I=1,N
PVAP=PC(I)*DEXP(5.3727DO*(1.DO+W(I))*(1.DO-TC(I)/T))/P
C1(I)=Z(I)*PVAP
           SUM1=SUH1+C1(I)
    40 CONTINUE
DO 50 I=1,N
           Y(I)=C1(I)/SUM1
    50 CONTINUE
* Calculation of the initial values of alpha and beta
        ALPHA=(Z(I1)-BETA*X1(I1)-X2(I1)*(1.DO-BETA))/
(Y(I1)-BETA*X1(I1)-X2(I1)*(1.DO-BETA))
       +
        BETA=BETA*(1.DO-ALPHA)
        IF (ALPHA .LT. 0.05D0) ALPHA=0.05D0
IF (ALPHA .GT. 0.95D0) ALPHA=0.95D0
IF (BETA .LT. 0.05D0) BETA=0.95D0
IF (BETA .GT. 0.95D0) BETA=0.95D0
        RETURN
        END
```

. `

```
*********
    SUBROUTINE WEG (WEGSTEIN'S METHOD TO OBTAIN NEW COMPOSITIONS)
                                                                 *
                                                                 *
*
****
                                                                 *
*
* Purpose:
 Calculate new compositions for a single phase using Wegstein's
                                                                 *
* acceleration method
* Parameters:
*
 -Input
*
 IC1
          Iteration number in which Wegstein's method was last used *
                     " " the flash calculation
*
  ITNUM
          Damping or acceleration factor
*
  W1
          Value of the objective function in the flash
*
  OBJ
  OBJL
                                         " " previous iteration *
*
*
  N
          Number of components
          Number of components
Mole fractions for a given phase
""""" " " in the previous iteration *
*
  X1
*
  X2
*
 -Output
*
          New value of the acceleration factor (W1>0). If it's
*
  W1
          input value is for damping (-0.6<=W1<=-0.2) it is not
*
          modified
*
*
  X1
          Mole fractions calculated with Wegstein's method
                                                                 *
*
*
                                                                 *
 Subroutines called:
*
  none
                                                                 *
* Called by:
                                                                 *
 FLASH2
*
                                                                 *
  FLASH3
*
                                                                 *
SUBROUTINE WEG(ITNUM, IC1, W1, OBJ, OBJL, N, X1, X2)
IMPLICIT REAL*8 (A-H, O-Z)
     DIMENSION X1(20), X2(20)
     IC1=ITNUM
     IF (W1 .GE. 0.DO) W1=OBJ/(OBJL-OBJ)
* Calculation of the new composition
     SUM=0.DO
     DO 10 I=1,N
       X1(I)=X1(I)+W1*(X1(I)-X2(I))
       IF (X1(I) .LT. 0.D0) X1(I)=0.D0
       SUM=SUM+X1(I)
  10 CONTINUE
     DO 20 I=1,N
       \overline{X1(I)} = \overline{X1(I)} / SUM
   20 CONTINUE
     RETURN
     END
```

```
*************
                                                                       *
*
             SUBROUTINE AMTO (DAMPING FACTOR CALCULATION)
*
                                                                       *
                                                                       *
*
***********
                                                                      **
*
 Purpose:
  Obtain the damping factor used in Wegstein's method
*
* Parameters:
*
 -Input
 OBJ,OBJL as described in subroutine WEG
*
  IC2, IC3, IC4, IC5 are control variables
*
 -Output
*
*
  L2
           Control variable: 0=no damping in the present iteration;
*
           1=damping in the present iteration
           Damping factor when the objective function increases significantly with respect to the last iteration
  ₩1
*
*
*
 Subroutines called:
  none
* Called by:
  FLASH2
  FLASH3
*
******************
      SUBROUTINE AMTO(OBJ,OBJL,L2,W1,IC2,IC3,IC4,IC5)
      IMPLICIT REAL*8 (A-H, O-Z)
      W2=(OBJL-OBJ)/OBJ
      IF (W2 .LE. -0.2D0) W2=-0.6D0
IF (W2 .LE. -0.2D0) W1=W2
IF (W2 .LE. -0.2D0 .AND. IC4 .EQ. 0) THEN
        IC4=1
      ELSE
        GO TO 10
      END IF
      IC2=7-IC3
      IC3=IC3+2
      IF (IC2 .LT. 1) IC2=1
   10 IF (W2 .GT. -0.2D0 .AND. OBJ .GT. IC2*0.001D0) GD TO 20
IF (W1 .LT. -0.2D0 .AND. OBJ .GT. IC2*0.001D0) THEN
        L2=L2+1
        IC5=1
      ELSE
        W1=0.D0
        IC4=0
      END IF
   20 RETURN
      END
```

```
*********
         SUBROUTINE GIBBS (GIBBS ENERGY OF MIXING CALCULATION)
* Purpose:
 Calculate the Gibbs energy of mixing for a system of n components *
* with a global composition z at a given temperature and pressure;
 the system may be VLL, LL, VL, L or V
*
* Parameters:
 -Input
  ITEQ, FUGC1, FUGC2, FUGC3 as defined in program PHASEQ
ALPHA, BETA, N, P, T, Z, TC, PC, W, PP, KIJ as defined in program PHASPLIT
          Mole fractions for the vapour (VL or VLL calc.) or for
          liq. 1 (LL calc.)
  X
          Mole fractions for the liquid (VL calc.) or for liq. 2
          (LL calc.)
          Mole fractions for liq. 1 (VLL calc.)
*
  X1
*
  X2
 -Output
*
          Gibbs energy of mixing (cal/gmole)
*
  DG
  Y
          Mole fractions for the vapour
*
                "
                          ...
                              liq. 1
*
  X1
            ...
                   н
                          н
  X2
                                     (the only liq. in VL calc.)
*
 Main variables:
          Reference pressure (1 atm)
Universal gas constant (1.987 cal/gmole K)
*
  PR
  R
          Gibbs energy of mixing for the vapour (cal/gmole)
  S1
*
*
  S2
                     -
                   ...
                            ...
                                 н
            ..
  S3
*
*
*
 Subroutines called:
*
  K
*
 Called by:
  PHASEO
*********
     SUBROUTINE GIBBS(ITEQ, N, P, T, ALPHA, BETA, Y, X, X1, X2, Z, TC, PC,
                      W, PP, KIJ, FUGC1, FUGC2, FUGC3, DG, IFLG)
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL*8 KEQ(20), KIJ(20,20)
     DIMENSION FUGC1(20), FUGC2(20), FUGC3(20), FUGC4(20), PC(20),
               PP(20),TC(20),W(20),X(20),X0(20),X1(20),X2(20)
     +
    +
               Y(20), Z(20), IFLG(0:20), APUR(20), BPUR(20), RM1(20),
     +
               RM2(20), RM3(20)
* Assignment of the reference pressure and the phase indicators
     PR=1.DO
     R=1.987D0
     IF (BETA .EQ. 1.DO .AND. ITEQ .EQ. 2) BETA=0.DO
     IF (ITEQ .EQ. 3 .AND. 1.DO-ALPHA-BETA .EQ. 0.DO .AND.
         BETA .NE. O.DO) BETA=O.DO
      IF (ITEQ .EQ. 1) THEN
        ALPHA=BETA
       BETA=0.DO
```

```
END IF
      IF (ITEQ .EQ. 2) ALPHA=0.DO
* Calculation of the fugacity coefficients when only one liquid phase
* is present
      IFLG(13)=1
      IF (.NOT. (ALPHA .EQ. O.DO .AND. (BETA .EQ. O.DO .OR.
          BETA .EQ. 1.DO))) GO TO 20
     +
      CALL K(X0,Z,TC,PC,W,PP,P,T,N,2,1,APUR,BPUR,FUGC3,FUGC4,
     +
              Z3,Z0,KIJ,KEQ,IFLG)
      DO 10 I=1,N
        Y(I)=0.D0
        X1(I)=0.D0
        X_2(I) = Z(I)
   10 CONTINUE
      GO TO 60
* Calculation of the fugacity coefficients when only a vapour phase
* is present
   20 IF (ALPHA .EQ. 1.DO .AND. BETA .EQ. 0.DO) THEN
CALL K(X0,Z,TC,PC,W,PP,P,T,N,1,1,APUR,BPUR,FUGC1,FUGC4,
                Z1,ZÓ,KÍJ,KEQ,IFLG)
      ELSE
        GO TO 40
      END IF
      DO 30 I=1,N
        Y(I)=Z(I)
        X1(I)=0.D0
        X2(I)=0.D0
        FUGC3(I)=FUGC2(I)
   30 CONTINUE
      GO TO 60
* Assignment of the variables according to the printing format
   40 DO 50 I=1.N
        IF (ITEQ .EQ. 1) THEN
          X_2(I) = X(I)
          FUGC3(I)=FUGC2(I)
          X1(I)=0.D0
          FUGC2(I)=0.D0
        ELSE IF (ITEQ .EQ. 2) THEN X1(I)=Y(I)
          X2(I)=X(I)
          FUGC3(I)=FUGC2(I)
          FUGC2(I)=FUGC1(I)
          Y(I)=0.D0
          FUGC1(I)=0.DO
        END IF
   50 CONTINUE
* Calculation of the Gibbs energy for each phase
   60 S1=0.D0
      S2=0.D0
      S3=0.D0
      ONMAB=1.DO-ALPHA-BETA
      DO 90 I=1,N
        IF (Z(I) .EQ. 0.DO) GO TO 90
IF (ONMAB .NE. 0.DO) THEN
          RM3(I)=R*T*DLOG(FUGC3(I)*X2(I)*P/PR)
        ELSE
```

```
GO TO 70
     END IF
S3=S3+X2(I)*RM3(I)
70
     IF (BETA .NE. O.DO) THEN
       RM2(I)=R*T*DLOG(FUGC2(I)*X1(I)*P/PR)
     ELSE
       GO TO 80
     END IF
S2=S2+X1(I)*RM2(I)
80
     IF (ALPHA .NE. O.DO) THEN
       RM1(I)=R*T*DLOG(FUGC1(I)*Y(I)*P/PR)
     ELSE
       GO TO 90
     END IF
     S1=S1+Y(I)*RM1(I)
90 CONTINUE
```

\* Calculation of the system's total Gibbs energy (cal/gmole)

-

```
DG=ALPHA*S1+BETA*S2+ONMAB*S3
IFLG(13)=0
RETURN
END
```

## Appendix C. Description and listings of the computer programs

```
*****************
              SUBROUTINE K (EQUILIBRIUM CONSTANTS CALCULATION)
**********
* Purpose:
 Calculate the fugacity coefficients for all components in both
* phases for VL and LL equilibria as well as the corresponding
* equilibrium constants. For VLL equilibria only the fugacity
* coefficients are calculated for a given phase each time the
 subroutine is called. The Soave-Redlich-Kwong or the Peng-Robinson equations of state are used to evaluate the thermodynamic
*
 properties
*
 Parameters:
  -Input
   TC,PC,W,PP,P,T,N,KIJ,IFLG as defined in program PHASPLIT
   X
            Mole fractions vector; for VL equilibria corresponds to
            the liquid comp., for LL to liquid 2 and for VLL is a
            dummy variable
   Y
            Mole fractions vector; for VL equilibria corresponds to the vapour comp., for LL to liquid 1 and for VLL is any
            of the phases as indicated by variable IT
            Indicator for the type of phase; in VL equilibria IT=1 means the phase with composition Y is a vapour; in LL equilibria IT=2 and means phase with composition Y is a
   IT
            liquid; in VLL equilibria IE=1 indicates the phase is a
            vapour and IT=2 that it is a liquid
   IE
            Control variable which indicates a flash calculation in
            which neither the temperature or the pressure change with
            respect to the last time the routine was executed. In
            such case parameters A and B for the pure components need
            not be calculated again
            Vector of A parameters for the pure components evaluated at the specified T and P % \left( {{{\bf{P}}_{{\rm{s}}}} \right)
   APUR
   BPUR
            Vector of B parameters for
            at the specified T and P
*
  -Output
   APUR, BPUR as defined above
   FUGC1
          Vector of the calculated fugacity coefficients; for VL
           equilibria corresponds to the vapour; for LL to liquid 1
           and for VLL to the specified phase with composition Y
   FUGC2
          Vector of the calculated fugacity coefficients; for VL
           equilibria corresponds to the liquid; for LL to liquid 2
           and for VLL is a dummy variable
   KEQ
           Vector of equilibrium constants for VL or LL equilibria
 Main variables
  L1,L2,U and W are equation of state parameters
                                                                             *
 Subroutines called:
*
  PAB
  AB
*
   ZETA
   FUG
*
 Called by:
  PHASPLIT
   INSHAH
   INGASE
```

291

```
INLIQ
                                                                               ¥
*
                                                                               *
  TESTA
*
  NELSTA
                                                                               *
*
*
   VAPSER
                                                                               *
*
  LIQSER
                                                                               *
*
   PHASEQ
                                                                               *
*
   IFLL
                                                                               *
   IFVL
*
                                                                               *
*
  FLASH2
                                                                               *
*
  FLASH3
                                                                               *
*
   GIBBS
                                                                               *
                                                                               *
******************
      SUBROUTINE K(X,Y,TC,PC,W,PP,P,T,N,IT,IE,APUR,BPUR,FUGC1,
FUGC2,Z1,Z2,KIJ,KEQ,IFLG)
     +
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KEQ(20), KIJ(20,20)
      DIMENSION FUGC1(20), FUGC2(20), IFLG(0:20), PC(20), PC(20),
TC(20), W(20), X(20), Y(20), APUR(20), BPUR(20)
      IF (IFLG(0) .EQ. 0) THEN
         U=1.D0
         V=0.D0
      ELSE
         U=2.D0
         V=-1.DO
      END IF
      IF (IE .EQ. 3 .AND. IFLG(4) .EQ. 1) GO TO 10
   CALL PAB(P,T,N,TC,PC,W,PP,APUR,BPUR,IFLG)
10 IF (IFLG(13) .EQ. 0) THEN
         IFLG(1)=0
         IFLG(2)=0
         IFLG(3)=0
      ELSE
        GO TO 30
      END IF
      CALL AB(X, N, APUR, BPUR, KIJ, A2, B2)
      CALL ZETA(2,U,V,A2,B2,B9,Z2,IFLG)
IF (IFLG(0) .EQ. 0) THEN
RL2=DLOG((Z2+B2)/Z2)
      ELSE
        RL2=DLOG((Z2+B2*(1.D0+DSQRT(2.D0)))/(Z2+B2*(1.D0-
                    DSQRT(2.D0)))/2.D0/DSQRT(2.D0)
     +
      END IF
      DO 20 I=1,N
        CALL FUG(X,I,N,Z2,A2,B2,B9,APUR,BPUR,KIJ,RL2,FUGC2(I))
   20 CONTINUE
   30 CALL AB(Y, N, APUR, BPUR, KIJ, A1, B1)
      IF (IFLG(2) .EQ. 1 .AND. IFLG(13) .EQ. 0 .AND. IT .EQ. 2)
     + THEN
        IFLG(3)=1
         IFLG(2)=0
      END IF
      CALL ZETA(IT, U, V, A1, B1, B8, Z1, IFLG)
      IF (IFLG(0) .EQ. 0) THEN
        RL1=DLOG((Z1+B1)/Z1)
      ELSE
        RL1=DLOG((Z1+B1*(1.D0+DSQRT(2.D0)))/(Z1+B1*(1.D0-
     +
                    DSQRT(2.D0)))/2.D0/DSQRT(2.D0)
      END IF
      DO 40 I=1,N
         CALL FUG(Y,I,N,Z1,A1,B1,B8,APUR,BPUR,KIJ,RL1,FUGC1(I))
   40 CONTINUE
```

```
IF (IFLG(13) .EQ. 1) GO TO 60
DO 50 I=1,N
KEQ(I)=FUGC2(I)/FUGC1(I)
50 CONTINUE
60 RETURN
END
```

. •

.

```
SUBROUTINE PAB (PURE COMPONENT PARAMETER A AND B CALCULATION)
*
                                                               *
**********
* Purpose:
  Obtain the pure component's attraction and repulsion parameters
                                                               *
*
* (in dimensionless form) from either the SRK or PR equations of
* state
* Parameters:
* -Input
 P,T,N,TC,PC,W,PP as defined in program PHASPLIT
* -Output
*
 APUR, BPUR as defined in subroutine K
*
* Called by:
*
  K
                                                               *
SUBROUTINE PAB(P,T,N,TC,PC,W,PP,APUR,BPUR,IFLG)
IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION IFLG(0:20), PC(20), PP(20), TC(20), W(20), APUR(20),
BPUR(20), R(20), Q(20)
    +
     IF (IFLG(0) .EQ. 1) THEN
       C=0.37464D0
       D=1.54226D0
       E=-0.26992D0
       A=0.45723553D0
       B=0.077796074D0
     ELSE
       C=0.48D0
       D=1.574D0
      E=-0.176D0
       A=0.42748023D0
       B=0.08664035D0
     END IF
     DO 30 I=1,N
       R(I)=C+D*W(I)+E*W(I)**2
       IF (T .GT. TC(I)) GO TO 10
       Q(I)=(1.D0+R(I)*(1.D0-DSQRT(T/TC(I)))-PP(I)*(1.D0-
            T/TC(I))*(0.7D0-T/TC(I)))**2
    +
       GO TO 20
       D=1.D0+R(I)/2.D0+0.3D0*PP(I)
  10
       Q(I)=DEXP((D-1.D0)/D*(1.D0-(T/TC(I))**D))**2
APUR(I)=A*P/PC(I)*(TC(I)/T)**2*Q(I)
  20
       BPUR(I) = B*P/PC(I)*TC(I)/T
  30 CONTINUE
     IFLG(4)=1
     RETURN
     END
```

```
********************
*
      SUBROUTINE AB (MIXTURE'S A AND B PARAMETER CALCULATION)
                                                            *
                                                            *
*
*
*
* Purpose:
  Obtain the mixture's attraction and repulsion parameters (in
                                                            *
  dimensionless form) from either the SRK or PR equations of state
                                                            *
* Parameters:
*
 -Input
 N, APUR, BPUR, KIJ as defined in subroutine K
*
  X
         Vector of mole fractions for the specified phase
*
*
                                                             *
* -Output
                                                            *
         Attraction parameter A for the mixture
Repulsion "B"""
*
  A
                                                            *
  В
                                                            *
*
*
*
 Called by:
                                                             *
*
 K
                                                             *
*************
     SUBROUTINE AB(X,N,APUR,BPUR,KIJ,A,B)
IMPLICIT REAL*8 (A-H,O-Z)
     REAL*8 KIJ(20,20)
     DIMENSION X(20), APUR(20), BPUR(20)
     A=0.D0
     B=0.D0
     DO 20 I=1,N
      DO 10 J=1,N
        A=A+X(I)*X(J)*DSQRT(APUR(I)*APUR(J))*(1.DO-KIJ(I,J))
  10
      CONTINUE
      B=B+X(I)*BPUR(I)
  20 CONTINUE
     RETURN
     END
```

```
*******
        SUBROUTINE ZETA (COMPRESSIBILITY FACTOR CALCULATION)
                                                                   *
  **********
**
                                                                 * * *
*
 Purpose:
  Obtain the mixture's compressibility factor at the specified T, P
  and composition from the SRK or PR equations of state
*
 Parameters:
*
 -Input
  IT, A, B, IFLG as defined in subroutine K
*
  U,V
          Equation of state parameters
*
 -Output
*
  BO
          Calculated value of the mixture's B parameter to correct
          the fugacity coeffs. for the liquid when an extrapolated
*
*
          value of the compressibility factor is needed
          For the vapour or when an extrapolated compressibility factor is not required for the liquid, BO is equal to B
*
  Z
          Compressibility factor obtained for the phase indicated
*
          by variable IT
 Called by:
*
  K
     ******
**
     SUBROUTINE ZETA(IT, U, V, A, B, BO, Z, IFLG)
     IMPLICIT REAL*8 (A-H, O-Z)
     DIMENSION IFLG(0:20)
     IF (IFLG(0) .EQ. 0) THEN
Y0=3.8473221019D0
       A0=4.933962452D0
       A2=4.66044838388D0
     ELSE
       Y0=3.9513730356D0
       A0=5.877359948D0
       A2=5.54485105556D0
     END IF
     A1=A/B
     D0=1.D0/Y0
     B1=U*B-B-1.D0
     C1=A+V*B*B-U*B*B-U*B
     D1=-V*B**3-V*B*B-A*B
     P=3.D0*C1-B1**2
     Q=B1**3+(27.D0*D1-9.D0*B1*C1)/2.D0
     D2=P**3+Q*Q
      IF (D2 .LT. O.DO) THEN
       H=DACOS(-Q/DSQRT(-P**3))
       GO TO 10
     END IF
     ONT=1.D0/3.D0
     RI = -Q + DSQRT(D2)
     RM=DABS(RI)/RI*DABS(RI)**ONT
     RJ = -Q - DSQRT(D2)
     RN=DABS(RJ)/RJ*DABS(RJ)**ONT
      Z = (RM + RN - B1)/3.D0
   10 IF (IT .EQ. 1) GO TO 40
```

\* Calculation for the liquid

```
IF (D2 .GE. 0.D0) GO TO 20
Z=(-B1+2.D0*DSQRT(-P)*DCOS(H/3.D0+2.D0/3.D0*
          3.14159265359D0))/3.D0
      IF (Z .GT. B) GO TO 20
       Z=(-B1+2.D0*DSORT(-P)*DCOS(H/3.D0))/3.D0
   20 D=B/Z
      F=1.D0/(1.D0-D)**2-A1*(2.D0*D+U*D*D)/(1.D0+U*D+V*D*D)**2
IF (Z .GT. B/D0 .OR. F .LT. 0.1D0) THEN
         GO TO 70
       ELSE
         GO TO 110
       END IF
* Extrapolation for the liquid
   30 \text{ IFLG}(2)=1
      IF (D .EQ. DO) THEN
F=1.DO/(1.DO-D)**2-A1*(2.DO*D+U*D*D)/(1.DO+U*D+V*D*D)**2
       ELSE
         F=F+0.1D0
      END IF
       C2=F*(D-0.7D0*D0)
      F6=D/(1.D0-D)-A1*D*D/(1.D0+U*D+V*D*D)
      CO=F6-C2*DLOG(D-0.7DO*DO)
      D=DEXP((B-CO)/C2)+0.7DO*DO
      Z=B/D
      GO TO 110
* Calculation for the vapour
   40 IF (D2 .GE. 0.D0) GO TO 50
      Z=(-B1+2.D0*DSQRT(-P)*DCOS(H/3.D0))/3.D0
   50 IF (A1 .LT. A0) GO TO 110
      D=B/Z
      F=1.D0/(1.D0-D)**2-A1*(2.D0*D+U*D*D)/(1.D0+U*D+V*D*D)**2
IF (Z .LT. B/D0 .OR. F .LT. 0.1D0) THEN
         GO TO 70
      ELSE
         GO TO 110
      END IF
* Extrapolation for the vapour
   60 IFLG(1)=1
      F6=D/(1.D0-D)-A1*D*D/(1.D0+U*D+V*D*D)
      F5=F+0.1D0
      C2=(-F5*((D+D0)/2.D0-D)+F6)/(2.D0*F6**2*D*((D+D0)/2.D0-
     +
           D)+F6**2*(D*D-((D+D0)/2.D0)**2))
      C3=-(F5+C2*2.D0*F6**2*D)/F6**2
      C0=1.D0/F6-C3*D-C2*D*D
      D = (-C3 - DSQRT(C3 + 2 - 4.D0 + C2 + (C0 - 1.D0/B)))/(2.D0 + C2)
      Z=B/D
      GO TO 110
* Calculation of reduced density in the limit of the allowed region
   70 IF (A1 .LT. A2) THEN
         D=D0
         GO TO 100
      END IF
      IF (IT .EQ. 2) THEN
        D=0.8D0
```

```
ELSE
      D=0.1D0
    END IF
    K=0
 80 K=K+1
    IF (K.GT. 25 .AND. IT .EQ. 2) THEN
      D=D0
       GO TO 100
    ELSE IF (K .GT. 25) THEN
       D=0.7D0*D0
       GO TO 100
    END IF
    F=1.D0/(1.D0-D)**2-A1*(2.D0*D+U*D*D)/(1.D0+U*D+V*D*D)**2-
   + 0.1D0
    F1=2.D0/(1.D0-D)**3-(A1*(1.D0+U*D+V*D*D)*(2.D0+2.D0*U*D)-
   + 2.D0*A1*(2.D0*D+U*D*D)*(U+2.D0*V*D))/(1.D0+U*D+V*D*D)
+ **3
    D3=D-F/F1
    IF (DABS(1.DO-D/D3) .LT. 1.D-5) GO TO 90
    D=D3
    GO TO 80
 90 IF ((D.LT. 0.D0 .OR. D .GT. 1.D0) .AND. IT .EQ .2) D=D0
IF ((D.LT. 0.D0 .OR. D .GT. 1.D0) .AND. IT .EQ .1)
   + D=0.7D0+D0
100 IF (IT .EQ. 2) THEN
GO TO 30
    ELSE
       GO TO 60
    END IF
110 IF (IFLG(2) .EQ. 1 .AND. IT .EQ. 2) THEN
B0=D/(1.D0-D)-A1*D*D/(1.D0+U*D+V*D*D)
    ELSE
       BO=B
    END IF
RETURN
    END
```

```
******************
         SUBROUTINE FUG (FUGACITY COEFFICIENTS CALCULATION)
*
                                                           *
*
*********
                                                           **
* Purpose:
*
  Calculate the fugacity coefficient for component i in the
 mixture for a given phase
*
* Parameters:
*
 -Input
 N, APUR, BPUR, KIJ as defined in subroutine K
*
  Z,A,B,BO as defined in subroutine ZETA
*
         Vector of mole fractions for the specified phase
*
 X
  Ι
         Component number for which the fugacity coefficient is to *
*
         be obtained
*
  RL
*
         Equation of state parameter
*
 -Output
*
  F
         Fugacity coefficient for component i in the specified
  В
         phase
*
*
 Called by:
*
  K
**********
     SUBROUTINE FUG(X,I,N,Z,A,B,BO,APUR,BPUR,KIJ,RL,F)
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL*8 KIJ(20,20)
     DIMENSION X(20), APUR(20), BPUR(20)
     SUM=0.DO
     DO 10 J=1,N
      SUM=SUM+X(J)*DSQRT(APUR(I)*APUR(J))*(1.DO-KIJ(I,J))
  10 CONTINUE
    F=DEXP(BPUR(I)/B*(Z-1.D0)-DLOG(Z-B)+A/B*RL*(BPUR(I)/B-
    + 2.D0*SUM/A))*B0/B
     RETURN
     END
```