Application of a non-linear transformation to the surface fraction of the UNIQUAC model and the performance analysis of the subsequent model (FlexQUAC-Q)

By

Thishendren Naidoo

[B. Sc. (Eng.)]
University of Kwazulu-Natal Durban
For the Degree of Master of Science (Chemical Engineering)
October 2007

Abstract

G^E-model and equations of state are used to describe and predict phase equilibria. Current models have varying capabilities and some display selectivity for certain special mixtures. While many models are superior to others in their performance, all models share a common deficiency, the inability to simultaneously describe vapour-liquid (VLE) and liquid–liquid equilibria (LLE). Current models require separate parameters to describe the two equilibria. This formed the motivation for a non-linear transformation which was formulated by Rarey (2005).

The transformation was applied to the concentration space. The clear advantage of such a transformation was that it could be easily applied to any model. The flexibility of the model was drastically increased. The effects were investigated on the local composition models, in particular the UNIQUAC model resulting in the FlexQUAC model. The model was used to regress a host of VLE and LLE data sets contained in the Dortmund Data Bank (DDB). The transformation had the desired effect on the flexibility of the model and the model was now able to describe VLE and LLE.

However a symmetric transformation applied to the concentration space might not be effective in the description of systems exhibiting large difference in molecular size. This is a clear disadvantage of the proposed FlexQUAC model. In order to allow the model to cater to asymmetric systems, the transformation is now applied to the surface fraction of the residual contribution of the UNIQUAC model. The Guggenheim-Staverman expression in the combinatorial part was not transformed. Both the original combinatorial term and the more suitable modification of Weidlich and Gmehling (1987) were used. The newly formed model was called the FlexQUAC-Q model.

The development of the FlexQUAC-Q model, derivation of activity coefficient expressions, model implementation and its performance analysis form the basis for this research study.

The activity coefficient of the new model had to be re-derived due to the application of the transformation to the residual contribution of the UNIQUAC equation. The computation of the activity coefficient was programmed in FORTRAN and integrated into the regression tool (RECVAL) of the Dortmund Data Bank (DDB). The RECVAL tool was used to regress data sets

contained in the DDB. Results obtained were comparable to those obtained using the GEQUAC model.

The regression was also performed in EXCEL for the three models (UNIQUAC, FlexQUAC, FlexQUAC-Q). The regression in EXCEL was more rigorous and was used for the comparison of the objective functions and to obtain a set of unique model parameters for each data set. The performance of the FlexQUAC-Q model was assessed utilizing the same data sets used to analyse the performance of the FlexQUAC model. The model's performance was assessed in the regression of 4741 binary VLE data sets, 13 ternary VLE data sets and carefully select ternary LLE cases.

The minor mean relative reduction of about 3% of the objective function using FlexQUAC-Q compared to FlexQUAC was observed compared to a reduction by about 53% relative to the UNIQUAC-results.

It was necessary to illustrate that the new model does not degenerate the model's existing capabilities (e.g. ability to predict multi-component mixtures from binary data) and that the model performs as well as or superior to the UNIQUAC model. FlexQUAC-Q performed similarly to FlexQUAC. However the improvement in the qualitative description of data sets exhibiting asymmetry is apparent. Herein lies the justification of such a modification and this illustrates the preference of such a model when asymmetric systems are being considered.

In addition, the FLEXQUAC-Q model can be adapted to be implemented into a group contribution method, a distinct advantage over the previous model FlexQUAC. The equations for the application of a non-linear transformation to a functional group activity coefficient model, UNIFAC are also explored in this study. The resulting model is referred to as FlexFaC.

PREFACE

The research presented in this thesis was performed at the University of KwaZulu-Natal, Durban from January 2004 to October 2007 (Part time basis from July 2005 – October 2007). The research was supervised by Professor D. Ramjugernath, Dr J. Rarey and Professor J. D. Raal.

This thesis is presented as the full requirement for the degree M.Sc. in Chemical Engineering. All the work presented in this thesis is original, unless otherwise stated and has not (in whole or part) been submitted previously to any tertiary institute as part of a degree.

Thishendren Naidoo

St. No. 200266703

As the candidate's supervisor, I, Prof. D. Ramjugernath, have approved this thesis for submission.

Prof. D. Ramjugernath

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

Nomenclature

List of Symbols

A	Helmhotz free energy
Cp	heat capacity
f_i	fugacity of pure component i
$\hat{f}_i$	fugacity of component i in a mixture
f^o	standard state fugacity
G	Gibbs free energy
H	enthalpy
J	joule
n	number of moles
N_A	Avagadro's number
P	pressure
q	surface parameter in UNIQUAC equation
r	volume parameter in UNIQUAC equation
R	universal gas constant
S	entropy
T	temperature
U	internal energy
V	molar volume
x	liquid mole fraction
y	vapour mole fraction
Z	coordination number in UNIQUAC equation

Greek Letters

α	non-randomness parameter in NRTL equation
γ	Activity coefficient
δ	additional parameter in FlexQUAC/ FlexQUAC-Q equation
τ	parameter in UNIQUAC equation

θ	surface fraction in UNIQUAC equation
ϕ	fugacity coefficient
μ	chemical potential
λ	Boltzmann factor

Superscripts

E	excess property
id	ideal solution
ig	ideal gas
o	standard state
R	residual property
∞	used to represent activity coefficient at infinite dilution

Subscripts

comb	combinatorial part in UNIQUAC equation
i, j	component identification
ij	interaction between components i and j
res	residual part in UNIQUAC equation

Overbars

-	partial molar property
^	property of component in a mixture

Abbreviations

AZD	azeotropic data
DDB	Dortmund Data Bank
GC	group contribution
LLE	liquid liquid equilibrium
SLE	solid liquid equilibrium
VLE	vapour liquid equilibrium

Chapter 1 Introduction

Everyday, man relies on his ability to accurately predict information and thereby reduce the margin for error and uncertainty. Scientific knowledge has burgeoned over time and has further developed our fundamental understanding in a plethora of fields and subjects. This understanding coupled with mathematics has enabled us to describe, model and quantify many aspects of interest. This can range from predicting daily weather patterns to economic indicators. Hence the applicability of predictive models can invariably be seen in everyday life.

Predictive models have become indispensable in the chemical industry, especially in the case where equilibrium data is unavailable. Phase equilibrium is perhaps one of the most important subjects in physical chemistry. Despite this being a fairly old field of research, much needs to be learned to improve the theoretical aspects of phase equilibria. Classical thermodynamics provides a framework for quantifying phase equilibria via pressure, temperature and chemical potential. At equilibrium these properties are equal for each component in all phases. The challenge lies in relating this chemical potential or fugacity to measurable quantities i.e. temperature, pressure and composition. In the case of real gas mixtures, the deviation from ideal gas mixtures is encompassed in a fugacity coefficient which is modeled by an appropriate equation of state. A fugacity coefficient can also be defined for the deviation in the liquid phase and can be modeled by an equation of state, preferably at elevated pressures. While in the case of a liquid phase, the ratio of fugacity of a component at some pressure, temperature and composition, to a corresponding fugacity at the mixture temperature and some specified pressure and composition, defines the activity coefficient of that component. The activity coefficient is modeled by excess

Gibbs free energy and this property quantifies the departure from an ideal solution in the case of the liquid phase.

Industries that use separation processes promoted further research into mixture thermodynamics. The accurate description of a multicomponent mixture became crucial for the design, synthesis and optimization of chemical processes. Separation processes like distillation are energy intensive and the ability to model and predict the behaviour of the multicomponent system in question is of paramount importance to ensure optimal and profitable operability. As a consequence, many commercial process simulators and physical property data banks have been developed which have radically improved our ability to model and predict multicomponent systems. The increase in computational speed of computers has also proven beneficial to the prediction of data.

Despite all these efforts, only a few mixture models are broadly used to correlate binary systems and predict the real behavior of multicomponent liquid non-electrolyte phases (Wilson, NRTL and UNIQUAC). All these models were developed more than 25 years ago and all of them suffer from several shortcomings discussed in detail by Rarey (2005).

Rarey (2005) developed a mathematical procedure to improve the existing models and the results were discussed in detail with reference to the UNIQUAC equation. The modified equation (FlexQUAC) was much better able to correlate binary phase equilibrium data and even allowed a simultaneous description of vapour-liquid and liquid-liquid equilibria.

The test of the new model focused to a great part on vapour-liquid equilibria and most data used described systems with components of rather similar size. As already suggested in the previous paper, in the case of mixtures of large and small molecules, a transformation of the mole fraction scale might not be a good choice. In these cases, the transformation would mainly affect the activity coefficient of the large component.

This research study presents a non-linear flexibilisation of the surface fraction in the residual part of the UNIQUAC equation (FlexQUAC-Q). G^E, when plotted as a function of the surface fraction

is usually a symmetric function with the maximum near 0.5. The combinatorial part of the equation is not transformed.

The performance of FlexQUAC-Q is compared to UNIQUAC and FlexQUAC using a similar set of data as in the previous paper.

Chapter 2 Theoretical Aspects of Equilibria

2.1 Introduction

In the realm of thermodynamics, a theoretically correct description of a liquid mixture and its behaviour remains quite complex. A thermodynamicist essentially measures different types of data experimentally and uses mathematical expressions to model trends in data so as to establish a tangible link between measurable and immeasurable quantities. Mathematical expressions can employ different levels of theory ranging from fully empirical to complex theoretical. Experiments are usually setup to record phase equilibrium data or excess properties. Phase equilibrium data could refer, for example to vapour-liquid equilibrium (VLE) or liquid-liquid equilibrium (LLE) depending on the nature of the chemical mixture and the presiding conditions of the mixture i.e. temperature and pressure.

The FlexQUAC-Q model that was developed in this research study is an excess Gibbs energy model based on the existing UNIQUAC model. Prior to the development and derivation of this model, an in depth study of some introductory thermodynamics is necessary.

The concept of Gibbs energy is introduced here. The Gibbs energy state function is an important generating property in thermodynamics. Phase equilibrium is examined in this chapter and the equations that describe it are presented. Chemical potential and its relation to phase equilibrium is also explained to help the reader to establish the link between this property and phase

equilibrium. Both fugacity and activity coefficient relate chemical potential to measurable properties. Fugacity coefficients are usually used to quantify the departure from ideality for real gases via residual properties. The activity coefficient is preferred when defining the departure of real liquids from ideality. In this research study the liquid phase departure from ideality for experimental VLE and LLE data was computed from derived FlexQUAC-Q activity coefficient expressions.

The concepts of LLE are discussed briefly. As part of the performance assessment of the FlexQUAC-Q model, ternary LLE is calculated and compared to the experimental data. Hence, an understanding of the types of ternary data was necessary. These are illustrated and discussed. The ternary LLE calculations in this research study were performed using the K-factor method. The algorithm for this method is detailed here.

Some commentary on data quality and model selection is presented here to emphasise the importance of accurate measurement of data. Consistent data can be inaccurate and lead to inaccurate prediction of data. It is vital that the reader understand how the quality of experimental data impacts on the accuracy of predicting data.

Since the focus of this research study is the development of an excess Gibbs energy model, some liquid solution theory is discussed. Some of the commonly used excess Gibbs energy models e.g. UNIQUAC, were developed from liquid solution theory. In this chapter, lattice theory, Flory Huggins and two liquid theory are discussed to give a reader an indication of how liquid solution theory evolved.

An overview of the excess Gibbs energy models is detailed here. The reviewed models include Margules, Redlich Kister, Van Laar, Wilson, NRTL and UNIQUAC. The advantages, disadvantages and the range of application for each model are discussed. Before undertaking the development of the FlexQUAC-Q model, it was vital to first understand the ability of each of the existing models. Since the FlexQUAC-Q model has its roots in the UNIQUAC model, a more comprehensive discussion of the model and its derivation is presented.

Group contribution methods are discussed briefly. The ASOG, UNIFAC and GEQUAC methods are discussed. The FlexQUAC-Q model can be extended to group contribution methods. This is discussed further in *Chapter 5*. The newly formed group contribution method, FlexFAC should be able to describe both VLE and LLE simultaneously. The GEQUAC model is a fairly complex group contribution method that achieves impressive results. However, these results are comparable to that of FlexQUAC. While the GEQUAC model has a theoretical basis, both FlexQUAC and FlexQUAC-Q are formulated from a non-linear transformation applied to the UNIQUAC model. In *Chapter 6*, the results obtained from the regression from the FlexQUAC-Q model is used to make a comparison with the GEQUAC model.

Thermodynamics has been elucidated in great depth by several authors including Abbott and Van Ness (1996) and Perry and Green (1998). An extensive description and discussion of various experimental apparatus was for example given by Raal and Muhlbauer (1998). This chapter proceeds to give a brief overview of the fundamental treatment of equilibria and other basic concepts which needed to be understood to develop the FlexQUAC-Q model.

2.2 Gibbs Energy

Classical thermodynamics provides a number of state functions, the most well known being internal energy (U), enthalphy (H), Helmholtz energy (A) and Gibbs free energy (G). Gibbs free energy is quantified in J/mol. Changes in these functions depend on the changes in two, or in case of open systems with material exchange with the surrounding, three state variables. Commonly used state variables are temperature T, entropy (S), pressure (P), volume (V), chemical potential (μ) and mole number (n). If these two or three state variables are kept constant, the state function is also a potential with a minimum value at equilibrium. The Gibbs energy is a convenient state function since it is a potential function at constant temperature, pressure and number of moles in the system. The Gibbs energy is an important generating property since it provides a tangible link between equilibrium, mathematics and classical thermodynamics.

For a closed system in equilibrium, the change in the total system Gibbs energy is given by:

$$d(nG) = (nV)dP - (nS)dT \qquad \text{2-1}$$

$$\left[\frac{\partial(nG)}{\partial P} \right]_{T,n} = nV$$

$$\left[\frac{\partial(nG)}{\partial T} \right]_{P,n} = -nS$$

where

- V - molar volume
- S - molar entropy
- P – pressure
- T – temperature
- n - mole number

Thus at constant T and P (the typical case of practical interest) the change of G in equilibrium must be zero. **Equation 2-1** is applicable to a system without exchange of material with the surrounding. To evaluate the vapour-liquid equilibrium problem one has to consider the case of mutual exchange of material between the phases.

In this case the Gibbs energy function is also influenced by a change in the amount of material. Hence,

$$nG = g(P,T,n_1,n_2,...n_i)$$

where n_i refers to the mole number of the components

$$d(nG) = \left[\frac{\partial(nG)}{\partial P} \right]_{T,n} dP + \left[\frac{\partial(nG)}{\partial T} \right]_{P,n} dT + \sum_i \left[\frac{\partial(nG)}{\partial n_i} \right]_{P,T,n_{j \neq i}} dn_j$$

$$d(nG) = (nV)dP - (nS)dT + \sum_i \left[\frac{\partial(nG)}{\partial n_i} \right]_{P,T,n_{j\neq i}} dn_j \qquad \text{2-2}$$

The last term in the equation above is termed the chemical potential.

$$\mu_i = \left[\frac{\partial(nG)}{\partial n_i} \right]_{P,T,n_{j\neq i}} \qquad \text{2-3}$$

Thus the fundamental property relation is derived.

$$d(nG) = (nV)dP - (nS)dT + \sum_i \mu_i dn_i \qquad \text{2-4}$$

The Gibbs energy function plays an integral role in the computation of other system properties.

2.3 Phase Equilibria

Equilibrium refers to a state of 'no change'. In the context of vapour-liquid equilibrium, it refers to a case when 'two or more phases reach a state of equilibrium wherein all tendency for further change has ceased' (Prausnitz, 1969).

The diagram below illustrates this concept of equilibrium exhibited in the VLE case. The phases are "closed" to the external environment since they cannot exchange mass beyond the system boundary. The internal system is regarded as "open" since mass and energy transfer is allowed via the phase boundary. The phases are in both thermal equilibrium ($T^\alpha = T^\beta$, zero[th] law of thermodynamics) and in mechanical equilibrium ($P^\alpha = P^\beta$).
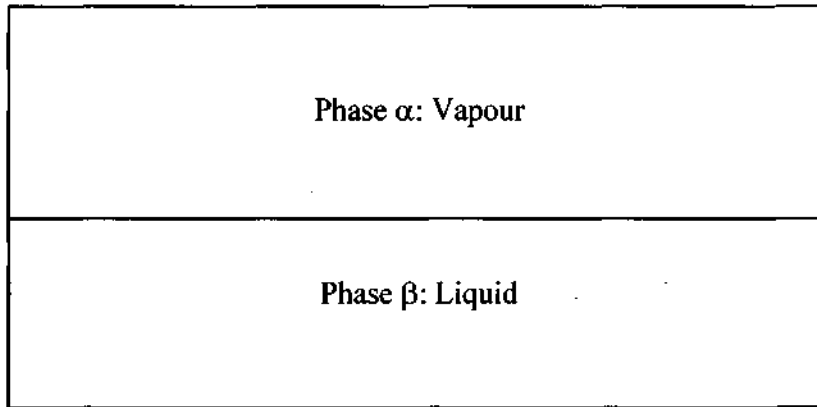
Figure 2-1 Depiction of vapour and liquid phases in an equilibrium state

The phases are denoted by 1 and 2.

Writing **Equation 2-4** in terms of the corresponding phases:

$$d\left(nG^{\alpha}\right) = \left(nV^{\alpha}\right)dP - \left(nS^{\alpha}\right)dT + \sum_{i} \mu_{i}^{\alpha} dn_{i}^{\alpha} \qquad \text{2-5}$$

$$d\left(nG^{\beta}\right) = \left(nV^{\beta}\right)dP - \left(nS^{\beta}\right)dT + \sum_{i} \mu_{i}^{\beta} dn_{i}^{\beta} \qquad \text{2-6}$$

The overall system change is found as the sum of the changes individual phases. **Equation 2-1** for a closed system implies that in order for equilibrium conditions to hold:

$$\sum_{i} \mu_{i}^{\alpha} dn_{i}^{\alpha} + \sum_{i} \mu_{i}^{\beta} dn_{i}^{\beta} = 0 \qquad \text{2-7}$$

Introducing the concept of mass conservation implies that

$$dn_{i}^{\alpha} = -dn_{i}^{\beta}$$

and

$$\sum_{i} \left(\mu_{i}^{\alpha} - \mu_{i}^{\beta}\right) dn_{i}^{\alpha} = 0$$

In order for this condition to be valid, it follows that

$$\mu_i^\alpha = \mu_i^\beta \qquad\qquad \text{2-8}$$

and this can be extended to any number of phases.

This is the key aspect that must be met for equilibrium to be satisfied and provides the starting point for one to relate μ to measurable properties.

2.4 Fugacity Coefficient

In order for the concept of chemical potential to be used in phase equilibrium, it needs to be quantified. However, it usually cannot be determined readily by experimental means. Thus it is necessary to relate this quantity to some measurable properties i.e. T, P and V. The concept of fugacity is utilized to relate chemical potential to measurable properties. Consider an ideal gas maintained at constant temperature:

$$dG = VdP = \frac{nRT}{P} dP = nRTd \ln P \qquad\qquad \text{2-9}$$

Integration leads to

$$G = G^* + RT \ln\left(\frac{P}{P^*}\right) \qquad\qquad \text{2-10}$$

where G* is the value of G at a reference pressure P*.

For a real mixture containing component i, the fugacity, denoted by $\hat{f}_i$ then

$$d\bar{G}_i = RTd \ln \frac{\hat{f}_i}{f^o}$$

- f^o refers to the standard fugacity. Common standard fugacities are the ideal gas state at 1 atm or that of the pure liquid.

Here G_i refers to the partial molar Gibbs free energy which is equivalent to μ_i. Therefore integration yields:

$$\mu_i = RT \ln \frac{\hat{f}_i}{f^o} + \theta(T)$$
2-11

Here $\theta(T)$ refers to the constant resulting from integration and is a function of temperature only. It is equal to the chemical potential of component i in the reference state. The criterion for equilibrium then emerges.

$$\hat{f}_i^\alpha = \hat{f}_i^\beta$$
2-12

The fugacity coefficient is defined as follows:

$$\varphi = \frac{f}{P}$$
2-13

With reference to a species in solution for component i, it becomes

$$\hat{\varphi}_i = \frac{\hat{f}_i}{y_i P}$$
2-14

The fugacity coefficient is used extensively to account for the departure of real gases from ideality via the use of residual properties. In general, a residual property is defined as

$$M^R = M - M^{ig}$$

where

- M - property of the real fluid
- M^{ig} - property of the ideal fluid

For detailed procedures on the computation of fugacity coefficient, the reader is referred to the texts by Abbott et al. (1996) and Prausnitz (1969).

2.5 Activity Coefficient

The computation of the fugacity coefficient requires integration of the difference between the Gibbs free energy of the ideal and real fluid from a state, where both are identical (ideal gas state at zero pressure) to the system state. This means that an equation of state has to be available, that describes the real fluid behaviour with sufficient accuracy at any pressure below the system pressure. In order to give reliable results for the fugacity coefficients in the liquid phase, it must also provide reliable results in the two-phase region. This is the reason why often a different procedure is used for the calculation of the liquid fugacity. Using the pure saturated liquid at system pressure and temperature as a convenient reference state, the description of the liquid phase real behaviour is facilitated by the introduction of an activity coefficient, γ_i.

$$\gamma_i = \frac{\hat{f}_i}{x_i f_i^S(P)} \qquad \text{2-15}$$

The standard state fugacity of pure component i is represented by $f_i^S(P)$. The liquid phase uses a property analogous to the residual property to account for the non-ideality. This property is the excess property.

$$M^E = M - M^{id}$$

M^{id} refers to the property of an ideal solution.

Ideal solution behaviour is characterized by the following equation:

$$\mu_i^{id} = G_i + RT \ln x_i \qquad \textbf{2-16}$$

Real solution behaviour can then be described by:

$$\mu_i = G_i + RT \ln \gamma_i x_i \qquad \textbf{2-17}$$

Therefore

$$\mu_i - \mu_i^{id} = \overline{G}_i - \overline{G}_i^{id} = RT \ln \gamma_i \qquad \textbf{2-18}$$

Thus one obtains the partial molar Gibbs energy.

$$\frac{\overline{G}_i^E}{RT} = \ln \gamma_i \qquad \textbf{2-19}$$

2.6 Computation Methods for VLE

Many engineering and design problems require the solution of the VLE. Consider a multicomponent system comprising N components then the independent variables are T, P, N-1 liquid phase mole fractions and N-1 vapour phase mole fractions. It follows that there are 2N independent variables. The phase rule dictates that N of these variables must be fixed in order to determine the N remaining variables. The remaining N variables are determined for the simultaneous solution of N equilibrium relations:

$$\hat{f}_i^l = \hat{f}_i^v \qquad (i=1, 2, ..., N) \qquad \textbf{2-20}$$

Typically, T or P and the liquid-phase or vapour-phase composition are specified which establishes the N variables. The N variables in question can then be determined.

In particular, many VLE systems exhibit low pressure that a simple equation of state is sufficient for the description of the vapour phase. Gibbs excess energy equations can be used to obtain liquid activity coefficients. The liquid phase fugacity of species i is:

$$\hat{f}_i^l = \gamma_i x_i f_i$$

and the corresponding vapour phase fugacity is:

$$\hat{f}_i^v = \hat{\phi}_i^v y_i P$$

It follows from **Equation 2-20** that

$$\gamma_i x_i f_i = \hat{\phi}_i y_i P \qquad\qquad \textbf{2-21}$$

This represents the gamma/phi formulation for VLE calculations. The fugacity f_i of the pure compressed liquid i is evaluated at T and P of the equilibrium mixture. This is done via two steps: first by calculating the fugacity coefficient of saturated vapour for pure species i at temperature T and vapour pressure $P = P_i^{sat}$ and then secondly by evaluating the change in fugacity of the liquid with a change in pressure to a value above or below P_i^{sat}. Thus f_i is derived as

$$f_i = \phi_i^{sat} P_i^{sat} \exp\left[-\frac{V_i^L\left(P - P_i^{sat}\right)}{RT} \right]$$

Equation 2-21 can be rewritten as

$$\gamma_i x_i P_i^{sat} = \Phi_i y_i P \qquad \textbf{2-22}$$

where $\Phi_i = \left(\dfrac{\hat{\phi}_i}{\phi_i^{sat}}\right)\exp\left[-\dfrac{V_i^L\left(P - P_i^{sat}\right)}{RT}\right]$

When the vapour and liquid phase can be considered as ideal then **Equation 2-22** reduces to a simple expression. In the case of an ideal gas then the fugacity coefficients $\hat{\phi}_i$ and ϕ_i^{sat} are unity and for all practical purposes $\Phi_i = 1$. The activity coefficients are also unity in the case of an ideal solution. Thus **Equation 2-22** becomes

$$y_i P = x_i P_i^{sat} \qquad \text{2-23}$$

This is Raoult's law. It represents the most rudimentary relation for VLE. Hence it fails to adequately describe real behaviour of most systems.

At low to moderate pressures, a reasonable assumption, that the ideal gas model adequately describes the vapour phase can be made. This provides a more realistic approach and the VLE relation can then be expressed as:

$$\gamma_i x_i P_i^{sat} = y_i P \qquad \text{2-24}$$

This relation is referred to as modified Raoult's Law. Although this provides a more accurate description than that of Raoult's law, its applicability is still limited to components that are sub-critical.

The gamma/phi method in general is restricted to systems containing components that are sub-critical. It is used typically when the system in question exhibits pressure no more than a few bars. In addition, it is adequate for the correlation of constant-temperature data. Although the local composition G^E models do contain some temperature dependence for the parameters, these are only an approximate (Perry, 1998).

Another possible VLE relation is where both liquid and vapour phases are described by an equation via fugacity coefficient. In this case the liquid and vapour phase are defined as:

$$\hat{f}_i^l = \hat{\phi}_i^l x_i P \quad - \qquad \text{Liquid}$$

$$\hat{f}_i^v = \hat{\phi}_i^v y_i P \quad - \qquad \text{Vapour}$$

Hence 2-20 becomes

$$x_i \hat{\phi}_i^l = y_i \hat{\phi}_i^v \qquad \qquad \text{2-25}$$

This approach has found its use in the high pressure domain. Here $\hat{\phi}$ is a function of T, P and composition and is evaluated from an appropriate equation of state with the aid of mixing rules. However, the use of equation of state for the liquid phase fugacity has been limited to systems with small to moderate deviations from ideal solution behaviour. This was attributed to inadequacies in empirical mixing rules. Wong and Sandler (1992) developed a new class of mixing rules for the cubic equation of state and this has improved its application to VLE. The Soave/Redlich/Kwong (SRK) and Peng/Robinson(PR) equations of state in combination with the Wong/Sandler mixing rules can accurately correlate and predict VLE data.

2.7 Liquid-liquid Equilibrium

For two liquid phases in equilibrium the fugacities of all the components are equal in both phases.

$$f_i' = f_i'' \qquad \qquad \text{2-26}$$

Using equations of state and fugacity coefficients, then

$$\left(x_i \varphi_i^L\right)' = \left(x_i \varphi_i^L\right)'' \qquad \qquad \text{2-27}$$

Using G^E model and activity coefficients, then

$$\left(x_i\gamma_i\right)' = \left(x_i\gamma_i\right)'' \qquad \text{2-28}$$

From the definition of activity,

$$a_i = x_i\gamma_i \qquad \text{2-29}$$

it follows that

$$a_i' = a_i'' \qquad \text{2-30}$$

2.8 Representation of Ternary Liquid-Liquid Equilibrium

Ternary data measured under isothermal conditions are represented on ternary diagrams.

Based on the properties of the constituent binary combinations, ternary systems are classified into three common categories.

Type	Description
1	One binary combination is partially miscible
2	Two binary combinations are partially miscible and the third type is completely miscible
3	All binary combinations are miscible.

Table 2-1 Classification of the different ternary systems
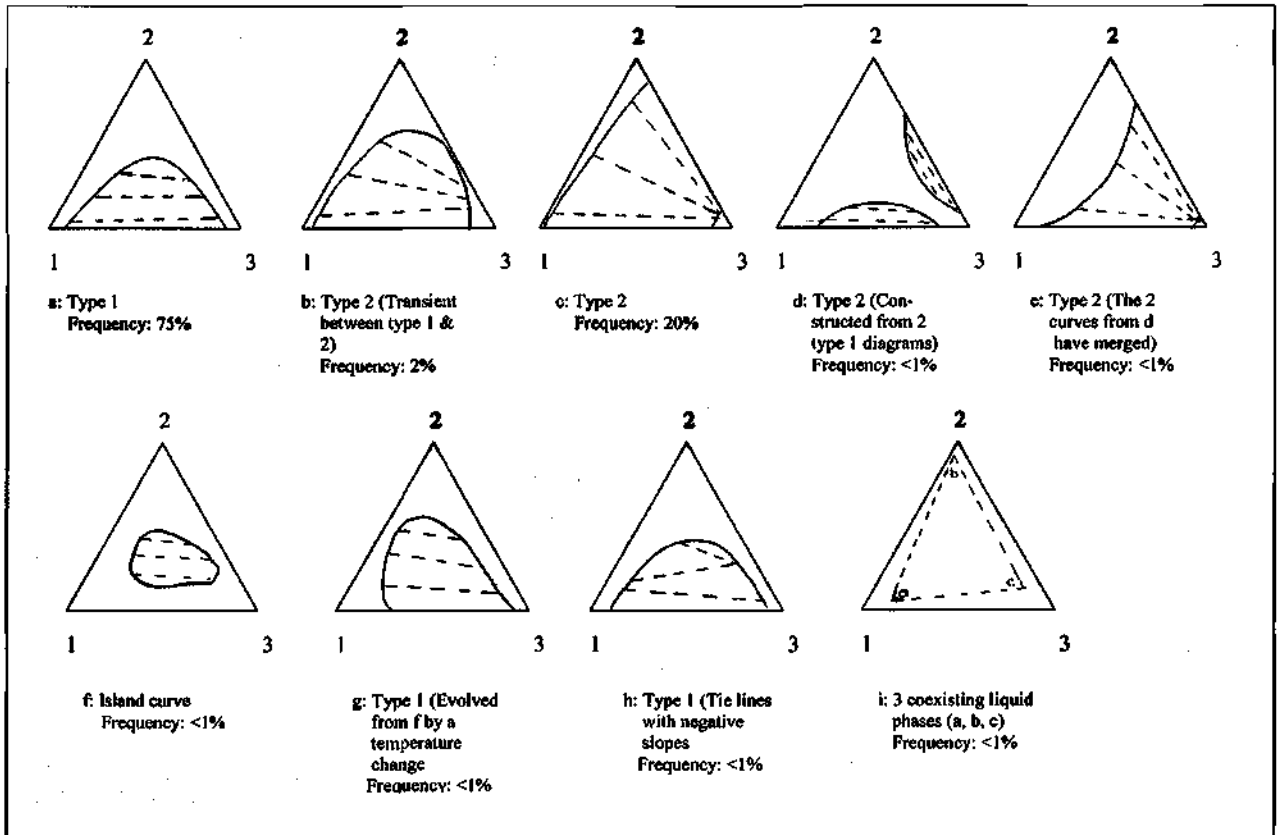
Figure 2-2 The different ternary LLE type diagrams

A typical type 1 system would be water-acetic acid-methylisobutylketone. N-butanol-butyl propionate-water is an example of a type 2 system. Often mixtures change type with a change in temperature. In **Figure 2-3** a system changes from type 2 to type 1 as temperature is increased. Types b, f and g can also be obtained from changes in temperature (**Figure 2-4**).
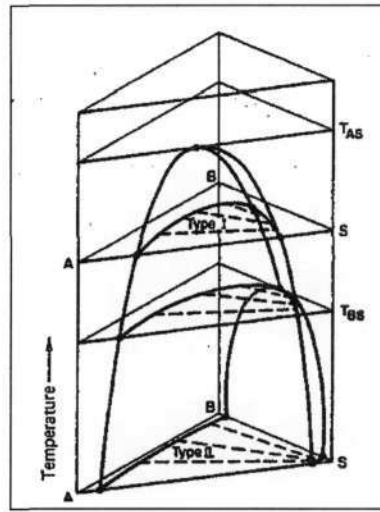
Figure 2-3 Effect of temperature on ternary LLE .A – feed solvent, B – solute, S – extraction solvent (Perry, 1997)
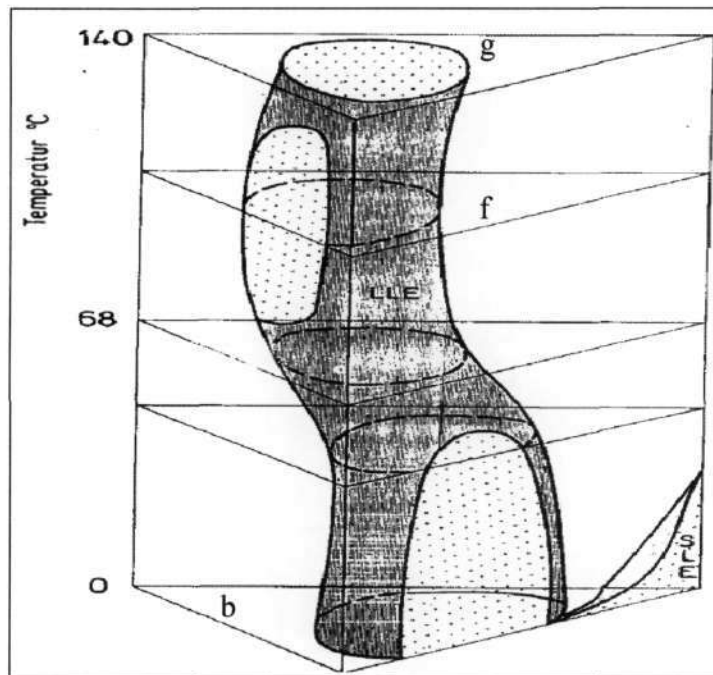
Figure 2-4 Effect of temperature on ternary LLE

2.9 Computation of Liquid-liquid equilibrium with the K-Factor Method
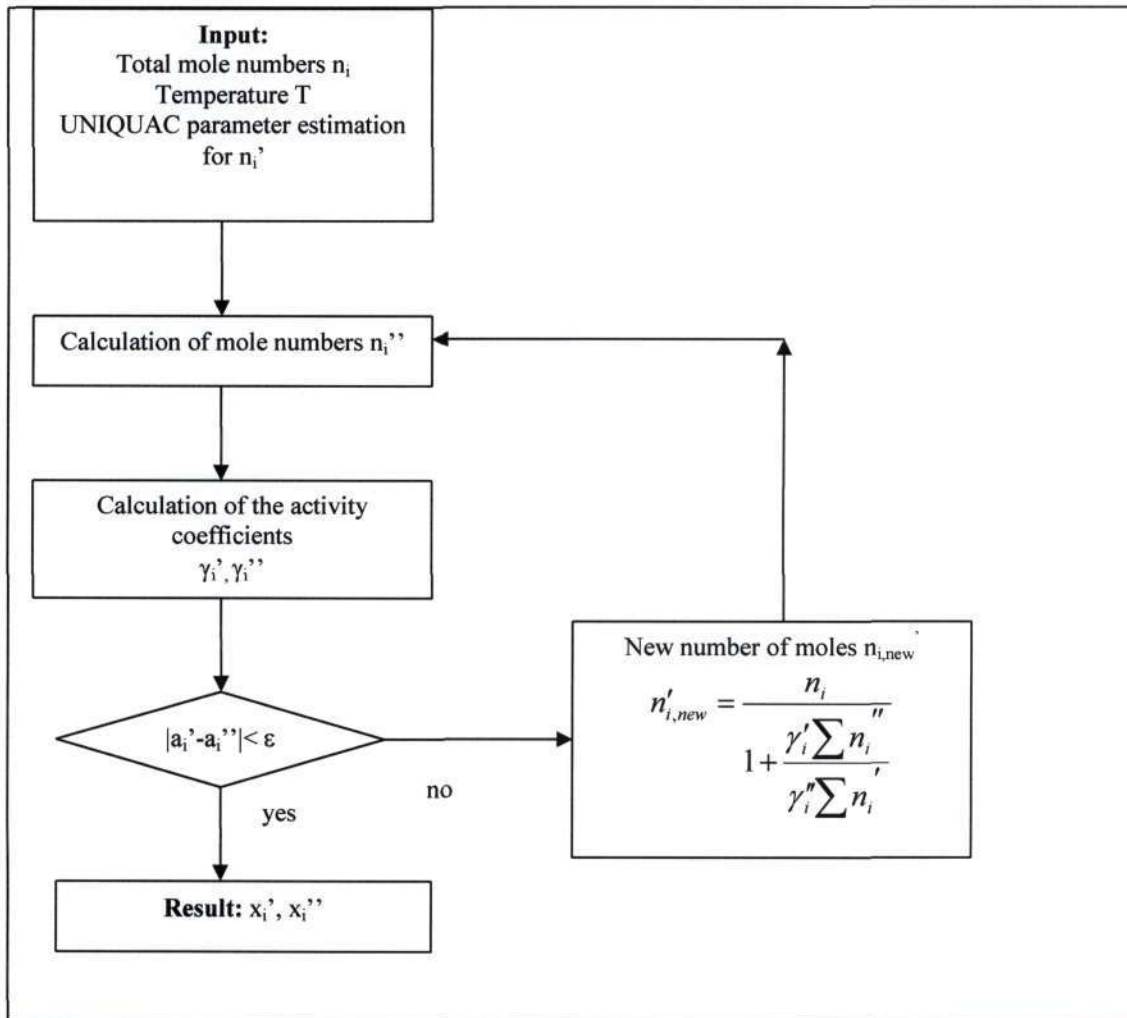
Figure 2-5 Algorithm for the calculation of multicomponent LLE using the K-Factor Method

2.10 Data Quality and Model Selection

It is critical that the thermodynamicist selects the most appropriate model for data reduction based on the chemical nature of the mixture (Raal and Muhlbauer, 1998). In cases where an applicable model is not apparent, several models may have to be evaluated to assist in choosing the best model.

The quality of equilibrium data has to also be addressed. Data has to meet some criteria in order to allow model fitting and prediction of data within a certain tolerable window of error. As a preliminary step to the regression process, data has to be checked for thermodynamic consistency. In the case of questionable data, one is strongly advised to re-measure data (Raal and Muhlbauer, 1998). Consistency tests based on the Gibbs-Duhem equation are used primarily to establish the credibility of VLE data. These testing procedures are detailed in Abbott et al., 1996. However, its application to high pressure VLE data is very difficult.

If data passes the consistency tests, it is still insufficient to classify it as accurate or of superior quality. One has to make the distinction between accuracy and precision in light of experimental work.

Consider the dartboard as an example which clearly illustrates the distinction between the two concepts.

good precision but
poor accuracy   poor precision and
                       poor accuracy   good precision and
                                              good accuracy
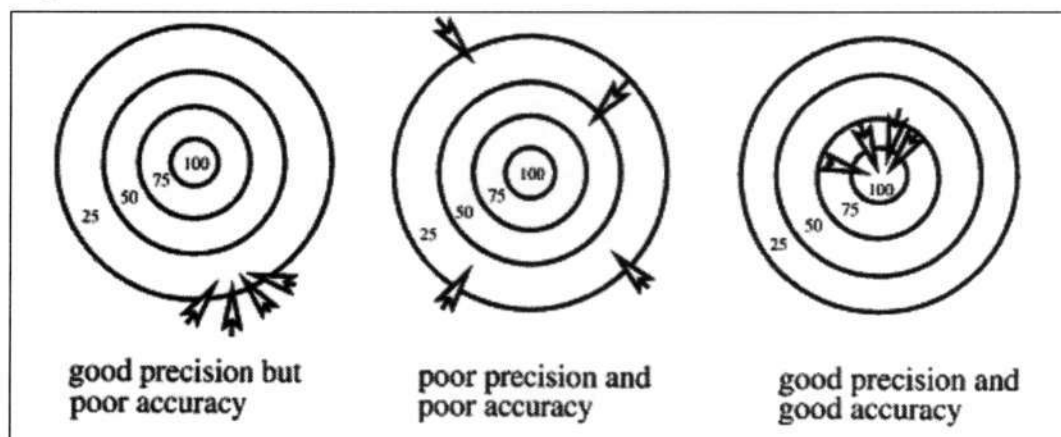
Figure Error! No text of specified style in document.**-1 Distinction between consistency and accuracy**

Accuracy of measurement refers to how close a measurement approaches the accepted or correct value. Precision refers to the spread of measurements. It is evident from the first dartboard that good precision does not necessarily imply a high degree of accuracy. The experimentalist thus aims to achieve a good combination of good precision and good accuracy by avoiding personal, systematic and random errors where possible.

One can thus conclude that although VLE data may be consistent, it could still be deemed inaccurate. Data may not conform to a model due to the presence of bad data points even though an appropriate model was used. In some cases, the data could be accurate but the selected model fails to reasonably reproduce the experimental data. For these reasons, it is essential to examine the data quality and thereafter select the best suited model.

2.10 Liquid Solution Theories and Evolution of G^E-Models

2.10.1 Introduction

Liquid theories have attempted to improve the quality of G^E-models by establishing a more substantial theoretical basis and thereby enhancing data correlation, extrapolation and prediction. The purpose of liquid theory is to be able to meticulously describe the behaviour of a real liquid mixture and to quantify the relevant liquid properties. A theoretically correct description of a liquid mixture is however an intricate and multi-faceted study and research in this field has not yielded a plausible theory that encompasses all dynamics of a liquid mixture. The chemical industry is expansive and vast and the notion of a sound theory that has a wide range of applicability seems almost inconceivable. Nevertheless existing theories provide a fair basis for the development and investigation of models.

This section aims to provide a brief overview of some of the theories that have been developed and to give the reader some insight into the evolution of G^E-models.

2.10.2 Lattice Theory

Lattice theory attempts to describe liquid behaviour by recognising the fact that the liquid phase is a transition between the solid and gas phase. It follows that a liquid can be envisioned as a real gas and the respective properties would be computed by an equation of state.

The latter option would then be to consider the liquid solution particles to have more restricted movement than that of the gas phase and hence the particles are more ordered in their configuration. The particles are then assumed to possess a semi-crystalline structure and this "structure" is termed a lattice. This forms the basis of the concept supporting lattice theory.

To adequately describe the behaviour of a liquid mixture one has to consider:

- Intermolecular forces between similar and dissimilar molecules
- Molecule size and shape and its effects on lattice configuration
- Magnitude of intermolecular forces and its effect on lattice configuration.

Figure 2-7 represents a mixture of 2 components and the configuration of the lattice after mixing. Using knowledge of statistical mechanics, expressions for the enthalpy and entropy of mixing can be found.
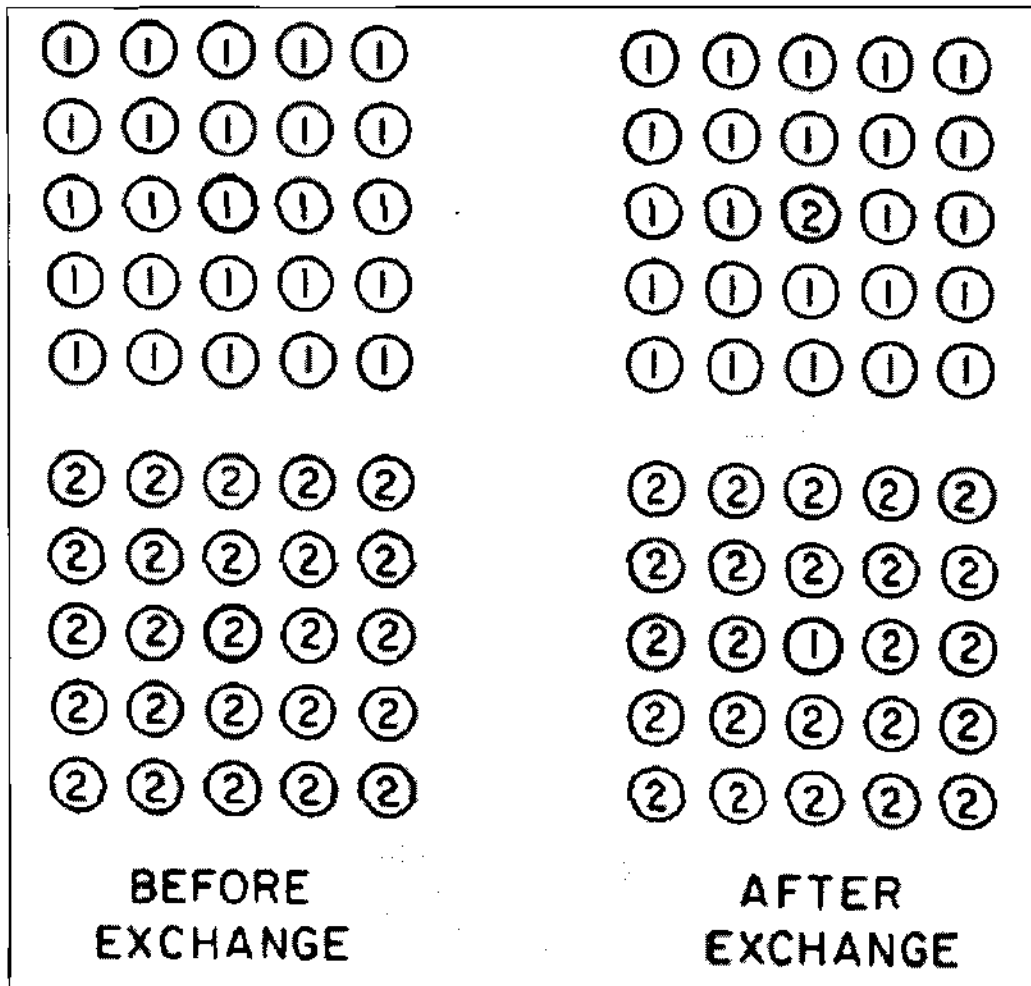
Figure 2-7 Lattice before and after interchange (Prausnitz, 1969)

The excess Gibbs energy is given by:

$$G^E = N_A \omega x_1 x_2 \qquad \text{2-31}$$

where

- N_A - Avagadro's Number
- ω - Interchange Energy

The interchange energy is defined as:

$$\omega = Z\left[\Gamma_{12} - 0.5(\Gamma_{11} + \Gamma_{22})\right] \qquad \text{2-32}$$

where

- Z - Coordination Number

- Γ_{1i} - Potential energy of pair 1-i

$$\ln \gamma_i = \frac{\omega}{kT} x_j^2 \qquad \text{2-33}$$

Here k refers to the Boltzmann constant.

This theory however assumes a completely random mixture. This argument can only be reasonable if all possible interactions (self association between the same components and cross association between different components) are either identical or much smaller than RT. In the case of a real mixture the energies of interaction are clearly not identical to those in the pure fluids.

Guggenheim utilised a quasi-chemical approximation in conjunction with lattice theory to simulate non-random mixtures (Prausnitz, 1969) which produced the following G^E expression:

$$\frac{G^E}{RT} = \frac{\omega}{kT} x_1 x_2 \left[1 - 0.5\left(\frac{2\omega}{zkT}\right) x_1 x_2 + \ldots\right] \qquad \text{2-34}$$

This equation does not produce superior predictive performance in the case of miscible mixtures. It does however prove far superior to lattice theory in the case of a mixture involving limited miscibility.

2.10.3 Flory-Huggins

This theory addresses the issue of non-zero entropy by considering the energy evolved upon mixing to be zero. This idealised solution is termed an athermal solution. The theory adopts the notion of a lattice theory.

Flory and Huggins assume a polymer molecule simulates chain behaviour and is composed of solvent molecule segments. Each segment now replaces a molecule position in the lattice. Volume fractions of the solvent and polymer are given respectively by:

$$\Phi_1 = \frac{n_1}{n_1 + mn_2} \qquad \Phi_2 = \frac{n_2}{n_1 + mn_2} \qquad \text{2-35}$$

where

- m - no. of segments
- n_1 = no. of solvent moles
- n_2 = no. of polymer moles

The excess Gibbs energy and activity coefficient expressions for an amorphous polymer are given by:

$$\frac{G^E}{RT} = \sum_i n_i \ln \Phi_i \qquad \text{2-36}$$

$$\ln \gamma_1 = \ln\left[1 - \left(1 - \frac{1}{m}\Phi_2\right)\right] + \left(1 - \frac{1}{m}\right)\Phi_2 \qquad \text{2-37}$$

However, since athermal behaviour is not realistic, the equation is modified by including an enthalpic term to account for energy of mixing.

$$\frac{G^E}{RT} = \sum_i n_i \ln \Phi_i + \chi\Phi_1\Phi_2\left(\bar{n}_1 + mn_2\right) \qquad \text{2-38}$$
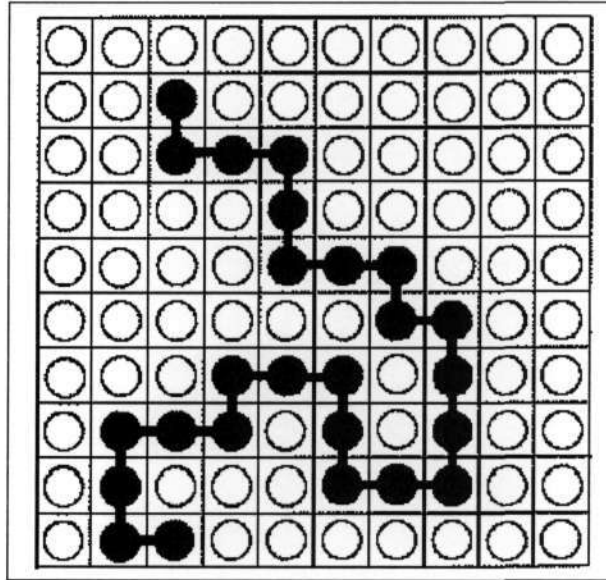
**Figure 2-8 Lattice model for a polymer chain in solution, symbols represent solvent molecules (▢)
and polymer-chain segments (▆)**

$$\ln \gamma_1 = \ln\left[1 - \left(1 - \frac{1}{m}\Phi_2\right)\right] + \left(1 - \frac{1}{m}\right)\Phi_2 + \chi\Phi_2^2 \qquad 2\text{-}39$$

where χ is the Flory interaction parameter derived from energy interaction of molecules.

2.10.4 Wilson's extension of Flory-Huggin's equation

Wilson used Flory and Huggin's equation and modified it by accounting for molecular interactions. Here Wilson considers that there are excess enthalpies and entropies, but that the excess volume is zero. Based on a binary case the ratio of the number of molecules around a central molecule can be expressed as the product of the ratio of mole fractions and the respective Boltzmann factors. The factors are representative of the potential energy interactions (between components i-i and i-j).

$$\frac{x_{ij}}{x_{ii}} = \frac{x_j \exp\left(\frac{-\lambda_{ij}}{RT}\right)}{x_i \exp\left(\frac{-\lambda_{ii}}{RT}\right)} \qquad \textbf{2-40}$$

where λ_{ij} and λ_{ii} are energies of interaction.

The local volume fractions are then defined as:

$$\xi_i = \frac{\upsilon_{ii} x_{ii}}{\upsilon_{ii} x_{ii} + \upsilon_{ij} x_{ij}} \qquad \textbf{2-41}$$

Wilson uses these local volume fractions to replace overall volume fractions Φ_i in the Flory-Huggins model.

$$\frac{G^E}{RT} = \sum_i x_i \ln \frac{\xi_i}{x_i} \qquad \textbf{2-42}$$

By introducing the parameters Λ_{ij} and Λ_{ji} the equation reduces to

$$\frac{G^E}{RT} = -\sum_i x_i \ln\left(\sum_j x_j \Lambda_{ij}\right) \qquad \textbf{2-43}$$

2.10.5 Two - Liquid Theory

This theory assumes that the properties of a mixture can be related to a hypothetical fluid and can be considered to be the composition averages of the constituent components. For the binary case, properties are derived by composition averages of the two hypothetical fluids.

A central molecule is contained in a region called a molecule cell and is surrounded by molecules from the mixture. Thus a binary mixture contains two cell-types with different cell molecules. Thus any extensive residual property M of the mixture can be found from $M^{(i)}$, the residual property of the fluid of cell type i.

$$M = x_1 M^{(1)} + x_2 M^{(2)}$$
2-44

This theory can be extended to an n-component mixture with n types of cells and this is referred to as n-fluid theory. This forms the basis for the derivation of the NRTL and UNIQUAC models which are discussed shortly.
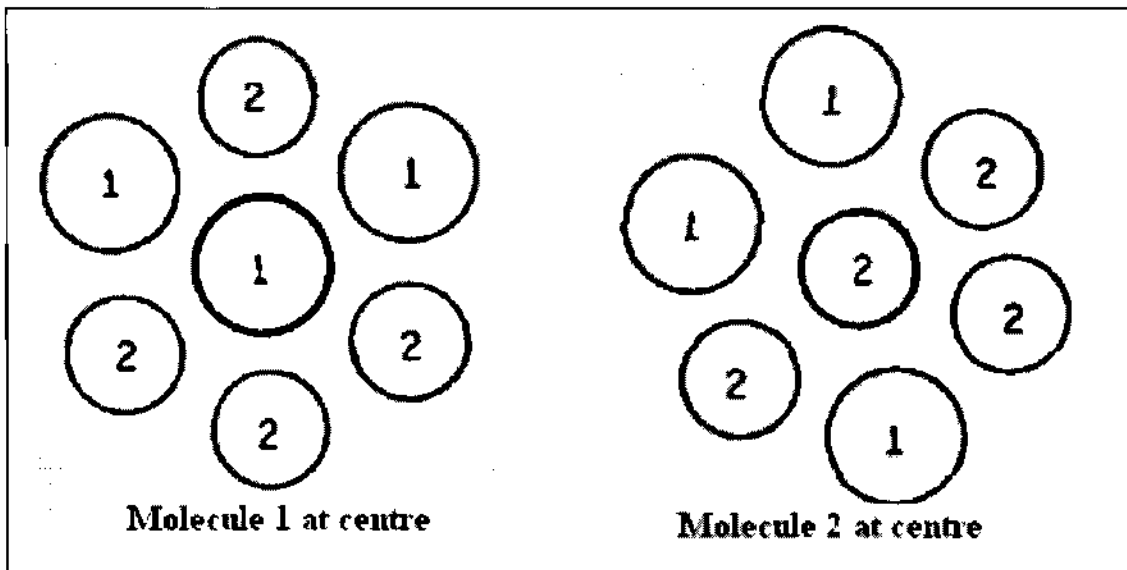
Figure 2-9 The two cell types forming the basis of the two-liquid theory

2.10 G^E-Models

2.10.1 Introduction

Since the focus of this research is the modification of G^E-models to improve their correlative and predictive ability, a brief review of current G^E-models is presented here. Models differ based on the number of adjustable parameters, applicability to binary and higher system of components, their nature (semi-theoretical or empirical) and degree of complexity and ease of computation. The chemical nature includes difference in molecular size and self and cross interaction of molecules in the mixture. Some models can be extended to multicomponent mixtures; however it is usually necessary to first compute binary parameters.

Activity coefficients are functions of temperature, pressure and composition. The pressure dependence is considered to have a negligible influence on the activity coefficient in the low pressure range (Abbott et al., 1996).

The interaction parameters may contain inherent temperature dependence. In order to establish temperature dependence for these parameters, a number of isothermal data sets would have to be regressed. In the case of regression of isobaric data, the temperature dependence of these parameters has to be accounted for and should not be ignored (Prausnitz, 1969).

This section outlines the applicability of models in addition to advantages and disadvantages to assist in choosing the best model.

2.10.2 Margules

This is perhaps the most rudimentary equation correlating G^E to composition. The simplest form is the Porter (2- suffix Margules) equation.

$$\frac{G^E}{RTx_1x_2} = A \qquad \text{2-45}$$

$$\ln \gamma_1 = Ax_2^2 \qquad \text{2-46}$$

$$\ln \gamma_2 = Ax_1^2 \qquad \text{2-47}$$

The use of this equation is limiting due to its obvious simplicity. It is symmetric with regard to the mole fractions x_1 and x_2. This is however unrealistic since most systems exhibit asymmetric behaviour and the model fails in this regard. Thus the model is used for a preliminary estimate of trends (Raal and Muhlbauer, 1998).

$$\frac{G^E}{RTx_1x_2} = A_{21}x_1 + A_{12}x_2 \qquad \text{2-48}$$

$$\ln \gamma_1 = x_2^2\left(A_{12} + 2(A_{21} - A_{12})x_1\right) \qquad \text{2-49}$$

$$\ln \gamma_2 = x_1^2\left(A_{21} + 2(A_{12} - A_{21})x_2\right) \qquad \text{2-50}$$

The 3-suffix expression above was derived to correlate more complex systems (A_{21} and A_{12} are the model constants). It does provide good reproduction of many non-ideal systems (Prausnitz, 1969). The model is empirical in nature and assumes that the components of the binary mixture have equal molecular size.

By the introduction of another term the 4-suffix equation is derived. This is appropriate when a number of accurate data points have been measured (Prausnitz, 1969).

Margules equations are only applicable to binary mixtures. Further, the models cannot be extended to multicomponent systems due to their empirical nature (Abbott et al., 1996).

2.10.3 Redlich-Kister

$$\frac{G^E}{RTx_1x_2} = A + B(x_1 - x_2) + C(x_1 - x_2)^2 + D(x_1 - x_2)^3 + \dots \quad \text{2-51}$$

Here B, C, and D are the model parameters that contain a temperature dependence that is determined from experimental data. The number of parameters chosen to represent a mixture is determined by the chemical nature of the mixture. Due to the model's construction the even powered terms are symmetric with regard to x and the odd powered terms are asymmetric and manipulate the shape of the G^E plot (Raal and Muhlbauer (1998)).

$$RT \ln \gamma_1 = a^{(1)}x_2^2 + b^{(1)}x_2^3 + c^{(1)}x_2^4 + d^{(1)}x_2^5 \quad \text{2-52}$$

where

- $a^{(1)} = A + 3B + 5C + 7D$
- $b^{(1)} = -4(B + 4C + 9D)$
- $c^{(1)} = 12(C + 5D)$
- $d^{(1)} = -32D$

This model is as effective as the 3- and 4-suffix Margules model.

2.10.4 Van Laar

By rewriting $\dfrac{G^E}{RTx_1x_2}$ as a reciprocal expression one obtains a polynomial similar to the Redlich-Kister expansion.

$$\frac{x_1x_2}{G^E/RT} = B + C(x_1 - x_2) \quad \text{2-53}$$

or

$$\frac{G^E}{x_1 x_2 RT} = \frac{A'_{12} A'_{21}}{A'_{12} x_1 + A'_{21} x_2} \qquad \text{2-54}$$

$$\ln \gamma_1 = A'_{12} \left[1 + \frac{A'_{12} x_1}{A'_{21} x_2} \right]^{-2} \qquad \text{2-55}$$

$$\ln \gamma_2 = A'_{21} \left[1 + \frac{A'_{21} x_2}{A'_{12} x_1} \right]^{-2} \qquad \text{2-56}$$

The Van Laar equation was proposed to accommodate for the differences in size of molecules and has its foundation in the Van der Waals equation. It can also be derived from Wohl's expansion with unlike molecule sizes. Despite the model's ability to account for molecule size differences, it is still inadequate in characterizing highly non-ideal systems. This model does not account for molecular interactions.

2.10.5 Wilson

This model is a pure entropic model. It describes Gibbs excess energy as $G^E = RT \sum_i \ln(f(x))$.

Unlike interactions that are used in the Wilson model are solely to quantify the local composition. Unlike the Guggenheim model, no enthalpic term is included. This results in problems in reproducing larger G^E values.

This equation is derived based on the concept of local composition. Wilson (1964) accounts for the size differences in molecules and the intermolecular energy interactions.

$$\frac{G^E}{RT} = -x_1 \ln(x_1 + x_2\Lambda_{12}) - x_2 \ln(x_2 + x_1\Lambda_{21})$$ 2-57

The adjustable parameters are defined as follows:

$$\Lambda_{ij} = \frac{v_j}{v_i} \exp\left[\frac{-\Delta\lambda_{ij}}{RT}\right]$$ 2-58

Here v_i refers to the pure liquid molar volumes of the components. λ_{ij} is the parameter characterising the molecular interactions between components i and j.

λ_{12} and λ_{21} can be considered to be independent of temperature over narrow temperature ranges (Prausnitz, 1998). Introducing the parameter's temperature dependence may not produce significant differences. The Wilson equation proves its worth in the cases of miscible mixtures, notably for mixtures containing polar and non-polar components. Its performance is superior to that of Van Laar and the 3- Suffix Margules in these cases.

The Wilson equation cannot be used for mixtures with extrema in the logarithms of activity coefficients (Prausnitz, 1998) and cannot predict liquid immiscibility. This limits its use to totally miscible systems. By including a parameter C, this problem was overcome, but this too was not without complications. The complexity of the extension to multicomponent mixtures becomes problematic. The C parameter increases the interdependence of the parameters.

Wilson can be extended to multicomponent mixtures where parameters are required for each binary pair in the multicomponent mixture.

$$\frac{G^E}{RT} = -\sum_i x_i \ln\sum_j (x_j\Lambda_{ij})$$ 2-59

$$\ln \gamma_k = -\ln\left(\sum_{j=1}^{m} x_j \Lambda_{kj}\right) + 1 - \sum_{i=1}^{m} \frac{x_i \Lambda_{ik}}{\sum_{j=1}^{m} x_j \Lambda_{ij}} \qquad \textbf{2-60}$$

2.10.6 NRTL

This model was developed by Renon and Prausnitz (1968) and is capable of describing liquid miscibility. It has wide applicability and noted applicability to highly non-ideal mixtures and mixtures exhibiting liquid immiscibility.

Unlike the Wilson equation, NRTL has the interaction energy in the calculation of the local composition (in the exponent) and for the calculation of the total G. However, NRTL has no explicit entropic part.

$$\frac{G^E}{RT} = x_1 x_2 \left(\frac{\tau_{21} G_{21}}{x_1 + x_2 G_{21}} + \frac{\tau_{12} G_{12}}{x_2 + x_1 G_{12}} \right) \qquad \textbf{2-61}$$

$$\ln \gamma_1 = x_2^2 \left[\tau_{21} \left(\frac{G_{21}}{x_1 + x_2 G_{21}} \right)^2 + \left(\frac{\tau_{12} G_{12}}{(x_2 + x_1 G_{12})^2} \right) \right] \qquad \textbf{2-62}$$

$$\ln \gamma_2 = x_1^2 \left[\tau_{12} \left(\frac{G_{12}}{x_2 + x_1 G_{12}} \right)^2 + \left(\frac{\tau_{21} G_{21}}{(x_1 + x_2 G_{21})^2} \right) \right] \qquad \textbf{2-63}$$

$$\tau_{ji} = \frac{g_{ji} - g_{ii}}{RT} \qquad \textbf{2-64}$$

$$G_{ij} = \exp(-\alpha_{ij} \tau_{ji}) \qquad \textbf{2-65}$$

In the case of a binary system the adjustable parameters are Δg_{12} and Δg_{21}, which represent the Gibbs free energy of interaction. The parameter α is an additional parameter and addresses the issue of non-randomness in solution. Monte Carlo simulations have shown that α_{12} ranges from 0.20 to 0.47 and thus its value is usually set at 0.3. (Prausnitz, 1969)

The model has limited explicit temperature dependence and the interdependence of parameters is increased (Raal and Muhlbauer, 1998).

The model can be extended to multicomponent mixtures.

$$\frac{G^E}{RT} = \sum_{i=1}^{m} x_i \frac{\sum_{j=1}^{m} \tau_{ji} G_{ji} x_j}{\sum_{l=1}^{m} G_{li} x_l} \qquad \text{2-66}$$

$$\ln \gamma_i = \frac{\sum_{j=1}^{m} \tau_{ji} G_{ji} x_j}{\sum_{l=1}^{m} G_{li} x_l} + \sum_{j=1}^{m} \frac{x_j G_{ij}}{\sum_{l=1}^{m} G_{lj} x_l} \left(\tau_{ij} - \frac{\sum_{r=1}^{m} x_r \tau_{rj} G_{rj}}{\sum_{l=1}^{m} G_{lj} x_l} \right) \qquad \text{2-67}$$

2.10.7 UNIQUAC

This equation is derived using the two-fluid theory. The model was derived to handle miscible mixtures using two parameters. Thereby it would be superior to other local composition models. The model is an extension of the quasi-chemical theory of Guggenheim.

Local composition equations for excess Gibbs energy are inconsistent when based on the one-fluid lattice model. UNIQUAC is based on the two-fluid theory and hence the inconsistencies do not apply. A brief derivation based on Maurer & Prausnitz (1978) is presented in *Appendix A: A1*

– *UNIQUAC Derivation* since UNIQUAC is the precursor to both the FlexQUAC and FlexQUAC-Q model.

$$\left(\frac{G^E}{RT}\right) = \left(\frac{G^E}{RT}\right)_{combinational} + \left(\frac{G^E}{RT}\right)_{residual} \qquad \textbf{2-68}$$

where

- $$\frac{G^E}{RT}_{combinational} = x_1 \ln\frac{\Phi_1}{x_1} + x_2 \ln\frac{\Phi_2}{x_2} + \frac{Z}{2}\left(q_1 x_1 \ln\frac{\theta_1}{\Phi_1} + q_2 x_2 \ln\frac{\theta_2}{\Phi_2}\right) \qquad 2\text{-}69$$

- $$\frac{G^E}{RT}_{residual} = -q_1 x_1 \ln(\theta_1 + \theta_2 \tau_{21}) - q_2 x_2 \ln(\theta_2 + \theta_1 \tau_{12}) \qquad 2\text{-}70$$

- $$\Phi_1 = \frac{x_1 r_1}{x_1 r_1 + x_2 r_2} \; ; \Phi_2 = \frac{x_2 r_2}{x_1 r_1 + x_2 r_2} \qquad 2\text{-}71$$

Unlike the two former local composition models, UNIQUAC contains both an entropic and energy interaction terms which account for two different aspects of real solution behaviour. This makes it superior to its predecessors. However UNIQUAC still utilizes the same simplified calculation of local composition as Wilson and NRTL.

The combinatorial part accounts for the varying sizes and shapes of molecules while the residual part factors in the presence of intermolecular force interactions (Prausnitz, 1969). The combinatorial part requires pure component size and surface fractions (r and Q). These are derived from Bondi (1968) or from group contributions (Raal and Muhlbauer, 1998). The binary parameters are contained in the residual part of the model. The coordination number Z is usually set to a value of ten.

To improve the performance of the model, for mixtures of waters and alcohols, Anderson and Prausnitz, (1978) adjusted the q values (q' in this case). q' is found to be smaller than q in the case of alcohols and water due to the presence of hydrogen bonding. For all other systems q = q'.

The two parameters, τ_{12} and τ_{21} are obtained from regression of experimental equilibrium data. Δu_{ij} is referred to as the characteristic energy and is considered to be a weak function of temperature (Prausnitz, 1969).

The corresponding activity coefficient expression is:

$$\ln \gamma_i = \ln \gamma_{i\,combinatorial} + \ln \gamma_{i\,residual}$$

$$\ln \gamma_{1\,combinatorial} = \ln \frac{\Phi_1}{x_1} + \frac{Z}{2} q_1 \ln \frac{\theta_1}{\Phi_1} + \Phi_2 \left(l_1 - \frac{r_1}{r_2} l_2 \right) \qquad 2\text{-}72$$

$$\ln \gamma_{1\,residual} = -q_1' \ln\left(\theta_1' + \theta_2' \tau_{21}\right) + \theta_2' q_1' \left(\frac{\tau_{21}}{\theta_1' + \theta_2' \tau_{21}} - \frac{\tau_{12}}{\theta_2' + \theta_1' \tau_{12}} \right) \quad 2\text{-}73$$

where

$$l_1 = \frac{Z}{2}(r_1 - q_1) - (r_1 - 1)$$

The UNIQUAC model can be extended to multicomponent mixtures.

$$\frac{G^E}{RT}_{combinatorial} = \sum_{i=1}^{m} x_i \ln \frac{\Phi_i}{x_i} + \frac{Z}{2} \sum_{i=1}^{m} q_i x_i \ln \frac{\theta_i}{\Phi_i} \qquad 2\text{-}74$$

$$\frac{G^E}{RT}_{residual} = -\sum_{i=1}^{m} q_i' x_i \ln\left(\sum_{j=1}^{m} \theta_j' \tau_{ji} \right) \qquad 2\text{-}75$$

$$\ln \gamma_i = \ln\frac{\Phi_i}{x_i} + \frac{Z}{2}q_i \ln\frac{\theta_i}{\Phi_i} + l_i - \frac{\Phi_i}{x_i}\sum_{j=1}^{m}x_j l_j - q_i'\ln\left(\sum_{j=1}^{m}\theta_j'\tau_{ji}\right) + q_i' - q_i'\sum_{j=1}^{m}\frac{\theta_j'\tau_{ij}}{\sum_{k=1}^{m}\theta_k'\tau_{kj}} \qquad 2\text{-}76$$

The major advantages of UNIQUAC that has made it superior to other models are:

- applicability to multicomponent mixtures
- applicability to LLE
- Inherent temperature dependency for many systems valid over a moderate range
- superior representation for molecules of widely different molecular sizes, suitable for non-ideal mixtures
- its basis for one of the most widely used predictive group contribution method UNIFAC

The model is however limited by its algebraic complexity and the availability of r and q parameters.

2.11 Group Contribution Methods

All the models discussed previously utilize interaction parameters which must be obtained from the regression of experimental data. When experimental data is limited, the use of group contribution methods is preferred. Here a molecule is fragmented into structural groups. It is assumed that the functional groups behave as if they were isolated from the molecule they are contained in. Thus the intermolecular interactions are "weighted sums of group-group interactions" (Prausnitz, 1998). Reduction of experimental data for structural group interaction then allows molecular interactions to be quantified. This section briefly reviews some group contribution methods.

2.11.1 ASOG

The Analytical solution of groups is based on the solution of groups by Wilson and Deal (1962). Here the chemical potential of a component i in solution (equivalent to lnγ) is considered to be the sum of two parts:

$$\ln \gamma_i = \ln \gamma_i^{FH} + \ln \gamma_i^{G} \qquad \text{2-77}$$

- $\ln \gamma_i^{FH}$ relates to the size contribution

- $\ln \gamma_i^{G}$ is related to the interaction of the structural groups of the molecule with the rest of the system

These terms in a sense take into account entropic and enthalpic contributions.

The size term has its roots in the Flory-Huggins theory:

$$\ln \gamma_i^{FH} = \ln R_i + 0.434(1 - R_i) \qquad \text{2-78}$$

- R_i refers to the ratio of solute groups to the total number of groups in the average liquid molecule

$$R_i = \frac{v_i^{FH}}{\sum_j v_j^{FH} x_j} \qquad \text{2-79}$$

- v_j^{FH} refers to the number of size groups in each molecular species in solution and the summation is taken over all the species

The interaction term is defined as the difference between the group contribution which occur in solution and in the molecular standard states. The summation of all these interactions are considered for the groups contained in the molecule.

$$\ln \gamma_i^G = \sum_k v_{ki} \left(\ln \Gamma_k - \Gamma_k^i \right) \qquad \text{2-80}$$

- v_{ki} refers to the number of interaction groups of type k in molecule i

- Γ represents the single function of the group composition for both the solution Γ and the molecular liquid standard state Γ^i

The term Γ is assumed to have the same value for all molecular mixtures which have the same group composition. "Γ can be considered the 'group activity coefficients' referred to the hypothetical standard pure groups"(Derr & Deal, 1969). This function also meets the Gibbs-Duhem criterion for the 'groups' since the activity for the molecules also satisfy this criterion.

The ASOG method proposes a definite analytical form for the Γ function. This is the distinction between ASOG and the solution of groups method. Solution of groups utilized binary data to obtain the group fraction dependence of the function Γ. This restricted its use since mixtures with only two groups could be handled. The analytical expression for this function allows for versatility to treat binary and multicomponent systems and is capable of handling many kinds of groups. Derr and Deal (1969) used Wilson as a basis for defining Γ since it related to excess free energy trends. Using the Wilson equation for group activity coefficients, the following equation is obtained for a group, k in a mixture comprising N groups:

$$\ln \Gamma_k = -\ln \sum_{i=1}^{N} X_i a_{ki} + 0.434 \left[1 - \sum_{i=1}^{N} \frac{X_i a_{ik}}{\sum_{m=1}^{} X_m a_{im}} \right] \qquad \text{2-81}$$

- a is the binary group parameter similar to binary model parameter (Note $a_{kk}=a_{ii}=a_{mm}=1$)
- X refers to the group fraction

The group fraction for group k is the ratio of the total number of groups of k to the total number of all types of groups in the mixture.

$$X_k = \frac{\sum_i x_i v_{ki}}{\sum_i x_i \sum_k v_{ki}}$$ 2-82

- x refers to the mole fraction

If one considers a solution comprising two groups (1 and 2) then **Equation 2-81** reduces to:

$$\ln \Gamma_1 = -\ln(X_1 + X_2 a_{12}) + 0.434\left[1 - \frac{X_1 \times 1}{(X_1 \times 1) + X_2 a_{12}} - \frac{X_2 a_{21}}{(X_2 \times 1) + X_1 a_{21}}\right]$$ 2-83

$$\ln \Gamma_2 = -\ln(X_2 + X_1 a_{21}) + 0.434\left[1 - \frac{X_2 \times 1}{(X_2 \times 1) + X_1 a_{21}} - \frac{X_1 a_{12}}{(X_1 \times 1) + X_2 a_{12}}\right]$$ 2-84

The following steps are required for use of ASOG:

1) Define the groups to be dealt with and their respective group numbers

2) Calculate group parameters from experimental data

3) Calculate activity coefficient for new mixtures

Typically a mixture is split into easily discernable chemical groups that have unique interactions. Consider the example of the acetone molecule. It could be fragmented into two methyl groups and one carbonylic interaction group with three size groups. The selection of the groups can be arbitrary and the total number of size groups does not have to correspond to the total number of interaction groups. In this case it would be possible to consider acetone as having two methyl with one carbonylic interaction groups (**Figure 2-10**) with four size groups or two methyl, one carbonylic carbon with one carbonylic oxygen interaction groups (**Figure 2-11**) with four size groups. Thus ASOG demonstrates flexibility in its application. However caution and logic should be exercised when determining the groups. This should be based on plausible interactions and molecule size. Group-pair parameters are obtained from activity coefficients of systems. This creates a matrix of group-pairs. These parameters can then be used in the calculation of activity coefficients in other new systems.

Obtaining a unique set of parameters can be problematic. Extrapolations of group parameters across large temperature ranges are not advisable.
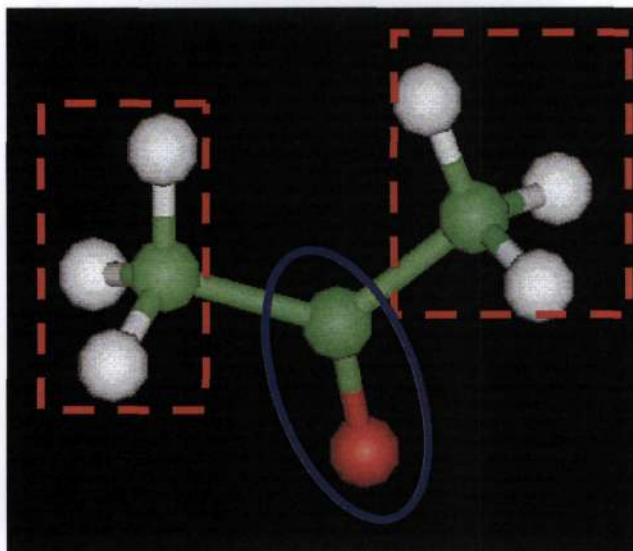
Figure 2-10 Acetone molecule with two methyl (highlighted in red) interaction groups and one carbonylic interaction group (highlighted in blue) (www.worldofmolecules.com)
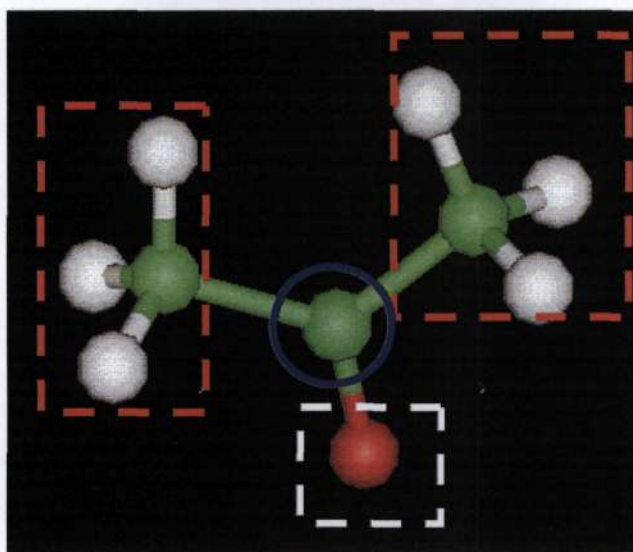
Figure 2-11 Acetone molecule with two methyl (highlighted in red) interaction groups, one carbonylic carbon interaction group (highlighted in blue) and one carbonylic interaction oxygen group (highlighted in white) (www.worldofmolecules.com)

2.11.2 UNIFAC

The UNIFAC method (Universal Functional Activity coefficient) uses the UNIQUAC model as its basis. It was developed by Fredenslund, Jones and Prausnitz (1975).
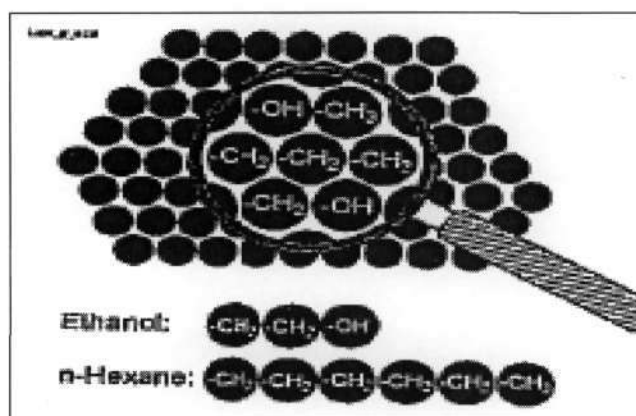
Figure 2-12 Depiction of how molecules are fragmented into structural groups

The UNIFAC method utilizes the solution of groups theory, whereby a system is regarded as a mixture of sub-molecular groups, e.g. CH_3--, $--OH$, $--COOH$ etc. rather than a mixture of molecules. The activity coefficients of the molecules are calculated from the activities of these groups. The interaction parameters required to find the group activities are regressed using a large database. These interaction parameters are independent of the molecules of which the groups form part.

The sub-molecular groups are divided into certain main groups. Interactions are only determined between the main groups. The advantage of this method is that the number of possible structural groups is much less than the number of different components comprising the entire molecule.

$$\ln \gamma_i^C = f(x_i, q_i, r_i) \qquad \text{2-85}$$

Relative Van der Waal's surface: $\qquad q_i = \sum_k v_k^{(i)} Q_k \qquad \text{2-86}$

Relative Van der Waal's volume: $r_i = \sum_k v_k^{(i)} R_k$ **2-87**

$$\ln \gamma_i^R = \sum_k v_k^{(i)} \left(\ln \Gamma_k - \ln \Gamma_k^{(i)} \right) \quad \textbf{2-88}$$

$$\Gamma_k = f\left(x_i, q_i, a_{nm}\right) \quad \textbf{2-89}$$

Group interaction parameters, a_{nm} are obtained from the regression of experimental VLE data.

There have been several modifications to UNIFAC. These include:

- Original UNIFAC
- Modified UNIFAC (Lyngby)
- Modified UNIFAC (Dortmund)
- Second-order UNIFAC
- UNIFAC-LL
- Polymer UNIFAC
- UNIFAC Free Volume

The modified UNIFAC (Dortmund) is maintained by Prof. Gmehling and co-workers. The parameters are fitted using the entire Dortmund Databank (DDB). Original UNIFAC lacked data for compounds of very different size. It had no qualitative information about $\gamma(T)$ and extrapolation to infinite dilution was not advisable.

Modified UNIFAC differs from original UNIFAC with regard to both the combinatorial and residual parts. Here the combinatorial part was modified to accurately account for systems that exhibit significant differences in molecular size. This gave improved results for asymmetric mixtures. The residual part was modified to include a better temperature dependence of the interaction parameters. This temperature dependence is crucial since UNIFAC interactions were usually fit to experimental VLE data close to 1 atm. This can produce high degree of error if extrapolation to higher/lower temperatures or pressures is required. In addition, modified

UNIFAC has a wider basis of data including γ^∞, H^E, Cp^E, SLE and LLE data. The excess enthalpies gave qualitative information about $\gamma(T)$. A combination of these modifications has made modified UNIFAC superior and more accurate than UNIFAC.

Modified UNIFAC uses the following objective function:

$$F = W_{VLE} \sum \Delta VLE + W_{AZD} \sum \Delta AZD + W_{H^E} \sum \Delta H^E + W_{Cp^E} \sum \Delta Cp^E + W_{\gamma^\infty} \sum \Delta \gamma^\infty + W_{LLE} \sum \Delta LLE + W_{SLE} \sum \Delta SLE$$

Data Type		Delivers the required information
VLE and azeotropic data		$\gamma = f(x)$
$H^E(Cp^E)$	Gibbs Helmhotz equation $$\frac{\partial \ln \gamma_i}{\partial(1/T)} = \frac{H_i^E}{R}$$	$\gamma = f(T)$ – support data at high temperature
γ^∞		The only reliable information in the dilute range and about asymmetric systems
SLE (Eutectic Systems)	$$\ln x_i^L \gamma_i^L = -\frac{\Delta h_{m,T_{m,i}}}{RT}\left(1 - \frac{T}{T_{m,i}}\right)$$	Supporting data at low temperature (T < 273 K)
LLE		Often the only information for strong real mixtures

Table 2-2 Data types used in the objective function and the type of information that they deliver
(Gmehling and Rarey, 2005)

The predictions obtained from UNIFAC are often very good provided there are interaction parameters available.
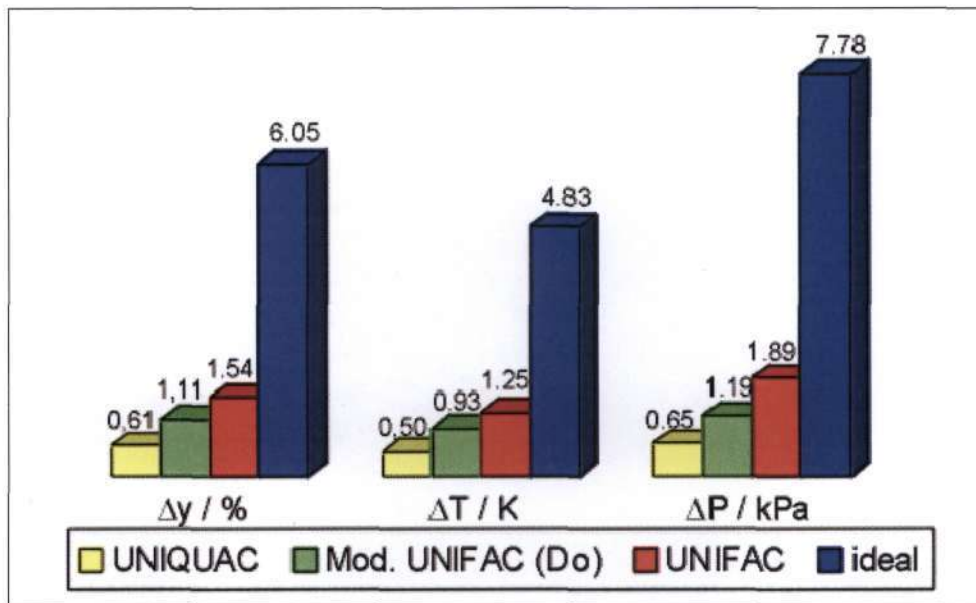
Figure 2-13 Relative deviations between experimental and predicted data for 3300 consistent VLE data sets (http://134.106.215.86/UNIFAC/)

Figure shows the relative deviation between the experimental and predicted (composition, temperature and pressure) for 3300 data sets as published by the UNIFAC consortium.

It can also be used to predict activity coefficients at infinite dilution. **Figure** shows the absolute and relative deviation for 12600 data points.

The UNIFAC and modified UNIFAC parameter matrices are being updated continuously. **Figure 2-15** illustrates the interaction matrix for modified UNIFAC.
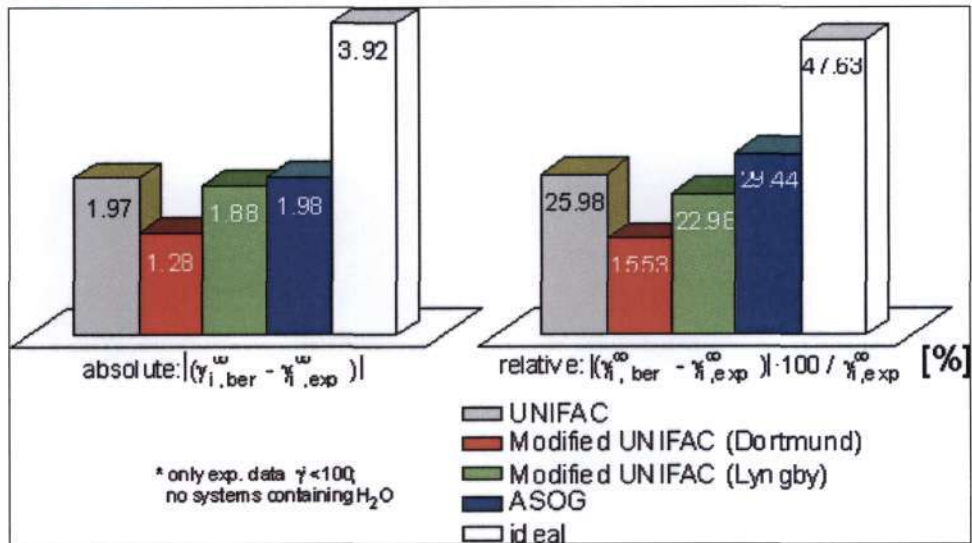
Figure 2-14 Infinite dilution activity coefficients prediction for comparison of various models

(http://134.106.215.86/UNIFAC/)
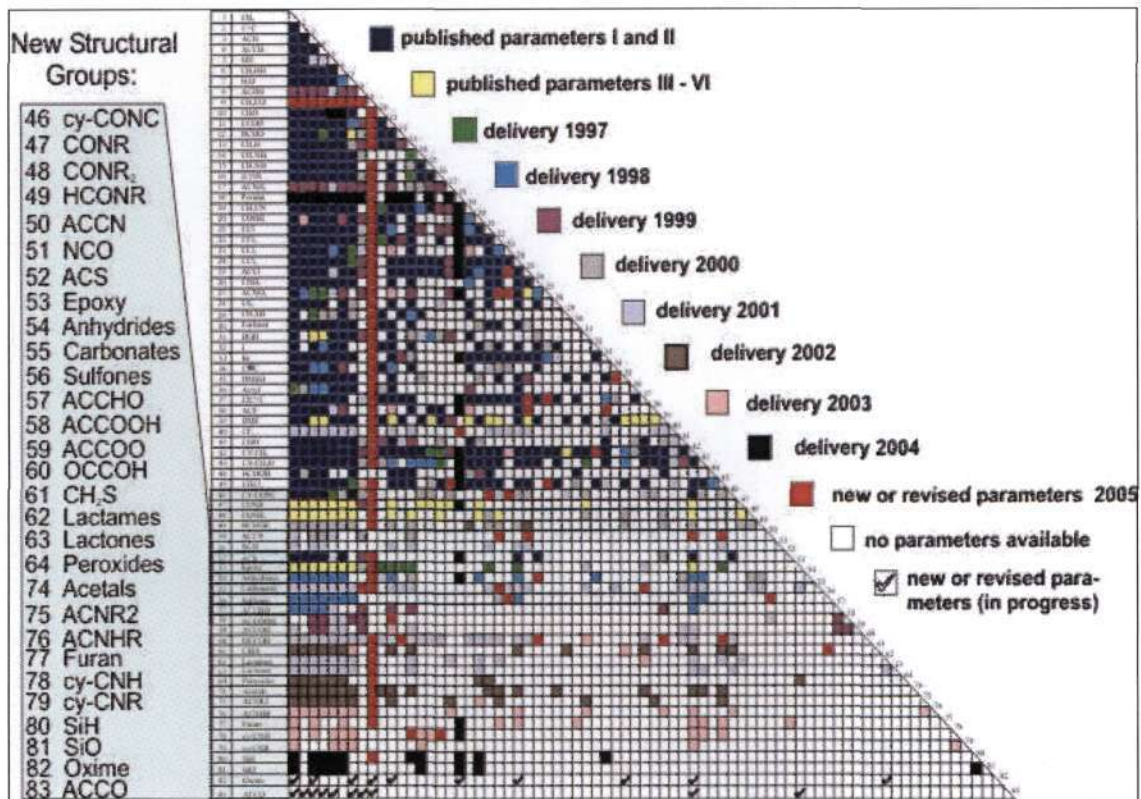
Figure 2-15 Modified UNIFAC (Dortmund) interaction matrix (http://134.106.215.86/UNIFAC/)

2.11.3 GEQUAC

Both UNIFAC and ASOG are based on the idea that a molecule interacts with its nearest neighbours. This concept was introduced by Guggenheim. However the use of local composition in semi-empirical models and group contribution methods are inconsistent. The selection of groups is arbitrary and hence different descriptions of the same molecule are obtained. With the aid of quantum mechanics, criteria have been established to determine physically consistent groups. This has been used with modified UNIFAC and has given better results. More exact approaches to Guggenheim's concept have produced, among others, the following models:

- TASQUAC (Kehiaian, Abusleme and Vera)
- GTASQUAC (Lacmann et al) – group contribution for multicomponent phase equilibrium of organic compounds
- DISQUAC (Kehiaian et al) – group surface version of Guggenheim-Barker quasi chemical lattice theory

Existing models do not adequately describe associating and non-associating mixtures. This can be attributed to the polar and hydrogen bonding interactions that occur between two positions of a functional group, the two poles of the dipole and the acceptor and donor site of the hydrogen bond. The GEQUAC model was derived with the intention of describing these interactions (Egner, Gaube & Pfennig, 1997).

$$G^E = G^E_{comb} + G^E_{int} \qquad 2\text{-}90$$

GEQUAC has two parts. The combinatorial part, G^E_{comb} is the same as the Guggenheim-Stavermann term which represents the entropy change of mixing attributed to the differences in molecular size and shape. G^E_{int} represents the interaction strength, influence of interaction, steric effects on mutual orientation and deviations from random mixing.

The model contains three adjustable parameters to describe mixtures. The intensity of interaction is described in terms of free energy.

$$\omega_{ji}^{G} = \omega_{ji}^{H} - T\omega_{ji}^{S} \qquad \text{2-91}$$

- ω_{ji}^{H} refers to the exchange-enthalpy parameter between the surfaces of groups i and j

- ω_{ji}^{S} refers to the exchange-entropy parameter between the surfaces of groups i and j

$$\omega_{ji}^{H} = 2\varepsilon_{ji} - \varepsilon_{ii} - \varepsilon_{jj} \qquad \text{2-92}$$

- ε refers to the interaction energy

The surface area of the site participating in the interaction is defined as:

$$q_i = \frac{A_i}{A_{ref}} \qquad \text{2-93}$$

- A_i refers to the surface area of site i
- $A_{ref}= 2.5 \times 10^9$ cm^2 mol^{-1}, the surface area of reference particle as defined for the UNIQUAC model

A molecule of component k, comprises M_k groups.

$$q_k = q_{VDW} = \sum_{i=1}^{M_k} q_i \qquad \text{2-94}$$

The volume parameter r, is obtained from normalization with the volume of a standard segment $V_{ref}= 15.17$ cm^3 mol^{-1} according to UNIQUAC.

The number of model parameters is determined by the fragmentation of the molecular structure. This is dependent on the type of mixture. If one considers a ketone-alkane or alcohol-alkane mixture, then four different surface areas can be defined (**Figure 2-16**). The alkane molecule can be considered as a weak interaction surface (4). Polar or hydrogen bonded molecules comprise two strong interacting surfaces areas 1 and 2, with the rest

of the molecule characterized by a weak interacting surface, 3. With this basis, 14 model parameters are considered.

The surface parameter for the alkane would correspond to that of the Van der Waals surface. Surface 3 of the strongly interacting component is defined as the difference between Van der Waal's surface of the entire molecule and the adjustable surface area of the two poles.

$$q_3 = q_{VDW} - q_1 - q_2 \qquad\qquad \text{2-95}$$
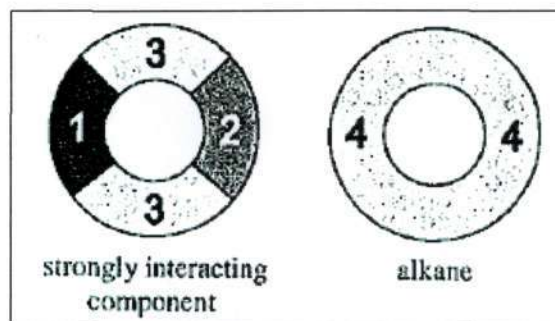
Figure 2-16Fragmentation of molecular surface for ketone-alkane and alcohol-alkane mixtures for the GEQUAC model (Ehlker & Pfennig, 2002)

Surface Sites	Surface parameter	
1	q_1	
2	q_2	
Group contact	**Exchange parameter**	
	Enthalpic	**Entropic**
1-2	ω_{12}^H	ω_{12}^S
1-3	ω_{13}^H	ω_{13}^S
2-3	ω_{23}^H	ω_{23}^S
1-4	ω_{14}^H	ω_{14}^S
2-4	ω_{24}^H	ω_{24}^S
3-4	ω_{34}^H	ω_{34}^S

Table 2-3 Model parameters for ketone-alkane and alcohol-alkane mixtures required for the GEQUAC model

Interaction energies for the weakly interacting surfaces, 3 can be considered small, thus ω_{34}^H is assumed to be negligible (ω_{jj}^H and ω_{ii}^H are zero by definition). Thirteen model parameters have to be determined and this requires a systematic approach to obtain a physically significant parameter set.

The objective function is defined as:

$$F\left(q_i, \omega_{ji}^H \omega_{ji}^S\right) = \sum \Delta VLE + \sum \Delta H^E \qquad \text{2-96}$$

- $$\Delta VLE_1 = \sum_{k-1}^{N}\left(\frac{P_k^{exp} - P_k^{GEQUAC}}{P_k^{exp}}\right)^2$$

- $$\Delta VLE_2 = \sum_{k-1}^{N}\left(\frac{T_k^{exp} - T_k^{GEQUAC}}{T_k^{exp}}\right)^2$$

- $$\Delta VLE_3 = \sum_{k-1}^{N}\left(\frac{y_k^{exp} - y_k^{GEQUAC}}{y_k^{exp}}\right)^2$$

- $$\Delta H^E = \sum_{k-1}^{N}\left(\frac{H_k^{E,exp} - H_k^{E,GEQUAC}}{H_k^{E,exp}}\right)^2$$

Since all parameters cannot be fitted simultaneously to all the binary data, priority has to be given to the surface parameters q_1 and q_2 of the strong interacting surface sites and the exchange enthalpy, ω_{12}^H between the surfaces of these 2 poles. q_1 and q_2 were varied in certain limits and objective function was evaluated.

Binary mixtures in a homologous series were investigated. The atomic charge of single ketone and alcohol molecules were investigated and it was found that the charge distribution was the same within a homologous series of ketones and alcohols. Slight deviations were observed for isomers. Hence all surface parameters of poles q_1 and q_2 are constant for all components of the homologous series of alcohols and ketones. ω_{ji}^H is the same for all binary mixtures within a series since they correspond to the same polar sites for each mixture. The components in each

series only differ in exchange entropy parameters. ω_{12}^S and ω_{13}^S refer to the strongly interacting components for different components in the homologous series and ω_{34}^S refer to the different components in the homologous series of alkanes. ω_{23}^S, ω_{14}^S and ω_{24}^S are set to a constant value. $\omega_{12}^H, \omega_{13}^H, \omega_{23}^H, \omega_{14}^H, \omega_{24}^H$ and surface parameters q_1 and q_2 are the adjustable parameters that have the same values for all binary mixtures in a homologous series.

Excellent results were obtained for the binary ketone-alkane and alcohol-alkane systems. GEQUAC describes the H^E data well. Data was also extrapolated to predict LLE, but the description is not very accurate.
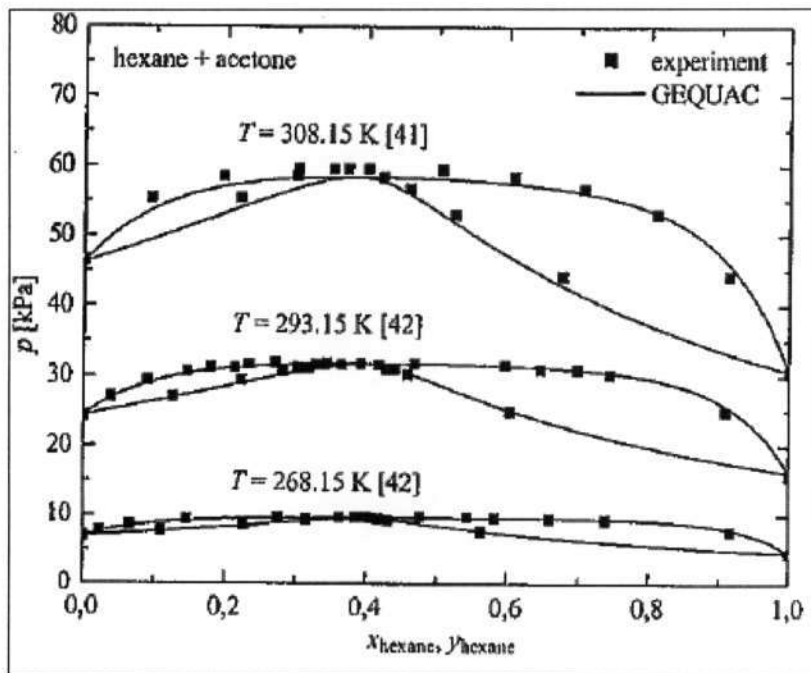
Figure 2-17 Hexane-acetone system as fitted by the GEQUAC model (Ehlker & Pfennig, 2002)
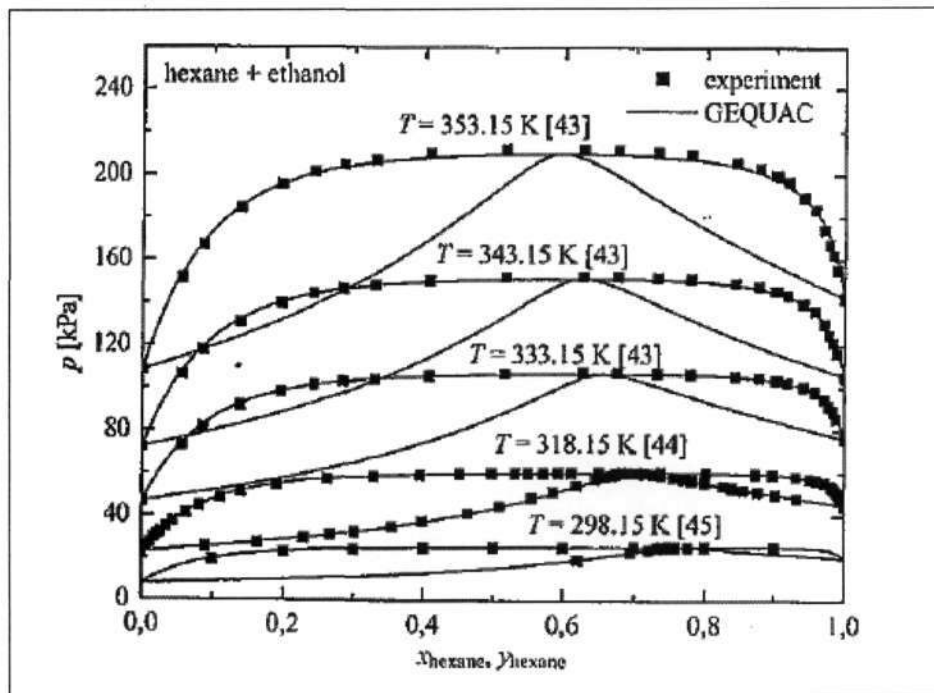
Figure 2-18 Hexane-ethanol system as fitted by the GEQUAC model (Ehlker & Pfennig, 2002)
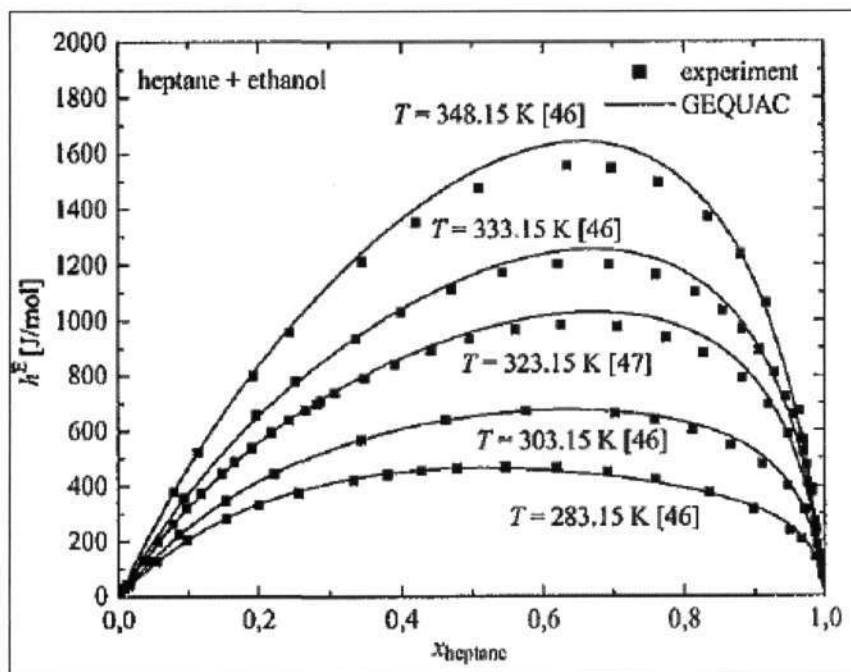
Figure 2-19 Heptane-ethanol H^E data as fitted by the GEQUAC model (Ehlker & Pfennig, 2002)
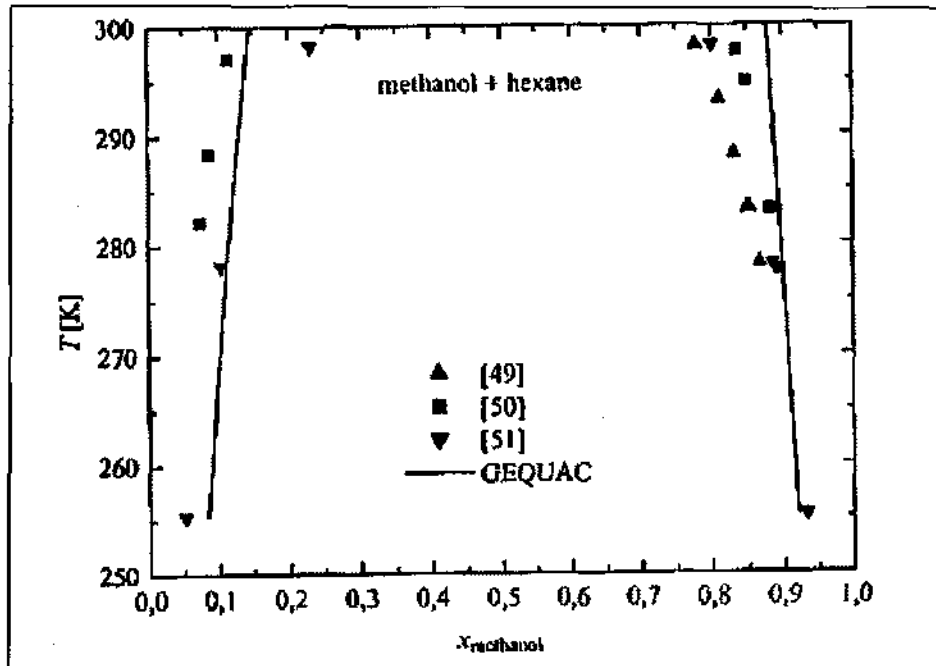
Figure 2-20 Prediction of LLE for methanol-hexane system (Ehlker & Pfennig, 2002)

While GEQUAC achieves good results, the model requires many parameters. GEQUAC has a plausible theoretical basis since it solves a quasichemical equation. The model is mathematically complex.

Chapter 3 Literature Review

3.1 Introduction

The chemical industry is an ever expanding industry with new components being synthesized to cater for varying industrial applications. Researchers have no doubt invested an immense amount of time, money and effort over decades to measure equilibrium data.

The results of years of arduous experimental work have been stored in computerized data banks e.g. the Dortmund Data Bank (DDB). The DDB was developed by J. Gmehling and co-workers. The current status of the DDB is:

- Vapour-liquid Equilibrium (VLE) Data - 52180 Isothermal/Isobaric Data sets
- γ^∞ - Infinite Dilution Activity Coefficients - 44750 Data points for pure solvents and mixtures
- H^E - Excess Enthalpy - 17400 Data sets
- Azeotropic Data - 14720 Data sets
- Liquid-liquid Equilibrium (LLE) Data - 14720 Data sets

The DDB has currently 52000 references from 1800 journals and for 18900 compounds. This helps the reader to further appreciate the rate at which the amount of physical property information is growing annually.

The experimental setup and measurement of equilibrium data has become a well established science over the years. However, it is still an expensive and complex undertaking depending on the nature of the chemical components. The cost of the measurement is also a further constraint to be factored in when considering the setup and operation of equipment. Equipment cost also varies depending on the operating conditions with the cost of high pressure equipment far outranking the cost of measuring low pressure equilibrium data. Multi-component data measurements are even more complex and the results are of questionable quality. The number of possible chemical combinations and permutations is often very large and thus one is limited by the availability of data.

Equilibrium data are a prerequisite for the design, synthesis and operation of separation equipment. Thus the design phase is dependent on the availability of equilibrium data. For cases where experimental data is not readily available, predictive methods become more crucial. In particular for multi-component systems, equilibrium data is predicted from binary interaction parameters obtained from experimental binary data.
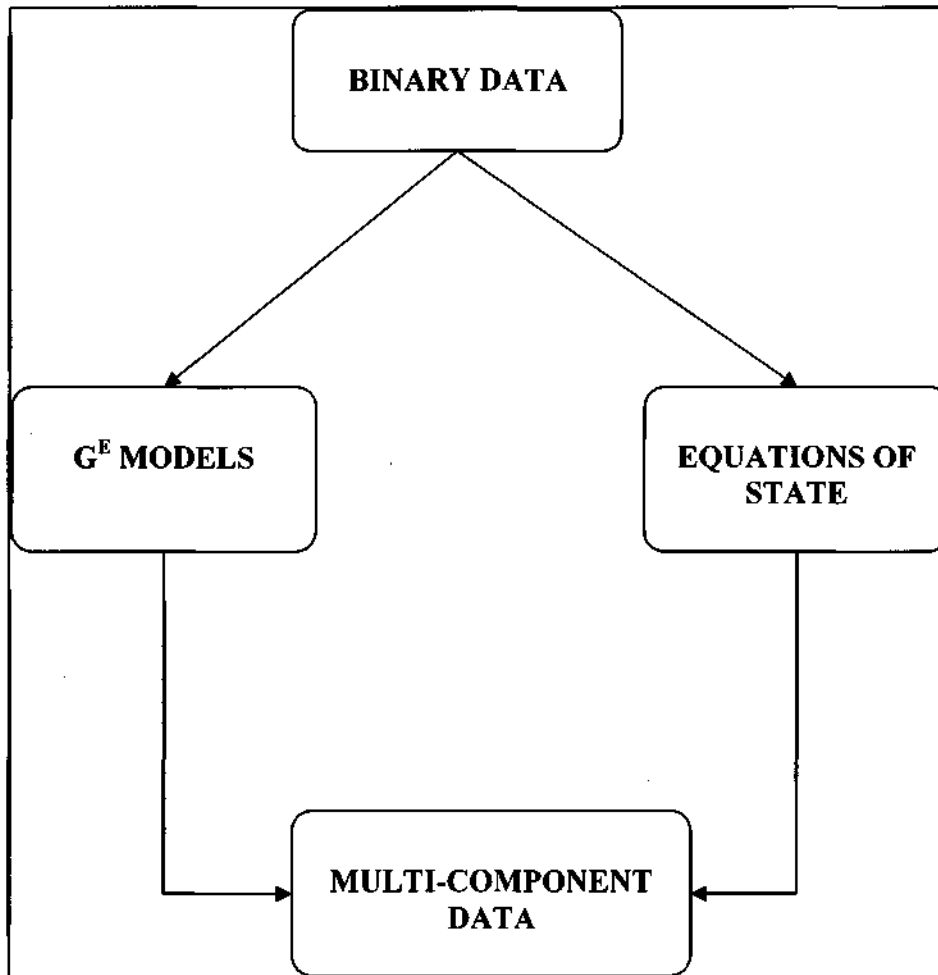
Figure 3-1: The link between binary and multi-component data

G^E-models and equations of state provide the link to predicting multi-component mixture behaviour. Inherent in the models are parameters derived from regression of experimental data. It follows that the parameters can only be used within a certain range of confidence dictated by the temperature and pressure of the data. This limits the use of the parameters to extrapolate data and impacts on the margin of error introduced. Further, this limits the extension of the use of the binary interaction parameters to predict multi-component data within a certain tolerable window of error. The quality of the binary data also determines the quality of the binary parameters and the confidence levels in predicting data. **Table 3.1** illustrates the availability of some of the more common G^E-models in commercially available process simulators. It also indicates whether the

model is capable of correlating and predicting liquid-liquid equilibria and multi-component phase behaviour.

MODEL	Aspen	HYSIM	PROII	Liquid–liquid equilibria	Multi-component Systems
Margules	-	X	-	Yes	No
Van Laar	X	X	X	Yes	No
Wilson	X	X	X	Yes	Yes
NRTL	X	X	X	Yes	Yes
UNIQUAC	X	X	X	Yes	Yes

Table 3.1 Implementation of excess Gibbs energy models in process simulators

This chapter carefully analyses the use of G^E-models in the prediction of equilibrium data. It highlights both their advantages and their disadvantages to help the reader understand their capabilities. The disadvantages have helped researchers to isolate the problem with current models i.e. the concentration dependence. The latter half of this chapter addresses methods employed by thermodynamicists, to improve G^E-models.

3.2 Advantage of Models

G^E-models and equations of state are tools for correlating and predicting the real behaviour of mixtures e.g. phase equilibrium. Current local composition G^E-models (Wilson, NRTL and UNIQUAC) boast the following features, which make them practical for engineering purposes:

- applicability to highly non-ideal mixtures

o G^E-models have evolved over time and have become able to correlate more non ideal systems. Simpler models like Margules were only able to handle symmetric systems. This deficiency prompted the development of models that were able to handle more realistic asymmetric chemical mixtures. Models gradually evolved to contain some theoretical basis which improved their predictive ability (local composition models – Wilson, NRTL and UNIQUAC). The Wilson equation showed distinct improvements for mixtures containing non-polar and polar components but was limited due to its inability to predict liquid immiscibility. NRTL and UNIQUAC were later developed and had the ability to predict liquid immiscibility in addition to highly non-ideal systems.

- extension to multi-component mixtures from binary data

  o Simpler models were not able to be extended to multi-component mixtures (Margules, Van Laar). The binary interaction parameters derived using the Wilson, NRTL and UNIQUAC models can be used for multi-component systems with considerable accuracy.

- improved extrapolation

  o Models like Margules and Van Laar lack a temperature dependency in their interaction parameters and hence cannot be used to extrapolate data to different temperatures. The local composition models have temperature dependency in their interaction parameters in the form of a Boltzmann term. These models can be used with some confidence to extrapolate data to different temperatures.

- mathematical simplicity and numerical stability

3.3 Disadvantages

3.3.1 Weak Theoretical Basis

G^E-models are developed either on an empirical, semi-empirical or theoretical basis. Models have evolved over time and are now more effective in describing equilibrium behaviour by accounting for different sizes of molecules and interactions between molecules. This is possible due to the

model's theoretical basis. If one improves this theoretical basis, then the model should theoretically become more widely applicable.

Often a better theoretical basis makes a model more unstable. As one accounts for more deviations from ideality in a model, the model rapidly steps up in complexity and the required computation time increases drastically. The model could become numerically unstable and parameter fitting may lead to multiple solutions. Intensive experimentation is required to quantify factors accounting for deviations from ideal behaviour. From a global perspective this is not a prudent solution to the problem. Models should ideally be fairly simple and practical for engineering purposes since this then serves to provide fair estimates or starting points for equipment design. In addition, the research and experimentation required to improve the model could be costly, complex and a time consuming process.

3.3.2 Poor Model Performance due to Few Parameters

Models have a varying number of parameters based on the complexity of the model. The predictive ability of the parameters is dependent on the quality of data and the ability of the model to adequately describe the mixture. A model can be modified differently by the introduction of parameters or by modifying the model equation itself.

Study of phase equilibrium data has revealed that pure component parameters can improve the description of almost all mixtures of practical interest since many of these of mixtures exhibit differences in both size and chemical nature.

Local composition models have been developed with this crucial idea as its focus. By including these pure component parameters the model's fitting capability is increased. The Wilson model utilises pure liquid molar volumes. The UNIQUAC model uses the r and q values which are representative of volume and surface areas.

3.3.3 Inability to Simultaneously Describe VLE and LLE

Although considerable time has been invested in developing activity coefficient models, no model to date can boast the ability to simultaneously describe VLE and LLE data. The correlation of VLE data from LLE derived parameters is very inaccurate. This is clearly illustrated in **Figure 3-2a-b.**
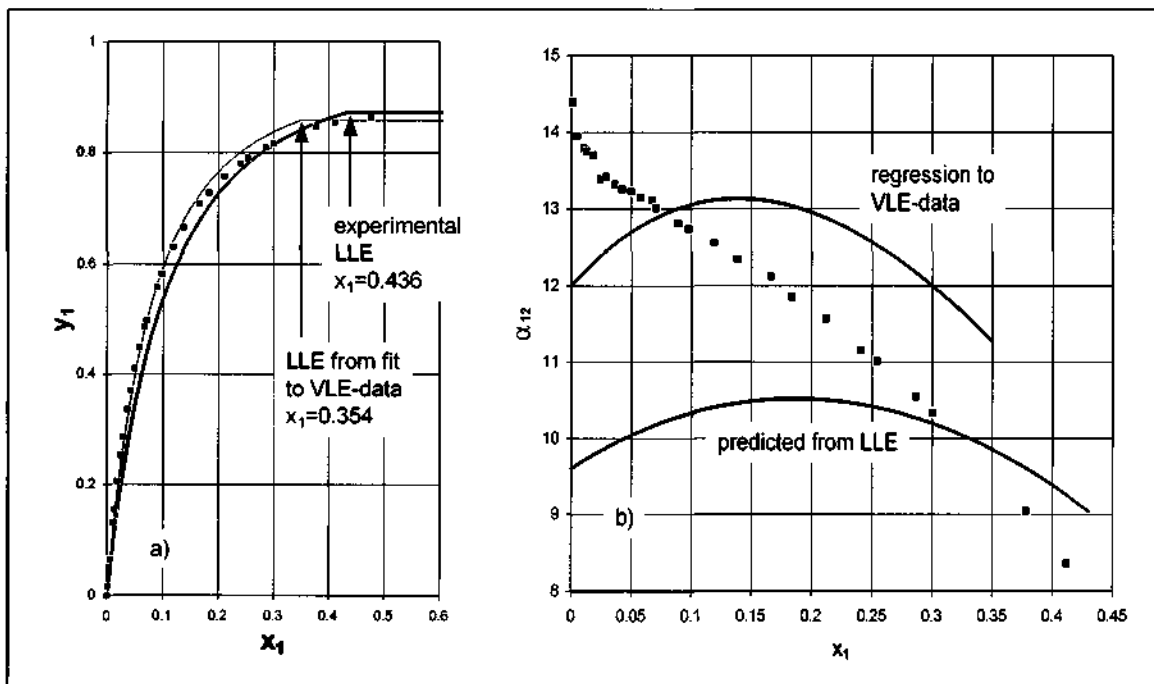
Figure 3-2 a) x-y data - b) Separation factor, α_{12} - calculated using UNIQUAC G^E-model parameters from LLE and from the regression of VLE-data for the system Water (1) - 1-Pentanol(2) (Cho, et al 1984) at 101.3 kPa together with experimental xyP-data (Rarey, 2005)

Figure 3-2a shows x-y data for the system Water (1)-1-Pentanol (2). The curve predicted from the LLE is not steep enough in the homogeneous region. In the case of the regression of the VLE data, the immiscible region is over estimated. In **Figure 3-2b** the separation factor, α_{12} is calculated from LLE parameters and fit by regression of VLE data. It is clear that the prediction of this parameter is unsatisfactory in both cases.
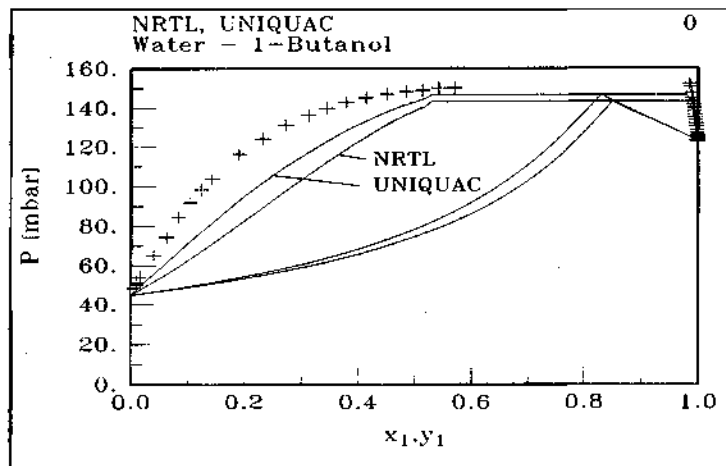
Figure 3-3 Prediction of VLE data (+ VLE data - Fischer, 1991) using NRTL and UNIQUAC G^E-model parameters from LLE at 50°C for Water (1)-1-Butanol

Figure 3-3 shows the correlation of VLE data from calculation using LLE parameters from local composition models NRTL and UNIQUAC. Both models fail to adequately describe the left hand side homogeneous regions. The curves generated using both models are clearly not steep enough in the homogeneous region and the result is an inaccurate description of the data.

These disadvantages help highlight the problematic areas of the models proposed thus far.

3.4 Improving the Concentration Dependence of G^E- Models

3.4.1 Introduction

The advantages and disadvantages presented above provide a holistic perspective of the capabilities of current models. It also highlights the fact that the problem lies in the concentration dependence of G^E-models and if one can propose plausible modifications while still retaining the model's current capabilities, then the model would truly be enhanced. This section reviews some of the methods employed by researchers to solve this problem, in particular the problem of simultaneously describing VLE and LLE.

Some researchers have invested time into exploring methods to improve the concentration dependence in G^E-models. The available methods that were explored by various researchers are very limited. Those methods that could be sourced from literature are discussed in this section.

3.4.2 Method 1: Combining Two Models

3.4.2.1 Introduction

A more flexible G^E-model could be obtained by combining two models, where both differ in their concentration dependence (Rarey, 2005). This newly formed G^E-model would have superior capabilities since the constituent models would compliment each other. In other words, the combined models would be able to correlate data in cases where the both models fail individually. The new model should then be able to simultaneously correlate VLE and LLE data provided the two models have different concentration dependences (Rarey, 2005).

3.4.2.2 Combination of Modified Wilson and Redlich Kister Equation

In support of the concept introduced above, an illustrative example is presented here. LLE data and excess enthalpies for two binary systems were correlated utilising a combined G^E-model. The systems investigated were methylcyclohexane (1)-methanol (2) and methylcyclohexane (1)-n, n-dimethylformamide (2) (Bendova et al. 2003). Data points for the binodal curves and tie-lines were determined for both systems and correlated. The LLE and excess enthalpy data was also predicted by the modified UNIFAC method for the purpose of comparison.

To correlate the data the Wilson and Redlich Kister equations combined. The newly formed Modified Wilson equation was then evaluated to assess its ability to correlate data.

$$Q = \left(\frac{G^E}{RT}\right)_{Wilson} + \left(\frac{G^E}{RT}\right)_{RedlichKister} \qquad \text{3-1}$$

$$\left(\frac{G^E}{RT}\right)_{Wilson} = -\sum_{i=1}^{N} x_i \ln \sum_{i=1}^{N} x_j \left(\frac{V_j}{V_i}\right) \exp\left(\frac{-a_{ij}}{T^*}\right) \qquad \text{3-2}$$

$$\left(\frac{G^E}{RT}\right)_{RedlichKister} = -\sum_{i=1}^{N}\sum_{j)i}^{N} x_i x_j \sum^{n_{ij}} B_{ijk}\left(x_i - x_j\right)^{k-1} \qquad 3\text{-}3$$

$$a_{ij} = \alpha_{ij} + \beta_{ij}T^* + \left(\frac{\gamma_{ij}}{T^*}\right) \qquad 3\text{-}4$$

$$B_{ijk} = \alpha_{ijk} + \beta_{ijk}T^* + \left(\frac{\gamma_{ijk}}{T^*}\right) \qquad 3\text{-}5$$

- $Q = \dfrac{G^E}{RT}$ - Dimensionless excess Gibbs energy

- α_{ij}, β_{ij}, γ_{ij}, α_{ijk}, β_{ijk} and γ_{ijk} – Temperature independent binary parameters

- V_i – molar volume

- x_i – mole fraction

- $T^* = \dfrac{T}{T_{ref}}$

- $T_{ref} = 300$ K

In the Redlich-Kister equation, the number of parameters, n_{ij} is dependent on the system to be correlated. T^* was introduced in this combined model as a means to improve numerical stability. **Equations 3-4 and 3-5** show the temperature dependence of the model parameters.

In this case, the correlation of data was achieved by use of the maximum-likelihood method. This method was appropriately chosen since it facilitates the correlation of various types of data

simultaneously. Specifically, the excess enthalpy data for the heterogeneous region could be included in the simultaneous correlation.

3.4.2.3 Evaluation of Combined Model Equation
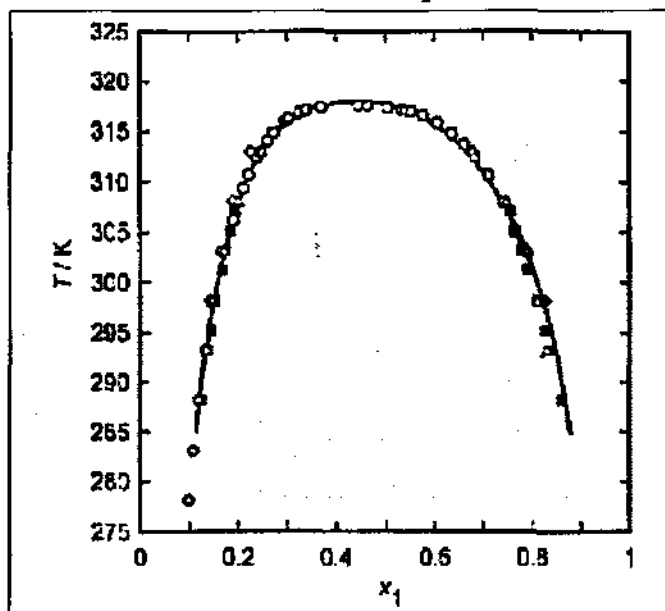
Figure 3-4 Liquid-liquid equilibrium for methylcyclohexane (1) and methanol (2) — Modified Wilson equation (Bendova et al, 2003)
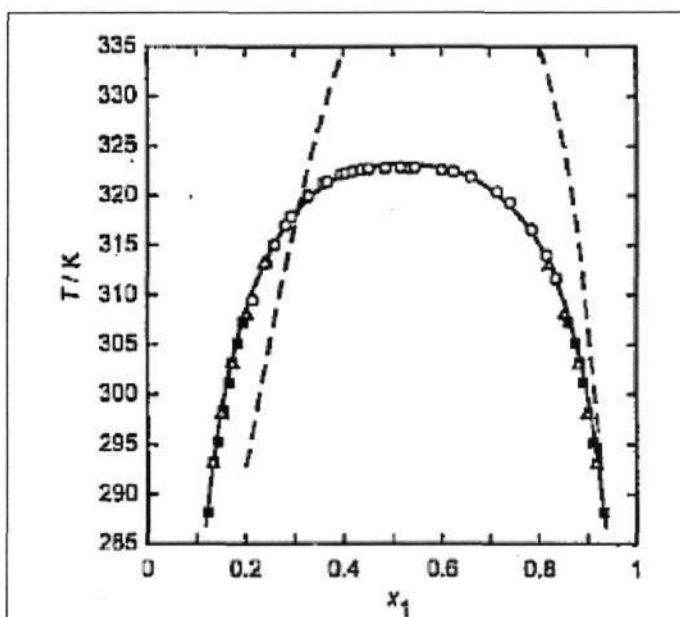
**Figure 3-5 Liquid-liquid equilibrium for methylcyclohexane (1) and n, n-dimethylformamide (2), ---
Modified Wilson equation; - - - Modified UNIFAC method (Bendova et al, 2003)**

Some literature values for the system methylcyclohexane-methanol were available and these were used in conjunction with measured LLE data for correlation purposes.

The modified Wilson equation fits the LLE data for both systems with excellent accuracy (**Figure 3-4 & Figure 3-5**). The prediction obtained from the UNIFAC method for the system methylcyclohexane (1) - n, n-dimethylformamide (2) is shown in **Figure 3-5**. UNIFAC over estimates the heterogeneous region when compared to the experimental data.

The correlation of excess enthalpy in the system methylcyclohexane (1) - n, n-dimethylformamide (2) (**Figure 3-7**) by the modified Wilson equation was excellent in comparison to the experimental data whilst that for the system methylcyclohexane (1) - methanol (2) (**Figure 3-6**) was satisfactory. UNIFAC is once again used to predict the excess enthalpy for both systems. The prediction obtained for the methylcyclohexane (1) - n, n-dimethylformamide (2) can be considered satisfactory. In the case of methylcyclohexane (1) - methanol (2), UNIFAC inaccurately predicts that the system is homogeneous in the entire concentration range at both temperatures, 293.15K and 398.15K.

The use of the modified Wilson equation in conjunction with the maximum-likelihood principle gave a very good correlation with the experimental data.

Advantage - Good description of both the LLE data and excess enthalpy using a combined model and maximum-likelihood principle.

Disadvantage – 7-8 Interaction parameters were required by the combined model equation to obtain a good correlation.

Disadvantage - Have to select constituent models with some knowledge of the systems being correlated to ensure optimal fitting of data by the combined model.
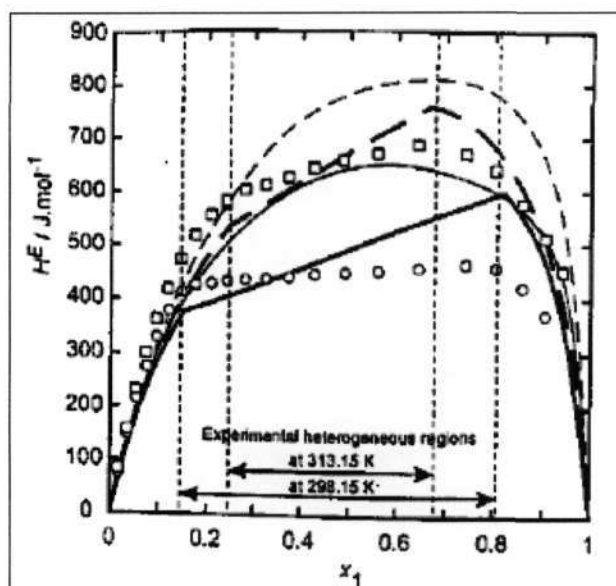
Figure 3-6 Excess enthalpy for methylcyclohexane(1) and methanol(2), ○, 298.15 K; □; 313.15K; (thin lines – the modified UNIFAC method, bold lines --- the modified Wilson equation) (Bendova et al, 2003)
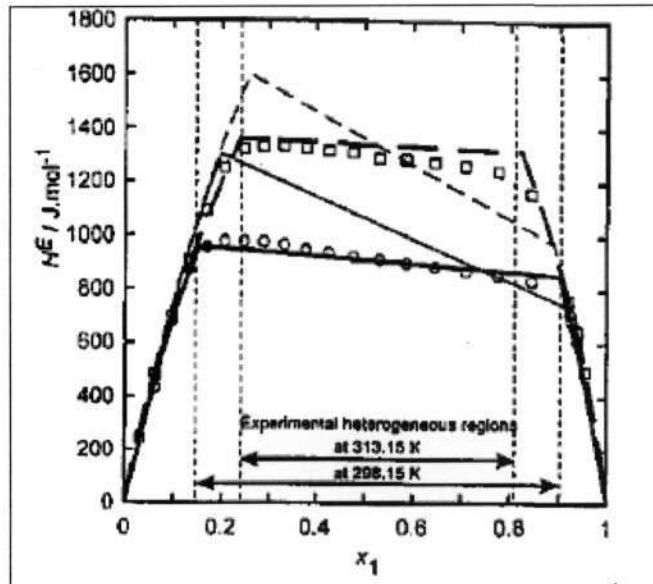
Figure 3-7 Excess enthalpy for methylcyclohexane(1) and n,n-dimethylformamide(2), ○, 298.15 K; □;

313.15K; (thin lines – the modified UNIFAC method, bold lines --- the modified Wilson equation)

(Bendova et al, 2003)

3.4.3 Use of Additional Parameters

3.4.3.1 Introduction

A lack of sufficient parameters is a major disadvantage that prevents models from adequately describing phase data. The use of additional parameter can drastically improve the fitting capability of a model but care should be taken that it does not introduce any intercorrelation between existing parameters. The parameter should also not introduce undue complexity with regards to use of the model.

3.4.3.2 Multiplication by a Concentration Dependent Factor

The work of Prausnitz and Cha (1985) addresses one aspect of the simultaneous representation of VLE and LLE. Their work is confined to the study of ternary systems, with the objective of improving the prediction of ternary LLE from binary data. This concept has great practical

potential in the design phase and operation of distillation equipment where liquid-liquid equilibrium is encountered in addition to VLE.

The method proposed is applicable to Type 1 ternary systems, i.e. a system of two miscible binaries and one partially miscible binary composed of non-electrolyte fluids. The method modifies the liquid phase non-ideality by modifying the excess Gibbs energy function and hence the activity coefficient expressions.

In predicting ternary LLE, the data of the three constituent binaries are required. Typically, models like UNIQUAC and NRTL are extended to ternary systems to obtain an expression for G^E as a function of mole fractions and binary interaction parameters. The set of parameters obtained from binary data is dependent on the quality of the experimental data that is regressed to fit G^E. Regression of binary VLE usually produces several optimal sets of binary parameters. To obtain a unique set of parameters, the experimental data has to be of high accuracy. The use of these parameters to extend to ternary systems in the case of ternary VLE is not a concern.

However, this is a concern in the case of ternary LLE. Ternary LLE exhibits more sensitivity to minor changes in molar excess Gibbs energy. Hence, changes in binary parameters have dramatic effects on ternary LLE. Coupled with the fact that binary VLE does not produce a unique set of binary parameters, one can conclude that ternary LLE cannot be predicted accurately from binary data alone.

Type 1 ternary LLE systems in particular cannot be predicted from binary data alone. Usually, the two phase region is over estimated. Prausnitz and Anderson (1978) attempted to rectify this shortcoming. Their study showed that a good description of ternary LLE data can be obtained from G^E model with binary parameters. However, the binary parameters must be obtained by utilizing both ternary LLE and binary data. Whilst the response for the prediction of LLE was distinctly better, the quality of VLE prediction was consequently reduced.

The work of Cha et al. (1985) examines the use of a correction factor that will enable the simultaneous correlation of both ternary LLE and binary VLE data with good accuracy.

3.4.3.3 Correction Factor

The regression of experimental binary data yields binary interaction parameters which are used to predict phase equilibrium data. In the case of ternary phase equilibria, there are 3 binary pairs which yield 3 sets of binary interaction parameters. Ternary equilibrium can then be described by these binary interaction parameters via G^E models written in terms of mole fractions which is

defined in this study as $\left(\dfrac{G^E}{RT}\right)^o_{123}$

Prausnitz proposes that $\left(\dfrac{G^E}{RT}\right)^o_{123}$ at constant temperature be multiplied by a factor C. C is empirical in nature and could be obtained from ternary LLE data. The correction factor has the following properties:

- C = 1 for binary data sets
- Dependent on composition

C is defined in such a manner so as to have minimal effects on the ternary VLE and thus retain the ability of the derived parameters to accurately predict VLE.

$$C = \exp\left[\alpha x_1^{\lambda_1} x_2^{\lambda_2} x_3^{\lambda_3}\right]$$ **3-6**

- α is a constant.

Therefore the new expression for G^E is

$$\left(\frac{G^E}{RT}\right)_{123} = C * \left(\frac{G^E}{RT}\right)^o_{123}$$ **3-7**

λ_i is a constant but has to be greater than or equal to unity to prevent singularities in the activity coefficients.

$$\lambda_1 = \left(\frac{\overline{x_1}}{\overline{x_2}}\right)\beta \qquad \textbf{3-8}$$

$$\lambda_2 = \left(\frac{\overline{x_2}}{\overline{x_3}}\right)\beta \qquad \textbf{3-9}$$

$$\beta = \lambda_3 \qquad \textbf{3-10}$$

Here $\overline{x_i}$ refers to the ternary composition when the absolute value of ln C is a maximum. Hence four adjustable parameters exist ($\alpha, \beta, \overline{x_1}$ and $\overline{x_2}$) for **Equation 3-6**.

Assuming, $\overline{x_1} = \overline{x_2} = \overline{x_3} = \dfrac{1}{3}$ then, $\beta = \lambda_1 = \lambda_2 = \lambda_3$

$$C = \exp\left|\alpha(x_1 x_2 x_3)^\beta\right| \qquad \textbf{3-11}$$

From **Equation 3-6 & 3-7**, the resulting activity coefficient expression is:

$$\ln \gamma_i = C \ln \gamma_i^0 + C\left(\frac{\lambda_i}{x_i} - \lambda_1 - \lambda_2 - \lambda_3\right) \ln C\left(\frac{G^E}{RT}\right)^0_{123} \qquad \textbf{3-12}$$

where $\ln \gamma_i^0 = \left[\dfrac{\partial\left(n_T \dfrac{G^E}{RT}\right)^0_{123}}{\partial n_i}\right]_{T,P,n_j}$

If the **Equation 3-11** is used instead then the activity coefficient expression reduces to

$$\ln \gamma_i = C \ln \gamma_i^0 + C\beta\left(\frac{1}{x_i} - 3\right) \ln C\left(\frac{G^E}{RT}\right)^0_{123} \qquad \textbf{3-13}$$

3.4.3.4 Evaluation of the Method

Nineteen ternary systems were investigated. The UNIQUAC equation was used to obtain the binary system interaction parameters for 16 ternary systems and the remaining 3 sets were regressed with NRTL. These parameters were then used to determine $\left(\dfrac{G^E}{RT}\right)^o_{123}$.

The ability to correlate data using the correction factor is dependent on the uncorrected model. Thus, the uncorrected model itself needs to be a fair approximation which requires the critical selection of binary interaction parameters. The binary data that is used should be in the same temperature range as the ternary system in question. In case of the partially miscible pair, parameters are obtained from mutual solubility data at the ternary system temperature. Here again, the accuracy of the mutual solubility data is crucial as minor adjustments in this data can dramatically improve the prediction of ternary data (Cha et al, 1985). It must be noted that if the uncorrected model results in an unsatisfactory description of the ternary data, then the correction factor will not improve the prediction of ternary data significantly since the binary parameters used for this system failed to give a fair description of the data.

Of the 19 ternary systems, 7 were calculated using **Equation 3-11** which is based on a simplifying assumption. **Equation 3-6** was used for the remaining 12 systems. It must be noted that the parameters determined in both these cases have no physical significance and are merely used to improve the correlation of the ternary LLE.

Figure 3-8 shows the correlation of 3 ternary systems where the binodal curves exhibit modest asymmetry. For these systems, **Equation 3-11** was used. Both cases are reflected here (with and without the correction). The case without correction shows the prediction obtained by using only binary data. It is evident that the description of the ternary data is unsatisfactory. By including the correction factor, C, the correlation of ternary data is excellent.

Figure 3-9 illustrates the effect of the correction factor as applied to 3 ternary systems in which the binodal curves exhibits strong asymmetry. Here, **Equation 3-6** was used for the calculation purposes. Once again, the use of the correction factor provides a superior correlation of ternary LLE data as compared to the prediction obtained from the binary data only.

Selectivity prediction also improved drastically (**Figure 3-10**). The predictive ability of the correction factor was further evaluated for the system cyclohexane, acetonitrile and benzene. Ternary parameters from 45° C are used to predict LLE at 25°C (**Figure 3-11**). In this case only the partially miscible pair's parameters were adjusted. The results achieved show good correlation of the system at both temperatures.
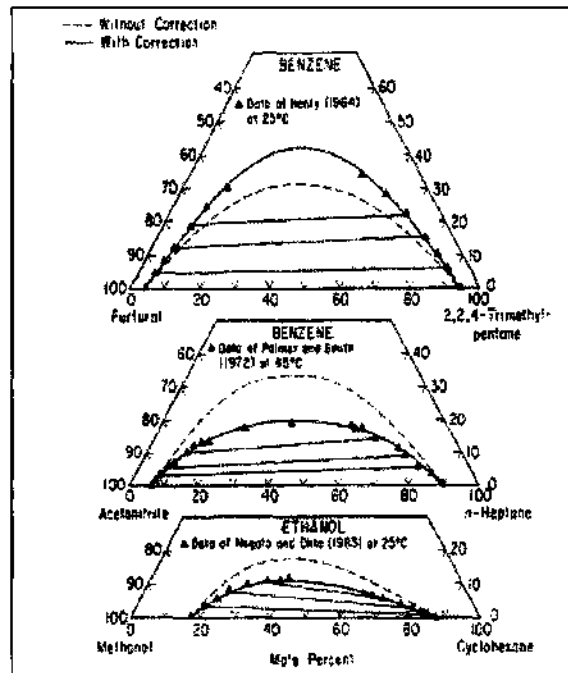
Figure 3-8 Calculated and observed liquid-liquid equilibria where the binodal curve shows only modest asymmetry (Prausnitz et al, 1985)

It is anticipated that the correction factor will not reach large values as the ternary LLE shows more sensitivity to activity coefficients and are not affected by pure-component vapour pressures. The systems studied here did not require large corrections to $\left(\dfrac{G^{E}}{RT}\right)^{o}_{123}$. Although, the correction factor is small, the effects on ternary LLE data have been dramatic. The effect on ternary VLE is expected to be minimal given that the magnitude of the correction factor C was never far removed from unity (Cha et al, 1985) and since ternary VLE is dictated by the accuracy of pure component vapour pressures rather than activity coefficients. This theory was only tested for a single ternary

system, acetonitrile(1)-n-heptane(2)-benzene(3). Here the correction factor did not compromise the accuracy in predicting the ternary VLE system. Cha et al. (1985) conclude that the effect of this correction factor on ternary VLE will not be significant.

Advantage - The correction factor was able to simultaneously correlate VLE and LLE data.

Disadvantage - The use of the correction factor is limited to the case of the ternary systems and may not be applicable to higher order systems.

Disadvantage – The correction factor ability to correlate ternary LLE is still dependent on model selection and binary interaction parameters. If the experimental data is not accurate or the uncorrected model fails to give a good description of the data, then the correction factor is futile.
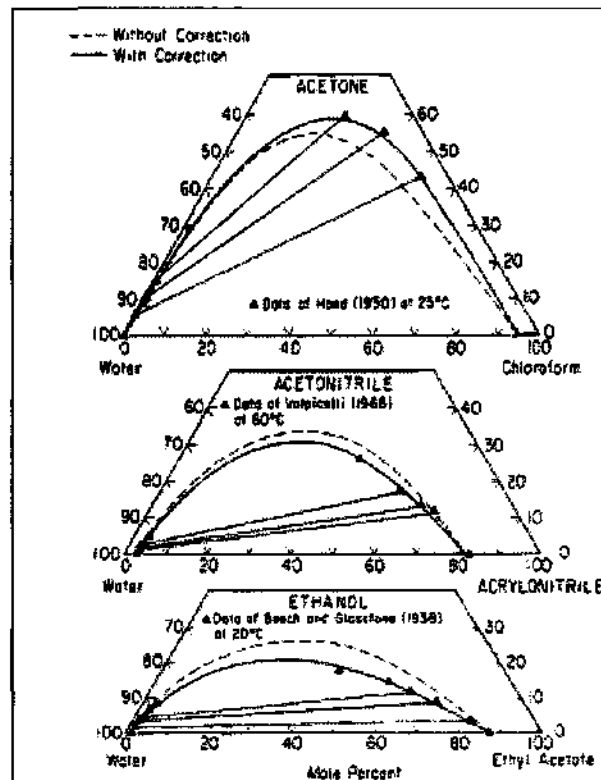
Figure 3-9 Calculated and observed liquid-liquid equilibria where the binodal curve shows strong asymmetry (Prausnitz et al, 1985)
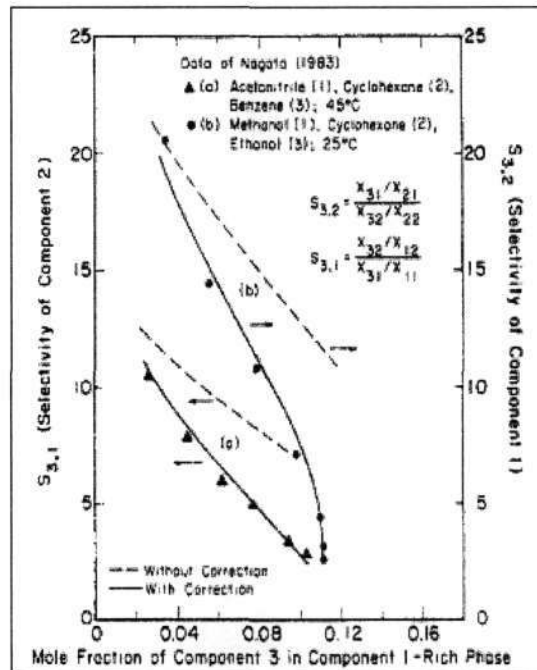
Figure 3-10 Calculated selectivities display drastic improvements when the correction factor is used (Prausnitz et al, 1985)
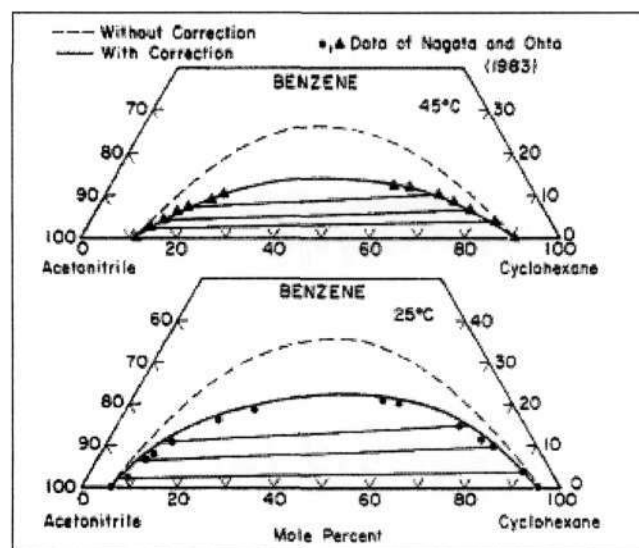
Figure 3-11 Predicted liquid-liquid equilibria for acetonitrile (1) – cyclohexane (2) and benzene (3) at 25^{0}C (Prausnitz et al, 1985)

3.4.4 Non-linear Concentration Transformation

3.4.4.1 Introduction

This empirical modification is proposed by Rarey (2005). The transformation allows one to simultaneously correlate VLE and LLE data. However, this transformation of the liquid composition does include an additional parameter. An additional parameter has to be used in order to simultaneously describe the VLE and LLE. (Rarey, 2005).

The transformation can be applied to any G^E-model. Rarey (2005) investigated the effects of such a transformation on the performance of local composition models. The Wilson model is not considered here since it is not applicable to LLE data. Thus the UNIQUAC and NRTL models are only considered in the performance analysis.

3.4.4.2 The Formulation of Transformation

Current two parameter G^E-models are able to manipulate the size and symmetry of G^E curves. The concentration dependence of the model then dictates the shape of the G^E curve (Rarey, 2005).
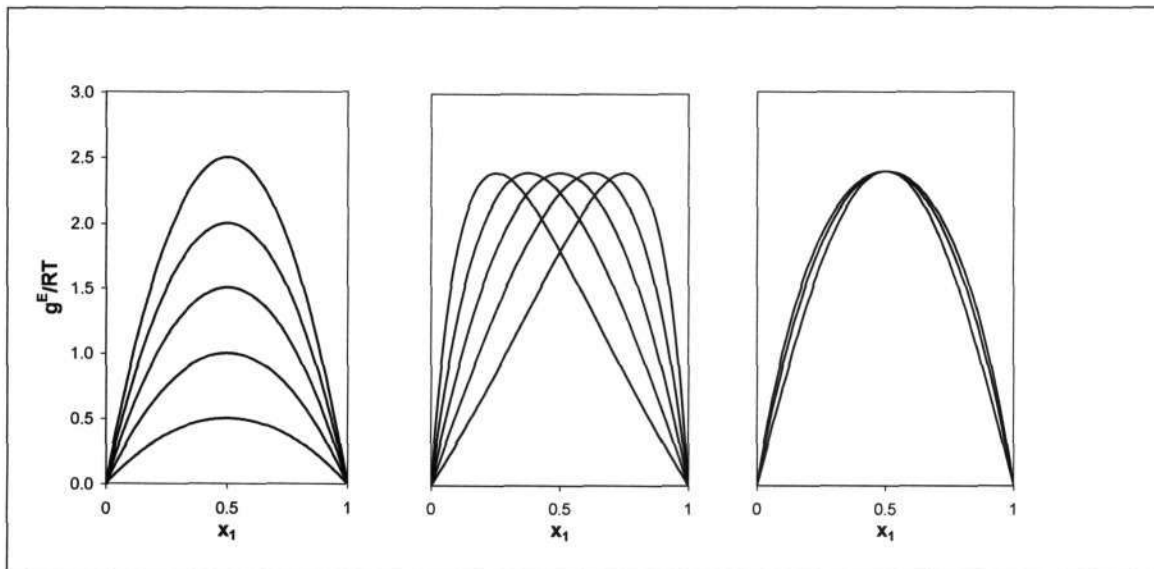
Figure 3-12 Depiction of the effects of size, symmetry and shape on G^E curves

(Rarey, 2005)

Therefore, the inclusion of an additional parameter should affect the shape of the G^E curve. One has to proceed carefully when proposing to incorporate an additional parameter. The following should be taken in cognisance with regards to the additional parameter:

- Will the inclusion of an additional parameter increase the degree of intercorrelation between the existing model parameters?
- Does this parameter compromise the model's existing capabilities?

Rarey (2005) implemented the additional parameter in the form of a symmetrical, non-linear concentration transformation f. The function f shows only little affect on the existing model parameters and hence the size and symmetry of the G^E is unaffected. The model is now redefined as:

$$G^E = G^E(f(x)) \qquad \text{3-14}$$

The mole fraction in the original G^E-model is replaced with the function f. The implementation of the transformation is subject to some constraints. It must:

- be continuous and differentiable

- satisfy the summation condition $\sum_i f_i = 1$ (derived from the fact that $\sum_i x_i = 1$)

- be defined for systems containing any number of components

- be symmetrical with respect to liquid molar composition in binary systems

- obey the boundary conditions, $f_i = 0$ when $x_i = 0$ and $f_i = 1$ when $x_i = 1$

The transformation proposed by Rarey (2005) is as follows:

$$f_i(x_i) = \frac{x_i\left(1 + \sum_k x_k \delta_{ik}\right)}{\sum_k x_k\left(1 + \sum_l x_l \delta_{kl}\right)}$$
3-15

Here d_{ik} is the additional parameter. It should be noted that $d_{ii} = 0$ and $d_{ij} = d_{ji}$.

Rarey (2005) also suggests another possible transformation that could be used. In this particular transformation, only the odd powers should be used.

$$f_i(x) = \frac{x_i\left(1 + \sum_k x_k \sum_l (x_i - x_k)^{l-1} d_{ikl}\right)}{\sum_k x_k\left(1 + \sum_l x_l \sum_m (x_k - x_l)^{m-1} d_{klm}\right)}$$
3-16

Multi-component systems can be described and the extension of the transformation using higher order parameters is possible.

$$f_i(x) = \frac{x_i\left(1 + \sum_k x_k \sum_l (x_i - x_k)^{l-1} d_{ikl} + \sum_k x_k \sum_l x_l e_{ikl} + \sum_k x_k \sum_l x_l \sum_m x_m f_{iklm} + \ldots\right)}{\sum_k x_k\left(1 + \sum_l x_l \sum_m (x_k - x_l)^{m-1} d_{klm} + \sum_l x_l \sum_m x_m e_{klm} + \sum_l x_l \sum_m x_m \sum_n x_n f_{klmn} + \ldots\right)}$$
3-17

3.7.3 Procedure for Calculating Activity Coefficient

The activity coefficient is derived from

$$\ln \gamma_i = \left(\frac{\partial (n_T Q)}{\partial n_i} \right)_{T,P,n_{j \neq i}} \qquad \text{3-18}$$

where $Q = \dfrac{G^E}{RT}$

This yields the following expression:

$$\ln \gamma_i = Q + n_T \left(\frac{\partial Q}{\partial n_i} \right)_{T,P,n_{j \neq i}} \qquad \text{3-19}$$

Now, G^E is a function of the transformation f(x), which is a function of composition. This has to be taken into account when deriving the new activity expression.

$$Q = Q(f(x))$$

The derivation of the activity coefficient requires the use of the chain rule.

$$\left(\frac{\partial Q}{\partial n_i} \right) = \left(\frac{\partial Q}{\partial f} \right)\left(\frac{\partial f}{\partial n_i} \right) \qquad \text{3-20}$$

Here $\left(\dfrac{\partial Q}{\partial n_i} \right)$ and $\left(\dfrac{\partial Q}{\partial f} \right)$ are vectors while $\left(\dfrac{\partial f}{\partial n_i} \right)$ represent a Jacobian or gradient matrix of concentration. Thus the expression for activity coefficient is:

$$\ln \gamma_i = Q(f) + \nabla f^T \left(\frac{\partial Q(f)}{\partial f} \right)_{T,P,n_{j \neq i}} \qquad \text{3-21}$$

In the binary case, the Jacobian matrix is derived from the following expression:

$$\nabla f = \frac{\partial f(x_i)}{\partial n_j} = \frac{\partial \left[\dfrac{x_i \left(1 + \sum_k x_k \delta_{ik} \right)}{\sum_k x_k (1 + \sum_l x_l \delta_{kl})} \right]}{\partial n_j} = \frac{\partial \left[\dfrac{n_i n_T + n_i \sum_k n_k \delta_{ik}}{n_T^2 \sum_k n_k \sum_l n_l \delta_{kl}} \right]}{\partial n_j} \qquad \text{3-22}$$

The differentiation of f has to be performed carefully because of the subscripts. Two cases have to be considered, one with identical subscripts and the other where the subscripts are not identical.

For the case where i = j the elements of the Jacobian have the form,

$$\frac{\partial f(x_i)}{\partial n_j} = \frac{1}{S_T} \left(1 + x_i + S_i - 2x_i \frac{(1+S_i)^2}{S_T} \right) \qquad \text{3-23}$$

For the latter case the elements of the Jacobian have the form,

$$\frac{\partial f(x_i)}{\partial n_j} = \frac{1}{S_T} \left(x_i (1 + d_{ij}) - 2x_i \frac{(1 + S_i)(1 + S_j)}{S_T} \right) \qquad \text{3-24}$$

where

- $S_i = \sum_k x_k d_{ik}$

- $S_T = 1 + \sum_k x_k S_k$

3.7.4 Performance of the FlexQUAC Model

The transformation was applied to the UNIQUAC model and since the purpose of the transformation was to improve the flexibility of the model, the subsequent model was named FlexQUAC.

The model was first tested with 4000 binary data sets containing pressure composition data (Px isothermal data) contained in the DDB. The mean relative squared deviation in pressure was used as the objective function to regress the data sets.

Naturally no big improvements were observed for the cases where the activity coefficients have no effect, since these are nearly ideal systems (**Figure 3-13**). The more non-ideal the systems gets, the higher the improvement is. At activity coefficients at infinite dilution greater than four, there is a constant improvement of more than 2.5 in the objective function. At coefficients greater than 15, the plot diverges signalling the onset of LLE behaviour and in these cases the quality of the data is very much reduced.
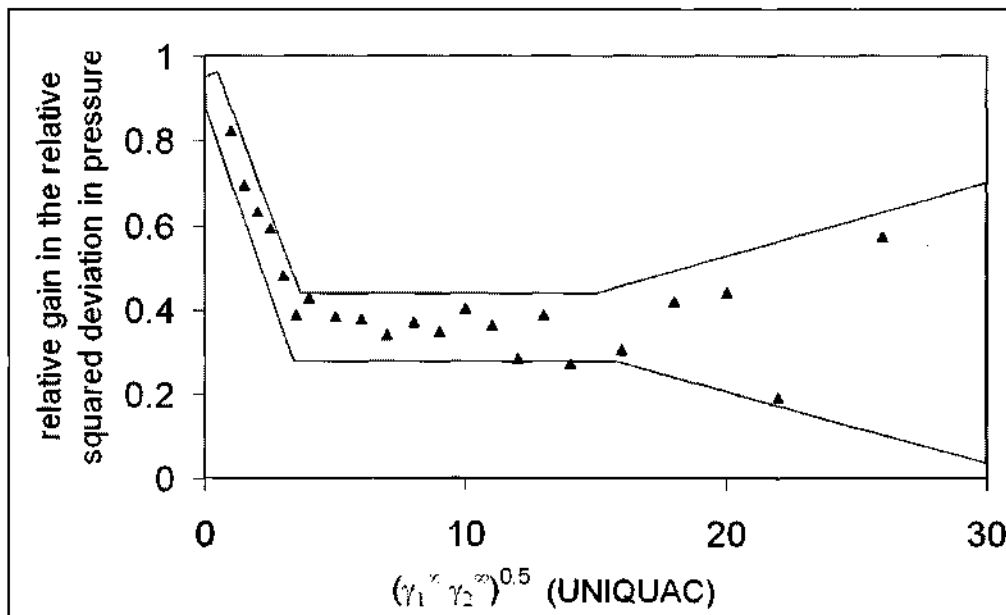
Figure 3-13 Relative Gain in objective function (▲) versus the geometric mean of the activity coefficients at infinite dilution calculated from the UNIQUAC regression (Rarey, 2005)

A typical example is given in **Figure 3-14** for the system tetrahydrofuran-water. The UNIQUAC fit of the data is not precise and the azeotropic point is not described well. FlexQUAC fits the data and describes the azeotropic point precisely. Although the difference between the two curves is small, the system is more accurately described by FlexQUAC.

With regards to ternary VLE, the intention was to illustrate that the new model does retain its ability to accurately predict higher order systems and that this ability is not compromised. Thirteen reliable ternary sets were chosen. **Figure 3-15** illustrates that FlexQUAC is superior to UNIQUAC in some cases or at worst predicts as well as the UNIQUAC model. **Table 3-2** illustrates the ternary systems investigated.
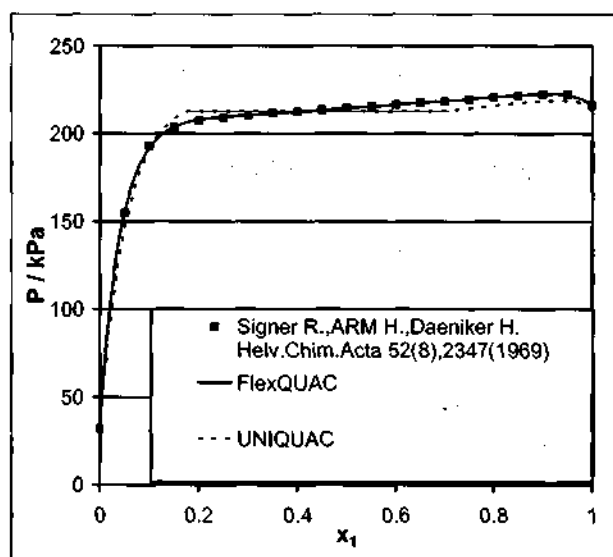
Figure 3-14 Regression results for the system tetrahydrofuran-water at T=298.15K using the UNIQUAC- and FlexQUAC-model (Rarey, 2005)
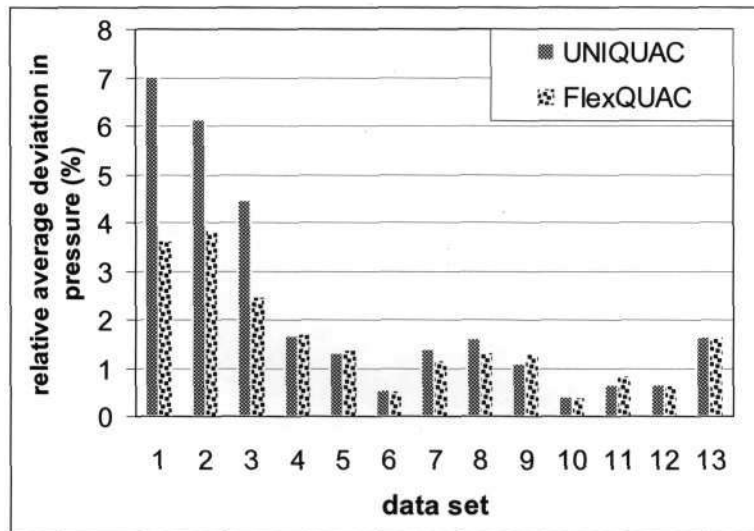
Figure 3-15 Comparison of the predictive performance between the UNIQUAC and FlexQUAC models for ternary mixtures (Rarey, 2005)
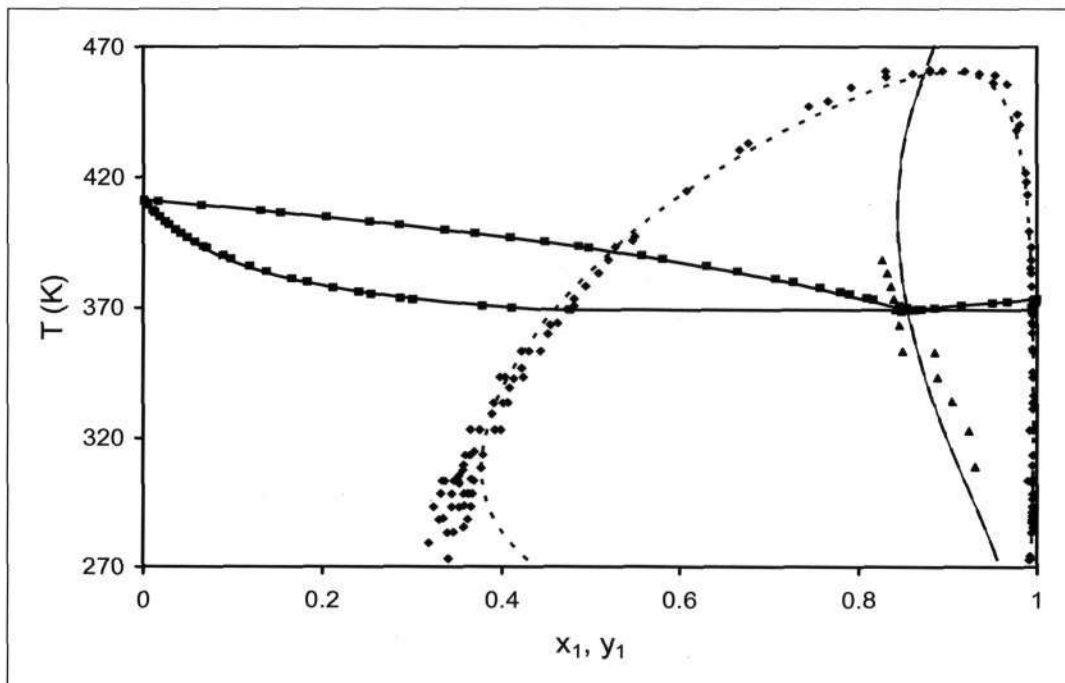
Figure 3-16 Simultaneous correlation of VLE and LLE using FlexQUAC G^E-model parameters from LLE data and VLE data for the system Water (1) – 1-Pentanol (2) (■ - VLE data, ♦ - LLE data, ▲ - azeotropic data, —— VLE calculation, ······ LLE calculation, ‾‾‾ azeotropic composition calculation) (Rarey, 2005)

Ternary Systems	T [K]	Mean Relative Deviation in P [%]	
		UNIQUAC	FlexQUAC
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	293.15	7.01	3.61
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	303.15	6.12	3.82
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	313.15	4.46	2.47
Acetone (1) - Methanol (2) - Chloroform (3)	313.15	1.68	1.71
Acetone (1) - Methanol (2) - Chloroform (3)	323.15	1.33	1.36
Acetone (1) - Methanol (2) - Chloroform (3)	323.15	0.56	0.51
Ethanol (1) - Acetonitrile (2) - Water (3)	323.15	1.41	1.16
Acetone (1) - Ethanol (2) - Water (3)	323.15	1.62	1.32
Ethanol (1) - Water (2) - 1,4-Dioxane (3)	323.15	1.08	1.29
Acetone (1) - Methylacetate (2) - Acetonitrile (3)	323.15	0.42	0.37
Benzene (1) - Cyclohexane (2) - Aniline (3)	343.15	0.65	0.83
n-Hexane (1) - Benzene (2) - Cyclohexane (3)	343.15	0.65	0.63
1-Heptene (1) - n-Heptane (2) - n-Octane (3)	328.15	1.64	1.64

Table 3-2 Ternary systems investigated in the evaluation of the FlexQUAC model (Rarey, 2005)

The model now adequately describes VLE and LLE simultaneously as illustrated in Table 3-2.

Chapter 4 Simultaneous Regression & Software Tools

4.1 Introduction

The regression of experimental phase equilibrium data facilitates parameter fitting using e.g. an excess Gibbs energy model. In **Figure 4-1**, the process of predicting equilibrium data is broken down into two steps. The first step involves the correlation of data to obtain model parameters and the second step involves the use of these model parameters in the appropriate excess Gibbs energy model or equation of state to interpolate and extrapolate thermodynamic data within a certain range of composition, temperature and pressure.

The accuracy of models is dictated by the model parameters and model equations. The experimental data may contain errors that can be attributed to measurement. Thus caution must be exercised in the regression step to account for possible errors. This then defines the level of confidence in model parameters and the subsequent predictions resulting from the model.

In order to assess the performance of the FlexQUAC-Q model in this study, phase equilibrium data had to be regressed simultaneously. This chapter reviews the important steps in this

regression process. The subject of simultaneous regression of phase equilibrium is explained further due to its relevance to this research study. Once model parameters are obtained from regression of data, different methods are used to calculate mixture data. These methods are discussed to illustrate how different types of data measurements are handled.

The process of regressing thermodynamic data is iterative and requires a certain criterion to be satisfied in order to terminate the algorithm. The criteria are contained in the form of an objective function. The regression process seeks to either minimize or maximize the objective function depending on the application. In the case of phase equilibrium data, the definition of a generic objective function would be the difference between the measured and calculated variable and the aim of the regression would be to minimize this function. The objective function is also dependent on the type of data measurements. The commonly used objective functions and the variations on its form are discussed here.

Since measured data have inherent errors, one has to account for these. This would minimize the error in the prediction of thermodynamic data. The maximum-likelihood principle accounts for these errors and is summarized here.

A regression algorithm is required for the regression of thermodynamic data. For the purposes of this research study, the regressions were made using the Simplex-Nelder Mead method. The theory of the method, the implementation and the application of this method in the Excel regression program is presented here to improve the reader's understanding.

The Dortmund Data Bank, DDB (Gmehling et. al., 2006) is used extensively in this research study to obtain thermodynamic mixture data and pure component data. Mixture data obtained here was regressed to assess the performance of the FlexQUAC-Q model in the excel program. The DDB's simultaneous regression software tool, RECVAL was also used extensively to regress mixture data in this study to compare the results with the excel regressions. RECVAL was also used to perform regression of mixture data using its array of features that allows the user to effectively manipulate the regression of data. A brief overview of this program is presented here to aid the reader's understanding of some of the program's innovative capabilities.
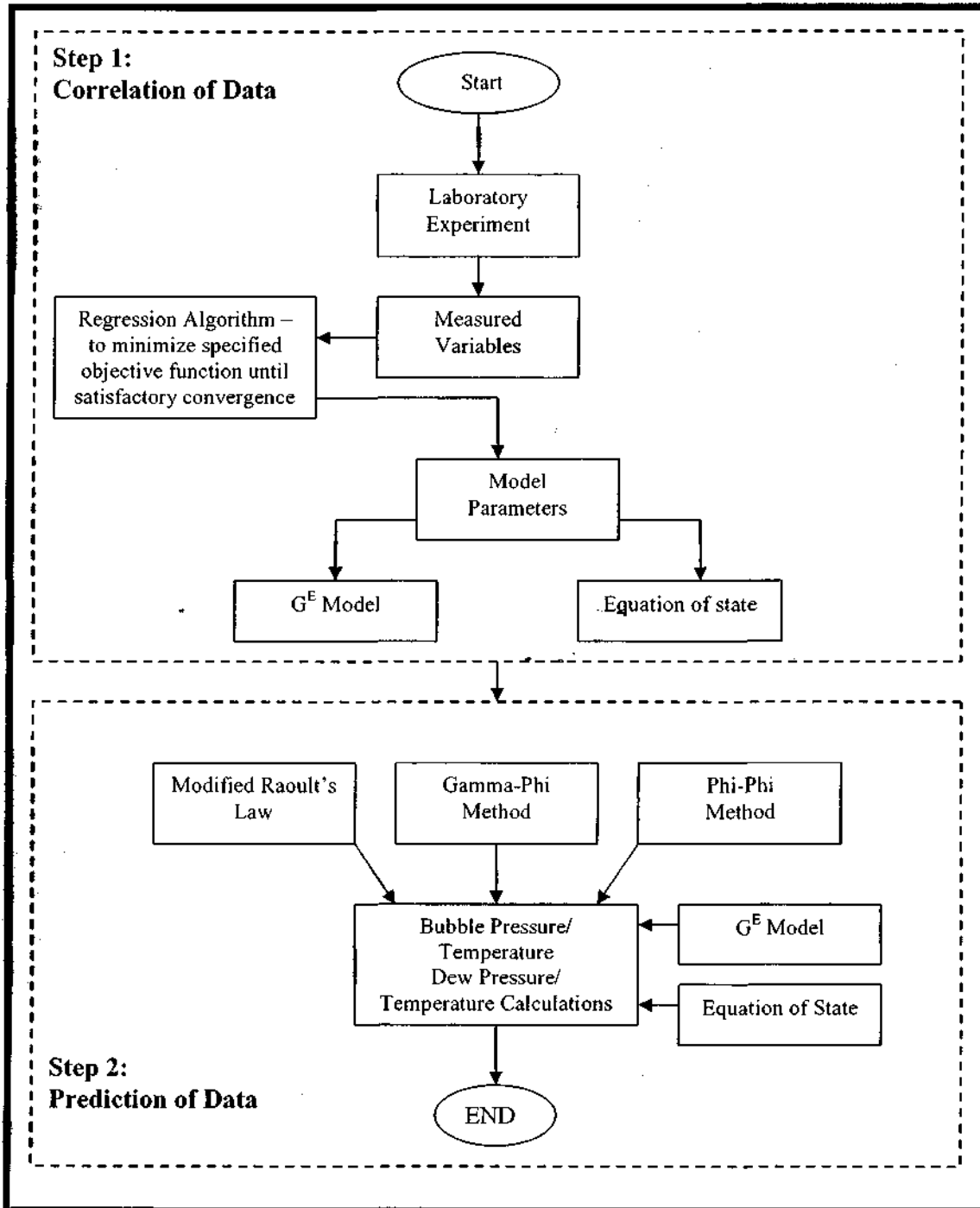
Figure 4-1 The sequence of steps required for regression and calculation of experimental data

4.2 Simultaneous Regression of Phase Equilibrium data

Simultaneous regression of different phase equilibrium data is a valuable tool for obtaining an accurate description of mixture properties. Different sources of experimental data are available. In addition to the data types outlined in **Table 4-1**, there exists also mutual solubility (LLE), azeotropic data, activity coefficients at infinite dilution (γ^∞ data), SLE, H^E and Cp^E data.

Since the number of data points measured usually exceeds the number of model parameters, a regression minimizing the deviation between the experimental data and model calculations has to be performed. Hence a unique set of model parameters that reproduces all experimental data with sufficient precision is not always obtained. Small errors in the binary parameters lead to larger errors in the prediction of multicomponent data.

Liquid-liquid equilibrium, in particular exhibits significant sensitivity to small changes in the activity coefficients (Prausnitz et al, 1980). Hence it is advised to include some LLE data in the regression to obtain reliable model parameters. Binary data are mostly inadequate for the prediction of ternary LLE data. In this case, some ternary data need to be utilized in the regression procedure to obtain suitable binary parameters. Usually ternary tie line data are regressed simultaneously together with binary VLE data to obtain optimal binary parameters. Consider an arbitrary system A, B and C where A-B is a partially miscible binary system and both B-C and A-C are completely miscible. For the partially miscible system A-B, the mutual solubility data is used to obtain the corresponding binary parameters. VLE data for the remaining miscible binaries are used to provide reasonable estimate of model parameters. An optimal set of parameters is obtained by simultaneous regression of both the VLE data and the ternary LLE data.

Figure 4-2 shows a ternary LLE system where both binary VLE and ternary tie-line data was used for regression purposes. The two-phase region was not well represented by original UNIQUAC. With the use of the ternary tie line data the description of the two phase region is greatly improved.
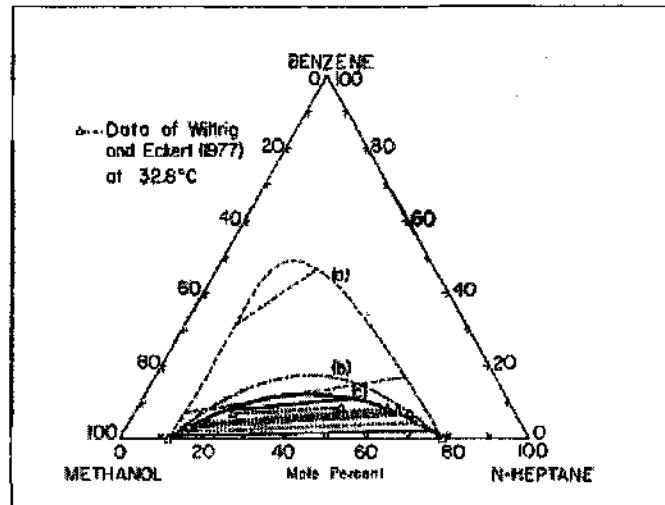
Figure 4-2 LLE for a type 1 system (a) original UNIQUAC (b) modified UNIQUAC for alcohol systems (c) Modified UNIQUAC with binary parameters calculated from binary VLE data and ternary tie-line data (Prausnitz, et al 1980)

4.3 Objective Function

The focus for regression of thermodynamic data is to minimize the deviation between the measured and calculated variable. Several types of objective function can be defined. One commonly used objective function is the relative mean squared deviation (RMSD):

$$F = \frac{1}{n} \sum_i^n \left(\frac{X_i^{exp} - X_i^{calc}}{X_i^{exp}} \right)^2 \qquad \text{4-1}$$

where

- X_i^{exp} refers to the experimental value of variable X for data point i

- X_i^{calc} refers to the calculated value of variable X for data point i

- n refers to the number of data points

Some other objective functions are:

$$F = \frac{1}{n} \sum \sum \left(X_{cal,i,j} - X_{exp,i,j} \right)^2 \qquad \text{4-2}$$

$$F = \frac{1}{n}\sum\sum\left|X_{cal,i,j} - X_{exp,i,j}\right| \qquad \text{4-3}$$

$$F = \frac{1}{n}\sum\sum\left(\frac{X_{cal,i,j} - X_{exp,i,j}}{X_{exp,i,j}}\right)^2 \qquad \text{4-4}$$

$$F = \frac{1}{n}\sum\sum\left|\frac{X_{cal,i,j} - X_{exp,i,j}}{X_{exp,i,j}}\right| \qquad \text{4-5}$$

In the case of simultaneous regression of different data types the objective function has to account for the deviations in different measured variables. Consider a regression of three different measured variables, X, Y and Z. The corresponding objective function would have the form:

$$F = \frac{1}{n}\sum_i^n\left[W_1\left(\frac{X_i^{exp} - X_i^{calc}}{X_i^{exp}}\right)^2 + W_2\left(\frac{Y_i^{exp} - Y_i^{calc}}{Y_i^{exp}}\right)^2 + W_3\left(\frac{Z_i^{exp} - Z_i^{calc}}{Z_i^{exp}}\right)^2\right] \qquad \text{4-6}$$

where W_1, W_2 and W_3 are weighting factors.

Weighting factors can be placed in the objective function to manipulate the weight of individual data points, data sets or data types.

In this study, all the binary VLE data was high precision xPT data. Hence the following objective function was used in regression:

$$F = \frac{1}{n}\sum_{i=1}^n\left(\frac{P_{exp,i} - P_{calc,i}}{P_{exp,i}}\right)^2 \qquad \text{4-7}$$

4.4 Calculation Method

Chapter 2 reviewed the calculation methods for thermodynamic mixture properties. The calculation method is dependent on the type of data measured and experimental setup. Data can be measured isothermally (constant temperature) or isobarically (constant pressure) or at constant

composition. Isothermal data are usually measured in static cells. In this case the pressure, as a function of liquid composition, can be obtained by varying the composition of the system in question and by measuring the pressure once phase equilibrium is reached. Thus temperature, pressure and liquid mole fractions are obtained experimentally and the vapour mole fraction can be calculated.

Table 4-1 illustrates the distinction between the various data types and the measured and calculated variables with the γ-φ method as its basis.

Computation type	Controlled variables	Computed Variable
Bubble pressure	x_i, T	P, y_i
Bubble temperature	x_i, P	T, y_i
Dew pressure	y_i, T	P, x_i
Dew temperature	y_i, P	T, x_i

Table 4-1 Different computation types for VLE data

The bubble pressure calculation algorithm is presented here (Figure 4-3) to better explain the procedure. In this case the objective function can be defined to minimize the error between computed pressure and experimental pressure or both computed and experimental pressure and vapour composition, y_i. The algorithm is an iterative one. Calculation methods for the other remaining types are readily accessible in many textbooks e.g. Smith & Van Ness et al. (1996).

In this case the regression of experimental data provides one with parameters for an appropriate excess Gibbs energy model. This model then generates activity coefficients and the unmeasured variables are then calculated by a bubble pressure calculation.

Isobaric data can be regressed in a similar manner as isothermal data. This is provided that the model parameters are not strongly dependent on temperature and can be considered negligible. In this case the unmeasured variable would be calculated by a bubble temperature iterative calculation.
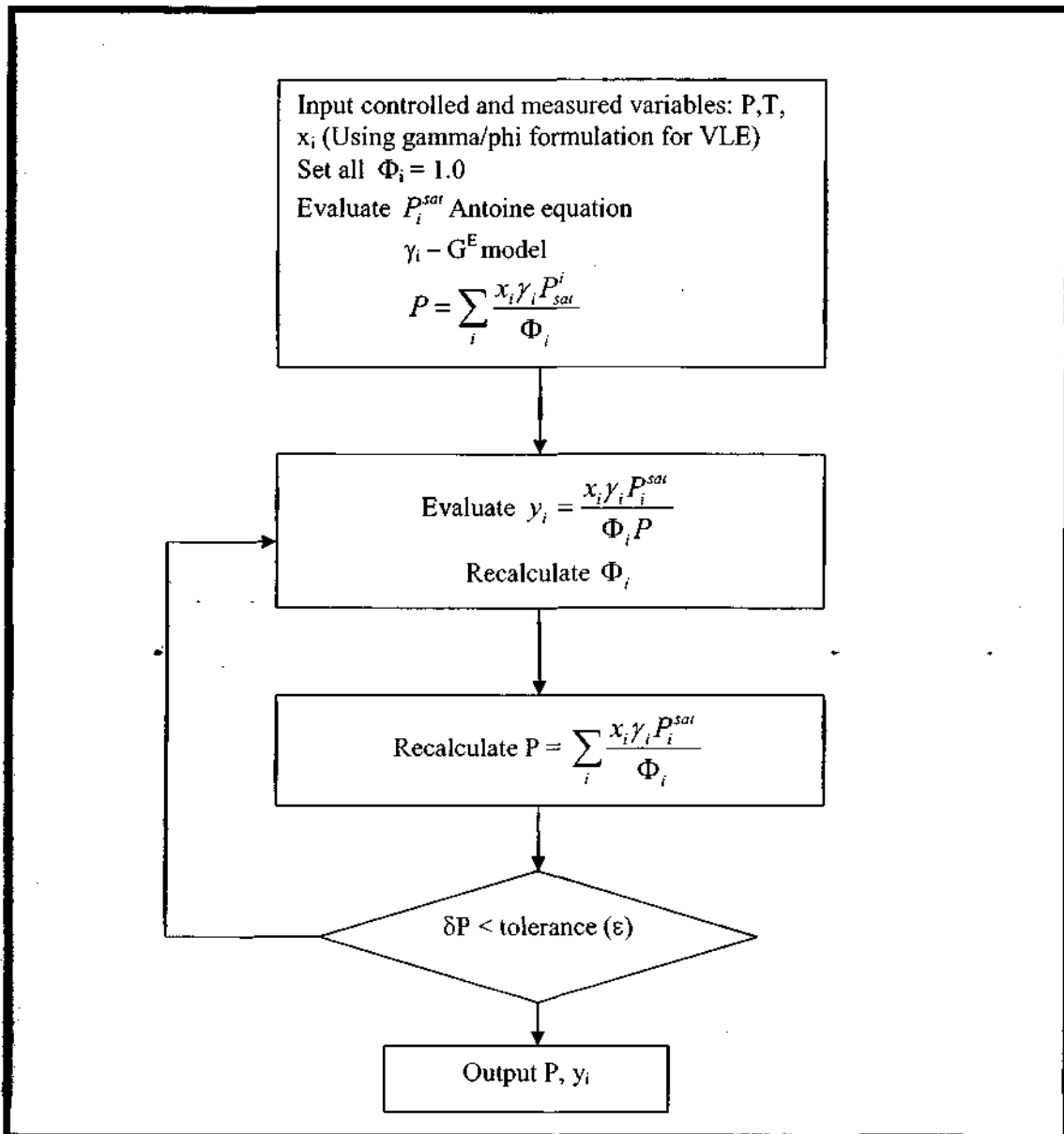
Figure 4-3 Bubble pressure computation procedure

4.5 Maximum-Likelihood Principle

It is anticipated that all measured data will contain some degree of error. Since this directly impacts the accuracy of model parameters, it is advisable to accommodate these errors statistically.

Experimental errors are either systematic or random. Systematic errors may arise due to a bias in the experimental method (Prausnitz, et al, 1980). The resulting data exhibit a departure from the proper value. Systematic errors can be limited to a certain concentration or pressure range. To avoid errors of this type, one needs to carefully analyse the experimental procedure and ensure good understanding of the equipment before undertaking the measurements.

Random errors, on the other hand can be accounted for statistically. These errors produce uncertainties in the model parameters. The errors are further exasperated when these random errors are coupled with the inadequacies of some models.

The maximum likelihood principle takes into account random errors of both controlled and measured variables. This principle postulates that an experimentally determined variable has a normal distribution about its exact value and a corresponding variance. Usually the variances are specific to the type of experimental equipment. The likelihood function is defined as the joint probability of the observed values of the variables for any set of true values of the variables, model parameters and error variances. "The best estimates of the measured variables are those which maximize this likelihood function with a normal distribution assumed for the experimental errors" (Prausnitz et al, 1980). This function is defined as follows:

$$S = \sum_{i=1}^{M} \left\{ \frac{\left(P_i^o - P_i^e\right)^2}{\sigma_{P_i}^2} + \frac{\left(T_i^o - T_i^e\right)^2}{\sigma_{T_i}^2} + \frac{\left(x_{ii}^o - x_{ii}^e\right)^2}{\sigma_{x_{ii}}^2} + \frac{\left(y_{ii}^o - y_{ii}^e\right)^2}{\sigma_{y_{ii}}^2} \right\} \qquad 4\text{-}8$$

where

- M - no. of data points
- e – measured variable
- o – estimated true value
- σ^2 – estimated variance of the measured variables

The model parameters that minimize this function represent the optimal parameters.

4.6 The Simplex-Nelder-Mead Method

4.6.1 The Theory of the Simplex-Nelder-Mead Method

The downhill simplex method was modified by Nelder and Mead (1965). This method is preferred since it requires only function evaluations. No derivatives are necessary. It can be inefficient considering the number of function evaluations it requires. This regression although slow, is very stable and is suited for the regression within this research study.

A simplex is the simplest body in N dimensional space. It consists of N+1 vertices and all their interconnecting line segments, polygonal faces. In two dimensional space a simplex would correspond to a triangle while in three dimensional space it would correspond to a tetrahedron. The simplex must enclose some finite N dimensional volume.

The simplex method is a systematic procedure for generating and testing the candidate vertex solutions to a non-linear program. The method requires an initial simplex to be defined in N dimensional space as an initial guess. Alternatively, values and step width can be specified for each parameter.

After the worst point (corresponds to the highest value of the objective function) is identified, this point is reflected along the centroid through the simplex. The objective function is evaluated at this point and if it corresponds to a new minimum, the simplex is expanded along this line.

However, if the new reflected point is worse than the previous point then the algorithm contracts the simplex in one dimension from the worst point. In the event that the new point is worse than

the previous points a multiple contraction towards the best point is undertaken. A combination of these steps (reflection, expansion, contraction) yields an optimal solution.

4.6.2 Reflection

The centroid of the N best points in the vertex is $\bar{x} = \frac{1}{n}\sum_{j \neq s}^{N} x_j$. x_s represents the point to be reflected. Let the reflection point be denoted by x_r located on the line joining the centroid to x_s. It is defined by:

$$\bar{x}_r = \bar{x} + \alpha(\bar{x} - x_s) \qquad 4\text{-}8$$

Here α is the reflection coefficient. If $F(x_z) > F(x_r) \geq F(x_b)$ then the reflected point, x_r is accepted replaces x_s. Here x_z represents the vertex which yields the second worst value of the target function while x_b is the vertex with the best value of the target function. The new simplex is re-evaluated.
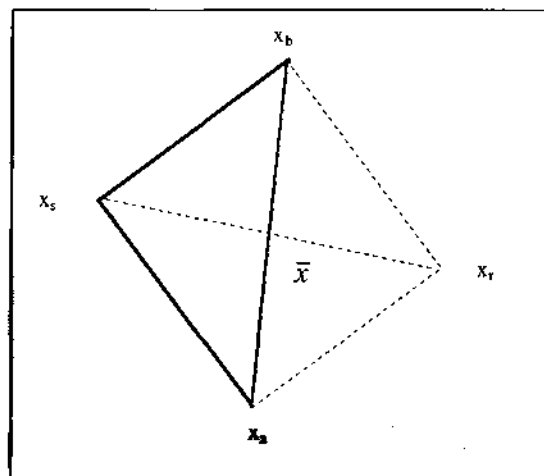
Figure 4-4 Representation of reflection for Simplex-Nelder-Mead algorithm

4.6.3 Expansion

If $F(x_r) < F(x_b)$ then the reflection has produced a new minimum. x_r is then expanded to a new point x_e.

$$x_e = \bar{x} + \gamma(x_r - \bar{x}) \qquad \text{4-9}$$

where

- γ refers to the expansion coefficient

If $F(x_e) < F(x_b)$, then x_s is replaced by x_e. In the case where $F(x_e) > F(x_b)$, then the expansion has failed and x_r replaces x_s before restarting.
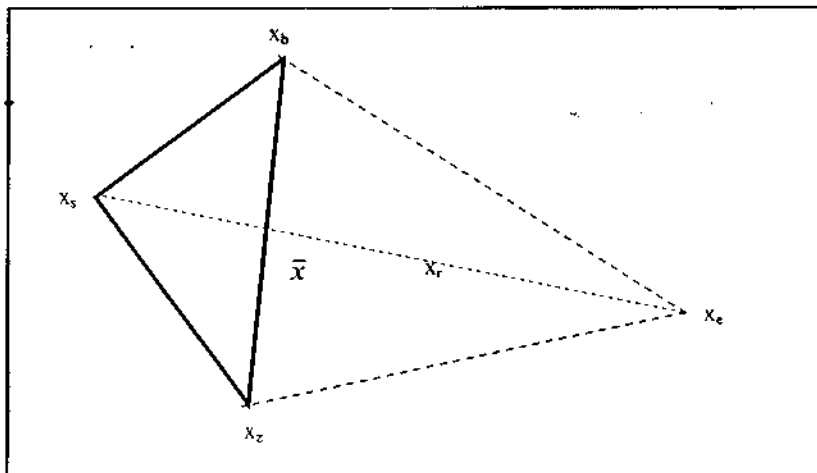
Figure 4-5 Representation of expansion for Simplex-Nelder-Mead algorithm

4.6.4 Contraction

If $F(x_r) > F(x_z)$ and/ or $F(x_r) > F(x_s)$ then a point x_c is generated by contraction

$$x_c = \bar{x} + \beta(x_s - \bar{x}) \qquad \text{4-10}$$

where

- β refers to the contraction coefficient

If $F(x_c) < F(x_s)$, then x_s is replaced by x_c. If $F(x_c) > F(x_s)$ then all x_j are replaced by $(x_j + x_b)/2$ and the entire process is restarted.
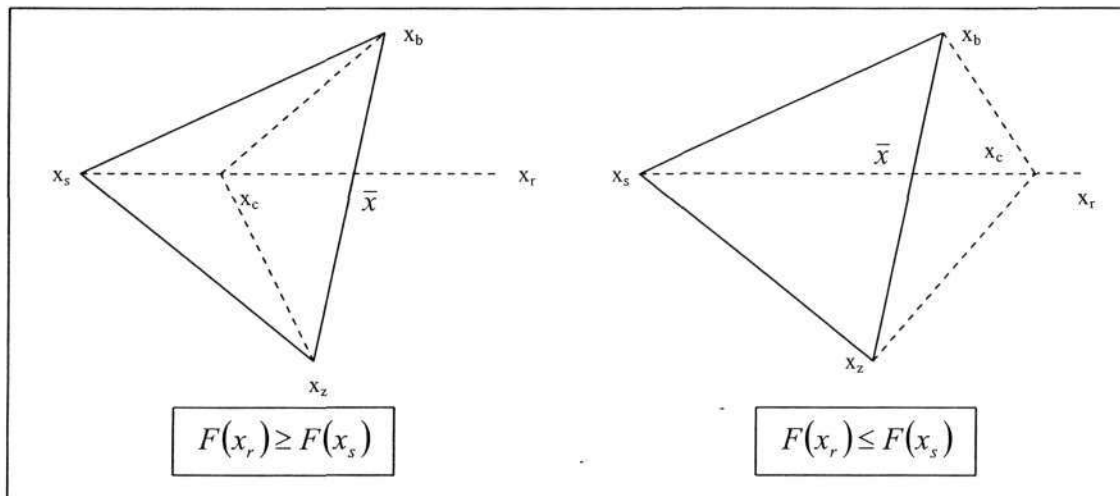
Figure 4-6 Representation of contraction for Simplex-Nelder-Mead algorithm

4.6.5 The Simplex-Nelder-Mead Algorithm

Start
Build Simplex

Calculate $F(x_j)$, $F(x_b)$, $F(x_z)$ and $F(x_s)$

$$\bar{x} = \frac{1}{n}\sum_{j \neq s} x_j$$

Reflection
$$\bar{x}_r = \bar{x} + \alpha(\bar{x} - x_s)$$

No — $F(x_r) < F(x_b)$ — Yes — $F(x_r) > F(x_z)$ — Yes — $F(x_r) < F(x_s)$ — No

Yes

Expansion
$$x_e = \bar{x} + \gamma(x_r - \bar{x})$$

Contraction
$$x_c = \bar{x} + \beta(x_r - \bar{x})$$

Contraction
$$x_c = \bar{x} + \beta(x_r - \bar{x})$$

$F(x_e) < F(x_r)$ — No

Yes

No

$F(x_c) > F(x_s)$ — No

Yes

$x_s = x_e$

$x_s = x_r$

$x_j = (x_j + x_b)/2$

$x_s = x_k$

Has the minimum been reached?

No

Yes

STOP
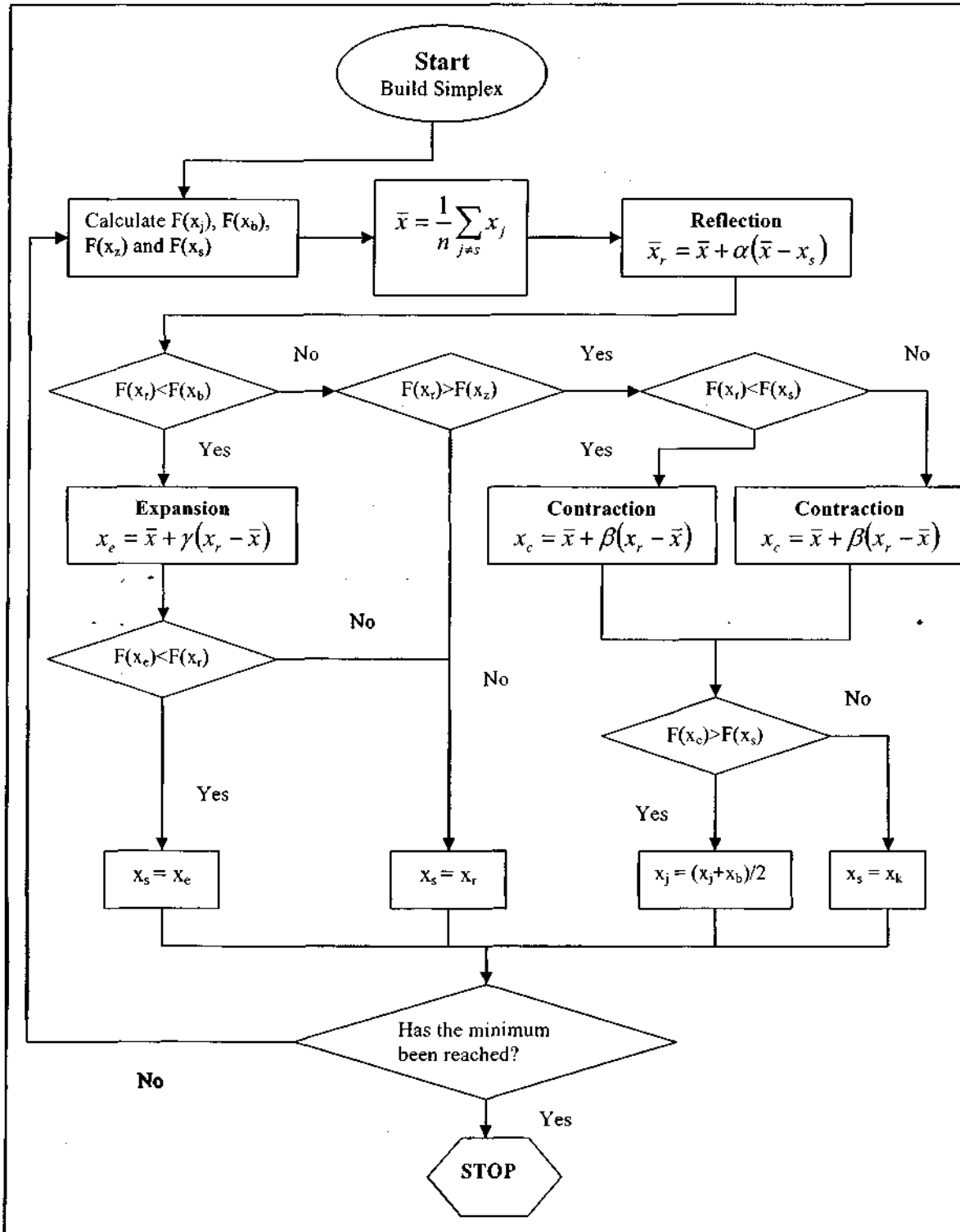
Figure 4-7 Flow diagram of the Simplex-Nelder-Mead Algorithm

4.6.6. Utilisation of the Simplex-Nelder-Mead Method

The Simplex-Nelder-Mead regression algorithm is used in an Excel program (**Figure 4-8**) to obtain model parameters for the FlexQUAC-Q model. The main program provides the initial set of parameters for the regression and then calls the *Datafit* subroutine which defines the auxiliary information for the regression. This sets the step width for the parameters, the stop criterion and the maximum number of iterations. These parameters are used by the Simplex-Nelder-Mead subroutine (DSIM) for the regression. These parameters are then used by the function AUX to perform the bubble temperature calculation and to calculate the objective function. The parameters obtained are then displayed on the Excel spreadsheet after the convergence criterion has been satisfied.
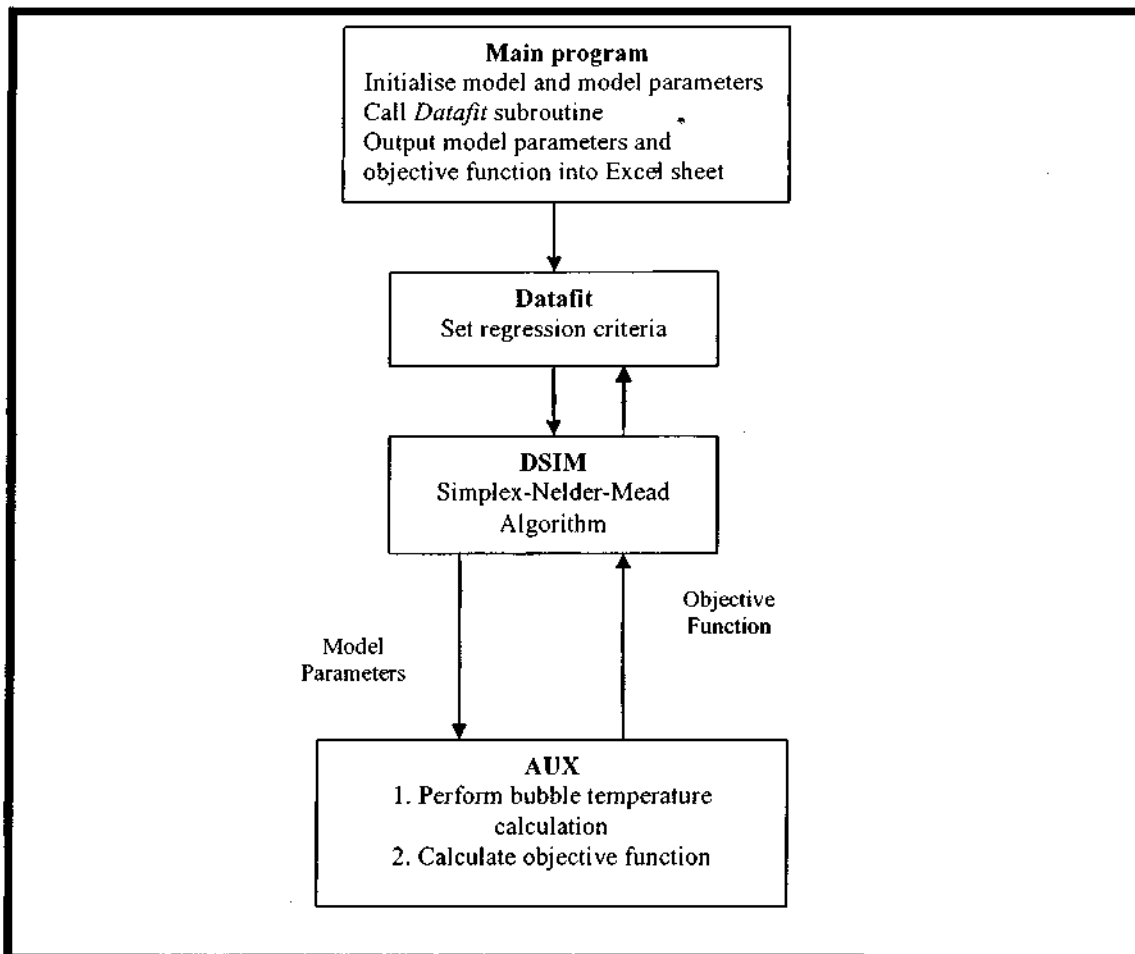
Figure 4-8 Flowchart of the Excel progam for data regression

4.7 Dortmund Data Bank (DDB)

4.7.1 History of the DDB

The Dortmund Data Bank (DDB) was initiated in 1973 by J. Gmehling at the University of Dortmund as a tool for the development of a group contribution G^E model for VLE in collaboration with Lyngby in Denmark. This was later extended to LLE, H^E, γ^∞, azeotropic data, Cp^E and SLE for the development of modified UNIFAC. Group contribution equations of state (PSRK and VTPR) for the VLE of low boiling compounds were then developed. Next VLE and GLE of electrolyte systems were stored. This led to the development of the electrolyte models, LIQUAC and LIFAC. Thereafter the pure component data base was initiated for the development of estimation methods for pure component properties. 2004 saw the launch of a polymer phase equilibrium data bank. In 1989 DDBST GmBH took over further development of the DDB. In 2000, the DDBST delivered the first windows version of the DDBSP which is a software package for data handling, correlation and estimation and process synthesis tools. Storing the huge amount of phase equilibrium and excess property data on computer provided the basis for the development of predictive methods. The DDB today has earned the reputation as the largest and most well established computerized data bank for both pure component and mixture properties.

The substantial amount of pure component and mixture data in the DDB allows one to fit model parameters to various thermodynamic models, including G^E models and equations of state for a large variety of mixtures. The data is indispensable in process simulation where reliable interaction parameters are required. The vast amount of data that often span a large temperature and composition ranges are of great importance for the development of group contribution methods. In addition the data has greatly aided the development of predictive methods.
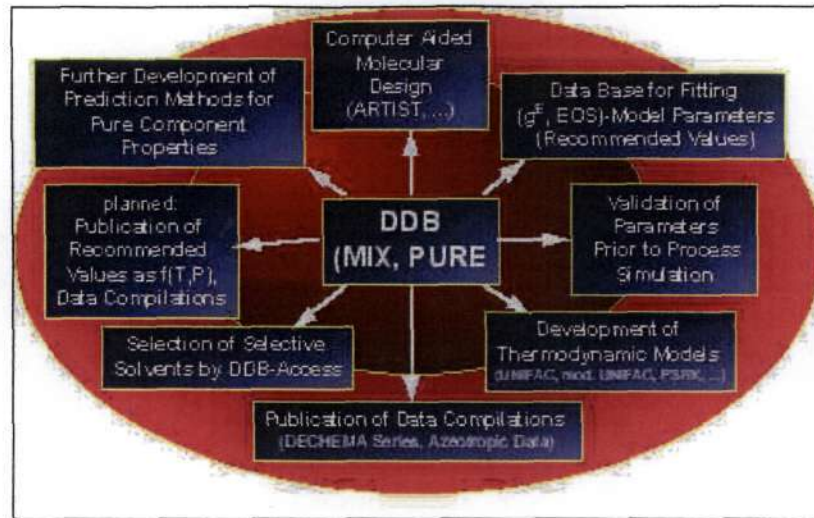
Figure 4-9 Different applications of the DDB (www.DDBST.de)

The DDB is updated on a yearly basis. **Figure 4-10** illustrates the phenomenal development of the DDB.
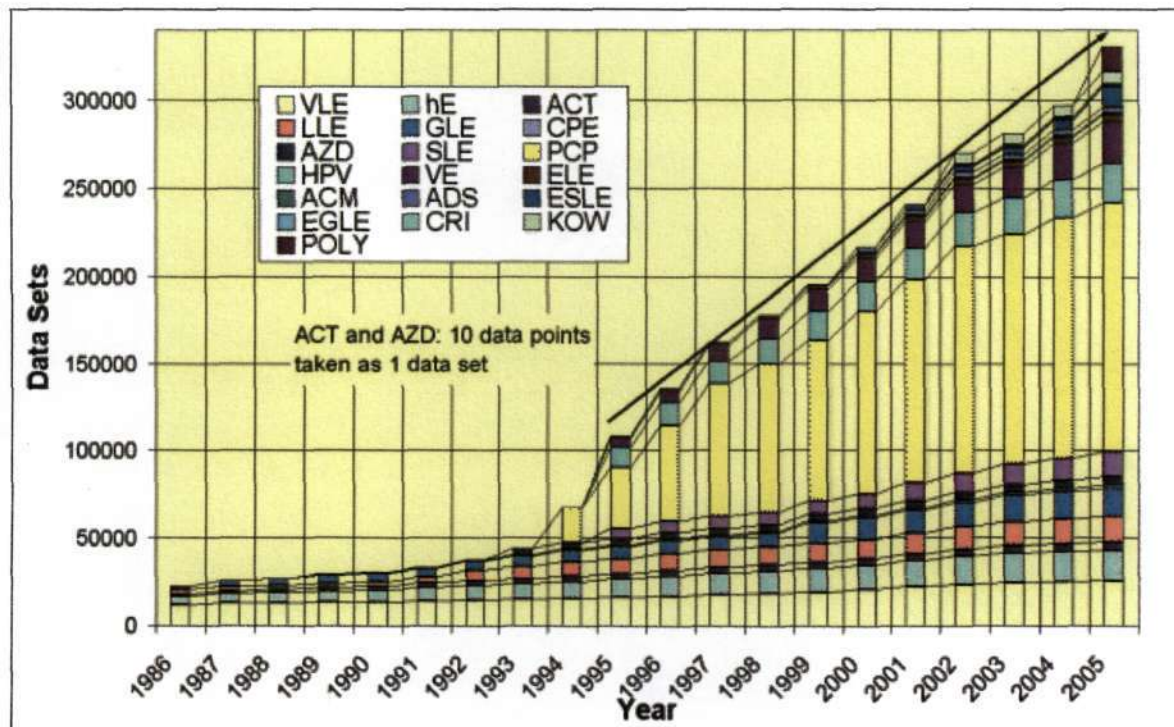
Figure 4-10 Rate of expansion of the DDB since 1986 (www.DDBST.de)

4.7.2 Software Tools

The DDB software package (DDBSP) contains a number of software applications tailored for use in industry. The software tools have a range of applicability ranging from data retrieval, graphical representation, calculations and property estimation. A short review of DDBSP is presented here to enable one to understand its capabilities. The RECVAL software package is presented in more detail since it is used extensively in this research.
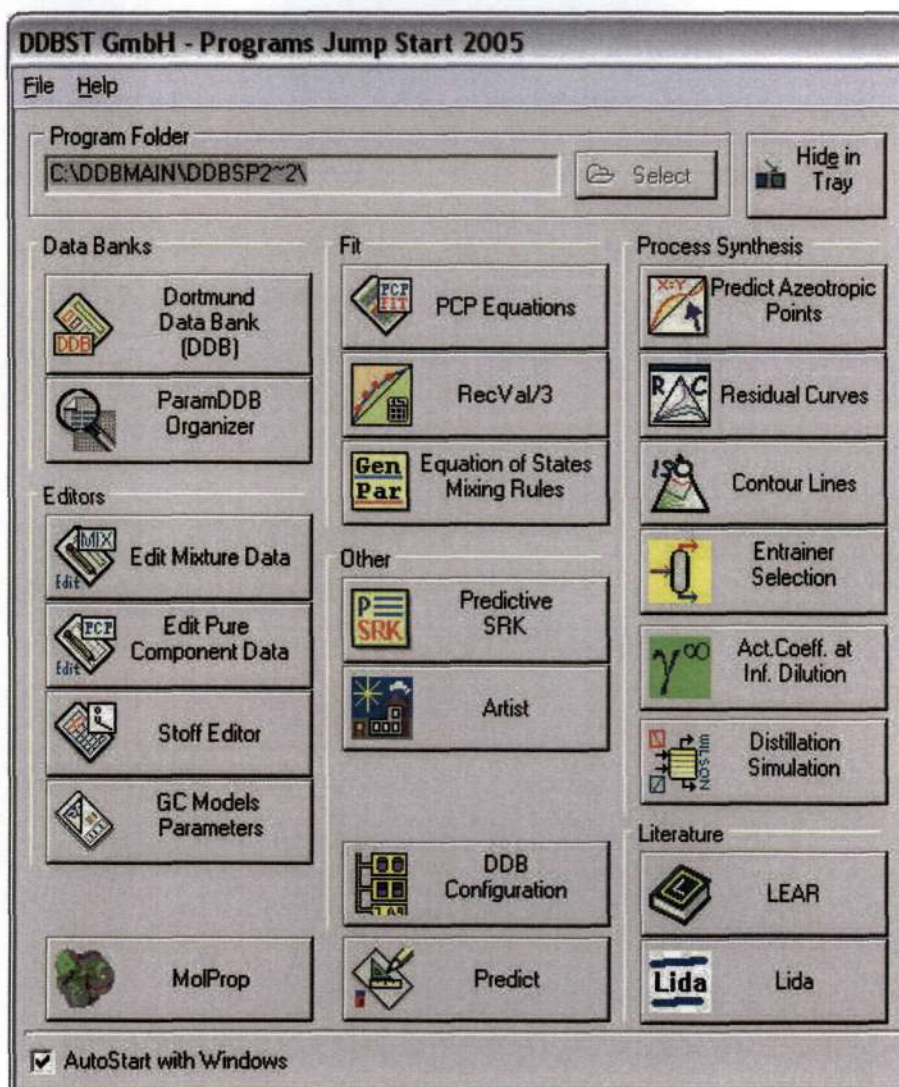
Figure 4-11 Various software tools available in the DDB software package (www.DDBST.de)

4.7.2.1 Data Viewer (Dortmund Data Bank)

This tool is used to perform data queries for pure component or mixture data.
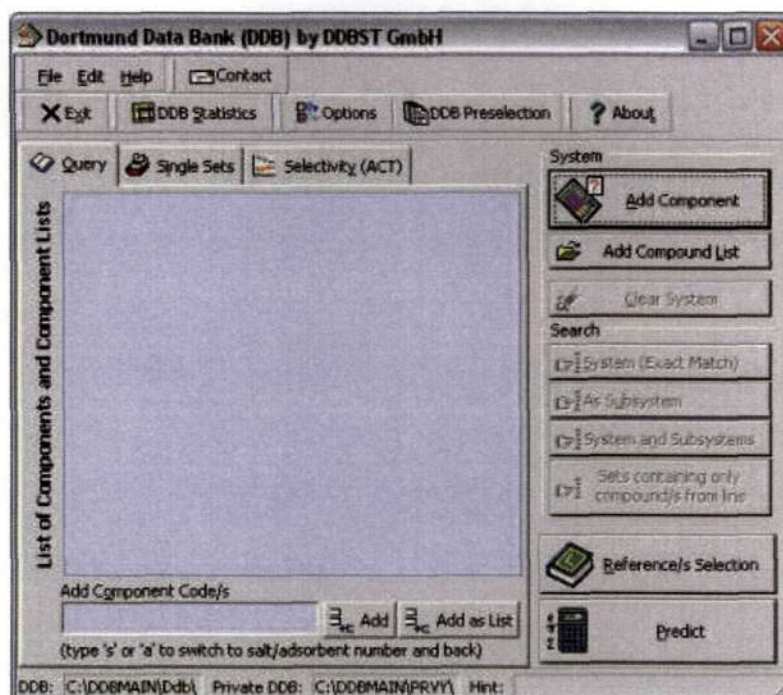
Figure 4-12 Query dialog of the data viewer (www.DDBST.de)

The user may specify a component or sets of components for data retrieval. The user can either enter a DDB pure component code number or search for the component of interest. After searching for the component(s) the data viewer enables the user to view and edit pure component or mixture data. The viewer provides a list of important parameters ranging from basic information (Antoine constants, molecular weights, boiling points etc.) to UNIFAC and UNIQUAC molecular parameters (**Figure 4-13**).

If one considers the mixture, ethanol and water, then the data viewer provides a comprehensive listing of all available binary data. The user can then navigate through the data to find the appropriate data required (**Figure 4-14**). By right clicking on a data set number, the user can view, plot and fit experimental data contained in the data set. The 'fit' option allows the user to select the G^E model or equation of state and the objective function.
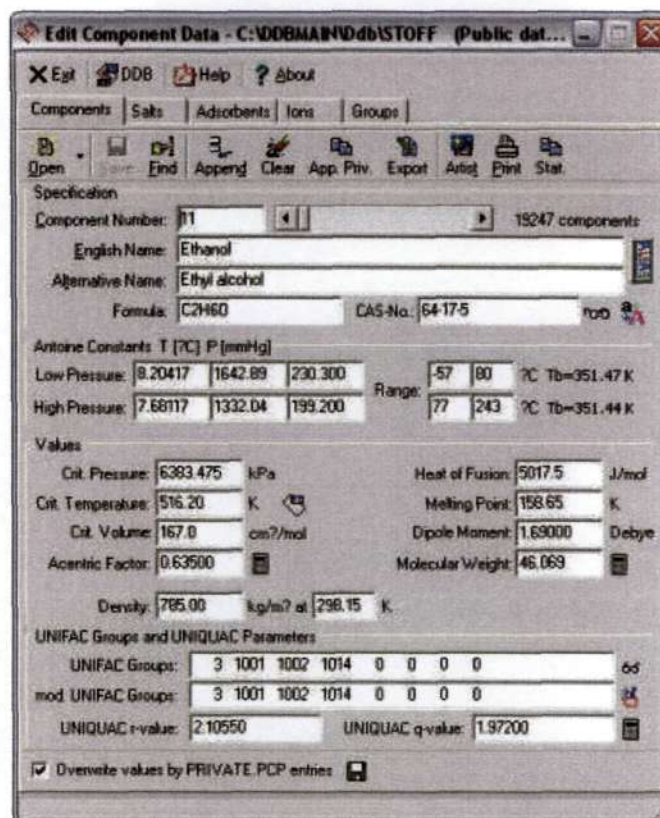
Figure 4-13 Pure component basic data for ethanol (www.DDBST.de)

4.7.2.2 ARTIST

This program is used for the estimation of pure component properties using group contribution methods. "Using a special coding scheme for molecular structures, ARTIST is able to take into account the chemical neighborhood of the individual groups thus implementing chemical know-how. Special features of some group contribution methods like topological indices and second order group corrections are automatically taken into account" (www.DDBST.de) .

4.7.2.3 Process Synthesis

This software tool is of particular importance to industry for the prediction of homogeneous and heterogeneous azeotropic compositions in multicomponent mixture, calculation of residual curves and contour lines and the selection of entrainers for various separation processes.

Figure 4-14 Various available binary data for the system ethanol-water (www.DDBST.de)

4.7.2.2 ARTIST

This program is used for the estimation of pure component properties using group contribution methods. "Using a special coding scheme for molecular structures, ARTIST is able to take into account the chemical neighborhood of the individual groups thus implementing chemical know-how. Special features of some group contribution methods like topological indices and second order group corrections are automatically taken into account" (www.DDBST.de) .

4.7.2.3 Process Synthesis

This software tool is of particular importance to industry for the prediction of homogeneous and heterogeneous azeotropic compositions in multicomponent mixture, calculation of residual curves and contour lines and the selection of entrainers for various separation processes.

4.7.2.4. Recval

This is the regression tool of the DDB used for the simultaneous regression of phase equilibria and excess properties. The data obtained in the *Mixview* data viewer for a specific chemical system can be exported to *Recval*.

Recval has the following options:

- *File* – Allows user to open projects, generates projects from mixture data files, save projects and exit the program
- *Edit* – Various options to specify the regression
- *Special* – Selectively remove data of different criteria
- *Run* – Initiate the regression of data
- *Results* – Displays calculation or regression results in tabular or graphical form
- *Options* – General options for calculations and plot specifications
- *Help* – Comprehensive help system pertaining to use of *Recval*

The *Recval* program is an excellent tool for regression of data. The user is given a multitude of options thereby achieving accurate results quickly and efficiently. The graphical interface of the

program allows for easy navigation and is user friendly. The capabilities presented here are but a superficial overview of the program and the true appeal of such a tool can only be realized once in use. Its versatility, speed and data handling options make it an indispensable software tool for regression of data especially in this research study.

4.7.2.4.1 Edit Menu

The **Edit** menu (**Figure 4-15**) allows the user to select from various options. This allows the user to select the most appropriate liquid and vapour phase models. The regression properties can also be tailored to specification. Initial interaction parameters and weighting factors can be assigned.

The '*Components, Properties*' option allows the user to set the models to calculate the pure component vapour pressures and liquid density. The vapour pressures can be computed from the Antoine, DIPPR or Wagner equations depending on which parameters are available.

With the '*Liquid Mixture Model Selection*' (**Figure 4-16**) the user can define the appropriate G^E model to be used. Currently the following models are available:

- Ideal (Raoult's Law) – calculation only
- Margules
- Van Laar (no h^E calculation)
- Wilson
- NRTL
- UNIQUAC
- FlexQUAC
- Redlich Kister
- Legendre Polynomial
- Scatchard-Hildebrand
- UNIFAC – calculation only
- mod. UNIFAC (Dortmund) – calculation only
- COSMO-RS

Recval supports the use of temperature dependence for interaction parameters. The user can specify the type of temperature dependence. The constant interaction parameters can adequately

predict the VLE data over a narrow temperature range. If the VLE data cover a larger temperature range then a temperature dependence of the interaction parameters is required. In the event that heats of mixing data are available, then temperature dependence should be utilized since they represent the first derivative of the excess Gibbs energy. If the data set contains heats of mixing data at various temperatures and they differ significantly; then quadratic temperature dependence should be utilized to describe the temperature dependence of the first derivative of G^E. This would also apply if excess heat capacity data is also available. Excess heat capacity data at different temperatures that differ significantly would require higher order temperature dependence. This however, is not supported in *Recval*.

The temperature dependence options available in *Recval* are:

- 1 - Par = A   - *constant interaction parameter*
- 2 - Par = A + B*T   - *temperature dependence*
- 3 - Par = A + B*T + C*T²   - *quadratic temperature dependence*
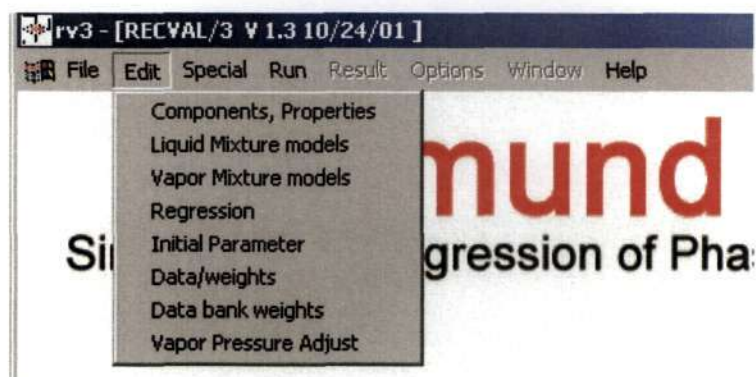
Figure 4-15 Options available in the 'Edit' menu
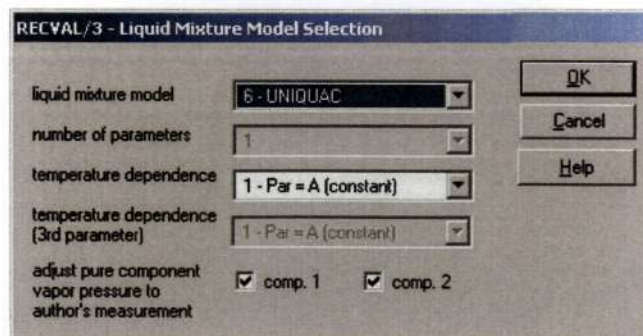
Figure 4-16 Liquid mixture model selection window
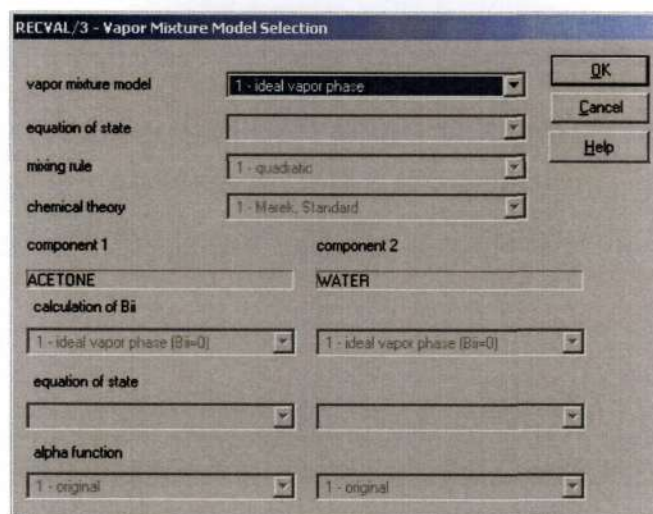
Figure 4-17 Vapour mixture model selection model

The 'Vapour mixture model Selection' (**Figure 4-17**) option currently supports the ideal vapour phase and cubic equations of state. In the event that one or both of the components are carboxylic acids, then the default model is set to the chemical theory model. For the cubic equations of state, the appropriate mixing rule and alpha functions have to be selected.

The 'Regression' (**Figure 4-18**) option allows the user to set the target criteria for convergence and the maximum number of iterations.
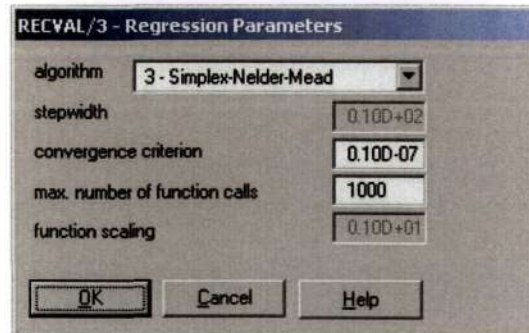
Figure 4-18 Regression parameter program window

The '*Initial parameter*' option allows the user to set initial values for the interaction parameters. It sets the step width for the regression. It also allows the user to fix a parameter value by deselecting it to exclude it from the regression or by setting its step width to zero. In addition the user can use the default starting values, previous regression results, previous starting values or load parameters from the DDB.

The '*data/weights*' option (**Figure 4-20**) allows the user to view the different data weights for each data set and data point. The details for each data set can be viewed i.e. type of measurement, no. of data points, the reference and consistency tests. The data weights, residual and objective function can also be viewed. Individual data points can also be viewed (**Figure 4-21**). The user has the option to remove a data set from the regression by hiding the data set.
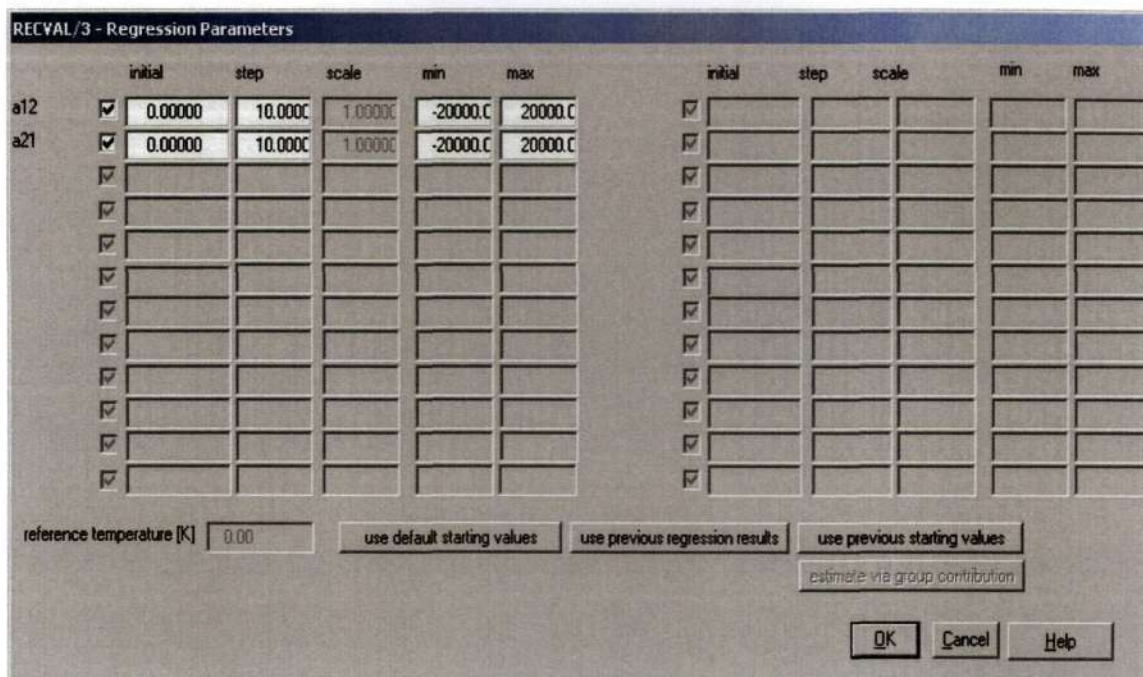
Figure 4-19 Initial parameter program window

The '*data bank weights*' (**Figure 4-22**) option allows the data bank weights to be adjusted. This defines the weighting of the different data types in the objective function. This is important since the objective function is adding properties that have different units with different orders of magnitude.

The '*Vapour Pressure Adjust*' option allows the user to modify the pure component vapor pressure values of the two components for each data set used to adjust the vapor pressure equation to fit the author's pure component data.

4.7.2.4.2 Special Menu

The '**Special**' option contains the following options:

o   *De-weight data in miscibility gap*

This option will set the weighting factors of all data points within the calculated miscibility gap to zero (not LLE data points).
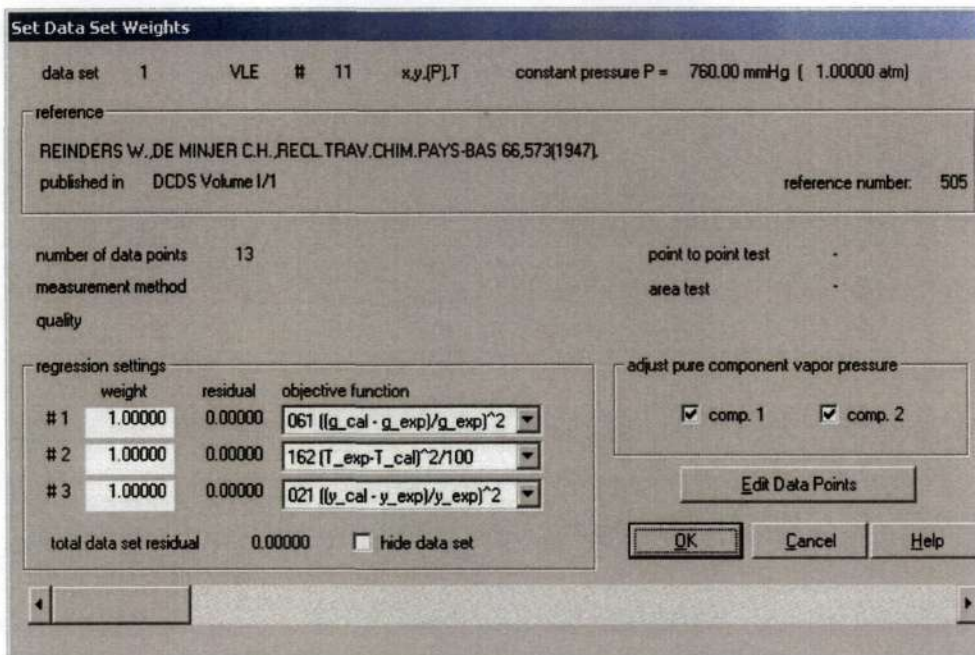
Figure 4-20 Data weights program window
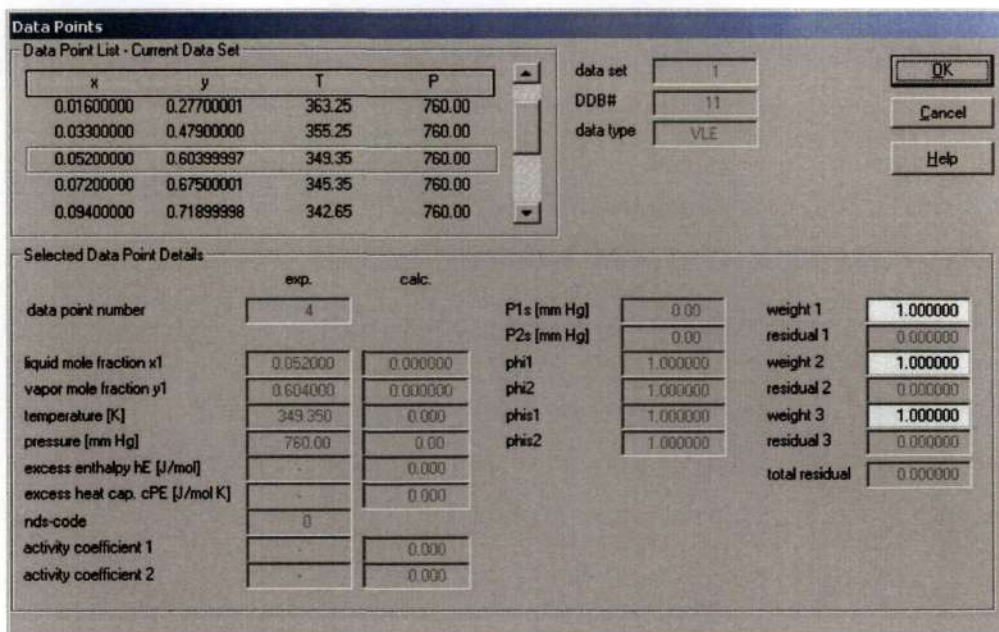
Figure 4-21 Data weights program window for individual data points
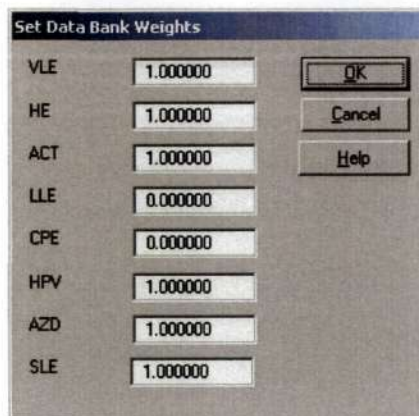
Figure 4-22 Data Bank weights program window

o   *De-weight LLE above 20 bar*

G^E models cannot describe the pressure dependence of LLE data measured above 20 bar. For this reason this data is de-weighted.

o   *De-weight inconsistent VLE*

All the data sets for which consistency tests were not done will be de-weighted. However consistency tests cannot be performed for all data sets in particular for the accurate static measurements and this would remove them from the regression completely.

4.7.2.4.3 Options menu

The 'Options' menu allows the user:

- To activate the calculation of azeotropes and LLE in a given temperature range with *'Regression/Calculations'* option
- The *'Plot'* option allows the user to change the settings for the graphics settings for the plots generated.

4.7.2.4.4 Run menu

This runs the regression based on the settings implemented by the user.

4.7.2.4.5 Results Menu

Figure 4-23 The various options available in the Results option

The '*Summary*' option displays the pure component parameters, the mixture data (model and binary interaction parameters), the data sets and their references and the objective function.

The '*Table*' option tabulates all experimental and calculated values from all regressed data sets.

The 'Graphics' (**Figure 4-24**) option generates various plots for the regressed data.
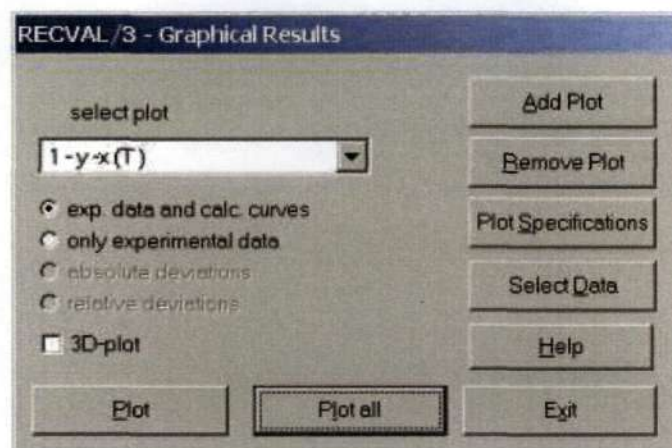
Figure 4-24 The graphics program window

The user can add a specific plot and format existing plots using '*Plot specifications*'. The '*Plot*' option initiates the program DDBMDIPLot which produces the graphical representation of data.

Chapter 5 Development & Implementation of the FlexQUAC-Q Model

5.1 Introduction

The development of the new model, FlexQUAC-Q, is based on a non-linear transformation proposed by Rarey (2005). The transformation was applied to the concentration space. The model, FlexQUAC achieved significant results: impressive reductions in the objective functions and has the ability to simultaneously correlate vapour-liquid and liquid-liquid equilibrium data. The FlexQUAC-Q model is developed with its core focus being asymmetric systems and with the intention of improving the prediction of equilibria data in these cases. In these cases the use of the FlexQUAC-Q model would be preferred to the use of the initial model, FlexQUAC.

This chapter reviews the work undertaken in this research study and is a meticulous guide into the development of the FlexQUAC-Q model. This chapter also reviews the steps for deriving,

implementing and subsequent testing of the FlexQUAC-Q model. First, the deficiencies of the FlexQUAC model are discussed to justify the development of the FlexQUAC-Q model. Thereafter, the derivation of the model is elucidated in detail. Next, the model was implemented and tested in both *Excel* and *Recval* (the regression tool for the DDB after the calculation procedure was programmed into Visual Fortran).

5.2 Deficiencies of the FlexQUAC Model

In the case of systems where the molecules are of comparable size, the performance of FlexQUAC is excellent. However, for the cases of mixtures where the difference in molecular size is considerable, then the use of the FlexQUAC model may not be a wise choice. In these asymmetric cases the non-linear transformation of the concentration space has a more pronounced effect on the activity coefficient of the larger activity coefficient.

Figure 5-1 illustrates the effects of the FlexQUAC model on the activity coefficient for increasing values of the third interaction parameter (δ_{12}). The UNIQUAC model is also represented here. The surface area ratio (ratio of the UNIQUAC q values) of the system represented is 2.72:1.4 where 2.72 is the q-value for the component tetrahydrofuran and 1.4 represents the q-value for the latter component, water. This ratio indicates that the system is typically asymmetric since the ratio is 1.94. The increase in δ_{12} appears to have a uniform effect on both sides of the plot. Upon closer examination the effects of the third interaction parameter has a more profound effect on the $\ln\gamma_i$ values in the dilute region. At molar composition of zero, the $\ln\gamma_i$ value increases from 6.68 to a value of 8 $(\delta_{12} = 0.2)$ compared to the 1.47 increase in the $\ln\gamma_i$ value at a molar composition of one.
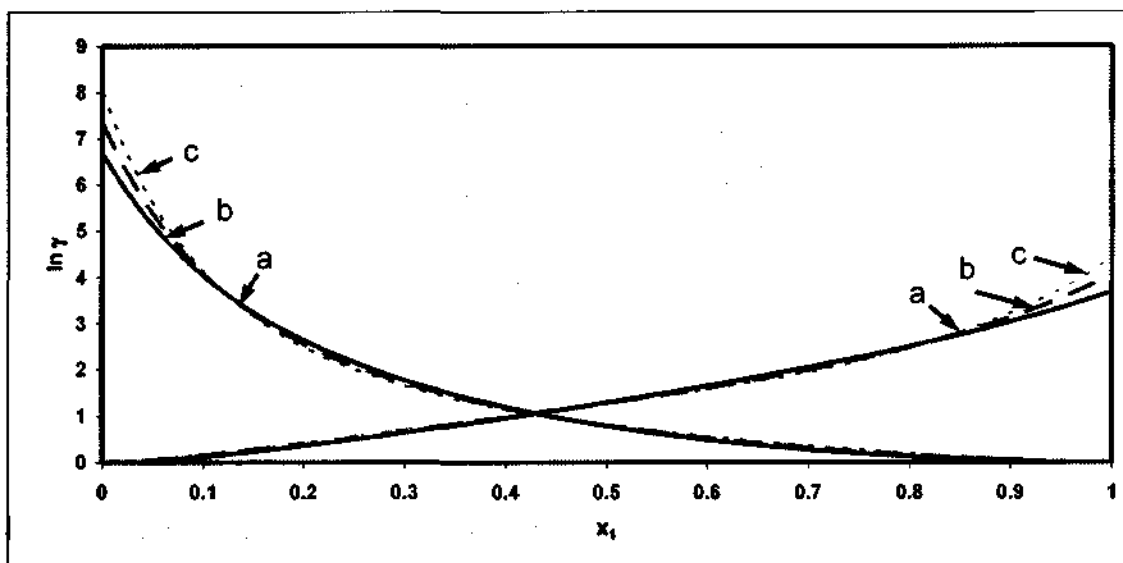
Figure 5-1 Comparison of the effects on lnγ values by comparing (a) UNIQUAC to FlexQUAC with third interaction parameter δ_{12} = (b) 0.1 and (c) 0.2 as a function of molar composition for the system water-tetrahydrofuran

Now consider the effects of the third parameter as expressed as a function of the surface fraction in **Figure 5-2.**
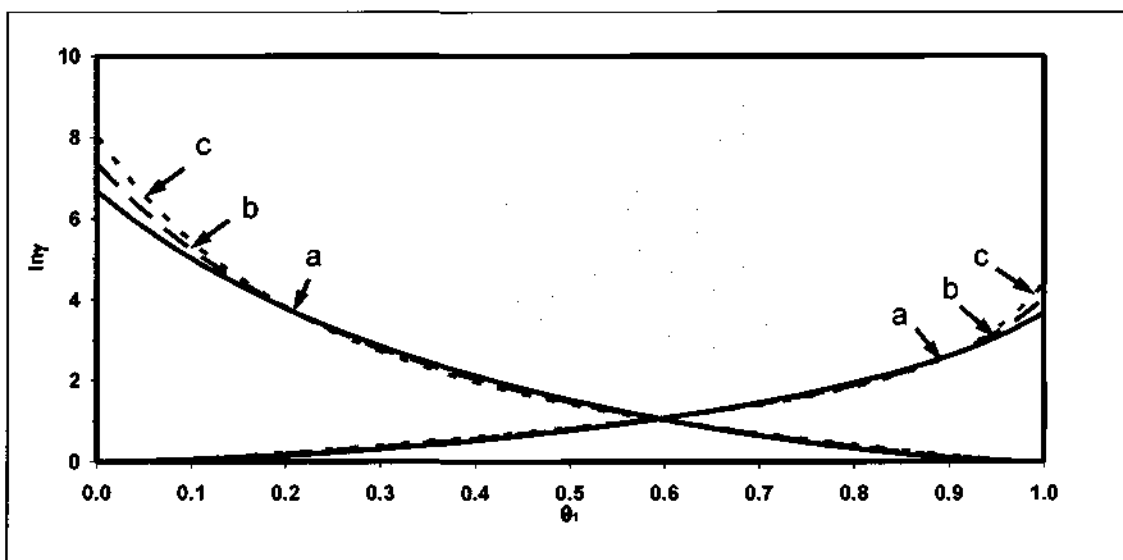
Figure 5-2 Comparison of the effects on lnγ values by comparing (a) UNIQUAC to FlexQUAC with third interaction parameter δ_{12} = (b) 0.1 and (c) 0.2 as a function of surface fraction for the system water-tetrahydrofuran

This plot serves to confirm the idea that the effects of the third parameter are more evident in the dilute range rather than the higher concentration range of the larger component, tetrahydrofuran in this case.

Prausnitz (1998) suggests that the nature of a system and the extent of departure from ideality can be gauged from a plot of the ratio of the logarithm of activity coefficient.
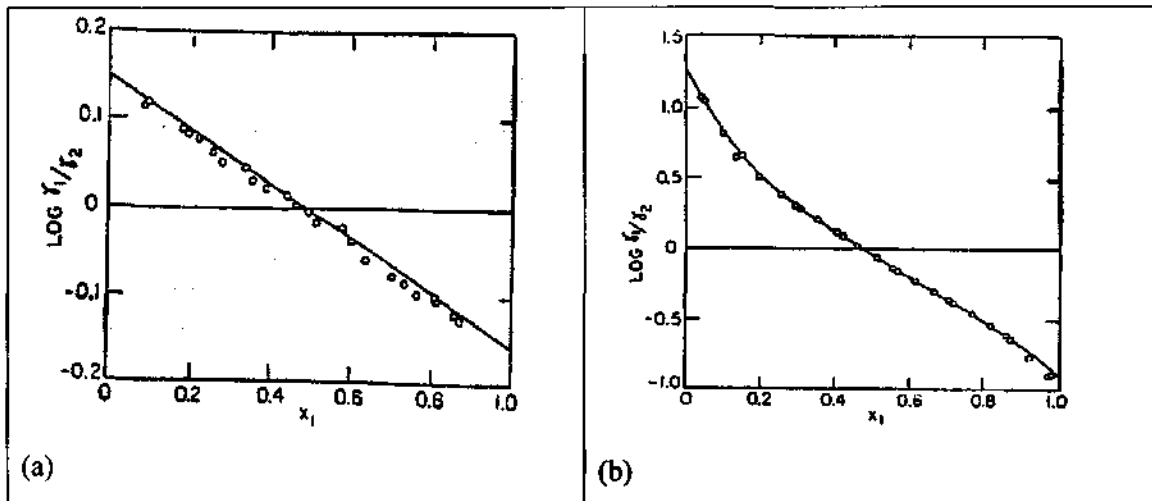
Figure 5-3 Logarithm of the ratio of activity coefficients versus mole fraction for various systems

(a) A simple system (b) A complex system (Prausnitz, 1998)

Simple systems as illustrated in **Figure 5-3a** result in a straight line. As one encounters more complex systems (**Figure 5-3b**) the line becomes a curve and in the case of more complex systems a point of inflection becomes apparent.

For asymmetric systems the straight line also becomes a curve. However if the logarithm of the ratio of the activity coefficient is plotted against the surface fraction, a straight line is obtained once more. This can be seen in **Figure 5-4**.

Figure 5-4 also shows the effect of flexibilisation of the mole fraction (FlexQUAC). This modification can be considered as physically unrealistic. When plotted as function of surface fraction, more complex systems usually show a symmetric deviation from linearity.

Based on these illustrations a flexibilisation of the UNIQUAC model based on a more adequate concentration scale like the surface fraction should lead to a better performance of the FlexQUAC model.
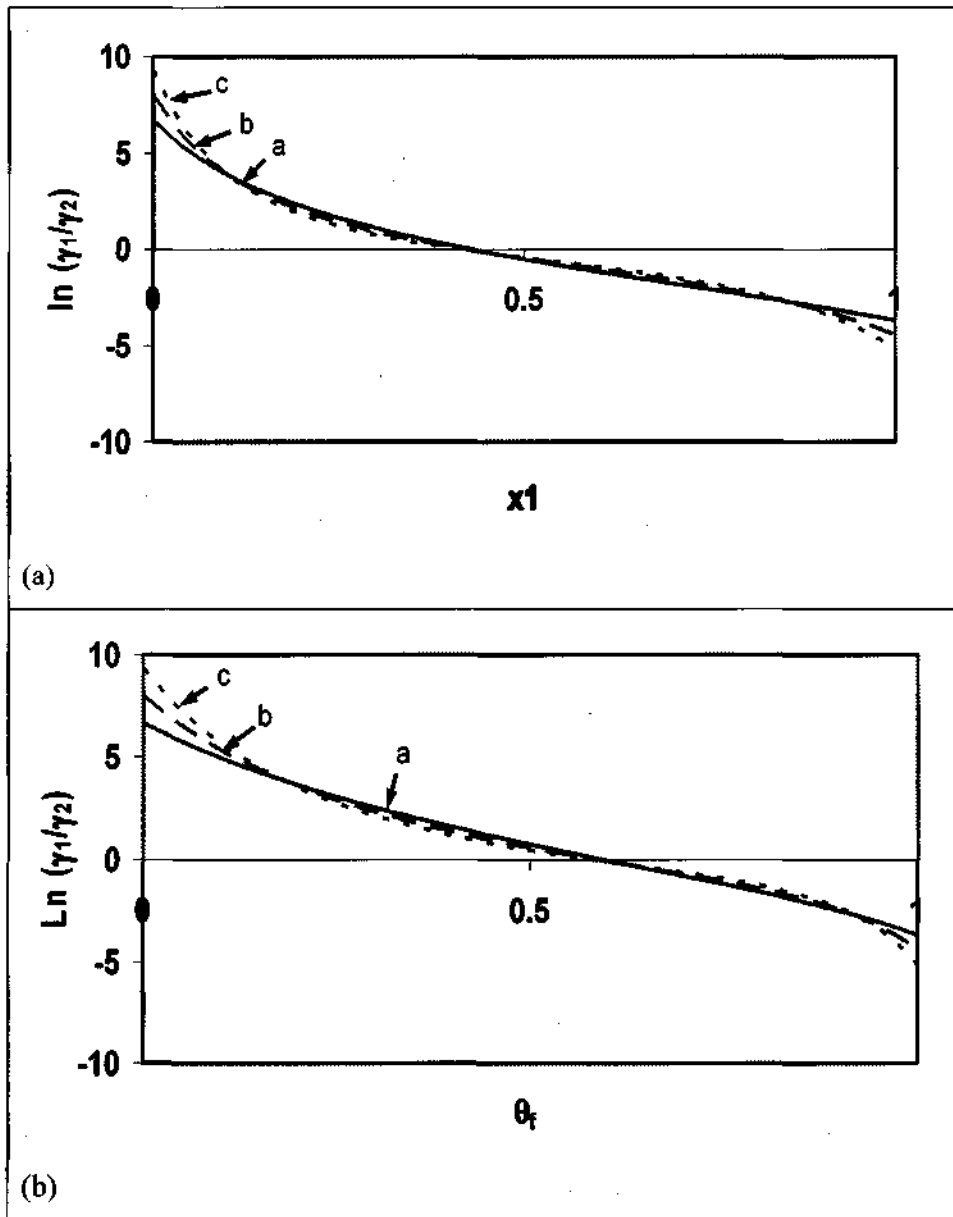
Figure 5-4 Logarithm of the ratio of activity coefficient versus (a) mole fraction (b) surface fraction for the purposes of comparison for varying values of the δ_{12} = (a) 0, (b) 0.2, (c) 0.4

5.3 Some Notes on the Combinatorial Part of UNIQUAC

In the UNIQUAC G^E model, G^E is calculated from two parts: the combinatorial and the residual part. The combinatorial part accounts for the size and shape differences of the molecules in the mixture while the intermolecular force interactions are accounted for in the residual part of the model.

$$G^E = G^E_{comb} + G^E_{res} \qquad\qquad \textbf{5-1}$$

The combinatorial part describes the G^E of a random athermal mixture via the Guggenheim-Staverman expression. The required relative Van der Waals surface and volume parameters r and q are usually calculated from the surfaces and volumes of the constituent structural groups of the molecules. In the case of for example n-alkanes, this yields the surface of stretched molecules, which are entropically improbable. Alkanes with a chain length of more than 6 or 7 carbon atoms will most likely be present in more or less spherical conformations. This means that while the volume parameter r increases linearly with chain length, the surface parameter should increase with chain length to the power of ⅔. Kikic et al (1980) empirically modified the combinatorial expression using this exponent for a better description of activity coefficients at infinite dilution of asymmetric systems and were able to describe retention times in gas chromatography. This modification was later carefully analyzed by Weidlich and Gmehling (1987), who found that an exponent of ¾ yields a nearly perfect description of asymmetric alkane-alkane systems. Thomas and Eckert (1984) had obtained similar results. These modifications are of great importance for group contribution methods based on the UNIQUAC equation (mod. UNIFAC, PSRK). In the case of the regression of binary mixture data of a single system, shortcomings of the combinatorial part can easily be compensated by the residual contribution.

Here the modification of Weidlich and Gmehling is used in order to achieve a more realistic separation of entropic and enthalpic effects:

$$\frac{G^E_{comb}}{RT} = \sum_i x_i \ln(\Phi'_i / x_i) + \frac{z}{2} \sum_i q_i x_i \ln\left(\frac{\Theta_i / x_i}{\Phi_i / x_i}\right) \qquad 5\text{-}2$$

with

$$\Theta_i / x_i = q_i \bigg/ \sum_j q_j x_j \qquad 5\text{-}3$$

$$\Phi_i / x_i = r_i \bigg/ \sum_j r_j x_j \qquad 5\text{-}4$$

$$\Phi'_i / x_i = \left(r_i^{3/4} x_i\right) \bigg/ \sum_j r_j^{3/4} x_j \qquad 5\text{-}5$$

5.4 Derivation of the FlexQUAC-Q Model

5.4.1 Re-formulation of the UNIQUAC-Equation

It is assumed in this study that the combinatorial part correctly describes the entropic effects of mixing. The combinatorial term of the UNIQUAC model is as follows:

$$\frac{G^E_{comb}}{RT} = \sum_i x_i \ln(\Phi'_i / x_i) + \frac{z}{2} \sum_i q_i x_i \ln\left(\frac{\Theta_i / x_i}{\Phi_i / x_i}\right) \qquad 5\text{-}6$$

Therefore flexibilisation should only be applied to the residual part. The residual part as such contains the binary parameters that are usually obtained from the regression of experimental data. In the case of molecules of different sizes, $G^E{}_{residual}$ (x) is a very unsymmetrical function. If $G^E{}_{residual}$ is plotted as a function of surface fraction then it is usually symmetrical. The surface fraction, Θ is defined as:

$$\Theta(\vec{x}) = \frac{x_i q_i}{\displaystyle\sum_{i=1}^{n} x_i q_i} \qquad 5\text{-}7$$

where x - liquid composition

     q - pure component molecular structure constants

     n - no. of components in the mixture

To avoid intercorrelation with the binary interaction parameters and to keep the number of adjustable parameters small, the FlexQUAC transformation function described in Chapter 3 should be symmetric ($\delta_{i,j} = \delta_{j,i}$). This can be achieved by applying the transformation to the surface fraction $\Theta(x)$ instead of the mole fraction x:

$$G^E_{residual} = G^E\left(\bar{\Theta}(\bar{x})\right) \qquad - \qquad \text{UNIQUAC}$$

$$G^E_{residual} = G^E \bar{f}\left(\left(\bar{\Theta}(\bar{x})\right)\right) \qquad - \qquad \text{FlexQUAC-Q}$$

The resulting model is called FlexQUAC-Q.

As a first step in the process of modifying the surface fraction, the UNIQUAC equation had to be defined as a function of surface fraction. As G^E is a molar property, the mole fraction as a variable cannot be completely removed from the expressions as the resulting equation could then not relate to a mole of mixture anymore. This problem was overcome by separating the function into the product of two functions, the first as a function of liquid composition and the latter as a function of surface fraction. While the second part calculates G^E for a unit surface of mixture ($G_{res}^{E,q}$), the first part converts this to a mole of mixture. As the molar relative van der Waals surface q of a mixture can be calculated as $q = \sum_i x_i q_i$, it holds that

$$G_{res}^E = \sum_i x_i q_i \cdot G_{res}^{E,q}$$

The original expression for the residual part of UNIQUAC is given by the following equation:

$$\left(\frac{G^E}{RT}\right)_{residual} = -\sum_{i=1}^{n} x_i q_i \ln S_i \qquad \textbf{5-8}$$

where S is defined as:

$$S_i = \sum_j \Theta(\vec{x})_j \tau_{ji} \qquad \text{5-9}$$

This can be written as:

$$\left(\frac{G^E}{RT}\right)_{residual} = -\sum_{i=1}^{n} x_i q_i \sum_{i=1}^{n} \Theta_i \ln S_i = -f_1(\vec{x}) f_2(\bar{\Theta}(\vec{x})) \qquad \text{5-10}$$

Thus the complete model FlexQUAC-Q equation is given by:

$$G^E = G^E_{comb} + RT \, f_1(\vec{x}) \, f_2(\vec{f}(\bar{\Theta}(\vec{x}))) \qquad \text{5-11}$$

5.4.2. Derivation of the Activity Coefficient Expressions

From **Equation 5-11** an expression for the activity coefficients had to be derived. The activity coefficients are defined as the partial molar derivative of the Gibbs energy.

$$RT \ln \gamma_i = \left(\frac{\partial(n_T G^E)}{\partial n_i}\right)_{T,P,n_{j \neq i}} \qquad \text{5-12}$$

or $\qquad \ln \gamma_i = \left(\frac{\partial(n_T Q)}{\partial n_i}\right)_{T,P,n_{j \neq i}}$ with $\qquad Q = \frac{G^E}{n_T RT}$ and $n_T = \sum_i n_i$

Here the total mole number n_T is equal to 1. As Q can be expressed by the sum, $Q_{comb} + Q_{res}$, both $\ln \bar{\gamma}_{comb}$ and $\ln \bar{\gamma}_{res}$ can be derived separately and $\ln \gamma_i$ can be expressed as:

$$\ln \gamma_i = \ln \gamma_{comb,i} + \ln \gamma_{res,i} \qquad \textbf{5-13}$$

The expression for $\ln \gamma_{comb,i}$ can be found in textbooks as

$$\ln \gamma_{comb,i} = 1 - \frac{\Phi_i'}{x_i} + \ln \frac{\Phi_i'}{x_i} + 5q_i \left(\ln \frac{\Theta_i}{\Phi_i} + \frac{\Phi_i}{\Theta_i} - 1 \right) \qquad \textbf{5-14}$$

In the case of the residual expression, the following expression can be used to determine the partial molar derivative.

$$\ln \gamma_{res,i} = Q_{res} + \left(\frac{\partial Q_{res}}{\partial n_i} \right)_{T,P,n_{j\ne i}}$$

$$= Q_{res} + \left(\frac{\partial \left(f_1(\vec{x}) \, f_2(\vec{f}(\vec{\Theta}(\vec{x}))) \right)}{\partial n_i} \right)_{T,P,n_{j\ne i}} \qquad \textbf{5-15}$$

The expression for $\left(\partial Q_{res} / \partial n_i \right)$ can easily be derived with the use of the chain rule of differentiation for the case of the UNIQUAC equation:

$$\left(\frac{\partial Q_{res}}{\partial n_i} \right)^T = \left(\frac{\partial f_1(\vec{x})}{\partial n_i} \right)^T f_2(\vec{\Theta}) + f_1(\vec{x}) \left(\frac{\partial f_2(\vec{\Theta})}{\partial \Theta_j} \right)^T \left(\frac{\partial \Theta_j}{\partial n_i} \right) \qquad \textbf{5-16}$$

where $\left(\partial Q_{res} / \partial n_i \right)$, $\left(\partial f_1(\vec{x}) / \partial n_i \right)$ and $\left(\partial f_2(\vec{\Theta}) / \partial \Theta_j \right)$ are vectors and the functions $f_1(\vec{x})$ and $f_2(\vec{x})$ are scalars. $\left(\partial \Theta_j / \partial n_i \right)$ is the gradient (Jacobian matrix $\nabla \Theta$) of the surface fraction vector $\vec{\Theta}$ with respect to the mole numbers.

However, in case of the FlexQUAC-Q model the function f_2 is now a function of $f(\vec{\Theta})$. In this case ∇f replaces $\nabla \Theta$ in the expression above and the new expression is:

$$\left(\frac{\partial Q_{res}}{\partial n_i} \right)^T = \left(\frac{\partial f_1(\vec{x})}{\partial n_i} \right)^T f_2(\vec{\Theta}) + f_1(\vec{x}) \left(\frac{\partial f_2(\vec{f}(\vec{\Theta}))}{\partial f(\vec{\Theta})_j} \right)^T \left(\frac{\partial f(\vec{\Theta})_j}{\partial n_i} \right) \qquad \textbf{5-17}$$

This leads to the following result for the residual activity coefficient:

$$\ln \gamma_{res,i} = Q_{res}(\bar{f}(\bar{\Theta})) + \left(\frac{\partial f_1(\bar{x})}{\partial n_i}\right)^T f_2(\bar{\Theta}) + f_1(\bar{x})\, \overline{\nabla f}^T \left(\frac{\partial f_2(\bar{f}(\bar{\Theta}))}{\partial f(\bar{\Theta})}\right)_{T,P,n_{j\neq i}} \qquad \text{5-18}$$

5.4.3 Derivation of Relevant Vectors

Having derived the activity coefficient equations, the relevant vectors and the Jacobian matrices can be derived.

The function $f_1(\bar{x})$ was rewritten in terms of mole numbers using the definition:

$$x_i = \frac{n_i}{n_T} \qquad \text{5-19}$$

where n_i refers to the mole number of component i.

Accordingly, $f_1(\bar{x})$ now becomes

$$f_1(\bar{x}) = \frac{1}{\sum_i n_i} \sum_i n_i q_i = f_{1a}(\bar{n}) f_{1b}(\bar{n}) \qquad \text{5-20}$$

In order to differentiate the above expression, the product rule was applied. The function $f_1(\bar{x})$ was segregated into two parts, $f_{1a}(\bar{n})$ and $f_{1b}(\bar{n})$ as defined above. Differentiation of the individual fragments yielded the following expressions.

$$\frac{\partial f_{1a}(n_i)}{\partial n_j} = -\frac{1}{\sum_i n_i^2}$$

5-21

$$\frac{\partial f_{1b}(n_i)}{\partial n_j} = q_j$$

5-22

Hence from the product rule,

$$\frac{\partial f_1(n_i)}{\partial n_j} = f_{1a}\frac{\partial f_{1b}}{\partial n_j} + f_{1b}\frac{\partial f_{1a}}{\partial n_j} = -\frac{\vec{q}}{\sum_i n_i} + \frac{\sum_i n_i q_i}{\left(\sum_i n_i\right)^2}$$

5-23

Re-substituting x for n leads to:

$$\frac{\partial f_1(\vec{n})}{\partial \vec{n}} = \sum_i (x_i q_i) - \vec{q}$$

5-24

The output from this derivation is a vector with the number of elements corresponding to the number of components contained in the mixture in question.

The scalar function $f_2(\vec{\Theta})$ is differentiated with respect to the surface fraction vector and yields a vector of partial derivatives.

The function $f_2(\vec{\Theta})$ can be rewritten as a product of two functions.

$$f_2(\vec{\Theta}) = \sum_i \Theta_i \ln \sum_k \Theta_k \tau_{kj} = \sum_i f_{2a}(\vec{\Theta}) f_{2b}(\vec{\Theta})$$

5-25

The product rule was used to differentiate these two functions. The differentiation was performed with respect to Θ_j.

$$\frac{df_{2a}(\Theta)}{d\Theta} = \frac{d\Theta_i}{d\Theta j} \ln \sum_{k=1}^{n} \Theta_k \tau_{k,i} + \Theta_i \frac{d \ln \sum_{k=1}^{n} \Theta_k \tau_{k,i}}{d\Theta_j} \qquad \text{5-26}$$

The derivation has to account for two cases, when subscripts i and j are identical and when they are not. This is crucial to obtaining the correct analytical expression. First, the case of different subscripts is considered.

$$\frac{d\Theta_i}{d\Theta j} = 0 \qquad \text{5-27}$$

and:

$$\frac{d \ln \sum_{k=1}^{n} \Theta_k \tau_{k,i}}{d\Theta_j} = \frac{\tau_{j,i}}{\sum_{k=1}^{n} \Theta_k \tau_{k,i}} \qquad \text{5-28}$$

The derivative of $f_2(\bar{\Theta})$ for this case is

$$\frac{\Theta_i \tau_{j,i}}{\sum_{k=1}^{n} \Theta_k \tau_{k,i}} \qquad \text{5-29}$$

Now the case of identical subscripts is derived. Equation 5-26 is still applicable, however in this case i and j are identical. Now

$$\frac{d\Theta_i}{d\Theta_j} = 1 \qquad \text{5-30}$$

$$\frac{d \ln \sum_{k=1}^{n} \Theta_k \tau_{k,j}}{d\Theta_j} = \frac{\tau_{i,j}}{\sum_{k=1}^{n} \Theta_k \tau_{k,j}} \qquad \text{5-31}$$

Thus in this case the derivative is

$$\ln \sum_{k=1}^{n} \Theta_k \tau_{k,j} + \frac{\Theta_i \tau_{i,j}}{\sum_{k=1}^{n} \Theta_k \tau_{k,j}} \qquad \text{5-32}$$

Thus combining both cases, the complete expression is obtained.

$$\frac{df_2(\vec{\Theta})}{d\vec{\Theta}} = \ln \sum_{k=1}^{n} \Theta_k \tau_{k,j} + \frac{\Theta_i \tau_{i,j}}{\sum_{k=1}^{n} \Theta_k \tau_{k,j}} + \sum_{i=1}^{n} \left(\frac{\Theta_i \tau_{j,j}}{\sum_{k=1}^{n} \Theta_k \tau_{k,i}} \right) - \frac{\Theta_i \tau_{i,j}}{\sum_{k=1}^{n} \Theta_k \tau_{k,j}} \qquad \text{5-33}$$

Note that the last term is required to remove the incorrect derivation for i = j introduced by the third term.

Now $\tau_{i,i}$ is 1 and the final result for the above expression is:

$$\frac{df_2(\vec{\Theta})}{d\Theta_j} = \ln \sum_{k} \Theta_k \tau_{k,j} + \sum_{k} \left(\frac{\Theta_i \tau_{j,k}}{\sum_{i} \Theta_i \tau_{i,k}} \right) \qquad \text{5-34}$$

5.4.4 Derivation of Jacobian Matrices

5.4.4.1 Analytical derivation for the differentiation of $\bar{\Theta}$ with respect to the mole number vector $\bar{n}$ (Jacobian matrix for UNIQUAC)

Next the Jacobian matrix, $\nabla\Theta$ was derived. The element of the surface fraction vector $\bar{\Theta}$ is defined as

$$\Theta(\bar{x})_i = \frac{n_i q_i}{\sum_k n_k q_k} \qquad 5\text{-}35$$

This expression was partially differentiated with respect to n_j and mole numbers were back substituted for mole fractions. Once again two cases have to be investigated. For the case of identical subscripts, the following expression was obtained:

$$\frac{d\Theta_i}{dn_i} = \frac{\sum_i x_i q_i q_j - x_i q_i q_j}{\left(\sum_i x_i q_i\right)^2} \qquad 5\text{-}36$$

This expression corresponds to the elements along the main diagonal of the matrix. The case of different subscripts yielded the following expression:

$$\frac{d\Theta_i}{dn_j} = \frac{-x_i q_i^2}{\left(\sum_k x_k q_k\right)^2} \qquad 5\text{-}37$$

This expression generated the rest of the elements for the matrix.

With the Jacobian matrix expressions derived, the derivative of $f_2(\vec{\Theta})$ can be determined. The product of the Jacobian matrix and the vector derivative of the surface fraction corresponds to the derivative of $f_2(\vec{n})$ with respect to mole numbers.

With the derivatives for both functions computed, the derivative of G^E with respect to mole numbers could now be undertaken using **Equation 5-16**.

5.4.4.2 Analytical derivation for the differentiation of $\bar{f}_{(\Theta)}$ with respect to the mole number vector $\vec{n}$ (Jacobian matrix for FlexQUAC-Q)

In FlexQUAC-Q, the modification of the UNIQUAC equation is only applied to the surface fraction, Θ. The modified surface fraction is

$$\bar{f}(\vec{\Theta}) = \frac{x_i q_i \left(1 + \left(\sum_k x_k q_k \delta_{i,k}\right)\right)}{\sum_k x_k q_k \left(1 + \left(\sum_l x_l q_l \delta_{k,l}\right)\right)} \qquad \text{5-38}$$

Equation 5-24 is still valid in this case. **Equation 5-34** now becomes

$$\frac{df_2(f(\vec{\Theta}))}{df(\vec{\Theta})_j} = \ln \sum_k f(\vec{\Theta})_k \tau_{k,j} + \sum_k \left(\frac{f(\vec{\Theta})_j \tau_{j,k}}{\sum_i f(\vec{\Theta})_i \tau_{i,k}}\right) \qquad \text{5-39}$$

The Jacobian matrix has to be derived. Thus **Equation 5-38** is differentiated with respect to mole numbers. As before, the surface fraction has to be first converted to an expression in terms of mole numbers.

$$f(\bar{\Theta})_i = n_i q_i \left(\frac{n_T + \sum_k n_k q_k \delta_{i,k}}{\sum_k n_k q_k \left(n_T + \sum_l n_l q_l \delta_{k,l} \right)} \right) = f_a \left(\frac{f_b}{f_c} \right) \qquad \text{5-39}$$

The expression is fragmented to facilitate the differentiation. Hence the differentiation result is

$$\frac{df(\bar{\Theta})}{dn_j} = \frac{df_a}{dn_j} \left(\frac{f_b}{f_c} \right) + \frac{d}{dn_j} \left(\frac{f_b}{f_c} \right) f_a \qquad \text{5-40}$$

Once again, there are two cases to consider with respect to the subscripts. The case of identical subscripts is considered first. In this case i = j. The differentiation for term f_a is

$$\frac{df_a}{dn_j} = q_i \qquad \text{5-41}$$

The differentiation of f_b / f_c follows the quotient rule, in which case the derivative is

$$\frac{d}{dn_j} \left(\frac{f_b}{f_c} \right) = \frac{f_b' f_c - f_c' f_b}{f_c^2} \qquad \text{5-42}$$

The differentiation of f_b leads to the following expression

$$\frac{df_b}{dn_j} = 1 + q_i \delta_{i,j} \qquad \text{5-43}$$

However the term $\delta_{i,j}$ is zero in this case and the second term of Equation 5-43 vanishes.

The differentiation of f_c results in

$$\frac{df_c}{dn_j} = q_i\left(n_T + \sum_l n_l q_l \delta_{l,j}\right) + \sum_k n_k q_k\left(1 + q_j \delta_{k,j}\right) \qquad \text{5-44}$$

Combining equations from 5-41 to 5-44 and substituting the result into **Equation 5-40**, one obtains the complete expression for the Jacobian elements for the case of identical subscripts. Using **Equation 5-40**, the complete expression in terms of mole fractions is

$$\frac{df(\vec{\Theta})_i}{dn_j} = q_i\left(\frac{1 + \sum_k x_k q_k \delta_{i,k}}{\sum_k x_k q_k\left(1 + \sum_l x_l q_l \delta_{k,l}\right)}\right) + x_i q_i \left(\frac{\sum_k x_k q_k\left(1 + \sum_l x_l q_l \delta_{k,l}\right)}{\sum_k x_k q_k\left(1 + \sum_l x_l q_l \delta_{k,l}\right)^2}\right) \ldots$$

$$\ldots + x_i q_i \left(\frac{\left(1 + \sum_k x_k q_k \delta_{i,k}\right)\left(q_i\left(1 + \sum_l x_l q_l \delta_{i,l}\right) + \sum_k x_k q_k\left(1 + q_j \delta_{k,j}\right)\right)}{\sum_k x_k q_k\left(1 + \sum_l x_l q_l \delta_{k,l}\right)^2}\right) \qquad \text{5-45}$$

For the case of different subscripts, the derivative of f_a with respect to mole numbers is reduced to zero. In this case the differentiation of f_b and f_c yield the following expressions.

$$\frac{df_b}{dn_j} = 1 + q_j \delta_{i,j} \qquad \text{5-46}$$

$$\frac{df_c}{dn_j} = q_j\left(n_T + \sum_l n_l q_l \delta_{j,l}\right) + \sum_k n_k q_k\left(1 + q_j \delta_{k,j}\right) \qquad \text{5-47}$$

Thus, the complete expression in this case is

$$\frac{df(\vec{\Theta})_i}{dn_j} = x_i q_i \left(\frac{\left[1 + q_j \delta_{i,j} \left(\sum_k x_k q_k \left(1 + \sum_l x_l q_l \delta_{k,l}\right)\right)\right] - \left(q_j \left(1 + \sum_l x_l q_l \delta_{j,l}\right) + \sum_k x_k q_k \left(1 + q_j \delta_{k,j}\right)\right)\left(1 + \sum_k x_k q_k \delta_{i,k}\right)}{\sum_k x_k q_k \left(1 + \sum_l x_l q_l \delta_{k,l}\right)^2} \right)$$

5-48

Using the auxiliary variables

$$S^A = 1 + \sum_j x_j q_j \delta_{i,j}$$

$$S^B = \sum_k x_k q_k S_k^A$$

$$S^C = \sum_k x_k q_k S_k^{A^2}$$

$$S_i^D = \sum_k x_k q_k \left(1 + q_i \delta_{k,j}\right)$$

this leads to

For $i = j$

$$\frac{df(\vec{\Theta})_i}{dn_j} = q_i \left(\frac{S_i^A}{S^B} + \frac{x_i}{S^C} \left(S^B + S_i^A \left(q_i S_i^A + S_i^D\right)\right) \right) \qquad \textbf{5-49}$$

For $i \neq j$

$$\frac{df(\vec{\Theta})_i}{dn_j} = x_i q_i \frac{S^B \left(1 + q_j \delta_{i,j}\right) - S_i^A \left(q_j S_j^A + S_j^D\right)}{S^C} \qquad \textbf{5-50}$$

With this derivation complete the activity coefficient can be computed from Equation **5-18**.

5.5 Calculation of Activity Coefficients using the Group Contribution Approach (FlexFAC)

In UNIFAC, the extension of the UNIQUAC model to group contribution, the residual activity coefficient of component i ($\gamma_{res,i}$) is calculated from the difference of the group activity coefficients of the structural groups in the mixture ($\vec{\Gamma}^M$) and the pure component i ($\vec{\Gamma}^{(i)}$):

$$\ln \gamma_{res,i} = \sum_k v_k^{(i)} (\Gamma_k^M - \Gamma_k^{(i)}) \quad \text{5-51}$$

where $v_k^{(i)}$ denotes the frequency of group k in component i. The group activity coefficients are calculated in the same way as in UNIQUAC. In FlexFAC, analogous to Equation 5-18 the following expressions are used:

$$\ln \Gamma_i^M = Q_{res}^M (\vec{f}(\vec{\Theta}^M)) + \left(\frac{\partial f_1(\vec{X}^M)}{\partial n_i} \right)^T f_2(\vec{\Theta}^M)$$
$$+ f_1(\vec{X}^M) \nabla \vec{f}^T \left(\frac{\partial f_2(\vec{f}(\vec{\Theta}^M))}{\partial f(\vec{\Theta}^M)} \right)_{T,P,n_{j\neq i}} \quad \text{5-52}$$

$$\ln \Gamma_k^{(i)} = Q_{res}^{(i)} (\vec{f}(\vec{\Theta}^{(i)})) + \left(\frac{\partial f_1(\vec{X}^{(i)})}{\partial n_k} \right)^T f_2(\vec{\Theta}^{(i)})$$
$$+ f_1(\vec{X}^{(i)}) \nabla \vec{f}^T \left(\frac{\partial f_2(\vec{f}(\vec{\Theta}^{(i)}))}{\partial f(\vec{\Theta}^{(i)})} \right)_{T,P,n_{j\neq k}} \quad \text{5-53}$$

where $\vec{X}^M$ and $\vec{X}^{(i)}$ are the group fraction vectors in the mixture and the pure component i. Q_{res}^M and $Q_{res}^{(i)}$ are calculated analogously to Equation 5-10 as

$$Q_{res}^M = f_1(\vec{X}^M) \, f_2(\vec{\Theta}(\vec{X}^M)) \quad \text{5-54}$$

$$Q_{res}^{(i)} = f_1(\vec{X}^{(i)}) \, f_2(\vec{\Theta}(\vec{X}^{(i)})) \quad \text{5-55}$$

5.6 Implementation of FlexQUAC-Q in Fortran

5.6.1 Preliminary Check

As a preliminary test, before any programming of the activity coefficient calculation subroutine was undertaken, the FlexQUAC and the re-formulated UNIQUAC model equations were checked against the original UNIQUAC model using a numerical example in MathCAD (*Appendix A – FlexQUAC-Q Derivation*).

Here the residual parts of the original UNIQUAC and re-formulated UNIQUAC equations were compared to see if the same result was obtained for any given input data. Hence, the both model equations were defined. The necessary derivatives and Jacobian matrices were defined. In addition, to further establish the accuracy of the analytical derivatives, numerical derivatives in terms of elementary finite difference methods were used as a rigorous checking device. This was a necessary exercise that would serve to identify analytical derivatives that were inaccurately defined.

For the reformulated version of the UNIQUAC model, the analytical derivatives of $f_1(\vec{x})$ were first determined in terms of mole numbers and then back substituted in terms of mole fractions. This exercise had a twofold benefit, initially as a checking device to identify any errors. This derivative was then compared to a numerical derivative. Since $f_1(\vec{x})$ is a function of mole fraction, the numerical derivative was computed in this way:

$$\frac{df_1(\vec{x})}{dn}\bigg|_{numerical} = \frac{f_1(\vec{x}+eps)-f_1(\vec{x}-eps)}{2(eps)} \quad \text{5-56}$$

Here eps refers to a tolerance of 10^{-8}. When the analytical and numerical derivatives are compared the difference should ideally be less then this tolerance.

Next, the derivative $f_2(\vec{\Theta})$ was derived together with its appropriate numerical derivative. In this case the numerical derivative is computed by a change in the surface fraction, Θ.

$$\frac{df_2(\bar{\Theta})}{d\Theta}\bigg|_{numerical} = \frac{f_2(\bar{\Theta}+eps)-f_2(\bar{\Theta}-eps)}{2(eps)} \qquad \text{5-57}$$

Here the both derivatives were found to be within acceptable tolerance.

As a next step, the Jacobian matrix ($\nabla\Theta$) is computed. Here the numerical derivative is expressed as:

$$\frac{d\Theta(\bar{x})}{dn}\bigg|_{numerical} = \frac{\Theta(\bar{x}+eps)-\Theta(\bar{x}-eps)}{2(eps)} \qquad \text{5-58}$$

Next, the derivatives of $f_2(\bar{\Theta})$ by mole numbers were computed analytically and numerically. Hence the residual activity coefficients were computed analytically and numerically. The comparison of the two results was good.

Similarly, the derivatives of $f_2(\bar{f}(\Theta(\bar{x})))$ by $f(\Theta)$ and the Jacobian ∇f were computed. In this case the following numerical derivatives apply:

$$\frac{df_2(\bar{f}(\bar{\Theta}))}{d\bar{f}(\bar{\Theta})}\bigg|_{j\ numerical} = \frac{f_2(\bar{f}(\bar{\Theta})+eps)-f_2(\bar{f}(\bar{\Theta})-eps)}{2(eps)} \qquad \text{5-59}$$

$$\frac{d\bar{f}(\bar{\Theta})}{dn}\bigg|_{numerical} = \frac{\bar{f}\left(\bar{\Theta}\left(\frac{\bar{x}-eps}{1-eps}\right)\right)-\bar{f}\left(\bar{\Theta}\left(\frac{\bar{x}+eps}{1+eps}\right)\right)}{2(eps)} \qquad \text{5-60}$$

Thereafter the numerical and analytical residual activity coefficients were compared and found to have negligible difference.

5.6.2 Simplifications in MathCAD

Some of the derived expressions were far too complex to be programmed as they appear in the previous MathCAD file (*Appendix A – A2 FlexQUAC-Q Derivation*). The derivatives and matrices require careful computation. In addition, the distinct disadvantages to programming such long expressions in FORTRAN were:

- Difficulty in computing and programming
- Difficulty to trace through code and identify errors

In short this approach was not an intelligent one, since troubleshooting would be problematic. A more prudent solution to this would be to simplify some the more complex expressions obtained in the derivations. This would ease the programming and enable one to manage the computation of complex expressions by fragmenting them. This would also help to isolate errors in computations to specific parts of expressions.

This simplification was undertaken in MathCAD once more using the same numerical example. (*cf Appendix A – A3 Fortran Implementation*) The purpose of computing the simplified formulae simultaneously with the new expressions were for comparison and to ensure that no errors were encountered in this simplification process.

The following variables (cf **Table 5-1**) were introduced as they appear frequently in other larger expressions and from a programming perspective, the computation of these terms would reduce the number of variables used, eliminate repetition and redundancy and facilitate better understanding of the code.

These variables reduce the complexity of some terms:

1) Definition of f_2

$$f2 := \sum_{i=1}^{n} x_i \cdot thetfx_i \cdot \ln\left(Sf_i\right)$$

2) Derivative of f_2 by $f(\Theta)$

$$df2_i := sum1_i + \ln\left(Sf_i\right)$$

Variable	Definition	Actual Equation
thetafx	$\text{thetafx}_i := Q_i \cdot \left(1 + \displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)$	$q_i \left(1 + \displaystyle\sum_{k=1}^{n} x_k q_k \delta_{i,k} \right)$
sthetfx	$\text{sthetfx} := \displaystyle\sum_i x_i \cdot \text{thetafx}_i$	$\displaystyle\sum_i x_i q_i \left(1 + \displaystyle\sum_{k=1}^{n} x_k q_k \delta_{i,k} \right)$
thetfx	$\text{thetfx}_i := \dfrac{\text{thetafx}_i}{\text{sthetfx}}$	$\dfrac{q_i \left(1 + \left(\displaystyle\sum_k x_k q_k \delta_{i,k} \right) \right)}{\displaystyle\sum_k x_k q_k \left(1 + \left(\displaystyle\sum_l x_l q_l \delta_{k,l} \right) \right)}$
Sf	$\text{Sf}_i := \displaystyle\sum_{j=1}^{n} x_j \cdot \text{thetfx}_j \cdot \tau_{j,i}$	$\displaystyle\sum_j f(\bar\Theta)_j \tau_{j,i}$
Sum1	$\text{sum1}_j := \displaystyle\sum_{i=1}^{n} \dfrac{x_i \cdot \text{thetfx}_i \cdot \tau_{j,i}}{\text{Sf}_i}$	$\displaystyle\sum_i \dfrac{f(\bar\Theta)_j \tau_{j,i}}{\displaystyle\sum_j f(\bar\Theta)_j \tau_{j,i}}$
SJ	$\text{SJ}_i := \displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i} \right)$	$\displaystyle\sum_{k=1}^{n} x_k q_k \left(1 + q_i \delta_{k,i} \right)$

Table 5-1 Summary of convenience variables that were defined due to the high frequency in which they appear in other definitions

JacobianFQ1 :=	for $i \in 1..n$
	for $m \in 1..n$
	$\displaystyle Jac_{m,i} \leftarrow -\left(thetfx_m \cdot x_i\right) - \frac{SJ_m \cdot x_i}{sthetfx} + \frac{Q_i \cdot x_i}{thetfx_i \cdot sthetfx}$
	$\displaystyle Jac_{m,i} \leftarrow Jac_{m,i} + \frac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{thetfx_i \cdot sthetfx}$ if $i \neq m$
	$Jac_{m,i} \leftarrow Jac_{m,i} + 1$ if $i = m$
	$Jac_{m,i} \leftarrow Jac_{m,i} \cdot thetfx_i$
	Jac

The above expressions were used to compute the Jacobian, ∇f. The Jacobian was analysed carefully to find repeated terms. As explained earlier, the differentiation to obtain the Jacobian ∇f involves two cases, one with identical subscripts and the latter which involve different subscripts. Hence the above expressions presents two different expressions for each case (one for i=m (main diagonal) and i≠m (rest of the matrix)). It was observed that some terms were common to both cases.

The Jacobian simplification follows a sequence of steps:

- *Step 1*: The terms in the two expressions are replaced by other variables

- *Step 2*: A new variable is defined, SJ since it appears in both cases

- *Step 3*: SJ is substituted into the expression and the term A is multiplied into the bracketed expression. In this case $A(B/C) = \Theta f$ as defined earlier

- *Step 4*: Θf is multiplied into the bracketed terms

- *Step 5*: The term $Q(B/C)$ is repeated and can be simplified to Θ_f / x

- *Step 6*: The term A/C is repeated and can be replaced by Θ_f / B

- *Step 7*: It is best to eliminate the mole fraction from the denominator of certain terms by the use of other variables to avoid division by zero. At this stage a more convenient variable can be introduced, $\Theta_{fx} \cdot \Theta_{fx} = \Theta/x$, therefore $\Theta/B = \Theta_{fx} x/B$.

- *Step 8*: It is now evident that a common factor of Θ_{fx} can be removed from both expression. Three terms are common to both expressions:

$$\frac{x_i}{B(x,Q,n)_i} - \Theta f(x,Q,n)_i \cdot x_i - \frac{SJ(x,Q,n)_i \cdot x_i}{C(x,Q)}$$

- *Step 9*: Hence, from careful observation and variable manipulation more compact expressions were derived for the Jacobian. As a final check, the simplified result is compared with the initial expression. The two correspond exactly.

5.6.3 Activity Coefficient Calculation Subroutine in FORTRAN

This section presents a description detailing the programming in the Fortran subroutine. To allow better understanding of the programming, the subroutine makes use of the variables previously defined in the Mathcad file (*cf Appendix A – A3 Fortran implementation* and **Table 5-1**). Some new variables are also summarised in the **Table 5-2**.

An algorithm (**Figure 5-5**) is also presented here to elucidate the programming as contained in the subroutine. The subroutine accomplishes the task of computing the activity coefficient via six steps:

1) Get interaction parameters
2) Calculate combinatorial activity coefficient
3) Perform auxiliary calculations for the new FlexQUAC-Q model
4) Calculate relevant derivatives ($\frac{df_1}{dn}, \frac{df_2}{dn}$ and Jacobian matrix)
5) Calculate the residual activity coefficient
6) Calculate the FlexQUAC-Q activity coefficient

Variable	Meaning
Lmodcomb34	Logical Variable - Activates the ¾ exponent on the combinatorial term
Lmodcomb23	Logical Variable - Activates the ⅔ exponent on the combinatorial term
lmodqflex	Logical Variable - Activates the FlexQUAC-Q model
x	Mole fraction
t	Temperature
act	Activity Coefficient
he	Excess Enthalpy
cpe	Excess heat capacity
aij	Interaction parameter
alphaij	δ_{ij}
aijt	Temperature dependence of interaction parameter
tau	τ_{ij}
SXR	$\sum_i x_i r_i$
SXQ	$\sum_i x_i q_i$
SXQP	$\sum_i x_i q_i'$
F	$\dfrac{q_i}{\sum_i x_i q_i}$
V	$\dfrac{r_i}{\sum_i x_i r_i}$
Lgamc	$\ln \gamma_{\text{combinatorial}}$
sthetfx	$\sum_i x_i q_i \left(1 + \sum_j x_j q_j \delta_{i,j}\right)$

Variable	Meaning
Sumk_array	$\left(1+\sum_j x_j q_j \delta_{i,j}\right)$
Thetfx	$q_i\left(1+\sum_j x_j q_j \delta_{i,j}\right)$
df1dn	Derivative of f1 wrt mole numbers
Jac	Jacobian matrix
df2dn	Derivative of f2 wrt mole numbers
lgamr	$\ln\gamma_{residual}$

Table 5-2 Summary of the variables and their definitions used in the Fortran subroutine

$$\left(\text{START}\right)$$

Call procedure setpar – *cf Section 1 Appendix B:B1 Implementation in Fortran*

- Calculate interaction parameters for given temperature and their temperature derivative (aij, aijt & alphaij)

Calculate Combinatorial Part – *cf Section 2 Appendix B:B1 Implementation in Fortran*

$$SXQ = \sum_i x_i q_i$$

$$SXR = \sum_i x_i r_i$$

$$F_i = q_i / SXQ = q_i / \sum_i x_i q_i = \Theta_i$$

$$V_i = r_i / SXR = r_i / \sum_i x_i r_i = \Phi_i$$

$$C1 = -1 + \Phi_i / \Theta_i + \ln(\Theta_i / \Phi_i)$$

The user selects the choice for combinatorial term

(a = ¾, 2/3 or 1)

$$\lg amc = \ln \gamma_{comb} = 1 - \frac{r_i^a}{\sum_i x_i r_i^a} + a \ln\left(\frac{r_i^a}{\sum_i x_i r_i^a}\right) - 5q_i\left(-1 + \ln\frac{\Theta_i}{\Phi_i} + \frac{\Phi_i}{\Theta_i}\right)$$

Cf Section 3 Appendix B:1.1 Implementation in Fortran

$$Tau = \tau_{ij} = \exp\left(-\Delta u_{ij} / RT\right)$$

$$sumk\_array_i = 1 + \sum_j x_j q_j \delta_{ij}$$

$$thetfx_i = q_i\left(1 + \sum_j x_j q_j \delta_{ij}\right)$$

$$sthetfx = \sum_i x_i q_i\left(1 + \sum_j x_j q_j \delta_{ij}\right)$$

$$thethaf_i = \frac{thetfx_i}{sthetfx} = \frac{q_i\left(1 + \sum_j x_j q_j \delta_{ij}\right)}{\sum_i x_i q_i\left(1 + \sum_j x_j q_j \delta_{ij}\right)}$$

$$Sf_i = \sum_j x_j \tau_{ji} \, thethaf_i = \sum_j \frac{x_j q_j \left(1 + \sum_k x_k q_k \delta_{jk}\right)}{\sum_l x_l q_l \left(1 + \sum_k x_k q_k \delta_{kl}\right)} \tau_{ji}$$

Cf Section 4 Appendix B:1.1 Implementation in Fortran
: Compute f1, f2 & $G^E_{residual}$

$$sumk = \sum_i x_i q_i$$

$$f1 = -sumk$$

$$f2 = x_i thethaf_i \ln(Sf_i) = \frac{x_i q_i \left(1 + \sum_j x_j q_j \delta_{ij}\right)}{\sum_i x_i q_i \left(1 + \sum_j x_j q_j \delta_{ij}\right)} \ln(Sf_i)$$

$$G^E_{Re\,sidual} = f1 \times f2$$

Cf Section 5 - Appendix B:1.1 Implementation in Fortran
: Derivative of f1

$$df1dn_i = -q_i - f1$$

Cf Section 6 - Appendix B:1.1 Implementation in Fortran
: Derivative of f2

$$df2dthet_i = \frac{df2}{d\Theta} = \sum_j \frac{x_j q_j \left(1 + \sum_k x_k q_k \delta_{kj}\right) \tau_{ij}}{\dfrac{\sum_l x_l q_l \left(1 + \sum_k x_k q_k \delta_{kl}\right)}{Sf_j}} + \ln Sf_i$$

Cont…

Cf Section 7 - Appendix B:1.1 Implementation in Fortran

Compute Jacobian Matrix

$$SJ_i = \sum_j x_j q_j \left(1 + q_i \delta_{ji}\right)$$

$$Jac_{j,i} = \frac{q_i x_i}{q_i \left(1 + \sum_k x_k q_k \delta_{k,i}\right)} - \frac{x_i q_j \left(1 + \sum_k x_k q_k \left(1 + q_j \delta_{k,j}\right)\right)}{\sum_k x_k q_k \left(1 + q_i \delta_{k,i}\right)} - \frac{x_i \sum_k x_k q_k \left(1 + q_j \delta_{k,j}\right)}{\sum_k x_k q_k \left(1 + \sum_l x_l q_l \delta_{k,l}\right)}$$

If i = j (main diagonal of Jacobian Matrix) then $Jac_{j,i} = Jac_{j,i} + 1$

Else $Jac_{j,i} = Jac_{j,i} + \dfrac{x_i q_i}{q_i \left(1 + \sum_j x_j q_j \delta_{i,j}\right)}$

$$Jac_{j,i} = Jac_{j,i} \times \frac{q_i \left(1 + \sum_j x_j q_j \delta_{i,j}\right)}{\sum_k x_k q_k \left(1 + \sum_j x_j q_j \delta_{i,j}\right)}$$

Cf Section 8 Appendix B:1.1 Implementation in Fortran
: Derivative of f2 wrt mole numbers

$$df2dn_i = \frac{df2}{dn} = \sum_j Jac_{i,j} \times df2dthet_i$$

Cf Section 9: Compute Activity Coefficient

$$\lg amr_i = \ln \gamma_{residual} = f1 \times df2dn_i + f2 \times df1dn_i + G^E_{residual}$$

$$act_i = \gamma = \exp\left(\lg amr_i + \lg amc_i\right)$$

END

Figure 5-5 Algorithm detailing the implementation of the FlexQUAC-Q model in Fortran

5.6.4 Implementation in RECVAL

With the subroutine complete, it was added to RECVAL, the regression software tool. To allow the user to select the appropriate model, a subroutine matflex was programmed. This subroutine uses logical variables to activate a specific model. In addition, this subroutine allows the user to choose from the different variations of the UNIQUAC combinatorial term.

In **Figure 5-6** RECVAL's liquid mixture model program window is illustrated. In the section model flavours, the user is prompted to select the appropriate. The user can select one of these options:

1. modqflex   – This activates the FlexQUAC-Q model with original combinatorial term

2. modcomb34   - This activates the ¾ modification of the combinatorial term

3. modcomb23   - This activate the $\dfrac{2}{3}$ modification of the combinatorial term

The combinatorial term options can be selected in conjunction with the FlexQUAC-Q option. However, only one combinatorial variation can be selected at a time.
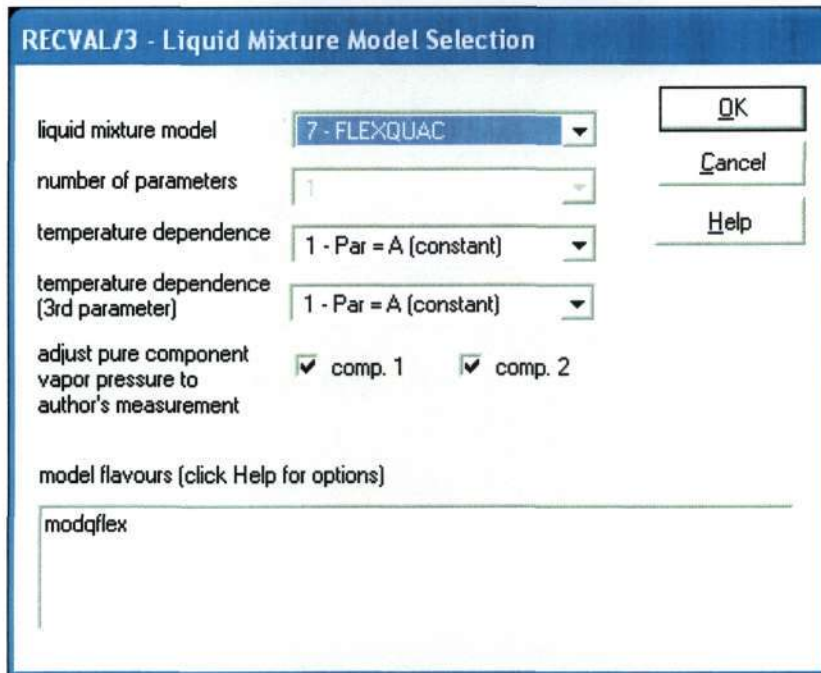
Figure 5-6 The RECVAL liquid selection model options, where the FlexQUAC-Q model can be enabled via the use of the control string, modqflex.

5.7 Implementation in Excel

With the use of macros, the regression of equilibrium data was possible. The programming code is contained in *Appendix B – B2 Implementation in Excel*. A simple algorithm is presented here to aid in the understanding of the regression procedure (**Figure 5-7**).

The program begins by reading data from the DDB and selecting the appropriate worksheet in Excel for data output. This Excel program makes use of a DDB add-in feature which a library of computational procedures utilised by the DDB software to perform various equilibrium data calculations.

The experimental equilibrium data is stored in a text file. This file is accessed and all the relevant data is displayed on the Excel worksheet. The file contains the DDB set no, component codes and

component names. It also enables the retrieval of pure component molecular constants, Antoine constants and the experimental VLE data.

A liquid model is then selected. Some preliminary calculations are then performed to obtain partial pressures and activity coefficients. The formulas for the calculation of pressure and the vapour mole fraction are set into the Excel worksheet.

An initial set of interaction parameters are set for the regression. The program first uses the UNIQUAC model. The interaction parameters are passed into a subroutine, datafit which contains the various criteria for the regression i.e. step width for each parameter, maximum number of iterations and stop criteria. This procedure then inputs these parameters into the simplex regression subroutine where the regression takes place. The results of the regression are displayed on the worksheet. The residual value from this regression is calculated from the sum of the deviation of the calculated pressure from the experimental pressure and is stored. The regression is made more robust by assigning a new set of parameters a further three times and performing the regression. In each instance the residual corresponding to the set of regressed parameters is stored. At the conclusion of the final regression, all the residuals are compared and the lowest residual corresponds to the optimal regression parameters. This set of parameters is used to regress the data once more.

Thereafter the infinite dilution activity coefficients and the relative absolute deviation in pressure are calculated.

The interaction parameters are reset and a similar procedure is used to regress the data using the FlexQUAC-Q model.

In the case of the ternary VLE data, the constituent binary VLE systems are considered first. These binary VLE systems are regressed and the parameters obtained are used in the calculation of the ternary VLE data.

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           ▼
```

- Read data from DDB
- Formats excel data sheet for output
- Selects either binary VLE regression or ternary VLE calculation – *cf Section 1 – Appendix B: 2.1 Excel code*

Opens data file and outputs the following on the spreadsheet (*cf Section 2 – Appendix B: 2.1 Excel code*):
- DDB Data set no
- No of data points
- Reference No
- Type of data (constant temperature or pressure)
- Retrieve component codes, empirical formula and component names
- Retrieve pure component molecular constants (r, q)
- Retrieve Antoine constants
- Retrieve experimental data

- Select liquid model - *cf Section 3 – Appendix B: 2.1 Excel code*

Adjust Vapour Pressure and calculate vapour pressure - *cf Section 4 – Appendix B -2.1 Excel code*

$$\ln P^{sat} = A + \frac{B}{T+C}$$

Calculate activity coefficients - *cf Section 5 – Appendix B -2.1 Excel code*

$$\gamma_i = \frac{y_i P}{x_i P_i^{sat}}$$

Calculate partial pressure - *cf Section 6 – Appendix B - 2.1 Excel code*

$$P_i = y_i P$$

Cont...

Write calculation formulas for P and y to spreadsheet - *cf Section 7 – Appendix B -2.1 Excel code*

$$P^{cal} = \sum_i x_i \gamma_i^{cal} P_i^{sat}$$

$$y_i^{cal} = \frac{x_i \gamma_i^{cal} P_i^{sat}}{P^{cal}}$$

Deviation:

$$Dev = \left(\frac{P^{exp} - P^{cal}}{P^{exp}} \right)^2$$

$$Re\,sidual = \sum \left[\left(\frac{P^{exp} - P^{cal}}{P^{exp}} \right)^2 \right]$$

Initialise regression parameters - *cf Section 6 – Appendix B – 2.1 Excel code*
$a_{12} = 50$ cal/mol
$a_{21} = 60$ cal/mol

Activate UNIQUAC model

Call Regression Procedure - *cf Section 9– Appendix B - 2.1 Excel code*

Calculate infinite dilution activity coefficients and Relative Absolute Deviation and output to worksheet - *cf Section 10 – Appendix B – 2.1 Excel code*

Activate FlexQUAC-Q model and initialise model parameters - *cf Section 11 – Appendix B – 2.1 Excel code*

Call Regression Procedure - *cf Section 12– Appendix B – 2.1 Excel code*

Calculate infinite dilution activity coefficients and Relative Absolute Deviation and output to worksheet - *cf Section 13 – Appendix B – 2.1 Excel code*

Figure 5-7 Algorithm detailing the implementation of the FlexQUAC-Q model in excel code

Chapter 6 Results & Discussion

6.1 Consistency Check

Due to the complexity of the mathematics *(Chapter 5)* involved in the derivation of the FlexQUAC-Q model and its restructuring and implementation in the programming code; one has to verify the correctness of the final implementation. It was vital to establish that the mathematics and the output of the programmed subroutine are consistent. As outlined previously in *Chapter 5* the model was first simplified in Mathcad and thereafter implemented in a Fortran subroutine and integrated into the existing DDB regression tool, RECVAL.

For the purpose of checking, data for the binary VLE system diethyl ether and acetonitrile were used (Joukovsky 1934). The data set was regressed using RECVAL program which contained the newly implemented Fortran subroutine to calculate activity coefficients for the FlexQUAC-Q model. This yielded optimal regression parameters. In addition, output statements were included at strategic points of the subroutine so that the numerical values of the pertinent variables can be

tabulated. The purpose of generating output to a text file is two fold; firstly, since it allows easy tracing through the subroutine code by monitoring the value of critical variables and secondly, it simplifies the process of identifying any errors substantially and the troubleshooting process. The Fortran subroutine generates a text file and the file records the values of the different values of the pertinent computational variables as the subroutine advances through the process of regression.

A Mathcad file was designed to generate numerical output for any given set of optimal parameters and mole fraction. Coupled with the Fortran output file, the process of identifying any source of errors became more manageable.

The pertinent variables and their respective values in both the Mathcad file and the Fortran file are summarised in **Table 6-1**. The output from both files is in excellent agreement. The rest of the activity coefficients are computed for the rest of the experimental data contained in the data file. This data is summarised in the mathcad file (*C1 - Consistency Check of Appendix C*) and the output from the Fortran file in a text file (*Flexq - cf Appendix C*). This data further corroborates the accuracy of the Fortran subroutine.

Variable	Mathcad	Fortran
x	$x_1 = 0; x_2 = 1$	
T	293 K	
Regression parameters	$\Delta u := \begin{pmatrix} 0 & 497.605 \\ -76.972 & 0 \end{pmatrix}$	$\delta := \begin{pmatrix} 0 & 0.1786 \\ 0.1786 & 0 \end{pmatrix}$
$\ln\gamma_{combinatorial}$	$\begin{bmatrix} -0.209 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.208 \\ 0 \end{bmatrix}$
τ	$\begin{bmatrix} 1.00 & 1.14 \\ 0.426 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 1.14 \\ 0.426 & 1.00 \end{bmatrix}$
sthetfx	1.724	1.723

Cont...

Variable	Mathcad	Fortran
Thetfx	$\begin{bmatrix} 2.288 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 2.288 \\ 1 \end{bmatrix}$
thetafx	$\begin{bmatrix} 3.945 \\ 1.723 \end{bmatrix}$	$\begin{bmatrix} 3.944 \\ 1.723 \end{bmatrix}$
thethaf	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
Sf	$\begin{bmatrix} 1.14 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1.14 \\ 1 \end{bmatrix}$
f1/ - sumk	-1.724	-1.723
f2	0	0
$G^E_{residual}$	0	0
$\dfrac{df_1}{dn}$	$\begin{bmatrix} -1.292 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1.292 \\ 0 \end{bmatrix}$
$\dfrac{df_2}{d\Theta}$	$\begin{bmatrix} 0.588 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.588 \\ 1 \end{bmatrix}$
SJ	$\begin{bmatrix} 2.653 \\ 1.724 \end{bmatrix}$	$\begin{bmatrix} 2.652 \\ 1.723 \end{bmatrix}$
Jac	$\begin{bmatrix} 2.288 & 0 \\ -2.288 & 0 \end{bmatrix}$	$\begin{bmatrix} 2.288 & 0 \\ -2.288 & 0 \end{bmatrix}$
$\dfrac{df_2}{dn}$	$\begin{bmatrix} -1.01 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1.01 \\ 0 \end{bmatrix}$
$\ln\gamma_{residual}$	$\begin{bmatrix} 1.743 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1.742 \\ 0 \end{bmatrix}$
γ	$\begin{bmatrix} 4.638 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 4.638 \\ 1 \end{bmatrix}$

Table 6-1 Comparison of accuracy of output data values in both Mathcad and Fortran for the system Diethyl-ether(1) and Acetonitrile(2) with $x_1 = 0$ and $x_2 = 1$ to assess accuracy of FlexQUAC-Q activity coefficient calculation

6.2 Performance Evaluation of FlexQUAC-Q Model

The consistency check served to confirm the accuracy of the Fortran subroutine and facilitated the subsequent performance evaluation of the FlexQUAC-Q model.

The performance analysis of the FlexQUAC-Q model followed a similar procedure to the one employed by Rarey (2005).

- Does the model give improved correlation of experimental binary VLE-data?
- Is the flexibilised model able to predict multicomponent VLE from binary data with a quality similar to (or even better than) the original model?
- Can the model be used for the simultaneous regression of VLE- and LLE-data?
- Is the model able to predict the ternary LLE-behavior from binary data alone?

For convenience the test of correlative and predictive ability with respect to vapour-liquid equilibria will be conducted using the same set of experimental data as in the previous work.

In this way, one can logically determine the effectiveness of the model compared to UNIQUAC and its precursor, the FlexQUAC model. In addition, the analysis would enable one to determine where the model shows marked improvement and verify if the model proves superior in the case of asymmetric systems. As a first step also, the FlexQUAC-Q model needs to demonstrate that is at least comparable with the former models and that it performance is not degenerative.

6.2.1 Binary VLE Data

6.2.1.1 Excel Regression

As a first test the FlexQUAC-Q-model was applied to the correlation of binary VLE data stored in the Dortmund Data Bank (DDB) using only one additional parameter. Different techniques are employed for the measurement of VLE data. Very high precision xPT-data are available using the static method as described by van Ness et. al. In this method the pressure in an equilibrium cell with known amounts of the components is measured at fixed temperature. The concentrations in the liquid and vapour phase are iteratively calculated using an nVT-flash calculation. A comparative test of the UNIQUAC, FlexQUAC and FlexQUAC-Q model was performed on this type of data whereby only data sets with 10 or more data points were used. The use of static measurements and much more precise pressure values (compared to the determination of the vapour composition) was always strongly advocated by Van Ness.

The resulting 4741 data sets were regressed individually using the Excel program with all three models with the mean relative squared deviation in pressure as objective function F:

$$F = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{P_{exp,i} - P_{calc,i}}{P_{exp,i}}\right)^2$$

6-1

The primary focus of the analysis was to establish the extent of the relative reduction of the objective function of the FlexQUAC-Q model compared to both FlexQUAC and UNIQUAC.

The objective functions (F) of the 4741 regressed binary VLE data sets (*cf Appendix C – C2 - Binary VLE results*) were arranged in ascending order for each of the three models. Each objective function corresponds to a percentage of the total data set and in so doing a cumulative percentage was created. Then, the deviation in pressure was plotted versus the percentage of data (**Figure 6-1**) to better understand the performance of the three models.

The minor mean relative reduction of about 3% of the objective function using FlexQUAC-Q compared to FlexQUAC was observed compared to a reduction by about 53% relative to the UNIQUAC-results. While in case of UNIQUAC approx. 48% of the data sets showed a final objective function of 10^{-3}, in case of the Flex-models this was achieved by about 63% of the data FlexQUAC-Q with the Weidlich and Gmehling combinatorial term behaves very similar to the FlexQUAC-Q model with the original combinatorial term.

This proves that both Flex-models perform nearly identical compared to UNIQUAC. It should be noted, that the improvement expected in case of asymmetric systems for the new transformation will not be strongly visible in case of vapour-liquid equilibria as the data sets are usually for components of similar vapour pressure and molecular size. Components very different in size usually have significantly different vapour pressures and present no problem in distillation. In case of liquid-liquid equilibria the vapour pressure has no influence and the phase equilibrium is determined only by the real behavior of the liquid.
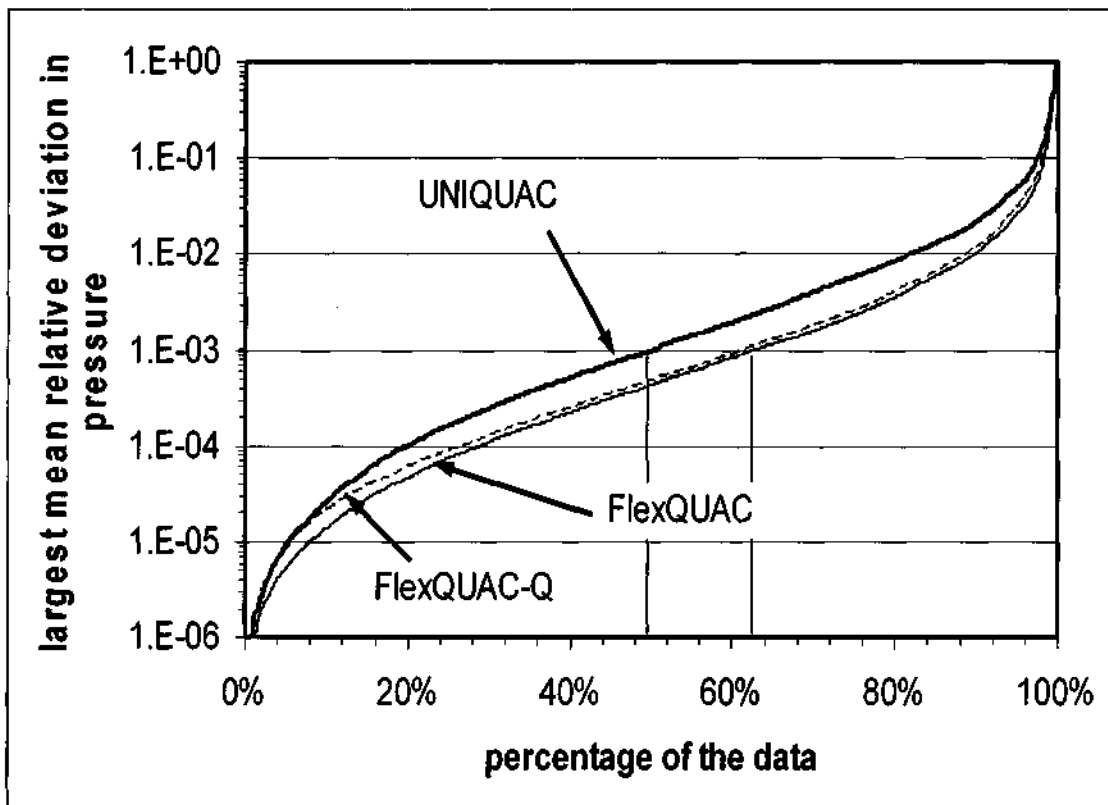
Figure 6-1 Largest deviation observed in a certain percentage of the sorted data set regressions

6.2.1.2 RECVAL Regression

Since RECVAL performs simultaneous regression of various types of data, it provided another way to test the performance of FlexQUAC-Q. Although the Flex-models were evaluated with a battery of binary VLE data, it would also be necessary to assess whether the model is able to correlate other data type simultaneously.

The functionality of RECVAL allows one to select a binary system from the DDB. Thereafter all the available experimental data sets of different types (activity coefficients, azeotropic data, excess heat capacity, excess enthalpy, VLE, LLE, SLE etc) can be exported to RECVAL for regression. In RECVAL, the regression can be adjusted with the use of weighting factors to assign more or exclusive importance to a specific type of data. The weighting factors should be set such that each data type gives similar total deviation.

However, not all experimental data are reliable and contradictory or false data result in erroneous predictions (*cf Chapter 2 – 2.10 Data quality and model selection*). RECVAL tracks the impact of each data point on the objective function by plotting residuals (**Figure 6-2**). Here one is able to ascertain which data set or data points have the largest deviation and these can be removed from the regression.

For the system tetrahydrofuran-water (**Figure 6-3**), UNIQUAC's fit of the data is not precise and the azeotropic point is not described well. FlexQUAC-Q performs as well as FlexQUAC. It fits the data and describes the azeotropic point precisely. Although the difference between the two curves is small, the qualitative description of the system is very different.
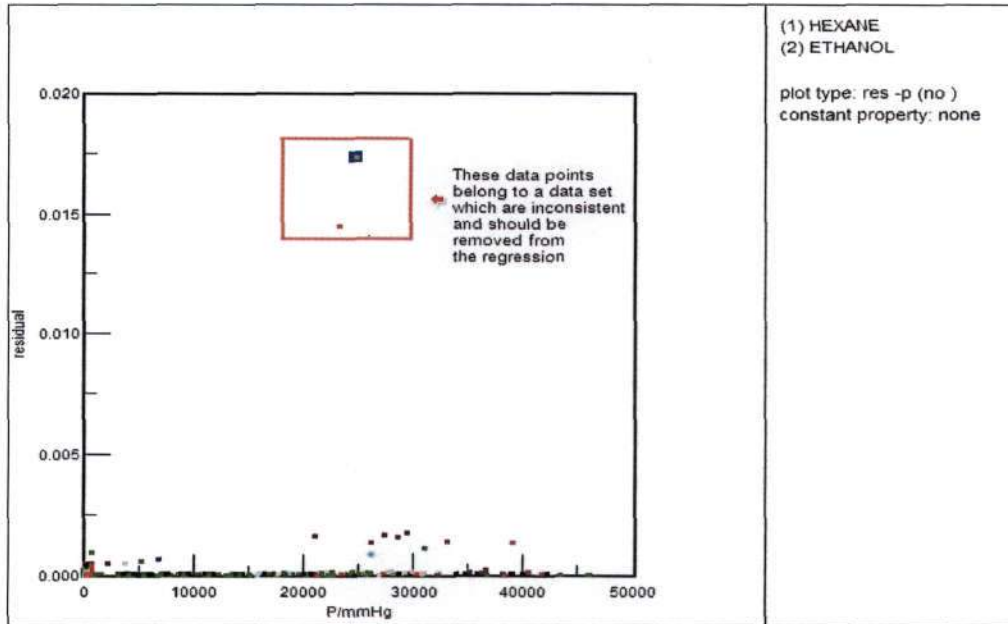
Figure 6-2 Residual plot for pressure in RECVAL for the system hexane-ethanol
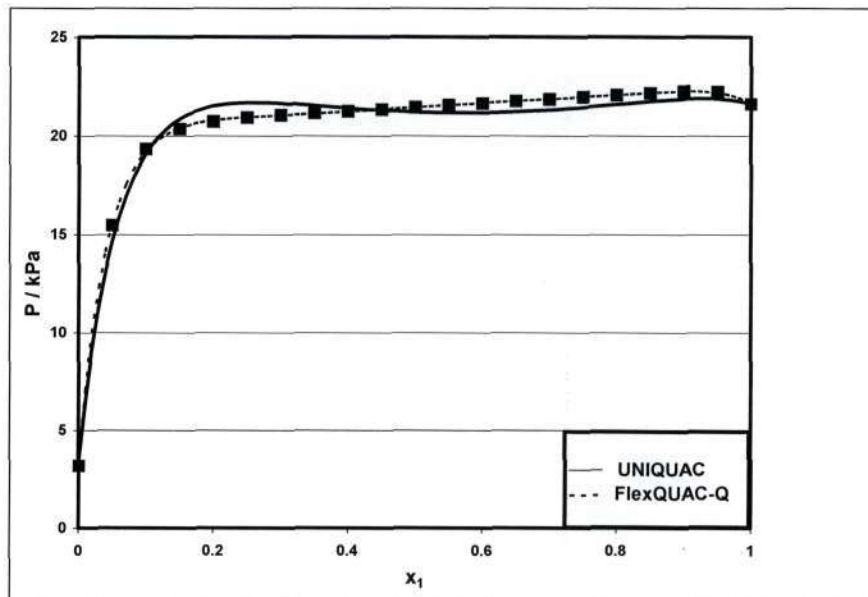
Figure 6-3 Description of tetrahydrofuran-water system at 298K (Signer et al , 1969) by the UNIQUAC and FlexQUAC-Q models

Consider the system hexane-acetone. The UNIQUAC surface parameters (q), are 3.856 and 2.3360 for hexane and acetone respectively. This represents a moderately asymmetric system. A simultaneous regression of available data for this system was done in RECVAL. **Table 6-2** shows the various data types used in the regression. Some data sets were omitted from the regression using the filtering process outlined above.

Data type	No. of data sets
VLE	19
LLE	11
Cp^E	1
H^E	10
Activity coefficient	48
Azeotropic data	69

Table 6-2 Different types of data sets that were simultaneously regressed for the system hexane-acetone

While UNIQUAC fails to give an accurate description in this case, both FlexQUAC and FlexQUAC-Q behave similarly and give an excellent description of the data. GEQUAC also provides an accurate description of the binary data (**Figure 6-5**). This model is an exact quasi-chemical multisegment model. Whilst GEQUAC provides such accuracy it must be noted that this model is complex and requires 14 parameters to obtain such an accurate description. This is in contrast to the FlexQUAC-Q which uses a non-linear transformation of surface fraction with only one additional parameter. FlexQUAC-Q is considerably simpler and produces the same accuracy.
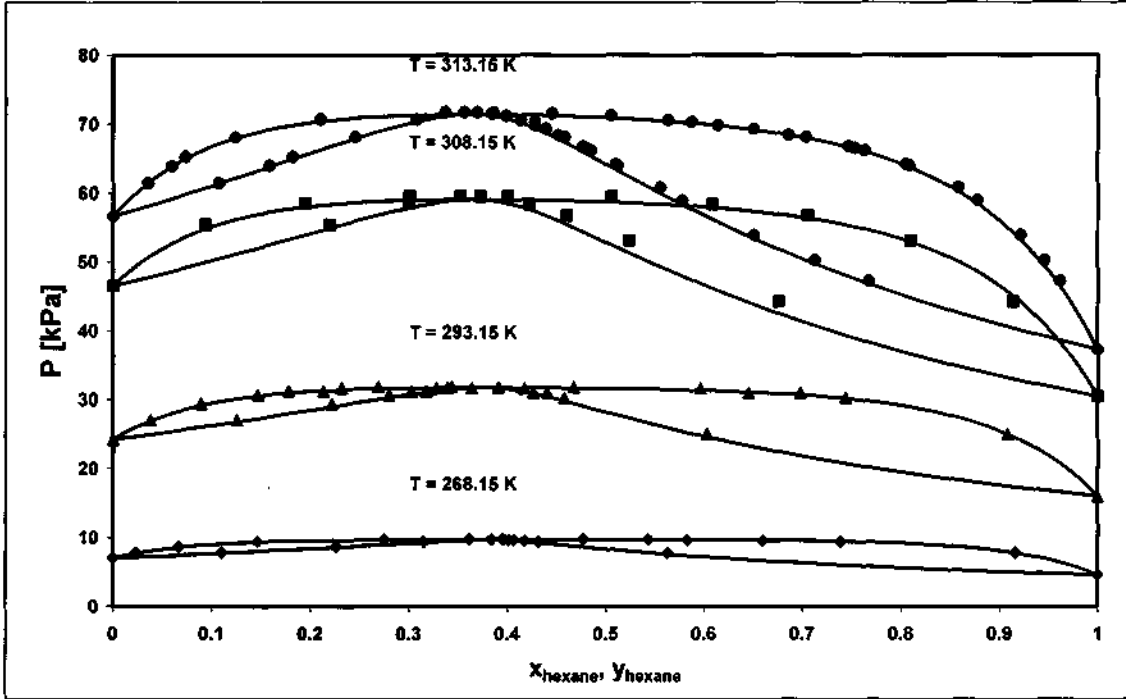
Figure 6-4 Hexane-acetone experimental data [T= 268.15K, 293.15K – (Rall et. al 1959), T = 308.15K – (Kudryavtseva et. al 1963), T = 313.15K – (Kolasinska et al. 1982)] fitted by the FlexQUAC-Q model
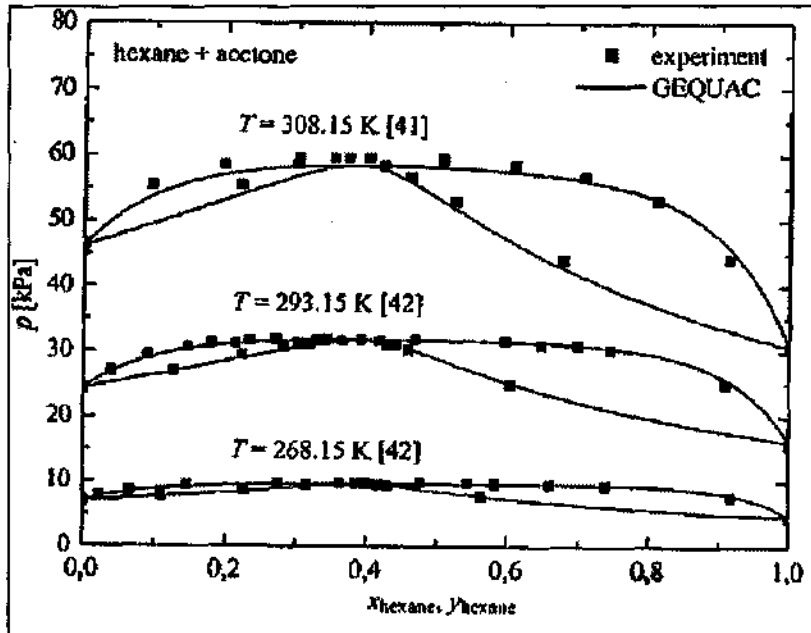
Figure 6-5 Hexane-acetone system as fitted by the GEQUAC model (Ehlker & Pfennig, 2002)

For the system ethanol-heptane, the UNIQUAC surface parameters are 1.972 (q_1) and 4.396 (q_2) respectively. This indicates that the system is asymmetric ($q_2/q_1 = 2.23$). This system was regressed simultaneously with the data sets detailed in Table 6-3.

Data type	No. of data sets
VLE	41
Cp^E	17
H^E	29
Activity coefficient	73
Azeotropic data	72

Table 6-3 Different types of data sets that were simultaneously regressed for the system ethanol-heptane

Figure 6-6 shows the correlation of H^E data by the FlexQUAC-Q model. Since the experimental data contain Cp^E data at various temperatures, it was appropriate to use quadratic temperature dependence for the interaction parameters in RECVAL. UNIQUAC failed to give a good description of H^E data. Once again the description of the data using the FlexQUAC-Q model is comparable to the accuracy of the description achieved by the GEQUAC (Figure 6-7) group contribution methods.
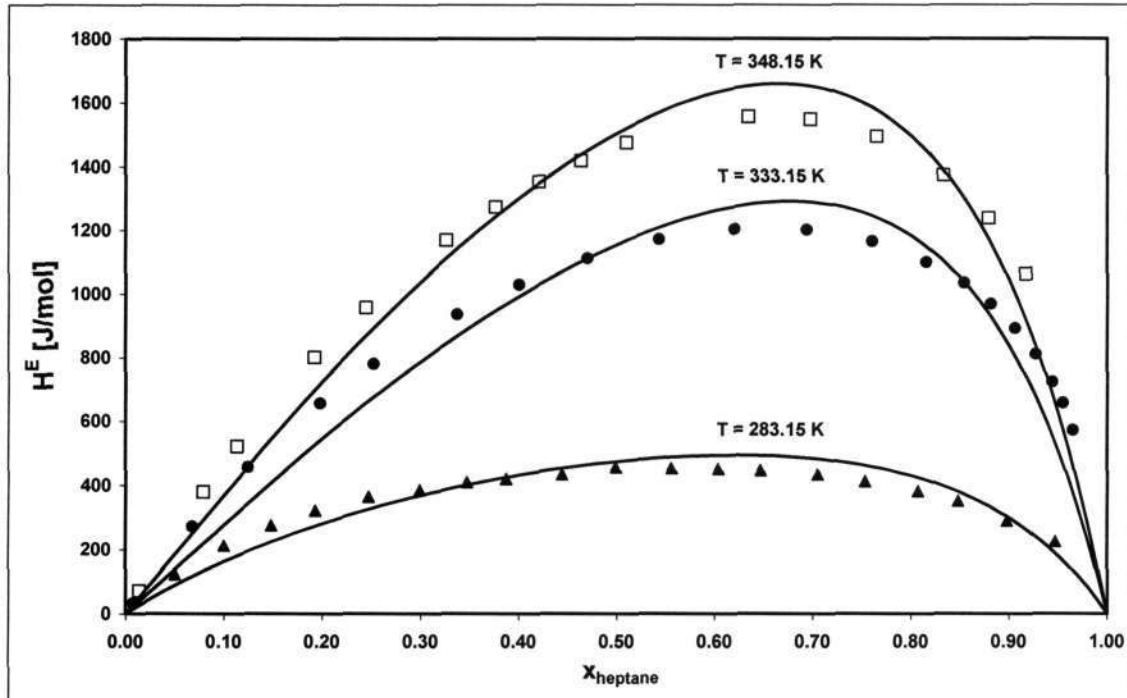
Figure 6-6 Heptane-ethanol H^E data (T= 283.15K, 323.15K – Lietzmann et.al 1994, T = 333.15K, 348.15K – Van Ness et.al 1976) as fitted by the FlexQUAC-Q model
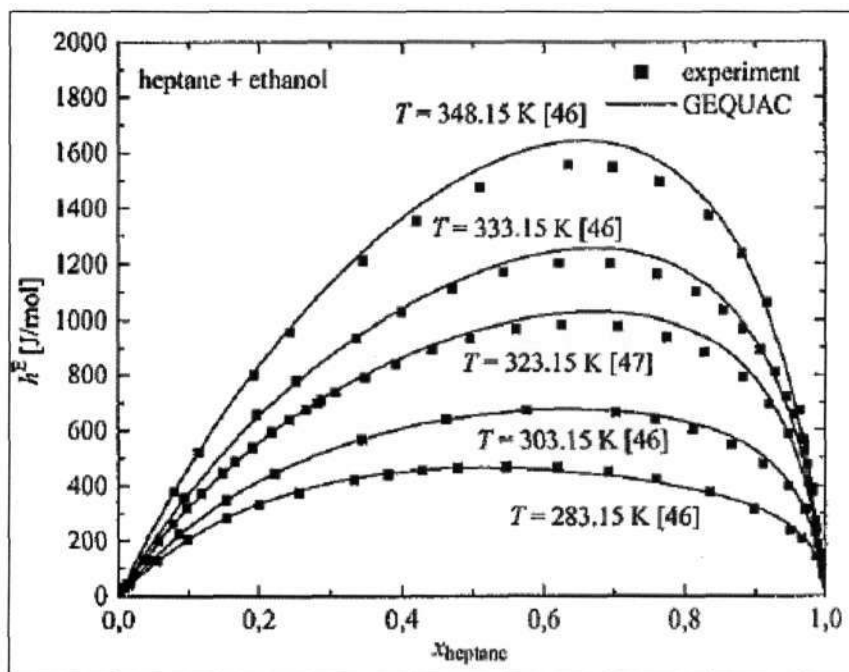
Figure 6-7 Heptane-ethanol H^E data as fitted by the GEQUAC model (Ehlker & Pfennig, 2002

Figure 6-8 shows the effect of flexibilisation with the FlexQUAC model for an arbitrary mixture with a surface area ratio of 6 to 1 as function of mole fraction and surface fraction. In the plot vs. mole fraction, most of the changes in the activity coefficients occur in the left side of the diagram. The modifications by flexibilisation on the other hand are similar on both sides. The effect of flexibilisation at low concentrations of component 1 is most pronounced while at high concentrations it is confined to a small range. FlexQUAC-Q should be able improve the quality of the data prediction based on this more adequate concentration scale.
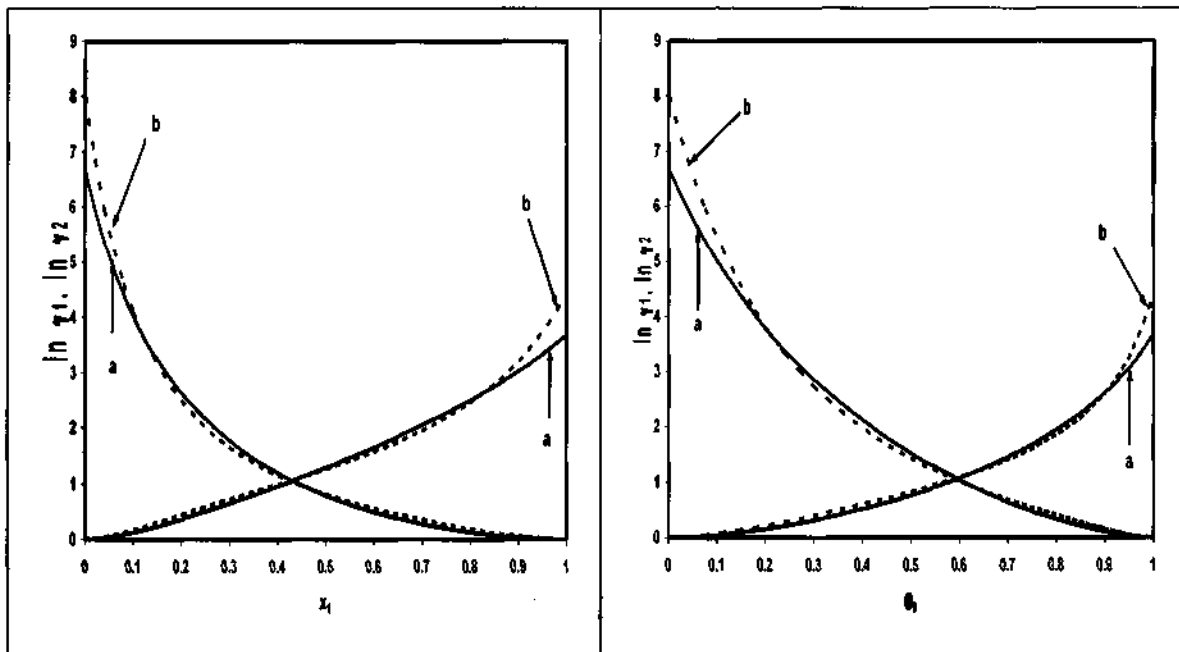
Figure 6-8 Effect of flexibilisation ((a) $\delta_{12}=0$, (b) $\delta_{12}=0.2$)) on an arbitrary mixture with a surface are ratio of 6 to 1.

Figure 6-9 shows the left and right homogeneous region of the moderately asymmetric mixture 2-butanol-water. The surface area ratio for this mixture is 2.178 ($q_{2\text{-butanol}}$ is 3.048 and q_{water} is 1.4). While on the right hand side of the miscibility gap a FlexQUAC parameter of 0.22 is sufficient, the pressure on the left hand side still looks underpredicted.

Figure 6-10 shows the regression of the system 2-butanol-water for various values of the interaction parameter, δ_{12}. A value of 0.20 for the parameter δ_{12} corresponds to the best description of the data for the left homogeneous region.
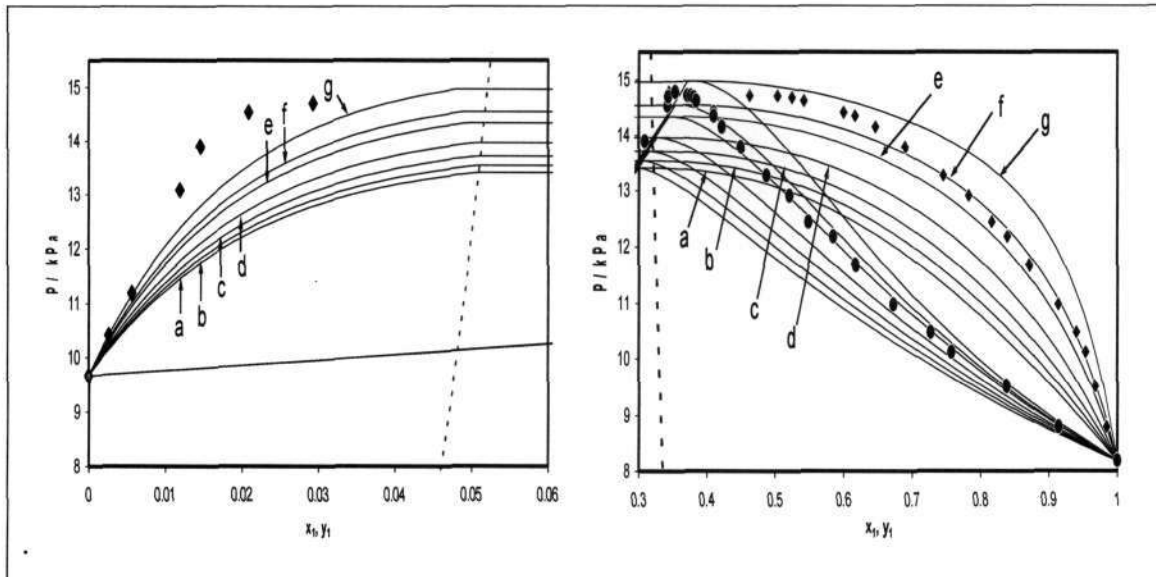
Figure 6-9 Left and right region of the Pxy-diagram for the system 2-Butanol (1) – Water (2) at 45.04°C together with experimental xyP-data (Escobedo-Alvarado G.N., Sandler S.I., 1999) and curves calculated from FlexQUAC using different values of δ_{12} (a: 0.0, b: 0.05, c: 0.1, d: 0.15, e: 0.2, f: 0.22, g: 0.25). Interaction parameters were regressed to liquid-liquid equilibrium (Marongiu B., Ferino I., et al, 1984.) composition (dashed lines)
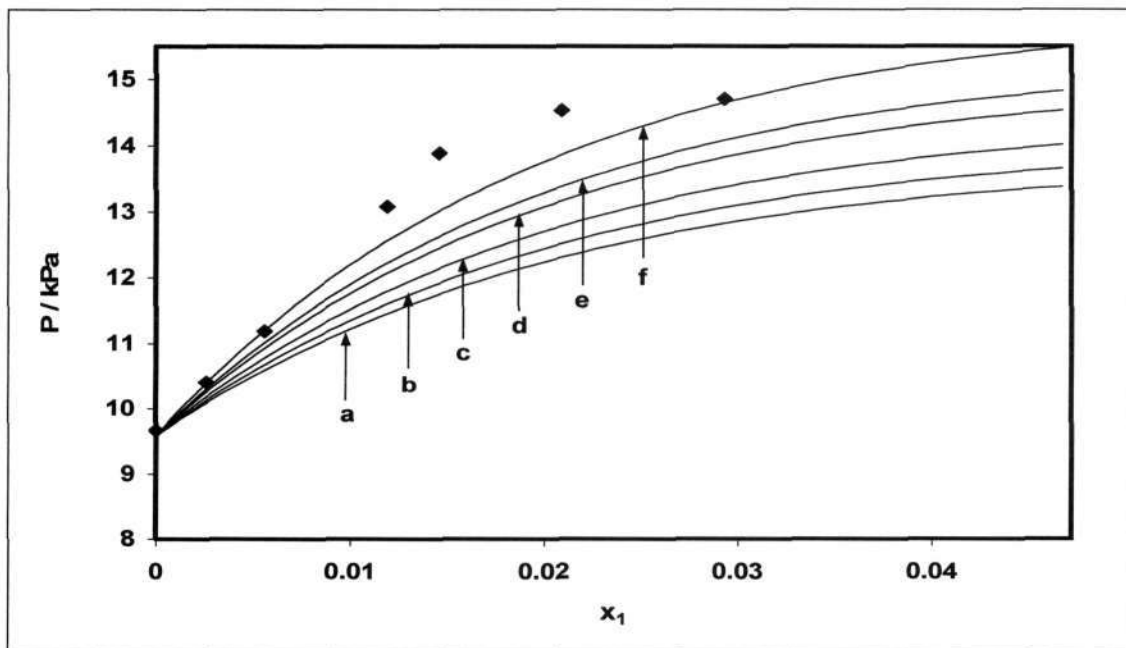
Figure 6-10 Left region of Pxy-diagram for the system 2-Butanol (1) – Water (2) at 45.04°C together with experimental xyP-data (Escobedo-Alvarado G.N., Sandler S.I., 1999) and curves calculated from FlexQUAC-Q using different values of δ_{12} (a: 0.0, b: 0.05, c: 0.1, d: 0.15, e: 0.17, f: 0.20). Interaction parameters were regressed to liquid-liquid equilibrium (Marongiu B., Ferino I., et al, 1984.) composition (dashed lines)

In accordance with the definition of both the FlexQUAC and FlexQUAC-Q models, a value of δ_{12} = 0 the both models degenerate back to the original UNIQUAC model. From **Figure 6-11** it is evident that UNIQUAC fails to give an accurate description of the data. In the case of FlexQUAC, a good description of the data is obtained at a value of approximately 0.23 (Rarey, 2005). Although a value of 0.20 gives a good description for the left region, the right region is not accurately described.

However, FlexQUAC-Q additional parameter δ_{12}, is intended to have a more pronounced effect on the larger component in the mixture, 2-butanol is this case. **Figure 6-10** confirms that FlexQUAC-Q parameter does have the desired effect on the curvature in the left region. For a gradual increase in the δ_{12} value, the fit of the data becomes better progressively. In the case of FlexQUAC, the curvature of the left region does not rise adequately as higher values of δ_{12} are approached and hence the description is not entirely accurate.
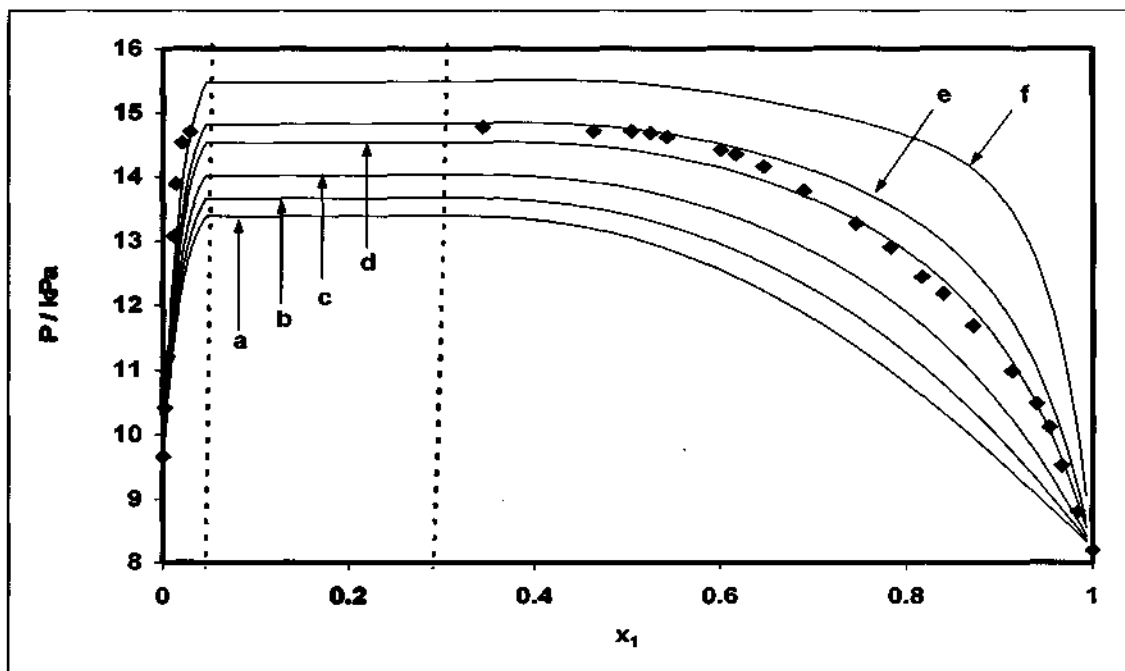
Figure 6-11 Pxy-diagram for the system 2-Butanol (1) – Water (2) at 45.04°C together with experimental xyP-data (Escobedo-Alvarado G.N., Sandler S.I., 1999) and curves calculated from FlexQUAC-Q using different values of δ_{12} (a: 0.0, b: 0.05, c: 0.1, d: 0.15, e: 0.17, f: 0.20). Interaction parameters were regressed to liquid-liquid equilibrium (Marongiu B., Ferino I., et al, 1984.) composition (dashed lines)

6.2.2 Ternary VLE Data

Ternary VLE data was then used to assess the performance of FlexQUAC-Q compared to UNIQUAC. The data of P. Giercyz (1964) was used to test the performance of the FlexQUAC model. Ternary homogeneous data sets were carefully selected and they were all measured in the same laboratory and the diversity of systems vary from strong positive to strong negative deviations from Raoult's law. The data set comprised 13 ternary VLE data sets and 39 binary VLE data sets. The binary data for the 39 data sets were regressed using the FlexQUAC-Q model (both original combinatorial and Weidlich & Gmehling's combinatorial term) and these parameters were then used to calculate the ternary VLE data (*cf Appendix C – C3 - Ternary VLE Results – Original Combinatorial and Ternary VLE results & Weidlich and Gmehling*).

DDB Ternary Data Set No.	Component 1	Component 2	Component 3	q_1	q_2	q_3	Maximum surface ratio		
							Comp 1 & 2	Comp 1 & 3	Comp 2 & 3
11564									
11565									
11566	Methanol	n-Hexane	Cyclohexane	1.432	3.856	3.24	2.693	2.263	1.190
11582									
11583									
6192	Acetone	Methanol	Chloroform	2.336	1.432	2.41	1.631	1.032	1.683
7057	Ethanol	Acetonitrile	Water	1.972	1.724	1.4	1.144	1.409	1.231
8209	Acetone	Ethanol	Water	2.336	1.972	1.4	1.185	1.669	1.409
7053	Ethanol	Water	1,4 Dioxane	1.972	1.4	2.64	1.409	1.339	1.886
5331	Acetone	Acetonitrile	Methanol	2.336	1.724	1.432	1.355	1.631	1.204
2654	Benzene	Cyclohexane	Aniline	2.4	3.24	2.816	1.350	1.173	1.151
2657	n-Hexane	Benzene	Cyclohexane	3.856	2.4	3.24	1.607	1.190	1.350
5971	1-Heptene	n-Heptane	n-Octane	4.184	4.396	4.936	1.051	1.180	1.123

Table 6-4 Ternary data set components and UNIQUAC surface fraction parameters (q) and calculated surface fraction ratio for the constituent binary systems

Table 6-4 lists the components for each of the thirteen ternary systems and the individual component UNIQUAC surface parameter, q. The maximum surface ratio is calculated as the ratio of the UNIQUAC surface parameter of the larger component to that of the smaller component. This gives an indication of the degree of asymmetry. From this calculation, one can infer that the systems are very moderately asymmetric, with a higher degree of asymmetry observed for the system ethanol- water- 1,4dioxane. Hence, it is expected that the FlexQUAC-Q model should not

produce any significant reduction in objective function as compared to the FlexQUAC model since the degree of asymmetry is not pronounced.

6.2.2.1 Verification of Regression of Constituent Binary VLE and Calculated Ternary VLE Data - Regression using Original Combinatorial Term

The ternary VLE system cyclohexane-methanol-hexane (DDB data set no. 11564) calculations in Excel were verified with the calculations from Mathcad using the original combinatorial term.

Binary VLE Data – Original Combinatorial Term										
Ternary Data Set	Binary Set No.	T^0 /K	No. of Data Pts	Component 1	Component 2	FlexQUAC-Q				
						a_{12}	a_{21}	δ_{12}	γ_1^∞	γ_2^∞
	11573	293.15	46	Methanol	Hexane	44.682	1212.436	0.078	37.879	27.853
	11570	293.15	25	Methanol	Cyclohexane	82.003	982.553	0.195	55.911	36.912
11564	11567	293.15	10	Hexane	Cyclohexane	-105.905	133.687	0.073	1.199	1.138

Table 6-5 Ternary VLE system cyclohexane-methanol-hexane with the constituent binary VLE data systems (Goral et. al. 2000)

The parameters summarized in Table 6-5 were used in the Mathcad calculations to verify the calculations of the activity coefficients at experimental liquid composition and the activity coefficients at infinite dilution.

Refer to *Appendix C - C3 - Ternary VLE Results and Calculations 11564 Ternary Check* (Printed file - The Mathcad file is available on CD in the *Appendix C* folder). Due to the large number of data sets for the binary systems methanol-hexane and methanol-cyclohexane (Data sets 11573 and 11570 respectively) only ten liquid molar compositions were selected for the calculation in Mathcad. The full experimental data set and calculated results are available on CD in the folder, *Appendix C - Ternary VLE Results and Calculation - Original Combinatorial*. The results from the Excel regression and the Mathcad files are in excellent agreement (cf Table 6-6 - Table 6-11).

x_1	γ_1	γ_2
0	37.879	1
0.0002	37.867	1
0.00076	37.422	1
0.1483	7.406	1.124
0.859	1.097	6.495
0.9502	1.017	13.483
0.9773	1.004	19.522
0.9915	1.001	24.178
0.9993	1	27.519
1	1	27.853

Table 6-6 Mathcad results for the calculation of the activity coefficient for system Methanol-hexane (Goral et. al. 2000)

Experimental Data		Calculated Data		
x_1	P	γ_1^{calc}	γ_2^{calc}	P_{calc}
0.00000	121.21	37.879	1	121.21
0.00002	121.36	37.866	1	121.28
0.00076	124.44	37.421	1.0000	123.89
0.14830	209.79	7.4062	1.1238	223.11
0.85920	209.04	1.0972	6.4954	202.78
0.95020	180.54	1.0168	13.482	177.78
0.97730	148.21	1.0039	19.521	149.39
0.99150	119.71	1.0005	24.178	121.65
0.99930	99.08	1.0000	27.518	99.78
1.00000	97.51	1	27.852	97.51

Table 6-7 Excel regression and calculation results for system Methanol-hexane (Goral et. al. 2000)

x_1	γ_1	γ_2
0	55.911	1
0.00058	55.171	1
0.00277	52.49	1
0.01225	42.7	1.002
0.05726	19.022	1.03
0.0901	12.11	1.068
0.8257	1.162	4.508
0.9002	1.073	7.551
0.939	1.033	11.679
1	1	36.912

Table 6-8 Mathcad results for the calculation of the activity coefficients for the system Methanol-cyclohexane (Oracz et. al. 1996)

Experimental Data		Calculated Data		
	P			P_{calc}
x_1	kPa	γ_1^{calc}	γ_2^{calc}	kPa
0.00000	77.53	55.911	1	77.53
0.00058	82.63	55.170	1.0000	80.61
0.00277	105.48	52.489	1.0000	91.50
0.01225	141.66	42.699	1.0016	127.73
0.05726	162.98	19.022	1.0303	181.55
0.09010	166.50	12.110	1.0677	181.75
0.82570	167.24	1.1616	4.5083	154.48
0.90020	161.85	1.0726	7.5507	152.61
0.93900	151.47	1.0330	11.679	149.86
1.00000	97.54	1	36.912	97.54

Table 6-9 Excel regression and calculation results for system Methanol-cyclohexane (Oracz et. al. 1996)

x_1	γ_1	γ_2
0	1.199	1
0.1004	1.124	1.003
0.1899	1.086	1.009
0.2941	1.059	1.017
0.3968	1	1.026
0.4928	1.029	1.036
0.5946	1.019	1.048
0.6901	1.012	1.061
0.7969	1.006	1.08
1	1	1.138

Table 6-10 Mathcad results for the calculation of the activity coefficients for the system Hexane-cyclohexane (Goral et. al. 2000)

Experimental Data		Calculated Data		
x_1	P	γ_1^{calc}	γ_2^{calc}	P_{calc}
0.00000	77.53	1.1987	1.0000	77.53
0.10040	83.66	1.1242	1.0031	83.64
0.18990	88.32	1.0863	1.0088	88.37
0.29410	93.45	1.0586	1.0170	93.40
0.39680	98.04	1.0406	1.0262	98.04
0.49280	102.14	1.0285	1.0359	102.17
0.59460	106.34	1.0187	1.0477	106.35
0.69010	110.14	1.0116	1.0610	110.11
0.79690	114.11	1.0055	1.0800	114.12
1.00000	121.20	1.0000	1.1381	121.20

Table 6-11 Excel regression and calculation results for system Hexane-cyclohexane (Goral et. al. 2000)

6.2.2.2 Ternary VLE Calculations from Binary Interaction Parameters

Table 6-12 and Figure 6-12 contains a summary of the mean relative deviation in pressure for the constituent binary VLE systems and the corresponding calculation of ternary VLE data accompanied with γ^∞ for the binary systems.

As in the case of FlexQUAC, FlexQUAC-Q performs better than UNIQUAC for systems with medium to large deviations from Raoult's law. This improved performance is seen for the ternary system Methanol (1) – n-Hexane(2) – Cyclohexane (3). Naturally, no improvements were observed for systems with small deviations from Raoult's law.

The objective for evaluating FlexQUAC-Q in calculating ternary VLE data from binary VLE data was purely to verify that the model retains its predictive capability. The results obtained illustrate that model modification does achieve this.
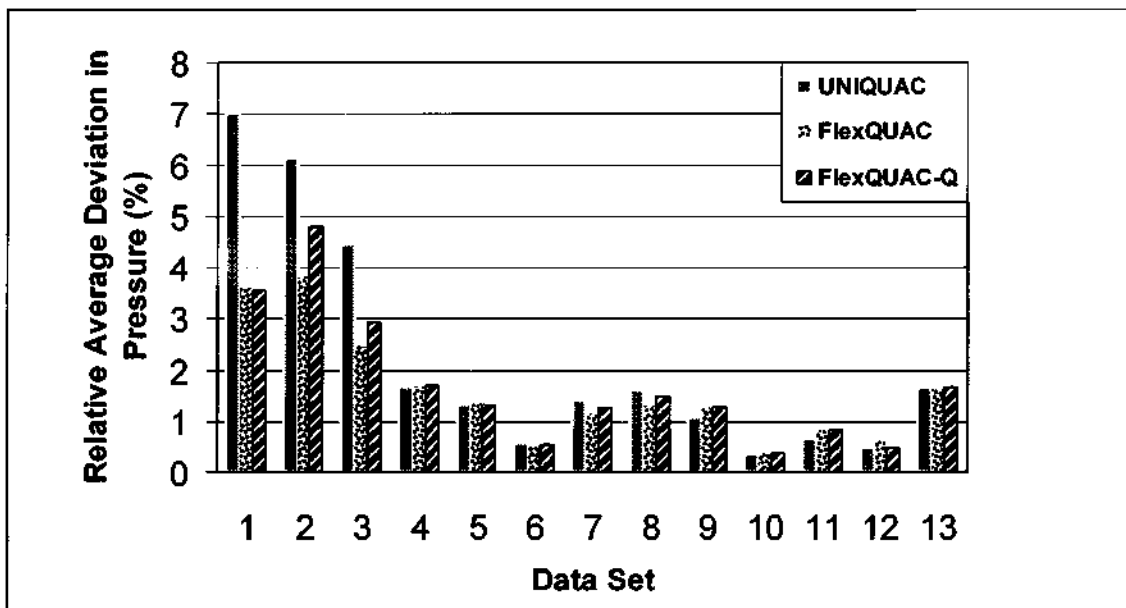
Figure 6-12 Comparison of the UNIQUAC, FlexQUAC and FlexQUAC-Q models for selected ternary mixtures

System	T/ K	RAD in P / %			γ^{∞} (FlexQUAC-Q)	
		UNIQUAC	FlexQUAC	FlexQUAC-Q	(1) in (2)	(2) in (1)
Methanol (1) - n-Hexane (2)	293.15	3.25	1.52	2.37	37.88	27.85
Methanol (1) - Cyclohexane (2)	293.15	7.49	4.68	6.59	55.91	36.91
n-Hexane (1) - Cyclohexane (2)	293.15	0.06	0.02	0.02	1.20	1.14
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	**293.15**	**7.01**	**3.61**	**3.54**		
Methanol (1) - Cyclohexane (2)	303.15	7.01	3.14	4.45	58.01	29.16
Methanol (1) - n-Hexane (2)	303.15	4.36	1.91	2.89	37.64	26.50
n-Hexane (1) - Cyclohexane (2)	303.15	0.07	0.06	0.06	1.17	1.10
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	**303.15**	**6.12**	**3.82**	**4.78**		
Methanol (1) - Cyclohexane (2)	313.15	4.17	2.62	4.17	58.08	7.01
Methanol (1) - n-Hexane (2)	313.15	3.30	0.77	1.43	36.13	25.70
n-Hexane (1) - Cyclohexane (2)	313.15	0.02	0.03	0.03	1.12	1.10
Methanol (1) - n-Hexane (2) - Cyclohexane (3)	**313.15**	**4.46**	**2.47**	**2.93**		
Acetone (1) - Chloroform (2)	313.15	0.34	0.35	0.37	0.36	0.46
Acetone (1) - Methanol (2)	313.15	0.06	0.02	0.02	2.01	1.98
Chloroform (1) - Methanol (2)	313.15	0.43	0.43	0.43	2.53	9.21
Acetone (1) - Methanol (2) - Chloroform (3)	**313.15**	**1.68**	**1.71**	**1.70**		
Acetone (1) - Chloroform (2)	323.15	0.34	0.35	0.37	0.36	0.46
Acetone (1) - Methanol (2)	323.15	0.06	0.02	0.02	2.01	1.98
Chloroform (1) - Methanol (2)	323.15	0.33	0.33	0.33	2.62	8.09
Acetone (1) - Methanol (2) - Chloroform (3)	**323.15**	**1.33**	**1.36**	**1.34**		
Acetone (1) - Chloroform (2)	323.15	0.36	0.36	0.36	0.33	0.53
Acetone (1) - Methanol (2)	323.15	0.19	0.08	0.09	2.01	2.05
Chloroform (1) - Methanol (2)	323.15	0.33	0.33	0.33	2.58	8.56
Acetone (1) - Methanol (2) - Chloroform (3)	**323.15**	**0.56**	**0.51**	**0.53**		
Ethanol (1) - Acetonitrile (2)	323.15	0.26	0.19	0.19	3.21	3.68
Acetonitrile (1) - Water (2)	323.15	1.23	0.16	0.18	12.93	7.79
Ethanol (1) - Water (2)	323.15	0.39	0.39	0.39	5.57	2.63
Ethanol (1) - Acetonitrile (2) - Water (3)	**323.15**	**1.41**	**1.16**	**1.27**		
Acetone (1) - Ethanol (2)	323.15	0.13	0.02	0.02	2.15	2.00
Acetone (1) - Water (2)	323.15	0.33	0.04	0.07	9.44	5.67
Ethanol (1) - Water (2)	323.15	0.23	0.23	0.23	5.38	2.67
Acetone (1) - Ethanol (2) - Water (3)	**323.15**	**1.62**	**1.32**	**1.50**		
Ethanol (1) - 1,4-Dioxane (2)	323.15	0.19	0.12	0.12	2.48	2.93
Water (1) - 1,4-Dioxane (2)	323.15	1.64	1.64	1.64	9.34	10.84
Ethanol (1) - Water (2)	323.15	0.21	0.21	0.21	5.65	2.61
Ethanol (1) - Water (2) - 1,4-Dioxane (3)	**323.15**	**1.08**	**1.29**	**1.27**		
Acetone (1) - Acetonitrile (2)	323.15	0.17	0.17	0.17	1.01	1.00
Acetone (1) - Methylacetate (2)	323.15	0.03	0.03	0.03	1.14	1.12
Methylacetate (1) - Acetonitrile (2)	323.15	0.06	0.04	0.04	1.29	1.34
Acetone (1) - Methylacetate (2) - Acetonitrile (3)	**323.15**	**0.42**	**0.37**	**0.36**		

Cont…

System	T/ K	RAD in P / %			γ8 (FlexQUAC-Q)	
		UNIQUAC	FlexQUAC	FlexQUAC-Q	(1) in (2)	(2) in (1)
Benzene (1) - Cyclohexane (2)	343.15	0.10	0.07	0.07	1.46	1.53
Cyclohexane (1) - Aniline (2)	343.15	0.97	0.33	0.34	7.25	9.31
Benzene (1) - Aniline (2)	343.15	0.22	0.22	0.24	1.85	2.20
Benzene (1) - Cyclohexane (2) - Aniline (3)	**343.15**	**0.65**	**0.83**	**0.83**		
Benzene (1) - Cyclohexane (2)	343.15	0.26	0.26	0.26	1.32	1.47
n-Hexane (1) - Cyclohexane (2)	343.15	0.25	0.25	0.25	1.05	1.04
n-Hexane (1) - Benzene (2)	343.15	0.17	0.17	0.17	1.54	1.56
n-Hexane (1) - Benzene (2) - Cyclohexane (3)	**343.15**	**0.65**	**0.63**	**0.49**		
1-Heptene (1) - n-Heptane (2)	328.15	0.03	0.03	0.03	1.08	1.10
1-Heptene (1) - n-Octane (2)	328.15	0.38	0.11	0.12	1.22	1.64
n-Heptane (1) - n-Octane (2)	328.15	0.11	0.11	0.12	1.07	1.21
1-Heptene (1) - n-Heptane (2) - n-Octane (3)	**328.15**	**1.64**	**1.64**	**1.67**		

Table 6-12 Calculation of ternary VLE data from binary interaction parameters

6.2.3 Simultaneous Description of VLE & LLE

Chapter 3 reviewed the common deficiencies of Gibbs excess energy models. A common deficiency shared by all models is the inability to simultaneously describe VLE and LLE.

FlexQUAC was able to simultaneously describe VLE and LLE as demonstrated by Rarey (2005). Although the transformation in the FlexQUAC-Q model is applied to the surface fraction the ability to simultaneously describe VLE and LLE should be retained.

The same example analysed in the case of the FlexQUAC is presented here. **Figure 3-2** illustrates the prediction of VLE using parameters obtained from the regression of VLE and LLE data using the UNIQUAC model.

Figure 6-13 shows the simultaneous description of the VLE, LLE and azeotropic data using the FlexQUAC-Q model.
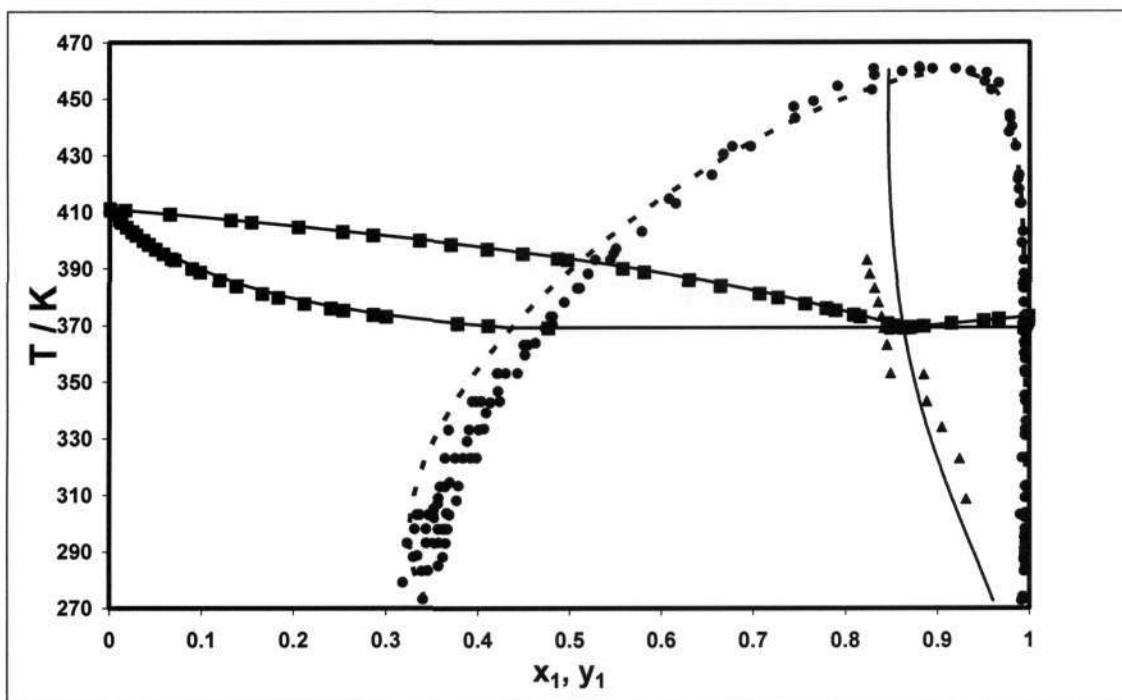
Figure 6-13 Simultaneous description of VLE and LLE using FlexQUAC-Q G^E model parameters from regression of LLE and VLE data for water (1)- 1-pentanol (2) [■ - VLE data (Cho et. al. 1984), ♦ - LLE data (DDB, 2006), ▲ – azeotropic data (DDB, 2006), --- VLE calculation, - - - LLE calculation, —— Azeotropic data]
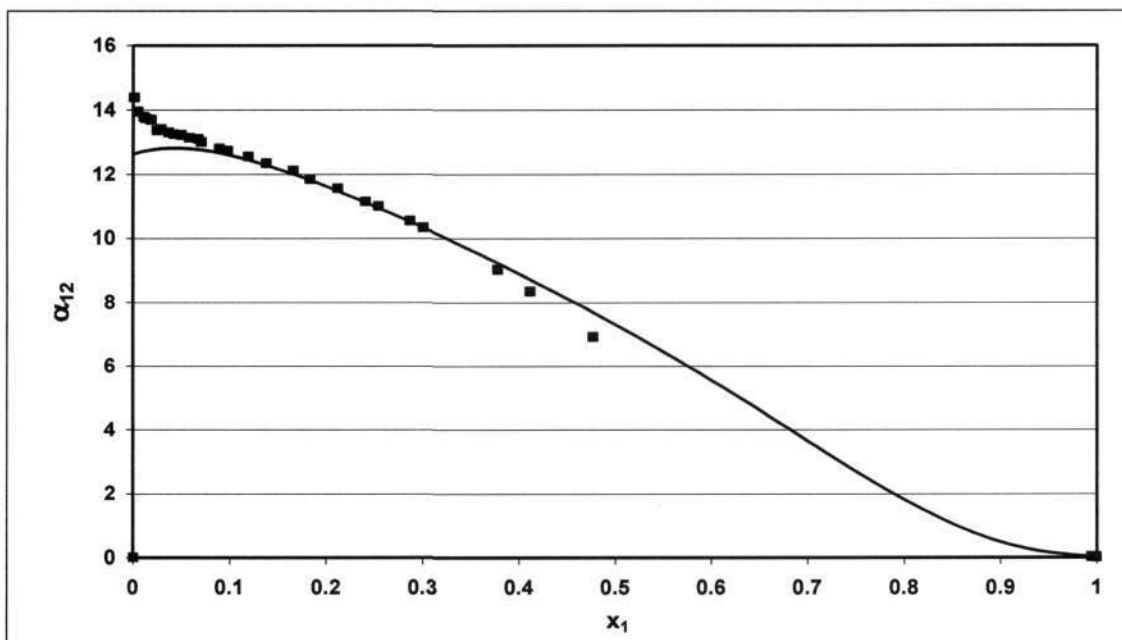
Figure 6-14 Separation factor as a function of liquid mole composition using FlexQUAC-Q G^E model parameters derived from the simultaneous regression of VLE and LLE data for the system Water (1) – 1-Pentanol (2) at 101.3kPa (Cho, Ochi & Kojima, 1984)

6.2.4 Ternary LLE Data

Ternary LLE calculations were performed using binary VLE and LLE data.

The ternary system Methanol (1)–Acetone (2) –Cyclohexane(3) (Campbell et. al. 1972) at 298.15K was investigated. Binary VLE data was regressed in the cases of the systems methanol-acetone and acetone-cyclohexane to yield binary interaction parameters. In the case of the system methanol-cyclohexane, LLE data was used solely to obtain the interaction parameters. As explained in *Chapter 4 – 4.1 Simultaneous Regression of Phase Equilibrium Data*, it is advisable that some LLE data be included in this type of regression to improve the accuracy in predicting ternary LLE.

No.	Component 1	Component 2	q_1	q_2	Maximum surface area ratio
1	Methanol	Acetone	1.432	2.336	1.63
2	Methanol	Cyclohexane	1.432	3.24	2.26
3	Acetone	Cyclohexane	2.336	3.24	1.39

Table 6-13 Maximum surface area ratio for each binary system

All systems exhibit moderate asymmetric behaviour and FlexQUAC-Q will not show any drastic improvements in the correlation of binary VLE data.

6.2.4.1 Regression of Binary Data in RECVAL

The binary data of each constituent system was regressed in RECVAL to obtain interaction parameters for the ternary LLE calculation.

Table 6-14 shows the various VLE and LLE data sets that were regressed for the respective binary constituent systems.

System	Data set		Temperature/ K
Methanol(1) – Acetone(2)	VLE Data	Puri et. al. 1974	298.15
		Tasic et. al. 1978	298.15
		Oracz et. al. 1996	293.15 / 303.15
		Campbell et. al. 1976	298.15
Methanol(1) – Cyclohexane(3)	LLE Data	Nagata et. al. 1983	298.15
Acetone(2) – Cyclohexane(3)	VLE Data	Rhim et. al. 1975	298.15
		Tamir et. al. 1981	298.15
		Campbell et. al. 1972	298.15
		Pierotti et. al. 1959	298.15

Table 6-14 The binary VLE and LLE data regressed for the purpose of obtaining binary interaction parameters

System	Interaction Parameters/ cal/mol	FlexQUAC-Q δ Parameter
Methanol(1) – Acetone(2)	$a_{12} =$ 164.0496	0
	$a_{21} =$ 79.7562	
Methanol(1) – Cyclohexane(3)	$a_{13} =$ 16.0264	0.004
	$a_{31} =$ 1287.072	
Acetone(2) – Cyclohexane(3)	$a_{23} =$ -14.6356	0.15
	$a_{32} =$ 499.1818	

Table 6-15 Binary interaction parameters obtained from regression for the three binary systems

6.2.4.2 Ternary LLE Calculation

The binary interaction parameters obtained from RECVAL were used in an Excel calculation program to perform the ternary LLE calculation.

The calculated activity coefficients in Excel were verified with the use of a MathCAD file – Refer to *Appendix C – Ternary LLE Calculations*. The two results are in excellent agreement (Table 6-16).

Excel Calculations						Mathcad Calculations		
Molar compositions calculated by K factor method			Calculated Activity Coefficients			Calculated Activity Coefficients		
x_1'	x_2'	x_3'	γ_1'	γ_2'	γ_3'	γ_1'	γ_2'	γ_3'
0.11996	0.00000	0.88004	7.49989	4.88610	1.05611	7.5	4.886	1.056
0.19836	0.03309	0.76856	4.25414	2.70253	1.18922	4.254	2.703	1.189
0.42971	0.09217	0.47813	1.87743	1.45976	1.87781	1.877	1.46	1.878

Table 6-16 Results from Excel calculation compared with Mathcad results to verify accuracy
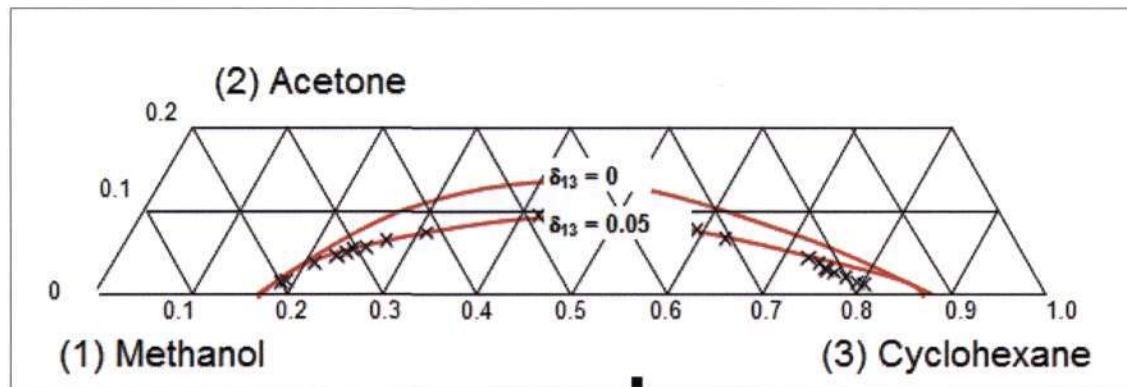
Figure 6-15 Binodale curves in the ternary system Methanol(1) – Acetone(2) – Cyclohexane (3) at 25°C (Nagata, 1984) from FlexQUAC-Q calculations

Figure 6-15 shows the results of the ternary LLE calculation and the effect of varying the FlexQUAC-Q parameter δ_{13}. $\delta_{13} = 0$ corresponds to the UNIQUAC equation. The immiscible region in this case is over estimated. As this parameter is increased, the prediction of the immiscible region improves ($\delta_{13} = 0.05$).

Chapter 7 Conclusion & Recommendations

In this study the non-linear transformation proposed by Rarey (2005) was further developed. It was observed that if the surface fraction of the UNIQUAC model was transformed in a similar way, then the resulting model (FlexQUAC-Q) could provide a more suitable description of asymmetric systems.

The transformation was applied to the residual part of the UNIQUAC equation and the activity coefficient equations had to be derived. The combinatorial term was not modified, however both the original Guggenheim-Stavermann expression and that of Weidlich and Gmehling (1987) were used in conjunction with the transformed residual term. Since G^E could not be expressed as an explicit function of surface fraction, the residual part was split into two parts. The first part was a function of liquid composition and the latter part was a function of surface fraction. This re-formulated version of the UNIQUAC model was tested against the original for verification and both were in excellent agreement. The transformation was then applied to the surface fraction in

the latter part and the activity coefficient expressions were derived and checked via both numerical and analytical derivatives.

The equations were first simplified and then programmed into a Fortran subroutine for implementation into the regression tool, RECVAL of the DDB. An EXCEL program was also developed. This program utilized the Fortran subroutine to compute the activity coefficients from the FlexQUAC-Q model. The regression in EXCEL provided the binary model parameters for comparing the UNIQUAC, FlexQUAC and FlexQUAC-Q (Guggenheim-Stavermann and Weidlich-Gmehling combinatorial terms) models.

The FlexQUAC-Q model was subjected to a similar performance evaluation as that of its precursor, the FlexQUAC model. Data from the DDB was utilized to compare the models with the objective function defined as the mean relative squared deviation in pressure. A minor mean relative reduction of about 3% of the objective function using FlexQUAC-Q compared to FlexQUAC was observed compared to a reduction by about 53% relative to the UNIQUAC-results. While in case of UNIQUAC approx. 48% of the data sets showed a final objective function of 10^{-3}, in case of the Flex-models this was achieved by about 63% of the data. This illustrated that the both FlexQUAC and FlexQUAC-Q behave similarly.

It was also observed that the results obtained using both FlexQUAC and FlexQUAC-Q models and be compared to that of the GEQUAC model. GEQUAC (Ehlker and Pfennig, 2002) is an exact quasi-chemical multisegment model, which follows a more exact approach to Guggenheim's concept of local composition. GEQUAC's ability to accurately predict non-ideal systems is noteworthy; however its mathematical complexity and large number of model parameters prove to be a disadvantage. Both Flex-models boast considerable simplicity when compared to GEQUAC and contain only one additional parameter in addition to the two existing UNIQUAC model parameters. Both models also produce accurate descriptions of non-ideal systems with a quality comparable to that of GEQUAC.

Due to its greatly increased flexibility, FlexQUAC-Q now allows one to regress a much larger range of binary data within their experimental uncertainty than was possible with NRTL and

UNIQUAC. This could significantly increase the reliability of pre-regressed parameters distributed e.g. with process simulation software.

Ternary VLE data calculations were performed with the FlexQUAC-Q model and compared to FlexQUAC and UNIQUAC. Model interaction parameters were obtained from the regression of the binary VLE data. The constituent binary VLE data sets of the thirteen ternary VLE data sets showed moderate asymmetry and it was expected that FlexQUAC-Q would not have any significant improvement in the correlation of data when compared to the FlexQUAC model. However, the correlation of the binary VLE data did improve when compared to UNIQUAC and consequently there was an improvement in the prediction of ternary VLE data. FlexQUAC-Q's prediction of the ternary VLE data is comparable to that of FlexQUAC. It is evident that the UNIQUAC model's predictive capability is retained in FlexQUAC-Q.

FlexQUAC was able to describe VLE and LLE simultaneously. The previously used system Water-1-Pentanol was used to test whether the ability to simultaneously correlate VLE and LLE was retained by FlexQUAC-Q. FlexQUAC-Q succeeded in correlating both VLE and LLE data.

Ternary LLE data was also used to evaluate the performance of FlexQUAC-Q. FlexQUAC-Q gives a better description of the immiscible region while UNIQUAC over estimates this region.

The performance evaluation of FlexQUAC-Q has revealed that it has no superior effect on the correlation of data when compared to FlexQUAC. Although FlexQUAC has achieved significant results and improved the accuracy and quality of data representation, its full benefit cannot be exploited since it cannot be extended to the group contribution method UNIFAC. This is due to manner in which the transformation was applied to UNIQUAC. In the case of FlexQUAC-Q, the transformation was applied directly to the surface fraction and hence the model can be extended to the group contribution method. The model equations were defined in this study and the subsequent model is referred to as FlexFAC.

Current group contribution methods cannot simultaneously describe VLE and LLE. Hence separate parameterizations are required. The newly formed FlexFAC model should be able to simultaneously describe both VLE and LLE. In addition to its use as a predictive G^E model,

FlexFAC can also be used in group contribution equation of states via a G^E mixing rule like PSRK, VTPR or MHV2.

As a next step a comparison of FlexFAC with the UNIFAC method should be conducted. A realistic test of the group contribution equation FlexFAC, either as a G^E-model or in combination with an equation of state via an appropriate G^E mixing rule, will require the regression of a group interaction parameter matrix to a large amount of data.

References

- Abbot, M. M., Van, H. C. and Ness (1996), 'Introduction to Chemical Engineering Thermodynamics', Prentice Hall.

- Abbott, M. M., Floes, J. K., Walsh, G. E. and Ness, H. C. V. (1975), 'Vapor-liquid equilibrium', AICHE 21(1), 72-76.

- Abbott, M. M. and Ness, H. C. V. (1975), 'Vapor-liquid equilibrium', AICHE 21(1), 62-76

- Anderson, T. F. and Prausnitz, J. M. (1978), Ind. Eng. Chem. Proc. Des. Dev. 17, 552-561.

- Aubry, M., Jayles, C., Guiglion, C. (1972), 'Equilibre Liquide-Vapeur des Melanges Dioxyde de Soufre-Chloro 1-Alcenes. Correlations sur les Coefficients d'Activite Dilution Infinie', Chim. Ind. Genie Chim. (Paris), 105(20), 1429 - 1435

- Balcazar-Ortiz, A. M., Patel, R. B., Abbott, M. M. and Ness, H. C. V (1979), 'Excess thermodynamic functions for ternary systems', J. Chem. Eng. Data 24(2), 133-136.

- Bendova, M., Rehak, K., Matous, J. and Novak, J. P. (2003), 'Liquid-liquid equilibrium and excess enthalpies in binary systems methylcyclohexane+methanol and methylcyclohexane+n,n-dmethylformamide', J. Chem. Eng. Data 48, 152-157.

- Brandstaetter, M. (1943), 'Isodimorphie bei einem angelblichen Fall von Mischkristallbildung nach Typus Ia', Z. Phys. Chem., 192, 82-84

- Cea, P., Laufente C., Sanz M. L., Royo F. M., Urieta J. S. (1995), 'Isobaric Vapor-Liquid Equilibrium Measurements on 2-Chlorobutane + Isomer Butanols at 60°C and 101.3kPa', J. Chem. Eng. Data, 40(3), 692 - 695

- Cha, T. H. and Prausnitz, J. M. (1985), 'Thermodynamic method for simultaneous representation of ternary vapor-liquid and liquid-liquid equilibria', Ind. Eng. Chem. Proc. Des. Dev. 24, 551-555.

- Chaudhry, M. M., Ness, H. C. V. and Abbott, M. M. (1980), 'Excess thermodynamic functions for ternary systems. 6. total pressure data and G^E for acetone-ethanol-water at 50 deg. C', J. Chem. Eng. Data 25(3), 254-257.

- Cheesman G.H., Ladner W.R. (1955), 'The variation of the heat of mixing with temperature', Proc.Roy.Soc.London Ser.A , 229, 387-395

- Cho, T-H., Ochi, K., Kojima, K. (1984), 'Isobaric vapour liquid for binary systems with limited miscibility, Water-n-amyl Alcohol and Water-Isoamyl Alcohol', Kagaka Kogaku Ronbunshu 10(2), 181-183

- Derr, E.L., Deal C.H. Jr. (1969), 'Analytical Solution of groups', Inst. Chem. Eng. Symp. Ser. (London)

- Dortmund Data Bank (DDB) (2006), DDBST GmBH (www.ddbst.de), Oldenburg, Germany

- Ehlker, G. H. and Pfenning Andreas (2002), 'Development of GEQUAC as a new group contribution method for strongly non-ideal mixtures', Fluid Phase Equilibria 203, 53-69

- Elsi, D. P. D., Pate, R. B., Abbott, M. M. and Ness, H. C. V. (1978), 'Excess thermodynamic function for ternary systems. 3. total pressure data and G^E for acetone-acetonitrile-methyl acetate at 50 deg. c', J. Chem. Eng. Data 23(3), 242-245.

- Gierycz, P. (1996), 'Comparison and verification of methods for correlation and prediction of VLE in multi-component systems', Polish Journal of Chemistry 70(1), 1373-1401.

- Gora, M., Kolasinska, G., Oracz, P. and Warycha, S. (1985), 'Vapor- liquid equilibria ii - the ternary system methanol-chloroform-acetone at 313.15K and 323.15K', Fluid Phase Equilibria 23, 89-116.

- Goral, M., Orac, P., Wilcek-Vera, G. and Warycha, S. (1995), 'Vapor-liquid equilibria viii - the ternary system cyclohexane-methanol-acetone at 313.15K', Fluid Phase Equilibria 112, 291-306.

- Goral, M., Oracz, P. and Warycha, S. (1998), 'Vapor-liquid equilibria xiii - the ternary system cyclohexane-methanol-hexane at 303.15K', FluidPhase Equilibria 152, 109-120.

- Goral, M., Oracz, P. and Warycha, S. (2000), 'Vapour-liquid equilibria xiv - the ternary system cyclohexane-methanol-hexane at 293.15K', Fluid Phase Equilibria 169, 85-99.

- Goral, M., P.Oracz and Warycha, S. (1997), 'Vapor-liquid equilibria vi – the ternary system acetone-hexane-cyclohexane at 313.15K', Fluid Phase Equilibria 129, 223-233.

- Goral, M.., Zawadski, S. (1993), " Vapour-liquid equilibria in non-polar mixtures. II. Carbon tetrachloride with alkylbenzenes and n-alkanes at 313.15K", Fluid Phase Equilibria., 90(1), 355 – 364

REFERENCES

- Kikic, I., Alessi, P., Rasmussen P. and Fredenslund A. (1980), 'On the Combinatorial Part of the UNIFAC and UNIQUAC Models', The Canadian Journal of Chemical Engineering Vol 58, 253-258

- Kolasinska, G., Goral, M. (1982), Giza J. Z. Phys. Chem. (Leipzig) 263(1)

- Kudryatseva, L. S., Susarev, M. P. (1963), J. Appl. Chem. USSR 36(7), 1419

- Oracz, P., Goral, M., Wilczek-Vera, G. and Warycha, S. (1996), 'Vapor-liquid equilibria x - the ternary system cyclohexane-methanol-acetone at 293.15K and 303.15K', Fluid Phase Equilibria 126, 71-92.

- Long R.B. (1963), 'Xylene Azeotropes with 1,1,2,2-Tetrachloroethane', J.Chem.Eng.Data, 8, p174-176,

- Ng H.-J., Chen C.-J., Razzaghi M. (1993), 'Vapor-Liquid Equilibria of Selected Aromatic Hydrocarbons in Triethylene Glycol', Fluid Phase Equilib., 82, p207-214

- Oracz, P. and Warycha, S. (1995), 'Vapor-liquid equilibria vii - the ternary system hexane-methanol-acetone at 313.15k', Fluid Phase Equilibria 108, 199-211.

- Perry, R. and Green, D. (1998), 'Perry's Chemical Engineers' Handbook, 7th edn', McGraw-Hill.

- Prausnitz, J. M. (1969), 'Molecular Thermodynamics of Fluid -Phase Equilibria', Prentice-Hall.

- Prausnitz, J. M., Lichenthaler, R. N., de Azevedo, E. G. (1999), 'Molecular Thermodynamics of Fluid -Phase Equilibria', 3rd edition, Prentice-Hall.

- Raal, J. and Muhlbauer, A. (1998), 'Phase Equilibria:Measurement and Computation', Taylor and Francis.

- Rall, W. and Schaefer K. (1959), Z. Elektrochem. 63(9/10), 1019

- Rarey, J. (2005), 'Extended flexibility for G^E-models and simultaneous description of VLE and LLE using a non-linear transformation of the concentration dependence', Ind. Eng. Chem. Proc. Des. Dev. 44, 7600-7608

- Ryland G. (1899), 'A Contribution to the Study of Liquid Mixtures of Constant Boiling-Point', Am. Chem. J., 22, 384-397

- Signer R., Arm H., Daeniker H. (1969), 'Untersuchungen uber das Verhalten organischer Mischphasen. 8. Mitteilung: Dampfdruecke, Dichten, thermodynamische Mischungfunktionen und Brechungsindices der binaeren systeme Wasser-

Tetrahydrofuran und Wasser-Diaethylaether bei 25°C', Helv. Chim. Acta. 52(8), 2347 – 2351

- Tamura, K., Osaki, A., Murakami, S., Ohji, H., Ogawa, H., Laurent, B., Grolier, J. P. E. (1999), 'Thermodynamic properties of binary mixtures{an alkoxyethanol + n-octane} Excess molar enthalpies and excess molar heat capacities at 298.15K', Fluid Phase Equilibria, 156, 137-147

- Tochigi K., Tiegs, D., Gmehling J. and Kojima K. (1990), 'Determination of new ASOG parameters', Journal of Chemical Engineering of Japan Vol 23 No. 4

- Walas, S. M. (1985), Phase Equilibria in Chemical Engineering, Butterworth, Boston.

- Wilson S. R., Patel R. B., Abbott M. M. and Ness H. C. V. (1979),'Excess thermodynamic functions for ternary systems', J. Chem. Eng. Data 24(2), 130-132.

- Wood, S. E. (1937), 'Deviations of Carbon Tetrachloride and Silicon Tetrachloride Solutions from Raoult's Law', J. Am. Chem. Soc., 59, 1510-1514

- Zabaloy M. S., Mabe G. D. B., Bottini S. B., Brignole E. A. (1993), 'Isothermal Vapour-Liquid Equilibrium Data for Binaries Propane-2-Propanol and Propylene-2-Propanol', J. Chem. Eng. Data, 38(1) 40 – 43

APPENDICES

APPENDIX A

APPENDIX A – UNIQUAC DERIVATION

A brief derivation based on Maurer & Prausnitz (1978) is presented here since UNIQUAC is the precursor to both the FlexQUAC and FlexQUAC-Q model.

Here a binary mixture containing molecules of component 1 and 2 with arbitrary size and shape is considered. Component 1 has r_1 segments with external surface area proportional to q_1. Analogously, component 2 has r_2 segments and external surface area proportional to q_2. For a unisegmental molecule which is small and spherical, $r = q = 1$. In the case of chain molecules the ratio of q to r becomes less than one and as the chain increases in number of segments, this ratio approaches 2/3.

In this derivation a unisegmental molecule is considered here. If one considers that the intermolecular forces are restricted to a small range and coupled with pair-wise additivity, then the energy required to vaporize this molecule to ideal gas state corresponds to $\frac{1}{2}Z^{(0)}U_{11}^{(0)}$. Here Z refers to the coordination number. Pure liquid is represented by (0) and $U_{11}^{(0)}$ refers to the potential energy of the two neigbouring molecules of the molecule 1. This molecule is then condensed into the hypothetical fluid (indicated by (1)). In this case the molecule now has $Z^{(1)}\theta_{11}$ neighbours of species 1 and $Z^{(1)}\theta_{21}$ neighbours of species 2. The local surface fraction of component 1 is defined as θ_{11} about central molecule 1. Similarly θ_{21} represents the local surface fraction of component 2, about the central molecule 1. Hence $\theta_{11} + \theta_{21} = 1$. With the assumption that $Z^{(1)}$ is the same as $Z^{(0)}$, then the energy associated with the condensation process is $\frac{1}{2}Z\left[\theta_{11}U_{11}^{(1)} + \theta_{21}U_{21}^{(1)}\right]$. With the same idea, molecule 2 can be vaporized from the pure liquid to a second hypothetical fluid.

In the context of x_1 moles of fluid 1 and x_2 moles of fluid 2, then the mixture extensive configurational property M is $M = x_1 M^{(1)} + x_2 M^{(2)}$ where $M^{(1)}$ and $M^{(2)}$ are the extensive configurational property for hypothetical fluid 1 and 2 respectively.

Thus the total energy of mixing, U^E corresponding to the transfer of x_1 moles of species 1 from the pure liquid 1 and x_2 moles of species 2 from the pure liquid 2 into the two-liquid mixture is

$$U^E = \tfrac{1}{2} Z x_1 N_A \left[q_1 \left(\theta_{11} U_{11}^{(1)} + \theta_{21} U_{21}^{(1)} - U_{11}^{(0)} \right) \right] + \tfrac{1}{2} Z x_2 N_A \left[q_2 \left(\theta_{22} U_{22}^{(2)} + \theta_{12} U_{12}^{(2)} - U_{22}^{(0)} \right) \right] \quad \text{A-1}$$

where N_A is Avagadro's number. The local surface fractions must obey the conservation equations; hence $\theta_{11} + \theta_{21} = 1$ and $\theta_{12} + \theta_{22} = 1$. Assume that $U_{11}^{(1)} = U_{11}^{(0)}$ and $U_{22}^{(1)} = U_{22}^{(0)}$ and **Equation A-1** reduces to

$$U^E = \tfrac{1}{2} Z N_A \left[x_1 \theta_{21} q_1 (U_{21} - U_{11}) + x_2 \theta_{12} q_2 (U_{12} - U_{22}) \right] \quad \text{A-2}$$

Wilson (1964) assumed that the local compositions are related to overall compositions via Boltzmann factors. This is the crucial idea in this derivation. Hence,

$$\frac{\theta_{21}}{\theta_{11}} = \frac{\theta_2}{\theta_1} \exp \left[-\tfrac{1}{2} Z \frac{(U_{21} - U_{11})}{kT} \right] \quad \text{A-3}$$

$$\frac{\theta_{12}}{\theta_{22}} = \frac{\theta_1}{\theta_2} \exp \left[-\tfrac{1}{2} Z \frac{(U_{12} - U_{22})}{kT} \right] \quad \text{A-4}$$

where

- $\theta_1 = \dfrac{x_1 q_1}{x_1 q_1 + x_2 q_2}$ and $\theta_2 = \dfrac{x_2 q_2}{x_1 q_1 + x_2 q_2}$

Thus the fundamental relation based on the two fluid theory is

$$U^E = x_1 q_1 \theta_{21} \Delta u_{21} + x_2 q_2 \theta_{12} \Delta u_{12} \quad \text{A-5}$$

where

- $\theta_{21} = \dfrac{\theta_2 \exp(-\Delta u_{21}/RT)}{\theta_1 + \theta_2 \exp(-\Delta u_{21}/RT)}$

- $\theta_{12} = \dfrac{\theta_1 \exp(-\Delta u_{12}/RT)}{\theta_2 + \theta_1 \exp(-\Delta u_{12}/RT)}$

- $\Delta u_{21} = \frac{1}{2} Z(U_{21} - U_{11}) N_A$

- $\Delta u_{12} = \frac{1}{2} Z(U_{12} - U_{22}) N_A$

Now the excess Helmhotz energy, A^E can be expressed as

$$\frac{d(A^E/T)}{d(1/T)} = U^E \qquad \text{A-6}$$

Integrating from $1/T_o$ to $1/T$ yields the following result

$$\frac{A^E}{T} = \int_{1/T_o}^{1/T} U^E d(1/T) + B \qquad \text{A-7}$$

where

- B is the constant of integration

B can be evaluated by letting 1/To approach zero.

At high temperatures, an athermal mixture is formed and Guggenheim's equation for athermal mixtures of molecules of arbitrary size and shape is suitable as a boundary condition.

$$\left(\frac{A^E}{RT}\right)_{athermal} = -\left(\frac{S^E}{R}\right)_{combinatorial} = x_1 \ln \frac{\Phi_1}{x_1} + x_2 \ln \frac{\Phi_2}{x_2} + \frac{1}{2} Z\left(q_1 x_1 \ln \frac{\theta_1}{\Phi_1} + q_2 x_2 \ln \frac{\theta_2}{\Phi_2}\right) \text{A-8}$$

where

- $\Phi_1 = \dfrac{x_1 r_1}{x_1 r_1 + x_2 r_2}$

- $\Phi_2 = \dfrac{x_2 r_2}{x_1 r_1 + x_2 r_2}$

Using the assumption that Δu_{21} and Δu_{12} are independent of temperature, then it follows at low pressure $(A^E)_{T,V} \approx (G^E)_{T,P}$. Hence

$$\left(\frac{A^E}{RT}\right)_{T,V} \approx \left(\frac{G^E}{RT}\right)_{T,P} = \left(\frac{G^E}{RT}\right)_{combinatorial} + \left(\frac{G^E}{RT}\right)_{residual} \qquad \text{A-9}$$

where

- $$\frac{G^E}{RT}_{combinatorial} = x_1 \ln\frac{\Phi_1}{x_1} + x_2 \ln\frac{\Phi_2}{x_2} + \frac{Z}{2}\left(q_1 x_1 \ln\frac{\theta_1}{\Phi_1} + q_2 x_2 \ln\frac{\theta_2}{\Phi_2}\right) \qquad \text{A-10}$$

- $$\frac{G^E}{RT}_{residual} = -q_1 x_1 \ln(\theta_1 + \theta_2 \tau_{21}) - q_2 x_2 \ln(\theta_2 + \theta_1 \tau_{12}) \qquad \text{A-11}$$

Appendix A

A2 - FlexQUAC-Q Derivation:

Test Using Numerical Example

TEST SYSTEM : 1.DIETHYL ETHER, 2. ACETONITRILE

Please note that all numerical evaluations are highlighted in yellow and have been calculated for $x_1 = 1$

Definition of parameters

general tolerance criterion

$$\text{eps1} \equiv 10^{-8}$$

number of components n

$$n := 2$$

range variables

$$k := 1..n \qquad j := 1..n \qquad i := 1..n$$

Liquid mole fraction

$$x(xx) := \begin{pmatrix} xx \\ 1 - xx \end{pmatrix}$$

The variable xx allows the user to perform the calculations for varying liquid mole compositions

UNIQUAC Parameters

$$R := \begin{pmatrix} 3.395 \\ 1.87 \end{pmatrix} \qquad Q := \begin{pmatrix} 3.016 \\ 1.724 \end{pmatrix}$$

Model interaction parameters

$$\Delta u := \begin{pmatrix} 0 & 497.605 \\ -76.972 & 0 \end{pmatrix}$$

FlexQUAC Q Parameter

$$\delta f(d) := \begin{pmatrix} 0 & d \\ d & 0 \end{pmatrix}$$

$$d := 0.1786$$

Universal Gas Constant cal/mol K

$$RG := 1.9872$$

System Temperature / K

$$T := 273.15 + 20.5$$

-192-

UNIQUAC - Residual Part

Formulae:

$$\tau(i,j,\Delta u,T) := \exp\left(\frac{-\Delta u_{i,j}}{RG \cdot T}\right)$$

$$\Theta o(x,Q,i,n) := \frac{\left(x_i \cdot Q_i\right)}{\displaystyle\sum_{k=1}^{n} x_k \cdot Q_k} \qquad \text{Surface Fraction}$$

$$So(x,Q,\Delta u,T,i,n) := \sum_{j=1}^{n} \Theta o(x,Q,j,n) \cdot \tau(j,i,\Delta u,T)$$

$$gE\_RTreso(x,T,\Delta u,Q,n) := -\sum_{i=1}^{n} x_i \cdot Q_i \cdot \ln\left(So(x,Q,\Delta u,T,i,n)\right) \qquad \text{Original residual part of UNIQUAC}$$

Section 1: UNIQUAC Model rewritten

$$gE\_RTreso1(x,T,\Delta u,Q,n) := -\sum_{i=1}^{n} x_i \cdot Q_i \cdot \sum_{i=1}^{n} \Theta o(x,Q,i,n) \cdot \ln\left(So(x,Q,\Delta u,T,i,n)\right)$$

$$gE\_RTreso1(x(1),T,\Delta u,Q,n) = 0$$

The function Θo can be defined as a vector:

$$\Theta(nn,Q,n) := \begin{array}{|l} \text{for } i \in 1..n \\[2ex] \quad ret_i \leftarrow \dfrac{\left(nn_i \cdot Q_i\right)}{\displaystyle\sum_{k=1}^{n} nn_k \cdot Q_k} \\[4ex] ret \end{array}$$

$$\Theta(x(1),Q,n) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Section 2: Derivation of G^E/RT by n for Original UNIQUAC using functions f1 and f2

G^E residual is the product of two functions:

$$f1x(x,Q) := -\sum_{i=1}^{n} x_i \cdot Q_i \qquad\qquad f2x(x,T,\Delta u,Q,n) := \sum_{i=1}^{n} \Theta o(x,Q,i,n) \cdot \ln\big(So(x,Q,\Delta u,T,i,n)\big)$$

Now we want to derive G^E by n. Following the product rule, this is

$$gE\_RTreso\_dn(x,T,\Delta u,Q,n) = f1x\_dn(x,Q) \cdot f2x(x,T,\Delta u,Q,n) + f1x(x,Q) \cdot f2x\_dn(x,T,\Delta u,Q,n)$$

Analytical Dervation of f1

Now $f1x(x,Q)$ can be re-written as a function of mole numbers $f1n(nn,Q)$

$$f1n(nn,Q) := \frac{-1}{\displaystyle\sum_{i=1}^{n} nn_i} \left(\sum_{i=1}^{n} nn_i \cdot Q_i\right)$$

-194-

This is the product of 2 functions f1an and f1bn

$$f1an(nn) := \frac{-1}{\displaystyle\sum_{i=1}^{n} nn_i}$$

$$f1bn(nn, Q) := \left(\sum_{i=1}^{n} nn_i \cdot Q_i\right)$$

with the derivations

$$f1an\_dn(nn) := \frac{1}{\left(\displaystyle\sum_{i=1}^{n} nn_i\right)^2}$$

$$f1bn\_dn(nn, Q, j) := Q_j$$

Using the product rule, it holds that

$$f1n\_dn(nn, Q, i) = f1an(nn) \cdot f1bn\_dn(nn, Q, i) + f1bn(nn, Q) \cdot f1an\_dn(nn)$$

This leads to

$$f1n\_dn(nn, Q, j) := \frac{-Q_j}{\displaystyle\sum_{i=1}^{n} nn_i} + \frac{\displaystyle\sum_{i=1}^{n} nn_i \cdot Q_i}{\left(\displaystyle\sum_{i=1}^{n} nn_i\right)^2}$$

Using $nn_i = x_i$ and $\displaystyle\sum_{i=1}^{n} nn_i = 1$ this leads to $f1\_dn(x, Q, j) := -Q_j + \displaystyle\sum_{i=1}^{n} x_i \cdot Q_i$

Putting f1 into vector form:

$$df1dn(x, Q, n) := \begin{vmatrix} \text{for } m \in 1..n \\ \quad ret_m \leftarrow -Q_m + \displaystyle\sum_{i=1}^{n} x_i \cdot Q_i \\ ret \end{vmatrix}$$

$$df1dn(x(1), Q, n) = \begin{pmatrix} 0 \\ 1.292 \end{pmatrix}$$

Numerical Derivation of f1

$$df1\Theta d\Theta N(x, Q, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \begin{array}{|l} \text{for } m \in 1..n \\ \quad eps_m \leftarrow 0 \\ eps_i \leftarrow eps1 \\ ret_i \leftarrow \left(\dfrac{f1n(x + eps, Q) - f1n(x - eps, Q)}{2.eps1} \right) \end{array} \\ ret \end{array}$$

$$df1\Theta d\Theta N(x(1), Q, n) = \begin{pmatrix} 0 \\ 1.292 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$df1\Theta d\Theta N(x(1), Q, n) - df1dn(x(1), Q, n) = \begin{pmatrix} 0 \\ -4.122 \times 10^{-9} \end{pmatrix}$$

Analytical Derivation of f2

Now f2 is a function of Θ and we want to flexibilize Θ. This means we need f2Θ_dΘ and dΘ_dn.
Because f2Θ is not a vector, f2Θ_dΘ is a vector and dΘ_dn is a Jacobian matrix.

f2Θ_dΘ is always the same (only depends on the UNIQUAC-equation).

dΘ_dn is different for original UNIQUAC and the Flex-version. Maybe you can ask a mathematician, how to combine the Jacobians dΘ_dx and dx_dn to get dΘ_dn. This would make things more easy.

from the UNIQUAC equation given above:
$$f2\Theta(x,T,\Delta u,Q,n) := \sum_{i=1}^{n} \Theta o(x,Q,i,n) \cdot \ln\left(\sum_{j=1}^{n} \Theta o(x,Q,j,n) \cdot \tau(j,i,\Delta u,T) \right)$$

again, this calls for the product rule

---

f2 is the sum of a product of two functions f2$_a$ and f2$_b$. This product is a scalar. The derivation is done by Θj. There are 2 cases: i = j and i <> j !!!!

$$f2\Theta a(\Theta,Q,i,n) := \Theta o(x,Q,i,n) \qquad\qquad f2\Theta b(\Theta,Q,i,n) := \ln\left(\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,i,\Delta u,T) \right)$$

We use the assumption i<>j for the derivation, then substract the wrong term (where i = j) and add the correct derivation of this term.

For i<>j the derivation of f2Θ_{ai} with respect to Θ_j is zero. The derivation of f2Θ_{bi} with respect to Θ_j is
$$\dfrac{\tau(j,i,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,i,\Delta u,T)}$$

So for the case i<>j following the product rule the derivation is
$$\sum_{i=1}^{n}\left(0 \cdot \ln\left(\sum_{j=1}^{n} \Theta o(x,Q,j,n) \cdot \tau(j,i,\Delta u,T) \right) + \Theta o(x,Q,i,n) \cdot \dfrac{\tau(j,i,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,i,\Delta u,T)} \right)$$

which is

$$\sum_{i=1}^{n} \left(\Theta o(x,Q,i,n) \cdot \frac{\tau(j,i,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,i,\Delta u,T)} \right)$$

we need to substract the case i = j for which we used the wrong derivation. In this case $\tau_{jj} = 1$

$$Case1(x,Q,j,n) := \sum_{i=1}^{n} \Theta o(x,Q,i,n) \cdot \frac{\tau(j,i,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,i,\Delta u,T)} - \Theta o(x,Q,j,n) \cdot \frac{\tau(j,j,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T)}$$

$Case1(x(1),Q,j,n) =$

0
1.141

For the i = j case the derivation of f2Θ_{ai} wrt Θ_j is 1. The derivation of f2Θ_{bi} wrt to Θ_j is $\dfrac{\tau(j,j,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T)}$ since $\tau_{jj} = 1$

So for the case i = j the derivation is: $\ln\left(\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T) \right) + \Theta o(x,Q,j,n) \cdot \dfrac{\tau(j,j,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T)}$

$$Case2(x,Q,j,n) := \left(\ln\left(\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T) \right) + \Theta o(x,Q,j,n) \cdot \frac{\tau(j,j,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta o(x,Q,k,n) \cdot \tau(k,j,\Delta u,T)} \right)$$

$$\text{Case2}(x(1), Q, j, n) =$$

1
-0.853

So the complete derivation is:

$$\text{df2}\Theta\text{d}\Theta\text{o}(x, Q, j, n) := \text{Case1}(x, Q, j, n) + \text{Case2}(x, Q, j, n)$$

$$\text{df2}\Theta\text{d}\Theta\text{o}(x(1), Q, j, n) =$$

1
0.288

Putting f2 in vector form:

$$\text{df2}\Theta\text{d}\Theta(x, Q, n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{ret}_i \leftarrow \text{df2}\Theta\text{d}\Theta\text{o}(x, Q, i, n) \\ \text{ret} \end{vmatrix}$$

$$\text{df2}\Theta\text{d}\Theta(x(1), Q, n) = \begin{pmatrix} 1 \\ 0.288 \end{pmatrix}$$

Numerical Derivation of f2

The numerical derivation has to be performed at $\Theta(x)$ but the change must be in Θ, not in x

$$f2\Theta(\Theta,T,\Delta u,Q,n) := \sum_{i=1}^{n} \Theta_i \cdot \ln\left(\sum_{j=1}^{n} \Theta_j \cdot \tau(j,i,\Delta u,T)\right)$$

$$\Theta(x(1),Q,n) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$df2\Theta d\Theta N(nn,Q,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \begin{vmatrix} \text{for } m \in 1..n \\ \quad eps_m \leftarrow 0 \\ eps_i \leftarrow eps1 \\ ret_i \leftarrow \left(\dfrac{f2\Theta(\Theta(nn,Q,n) + eps,T,\Delta u,Q,n) - f2\Theta(\Theta(nn,Q,n) - eps,T,\Delta u,Q,n)}{2.eps1}\right) \end{vmatrix} \\ ret \end{vmatrix}$$

$$df2\Theta d\Theta N(x(1),Q,n) = \begin{pmatrix} 1 \\ 0.288 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$df2\Theta d\Theta N(x(1),Q,n) - df2\Theta d\Theta(x(1),Q,n) = \begin{pmatrix} -5.264 \times 10^{-10} \\ -4.301 \times 10^{-11} \end{pmatrix}$$

Via Partial Numerical Derivation of of each Summand in f2

$$f2\Theta i(\Theta, T, \Delta u, Q, n, i) := \Theta_i \cdot \ln\left(\sum_{j=1}^{n} \Theta_j \cdot \tau(j, i, \Delta u, T)\right)$$

$$df2\Theta id\Theta jN(nn, Q, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \begin{array}{|l} \text{for } m \in 1..n \\ \quad eps_m \leftarrow 0 \\ eps_i \leftarrow eps1 \\ \text{for } j \in 1..n \\ \quad ret_{i,j} \leftarrow \left(\dfrac{f2\Theta i(\Theta(nn,Q,n) + eps, T, \Delta u, Q, n, j) - f2\Theta i(\Theta(nn,Q,n) - eps, T, \Delta u, Q, n, j)}{2.eps1}\right) \end{array} \\ ret \end{array}$$

$$df2\Theta id\Theta jN(x(1), Q, n) = \begin{pmatrix} 1 & 0 \\ 1.141 & -0.853 \end{pmatrix}$$

⊡

$$\sum_{j=1}^{n} df2\Theta id\Theta jN(x(1), Q, n)_{i,j} =$$

1
0.288

Section 3: Derivation of Jacobian matrix for surface fraction

▾

Analytical Derivation of Θ by n

$$\text{Jacobian}\Theta(x,Q) := \begin{array}{|l} \text{for } k \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \text{Jac}_{k,m} \leftarrow \dfrac{Q_m\left(\sum\limits_{k=1}^{n} x_k \cdot Q_k\right) - x_k \cdot Q_k \cdot Q_m}{\left(\sum\limits_{k=1}^{n} x_k \cdot Q_k\right)^2} \quad \text{if } k = m \\ \qquad \text{Jac}_{k,m} \leftarrow \dfrac{-x_m \cdot Q_m \cdot Q_k}{\left(\sum\limits_{k=1}^{n} x_k \cdot Q_k\right)^2} \quad \text{if } k \neq m \\ \text{Jac} \end{array}$$

$$\text{Jacobian}\Theta(x(1),Q) = \begin{pmatrix} 0 & 0 \\ -0.572 & 0.572 \end{pmatrix}$$

Numerical Derivation of Θ by n

$$\text{JacobianN}\Theta(x,Q,n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \begin{array}{|l} \text{for } m \in 1..n \\ \quad eps_m \leftarrow 0. \\ eps_i \leftarrow eps1 \\ x1 \leftarrow \Theta\left(\dfrac{x - eps}{1 - eps1}, Q, n\right) \\ x2 \leftarrow \Theta\left(\dfrac{x + eps}{1 + eps1}, Q, n\right) \\ \text{for } m \in 1..n \\ \quad \text{Jac}_{i,m} \leftarrow \dfrac{x2_m - x1_m}{2 \cdot eps1} \end{array} \\ \text{Jac} \end{array}$$

$$\text{JacobianN}\Theta(x(1),Q,n) = \begin{pmatrix} 0 & 0 \\ -0.572 & 0.572 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$\text{Jacobian}\Theta(x(0.098),Q) - \text{JacobianN}\Theta(x(0.098),Q,n) = \begin{pmatrix} 1.369 \times 10^{-9} & 1.846 \times 10^{-11} \\ -1.795 \times 10^{-10} & 4.343 \times 10^{-9} \end{pmatrix}$$

Now f2 was derived by Θo. This derivative need to be multiplied by the Jacobian of Θ by n to calculate the derivative of f2 by n.

$\text{df2dn}(x,Q,n) := \text{Jacobian}\Theta(x,Q) \cdot \text{df2}\Theta\text{d}\Theta(x,Q,n)$

$$\text{df2dn}(x(1),Q,n) = \begin{pmatrix} 0 \\ -0.407 \end{pmatrix}$$

Now test df2dn numerically

$\text{df2dnN}(x,Q,n) := \text{JacobianN}\Theta(x,Q,n) \cdot \text{df2}\Theta\text{d}\Theta\text{N}(x,Q,n)$

$$\text{df2dnN}(x(1),Q,n) = \begin{pmatrix} 0 \\ -0.407 \end{pmatrix}$$

This gives for dgE_res/RT / dn

$$dgE\_RTres1\_dn(x, T, \Delta u, Q, n) := \left(df1dn(x, Q, n) \cdot f2x(x, T, \Delta u, Q, n) + f1x(x, Q) \cdot df2dn(x, Q, n) \right)$$

$$dgE\_RTres1\_dn(x(1), T, \Delta u, Q, n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

Test dgE_res/RT / dn numerically

$$dgE\_RTres1\_dnN(x, T, \Delta u, Q, n) := \left(df1\Theta d\Theta N(x, Q, n) \cdot f2x(x, T, \Delta u, Q, n) + f1x(x, Q) \cdot df2dnN(x, Q, n) \right)$$

$$dgE\_RTres1\_dnN(x(1), T, \Delta u, Q, n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

Calculate ln_gamma_res using dgE_res/RT / dn

$$\ln\gamma\_res1(x, T, \Delta u, Q, n) := dgE\_RTres1\_dn(x, T, \Delta u, Q, n) + gE\_RTreso1(x, T, \Delta u, Q, n)$$

$$\ln\gamma\_res1(x(1), T, \Delta u, Q, n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

$$\ln\gamma\_res1N(x,T,\Delta u,Q,n) := dgE\_RTres1\_dnN(x,T,\Delta u,Q,n) + gE\_RTreso1(x,T,\Delta u,Q,n)$$

$$\ln\gamma\_res1N(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

FLEXQUAC-Q

G^E residual [FLEXQUAC] is the product of two functions:

$$\Theta f(x,Q,\delta,i,n) := \left[\frac{x_i \cdot Q_i \cdot \left[1 + \left(\displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right) \right]}{\left[\displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \displaystyle\sum_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l} \right) \right]} \right]$$

New surface fraction definition which includes the FlexQUAC-Q parameter

$$\Theta fn(x,\delta,Q,n) := \begin{array}{|l} \text{for } i \in 1..n \\[2em] \quad ret_i \leftarrow x_i \cdot Q_i \cdot \left[\dfrac{1 + \left(\displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)}{\left[\displaystyle\sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \displaystyle\sum_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l} \right) \right]} \right] \\[2em] ret \end{array}$$

-205-

$$Sf(x,Q,\Delta u,T,\delta,i,n) := \sum_{j=1}^{n} \Theta f(x,Q,\delta,j,n) \cdot \tau(j,i,\Delta u,T)$$

$$Sf(x(1),Q,\Delta u,T,\delta f(d),1,n) = 1$$

$$Sf(x(1),Q,\Delta u,T,\delta f(d),2,n) = 0.426$$

$$f1x(x,Q) := -\sum_{i=1}^{n} x_i \cdot Q_i \qquad f2\Theta f(x,T,Q,\delta,n) := \sum_{i=1}^{n} \Theta f(x,Q,\delta,i,n) \cdot \ln(Sf(x,Q,\Delta u,T,\delta,i,n))$$

$$f1x(x(1),Q) = -3.016$$

$$f2\Theta f(x(1),T,Q,\delta f(d),n) = 0$$

$$gE\_RTresf(x,T,\Delta u,Q,\delta,n) := -\sum_{i=1}^{n} x_i \cdot Q_i \cdot \sum_{i=1}^{n} \Theta f(x,Q,\delta,i,n) \cdot \ln(Sf(x,Q,\Delta u,T,\delta,i,n))$$

$$gE\_RTresf(x(1),T,\Delta u,Q,\delta f(d),n) = 0$$

Section 4: Analytical Derivation of f2 by Θf

$$df2\Theta fd\Theta f(x,Q,\delta,j,n) := \sum_{i=1}^{n} \left(\frac{\Theta f(x,Q,\delta,i,n) \cdot \tau(j,i,\Delta u,T)}{\displaystyle\sum_{k=1}^{n} \Theta f(x,Q,\delta,k,n) \cdot \tau(k,i,\Delta u,T)} \right) + \ln\left(\sum_{k=1}^{n} \Theta f(x,Q,\delta,k,n) \cdot \tau(k,j,\Delta u,T) \right)$$

Putting in vector form:

$$df2d\Theta f(x,Q,\delta,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad ret_i \leftarrow df2\Theta fd\Theta f(x,Q,\delta,i,n) \\ ret \end{vmatrix}$$

$$df2d\Theta f(x(1),Q,\delta f(d),n) = \begin{pmatrix} 1 \\ 0.288 \end{pmatrix}$$

Numerical Derivation of f2 by Θf

The numerical derivation has to be performed at $\Theta(x)$ but the change must be in Θ, not in x

$$f2\Theta f(\Theta fn,T,\Delta u,Q,n) := \sum_{i=1}^{n} \Theta fn_i \cdot \ln\left(\sum_{j=1}^{n} \Theta fn_j \cdot \tau(j,i,\Delta u,T)\right)$$

$$df2\Theta fd\Theta N(x,Q,\delta,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \begin{vmatrix} \text{for } m \in 1..n \\ \quad eps_m \leftarrow 0 \\ eps_i \leftarrow eps1 \\ ret_i \leftarrow \left(\dfrac{f2\Theta f(\Theta fn(x,\delta,Q,n) + eps,T,\Delta u,Q,n) - f2\Theta f(\Theta fn(x,\delta,Q,n) - eps,T,\Delta u,Q,n)}{2.eps1}\right) \end{vmatrix} \\ ret \end{vmatrix}$$

$$df2\Theta fd\Theta N(x(1),Q,\delta f(d),n) = \begin{pmatrix} 1 \\ 0.288 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$\text{df2}\Theta\text{fd}\Theta\text{N}(x(1),Q,\delta f(d),n) - \text{df2d}\Theta f(x(1),Q,\delta f(d),n) = \begin{pmatrix} -5.264 \times 10^{-10} \\ -4.301 \times 10^{-11} \end{pmatrix}$$

Section 5: Analytical Derivation of Θf by n

The derivation of thetha by n is made easier by breaking the expression into fragments and then using differentiation rules to obtain the derivation.

Theta = A (B/C)

Hence it follows:

Θfn derived by n gives $\left(\dfrac{d}{dn}A\right).\dfrac{B}{C} + \left(\dfrac{d}{dn}\dfrac{B}{C}\right).A$

$\dfrac{B}{C}$ derived by n gives $\dfrac{\left(\dfrac{d}{dn}B\right)C + \left(\dfrac{d}{dn}C\right).B}{C^2}$

-208-

$$\text{JacobianFQ1}(x, Q, \delta, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \end{array}$$

$$\text{Jac}_{m,i} \leftarrow x_i \cdot Q_i \cdot \frac{\left(1 + Q_m \cdot \delta_{i,m}\right) \cdot \left[\sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l}\right)\right] - \left[Q_m \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{m,l}\right) + \sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_m \cdot \delta_{k,m}\right.\right.}{\left[\sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l}\right)\right]^2}$$

$$\text{Jac}_{i,m} \leftarrow Q_i \cdot \left[\frac{1 + \sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k}}{\sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l}\right)}\right] + x_i \cdot Q_i \cdot \frac{\left[\sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{k,l}\right)\right] - \left[Q_i \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l \cdot \delta_{i,l}\right) + \right.}{\left[\sum\limits_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + \sum\limits_{l=1}^{n} x_l \cdot Q_l\right.\right.}$$

Jac

$$\text{JacobianFQ1}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 & 0 \\ -0.88 & 0.88 \end{pmatrix}$$

Numerical Derivation of Θf by n

$$\text{JacobianN2}_{FQ}(nn, Q, \delta, n) := \begin{array}{|l}
\text{for } i \in 1..n \\
\quad \begin{array}{|l}
\text{for } m \in 1..n \\
\quad eps_m \leftarrow 0. \\
eps_i \leftarrow eps1 \\
nn1 \leftarrow \Theta fn\left(\dfrac{nn - eps}{1 - eps1}, \delta, Q, n\right) \\
nn2 \leftarrow \Theta fn\left(\dfrac{nn + eps}{1 + eps1}, \delta, Q, n\right) \\
\text{for } m \in 1..n \\
\quad Jac_{i,m} \leftarrow \dfrac{nn2_m - nn1_m}{2 \cdot eps1}
\end{array} \\
Jac
\end{array}$$

$$\text{JacobianN2}_{FQ}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 & 0 \\ -0.88 & 0.88 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$\text{JacobianFQ1}(x(1), Q, \delta f(d), n) - \text{JacobianN2}_{FQ}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 & 0 \\ -1.717 \times 10^{-8} & 0 \end{pmatrix}$$

Now f2 was derived by Θf. This derivative need to be multiplied by the Jacobian of Θ by n to calculate the derivative of f2 by n.

$$\text{df2fdn}(x, Q, \delta, n) := \text{JacobianFQ1}(x, Q, \delta, n) \cdot \text{df2d}\Theta\text{f}(x, Q, \delta, n)$$

$$\text{df2fdn}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 \\ -0.626 \end{pmatrix}$$

Now test df2dn numerically

$$\text{df2fdnN}(x, Q, \delta, n) := \text{JacobianN2}_{FQ}(x, Q, \delta, n) \cdot \text{df2}\Theta\text{fd}\Theta\text{N}(x, Q, \delta, n)$$

$$\text{df2fdnN}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 \\ -0.626 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$\text{df2fdn}(x(1), Q, \delta f(d), n) - \text{df2fdnN}(x(1), Q, \delta f(d), n) = \begin{pmatrix} 0 \\ -1.76 \times 10^{-8} \end{pmatrix}$$

$$f2\Theta f(x,T,Q,\delta,n) := \sum_{i=1}^{n} \Theta f(x,Q,\delta,i,n) \cdot \ln(Sf(x,Q,\Delta u,T,\delta,i,n))$$

$$dgE\_RTresf\_dn(x,T,Q,\delta,n) := f2\Theta f(x,T,Q,\delta,n) \cdot df1dn(x,Q,n) + f1x(x,Q) \cdot df2fdnN(x,Q,\delta,n)$$

$$dgE\_RTresf\_dn(x(1),T,Q,\delta f(d),n) = \begin{pmatrix} 0 \\ 1.888 \end{pmatrix}$$

Test dgE_resf/RT / dn numerically

$$f2\Theta f(x,T,Q,\delta,n) := \sum_{i=1}^{n} \Theta f(x,Q,\delta,i,n) \cdot \ln(Sf(x,Q,\Delta u,T,\delta,i,n))$$

$$dgE\_RTresf\_dnN(x,T,\Delta u,Q,\delta,n) := \left(f2\Theta f(x,T,Q,\delta,n) \cdot df1dn(x,Q,n) + f1x(x,Q) \cdot df2fdnN(x,Q,\delta,n) \right)$$

$$dgE\_RTresf\_dnN(x(1),T,\Delta u,Q,\delta f(d),n) = \begin{pmatrix} 0 \\ 1.888 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$dgE\_RTresf\_dn(x(1),T,Q,\delta f(d),n) - dgE\_RTresf\_dnN(x(1),T,\Delta u,Q,\delta f(d),n) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Calculate ln_gamma_res using dgE_resf/RT / dn

$\ln\gamma\_resf(x, T, \Delta u, Q, \delta, n) := dgE\_RTresf\_dn(x, T, Q, \delta, n) + gE\_RTresf(x, T, \Delta u, Q, \delta, n)$

$$\ln\gamma\_resf(x(1), T, \Delta u, Q, \delta f(d), n) = \begin{pmatrix} 0 \\ 1.888 \end{pmatrix}$$

Calculate ln_gamma_res numerically

$\ln\gamma\_resf1N(x, T, \Delta u, Q, \delta, n) := dgE\_RTresf\_dnN(x, T, \Delta u, Q, \delta, n) + gE\_RTresf(x, T, \Delta u, Q, \delta, n)$

$$\ln\gamma\_resf1N(x(1), T, \Delta u, Q, \delta f(d), n) = \begin{pmatrix} 0 \\ 1.888 \end{pmatrix}$$

Comparison of Analytical and Numerical derivatives

$$\ln\gamma\_resf(x(1), T, \Delta u, Q, \delta f(d), n) - \ln\gamma\_resf1N(x(1), T, \Delta u, Q, \delta f(d), n) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Combinatorial Part

$$r(x,R,n) := \sum_{i=1}^{n} x_i \cdot R_i \quad q(x,Q,n) := \sum_{i=1}^{n} x_i \cdot Q_i \quad \phi(x,R,i,n) := \frac{R_i}{r(x,R,n)} \quad \Theta(x,Q,i,n) := \frac{Q_i}{q(x,Q,n)}$$

$$gE\_RTcomb(x,T,\Delta u,R,Q,n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x,R,i,n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)}\right) \right)$$

Analytical Expression for Acticity Coefficient for Original UNIQUAC equation

$$\ln\gamma\_combo(x,T,R,\Delta u,Q,n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad ret_i \leftarrow 1. - \phi(x,R,i,n) + \ln(\phi(x,R,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)}\right) + \frac{\phi(x,R,i,n)}{\Theta(x,Q,i,n)} - 1 \right) \\ ret \end{array}$$

$$\ln\gamma\_combo(x(1),T,R,\Delta u,Q,n) = \begin{pmatrix} 0 \\ -0.141 \end{pmatrix}$$

$$So(x,Q,\Delta u,T,i,n) := \sum_{j=1}^{n} \Theta o(x,Q,j,n) \cdot \tau(j,i,\Delta u,T)$$

$$\text{ln}\gamma\_\text{reso}(x,T,\Delta u,Q,n) := \left|\begin{array}{l} \text{for } i \in 1..n \\ \quad \text{ret}_i \leftarrow Q_i \cdot \left[1. - \ln(\text{So}(x,Q,\Delta u,T,i,n)) - \sum_{k=1}^{n}\left(\dfrac{\Theta o(x,Q,k,n) \cdot \tau(i,k,\Delta u,T)}{\text{So}(x,Q,\Delta u,T,k,n)}\right)\right] \\ \text{ret} \end{array}\right.$$

$$\text{ln}\gamma(x,T,\Delta u,R,Q,n) := \left|\begin{array}{l} \text{for } i \in 1..n \\ \quad \text{ret}_i \leftarrow 1. - \phi(x,R,i,n) + \ln(\phi(x,R,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\dfrac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)}\right) + \dfrac{\phi(x,R,i,n)}{\Theta(x,Q,i,n)} - 1\right) \dots \\ \qquad + Q_i \cdot \left[1. - \ln(\text{So}(x,Q,\Delta u,T,i,n)) - \sum_{k=1}^{n}\left(\dfrac{\Theta(x,Q,k,n) \cdot x_k \cdot \tau(i,k,\Delta u,T)}{\text{So}(x,Q,\Delta u,T,k,n)}\right)\right] \\ \text{ret} \end{array}\right.$$

$$\text{ln}\gamma(x(1),T,\Delta u,R,Q,n) = \begin{pmatrix} 0 \\ 1.086 \end{pmatrix}$$

$$\text{ln}\gamma\_\text{o}(x,T,R,\Delta u,Q,n) := \text{ln}\gamma\_\text{combo}(x,T,R,\Delta u,Q,n) + \text{ln}\gamma\_\text{reso}(x,T,\Delta u,Q,n)$$

$$\text{ln}\gamma\_\text{o}(x(1),T,R,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 1.086 \end{pmatrix}$$

$$\exp(\text{ln}\gamma\_\text{o}(x(1),T,R,\Delta u,Q,n)) = \begin{pmatrix} 1 \\ 2.962 \end{pmatrix}$$

Original UNIQUAC activity coefficient

Original UNIQUAC

$$\ln\gamma\_reso(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

Analytical (UNIQUAC rewritten)

$$\ln\gamma\_res1(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

Numerical (UNIQUAC Rewritten)

$$\ln\gamma\_res1N(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 1.227 \end{pmatrix}$$

Comparison to Original UNIQUAC Model

$$\ln\gamma\_reso(x(1),T,\Delta u,Q,n) - \ln\gamma\_res1(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Reformulated UNIQUAC (Analytical) compared to original UNIQUAC

$$\ln\gamma\_reso(x(1),T,\Delta u,Q,n) - \ln\gamma\_res1N(x(1),T,\Delta u,Q,n) = \begin{pmatrix} 0 \\ 3.539 \times 10^{-9} \end{pmatrix}$$

Reformulated UNIQUAC (Numerical) compared to original UNIQUAC

FlexQUAC-Q

$$\ln\gamma f(x,T,R,\Delta u,Q,\delta,n) := \ln\gamma\_combo(x,T,R,\Delta u,Q,n) + \ln\gamma\_resf(x,T,\Delta u,Q,\delta,n)$$

$$\ln\gamma f(x(1),T,R,\Delta u,Q,\delta f(d),n) = \begin{pmatrix} 0 \\ 1.747 \end{pmatrix}$$

$$\exp(\ln\gamma f(x(1),T,R,\Delta u,Q,\delta f(d),n)) = \begin{pmatrix} 1 \\ 5.735 \end{pmatrix}$$ **FlexQUAC-Q activity coefficients**

Appendix A

A3 - Fortran Implementation

Test Using Numerical Example

Components: 1 - acetone   acetone := 1 $n := 3$

               2 - chloroform   chloroform := 2 $i := 1..n$

               3 - methanol   methanol := 3 $j := 1..n$

Pure Component Parameters:

general $eps1 \equiv 10^{-8}$

$$R := \begin{pmatrix} 2.5735 \\ 2.87 \\ 1.4311 \end{pmatrix} \qquad Q := \begin{pmatrix} 2.336 \\ 2.41 \\ 1.432 \end{pmatrix}$$

FlexQUAC-Q Parameters:

$$\Delta u := \frac{\begin{pmatrix} 0 & -340.663 & 431.709 \\ 169.453 & 0 & 1307.65 \\ -104.734 & -273.252 & 0 \end{pmatrix} K}{1.98721} \qquad \delta := \begin{pmatrix} 0 & 0.1 & 0.2 \\ 0.1 & 0 & 0.3 \\ 0.2 & 0.3 & 0 \end{pmatrix} \qquad T := 333K$$

Mole fractions:

$$x := \begin{pmatrix} .3 \\ .3 \\ .4 \end{pmatrix}$$

Residual Part

FLEXQUAC - Q

The complex terms of the original are simplified into more convenient terms to enable efficient programming in the FORTRAN subroutine. The simplified terms (shown in yellow) are compared to the original derivation in the previous file (shown in green).

$$\tau_{i,j} := \exp\left(\frac{-\Delta u_{i,j}}{T}\right)$$

$$\tau = \begin{pmatrix} 1 & 1.673 & 0.521 \\ 0.774 & 1 & 0.139 \\ 1.171 & 1.511 & 1 \end{pmatrix}$$

$$\text{thetfx}_i := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)$$

$$\text{sthetfx} := \sum_i x_i \cdot \text{thetfx}_i$$

$$\text{sthetfx} = 2.507 \qquad \boxed{2.507}$$

$$\text{thetfx}_i := \frac{\text{thetfx}_i}{\text{sthetfx}}$$

$$\text{thetfx} = \begin{pmatrix} 1.106 \\ 1.194 \\ 0.775 \end{pmatrix}$$

$$Sf_i := \sum_{j=1}^{n} x_j \cdot \text{thetfx}_j \cdot \tau_{j,i}$$

$$Sf = \begin{pmatrix} 0.972 \\ 1.382 \\ 0.532 \end{pmatrix}$$

G^E residual

$$f1 := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f1 = -1.997 \qquad \boxed{-1.997}$$

$$f2 := \sum_{i=1}^{n} x_i \cdot \text{thetfx}_i \cdot \ln(Sf_i)$$

$$f2 = -0.089 \qquad \boxed{-0.089}$$

$$gE\_RTresf := f1 \cdot f2$$

$$gE\_RTresf = 0.177 \qquad \boxed{0.177}$$

Auxillary Variables

$$\text{sum1}_j := \sum_{i=1}^{n} \frac{x_i \cdot \text{thetfx}_i \cdot \tau_{j,i}}{Sf_i}$$

$$\text{df2}_i := \text{sum1}_i + \ln(Sf_i)$$

$$\text{df2} = \begin{pmatrix} 1.05 \\ 0.928 \\ 0.744 \end{pmatrix} \qquad \begin{pmatrix} 1.05 \\ 0.928 \\ 0.744 \end{pmatrix}$$

$$SJ_i := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$SJ = \begin{pmatrix} 2.433 \\ 2.58 \\ 2.508 \end{pmatrix} \qquad \begin{pmatrix} 2.433 \\ 2.58 \\ 2.508 \end{pmatrix}$$

Jacobian Matrix

$$
\text{JacobianFQ1} :=
\begin{array}{|l}
\text{for } i \in 1..n \\
\quad \text{for } m \in 1..n \\
\qquad \begin{array}{|l}
\text{Jac}_{m,i} \leftarrow -\left(\text{thetfx}_m \cdot x_i\right) - \dfrac{SJ_m \cdot x_i}{\text{sthetfx}} + \dfrac{Q_i \cdot x_i}{\text{thetfx}_i \cdot \text{sthetfx}} \\[2mm]
\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{\text{thetfx}_i \cdot \text{sthetfx}} \quad \text{if } i \neq m \\[2mm]
\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + 1 \quad \text{if } i = m \\[2mm]
\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} \cdot \text{thetfx}_i
\end{array} \\
\text{Jac}
\end{array}
$$

$$\text{JacobianFQ1} = \begin{pmatrix} 0.697 & -0.388 & -0.309 \\ -0.391 & 0.686 & -0.296 \\ -0.229 & -0.224 & 0.453 \end{pmatrix} \qquad \begin{pmatrix} 0.697 & -0.388 & -0.309 \\ -0.391 & 0.686 & -0.296 \\ -0.229 & -0.224 & 0.453 \end{pmatrix}$$

Derivatives

df2/dn

df2dn := JacobianFQ1 · df2

$$df2dn = \begin{pmatrix} 0.142 \\ 6.504 \times 10^{-3} \\ -0.111 \end{pmatrix}$$

$$\begin{pmatrix} 0.142 \\ 6.504 \times 10^{-3} \\ -0.111 \end{pmatrix}$$

d(G^E/RT)/dn

dgE_RTresf := f1 · df2dn + f2 · (−Q − f1)

$$dgE\_RTresf = \begin{pmatrix} -0.254 \\ 0.024 \\ 0.172 \end{pmatrix}$$

$$\begin{pmatrix} -0.254 \\ 0.024 \\ 0.172 \end{pmatrix}$$

ln$\gamma$$_{residual}$

lnγ_resf := dgE_RTresf + gE_RTresf

$$ln\gamma\_resf = \begin{pmatrix} -0.076 \\ 0.201 \\ 0.35 \end{pmatrix}$$

$$\begin{pmatrix} -0.076 \\ 0.201 \\ 0.35 \end{pmatrix}$$

APPENDIX B

B1. Implementation in Fortran

1.1 FlexQUAC-Q Code:

This is the fortran subroutine code to calculate the activity coefficient using the FlexQUAC-Q model.

```
C---------------------------------------------------------------------

      integer*2 function iactqflex (x,T,act,he,cpe,pmh,pmc)

C---------------------------------------------------------------------

c     FlexQUAC-Q flexibilizes the surface fraction in the residual part only

c

C---------------------------------------------------------------------
```

This section contains all the pertinent variable declarations		
	use	peqmodule
	use	dflib
	use	dflogm
	implicit none	
	logical*4	lmodcomb34, lmodcomb23, lmodqflex, lmodflexnum
	common	/flexquac/ lmodcomb34, lmodcomb23, lmodqflex, lmodflexnum
c	character	cbuf*80
c	integer(4)	retlog
	character	filescrp*255
	real*8	x(MKO),t,act(MKO),he,cpe,pmh(MKO),pmc(MKO)
	real*8	aij(mko,mko),
>		aijt(mko,mko),
>		alphaij(mko,mko),
>		TAU(mko,mko),ftau,
>		sxr,sxq,sxqp,ss(mko),ss1(mko),dgE_dn(mko),f1,f2,
>		sftji,sftu,
>		F(mko),Fp(mko),
>		V(mko),
>		x1(mko),

```
>           rg,

>           xsum,

>           xsave,

>           her,

>           dQdx(mko), dQdn(mko), Jac(mko,mko), Q,sum, sumd(mko), c1,
            lgamc(mko),

>           B(mko),

>           C,

>           Sf(mko), thetafx(mko), df1dn(mko), df2dn(mko),lgamr(mko),

>           df2dthet(mko),

>           SJ(mko),

>           sthetfx,

>           sumk_array(mko),

>           thetfx(mko),

>           thethaf(mko), sumk,ge



       integer    i,j,ia,i1,ki,kj, ierr, ilog
```

This section creates a text file for computation output to allow for comparison between FORTRAN and Mathcad results

```
       ilog = 1
       if (ilog.eq.1) then
               open (186, file = filescrp('flexq.txt'), access='append')
               write (186,*) ' start subroutine ---------------------------'
               write (186,*) ' x = ', (x(i),i=1,peq%ncomp)
               write (186,*) ' T = ', T
       endif
```

Section 1: Calculate interaction parameters for given temperature and their temperature derivative and outputs the values to the text file

---

```
RG=1.98721d0

Call setpar (t,aij,aijt,alphaij)


            if (ilog.eq.1) then
                    write (186,*) ' aij'
                    do j=1, peq%ncomp
                            write (186,*)  (aij(i,j),i=1,peq%ncomp)
                    enddo
                    write (186,*) ' aijt'
                    do j=1, peq%ncomp
                            write (186,*)  (aijt(i,j),i=1,peq%ncomp)
                    enddo
                    write (186,*) ' delta'
                    do j=1, peq%ncomp
                            write (186,*)  (alphaij(i,j),i=1,peq%ncomp)
                    enddo
            endif
```

Section 2: Calculate combinatorial part (UNIQUAC, unchanged or UNIQUAC-3/4)

```
SXQ     = 0.
SXQP    = 0.
SXR     = 0.
DO KJ = 1,peq%NCOMP
        SXQ = SXQ + X(KJ) * peq%uniq(KJ)
        SXR = SXR + X(KJ) * peq%unir(KJ)
    enddo
DO KI = 1,peq%NCOMP
        F(KI)   = peq%uniq(KI) / SXQ
        V(KI)   = peq%unir(KI) / SXR
        C1              = -1.d0 + V(KI)/F(KI) + LOG(F(KI)/V(KI))
```

The user determines the choice of combinatorial term

```
                    if (lmodcomb34) then
                              LGAMC(KI) = 1.d0 - V(KI) ** .75d0 + .75d0 * LOG(V(KI)) - 5.d0
peq%uniq(KI) * C1
                    else if (lmodcomb23) then
                              LGAMC(KI) = 1.d0 - V(KI) ** .66666666d0 + .66666666d0 *
LOG(V(KI)) - 5.d0 * peq%uniq(KI) * C1
                    else
                              LGAMC(KI) = 1.d0 - V(KI) + LOG(V(KI)) + 5.d0 * peq%uniq(KI) *
C1
                    endif
              enddo


                    if (ilog.eq.1) then
                              write (186,*) ' lgamc'
                              write (186,*)  (lgamc(i),i=1,peq%ncomp)
                    endif
```

Section 3: Calculation of variables and output the values to the text file

```
c       Calculate τ and outputs the value to the text file
    do i = 1,peq%NCOMP
              do j = 1,peq%NCOMP
                    tau(i,j) = dexp(dble(-aij(i,j)/(RG*T)))
              enddo
        enddo


              if (ilog.eq.1) then
                    write (186,*) ' tau'
                    do j=1, peq%ncomp
                              write (186,*)  (tau(i,j),i=1,peq%ncomp)
                    enddo
              endif
```

```
      if (abs(x(1)-0.03080).lt.1.e-5) then
              continue
      endif
```

The following variables (sthetfx & thetfx) are defined and calculated as per the
Mathcad file. The output is sent to the text file

```
      sthetfx = 0.
      do i = 1, peq%ncomp
   sumk_array(i) = 1.
   do j = 1, peq%ncomp
      sumk_array(i) = sumk_array(i) + x(j) * peq%uniq(j) * alphaij(i, j)
   enddo
   thetfx(i) = peq%uniq(i) * sumk_array(i)
   sthetfx = sthetfx + (x(i) * thetfx(i))
      enddo


              if (ilog.eq.1) then
                      write (186,*) ' sthetfx'
                      write (186,*)  sthetfx
                      write (186,*) ' thetfx'
                      write (186,*)  (thetfx(i),i=1,peq%ncomp)
              endif


c     Computation of Sf and computation of surface fraction


   do i = 1, peq%ncomp
              thethaf(i) = thetfx(i) / sthetfx
   enddo


              if (ilog.eq.1) then
                      write (186,*) ' thethaf'
                      write (186,*)  (thethaf(i),i=1,peq%ncomp)
```

```
                    endif



do i = 1, peq%ncomp
          Sf(i) = 0
   do j = 1, peq%ncomp
                    Sf(i) = Sf(i) + x(j) * tau(j, i) * thethaf(j)
   enddo
enddo



                    if (ilog.eq.1) then
                            write (186,*) ' Sf'
                            write (186,*)  (Sf(i),i=1,peq%ncomp)
                    endif
```

Section 4: Computation of f1, f2 and $G^E_{residual}$ and outputs the values to the text file

```
c    Computation of f1 and f2
     sumk = 0.
     Do i = 1, peq%ncomp
        sumk = sumk + x(i) * peq%uniq(i)
     enddo
                    if (ilog.eq.1) then
                            write (186,*) ' sumk'
                            write (186,*)  sumk
                    endif


         f1 = -sumk
     sumk = 0.
     do i = 1, peq%ncomp
         sumk = sumk + x(i) * thethaf(i) * Log(Sf(i))
     enddo
     f2 = sumk
```

```
                    if (ilog.eq.1) then
                            write (186,*) ' f2'
                            write (186,*)  f2
                    endif


c    Computation of G^E_residual
     Ge = f1 * f2


                    if (ilog.eq.1) then
                            write (186,*) ' Ge'
                            write (186,*)  Ge
                    endif
```

Section 5 – Compute derivative of f1 and record output in text file

```
    Do i = 1, peq%ncomp
            df1dn(i) = -peq%uniq(i) - f1
    enddo


                    if (ilog.eq.1) then
                            write (186,*) ' df1dn'
                            write (186,*)  (df1dn(i),i=1,peq%ncomp)
                    endif
```

Section 6: Derivative of f2 wrt to surface fraction and output of value to text file

```
      Do i = 1, peq%ncomp
            sumk_array(i) = 0.
```

```
        do j = 1, peq%ncomp
                sumk_array(i) = sumk_array(i) + (x(j) * tau(i, j) * thethaf(j)) / Sf(j)
        enddo
        df2dthet(i) = sumk_array(i) + Log(Sf(i))
enddo


        if (ilog.eq.1) then
                write (186,*) ' df2dthet'
                write (186,*)  (df2dthet(i),i=1,peq%ncomp)
        endif
```

Section 7: Computation of Jacobian Matrix

```
Do i = 1, peq%ncomp
        SJ(i) = 0.
        Do j = 1, peq%ncomp
                SJ(i) = SJ(i) + x(j) * peq%uniq(j) * (1 + peq%uniq(i) * alphaij(j, i))
        enddo
enddo
        if (ilog.eq.1) then
                write (186,*) ' SJ'
                write (186,*)  (SJ(i),i=1,peq%ncomp)
        endif

do i = 1, peq%ncomp
        do j = 1, peq%ncomp
                Jac(j, i) = (peq%uniq(i) * x(i)) / (thethaf(i) * sthetfx)
>                                  + (-(thethaf(j) * x(i)) - ((SJ(j) * x(i)) / sthetfx))
                If (i .eq. j) Jac(j, i) = (Jac(j, i) + 1)
                If (i .ne. j) Jac(j, i) = Jac(j, i) + (x(i) * peq%uniq(i)
>                                  * peq%uniq(j) * alphaij(i, j)) / (thethaf(i) * sthetfx)
                Jac(j, i) = Jac(j, i) * thethaf(i)
```

```
              enddo
      enddo


          if (ilog.eq.1) then
                  write (186,*) ' Jac'
                  do j=1, peq%ncomp
                          write (186,*)  (Jac(i,j),i=1,peq%ncomp)
                  enddo
          endif
```

Section 8: Compute derivative of f2 wrt to mole numbers

```
do i = 1, peq%ncomp
          df2dn(i) = 0.
          do j = 1, peq%ncomp
                  df2dn(i) = df2dn(i) + Jac(i, j) * df2dthet(j)
          enddo
enddo
          if (ilog.eq.1) then
                  write (186,*) ' df2dn'
                  write (186,*)  (df2dn(i),i=1,peq%ncomp)
          endif
```

Section 9: Computation of $G^E_{residual}$ and residual activity coefficients

```
      sumk = 0
      do i = 1,peq%ncomp
              lgamr(i) =  f1 * df2dn(i) + f2 * df1dn(i) + Ge
              act(i) = EXP(lgamr(i) + lgamc(i))
      enddo


          if (ilog.eq.1) then
                  write (186,*) ' lgamr'
```

```
                                    write (186,*)  (lgamr(i),i=1,peq%ncomp)

                                    write (186,*) ' act'

                                    write (186,*)  (act(i),i=1,peq%ncomp)

                    endif

        iactqflex=0

        if (ilog.eq.1) then

                    close (186)

        endif


        return

        end
```

1.2 Model Selection Code

The code of this subroutine allows the user to select between the FlexQUAC and FlexQUAC-Q model.

```
C-----------------------------------------------------------------

        SUBROUTINE matflex (IERROR)

C-----------------------------------------------------------------
```

C   model initialisation routine for FlexQUAC, QFlexQUAC

C---

```
        Variable dec
        use             user32
        use             kernel32
        use             dflogm
        use             dfcom
        use             dfauto
        use             peqmodule
     IMPLICIT none
        logical*4       lmodcomb34, lmodcomb23, lmodqflex,lmodflexnum
        common          /flexquac/ lmodcomb34, lmodcomb23,
lmodqflex,lmodflexnum
        integer         ierror, index
        integer*4       ret, ghwndMain
        ghwndMain = 0


        peq%iusedcp = 1              ! use delta_cP for SLE calculation
```

```
c       get model control options from control-string


c       lmodcomb34 to use the 3/4-modification of the combinatorial part (as in
mod. UNIFAC)
        lmodcomb34 = .false.
        if (index(peq%control,'modcomb34').ne.0) lmodcomb34 = .true.
c       lmodcomb23 to use the 2/3-modification of the combinatorial part (Kikic)
        lmodcomb23 = .false.
        if (index(peq%control,'modcomb23').ne.0) lmodcomb23 = .true.


c       only one of modcomb23 and modcomb34 allowed
        if (lmodcomb23.and.lmodcomb34) then
```

```
           ret = MESSAGEBOX( ghwndMain, 'only one of modcomb23 and
modcomb34 allowed. modcomb34 selected.'c,
    >    'Warning'c, MB_OK)
           lmodcomb23 = .false.
       endif
```

```
c      lmodqflex to use the surface fraction flexibilisation only in the
combinatorial part
       lmodqflex = .false.
       if (index(peq%control,'modqflex').ne.0) lmodqflex = .true.


c      numerical flexibilisation for testing purpose
       lmodflexnum = .false.
       if (index(peq%control,'numflex').ne.0) lmodflexnum = .true.


    return
    end
```

B2. Implementation in Excel

2.1 Excel Code

```
Sub Main()


Variable Declarations
Dim inkr(8) As Long, inkrmod(8) As Long, xxx As Single, stringscr As String * 2
```

```
Dim x(1 To 5) As Double, y(5) As Double, he As Double, cpe As Double, dq(5) As
Double, grad_x(5, 5) As Double, grad_fx(5, 5) As Double, dQdn(5) As Double
Dim act(1 To 5) As Double, pmh(1 To 5) As Double, pmc(1 To 5) As Double
Dim single_set As Integer, irowstart As Integer, datafile As String, irow As Integer, i As
Integer, j As Integer
Dim imiss As Integer, icod As Integer, iso As Integer, ibest As Integer
Dim ename As String, dname As String, empform As String, casn As String
Dim runi As Single, quni As Single, q As Double
Dim antA As Single, antB As Single, antC As Single, iberL As Integer, iberH As Integer
Dim anthA As Single, anthB As Single, anthC As Single, iberhL As Integer, iberhH As
Integer
Dim ilast As Integer, xn As Double, t As Double, p As Double
Dim iret As Long
```

```
Section 1
Call DDB_2.D_ReadDDBenv
Worksheets("Data_Sheet").Select
Call prepare_d_sheet  ' Formats the excel spreadsheet

single_set = 0
irowstart = 3
imodel = 2
fit_tertest = True
rework_fq = True      ' this goes through the results and refitts the cases where FQ
is worse than F
workpath = "C:\Thishen\Masters Work\flexquac Q projects\EXCEL"
'datafile = "VLE_binary.TXT"
'datafile = "VLE_all1.TXT"
If fit_tertest Then
   datafile = "VLE_ternary_test.TXT"
End If
nline_tertest = 2
```

```
Dim wst3 As Object
Set wst3 = Worksheets("Tabelle3")


If rework_fq Then
    Dim set2line(30000)
        For jlinerw = 4 To 6000
            buf = wst3.Cells(jlinerw, 1)
            If IsNumeric(buf) And buf <> "" Then
                set2line(buf) = jlinerw
            End If
        Next jlinerw
End If
```

```
Section 2
Open workpath & datafile For Input As #30
irow = 2
While Not EOF(30)
1:
Worksheets("Data_Sheet").Cells(23, 2) = 0#   ' set d to 0
If fit_tertest = True Then
    nset_tertest = Worksheets("tertest").Cells(nline_tertest, 6)
    Worksheets("Data_Sheet").Cells(2, 12) = nset_tertest
End If
' read dataset from file
DoEvents
With vleset
    Input #30, .ncomp, .iso, .nval, .PT, .nref, .icod(1), .icod(2), .icod(3), .icod(4), .iset, i, i,
.iddb
    For i = 1 To .nval
        For j = 1 To .ncomp - 1
            Input #30, .x(j, i)
```

```
        Next j
        For j = 1 To .ncomp - 1
            Input #30, .y(j, i)
        Next j
        Select Case .iso
        Case 1, 3, 5, 7
            Input #30, .p(i)
            .t(i) = .PT - 273.15
        Case 2, 4, 6, 8
            Input #30, .t(i)
            .p(i) = .PT
        Case 9
            Input #30, .p(i), .t(i)
        End Select
    Next i
End With


If vleset.ncomp <> 2 And fit_tertest = False Then
    GoTo 1
End If


' use dataset?
Worksheets("Data_Sheet").Cells(2, 10) = vleset.iset
If fit_tertest <> True Then
    If vleset.nval < 10 Then GoTo 1
    If vleset.iset = 5049 Then GoTo 1 ' this dataset has one pure component vapor
pressure = 0
    If vleset.iset = 23067 Then GoTo 1 ' this dataset has one pure component vapor
pressure = 0
    If vleset.iso <> 3 Then GoTo 1
    If rework_fq Then
        If set2line(vleset.iset) < 1 Then GoTo 1
```

```
      'If set2line(vleset.iset) < 310 Then GoTo 1
    ' If wst3.Cells(set2line(vleset.iset), 23) < 1.05 Then GoTo 1


    End If
Else
   If vleset.iset <> nset_tertest Then
      GoTo 1
   Else
      iii = iii + 1
   End If
End If
irow = irow + 1
If rework_fq Then
   irow = set2line(vleset.iset)
End If


If irow < irowstart Then GoTo 1


'write dataset to sheet


Range("A26:P200").Select
Selection.ClearContents
Range("A1").Select
With Worksheets("Data_Sheet")
.Cells(4, 3) = vleset.iset
.Cells(5, 3) = vleset.iso
.Cells(6, 3) = vleset.nval
.Cells(7, 3) = vleset.PT
Select Case vleset.iso
Case 1, 3, 5, 7
   .Cells(7, 4) = "Deg C"
Case 2, 4, 6, 8
```

```
   .Cells(7, 4) = "mm Hg"
Case 9
   .Cells(7, 3) = ""
End Select
.Cells(8, 3) = vleset.nref
.Cells(10, 3) = vleset.icod(1)
.Cells(11, 3) = vleset.icod(2)
.Cells(4, 3) = vleset.iset
.Cells(4, 3) = vleset.iset
For i = 1 To vleset.nval
   .Cells(25 + i, 1) = vleset.x(1, i)
   .Cells(25 + i, 2) = vleset.y(1, i)
   .Cells(25 + i, 3) = vleset.p(i)
   .Cells(25 + i, 4) = vleset.t(i)
Next i
' MsgBox "continue?"


'retrieve additional information


imiss = 0
For i = 1 To vleset.ncomp
   icod = vleset.icod(i)
   Call DDB_Stoff_name(icod, ename, dname, empform, casn)
   .Cells(9 + i, 4) = empform
   .Cells(9 + i, 5) = ename
   Call DDB_Stoff_rqinkr(icod, runi, quni, inkr, inkrmod)
   .Cells(14 + i, 2) = runi
   If runi < 0.01 Then imiss = 1
   vleset.runi(i) = runi
   .Cells(14 + i, 3) = quni
   vleset.quni(i) = quni
   If quni < 0.01 Then imiss = 1
```

```
    Call DDB_Stoff_antoine(icod, antA, antB, antC, iberL, iberH, anthA, anthB, anthC,
iberhL, iberhH)
    If antA < 0.01 Then imiss = 1
    .Cells(14 + i, 4) = antA
    .Cells(14 + i, 5) = antB
    .Cells(14 + i, 6) = antC
    vleset.Ant(1, i) = antA
    vleset.Ant(2, i) = antB
    vleset.Ant(3, i) = antC
    .Cells(14 + i, 7) = iberL
    .Cells(14 + i, 8) = iberH
Next i
```

```
Section 3 – Select liquid model
Call peq.settype(0, itype_lmodel, ilmodel_FlexQUAC)
Call peq.settype(0, itype_ncomp, CLng(vleset.ncomp))
For i = 1 To vleset.ncomp
    Call peq.settype(i, itype_icod, CLng(vleset.icod(i)))
Next i


For i = 1 To vleset.ncomp
    Call peq.settype(i, itype_vap, 1) ' 1 - Antoine ( low pressure)
    Call peq.settype(i, itype_den, 1) ' 1 - no Poynting correction
    Call peq.settype(i, itype_adj, 1) ' adjust
Next i


Call peq.init_pure(0)    ' read pure component properties from STOFF1 etc.
Call peq.init_lmodel     ' initialize liquid model
```

```
Section 4
Adjust pure component vapor pressure

```

```
For j = 1 To vleset.ncomp - 1
   vleset.padj(j) = 1
   For i = 1 To vleset.nval
      If vleset.x(j, i) > 0.9999999 Then
         vleset.padj(j) = vleset.p(i) / (10 ^ (vleset.Ant(1, j) - vleset.Ant(2, j) / (vleset.Ant(3,
j) + vleset.t(i))))
         If vleset.padj(j) < 0.1 Then vleset.padj(j) = 1
      End If
   Next i
Next j
ilast = vleset.ncomp
vleset.padj(ilast) = 1
For i = 1 To vleset.nval
   xn = 1#
   For j = 1 To ilast - 1
      xn = xn - vleset.x(j, i)
   Next j
   If xn > 0.99999999 Then
      vleset.padj(ilast) = vleset.p(i) / (10 ^ (vleset.Ant(1, ilast) - vleset.Ant(2, ilast) /
(vleset.Ant(3, ilast) + vleset.t(i))))
      If vleset.padj(ilast) < 0.1 Then vleset.padj(ilast) = 1
   End If
Next i
For j = 1 To ilast
   .Cells(14 + j, 9) = vleset.padj(j)
   Call peq.setpar(j, 1, 1, 3, Log(vleset.padj(j)) / Log(10#))
Next j


Calculate pure component vapor pressures


If vleset.iso <> 6 And vleset.iso <> 9 Then
   For i = 1 To vleset.nval
```

```
    t = .Cells(25 + i, 4)
    For j = 1 To 2
        p = vleset.padj(j) * 10 ^ (.Cells(14 + j, 4) - .Cells(14 + j, 5) / (.Cells(14 + j, 6) + t))
        .Cells(25 + i, 4 + j) = p
        vleset.ps(j, i) = p
    Next j
    Next i
End If
```

```
Section 5 - Calculate activity coefficients


If vleset.iso < 3 Then
    For i = 1 To vleset.nval
        p = .Cells(25 + i, 3)
        If .Cells(25 + i, 1) > 0.0000001 And .Cells(25 + i, 1) < 0.9999999 Then
            .Cells(25 + i, 7) = .Cells(25 + i, 2) * p / .Cells(25 + i, 1) / .Cells(25 + i, 5)
            .Cells(25 + i, 8) = (1# - .Cells(25 + i, 2)) * p / (1# - .Cells(25 + i, 1)) / .Cells(25 + i,
6)
            vleset.act(1, i) = .Cells(25 + i, 7)
            vleset.act(2, i) = .Cells(25 + i, 8)
        End If
    Next i
End If
Section 6 - Calculate partial pressures


If vleset.iso < 3 Or iso = 6 Then
    For i = 1 To vleset.nval
        p = .Cells(25 + i, 3)
        .Cells(25 + i, 9) = .Cells(25 + i, 2) * p
        .Cells(25 + i, 10) = (1# - .Cells(25 + i, 2)) * p
    Next i
End If
```

Section 7 - write formulas for P and y to grid

For i = 1 To vleset.nval

   .Cells(25 + i, 13) = "=RC[-12]*RC[-2]*RC[-8] +(1.-RC[-12])*RC[-1]*RC[-7]"

   .Cells(25 + i, 14) = "=RC[-13]*RC[-3]*RC[-9]/RC[-1]"

   .Cells(25 + i, 15) = "=((RC[-12]-RC[-2])/RC[-12])^2"

Next i

.Cells(21, 6) = "=SUM(R[5]C[9]:R[" & CStr(24 + vleset.nval) & "]C[9])"

Section 8

Application.Calculation = xlManual

Set initial parameters for original model

ibest = 1

.Cells(20, 2) = 50#

.Cells(21, 2) = 60#

.Cells(22, 2) = 0.3

.Cells(23, 2) = 0#

.Cells(23, 3) = 0#

Worksheets("Tabelle3").Select

ilog = Application.Wait(Now() + TimeValue("0:00:03"))

Worksheets("Data_Sheet").Select

If rework_fq Then

   GoTo second_model

End If

Call peq.settype(0, itype_Imodel, ilmodel_UNIQUAC)

Call peq.setstring(1, "")

Section 9 – Regression Procedure

```
Call datafit
Call results2sheet
res(1, 1) = .Cells(21, 6)
res(1, 2) = .Cells(20, 2)
res(1, 3) = .Cells(21, 2)
res(1, 4) = .Cells(22, 2)
res(1, 5) = .Cells(23, 2)


.Cells(20, 2) = -1# * res(1, 2)
If (.Cells(20, 2) > 5000#) Then .Cells(20, 2) = .Cells(20, 2) / 10#
If (.Cells(20, 3) > 5000#) Then .Cells(21, 2) = .Cells(21, 2) / 10#
Call datafit
Call results2sheet
res(2, 1) = .Cells(21, 6)
res(2, 2) = .Cells(20, 2)
res(2, 3) = .Cells(21, 2)
res(2, 4) = .Cells(22, 2)
res(2, 5) = .Cells(23, 2)
If res(2, 1) < res(ibest, 1) Then ibest = 2


.Cells(20, 2) = res(1, 2)
.Cells(21, 2) = -1# * res(1, 3)
If (.Cells(20, 2) > 5000#) Then .Cells(20, 2) = .Cells(20, 2) / 10#
If (.Cells(20, 3) > 5000#) Then .Cells(21, 2) = .Cells(21, 2) / 10#
.Cells(22, 2) = res(1, 4)
.Cells(23, 2) = res(1, 5)
Call datafit
Call results2sheet
res(3, 1) = .Cells(21, 6)
res(3, 2) = .Cells(20, 2)
res(3, 3) = .Cells(21, 2)
```

```
res(3, 4) = .Cells(22, 2)

res(3, 5) = .Cells(23, 2)

If res(3, 1) < res(ibest, 1) Then ibest = 3


.Cells(20, 2) = -1# * res(1, 2)

.Cells(21, 2) = -1# * res(1, 3)

If (.Cells(20, 2) > 5000#) Then .Cells(20, 2) = .Cells(20, 2) / 10#

If (.Cells(20, 3) > 5000#) Then .Cells(21, 2) = .Cells(21, 2) / 10#

.Cells(22, 2) = res(1, 4)

.Cells(23, 2) = res(1, 5)

Call datafit

Call results2sheet

res(4, 1) = .Cells(21, 6)

res(4, 2) = .Cells(20, 2)

res(4, 3) = .Cells(21, 2)

res(4, 4) = .Cells(22, 2)

res(4, 5) = .Cells(23, 2)

If res(4, 1) < res(ibest, 1) Then ibest = 4


.Cells(20, 2) = res(ibest, 2)

.Cells(21, 2) = res(ibest, 3)

.Cells(22, 2) = res(ibest, 4)

.Cells(23, 2) = res(ibest, 5)

Call datafit

Call results2sheet


Calculate

Application.ScreenUpdating = False

End With
```

Section 10 - Calculate gamma infinite

```
x(1) = 0#
x(2) = 1#
t = vleset.t(1) + 273.15
On Error Resume Next
Call peq.actcal(x, t, act, he, cpe, pmh, pmc, iret)
On Error GoTo 0
Sheets("Data_Sheet").Cells(20, 9) = act(1)
x(1) = 1#
x(2) = 0#
t = vleset.t(1) + 273.15
On Error Resume Next
Call peq.actcal(x, t, act, he, cpe, pmh, pmc, iret)
On Error GoTo 0
Sheets("Data_Sheet").Cells(21, 9) = act(2)
Sheets("Data_Sheet").Select


' Calculate Relative absolute deviation in pressure
pdev = 0#
For i = 1 To vleset.nval
    p = Sheets("Data_Sheet").Cells(25 + i, 3)
    pcal = Sheets("Data_Sheet").Cells(25 + i, 13)
    pdev = pdev + Abs((p - pcal) / p)
Next i
pdev = pdev / vleset.nval * 100#
Sheets("Data_Sheet").Cells(3, 10) = pdev


' Copy results to Excel spreadsheet
Call copy_cell("C4", "a" & CStr(irow + 1))
Call copy_cell("C5", "b" & CStr(irow + 1))
Call copy_cell("C6", "c" & CStr(irow + 1))
Call copy_cell("C7", "d" & CStr(irow + 1))
Call copy_cell("C8", "e" & CStr(irow + 1))
```

```
Call copy_cell("C10", "f" & CStr(irow + 1))

Call copy_cell("C11", "g" & CStr(irow + 1))

Call copy_cell("B20", "h" & CStr(irow + 1))

Call copy_cell("B21", "i" & CStr(irow + 1))

Call copy_cell("B22", "j" & CStr(irow + 1))

Call copy_cell("F21", "k" & CStr(irow + 1))

Call copy_cell("I20", "z" & CStr(irow + 1))

Call copy_cell("I21", "aa" & CStr(irow + 1))

Call copy_cell("J3", "ab" & CStr(irow + 1))


If single_set = 1 Then

    Sheets("Data_Sheet").Select

    Sheets("Data_Sheet").Copy Before:=Sheets(1)

    Sheets("Data_Sheet (2)").Select

    Sheets("Data_Sheet (2)").Name = "UNIQUAC"

    Sheets("Data_Sheet").Select

End If


Application.ScreenUpdating = True

DoEvents
```

```
Section 11: Second regression using FlexQUAC-Q model


Call peq.settype(0, itype_lmodel, ilmodel_FlexQUAC)

'Call peq.setstring(1, "modqflex modcomb34")

Call peq.setstring(1, "modqflex")

'Call peq.setstring(1, "")


With Sheets("Data_Sheet")

If Not rework_fq Then

    .Cells(20, 2) = .Cells(20, 2) * 0.9
```

```
.Cells(21, 2) = .Cells(21, 2) * 1.05

.Cells(23, 2) = 0.1

.Cells(23, 3) = 0.05

If .Cells(22, 3) > 0.599 Then .Cells(22, 2) = 0.3
```

Section 12 – FlexQUAC-Q data regression

```
ibest = 1

Call datafit

Call results2sheet

.Cells(23, 3) = 0.05

Call datafit

Call results2sheet

res(1, 1) = .Cells(21, 6)

res(1, 2) = .Cells(20, 2)

res(1, 3) = .Cells(21, 2)

res(1, 4) = .Cells(22, 2)

res(1, 5) = .Cells(23, 2)


.Cells(23, 3) = 0.05

Call datafit

Call results2sheet

If IsNumeric(.Cells(21, 6)) Then

    res(2, 1) = .Cells(21, 6)

Else

    res(2, 1) = 1E+30

End If

res(2, 2) = .Cells(20, 2)

res(2, 3) = .Cells(21, 2)

res(2, 4) = .Cells(22, 2)

res(2, 5) = .Cells(23, 2)

If res(2, 1) < res(ibest, 1) Then ibest = 2
```

```
'.Cells(20, 2) = res(1, 2)
'.Cells(21, 2) = -1# * res(1, 3)
'.Cells(22, 2) = res(1, 4)
'.Cells(23, 2) = 0.1
'Call datafit
'res(3, 1) = .Cells(21, 6)
'res(3, 2) = .Cells(20, 2)
'res(3, 3) = .Cells(21, 2)
'res(3, 4) = .Cells(22, 2)
'.Cells(23, 2) = 0.1
'If res(3, 1) < res(ibest, 1) Then ibest = 3


'.Cells(20, 2) = -1# * res(1, 2)
'.Cells(21, 2) = -1# * res(1, 3)
'.Cells(22, 2) = res(1, 4)
'.Cells(23, 2) = 0.1
'Call datafit
'res(4, 1) = .Cells(21, 6)
'res(4, 2) = .Cells(20, 2)
'res(4, 3) = .Cells(21, 2)
'res(4, 4) = .Cells(22, 2)
'res(4, 5) = .Cells(23, 2)
'If res(4, 1) < res(ibest, 1) Then ibest = 4


.Cells(20, 2) = res(ibest, 2)
.Cells(21, 2) = res(ibest, 3)
.Cells(22, 2) = res(ibest, 4)
.Cells(23, 2) = res(ibest, 5)
Call datafit
Call results2sheet
Else
.Cells(20, 2) = wst3.Cells(set2line(vleset.iset), 12)
```

```
    .Cells(21, 2) = wst3.Cells(set2line(vleset.iset), 13)

    .Cells(22, 2) = wst3.Cells(set2line(vleset.iset), 14)

    .Cells(23, 2) = wst3.Cells(set2line(vleset.iset), 15)

    .Cells(23, 3) = wst3.Cells(set2line(vleset.iset), 16)

    Call datafit

    Call results2sheet

End If




Calculate

Application.ScreenUpdating = False

End With
```

```
Section 13

Calculate gamma infinite

peq.Log (0)

x(1) = 0#

x(2) = 1#

t = vleset.t(1) + 273.15

On Error Resume Next

Call peq.actcal(x, t, act, he, cpe, pmh, pmc, iret)

On Error GoTo 0

Sheets("Data_Sheet").Cells(20, 9) = act(1)

x(1) = 1#

x(2) = 0#

t = vleset.t(1) + 273.15

On Error Resume Next

Call peq.actcal(x, t, act, he, cpe, pmh, pmc, iret)

On Error GoTo 0

Sheets("Data_Sheet").Cells(21, 9) = act(2)

peq.Log (0)
```

Calculate relative absolute deviation in pressure

```
pdev = 0#
For i = 1 To vleset.nval
    p = Sheets("Data_Sheet").Cells(25 + i, 3)
    pcal = Sheets("Data_Sheet").Cells(25 + i, 13)
    pdev = pdev + Abs((p - pcal) / p)
Next i
pdev = pdev / vleset.nval * 100#
Sheets("Data_Sheet").Cells(3, 10) = pdev
```

Copy results to Excel spreadsheet

```
If Not rework_fq Then
    Call copy_cell("B20", "l" & CStr(irow + 1))
    Call copy_cell("B21", "m" & CStr(irow + 1))
    Call copy_cell("B22", "n" & CStr(irow + 1))
    Call copy_cell("B23", "o" & CStr(irow + 1))
    Call copy_cell("F21", "p" & CStr(irow + 1))
    Call copy_cell("I20", "ad" & CStr(irow + 1))
    Call copy_cell("I21", "ae" & CStr(irow + 1))
    Call copy_cell("J3", "af" & CStr(irow + 1))
Else
    Call copy_cell("B20", "q" & CStr(irow))
    Call copy_cell("B21", "r" & CStr(irow))
    Call copy_cell("B22", "s" & CStr(irow))
    Call copy_cell("B23", "t" & CStr(irow))
    Call copy_cell("F21", "u" & CStr(irow))
    Call copy_cell("I20", "ag" & CStr(irow))
    Call copy_cell("I21", "ah" & CStr(irow))
    Call copy_cell("J3", "ai" & CStr(irow))

End If
Sheets("Data_Sheet").Select
```

```
Application.Calculation = xlAutomatic
Application.ScreenUpdating = True


If single_set = 1 Then
    Sheets("Data_Sheet").Select
    Sheets("Data_Sheet").Copy Before:=Sheets(1)
    Sheets("Data_Sheet (2)").Select
    Sheets("Data_Sheet (2)").Name = "FLEXQUAC"
    Sheets("Data_Sheet").Select
    Sheets("x-P").Select
    ActiveChart.ChartArea.Select
    ActiveChart.SeriesCollection(2).Values = "=UNIQUAC!R26C13:R200C13"
    ActiveChart.SeriesCollection(3).Values = "=FLEXQUAC!R26C13:R200C13"
End If
Sheets("Data_Sheet").Select
If fit_tertest = True Then
    Close (30)
    Open workpath & datafile For Input As #30
    nline_tertest = nline_tertest + 1
End If
If irow Mod 30 = 0 Then ActiveWorkbook.Save
Wend
Close (30)
End Sub
```

APPENDIX C

APPENDIX C

Test Using Numerical Example - VLE Data Set 420

BINARY SYSTEM - DIETHYL ETHER, ACETONITRILE

$$n := 2$$
$$i := 1 .. n \quad \text{general} \qquad eps1 \equiv 10^{-8}$$
$$j := 1 .. n$$

Parameters

Pure Component Parameters

$$R1 := \begin{pmatrix} 3.395 \\ 1.87 \end{pmatrix} \qquad Q := \begin{pmatrix} 3.016 \\ 1.724 \end{pmatrix}$$

FlexQUAC-Q Model Parameters

$$\Delta u := \begin{pmatrix} 0 & 497.605 \\ -76.972 & 0 \end{pmatrix} \qquad \delta := \begin{pmatrix} 0 & 0.1786 \\ 0.1786 & 0 \end{pmatrix}$$

Temperature

$$T := 273.15 + 20.50$$

Pure Gas Constant

$$R := 1.98721$$

Liquid Mole Fraction

$$x := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

FlexQUAC - Q: Residual Part

$$\tau_{i,j} := \exp\left(\frac{-\Delta u_{i,j}}{R \cdot T} \right) \qquad\qquad \tau = \begin{pmatrix} 1 & 0.426 \\ 1.141 & 1 \end{pmatrix}$$

Surface Fraction

$$\text{thetafx}_i := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)$$

$$\begin{array}{l} \text{thetafx}_i = \\ \boxed{\begin{array}{c} 3.945 \\ \hline 1.724 \end{array}} \end{array}$$

$$\text{sthetfx} := \sum_{i} x_i \cdot \text{thetafx}_i$$

$$\text{sthetfx} = 1.724$$

$$\text{thetfx}_i := \frac{\text{thetafx}_i}{\text{sthetfx}}$$

$$\text{thetfx} = \begin{pmatrix} 2.288 \\ 1 \end{pmatrix}$$

$$\text{thethaf}_i := x_i \cdot \text{thetfx}_i$$

$$\begin{array}{l} \text{thethaf}_i = \\ \boxed{\begin{array}{c} 0 \\ \hline 1 \end{array}} \end{array}$$

$$\text{Sf}_i := \sum_{j=1}^{n} x_j \cdot \text{thetfx}_j \cdot \tau_{j,i}$$

$$\text{Sf} = \begin{pmatrix} 1.141 \\ 1 \end{pmatrix}$$

Residual Gibbs Energy

$$\text{f1} := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$\text{f1} = -1.724$$

$$\text{f2} := \sum_{i=1}^{n} x_i \cdot \text{thetfx}_i \cdot \ln\left(\text{Sf}_i\right)$$

$$\text{f2} = 0$$

$$\text{gE\_RTresf} := \text{f1} \cdot \text{f2}$$

$$\text{gE\_RTresf} = 0$$

Derivative of f1 by mole numbers

$$\text{df1dn} := -Q - \text{f1}$$

$$\text{df1dn} = \begin{pmatrix} -1.292 \\ 0 \end{pmatrix}$$

-254-

Derivative of f2 by surface fraction

$$\text{sum1}_j := \sum_{i=1}^{n} \frac{x_i \cdot \text{thetfx}_i \cdot \tau_{j,i}}{\text{Sf}_i}$$

$$\text{df2}_i := \text{sum1}_i + \ln\left(\text{Sf}_i\right)$$

$$\text{df2} = \begin{pmatrix} 0.558 \\ 1 \end{pmatrix}$$

Jacobian matrix

$$\text{SJ}_i := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$\text{SJ} = \begin{pmatrix} 2.653 \\ 1.724 \end{pmatrix}$$

$\text{JacobianFQ1} :=$ | for $i \in 1..n$
   for $m \in 1..n$

   $\text{Jac}_{m,i} \leftarrow -\left(\text{thetfx}_m \cdot x_i\right) - \dfrac{\text{SJ}_m \cdot x_i}{\text{sthetfx}} + \dfrac{Q_i \cdot x_i}{\text{thetfx}_i \cdot \text{sthetfx}}$

   $\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{\text{thetfx}_i \cdot \text{sthetfx}}$ if $i \neq m$

   $\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + 1$ if $i = m$

   $\text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} \cdot \text{thetfx}_i$

 Jac

$$\text{JacobianFQ1} = \begin{pmatrix} 2.288 & -2.288 \\ 0 & 0 \end{pmatrix}$$

$$df2dn := JacobianFQ1 \cdot df2$$

$$df2dn = \begin{pmatrix} -1.011 \\ 0 \end{pmatrix}$$

Residual Activity Coefficient

$$dgE\_RTresf := f1 \cdot df2dn + f2 \cdot (-Q - f1)$$

$$dgE\_RTresf = \begin{pmatrix} 1.743 \\ 0 \end{pmatrix}$$

$$ln\gamma\_resf := dgE\_RTresf + gE\_RTresf$$

$$ln\gamma\_resf = \begin{pmatrix} 1.743 \\ 0 \end{pmatrix}$$

Combinatorial Part

$$r(x, R1) := \sum_{i=1}^{n} x_i \cdot R1_i \qquad\qquad q(x, Q) := \sum_{i=1}^{n} x_i \cdot Q_i$$

$$\phi(x, R, i, n) := \frac{R1_i}{r(x, R1)} \qquad\qquad \Theta(x, Q, i, n) := \frac{Q_i}{q(x, Q)}$$

Combinatorial Term - Gibbs Energy

$$gE\_RTcomb(x, T, \Delta u, R1, Q, n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x, R1, i, n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x, Q, i, n)}{\phi(x, R, i, n)} \right) \right)$$

$$\ln\gamma\_combo(x, T, \Delta u, Q, n) := \left| \begin{array}{l} \text{for } i \in 1 .. n \\[2mm] \quad ret_i \leftarrow 1. - \phi(x, R1, i, n) + \ln(\phi(x, R1, i, n)) + 5 \cdot Q_i \cdot \left(\ln\left(\dfrac{\Theta(x, Q, i, n)}{\phi(x, R1, i, n)} \right) + \dfrac{\phi(x, I}{\Theta(x,} \right. \\[4mm] ret \end{array} \right.$$

$$\ln\gamma\_combo(x, T, \Delta u, Q, n) = \begin{pmatrix} -0.209 \\ 0 \end{pmatrix}$$

Activity Coefficient

$$\ln\gamma(x, T, \Delta u, Q, n) := \ln\gamma\_combo(x, T, \Delta u, Q, n) + \ln\gamma\_resf$$

$$\ln\gamma(x, T, \Delta u, Q, n) = \begin{pmatrix} 1.534 \\ 0 \end{pmatrix}$$

$$\exp(\ln\gamma(x, T, \Delta u, Q, n)) = \begin{pmatrix} 4.638 \\ 1 \end{pmatrix}$$

Summary of Results

$$x0 := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \exp(\ln\gamma(x0,T,\Delta u,Q,n)) = \begin{pmatrix} 4.638 \\ 1 \end{pmatrix}$$

$$x1 := \begin{pmatrix} 0.0980 \\ 0.902 \end{pmatrix} \qquad \exp(\ln\gamma(x1,T,\Delta u,Q,n)) = \begin{pmatrix} 4.897 \\ 0.997 \end{pmatrix}$$

$$x2 := \begin{pmatrix} 0.2040 \\ 0.796 \end{pmatrix} \qquad \exp(\ln\gamma(x2,T,\Delta u,Q,n)) = \begin{pmatrix} 5.123 \\ 0.989 \end{pmatrix}$$

$$x3 := \begin{pmatrix} 0.38 \\ 0.62 \end{pmatrix} \qquad \exp(\ln\gamma(x3,T,\Delta u,Q,n)) = \begin{pmatrix} 5.396 \\ 0.969 \end{pmatrix}$$

$$x4 := \begin{pmatrix} 0.7530 \\ 0.247 \end{pmatrix} \qquad \exp(\ln\gamma(x4,T,\Delta u,Q,n)) = \begin{pmatrix} 5.674 \\ 0.91 \end{pmatrix}$$

$$x5 := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \exp(\ln\gamma(x5,T,\Delta u,Q,n)) = \begin{pmatrix} 5.714 \\ 0.868 \end{pmatrix}$$

Results of Recval

data set : VLE   420   type of data: x,y,P,(T)
reference : JOUKOVSKY N.I.,Bull.Soc.Chim.Belg. 43(10),397(1934).

consistency tests: 1-- 2-o

constant values:
temperature   = 20.50 degree C
exp. sat. vapor pressure Ps1 =   447.10 mm Hg
calc. sat. vapor pressure Ps1 =   447.10 mm Hg
exp. sat. vapor pressure Ps2 =   70.60 mm Hg
calc. sat. vapor pressure Ps2 =   70.60 mm Hg

P [mm Hg] exp.	P [mm Hg] calc.	x 1 [-] exp.	x 1 [-] calc.	y 1 [-] exp.	y 1 [-] calc.	act 1 [-] exp.	act 1 [-] calc.	act 2 [-] exp.	act 2 [-] calc.
70.60	70.60	0.0000	0.0000	0.0000	0.0000	0.00000	4.63812	1.00000	1.00000
176.00	176.78	0.0980	0.0980	0.6150	0.6299	2.47034	2.54132	1.06405	1.02752
251.00	232.82	0.2040	0.2040	0.7500	0.7391	2.06395	1.88657	1.11660	1.08106
321.60	307.60	0.3800	0.3800	0.8310	0.8318	1.57300	1.50591	1.24167	1.18216
405.00	414.99	0.7530	0.7530	0.9310	0.9257	1.11997	1.14112	1.60252	1.76718
447.10	447.10	1.0000	1.0000	1.0000	1.0000	1.00000	1.00000	0.00000	5.73548

APPENDIX C

Parameters:

$$n := 2$$
$$i := 1..n$$
$$j := 1..n \quad \text{general} \qquad eps1 \equiv 10^{-8}$$

Pure Component Parameters

DDB Data Set No: 11573 (Set 1)

Components: 1. Methanol
2. Hexane

$$R\_set1 := \begin{pmatrix} 1.4311 \\ 4.4998 \end{pmatrix} \quad Q\_set1 := \begin{pmatrix} 1.4320 \\ 3.856 \end{pmatrix}$$

FlexQUAC-Q Model Parameters

Set 1 - 11573

$$\Delta u\_set1 := \begin{pmatrix} 0 & 44.68176 \\ 1212.435978 & 0 \end{pmatrix} \qquad \delta\_set1 := \begin{pmatrix} 0 & 0.07816629 \\ 0.07816629 & 0 \end{pmatrix}$$

Temperature

$$T := 273.15 + 20$$

Pure Gas Constant

$$R := 1.98721$$

Mole Fraction

$$x(xx) := \begin{pmatrix} xx \\ 1 - xx \end{pmatrix}$$

-259-

$$\tau(\Delta u, i, j) := \exp\left(\frac{-\Delta u_{i,j}}{R \cdot T}\right)$$

$$\tau n(\Delta u, n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{for } j \in 1..n \\ \qquad \text{ret}_{i,j} \leftarrow \tau(\Delta u, i, j) \\ \text{ret} \end{vmatrix}$$

Surface Fraction

$$\text{thetafx}(x, Q, \delta, i) := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k}\right)$$

$$\text{sthetfx}(x, Q, \delta) := \sum_i x_i \cdot \text{thetafx}(x, Q, \delta, i)$$

$$\text{thetfx}(x, Q, \delta, i) := \frac{\text{thetafx}(x, Q, \delta, i)}{\text{sthetfx}(x, Q, \delta)}$$

$$\text{thethaf}(x, Q, \delta, i) := x_i \cdot \text{thetfx}((x, Q, \delta, i))$$

$$\text{Sf}(x, \Delta u, Q, \delta, i) := \sum_{j=1}^{n} x_j \cdot \text{thetfx}(x, Q, \delta, j) \cdot \tau n(\Delta u, n)_{j,i}$$

Residual Gibbs Energy

$$\text{fl}(x, Q) := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f2(x, \Delta u, Q, \delta) := \sum_{i=1}^{n} x_i \cdot thetfx(x, Q, \delta, i) \cdot \ln(Sf(x, \Delta u, Q, \delta, i))$$

$$gE\_RTresf(x, \Delta u, Q, \delta) := f1(x, Q) \cdot f2(x, \Delta u, Q, \delta)$$

Derivative of f1 by mole numbers

$$df1dn(x, Q) := -Q - f1(x, Q)$$

Derivative of f2 by surface fraction

$$sum1(x, \Delta u, Q, \delta, j) := \sum_{i=1}^{n} \frac{x_i \cdot thetfx(x, Q, \delta, i) \cdot \tau n(\Delta u, n)_{j, i}}{Sf(x, \Delta u, Q, \delta, i)}$$

$$df2(x, \Delta u, Q, \delta, i) := sum1(x, \Delta u, Q, \delta, i) + \ln(Sf(x, \Delta u, Q, \delta, i))$$

$$df2n(x, \Delta u, Q, \delta, n) := \begin{vmatrix} \text{for } i \in 1 .. n \\ \quad ret_i \leftarrow df2(x, \Delta u, Q, \delta, i) \\ ret \end{vmatrix}$$

Jacobian matrix

$$SJ(x, Q, \delta, i) := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k, i}\right)$$

$$\text{JacobianFQ1}(x,Q,\delta) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \begin{array}{|l} \text{Jac}_{m,i} \leftarrow -\left(\text{thetfx}(x,Q,\delta,m) \cdot x_i\right) - \dfrac{\text{SJ}(x,Q,\delta,m) \cdot x_i}{\text{sthetfx}(x,Q,\delta)} + \dfrac{Q_i \cdot x_i}{\text{thetfx}(x,Q,\delta,i) \cdot \text{sthetfx}(} \\[3mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{\text{thetfx}(x,Q,\delta,i) \cdot \text{sthetfx}(x,Q,\delta)} \quad \text{if } i \neq m \\[3mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + 1 \quad \text{if } i = m \\[2mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} \cdot \text{thetfx}(x,Q,\delta,i) \end{array} \\ \text{Jac} \end{array}$$

Derivative of f2 by mole numbers

$$\text{df2dn}(x,\Delta u,Q,\delta) := \text{JacobianFQ1}(x,Q,\delta) \cdot \text{df2n}(x,\Delta u,Q,\delta,n)$$

Residual Activity Coefficient

$$\text{dgE\_RTresf}(x,\Delta u,Q,\delta) := \text{f1}(x,Q) \cdot \text{df2dn}(x,\Delta u,Q,\delta) + \text{f2}(x,\Delta u,Q,\delta) \cdot (-Q - \text{f1}(x,Q))$$

$$\text{ln}\gamma\_\text{resf}(x,\Delta u,Q,\delta) := \text{dgE\_RTresf}(x,\Delta u,Q,\delta) + \text{gE\_RTresf}(x,\Delta u,Q,\delta)$$

Combinatorial Part

$$r(x,R\_1) := \sum_{i=1}^{n} x_i \cdot R\_1_i \qquad\qquad q(x,Q) := \sum_{i=1}^{n} x_i \cdot Q_i$$

$$\phi(x,R\_1,i,n) := \frac{R\_1_i}{r(x,R\_1)} \qquad\qquad \Theta(x,Q,i,n) := \frac{Q_i}{q(x,Q)}$$

Combinatorial Term - Gibbs Energy

$$gE\_RTcomb(x,T,\Delta u,R\_1,Q,n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x,R\_1,i,n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R\_1,i,n)} \right) \right)$$

Combinatorial Activity Coefficient

$$\ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \\ ret_i \leftarrow 1. - \phi(x,R\_1,i,n) + \ln(\phi(x,R\_1,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R\_1,i,n)} \right) \right) \cdot \\ \\ ret \end{vmatrix}$$

Activity Coefficient

$$\ln\gamma(x,T,\Delta u,Q,R\_1,\delta,n) := \ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) + \ln\gamma\_resf(x,\Delta u,Q,\delta)$$

Summary of Results

Due to the large number of data points only ten experimental points were chosen for evaluation

Data Set: 11573

$$m := 1..10 \qquad xx1 := \begin{pmatrix} 0 \\ 0.00002 \\ 0.00076 \\ 0.14830 \\ .85920 \\ .95020 \\ .97730 \\ .99150 \\ .99930 \\ 1 \end{pmatrix}$$

-263-

x1	y1	ACT$_1^{calc}$	ACT$_2^{calc}$
0.00000	0.00000	37.8790355	1
0.00002	0.00000	37.8669076	1
0.00003	0.00000	37.8608456	1.00000001
0.00027	0.00000	37.7157579	1.00000058
0.00076	0.00000	37.4219022	1.00000461
0.00188	0.00000	36.7619577	1.00002812
0.00233	0.00000	36.5013114	1.00004313
0.00609	0.00000	34.419854	1.00029106
0.00829	0.00000	33.2774383	1.00053554
0.02319	0.00000	26.7700909	1.00400337
0.02443	0.00000	26.3113184	1.00442676
0.04142	0.00000	21.005933	1.01213088
0.07120	0.00000	14.8285048	1.03325778
0.08599	0.00000	12.7115861	1.0469099
0.10228	0.00000	10.8640556	1.06411922
0.14830	0.00000	7.40620343	1.12380039
0.85920	0.00000	1.09720694	6.49547206
0.90070	0.00000	1.05530301	8.65868707
0.93490	0.00000	1.02704195	11.7521736
0.93970	0.00000	1.02366029	12.3462347
0.94420	0.00000	1.02064372	12.9519807
0.95020	0.00000	1.016868	13.8427652
0.95400	0.00000	1.0146315	14.4621654
0.95540	0.00000	1.01383926	14.7022183
0.96600	0.00000	1.00843618	16.7576171
0.97200	0.00000	1.00588408	18.1399616
0.97730	0.00000	1.0039668	19.521622
0.98240	0.00000	1.00244497	21.0173987
0.98393	0.00000	1.00205394	21.5018201
0.98666	0.00000	1.00143491	22.4110961
0.98710	0.00000	1.00134481	22.5632939
0.98989	0.00000	1.00083785	23.5671428
0.99150	0.00000	1.00059717	24.1785112
0.99268	0.00000	1.0004456	24.642375
0.99430	0.00000	1.00027249	25.302006
0.99488	0.00000	1.00022052	25.5448249
0.99670	0.00000	1.0000925	26.3305069
0.99713	0.00000	1.00007012	26.52156
0.99750	0.00000	1.00005331	26.6876788
0.99848	0.00000	1.00001981	27.1354329
0.99850	0.00000	1.00001929	27.144675
0.99895	0.00000	1.00000948	27.3542508
0.99920	0.00000	1.00000551	27.4717399
0.99930	0.00000	1.00000422	27.5189628
0.99962	0.00000	1.00000125	27.6708949
1.00000	0.00000	1	27.8529634

Calculated activity coefficient 1

$$\exp\left(\ln\gamma\left(x\left(xx1_m\right), T, \Delta u\_set1, Q\_set1, R\_set1, \delta\_set1, n\right)_1\right) =$$

37.879
37.867
37.422
7.406
1.097
1.017
1.004
1.001
1
1

Calculated activity coefficient 2

$$\exp\left(\ln\gamma\left(x\left(xx1_m\right), T, \Delta u\_set1, Q\_set1, R\_set1, \delta\_set1, n\right)_2\right) =$$

1
1
1
1.124
6.495
13.843
19.522
24.178
27.519
27.853

APPENDIX C

$$n := 2$$
$$i := 1 .. n \quad \text{general} \qquad eps1 \equiv 10^{-8}$$
$$j := 1 .. n$$

Parameters:

Pure Component Parameters

DDB Data Set No: 11570 (Set 2)

Components: 1. Methanol
2. Cyclohexane

$$R\_set2 := \begin{pmatrix} 1.4311 \\ 4.0464 \end{pmatrix} \quad Q\_set2 := \begin{pmatrix} 1.4320 \\ 3.24 \end{pmatrix}$$

FlexQUAC-Q Model Parameters

Set 2 - 11570

$$\Delta u\_set2 := \begin{pmatrix} 0 & 82.00264 \\ 982.5527 & 0 \end{pmatrix} \qquad \delta\_set2 := \begin{pmatrix} 0 & 0.195406 \\ 0.195406 & 0 \end{pmatrix}$$

Temperature

$$T := 273.15 + 20$$

Pure Gas Constant

$$R := 1.98721$$

Mole Fraction

$$x(xx) := \begin{pmatrix} xx \\ 1 - xx \end{pmatrix}$$

$$\tau(\Delta u, i, j) := \exp\left(\frac{-\Delta u_{i,j}}{R \cdot T}\right)$$

$$\tau n(\Delta u, n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{for } j \in 1..n \\ \quad\quad \text{ret}_{i,j} \leftarrow \tau(\Delta u, i, j) \\ \text{ret} \end{vmatrix}$$

Surface Fraction

$$\text{thetafx}(x, Q, \delta, i) := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)$$

$$\text{sthetfx}(x, Q, \delta) := \sum_{i} x_i \cdot \text{thetafx}(x, Q, \delta, i)$$

$$\text{thetfx}(x, Q, \delta, i) := \frac{\text{thetafx}(x, Q, \delta, i)}{\text{sthetfx}(x, Q, \delta)}$$

$$\text{thethaf}(x, Q, \delta, i) := x_i \cdot \text{thetfx}((x, Q, \delta, i))$$

$$\text{Sf}(x, \Delta u, Q, \delta, i) := \sum_{j=1}^{n} x_j \cdot \text{thetfx}(x, Q, \delta, j) \cdot \tau n(\Delta u, n)_{j,i}$$

Residual Gibbs Energy

$$f1(x, Q) := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f2(x, \Delta u, Q, \delta) := \sum_{i=1}^{n} x_i \cdot \text{thetfx}(x, Q, \delta, i) \cdot \ln(\text{Sf}(x, \Delta u, Q, \delta, i))$$

$$gE\_RTresf(x, \Delta u, Q, \delta) := f1(x, Q) \cdot f2(x, \Delta u, Q, \delta)$$

Derivative of f1 by mole numbers

$$df1dn(x, Q) := -Q - f1(x, Q)$$

Derivative of f2 by surface fraction

$$sum1(x, \Delta u, Q, \delta, j) := \sum_{i=1}^{n} \frac{x_i \cdot thetfx(x, Q, \delta, i) \cdot \tau n(\Delta u, n)_{j,i}}{Sf(x, \Delta u, Q, \delta, i)}$$

$$df2(x, \Delta u, Q, \delta, i) := sum1(x, \Delta u, Q, \delta, i) + \ln(Sf(x, \Delta u, Q, \delta, i))$$

$$df2n(x, \Delta u, Q, \delta, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad ret_i \leftarrow df2(x, \Delta u, Q, \delta, i) \\ ret \end{array}$$

Jacobian matrix

$$SJ(x, Q, \delta, i) := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$JacobianFQ1(x, Q, \delta) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \begin{array}{|l} Jac_{m,i} \leftarrow -\left(thetfx(x, Q, \delta, m) \cdot x_i\right) - \dfrac{SJ(x, Q, \delta, m) \cdot x_i}{sthetfx(x, Q, \delta)} + \dfrac{Q_i \cdot x_i}{thetfx(x, Q, \delta, i) \cdot sthetfx(} \\ Jac_{m,i} \leftarrow Jac_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{thetfx(x, Q, \delta, i) \cdot sthetfx(x, Q, \delta)} \quad \text{if } i \neq m \\ Jac_{m,i} \leftarrow Jac_{m,i} + 1 \quad \text{if } i = m \\ Jac_{m,i} \leftarrow Jac_{m,i} \cdot thetfx(x, Q, \delta, i) \end{array} \\ Jac \end{array}$$

Derivative of f2 by mole numbers

$$df2dn(x, \Delta u, Q, \delta) := JacobianFQ1(x, Q, \delta) \cdot df2n(x, \Delta u, Q, \delta, n)$$

Residual Activity Coefficient

$$dgE\_RTresf(x, \Delta u, Q, \delta) := f1(x, Q) \cdot df2dn(x, \Delta u, Q, \delta) + f2(x, \Delta u, Q, \delta) \cdot (-Q - f1(x, Q))$$

$$\ln\gamma\_resf(x, \Delta u, Q, \delta) := dgE\_RTresf(x, \Delta u, Q, \delta) + gE\_RTresf(x, \Delta u, Q, \delta)$$

Combinatorial Part

$$r(x, R\_1) := \sum_{i=1}^{n} x_i \cdot R\_1_i \qquad\qquad q(x, Q) := \sum_{i=1}^{n} x_i \cdot Q_i$$

$$\phi(x, R\_1, i, n) := \frac{R\_1_i}{r(x, R\_1)} \qquad\qquad \Theta(x, Q, i, n) := \frac{Q_i}{q(x, Q)}$$

Combinatorial Term - Gibbs Energy

$$gE\_RTcomb(x, T, \Delta u, R\_1, Q, n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x, R\_1, i, n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x, Q, i, n)}{\phi(x, R\_1, i, n)} \right) \right)$$

$$\ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad ret_i \leftarrow 1. - \phi(x,R\_1,i,n) + \ln(\phi(x,R\_1,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R\_1,i,n)} \right) \right) \\ ret \end{vmatrix}$$

Activity Coefficient

$$\ln\gamma(x,T,\Delta u,Q,R\_1,\delta,n) := \ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) + \ln\gamma\_resf(x,\Delta u,Q,\delta)$$

Summary of Results

Due to the large number of data points only ten experimental points were chosen for evaluation

Data Set: 11570

$$xx2 := \begin{pmatrix} 0 \\ 0.00058 \\ 0.00277 \\ 0.01225 \\ 0.05726 \\ 0.09010 \\ 0.82570 \\ 0.90020 \\ 0.93900 \\ 1 \end{pmatrix}$$

$$m := 1..10$$

-269-

Calculated activity coefficient 1

x1	y1	ACT$_1^{calc}$	ACT$_2^{calc}$
0.00000	0.00000	55.9110671	1
0.00007	0.00000	55.8210037	1.00000006
0.00058	0.00000	55.1706174	1.00000386
0.00152	0.00000	53.9980554	1.00002644
0.00277	0.00000	52.4896917	1.00008732
0.00402	0.00000	51.0370477	1.00018291
0.00730	0.00000	47.4725252	1.00059476
0.01225	0.00000	42.6995838	1.00164052
0.02257	0.00000	34.6503068	1.00534241
0.04035	0.00000	25.0243742	1.01597045
0.05726	0.00000	19.022098	1.03032453
0.08700	0.00000	12.5869184	1.06372872
0.09010	0.00000	12.1101572	1.06772649
0.14040	0.00000	7.11176535	1.14356175
0.50980	0.00000	1.66374603	2.04791272
0.82570	0.00000	1.16160115	4.50838104
0.82860	0.00000	1.1579917	4.57602772
0.85060	0.00000	1.13087982	5.18143162
0.86600	0.00000	1.11226364	5.73011147
0.90020	0.00000	1.07264952	7.55071402
0.91350	0.00000	1.05822761	8.61589462
0.93900	0.00000	1.03307152	11.6794471
0.96090	0.00000	1.01539091	16.2487462
0.98110	0.00000	1.00407599	23.7524267
1.00000	0.00000	1	36.9120802

$\exp\left(\ln\gamma\left(x\left(xx2_m\right), T, \Delta u\_set2, Q\_set2, R\_set2, \delta\_set2, n\right)_1 \right)$:

55.911
55.171
52.49
42.7
19.022
12.11
1.162
1.073
1.033
1

Calculated activity coefficient 2

$\exp\left(\ln\gamma\left(x\left(xx2_m\right), T, \Delta u\_set2, Q\_set2, R\_set2, \delta\_set2, n\right)_2 \right)$:

1
1
1
1.002
1.03
1.068
4.508
7.551
11.679
36.912

APPENDIX C

$$n := 2$$
$$i := 1 .. n$$
$$j := 1 .. n$$ general $$\quad eps1 \equiv 10^{-8}$$

Parameters:

Pure Component Parameters

DDB Data Set No: 11567 (Set 3)

Components: 1. Hexane
2. Cyclohexane

$$R\_set3 := \begin{pmatrix} 4.4998 \\ 4.0464 \end{pmatrix} \quad Q\_set3 := \begin{pmatrix} 3.856 \\ 3.24 \end{pmatrix}$$

FlexQUAC-Q Model Parameters

$$\Delta u\_set3 := \begin{pmatrix} 0 & -105.905 \\ 133.687 & 0 \end{pmatrix} \qquad \delta\_set3 := \begin{pmatrix} 0 & 0.073377 \\ 0.073377 & 0 \end{pmatrix}$$

Temperature

$$T := 273.15 + 20$$

Pure Gas Constant

$$R := 1.98721$$

Mole Fraction

$$x(xx) := \begin{pmatrix} xx \\ 1 - xx \end{pmatrix}$$

$$\tau(\Delta u, i, j) := \exp\left(\frac{-\Delta u_{i,j}}{R \cdot T}\right)$$

$$\tau n(\Delta u, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad \text{for } j \in 1..n \\ \quad\quad \text{ret}_{i,j} \leftarrow \tau(\Delta u, i, j) \\ \text{ret} \end{array}$$

Surface Fraction

$$\text{thetafx}(x, Q, \delta, i) := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k}\right)$$

$$\text{sthetfx}(x, Q, \delta) := \sum_i x_i \cdot \text{thetafx}(x, Q, \delta, i)$$

$$\text{thetfx}(x, Q, \delta, i) := \frac{\text{thetafx}(x, Q, \delta, i)}{\text{sthetfx}(x, Q, \delta)}$$

$$\text{thethaf}(x, Q, \delta, i) := x_i \cdot \text{thetfx}((x, Q, \delta, i))$$

$$\text{Sf}(x, \Delta u, Q, \delta, i) := \sum_{j=1}^{n} x_j \cdot \text{thetfx}(x, Q, \delta, j) \cdot \tau n(\Delta u, n)_{j,i}$$

Residual Gibbs Energy

$$f1(x, Q) := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f2(x, \Delta u, Q, \delta) := \sum_{i=1}^{n} x_i \cdot \text{thetfx}(x, Q, \delta, i) \cdot \ln(\text{Sf}(x, \Delta u, Q, \delta, i))$$

$$\text{gE\_RTresf}(x, \Delta u, Q, \delta) := f1(x, Q) \cdot f2(x, \Delta u, Q, \delta)$$

Derivative of f1 by mole numbers

$$\text{df1dn}(x, Q) := -Q - f1(x, Q)$$

Derivative of f2 by surface fraction

$$\text{sum1}(x, \Delta u, Q, \delta, j) := \sum_{i=1}^{n} \frac{x_i \cdot \text{thetfx}(x, Q, \delta, i) \cdot \text{tn}(\Delta u, n)_{j,i}}{\text{Sf}(x, \Delta u, Q, \delta, i)}$$

$$\text{df2}(x, \Delta u, Q, \delta, i) := \text{sum1}(x, \Delta u, Q, \delta, i) + \ln(\text{Sf}(x, \Delta u, Q, \delta, i))$$

$$\text{df2n}(x, \Delta u, Q, \delta, n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{ret}_i \leftarrow \text{df2}(x, \Delta u, Q, \delta, i) \\ \text{ret} \end{vmatrix}$$

Jacobian matrix

$$\text{SJ}(x, Q, \delta, i) := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$\text{JacobianFQ1}(x, Q, \delta) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \begin{vmatrix} \text{Jac}_{m,i} \leftarrow -\left(\text{thetfx}(x, Q, \delta, m) \cdot x_i\right) - \frac{\text{SJ}(x, Q, \delta, m) \cdot x_i}{\text{sthetfx}(x, Q, \delta)} + \frac{Q_i \cdot x_i}{\text{thetfx}(x, Q, \delta, i) \cdot \text{sthetfx}(} \\ \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + \frac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{\text{thetfx}(x, Q, \delta, i) \cdot \text{sthetfx}(x, Q, \delta)} \text{ if } i \neq m \\ \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + 1 \text{ if } i = m \\ \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} \cdot \text{thetfx}(x, Q, \delta, i) \end{vmatrix} \\ \text{Jac} \end{vmatrix}$$

-273-

$$df2dn(x, \Delta u, Q, \delta) := JacobianFQ1(x, Q, \delta) \cdot df2n(x, \Delta u, Q, \delta, n)$$

Residual Activity Coefficient

$$dgE\_RTresf(x, \Delta u, Q, \delta) := f1(x, Q) \cdot df2dn(x, \Delta u, Q, \delta) + f2(x, \Delta u, Q, \delta) \cdot (-Q - f1(x, Q))$$

$$ln\gamma\_resf(x, \Delta u, Q, \delta) := dgE\_RTresf(x, \Delta u, Q, \delta) + gE\_RTresf(x, \Delta u, Q, \delta)$$

Combinatorial Part

$$r(x, R\_1) := \sum_{i=1}^{n} x_i \cdot R\_1_i \qquad\qquad q(x, Q) := \sum_{i=1}^{n} x_i \cdot Q_i$$

$$\phi(x, R\_1, i, n) := \frac{R\_1_i}{r(x, R\_1)} \qquad\qquad \Theta(x, Q, i, n) := \frac{Q_i}{q(x, Q)}$$

Combinatorial Term - Gibbs Energy

$$gE\_RTcomb(x, T, \Delta u, R\_1, Q, n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x, R\_1, i, n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x, Q, i, n)}{\phi(x, R\_1, i, n)} \right) \right)$$

$$\ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \\ ret_i \leftarrow 1. - \phi(x,R\_1,i,n) + \ln(\phi(x,R\_1,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R\_1,i,n)} \right) \right) \cdot \\ \\ ret \end{vmatrix}$$

$$\ln\gamma(x,T,\Delta u,Q,R\_1,\delta,n) := \ln\gamma\_combo(x,T,\Delta u,Q,R\_1,n) + \ln\gamma\_resf(x,\Delta u,Q,\delta)$$

Summary of Results

Due to the large number of data points only ten experimental points were chosen for evaluation

Data Set: 11567

$$m := 1..10 \qquad xx3 := \begin{pmatrix} 0 \\ 0.10040 \\ 0.18990 \\ 0.29410 \\ .39680 \\ .49280 \\ .59460 \\ .69010 \\ .79690 \\ 1 \end{pmatrix}$$

x1	y1	ACT$_1$^{calc}	ACT$_2$^{calc}
0.00000	0.00000	1.19876306	1
0.10040	0.00000	1.1242232	1.00311836
0.18990	0.00000	1.08632083	1.00885893
0.29410	0.00000	1.05866376	1.01708316
0.39680	0.00000	1.04066399	1.02626148
0.49280	0.00000	1.0285493	1.03590216
0.59460	0.00000	1.01878546	1.04773029
0.69010	0.00000	1.01163415	1.06108816
0.79690	0.00000	1.00550579	1.08004384
1.00000	0.00000	1	1.13814471

Calculated activity coefficient 1

$$\exp\left(\ln\gamma\left(x\left(xx3_m\right), T, \Delta u\_set3, Q\_set3, R\_set3, \delta\_set3, n\right)_1\right) =$$

1.199
1.124
1.086
1.059
1.041
1.029
1.019
1.012
1.006
1

Calculated activity coefficient 2

$$\exp\left(\ln\gamma\left(x\left(xx3_m\right), T, \Delta u\_set3, Q\_set3, R\_set3, \delta\_set3, n\right)_2\right) =$$

1
1.003
1.009
1.017
1.026
1.036
1.048
1.061
1.08
1.138

-276-

APPENDIX C

Testing Ternary VLE Data Set 11564

TERNARY SYSTEM - CYCLOHEXANE, METHANOL & HEXANE

$n := 3$

$i := 1 .. n$ general $eps1 \equiv 10^{-8}$

$j := 1 .. n$

Parameters

Pure Component Parameters

$$R1 := \begin{pmatrix} 4.0464 \\ 1.4311 \\ 4.4998 \end{pmatrix} \qquad Q := \begin{pmatrix} 3.24 \\ 1.4320 \\ 3.856 \end{pmatrix}$$

FlexQUAC-Q Model Parameters

$$\Delta u := \begin{pmatrix} 0 & 982.5527 & 133.687 \\ 82.00264 & 0 & 44.68176 \\ -105.905 & 1212.436 & 0 \end{pmatrix}$$

$$\delta := \begin{pmatrix} 0 & 0.195406 & 0.073377 \\ 0.195406 & 0 & 0.07816629 \\ 0.073377 & 0.07816629 & 0 \end{pmatrix}$$

Temperature

$T := 273.15 + 20$

Pure Gas Constant

$R := 1.98721$

Mole Fraction

$$x(xx, xx1) := \begin{pmatrix} xx \\ xx1 \\ 1 - xx - xx1 \end{pmatrix}$$

-277-

$$\tau_{i,j} := \exp\left(\frac{-\Delta u_{i,j}}{R \cdot T}\right)$$

Surface Fraction

$$\text{thetafx}(x,i) := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k}\right)$$

$$\text{sthetfx}(x) := \sum_i x_i \cdot \text{thetafx}(x,i)$$

$$\text{thetfx}(x,i) := \frac{\text{thetafx}(x,i)}{\text{sthetfx}(x)}$$

$$\text{thethaf}(x,i) := x_i \cdot \text{thetfx}(x,i)$$

$$\text{Sf}(x,i) := \sum_{j=1}^{n} x_j \cdot \text{thetfx}(x,j) \cdot \tau_{j,i}$$

Residual Gibbs Energy

$$f1(x) := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f2(x) := \sum_{i=1}^{n} x_i \cdot \text{thetfx}(x,i) \cdot \ln(\text{Sf}(x,i))$$

$$\text{gE\_RTresf}(x) := f1(x) \cdot f2(x)$$

-278-

$$df1dn(x) := -Q - f1(x)$$

$$sum1(x,j) := \sum_{i=1}^{n} \frac{x_i \cdot thetfx(x,i) \cdot \tau_{j,i}}{Sf(x,i)}$$

$$df2(x,i) := sum1(x,i) + \ln(Sf(x,i))$$

$$df2n(x,n) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad ret_i \leftarrow df2(x,i) \\ ret \end{vmatrix}$$

$$SJ(x,i) := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$JacobianFQ1(x) := \begin{vmatrix} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \begin{vmatrix} Jac_{m,i} \leftarrow -\left(thetfx(x,m) \cdot x_i\right) - \dfrac{SJ(x,m) \cdot x_i}{sthetfx(x)} + \dfrac{Q_i \cdot x_i}{thetfx(x,i) \cdot sthetfx(x)} \\ Jac_{m,i} \leftarrow Jac_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{thetfx(x,i) \cdot sthetfx(x)} \quad \text{if } i \neq m \\ Jac_{m,i} \leftarrow Jac_{m,i} + 1 \quad \text{if } i = m \\ Jac_{m,i} \leftarrow Jac_{m,i} \cdot thetfx(x,i) \end{vmatrix} \\ Jac \end{vmatrix}$$

-279-

$$df2dn(x) := JacobianFQ1(x) \cdot df2n(x,n)$$

Residual Activity Coefficient

$$dgE\_RTresf(x) := f1(x) \cdot df2dn(x) + f2(x) \cdot (-Q - f1(x))$$

$$ln\gamma\_resf(x) := dgE\_RTresf(x) + gE\_RTresf(x)$$

Combinatorial Part

$$r(x,R1) := \sum_{i=1}^{n} x_i \cdot R1_i \qquad\qquad q(x,Q) := \sum_{i=1}^{n} x_i \cdot Q_i$$

$$\phi(x,R,i,n) := \frac{R1_i}{r(x,R1)} \qquad\qquad \Theta(x,Q,i,n) := \frac{Q_i}{q(x,Q)}$$

Combinatorial Term - Gibbs Energy

$$gE\_RTcomb(x,T,\Delta u,R1,Q,n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x,R1,i,n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)} \right) \right)$$

Combinatorial Activity Coefficient

$$\ln\gamma\_combo(x, T, \Delta u, Q, n) := \begin{array}{|l} \text{for } i \in 1..n \\ \quad ret_i \leftarrow 1. - \phi(x, R1, i, n) + \ln(\phi(x, R1, i, n)) + 5 \cdot Q_i \cdot \left(\ln\left(\frac{\Theta(x, Q, i, n)}{\phi(x, R1, i, n)} \right) + \frac{\phi(x, I)}{\Theta(x,} \right. \\ ret \end{array}$$

Activity Coefficient

$$\ln\gamma(x, T, \Delta u, Q, n) := \ln\gamma\_combo(x, T, \Delta u, Q, n) + \ln\gamma\_resf(x)$$

Summary of Results

$$x1 := \begin{pmatrix} .7620 \\ 0.68320 \\ .16510 \\ 0.05470 \\ .51900 \\ .0992 \\ .0982 \\ .4313 \\ .1192 \\ .3072 \\ 0.0723 \\ 0.0242 \\ 0.1817 \\ 0.0314 \\ 0.0138 \end{pmatrix} \qquad x2 := \begin{pmatrix} 0.05750 \\ 0.15570 \\ 0.79600 \\ 0.93160 \\ 0.20010 \\ 0.8471 \\ 0.8128 \\ 0.1274 \\ 0.7589 \\ 0.1262 \\ 0.7944 \\ 0.9304 \\ 0.1297 \\ 0.8496 \\ 0.9313 \end{pmatrix} \qquad m := 1..15$$

		FlexQUAC-Q			P^{calc}	P_{dev}
x_1	x_2	ACT_1^{calc}	ACT_2^{calc}	ACT_3^{calc}	kPa	kPa
0.76270	0.05750	1.046521425	15.77245331	1.055819869	230.8555	0.207808
0.71780	0.11300	1.111746669	8.683210373	1.08885026	239.5729	4.485532
0.68320	0.15570	1.175491452	6.110939026	1.142140627	236.2022	-0.045061
0.65260	0.19350	1.239710689	4.745593071	1.205356717	232.7506	-3.763261
0.21200	0.73800	3.237323761	1.272613645	3.772146463	223.2851	-13.37539
0.16510	0.79600	4.001228333	1.194775939	4.596377373	220.5814	-16.02582
0.13440	0.83390	4.820527554	1.146621943	5.413010597	218.7671	-17.58679
0.10320	0.87110	6.132819176	1.10152328	6.626318932	217.4477	-16.65303
0.05470	0.93160	10.98756218	1.037474513	10.65252209	211.1212	-6.820871
0.59530	0.08250	1.081368566	11.15514278	1.061445117	241.2257	-1.167669
0.54520	0.15970	1.180840969	5.935912609	1.150584698	244.4242	-1.075595
0.51900	0.20010	1.245008707	4.627930164	1.218364358	242.261	-3.878791
0.1669	0.7427	3.319094419	1.264445424	3.774218559	234.2579	-12.41516
0.1293	0.8007	4.164166927	1.185048103	4.651935101	231.4064	-15.37329
0.1217	0.8129	4.413815022	1.169265389	4.899792194	230.6482	-15.93149
0.0992	0.8471	5.346011639	1.126492262	5.784268379	228.8402	-16.57961
0.0846	0.87	6.249540806	1.099438071	6.603644848	227.2196	-15.93371
0.0459	0.9294	10.99851704	1.037687182	10.51953793	219.3188	-7.635908
0.0982	0.8128	4.498530865	1.166317701	4.897271156	239.1165	-14.66267
0.0705	0.8656	6.203115463	1.101709247	6.458520412	235.6493	-15.14337
0.0383	0.9269	10.94617653	1.038428426	10.35733032	226.4862	-8.347845
0.455	0.0795	1.082150817	11.26332855	1.0582546	246.7018	-3.410954
0.4313	0.1274	1.132399321	7.522862434	1.10923326	253.9609	0.381784
0.4144	0.1617	1.176479697	5.922548294	1.157258272	253.9462	-0.566177
0.1192	0.7589	3.568854094	1.23870194	3.959941149	243.9713	-10.54108
0.1005	0.7967	4.180143833	1.186648846	4.570548058	242.0387	-13.60691
0.0573	0.8841	7.20947361	1.08037889	7.303123951	235.8115	-14.22122
0.3186	0.0938	1.092914343	10.02684498	1.070585012	259.7089	1.196842
0.3072	0.1262	1.122941852	7.728416443	1.108476281	263.7281	2.682926
0.294	0.1638	1.165987372	5.965173244	1.161972165	264.0503	2.338492
0.0723	0.7944	4.23600769	1.18604064	4.541873932	251.7709	-11.56747
0.0641	0.8157	4.716980934	1.15803647	4.989784718	250.7666	-12.963
0.0453	0.8698	6.65099144	1.093111634	6.703318596	246.5026	-13.66265
0.0242	0.9304	11.83656693	1.033547759	10.86916542	234.1463	-7.807158
0.0246	0.9313	11.93829346	1.032914996	10.9572649	233.2826	-8.030854
0.1922	0.0797	1.078157783	11.87164211	1.052734256	268.0845	3.839528
0.1817	0.1297	1.111484289	7.806948662	1.110529184	275.8677	7.836432
0.1793	0.1412	1.121904612	7.178976536	1.126340747	276.0324	7.441099
0.0314	0.8496	5.959610462	1.112957954	6.00107336	257.4666	-11.32461
0.0259	0.8707	6.942171574	1.089020371	6.828286648	255.7456	-11.83238
0.0138	0.9313	12.35376072	1.031419158	11.11178398	240.8683	-7.671285

Activity Coefficient 1

$$\exp\left(\ln\gamma\left(x\left(x1_m, x2_m\right), T, \Delta u, Q, n\right)_1\right) =$$

1.047
1.175
4.001
10.988
1.245
5.346
4.499
1.132
3.569
1.123
4.236
11.837
1.111
5.96
12.354

Activity Coefficient 2

$$\exp\left(\ln\gamma\left(x\left(x1_m, x2_m\right), T, \Delta u, Q, n\right)_2\right) =$$

15.764
6.111
1.195
1.037
4.628
1.126
1.166
7.523
1.239
7.728
1.186
1.034
7.807
1.113
1.031

Activity Coefficient 3

$$\exp\left(\ln\gamma\left(x\left(x1_m, x2_m\right), T, \Delta u, Q, n\right)_3\right) =$$

1.056
1.142
4.596
10.653
1.218
5.784
4.897
1.109
3.96
1.108
4.542
10.869
1.111
6.001
11.112

Appendix C

Ternary LLE Calculation - Methanol - Acetone Cyclohexane

Test Using Numerical Example

Components:

1 - methanol	methanol := 1	n := 3
2 - acetone	acetone := 2	i := 1 .. n
3 - cyclohexane	cyclohexane := 3	j := 1 .. n

Pure Component Parameters:

general $\text{eps1} \equiv 10^{-8}$

$$R := \begin{pmatrix} 1.4311 \\ 2.5735 \\ 4.0464 \end{pmatrix} \qquad Q := \begin{pmatrix} 1.432 \\ 2.3360 \\ 3.24 \end{pmatrix}$$

FlexQUAC-Q Parameters:

$GC := 1.98721$

$$\Delta u := \begin{pmatrix} 0 & 164.0496 & 16.0264 \\ 79.7562 & 0 & -14.6356 \\ 1287.072 & 499.1818 & 0 \end{pmatrix} \qquad \delta := \begin{pmatrix} 0 & 0 & 0.0004 \\ 0 & 0 & 0.15 \\ 0.0004 & 0.15 & 0 \end{pmatrix} \qquad \text{tesla} := 298.15$$

Mole fractions:

$$\tau_{i,j} := \exp\left(\frac{-\Delta u_{i,j}}{GC \cdot \text{tesla}} \right) \qquad \tau = \begin{pmatrix} 1 & 0.758 & 0.973 \\ 0.874 & 1 & 1.025 \\ 0.114 & 0.431 & 1 \end{pmatrix}$$

Combinatorial Part

$$r(x,R,n) := \sum_{i=1}^{n} x_i \cdot R_i \quad q(x,Q,n) := \sum_{i=1}^{n} x_i \cdot Q_i \qquad \phi(x,R,i,n) := \frac{R_i}{r(x,R,n)} \quad \Theta(x,Q,i,n) := \frac{Q_i}{q(x,Q,n)}$$

$$gE\_RTcomb(x,T,\Delta u,R,Q,n) := \sum_{i=1}^{n} \left(x_i \cdot \ln(\phi(x,R,i,n)) + 5 \cdot Q_i \cdot \ln\left(\frac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)} \right) \right)$$

$$\ln\gamma\_combo(x,T,R,\Delta g,Q,n) := \begin{array}{|l} \text{for } i \in 1..n \\[6pt] \quad ret_i \leftarrow 1. - \phi(x,R,i,n) + \ln(\phi(x,R,i,n)) + 5 \cdot Q_i \cdot \left(\ln\left(\dfrac{\Theta(x,Q,i,n)}{\phi(x,R,i,n)} \right) \right) + \dfrac{\phi(x,F}{\Theta(x,(} \\[6pt] ret \end{array}$$

FLEXQUAC - Q

Residual Part

$$thetafx(x,Q,\delta,i,n) := Q_i \cdot \left(1 + \sum_{k=1}^{n} x_k \cdot Q_k \cdot \delta_{i,k} \right)$$

$$sthetfx(x,Q,n) := \sum_i x_i \cdot thetafx(x,Q,\delta,i,n)$$

$$thetfx(x,Q,\delta,i,n) := \frac{thetafx(x,Q,\delta,i,n)}{sthetfx(x,Q,n)}$$

$$Sf(x,Q,\delta,i,n) := \sum_{j=1}^{n} x_j \cdot thetfx(x,Q,\delta,j,n) \cdot \tau_{j,i}$$

G^E residual

$$f1(x,Q) := -\sum_{i=1}^{n} x_i \cdot Q_i$$

$$f2(x,Q,\delta,n) := \sum_{i=1}^{n} x_i \cdot thetfx(x,Q,\delta,i,n) \cdot \ln(Sf(x,Q,\delta,i,n))$$

$$gE\_RTresf(x,Q,\delta) := f1(x,Q) \cdot f2(x,Q,\delta,n)$$

-284-

$$\text{sum1}(x,Q,\delta,j,n) := \sum_{i=1}^{n} \frac{x_i \cdot \text{thetfx}(x,Q,\delta,i,n) \cdot \tau_{j,i}}{\text{Sf}(x,Q,\delta,i,n)}$$

$$\text{df2}(x,Q,\delta,i,n) := \text{sum1}(x,Q,\delta,i,n) + \ln(\text{Sf}(x,Q,\delta,i,n))$$

$$\text{df2}(x,Q,\delta,i,n) =$$

∎

$$\text{df2n}(x,Q,\delta,n) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{ret}_i \leftarrow \text{df2}(x,Q,\delta,i,n) \\ \text{ret} \end{array} \right.$$

$$\text{SJ}(x,Q,\delta,i,n) := \sum_{k=1}^{n} x_k \cdot Q_k \cdot \left(1 + Q_i \cdot \delta_{k,i}\right)$$

$$\text{JacobianFQ1}(x,Q,\delta,n) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{for } m \in 1..n \\ \qquad \left| \begin{array}{l} \text{Jac}_{m,i} \leftarrow -\left(\text{thetfx}(x,Q,\delta,m,n) \cdot x_i\right) - \dfrac{\text{SJ}(x,Q,\delta,m,n) \cdot x_i}{\text{sthetfx}(x,Q,n)} + \dfrac{Q_i}{\text{thetfx}(x,Q,\delta,i,n)} \\[3mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + \dfrac{x_i \cdot Q_i \cdot Q_m \cdot \delta_{i,m}}{\text{thetfx}(x,Q,\delta,i,n) \cdot \text{sthetfx}(x,Q,n)} \text{ if } i \neq m \\[3mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} + 1 \text{ if } i = m \\[2mm] \text{Jac}_{m,i} \leftarrow \text{Jac}_{m,i} \cdot \text{thetfx}(x,Q,\delta,i,n) \end{array} \right. \\ \text{Jac} \end{array} \right.$$

Derivatives

df2/dn

$$\text{df2dn}(x,Q,\delta,n) := \text{JacobianFQ1}(x,Q,\delta,n) \cdot \text{df2n}(x,Q,\delta,n)$$

d(G^E/RT)/dn

$$\text{dgE\_RTresf}(x,Q,\delta,n) := \text{f1}(x,Q) \cdot \text{df2dn}(x,Q,\delta,n) + \text{f2}(x,Q,\delta,n) \cdot (-Q - \text{f1}(x,Q))$$

ln$\gamma_{residual}$

$$\text{ln}\gamma\_\text{resf}(x,Q,\delta,n) := \text{dgE\_RTresf}(x,Q,\delta,n) + \text{gE\_RTresf}(x,Q,\delta)$$

lnγ

$$\text{ln}\gamma(x,i,R,Q,n) := \text{ln}\gamma\_\text{resf}(x,Q,\delta,n)_i + \text{ln}\gamma\_\text{combo}(x,T,R,\Delta u,Q,n)_i$$

$$\gamma(x,i,R,Q,n) := \exp(\text{ln}\gamma(x,i,R,Q,n))$$

Excel calculated molar composition from K factor method:

$$x1 := \begin{pmatrix} .119963 \\ 0 \\ .880037 \end{pmatrix} \qquad x2 := \begin{pmatrix} 0.198356 \\ 0.033085 \\ 0.768559 \end{pmatrix} \qquad x3 := \begin{pmatrix} 0.429705 \\ 0.092166 \\ 0.478129 \end{pmatrix}$$

Calculated activity coefficients

$\gamma(x1,i,R,Q,n) =$

7.5
4.886
1.056

$\gamma(x2,i,R,Q,n) =$

4.254
2.703
1.189

$\gamma(x3,i,R,Q,n) =$

1.877
1.46
1.878