TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Notation

A	Internediate variable used in the calculation of parameter a
A_o	Constant used in the estimation of the boiling point temperature
a	Energy parameter of Peng-Robinson Equation of State (bar cm^6/mol^2)
a_c	Parameter in Peng-Robinson Equation of State
B	Second virial coefficient
B_o	Constant used in the estimation of the boiling point temperature
b	co-volume parameter of Peng-Robinson Equation of State (cm^3/mol)
C	Third virial coefficient
C_i	Constants used in the estimation of parameters a and m (i = 1,2…5)
C_j	Contribution of the molecular group j to the estimation of the parameter c
C_o	Constant used in the estimation of the boiling point temperature
C_{PL}	Saturated liquid heat capacities
c	volume correction parameter of Peng-Robinson Equation of State (cm^3/mol)
D	Forth virial coefficient
D_x	Distance from the tangent plane to the Gibbs free energy surface
D_x^*	Normalized distance from the tangent plane to the Gibbs free energy surface
f	fugacity (bar)
f_i	function used in the estimation of parameter a (i = 1,2)
f_i^l	Fugacity value for the component i in the liquid phase (bar)
f_i^v	Fugacity value for the component i in the vapour phase (bar)
g_i	stationary point condition (QNSS algorithm)
h_i	Intermediate variable in the stability test calculation
I_k	Number of occurrences
K_i	Equilibrium constant
k_{ij}	Interaction parameter
M_j	Contribution of the molecular group j to the estimation of the shape parameter m
m	Shape parameter

Nc	Number of carbons in the molecule
N_j	Number of groups of type j
N_p	Number of measurements
n	Number of moles
p	Pressure (bar)
pc	Critical pressure (bar)
R	Universal gas constant (bar cm^3 / (mol K))
S	Intermediate variable used in the estimation of parameters b, m and c
T	Temperature (K)
T_b	Boiling point temperature (K)
T_b^{GC}	Boiling point temperature estimated using the group contribution method (K)
T_c	Critical temperature (K)
T_r	Reduced Temperature (K)
U	Total internal energy (J)
V	Total volume (cm_3)
v	Molar volume (cm^3/mol)
$v_{corrected}$	Corrected molar volume (cm^3/mol)
V_{Wj}	Contribution of the j^{th} group to the Van der Waals volume
w	acentric factor
W_L	Speed of sound in saturated liquids
X	Constant used in the estimation of parameter a
$\bar{x}$	Vector of liquid molar compositions
x_i	Molar fraction of component i in the liquid phase
Y	Constant used in the estimation of parameter a
y_i	Molar fraction of component i in vapour phase
$\bar{y}$	Vector of vapour molar compositions
Z	Compressibility factor
z_i	molar fraction of component i in the feed

Symbols

α	Parameter in Peng-Robinson Equation of State

α_i	Intermediate variable used in the estimation of parameter c (i = 0, 1, 2 and b)
β	Amount of vapour divided by the amount of feed
β_i	Intermediate variable used in the estimation of parameter c (i = 0, 1, 2 and b)
ΔH_{VAP}	Heats of vaporization
δV_{WK}	Correction to van der Walls volume
$\delta_r(X)$	Relative error of variable X
φ_i	Fugacity coefficient of component i
μ_i	Chemical potential of component i
ρ	Density (g/cm^3)
$\bar{\xi}$	Intermediate variable in the QNSS algorithm
σ	Intermediate variable in the QNSS algorithm
τ_{bj}	Contribution of the molecular group j to the estimation of the boiling point temperature

Abbreviations

AVR	Athabasca Vacuum Residue
EOS	Equation of State
CME	Chemical and Materials Engineering
GC	Group Contribution
LS	Liquid-solid
L_1L_2	Liquid-liquid
MCR	Micro Carbon Residue
NMR	Nuclear Magnetic Resonance
PVT	Pressure-Volume-Temperature
QNSS	Quasi Newton Successive Substitution
SARA	Saturates, Aromatics, Resins, and Asphaltehens
VL	Vapour-liquid
VLE	Vapour liquid equilibrium
VL_1L_2S	Vapour-liquid-liquid-solid
VLS	Vapor-liquid-solid
VS	Vapour-solid

1. INTRODUCTION

1.1. Motivation

Thermophysical properties have been defined as those affecting the transfer of heat without changing the chemical identity of a material. Usually, these variables include temperature-pressure-density relationships, thermal conductivity and diffusivity, heat capacity, thermal expansion and thermal radiative properties, as well as viscosity and mass and thermal diffusion coefficients, speed of sound, surface and interfacial tension in fluids. Thermophysical properties play an important role in many industries and research areas as they define compounds in a precise way, which is needed for the modeling and forecast of events in many processes.

In the petroleum industry, accurate and consistent values of thermophysical properties are a key element in simulation, design, and analysis of products and processes. However, for heavy hydrocarbons the required properties are not often available in the literature or experimental values are very difficult to obtain due to thermal decomposition at temperatures of interest. In these cases thermodynamic models should be used.

In the case of phase behaviour calculations (Pressure-Temperature-Density relationships), cubic equations of state (EOS) have been widely used due to its performance and simplicity. They require the computation of three parameters (energy parameter a, co-volume b and volume correction c). Usually these parameters are estimated from critical properties (critical temperature T_c, and critical pressure p_c) and acentric factor of the components. However, due to thermal cracking, critical properties are not available for high molecular weight components.

In order to avoid the need for critical properties in the estimation of parameters of the EOS, another approach should be taken. Correlations can be considered as one option, but the ones available in the literature are often restricted to limited compounds and temperature or pressure conditions. As an alternative, a new approach developed specifically for heavy hydrocarbons considers the estimation of the parameters of the EOS from a Group Contribution (GC) concept, using information from the molecular structure and the boiling point temperature of the components (Coniglio et al., 2000).

Consequently, the present work focuses on the estimation of thermophysical properties of heavy hydrocarbon compounds using the Peng-Robinson equation of state, where parameters are estimated using a group contribution method. This approach requires only information about the molecular structure of the components. The sub molecular groups, employed by the method, are based on the work of Coniglio et al. (2000). A computer program implementing the algorithm will be prepared and tested for known pure compounds and for mixtures. As a final step the method will be applied with Athabasca Vacuum Residue (AVR).

1.2. Background

There are two important research projects that relate directly to the present study and these are presented in this section. The first project concerns the molecular structure of AVR and constitutes the main input data for the group contribution model. The second project, a forerunner to the present work and completed recently

(Mahmoodaghdam et al., 2002), also applied a group contribution approach to estimate thermophysical properties of AVR. Its results are compared with the performance of the group contribution model developed in the present study.

1.2.1. Athabasca Vacuum Residue and its molecular structure

Bitumen is a heavy black viscous hydrocarbon fluid that must be treated to convert it into an upgraded crude oil before it can be used by refineries to produce gasoline and diesel fuels. It also requires dilution with lighter hydrocarbons to make it transportable by pipeline.

Alberta's oil sands comprise one of the world's two largest sources of bitumen; the other is in Venezuela. Oil sands are found in three places in Alberta -the Athabasca, Peace River and Cold Lake regions- and cover a total of nearly 140,800 Km^2.

Oil sands currently represent 54 per cent of Alberta's total oil production, and about one-third of all the oil produced in Canada. By 2005, oil sands production is expected to represent 50 per cent of Canada's total crude oil output, and 10 per cent of North American production (www.energy.gov.ab.ca, 2006).

The extraction of bitumen is accomplished by two main approaches; the first one is via the Clark hot water extraction process and the second one is via in situ bitumen production. In the first case the near surface deposits of bitumen allow the exploitation through surface mining techniques. In the second case, bitumen deposits are present at depths greater than 30-60 m and open pit mining becomes an uneconomically approach, in these cases in situ bitumen production is applied using steam to heat the oil in place increasing permeability values and allowing the flow of bitumen from the wells to surface facilities (Gray, 2004).

The bitumen obtained from the extraction stage is not suitable for conventional refining due to its viscosity and impurities, such as nitrogen, sulphur, minerals and metals. Consequently, prior to be refined the bitumen undertakes an upgrading process. The product of the upgrading process is a "synthetic" crude oil, stable with no residue content, which can be sold as a conventional light sweet crude. Different schemes of upgrading can be applied; the most commons processes encountered within the province of Alberta are, thermal cracking and hydrogen addition processes (Gray, 2004).

The Syncrude Ltd. Process Scheme for upgrading is presented in Figure 1.1. In general, atmospheric and vacuum distillations are processes that take place in the feed separation stage. Atmospheric distillation to a boiling point up to 343 °C is used to recover some distillate from bitumen. However, removal of more distillate requires use of vacuum distillation at pressures of 3-5 KPa to recover gas oil to equivalent boiling points as high as 524 °C (Gray, 2004). The top products are sent to a secondary upgrading stage while the bottoms undergo primary upgrading. The bottom product of the vacuum distillation, Athabasca Vacuum Residue (AVR), is the focus of the present study and some of its thermophysical properties will be estimated using a group contribution model presented in Chapter 4. Properties of AVR are shown in Table 1.1.

Table 1.1 Properties of Athabasca Vacuum Residue (Gray, 2004)

	AVR
°API Gravity	1.6
Density (Kg/m^3)	1063.1
Sulfur (wt%)	5.7
Nitrogen (wppm)	5820
MCR, Solids Free (wt%)	20.7
Solids, Toluene Insolubles (wt%)	0.36
Nickel (wppm)	125
Vanadium (wppm)	319

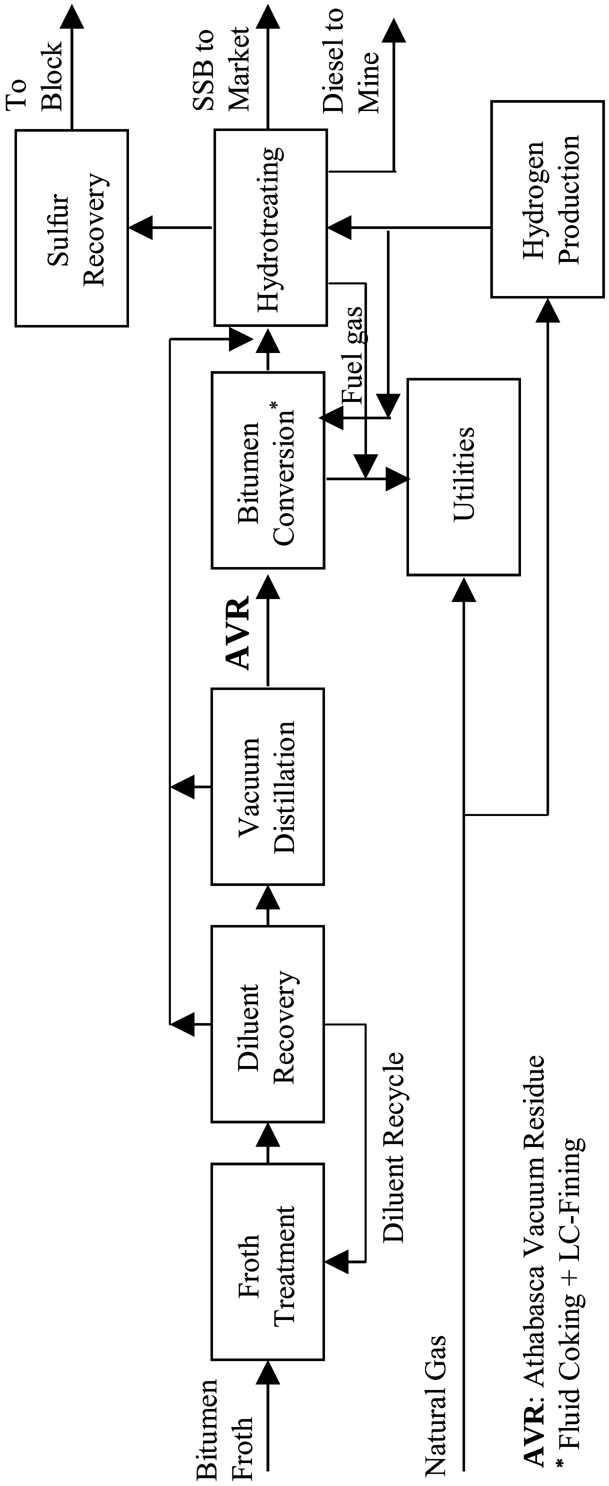


Figure 1.1- Syncrude Process Scheme - Post 1999. (Gray, 2004)

Defining the molecular structure of heavy hydrocarbon fractions remains a challenge with ongoing research efforts focusing on this issue. However, experimental analytical data such as elemental composition, C13 NMR, apparent molecular weight, density and aromaticity are readily available. Vapour pressure, and viscosity are more difficult to obtain reproducibly. The main problem is converting the available experimental data into molecular structures.

One project dealt with this issue specifically for AVR (Sheremata, 2005). It developed molecular representations using Monte Carlo construction method for AVR. Ten fractions were generated from AVR with supercritical fluid extraction. These samples were provided by Syncrude Ltd.. Following analytical characterization, a non linear optimization method was used to select molecules that were consistent with molecular weight, elemental and NMR spectroscopy data and elemental analysis. Starting with 100 molecules, each fraction was represented by a small subset of up to six molecules using a sequential optimization and are consistent with the available experimental data.

The molecular representation determined for AVR (Sheremata, 2005) is used in the present work for the estimation of thermophysical properties of this material.

1.2.2. Estimation of thermophysical properties of Athabasca Vacuum Residue using a single molecule representation

In a first attempt to model thermophysical properties of AVR, a research project was conducted using a group contribution based equation of state (Mahmoodaghdam, 2002). In that study a one molecule representation was used, the characteristics of the molecule were obtained from the molecular study available at that time for AVR (Sheremata, 2002). The molecule selected had an average molecular weight of 1750 and a normal boiling point temperature of 507 °C. The group contribution model used in the study was proposed by Coniglio et al. (2000) and it was designed to estimate thermophysical properties of heavy hydrocarbons with molecular weights higher than 250 and normal boiling point temperatures ranging from 7 – 477 °C. The group contribution model using a one molecule representation failed to predict density and vapour pressure of AVR. The results obtained are discussed and compared with the present study in Chapter 4.

1.3. Research Objectives

The present study evaluates an approach for estimating the thermophysical properties of heavy hydrocarbons and Athabasca Vacuum Residue in particular. In this approach, the Peng-Robinson Equation of State (Peng et al., 1976) is applied, but the parameters of the equation will be evaluated using a group contribution method proposed by Coniglio et al. (2000), only the molecular structure of the components is required. The first research objective is to build a computational model to estimate thermophysical properties of heavy hydrocarbon compounds and extend the approach to mixtures using the group contribution method proposed by Coniglio et al.(2000). The second objective is to estimate thermophysical properties of Athabasca Vacuum Residue using the group contribution model developed. The third objective is to compare model performance with available experimental vapour pressure and density data for this material.

2. LITERATURE REVIEW

Finding reliable values for thermophysical properties is the key to successful simulations, which constitute the main tool used nowadays in the design and optimization of processes. Among all thermophysical properties, vapour pressure and density are important variables. For example, vapour pressure is required in many calculations related to safety as well as design and operation of various units. On the other hand, density is perhaps one of the most relevant physical properties of a fluid, since in addition to its direct use in size calculations it is needed to predict other thermodynamic properties and in some cases to estimate transport properties of dense fluids. Both of these properties can be predicted using pressure-volume-temperature (PVT) relationships. The present study focuses on the establishment of a PVT relationship for heavy hydrocarbons with a special application to Athabasca Vacuum Residue (AVR). In this chapter, PVT relationship options are presented. The group contribution model proposed by Coniglio et al. (2000) is described in detail, as it is the departure point for PVT relationship development in the present work.

Key features required in the present work include: parameter estimation for compounds in the absence of data for critical or boiling properties; easy extension to mixtures; method suitable for thermophysical property and phase behaviour prediction for large molecules combined with small ones. Chapter 2 is organized to present first a general description of phase equilibrium and its basic equations (Section 2.1). Secondly, cubic equations of state are described with an emphasis in Peng-Robinson Equation of State in Section 2.2. Furthermore, Sections 2.3

and 2.4 show in detail the Group Contribution Concept and the Group Contribution Based Equation of State which is the base for the computational model to be developed in the present study. Finally, Section 2.5 describes the extension of the Group Contribution method to mixtures and the last section focuses on the numerical methods required to solve the equilibrium equations.

2.1. Phase Equilibrium

Phase equilibrium calculations are one of the biggest applications of Equations of State (EOS) and perhaps the most important type of calculations in the petroleum industry (Riazi, 2005). In petroleum production, phase equilibrium calculations lead to the determination of the composition and amount of oil and gas produced, Pressure-Temperature diagrams to determine the hydrocarbon phases in reservoirs, solubility of solids in liquids and solid deposition (wax and asphaltene) or hydrate formation. In addition, these calculations are an essential and recurrent element in the simulation of chemical processes in the refining and petrochemical industry.

Any substance can exist in any of these states: liquid, vapour and solid. A system is considered to be at equilibrium when there is no tendency to change. Therefore, in the case of phase equilibrium, a system will be in equilibrium when there is no change in pressure, temperature and compositions in each phase (Riazi, 2005).

A pure substance can exhibit four different equilibrium types: vapour-liquid (VL), vapour-solid (VS), liquid-solid (LS) and vapour-liquid-solid (VLS), the latest only occurs at the triple point. On the other hand a multicomponent system can exhibit multiple combinations of phase equilibrium in addition to the previous, such as liquid-liquid (L_1L_2) or vapour-liquid-liquid-solid (VL_1L_2S) making the calculations more abstract.

Among the equilibrium states, vapour liquid equilibrium (VLE) is one of the most important states. Since it is seen in hydrocarbon mixtures at all stages of oil production and refining. The present study will focus its attention in this kind of equilibrium.

The variables involved in VLE calculations are:

z_i = molar fraction of component i in the feed

T = temperature

p = pressure

x_i= molar fraction of component i in the liquid phase

$y_{i\,=}$ molar fraction of component i in vapour phase

v= molar volume

Z=compressibility factor

Generally, there are five types of VLE calculations:

- *Flash Calculations:* z_i, T and p are known while x_i, y_i and the fraction of vapour are the unknown parameters
- *Bubble Point Calculations:* in this case a first vapour molecule is formed within a liquid phase; the corresponding pressure is called bubble point pressure. There are two types of calculations. In the first type T and x_i are known while pressure (bubble-p) and y_i are the unknown variables. In the second case p and x_i are known while T (bubble-T) and y_i are the unknown

- *Dew Point Calculations:* a vapour phase of known composition starts to condense the first molecule of liquid; this point is called Dew Point. There are two type of calculations, when pressure is unknown (Dew-P) and when T is the unknown variable (Dew-T)

2.1.1. Basic equations of phase equilibrium

The equations presented in this section apply for all types of equilibrium calculations. Considering one mole of a multicomponent mixture, where n_c is the number of components present in the mixture, the following set of equations is used to solve a two phase equilibrium calculation (Michelsen, 1993):

- *n_c Material Balance Equations:*

$$(1-\beta)\, x_i + \beta y_i - z_i = 0 \qquad 2.1$$

- *n_c Equilibrium Relations:*

$$f_i^l(T,p,\overline{x}) - f_i^v(T,p,\overline{y}) = 0 \qquad 2.2$$

- *1 Summation of the mole fractions relation:*

$$\sum_{i=1}^{n_c}(y_i - x_i) = 0 \qquad 2.3$$

Where, β is the amount of vapour divided by the amount of feed and $\overline{x}$ and $\overline{y}$ refer to the liquid and vapour compositions respectively. f_i^l and f_i^v represent the fugacity values for the component i in the liquid and vapour phases.

The total number of equations becomes (2 n_c + 1), while the number of variables remains (2 n_c + 3). The two additional specifications usually assign values to two of the independent variables such as p and β or T and β.

In addition to Equations 2.1-2.3, thermodynamically consistent models are necessary for calculation of liquid and vapour properties (fugacity coefficients, densities and sometimes enthalpies) and all properties for a given phase should be calculated by the same thermodynamic model. Usually, an equation of state is used to estimate the properties in both phases but it is also possible to use an equation of state for the vapour phase and an activity model for the liquid phase.

2.2. Cubic Equations of State (EOS)

Cubic equations of state are a group of equations obtained by modifying the van der Waals equation of State (van der Waals, 1873) and are mathematical expressions that relate pressure, volume and temperature. According to van der Waals assumptions, molecules have a finite diameter, thus making a part of the volume unavailable for molecular motion (excluded volume), and intermolecular attraction decreases the pressure. Equations 2.4-2.5 present a general form for cubic EOS (Sengers et al., 2000):

$$p = \frac{RT}{v - b} - p_{att}(T, v) \qquad 2.4$$

$$p_{att}(T, v) = \frac{a}{v(v + d) + e(v - d)} \qquad 2.5$$

Where, a, b, d and e can be constants or functions of temperature and some fluid properties such as acentric factor, normal boiling point, etc. and R is the universal gas constant.

Although the cubic equations of state are not the most appropriate models for the representation of pure fluid properties, these are the most frequently used equations for practical applications. This is due to the fact that they offer the best balance between accuracy, reliability, simplicity and speed of computation. In addition, they have the advantage of representing multiple phases and multicomponent mixtures using the same model.

2.2.1. Peng-Robinson Equation of State

Peng-Robinson EOS is one of the most popular cubic EOS. It is applied in many areas such as research, design and simulation of chemical processes and many other applications within the oil industry. It was developed during the 1970s and one of its main characteristics is that it does not introduce additional parameter beyond the original two presented in van der Waals EOS. However, it introduces two important modifications to van der Waals EOS (Peng et al., 1976):

- Use of a temperature dependent parameter a, as presented in Equation 2.6. The relation to estimate $\alpha(T)$ is taken from Soave (1972) and it depends on the acentric factor and the critical temperature as shown in Equations 2.7-2.10.

$$a = a_c \alpha(T) \tag{2.6}$$

$$a_c = \frac{0.45724 R^2 T^2}{p_c} \tag{2.7}$$

$$\alpha = \left[1 + m(1 - T_r^{1/2})\right]^2 \tag{2.8}$$

$$m = 0.480 + 1.574\omega - 0.175\omega^2 \tag{2.9}$$

$$T_r = \frac{T}{T_c} \tag{2.10}$$

Where, ω refers to the acentric factor and T_c is the critical temperature.

- Modification to the functional form of the pressure-volume relationship. Peng and Robinson recognized that the critical compressibility factor (Z_c) of the Redlich-Kwong EOS (1949), which is equal to 0.333, is greater than the compressibility factors of practically all fluids (Sengers et al., 2000). Therefore, they postulated a function that reduces Z_c to 0.307 as presented in Equation 2.11.

$$p = \frac{RT}{v - b} - \frac{a(T)}{v(v + b) + b(v - b)} \tag{2.11}$$

The second parameter of Peng-Robinson EOS is a function of the critical temperature (T_c) and pressure (p_c) of the substance and is calculated using the expression presented in Equation 2.12:

$$b = \frac{0.07780RT_c}{p_c} \tag{2.12}$$

This model provides reasonable accuracy for VLE calculations especially for liquid volumes of medium-size hydrocarbons and other compounds with intermediate values of the acentric factor. In addition, the extension to mixture calculations is relatively easy using mixing and combining rules of any complexity.

2.3. Group Contribution Concept

To obtain coefficients for EOS using the Group Contribution (GC) concept, a molecule is divided into a set of functional groups whose properties have been regressed exogenously and comprise a data bank. The contributions of the funcitional groups are combined to obtain properties of complete molecules. This concept can be applied in many areas; the most common applications are to obtain critical properties or boiling and freezing points of the components, and for use in EOS.

The GC concept assumes that the intermolecular forces that determine the properties of interest depend primarily on the bonds between the atoms of the molecules as well as on the nature of the atoms involved and that functional groups can be treated independently of their arrangements or their neighbours. The identity of functional groups and the number of functional groups are normally assumed in advance and their contributions are obtained by fitting available experimental data for vapour pressure, density, critical properties etc.. Corrections for specific multigroup, conformational or resonance effects can also be included (Poling et al., 2001).

The most applied GC methods found in the literature to estimate critical properties are the method of Joback (1984;1987), Constantinou and Gani (1994), Wilson and Jasperson (1996), and Marrero and Pardillo (1999). Other GC methods estimate boiling or freezing points. Finally, a different approach of using the GC concept in VLE calculations is to apply the GC method to estimate parameters of an EOS, this approach is described next with the specific description of the GC method proposed by Coniglio et al. (2000).

2.4. Group Contribution Based Equations of State

The Group Contribution Based Equations of State are equations of state where the parameters appearing in them are evaluated using a group contribution method. In the present section the Group Contribution Based Equation of State presented by Coniglio et al. (2000) is described in detail.

The cubic EOS used as departure function is presented in Equation 2.13. It is based on Peng-Robinson EOS (Peng et al., 1976) along with the cubic EOS-consistent volume correction introduced by Peneloux et al. and Rauzy (1982).

$$P = \frac{RT}{v - b} - \frac{a(T)}{v^2 + 2bv - b^2} \quad \text{with} \quad v_{\text{corrected}} v - c(T) \qquad 2.13$$

The parameters a, b and c of the EOS have been modified to fit the estimation of high boiling hydrocarbon properties and are estimated through a GC concept.

2.4.1. *Characteristics of the GC Method*

The GC concept developed by Coniglio et al., includes seven basic molecular groups and certain structural increments that take into account deviations of specific molecules from the general approach. The basic molecular groups are shown in Table 2.1.

Table 2.1- Basic Molecular Groups (Coniglio et al., 2000)

Alkanes	**Aromatics**
CH_3	ACH
CH_2	AC substituded
CH	AC condensed
C quaternary	

The performance of a predictive model to estimate thermophysical properties depends not only on molecular group definition and related analytical expressions, but also on the quality and number of measurements selected to fit group parameters. The GC method developed by Coniglio et al. considered a variety of hydrocarbons including alkanes, naphthenes, alkylbenzenes and polynuclear aromatics. Two basic databases were used (A and B), the first one was used to estimate the GC parameters, and includes vapour pressure and density measurements. The second database was used to check the thermodynamic consistency of the method, and contains heats of vaporization (ΔH_{VAP}), saturated liquid heat capacities (C_{PL}) and speed of sound in saturated liquids (W_L). The compounds included in Database B were selected among very high boiling compounds in order to test the predictive abilities of the method. Table 2.2 presents the main characteristics of the two databases.

Furthermore, for the vapour pressure database, a screening process to check consistency of the experimental data was made. Two sets of measurements made by two different authors could only exhibit a discrepancy between 2-12 % between them.

Table 2.2- Characteristics of GC Method Databases (Coniglio et al., 2000)

Database	Number of Compounds	T_b Interval (K)	Measurement Interval
A1.- Vapour Pressure	128	300-600	10^{-6} – 7 (bar)
A2.- Liquid Density	121	300-600	triple point – boiling point pressure = 2 bar
B1.- ΔH_{VAP}	69	-	triple point – boiling point pressure = 2 bar
B2.- C_{PL}	69	-	triple point – boiling point pressure = 2 bar
B3.- W_L	scarce	-	atmospheric pressure

2.4.2. Estimation of parameter b

The calculation of the co-volume parameter b is presented in equation 2.14. It is assumed to be proportional to the Van der Waals volume of the molecular groups (Bondi, 1968) and the van der Waals volume of methane is taken as a reference.

$$b = b_{CH4} \frac{\left(\sum_{j=1}^{7} V_{Wj} N_j + \sum_{j=1}^{3} \delta V_{Wk} I_k \right)}{V_{WCH_4}} \qquad 2.14$$

V_{Wj} is the contribution of the j^{th} group to the Van der Waals volume and N_j is the number of groups of type j. δV_{WK}, represents a correction introduced by the method to special cases and I_k represents the number of corresponding occurrences.

The methane co-volume b_{CH4} is obtained from its critical properties and has a value of b_{CH4}=26.80 cm^3/mol. The van der Waals volume for all the molecular groups are obtained from Bondi (1968).

2.4.3. Estimation of temperature dependent parameter a

An exponential form for the parameter a(T) was selected by Coniglio et al. The original expression was proposed by Melhem (1989) and then modified in order to fit heavy hydrocarbon properties. This expression meets three basic requirements:

1. It is positive and finite for all values of temperature.
2. It approaches a finite value (close to zero) as temperature raises to infinity.
3. It is second order continuous (possesses a second derivative). Therefore, heat capacities at constant volume and at constant pressure can be calculated.

Equation 2.15 presents the expression that estimates parameter a(T). Temperature is the principal variable and it takes as parameters, the boiling point temperature and the shape parameter of compounds. $a(T_b)$ is the value of a(T) at the normal boiling temperature and should be estimated first following the procedure presented in section 2.4.5.

$$a(T) = a(T_b)\exp\left\{ f_1(m)\left[1-\left(\frac{T}{T_b}\right)^X\right] - f_2(m)\left[1-\left(\frac{T}{T_b}\right)^Y\right]\right\} \qquad 2.15$$

The shape parameter m, which is characteristic of each compound, has a role similar to the acentric factor (Rogalski et al., 1991) and it is estimated through the group contribution concept presented in the next section.

The expressions to evaluate f_1 and f_2 are presented below – equations 2.16 and 2.17. The values of parameters X and Y were optimized to fit all thermophysical properties considered in the databases (Coniglio et al.): X= 0.4, Y=1/X, C_1=0.787820 and C_2=-16.2469.

$$f_1(m) = \left[\frac{C_2}{XC_2 - Y}\right]m - \left[\frac{C_1 Y}{XC_2 - Y}\right] \qquad 2.16$$

$$f_2(m) = \left[\frac{1}{XC_2 - Y}\right]m - \left[\frac{C_1 X}{XC_2 - Y}\right] \qquad 2.17$$

2.4.4. Estimation of the shape parameter m

The expression for the estimation of the shape parameter m presented by Coniglio et al. has a logarithmic form - equations 2.18 and 2.19, where the values for coefficients C_3, C_4 and C_5 are 0.30048, 0.08425 and $0.88/\sqrt{C_4}$ respectively.

$$\mathrm{m} = \mathrm{C}_3 + \mathrm{S} - \mathrm{C}_5 \ln(1 + \mathrm{C}_4 \mathrm{S}^2) \qquad 2.18$$

$$\mathrm{S} = \sum_{j=1}^{7} M_{\mathrm{j}} \mathrm{N}_{\mathrm{j}} + \sum_{k=1}^{6} \delta m_k I_k \qquad 2.19$$

The first term in Equation 2.19 corresponds to the contribution of the 7 basic molecular groups to the shape parameter while the second term takes into consideration structural increments related to special cases with a number of occurrences of $\mathrm{I_k}$. The complete set of parameters of Equations 2.18-2.19 is presented in Table 2.3.

There is a list of five light hydrocarbons requiring an additional correction to the shape parameter m. Their shape parameter is estimated through Equations 2.20 and 2.21 with values for Δm presented in Table 2.4.

$$\mathrm{m} = \mathrm{C}_3 + \mathrm{S} - \mathrm{C}_5 \ln(1 + \mathrm{C}_4 \mathrm{S}^2) + \Delta \mathrm{m} \qquad 2.20$$

$$\mathrm{S} = \sum_{j=1}^{7} M_{\mathrm{j}} \mathrm{N}_{\mathrm{j}} \qquad 2.21$$

Table 2.3- Group Contributions and Structural Increments for Parameter m (Coniglio et al., 2000)

Groups	**M_j**	**Groups**	**M_j**
alkanes		aromatics	
CH_3	0.04963	ACH	0.03208
CH_2	0.05024	AC substituted	0.04232
CH	0.02920	AC condensed rings	0.00714
C quaternary	0.00000		

structural increments	**δm**
Normal Alkanes: Nc= number of carbons in molecule $I_1=N_c$ for $N_c<7$ $I_1=N_c^{-0.5}$ for $N_c>7$	0.07491
Branched Alkanes: $N_{c,p}$= number of carbons in main chain N_{subst}= number of substitutions $I_2=N_{c,p} + N_{subst} - 6$ for $N_c<8$ $I_2=2$ for $N_c>8$	0.01126
Ring Systems: per cyclopentyl ring: I_3	-0.03804
per cyclohexyl ring: I_4	-0.07293
per isopropyl or terbutyl group attached to an aromatic or naphthenic group: I_5	0.01433
Polynuclear aromatics: $I_6=1$ for naphthalene and derived compounds $I_6=(-1)^{\alpha}\, 0.15(N_{c,2}+2N_{c,3})$ for other compounds α=number of aromatic rings in the molecule $N_{c,2}$ and $N_{c,3}$= number of aromatic condensed ring carbon atoms in common with two or three rings respectively	0.01561

Table 2.4.- Compounds Requiring Correction Δm

Compound	**Δm**
cyclopentane	-0.03911
cyclohexane	-0.04600
isopropylcyclopentane	-0.04326
benzene	0.03196
toluene	0.00972

2.4.5. Estimation of the parameter a at the boiling point temperature a(T_b)

The calculation of parameter a(T) requires first the estimation of parameter a at boiling point temperature a(T_b) where the corresponding pressure is set to P_o=1.01325 bar. The procedure to estimate a(T_b) is iterative. a is changed until vapour pressure and boiling temperature converge to the set values T_b and P_o. The method is illustrated in Figure 2.1.

For convenience, in the proposed iterative algorithm by Coniglio et al. the variable A = a/(RT) is introduced to substitute for the use of a. In addition, the chosen values for $A^{(1)}$ and β (Figure 2.1) facilitate convergence.

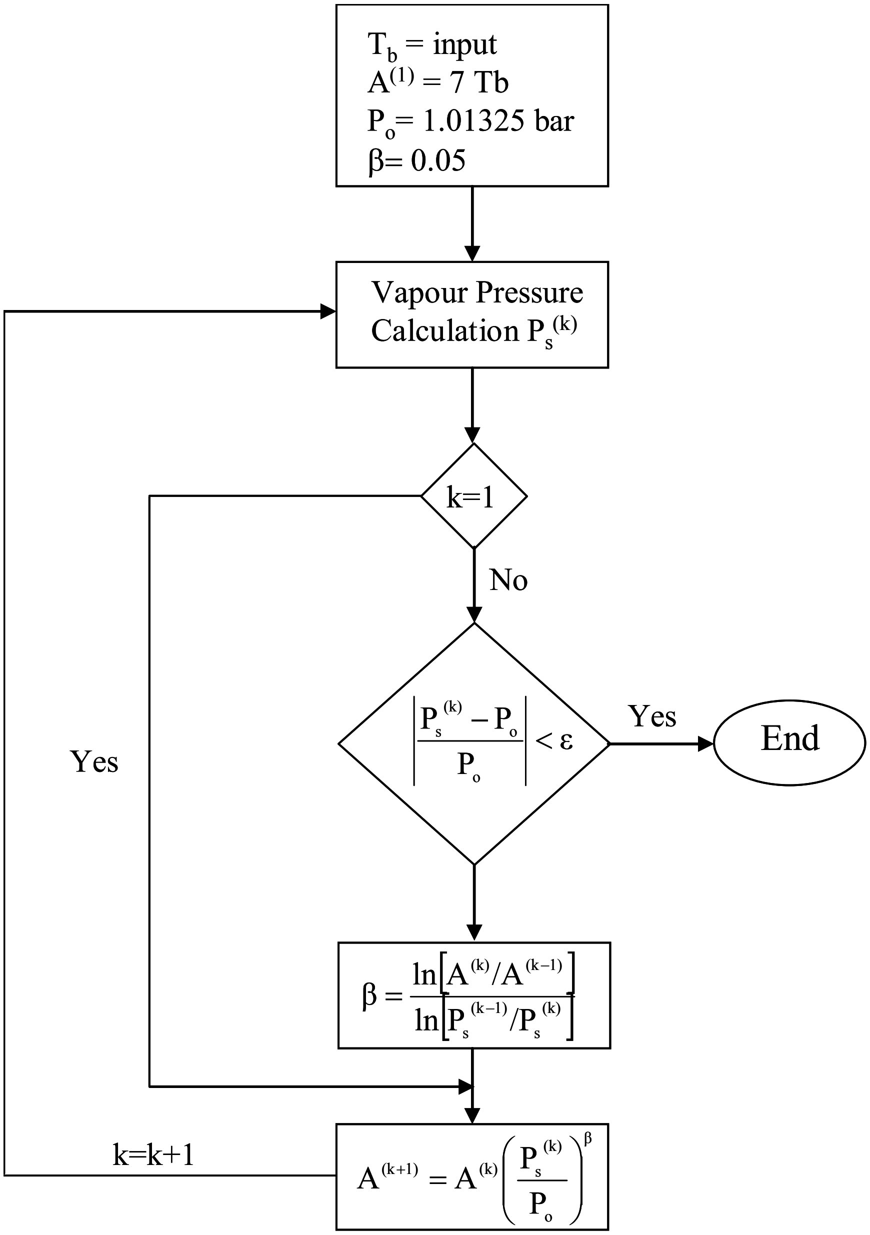


Figure 2.1- Flowchart for calculation of parameter a(T_b). (Coniglio et al., 2000)

2.4.6. Estimation of the temperature dependent parameter c

The expression proposed by Coniglio et al. to estimate the volume correction parameter c depends on the shape parameter and the boiling point temperature of the compounds present. In addition, it requires the estimation of the parameter S defined in

Equation 2.26 through the group contribution concept. The complete set of equations required to estimate parameter c is presented in Equations 2.22-2.26.

$$c(T) = c(T_b)\left[1 + \alpha_o(1 - Y) + \beta_o(1 - Y)^2\right] + (\sqrt{2} - 1)b \qquad 2.22$$

$$Y = \exp\left(1 - \frac{T}{T_b}\right) \qquad 2.23$$

$$\alpha_o = \alpha_1 T_b^{\,2} + \alpha_2 \quad \text{and} \quad \beta_o = \beta_1 T_b + \beta_2 \qquad 2.24$$

$$c(T_b) = \alpha_b T_b + \beta_b m^2 - S \qquad 2.25$$

$$S = \sum_{j=1}^{7} C_j N_j + \sum_{k=1}^{3} \delta C_k I_k \qquad 2.26$$

With α_1=1.89213 10^{-6}, α_2=-0.25116, β_1=2.20483 10^{-3}, β_2=-1.22706, α_b=0.27468 and β_b=-50.94930.

Using the same approach described for the estimation of parameter m, the estimation of the parameter S in this case, takes into consideration the contributions from the seven molecular groups of the method and special cases as well. The group contributions and structural increments for the estimation of the volume correction at boiling point temperature are presented in Table 2.5.

Table 2.5- Group Contributions and Structural Increments for Parameter c(T_b) (Coniglio et al., 2000)

Groups	**C_j (cm^3/mol)**	**Groups**	**C_j (cm^3/mol)**
alkanes		aromatics	
CH_3	35.9209	ACH	19.3874
CH_2	13.2044	AC substituted	-4.2938
CH	-11.4445	AC condensed	7.4171
C quaternary	-36.3578		

structural increments	**δC (cm^3/mol)**
per cyclopentyl and per cyclohexyl ring singly bonded or in condensed cyclic naphthene in trans conformation: I_1	44.2322
per cyclopentyl and per cyclohexyl ring singly bonded or in condensed cyclic naphthene in cis conformation: I_2	46.1907
Per methylene ring condensed to aromatic ring (system): I_3	46.1120

2.4.7. Estimation of boiling point temperature using the group contribution concept

The proposed GC method (Coniglio et al., 2000) uses the boiling point temperature of the compounds present as reference temperatures. However, this datum is frequently unavailable experimentally for large organic molecules due to thermal decomposition. In these cases, T_b must be estimated. Various estimation methods for T_b can be found in the literature. However, due to the importance of this parameter in the estimation of vapour pressures, a higher degree of accuracy is necessary. As a consequence, a GC approach was proposed for the estimation of T_b (Coniglio et al., 2001), where the parameters were fit using the same databases as the original GC method (Coniglio et al., 2000).

The proposed model to estimate T_b (Coniglio et al., 2001) is presented in equations 2.27 and 2.28. The calculation of S requires the application of the GC concept.

$$T_b^{GC} = A_o + B_o(\ln S)^2 + C_o \frac{b}{m} \qquad 2.27$$

$$S = \sum_{j=1}^{7} \tau_{bj} N_j + \sum_{k=1}^{15} \delta\tau_{bj} I_k \qquad 2.28$$

A_o=258.257 K, B_o=49.6530 K and Co=-1.35746(2-√2) Kcm^3/mol. b and m are respectively, the covolume parameter b and the shape parameter m, both defined previously in the original GC method (Coniglio et al., 2000). The basic functional groups and structural corrections for the estimation of S are presented in Table 2.6.

Table 2.6- Group Contributions and Structural Increments for Parameter T_b (Coniglio et al., 2001)

Groups	τ_b	**Groups**	τ_b
alkanes		aromatics	
CH_3	0.803931	ACH	1.39979
CH_2	1.77186	AC substituted	3.06025
CH	2.56070	AC condensed	3.57473
C quaternary	3.32912		

structural increments	$\delta\tau_b$
Normal Alkanes: Nc= number of carbons in molecule $I_1=N_c$ for $N_c<7$ $I_1=N_c^{-0.5}$ for $N_c>7$	-1.57503
Branched Alkanes: $N_{c,p}$= number of carbons in main chain N_{subst}= number of substitutions	

$I_2=N_{c,p} + N_{subst} - 6$ for $N_c<8$ $I_2=2$ for $N_c>8$	-0.209878
Nonaromatic Branched Systems:	
I_3= occurrence of CH(CH$_3$)- CH(CH$_3$) structure	0.161289
I_4= occurrence of CH(CH$_3$)- CH(CH$_3$)$_2$ structure	0.454852
I_5= occurrence of CH(CH$_3$)$_2$- CH(CH$_3$)$_2$ structure	1.21475
Naphthenes (five or six membered rings):	
I_6=occurrence of five-membered ring	-1.93623
I_7=occurrence of six-membered ring	-1.25952
$I_8 = (-1)^{\lambda 1-1}$ per cis double branching [1-2 or 1-3] in five or [1-2,1-3 or 1-4] in six membered rings $\lambda 1$= position of the first cis branching	0.166022
Aromatic systems:	
I_9= occurrence of CH$_3$-AC structure	-0.678644
I_{10}= occurrence of CH$_2$(noncyclic)-AC structure	-0.913446
I_{11}= occurrence of CH$_2$(cyclic)-AC structure	-0.542889
I_{12}= occurrence of [CH(noncyclic)-AC or C(noncyclic)-AC] structure	-1.24178
I_{13}= occurrence of substitution in position 2, 2', 6 or 6' in biphenyl structure	-1.30161
I_{14}= occurrence of α-vicinal noncyclic disturbation	0.231818
Condensed Polynuclear Aromatics:	
I_{15}=1 for naphthalene and derived compounds $I_6=(-1)^{\alpha}\, 0.15(N_{c,2}+2N_{c,3})$ for other compounds α=number of aromatic rings in the molecule $N_{c,2}$ and $N_{c,3}$= number of aromatic condensed ring carbon atoms in common with two or three rings respectively	-1.17975

As noted above, vapour pressures calculated via the GC method are very sensitive to errors in T_b. For example, a deviation in T_b equal to 1 K leads to a deviation in vapour pressure of 5 % (Coniglio et al., 2000). Deviations observed by the authors did not exceed 0.3 K for compounds in Database B. Vapour pressures were calculated to within experimental accuracy.

2.4.8. Performance and limitations of the GC Method

The GC method was tested by the authors (Coniglio et al., 2000) with 78 hydrocarbon compounds, 23 compounds with known T_b and 55 with unknown T_b. The maximum average deviation was 3.58 % for compounds with known T_b. Vapour pressure predictions for hydrocarbons with unknown T_b have the same accuracy as those with known T_b. These results are reproduced in Table 2.7. N_p refers to the number of measurements and the percent relative error is calculated using Equation 2.29.

$$\delta_r(X)\,\% = \frac{100}{N_p}\sum_{i=1}^{N_p}\left|\frac{X_i^{exp} - X_i^{calc}}{X_i^{exp}}\right| \qquad 2.29$$

Table 2.7.- Vapour Pressure Predictions GC method (Coniglio et al., 2001)

Class of hydrocarbons	**Experimental T_b known (23 compounds)**		**T_b estimated via the proposed GC method (55 compounds)**	
	Np	$\delta_r(P_s)$ (%)	Np	$\delta_r(P_s)$ (%)
normal alkanes	218	1.33	190	1.79
branched alkanes			82	2.96
cyclopentanes	4	0.39	15	0.44
cyclohexanes	194	1.40	119	1.05
alkylbenzenes	158	3.58	519	2.10
polynuclear aromatics	124	1.80	196	2.05
overall	698	1.94	1121	1.97

This GC method will be tested in the present study for large multifunctional hydrocarbon compounds associated the bitumen fractions. A detailed analysis of the performance of this approach for this application is presented in Chapter 4.

2.4.9. Simplification of the GC method applicable to mixtures (Crampon et al., 2003)

In the previous sections the original version of the GC method proposed by Coniglio et al. was presented in detail. Based on the information presented by the authors, the method has acceptable performance for VLE calculations of heavy hydrocarbons. However, the applicability of the method as a software tool for general mixtures is difficult due to the many corrections introduced as pseudo structural increments. A simplification was developed by Crampon et al. (2003). It eliminates most of the specific structural increments and increases the number of molecular groups. The performance of the simplified method was tested and deviations in the order of the magnitude of the experimental uncertainties (1 %) were obtained. Based on the good performance and simplicity of this version of the GC method, it was applied in the present work to estimate vapour pressures of mixtures.

The simplified GC method applies the same Peng-Robinson based Equation of State and the GC concept to evaluate the parameters as the GC method proposed by Coniglio et al. Only in this case volume correction was not introduced. The underlying equation of state is the same as the original version developed by Peng et al:

$$P = \frac{RT}{v-b} - \frac{a(T)}{v^2 + 2bv - b^2} \qquad 2.30$$

The parameter b is calculated using the same equation as in the original GC method (Equation 2.14), except for the elimination of the structural increments δV_{Wk}.

$$b = b_{CH4} \frac{\left(\sum_{j=1}^{7} V_{W_j} N_j \right)}{V_{WCH_4}} \qquad 2.31$$

The parameter a(T) is calculated using the same expressions presented in the original version of the method (Coniglio et al., 2000). However, in this new version of the method, the parameter m is calculated using the expression proposed by Trassy (1998), but without taking into account the correction terms introduced for some specific molecules.

$$m = 0.23269 S^{\left(\frac{1}{S^{0.6}}+0.5\right)} + 0.08781 \ln(S) + 0.59180 \qquad 2.32$$

S is again an intermediate variable calculated by group contributions:

$$S = \sum_{j=1}^{N_G} M_j N_j \qquad 2.33$$

Where N_j is the number of occurrences of group j.

This new version of the GC method (Crampon et al. 2003) increases the number of molecular groups from 7 to 19, eliminates the specific corrections that were introduced as structural increments and introduces a new expression for the shape parameter. Table 2.8 shows all the new molecular groups along with the values for the van der Waals volumes V_j (used in the estimation of parameter b) and the group contributions for the estimation of the shape parameter m (M_j), which is calculated with Equations 2.34-2.35.

$$m = 0.23269 S^{\left(\frac{1}{S^{0.6}}+0.5\right)} + 0.08781 \ln(S) + 0.59180 \qquad 2.34$$

$$S = \sum_{j=1}^{19} M_j N_j \qquad 2.35$$

Table 2.8- Group Contributions for the van der Waals Volumes and the shape factor m (Crampon et al., 2003)

Groups	**V_j**	**M_j**
Alkanes		
CH_3	13.67	0.085492
CH_2	10.23	0.082860
CH	6.78	0.047033
C	3.33	-0.028020
Naphthenes		
CH_2	10.23	0.062716
CH	6.78	0.034236
C	3.33	-0.010213
CH (ring/ring	6.78	0.010039
C (ring/aromatic	3.33	0.051147
Aromatic		
CH	8.06	0.050476
C	5.54	0.071528
C condensed	4.74	0.013697
Aliphatic Alkanes		
=CH2	11.94	0.059938
=CH-	8.47	0.069836
=C<	5.01	0.060085
=C=	6.96	0.112156
Double bond near	8.47	0.092399
Aliphatic Alkyles		
C	8.05	0.141491
CH	11.55	0.138136

Crampon et al. (2003) tested the method using a database of more than 128 hydrocarbons including n-alkanes up to n-triacontane and other compounds such as fluorene and acenaphthene, obtaining precisions very similar to the original version and always better than using the Peng-Robinson equation of state where parameters are obtained from the critical properties of the compounds.

2.5. Application of the GC Method to mixtures

In the previous section the GC method was presented in detail for pressure-volume calculations for single compounds. A key objective of the present study is to estimate vapour pressures and densities of multicomponent heavy hydrocarbon mixtures, including Athabasca Vacuum Residue. In order to extend the calculations to mixtures, the parameters a(T) and b for the mixture must be estimated from information pertaining to single compounds or pseudo compounds present. Consequently, a mixing rule must be introduced to the GC-EOS.

Numerous mixing rule options are available in the literature (Sengers et al., 2000). The most common one is the classical quadratic mixing rule. Others include composition-dependent combining rules, density dependent mixing rules and combining equations of state with Excess Gibbs Free Energy models.

Although mixing rules per se play an important role in vapour pressure calculations, the present study focuses on the analysis of the GC method and its performance related to the estimation of vapour pressures and densities of heavy hydrocarbon mixtures instead of analysing the effect of mixing rules on calculations. Therefore, the van der Waals classical quadratic mixing rules are applied.

The classical quadratic mixing rules can be deduced from the composition dependence of virial coefficients by expanding the original van der Waals Equation of State as a power series around zero density (Sengers et al., 2000):

$$Z = 1 + \sum_{i=1}^{\infty}\left(\frac{b}{v}\right)^{i} - \frac{a}{RTv} \quad 2.36$$

The second, third and higher virial coefficients are B=b-a / (RT), C=b^2, D=b^3, and so on. The second virial coefficient has a quadratic composition dependence. Therefore, in order to maintain consistency, the parameters a and b can be at most quadratic functions of composition. In addition, the third virial coefficient imposes an extra restriction, that the parameter b should only be a linear function of composition. Consequently, the classical quadratic mixing rules become:

$$a = \sum\sum x_i x_j a_{ij} \quad 2.37$$

The term a_{ij} refers to the geometric mean applied to the pure compound parameters a_{ii} and a_{jj}, as it is shown in Equation 2.38:

$$a_{ij} = (a_{ii}a_{jj})^{1/2}(1 - k_{ij}) \quad 2.38$$

k_{ij} is an adjustable binary parameter, also known as interaction parameter.

The interaction parameter is usually calculated through regression analysis of experimental phase equilibrium data, although some predictive correlations have been proposed for some mixtures (Riazi, 2005). The most common property regressed is bubble pressure (Valderrama, 2003).

The parameter b for the mixture is expressed as a linear function of composition:

$$b = \sum x_i b_i \quad 2.39$$

These classical quadratic mixing rules are, in general, suitable for the representation of phase equilibrium in multicomponent systems containing nonpolar and weakly polar components (Sengers et al., 2000), which represent the type of compounds that are evaluated in the present study.

2.6. Phase Equilibrium Calculations

This section presents the methods selected to solve the equilibrium problem using the GC-EOS model. In section 2.6.1 the single component equilibrium calculations are explained and in sections 2.6.2 and 2.6.3 the stability criteria and the tangent plane criterion are presented, which constitute the methods selected to solve the equilibrium problem in multicomponent mixtures.

2.6.1. Equilibrium calculations for single compounds

Vapour pressures and saturated liquid densities for single compounds are calculated following the GC method presented by Coniglio et al. (2000).

The calculation of vapour pressures and densities of single compounds involves an iterative calculation where the value for pressure is adjusted until the fugacities of both phases are equal. At the end the value for pressure becomes the vapour pressure at the given temperature

However, in order to mathematically solve the problem the EOS should be rearranged as follows (Cartlidge, 1997):

$$Z^3 + (B-1)Z^2 + (A-3B^2-2B)Z - AB + B^2 + B^3 = 0 \qquad 2.40$$

$$Z = \frac{pv}{RT} \qquad 2.41$$

$$A = \frac{ap}{R^2T^2} \qquad 2.42$$

$$B = \frac{bp}{RT} \qquad 2.43$$

Equation 2.40 represents Peng-Robinson EOS in the cubic form, where Z is the compressibility factor defined in Equation 2.41. The Equation is solved to obtain three roots, where Z_{min} and Z_{max}, when real and different, represent the liquid and vapour compressibility factors respectively.

The fugacity computation is made with Equation 2.44 (Cartlidge, 1997):

$$\ln\left(\frac{f}{p}\right) = (Z-1) - \ln(Z-B) - \frac{A}{2\sqrt{2}B}\ln\left[\frac{Z+(1+\sqrt{2})B}{Z+(1-\sqrt{2})B}\right] \qquad 2.44$$

Where,

f = fugacity

f^L and f^V = liquid and vapour fugacities respectively

If after starting the iteration process presented in Figure there is only one equilibrium phase (Zmin= Zmax), the value for pressure is updated adding a certain amount Δp until the two phases region is reached.

The process updates pressure using Equation 2.45 which was proposed in the GC method of Coniglio et al. (2000):

$$p^{(k+1)} = \left|\frac{p^{(k)}}{Z_{max} - Z_{min}}\left[(Z_{max} - Z_{min}) - (f^V - f^L)\right]\right| \qquad 2.45$$

The saturated liquid density (ρ_L, mol/cm^3) is calculated using Equation 2.46, which is another way of presenting Equation 2.41 and Z is substituted with Zmin:

$$\rho_L = \frac{p}{Z_{min}RT} \qquad 2.46$$

2.6.2. Stability Criteria

Stability criteria and the tangent plane criterion have been used widely for phase equilibrium calculations (Michelsen, 1993). Algorithms have been developed to test whether results from flash calculations correspond to a globally stable system, and to introduce additional phases in the event that the system obtained is not stable. In addition, the tangent plane criterion can be used in the calculation of saturation pressures or temperatures for homogeneous phases (Nghiem et al., 1984). This section introduces the conditions for thermodynamic equilibrium and the related stability criteria. In section 2.6.3 the tangent plane criterion and how it can be applied to saturation point calculations are described.

Gibbs in one of his famous papers ("On the Equilibrium of Heterogeneous Substances", 1876) showed the conditions necessary for a homogeneous system to be at equilibrium and stable:

$$\delta U + p\delta V - T\delta S - \sum_i^n \mu_i \delta m_i = 0 \qquad \text{(equilibrium)} \qquad 2.47$$

$$\Delta U + p\Delta V - T\Delta S - \sum_i^n \mu_i \Delta m_i > 0 \qquad \text{(stability)} \qquad 2.48$$

Where, U, V, S, and m_i denote, respectively, total internal energy, volume, entropy, and mass of components i and T, p and μ_i represent temperature, pressure and potential of component i. In Equation 2.47, δ denotes infinitesimal variations while Δ represent finite variation of the variables.

Introducing the definition of the Gibbs free energy (Equation 2.49) into equations 2.47 and 2.48, and considering a closed system at constant temperature and pressure, the stability criteria can be expressed as follows (Cartlidge C. R., 1997):

$$G \equiv U + pV - TS \tag{2.49}$$

$$\delta G = 0 \quad \text{(equilibrium)} \tag{2.50}$$

$$\Delta G > 0 \quad \text{(stability)} \tag{2.51}$$

From Equation 2.51, for an equilibrium state to be stable, it must correspond to a global minimum in the Gibbs free energy of the system.

Based on Equations 2.50 and 2.51 three equilibrium states can be defined:

- *Metastable State:* the system fulfills the equilibrium condition but not the stability requirement. Therefore, it is stable with respect to its adjacent states and unstable with respect to a distant state.
- *Stable State:* the system is stable with respect to both adjacent and distant states and any variation in the state of the system results in a positive variation in the Gibbs free energy.
- *Unstable State:* The equilibrium state is unstable with respect to any adjacent state.

2.6.3. The Tangent Plane Criterion

Based on the stability criteria presented in the previous section, the tangent plane criterion analyses the Gibbs free energy surface and tangents to minimize the total Gibbs free energy of a system (stable equilibrium) and then determine the number of phases present and their compositions given an overall composition, temperature and pressure.

The concept of the tangent plane criterion for a binary mixture is presented in Figure 2.2 (Cartlidge C. R., 1997). The diagram shows the Gibbs free energy versus composition for a binary system A + B at a given temperature and pressure. The curve is obtained calculating the Gibbs free energy of a hypothetical single phase system at all compositions. The three convex globes of the G curve represent possible equilibrium phases.

A tangent to the G curve, with two points of contact, represents two equilibrium phases. In this case the two phase systems are denoted with G_2 and G_2' (which lies on the tangent line on the feed composition), in both cases the Gibbs free energy G is less than the Gibbs free energy of the homogeneous phase, which means that the two phases system is more stable than the single state. However, the second tangent with G_2 has a lower value for the Gibbs free energy ($G_2 < G_2$'), consequently, G_2 represents the stable solution and G_2' the unstable solution. Once the tangent with the lowest Gibbs free energy for the mixture is identified, the composition and fraction of the two equilibrium phases can be obtained.

For multiphase or multicomponent mixtures, application of the tangent plane criterion becomes more abstract and a numerical method is required. It is important to recall that even binary mixtures can have up to four phases in equilibrium (L_1L_2VS). Methods for the solution of this problem have been developed and are presented in section 2.6.4.

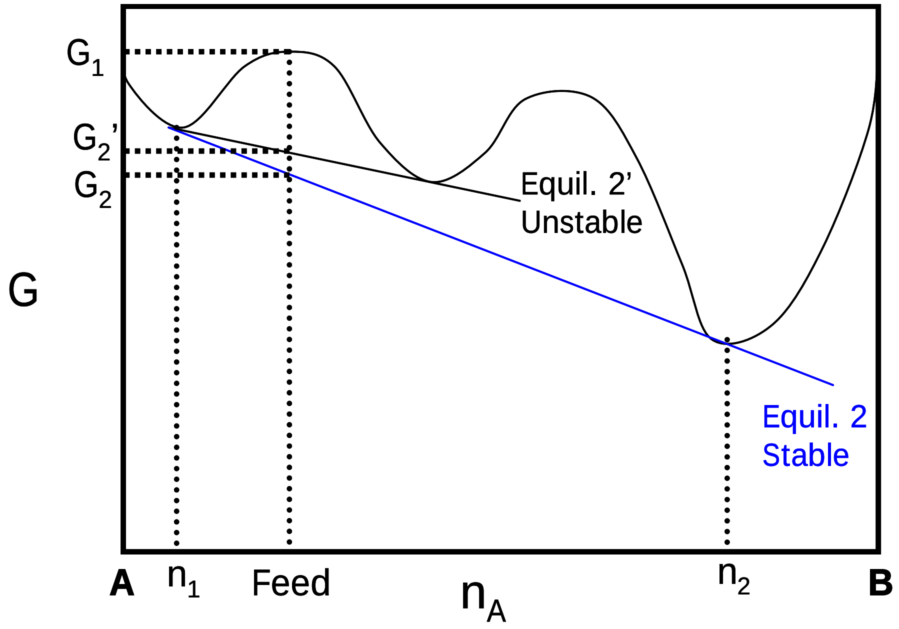


Figure 2.2- Tangent Plane Criterion for a Binary Mixture (Cartlidge C. R., 1997)

2.6.4. Numerical Solutions in Equilibrium Calculations

When using the tangent plane criterion in saturation points calculations a numerical method is required. The method should be reliable and robust, using the minimum iterations steps and arriving to the right solution. Nghiem et al. (1985) proposed and compared a series of numerical methods to solve the tangent plane criterion problem. After comparing the performance of different methods, he suggested the use of the Quasi Newton Successive Substitution method (QNSS) in the case of saturation point calculations. This method will be applied in the present work for the GC multicomponent model and will be described next. Figure 2.3 presents the solution scheme for vapour pressures using the tangent plane criterion and the QNSS method.

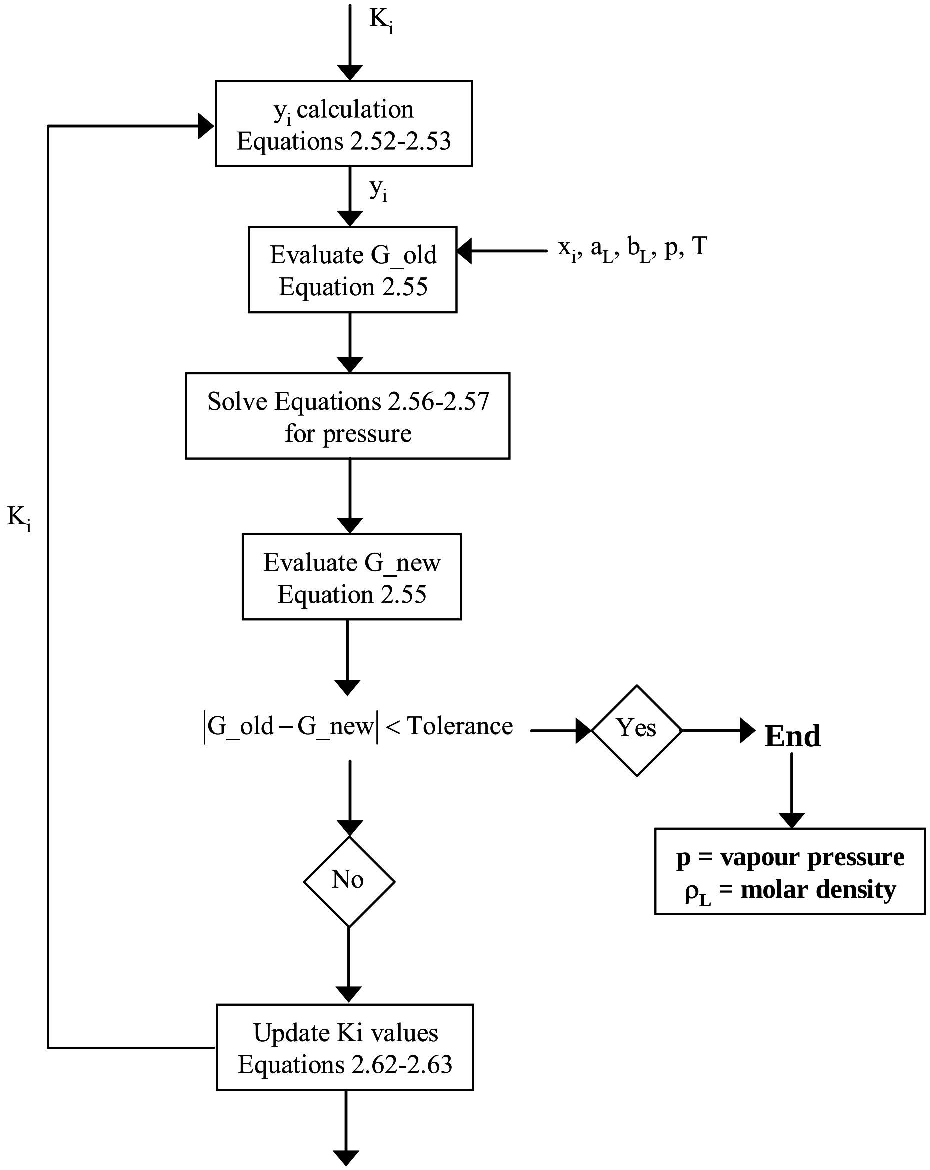


Figure 2.3- Solution Scheme for Vapour Pressures. Multicomponent Model (Nghiem et al., 1985)

The general procedure proposed by Nghiem et al. (1985) with the QNSS algorithm can be explained al follows:

1. Start with first guesses for p and K_i
2. Evaluate the stationary point equation for g_i (Equation 2.54)

3. Find saturation points where the distance from the tangent plane to the Gibbs free energy surface is zero (Solve for pressure) at constant compositions (Equation 2.56-2.57). Usually, Newton method is applied.
4. Evaluate new value of g_i (Equation 2.54)
5. Compare g_i from step 2 with g_i from step 4. If the result is less than the tolerance the iterative process ends.
6. Update K_i values following the QNSS updating algorithm (Equations 2.62-2.63)
7. Repeat steps 2-6 until convergence is reached

The Equations involved in the process are presented next:

- *Equations to calculate y_i from the equilibrium K_i values (Nghiem et al., 1985):*

$$Y_i = K_i x_i \qquad i=1,\ldots,n_c \tag{2.52}$$

$$y_i = \frac{Y_i}{\sum_{j=1}^{nc} Y_j} \qquad i=1,\ldots,nc \tag{2.53}$$

- *Evaluation of the variable g_i (stationary point condition) (Nghiem et al., 1985):*

$$g_i \equiv \ln K_i + \ln\varphi_i(\overline{y},p,T) - \ln\varphi_i(\overline{x},p,T) = 0 \qquad i=1,\ldots,n_c \tag{2.54}$$

Where,

φ_i = Fugacity coefficient of component i (defined in Equations 3.21-3.22)

$\overline{y}$ = Molar composition of the vapour phase: $[y_1,\ldots, y_{nc}]$

$\overline{x}$ = Molar composition of the liquid phase: $[x_1,\ldots, x_{nc}]$

In order to check the convergence of Equation 3.18 to zero, the norm of the vector $[\bar{g}]$ is calculated:

$$G\_ = \mathrm{norm}(\bar{g}) \qquad 2.55$$

Where,

$\bar{g} = [g_1, \ldots, g_{nc}]$

- *Evaluation of the normalized distance from the tangent plane to the Gibbs free energy surface (Nghiem et al., 1985):*

$$D_x^*(\bar{y}, p, T) = \frac{D_x(\bar{y}, p, T)}{RT} \qquad 2.56$$

$$D_x(\bar{y}, p, T) = RT\sum_{i=1}^{nc} y_i \ln\left[\frac{f_i(\bar{y}, p, T)}{f_i(\bar{x}, p, T)}\right] = 0 \qquad 2.57$$

Where,

D_x^* = normalized distance from the tangent plane to the Gibbs free energy surface

D_x = distance from the tangent plane to the Gibbs free energy surface

- *Calculation of Fugacities (Cartlidge, 1997):*

$$\ln\left(\frac{f_i}{x_i p}\right) = \frac{b_i}{b}(Z-1) - \ln(Z-B) - \left(\frac{A}{2\sqrt{2}B}\right)\left(\frac{2}{a}\sum_{j=1}^{nc} x_j (a_i a_j)^{1/2}(1-k_{ij}) - \frac{b_i}{b}\right)\ln\left[\frac{Z+(1+\sqrt{2})B}{Z+(1-\sqrt{2})B}\right] \qquad 2.58$$

$$\varphi_i = \frac{f_i}{x_i p} \qquad 2.59$$

Where,

f_i = fugacity of component I (bar)

A, B = defined in Equations 3.6 and 3.7 respectively

a_i, b_i = parameters of Peng-Robinson EOS for the component i

a, b = parameters of Peng-Robinson EOS for the mixture

Equation 2.58 can be applied to both phases, substituting the corresponding compositions x_i or y_i and the specific compressibility factor, Z. When the EOS has more than one root for Z, the root which gives the lowest Gibbs free energy is selected (computed from Equation 2.60).

$$\frac{\left(G - G^{\text{ideal}}\right)}{RT} = Z - 1 - \ln(Z - B) - \frac{A}{B2\sqrt{2}} \ln\left[\frac{Z + (1+\sqrt{2})B}{Z + (1-\sqrt{2})B}\right] \qquad 2.60$$

Where,

G = Gibbs free energy (J)

G^{ideal} = Gibbs free energy in the ideal state (J)

- *QNSS updating algorithm:* The method proposed by Nghiem et al. (1985) to solve the tangent plane criterion in vapour pressure calculations, employs an accelerated successive substitution algorithm, the QNSS algorithm, for updating the equilibrium constants K_i. The suggested equations are:

$$\bar{\xi} \equiv (\ln K_i, \cdots, \ln K_{n_c}) \qquad 2.61$$

$$\Delta\xi = -\sigma^{(k)} \bar{g}^{(k)} \qquad 2.62$$

And σ is a scalar given by:

$$\sigma^{(k)} = \frac{\Delta\bar{\xi}^{(k-1)} \bar{g}^{(k-1)}}{\Delta\bar{\xi}^{(k-1)} (\bar{g}^{(k)} - \bar{g}^{(k-1)})} \sigma^{(k-1)} \qquad 2.63$$

The process is initialized with $\sigma^{(0)} = 1$ and the step size is limited to $\left|\sigma^{(k)}\right| \leq 30$.

The QNSS method is started with a first guess for pressure and the equilibrium K_i values (or with vapour copositions). These two variables and their first estimates play an important role in the convergence of the GC model and, only good estimates for them can improve the convergence to the right solution. That is the reason for selecting the stability test calculation (Michelsen, 1982) as the algorithm to estimate this first guesses of p and K_i.

The stability test calculation refers to the model created to estimate the first guesses for pressure and vapour compositions that are required as an input for the GC multicomponent model.

This method is based also in the tangent plane criterion and can be divided into two main parts. The first one that estimates the equilibrium values K_i and the second one that estimates pressure. Each part is described next.

1. *Estimation of K_i values:* To estimate first guess values for the equilibrium K_i values a stability test calculation is computed, which is a method proposed by Michelsen (1982). It takes as an input a specific value for pressure and temperature; these values do not need to be close to the phase boundary.

   This process starts with the stability criterion presented in Equations 2.56-2.57 with the distance from the tangent plane to the Gibbs free energy surface equal to zero. Rearranging Equation 2.57 (Michelsen, 1982):

$$D_x^*(\overline{y}, p, T) = \sum_{i=1}^{nc} y_i(\ln y_i + \ln\varphi_i - h_i) = 0 \qquad 2.64$$

Where,

$$h_i = \ln x_i + \ln \varphi_i(\bar{x}) \qquad 2.65$$

The stationary criterion is:

$$k_i = \ln y_i + \ln \varphi_i - h_i \quad i=(1,\ldots, n_c) \qquad 2.66$$

Where,

k = dimensionless chemical potential difference

Defining a new variable:

$$Y_i = \exp(-k_i) \qquad 2.67$$

The stationary criterion becomes:

$$\ln Y_i + \ln \varphi_i - h_i = 0 \quad i=(1,\ldots, n_c) \qquad 2.68$$

Y_i can be interpreted as mole numbers and have the same definition presented in Equation 2.53. In Addition:

$$Y_i = K_i x_i \qquad 2.69$$

Solutions to Equation 2.68 are the stationary points; stability is verified provided that $k \geq 0$, corresponding to $\sum_i Y_i \leq 1$. The solution can be obtained with direct substitution and subsequent iterates are determined by:

$$\ln Y_i^{(t+1)} = h_i - \ln \varphi_i^{(t)} \qquad 2.70$$

This method can be started with any values of K_i. However, for hydrocarbon systems a good approximation would be (Michelsen, 1982):

$$K_i = \frac{p_{ci}}{p} \exp\left[5.42\left(1 - \frac{T_{ci}}{T}\right)\right] \qquad 2.71$$

2. *Estimation of pressure:* The first guess for pressure is obtained from the proposal developed by Michelsen (1985). It takes the K_i values estimated from the stability test calculation at a specified pressure and temperature. Then two sets of $Y_i^{(0)}$ values are defined:

$$Y_{i,1}^{(0)} = K_i^{(0)} x_i \tag{2.72}$$

$$Y_{i,2}^{(0)} = \frac{1}{K_i^{(0)}} x_i \tag{2.73}$$

When the saturation temperature is specified, Equation 2.57 can be solved to get the pressure where the distance from the tangent plane to the Gibbs free energy surface is zero. It is a one dimensional search and is computed with the two different sets of Y_i defined in Equations 2.72-2.73.

As a result two different pressures are obtained. Any of these pressures can be used as the starting point in the vapour pressure calculation. However, it was observed (Michelsen, 1985) that depending on the starting guess value used, a different solution can be obtained. Some of the solutions may lie on the stability limit and are physically meaningful.

3. METHODOLOGY

The methodologies applied to accomplish the proposed research objectives are presented in this chapter. The organization of the methods based on their type and the order of their application in the study are described. This chapter includes an explanation for how the GC method is implemented as a computational model to evaluate thermophysical properties, the solution algorithms for all the routines in the program, and the performance criteria vapour pressure and density estimates.

3.1. Scope of Project

The key tasks in the present work, are to:

- Build a computational model to estimate thermophysical properties of heavy hydrocarbon compounds using the group contribution method proposed by Coniglio et al..(2000).
- Extend the application of the group contribution method proposed by Coniglio et al. (2000) to multicomponent mixtures and apply it in a computational model.
- Estimate thermophysical properties of Athabasca Vacuum Residue using the group contribution model
- Compare model performance with available experimental vapour pressure and density data for this material.

The GC method proposed by Coniglio et al. (2000) and its update (Crampon et al., 2004), both presented in Chapter 2, were implemented in MATLAB. Implementation of the GC method requires the selection of solution algorithms and numerical methods to solve the set of equations. These methods are presented in detail in Section 3.2.

Once the GC model was completed, it was necessary to test it and tune it as required. A series of bubble pressure and density calculations for known compounds and mixtures were performed using the GC model and the results were compared with available experimental data. The compounds were selected based on their similarity with the type of hydrocarbons encountered in AVR.

With the performance of the GC model tested, it is possible to make bubble pressure and density estimates for AVR. The inputs for the model include the molecular structure and molar mass which were obtained from previous research (Sheremata, 2005). Experimental bubble pressure and density data for AVR (Zou, 2003) were then compared with predictions. This comparison and analysis is presented in Chapter 4. Figure 3.1 presents a schematic representation of the methodology that is followed in the present study.

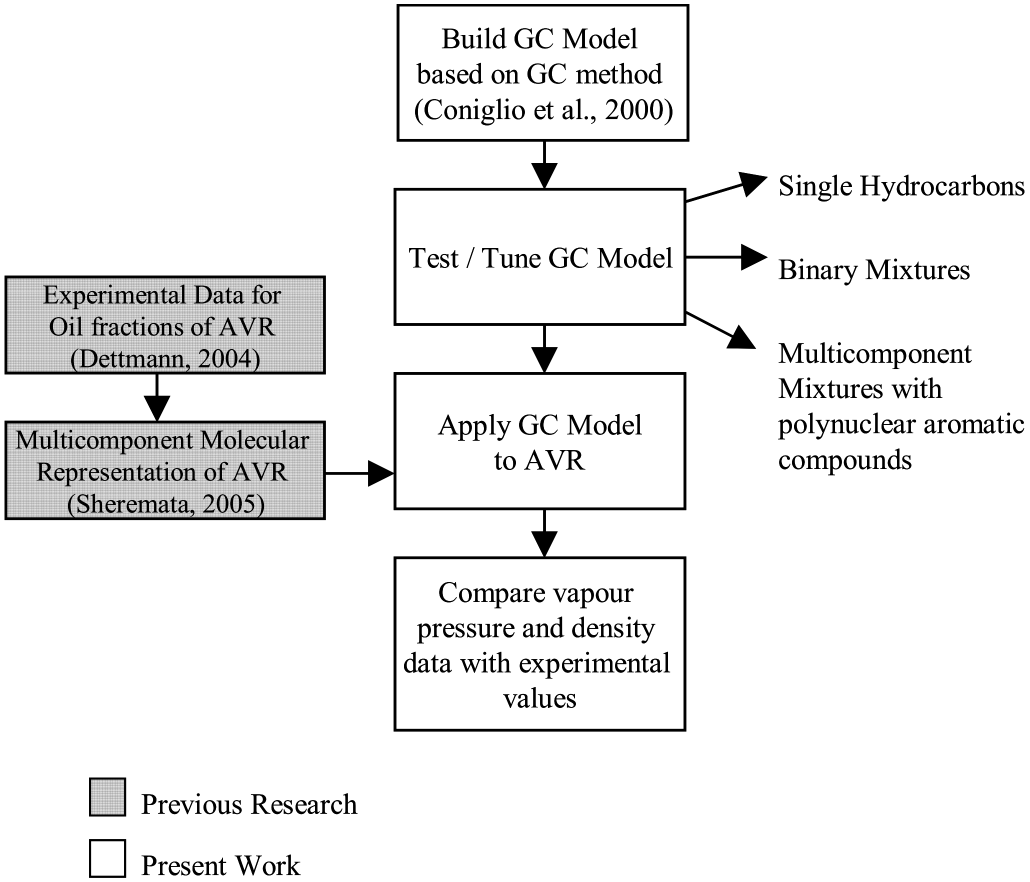


Figure 3.1- Scope of Project

3.2. GC Model Implementation

Key decisions include:

- Selection of the programming language and software
- Organization of the model
- Selection of solution algorithms

MATLAB was selected as the programming languages because the user interface is “friendly” and because arrays do not require dimensioning. This later feature allows matrixes and vectors to be handled in an easy way compared with scalar languages such as C or Fortran. In addition, display of results in matrix form and plots are easily transferable to Excel or any other data processor. MATLAB is also used by numerous groups in the CME department, and it is therefore a suitable platform for further development of the model.

Two complete programs were developed; the first one focused on pure fluids and was a direct application of the GC method. It was a benchmark program for learning and method validation. The second program extended the GC method to multicomponent mixtures through the incorporation of mixing rules. The multicomponent program does estimate properties of pure fluids but the solution algorithms and other programming details differ. In addition to these two main programs a third one was prepared. This latter program provides starting estimates for bubble pressure and vapour compositions, which are part of the input information of the multicomponent program. The organization of the model is presented in Figure 3.2.

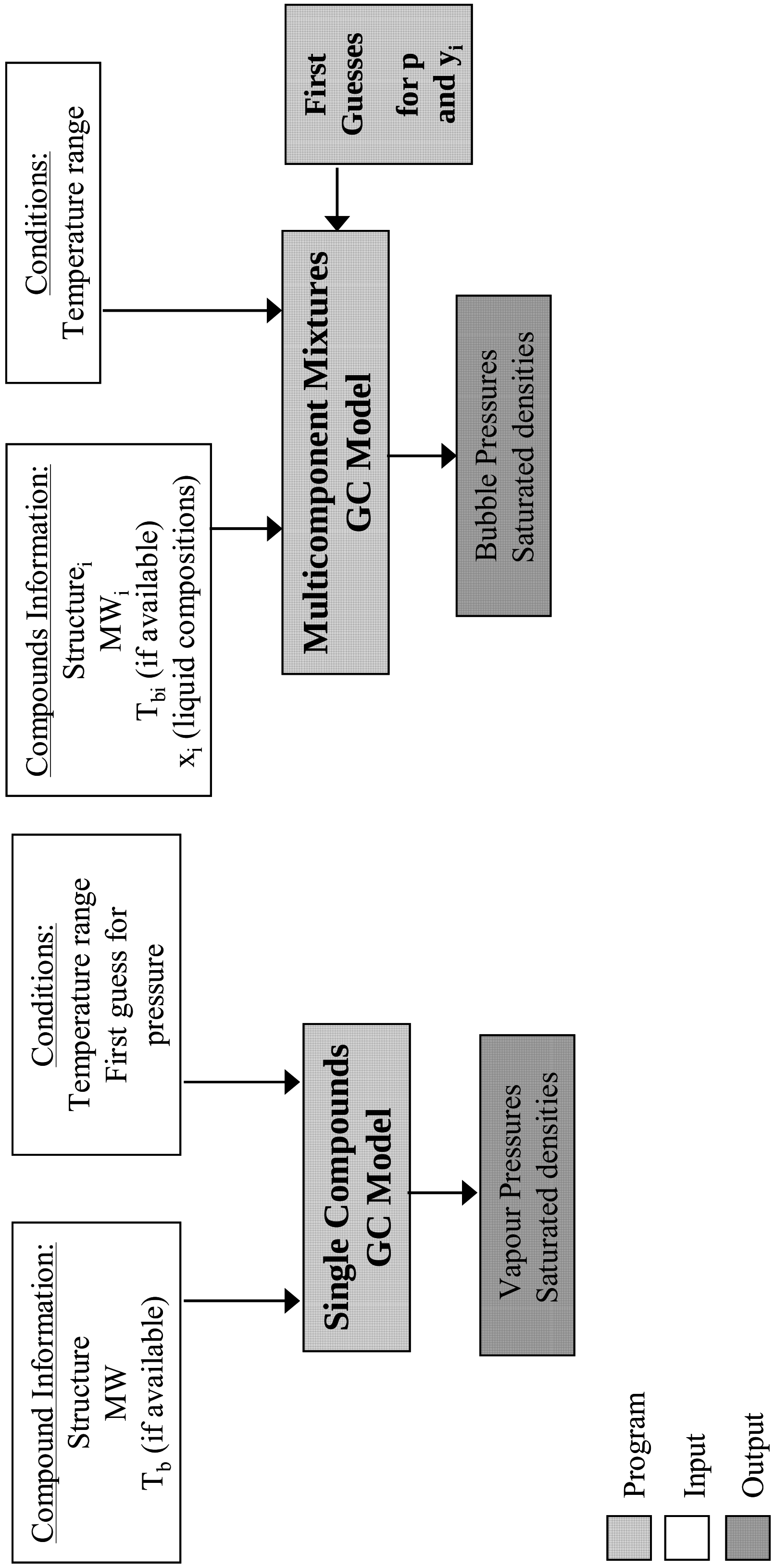


Figure 3.2- GC Model Organization

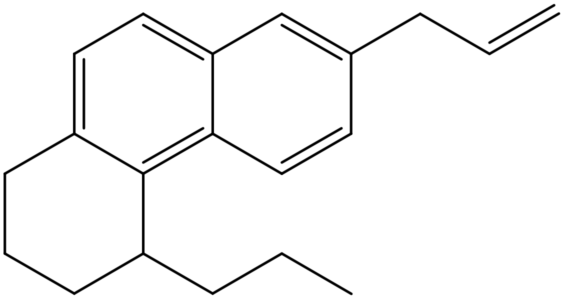
The selection of solution algorithms is made in order to obtain very robust programs in terms of convergence and precision of the solutions. The different solution algorithms are presented in detail in Sections 3.2.2 and 3.2.3. Before addressing these issues, an explanation for how the molecules proposed by Sheremata 2005 are decomposed into the molecular substructures required by the GC method.

3.2.1. Decomposition of a molecule into GC method molecular substructures

The application of the GC method (Coniglio et al., 2000) presented in Chapter 2 requires the molecular substructure of the components present in a particular form. The molecular substructures or groups are described by Crampon et al. (2004) (Table 2.8) and these are used in the equations that estimates the parameters for the Peng-Robinson EOS: a, b and c.

The decomposition process is illustrated for a sample molecule, Figure 3.3, which is decomposed into the molecular groups shown in Table 3.1. The resulting vector, in addition to the composition and molar mass are the only input for GC model calculations related to this specific compound. The form of the vector is:

$$\text{structure } (C20H24) = [1\ 3\ 0\ 0\ 3\ 1\ 0\ 0\ 2\ 5\ 1\ 2\ 1\ 1\ 0\ 0\ 0\ 0\ 0] \qquad 3.1$$



$C_{20}H_{24}$
MW = 264.4

Figure 3.3- Sample Molecule for Decomposition

The decomposition of the molecule is as follows:

Table 3.1- Decomposition into Molecular Groups

# Group	**Molecular Group Name**	**N_j**
	Alkanes:	
1	CH_3	1
2	CH_2	3
3	CH	0
4	C	0
	Naphthenes:	
5	CH_2	3
6	CH	1
7	C	0
8	CH (ring/ring junction)	0
9	C (ring/aromatic junction)	2
	Aromatics:	
10	CH	5
11	C (substituded)	1
12	C (condensed)	2
	Aliphatic Alkenes:	
13	$=CH_2$	1
14	=CH-	1
15	=C<	0

16	=C=	0
	Double bond near aromatic ring:	
17	=CH-aromatic ring	0
	Aliphatic Alkynes:	
18	≡C-	0
19	≡CH	0

N_j represents the number of contributions of the molecular group j.

3.2.2. Algorithm for pure fluids

The Solution algorithm developed to solve for vapour pressures and saturated liquid densities of single compounds follows the GC method presented in Chapter 2. Figure 3.4 shows the general form of the algorithm. Each step is explained in detail in this section.

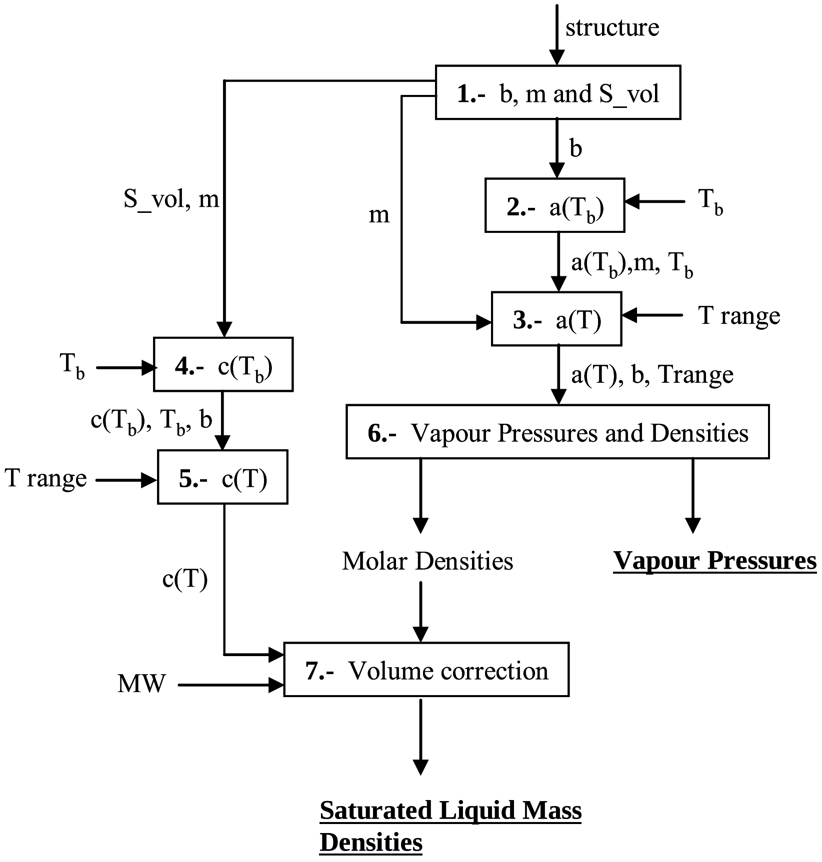


Figure 3.4- Solution Algorithm for Pure Fluids

Following the structure of the GC model presented in Figure 3.4, the procedure to solve for vapour pressures and densities is as follows:

- *Step 1:* The first step takes the structure of the compound as an input and following equations from the GC method (Coniglio et al., 2000) and (Crampon et al., 2003) presented in Chapter 2, the parameters b, m and S_vol are estimated:

  b: Equation 2.31

  m: Equation 2.32 and 2.33

S_vol: Equation 2.26 without incremental corrections and taking the contribution of the 25 molecular groups:

$$S = \sum_{j=1}^{25} C_j N_j \qquad 3.2$$

Note that the values of C_j were taken from the original GC method (Coniglio et al.) and only add volume corrections to 7 of the 25 molecular groups used in the GC model. Results with and without volume correction are presented in Chapter 4.

- *Step 2:* In the second step the parameter $a(T_b)$ at the boiling point temperature is estimated using the algorithm proposed by Coniglio et al. (2000) and presented in Chapter 2, Figure 2.1. The input information is the parameter b and the boiling point temperature of the compound T_b, (if available). The GC model starts an iterative process changing the value for the parameter a (Peng-Robinson EOS) until the vapour pressure matches the boiling point pressure 1.01325 bar. The details for vapour pressure calculation in this step are explained in Figure 3.5. An iterative process is conducted starting with a set value for pressure (usually 1 bar) and updating it using the method of Coniglio et al. (2000) until convergence is obtained.

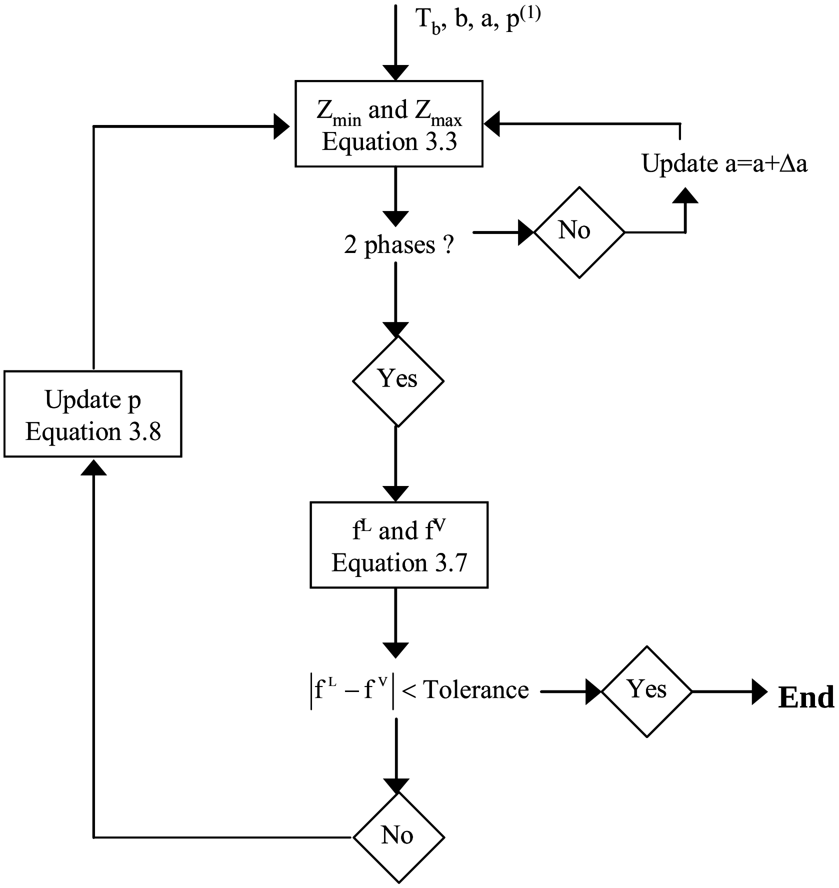


Figure 3.5- Vapour Pressure Estimation for $a(T_b)$

- *Step 3:* This step shown in Figure 3.4 estimates the parameter a for all the values of T that are of interest. It applies Equations 2.15-2.17. The input information are the parameters b, m (from Step 1) and $a(T_b)$ (from Step 2). The calculation is repeated for all temperatures and the output of the calculation is a vector with the values of parameter a at each T.

- *Step 4:* In this step the parameter c at the boiling point temperature ($c(T_b)$) is estimated using Equation 2.25 which is a function of the shape parameter m and S_vol (from Step 1) and the boiling point temperature T_b.

- *Step 5:* The volume correction parameter c is calculated for all T of interest applying Equations 2.22-2.24. The output is a vector with each one of the values for c(T).

- *Step 6:* Vapour pressure and densities estimation. This algorithm is very similar to the vapour pressure estimation used in $a(T_b)$ calculation with the exception of the first cycle necessary when only one equilibrium phase is present. In this case, instead of changing the parameter a, the value for pressure is adjusted until the two phases region is obtained. The process converges when the chemical potential of both phases are equal. At the end, the value for pressure becomes the vapour pressure at the given temperature. This algorithm is repeated for all temperatures of interest, providing a set of vapour pressure points. The saturated liquid density (ρ_L, mol/cm^3) is then calculated using Equation 2.46

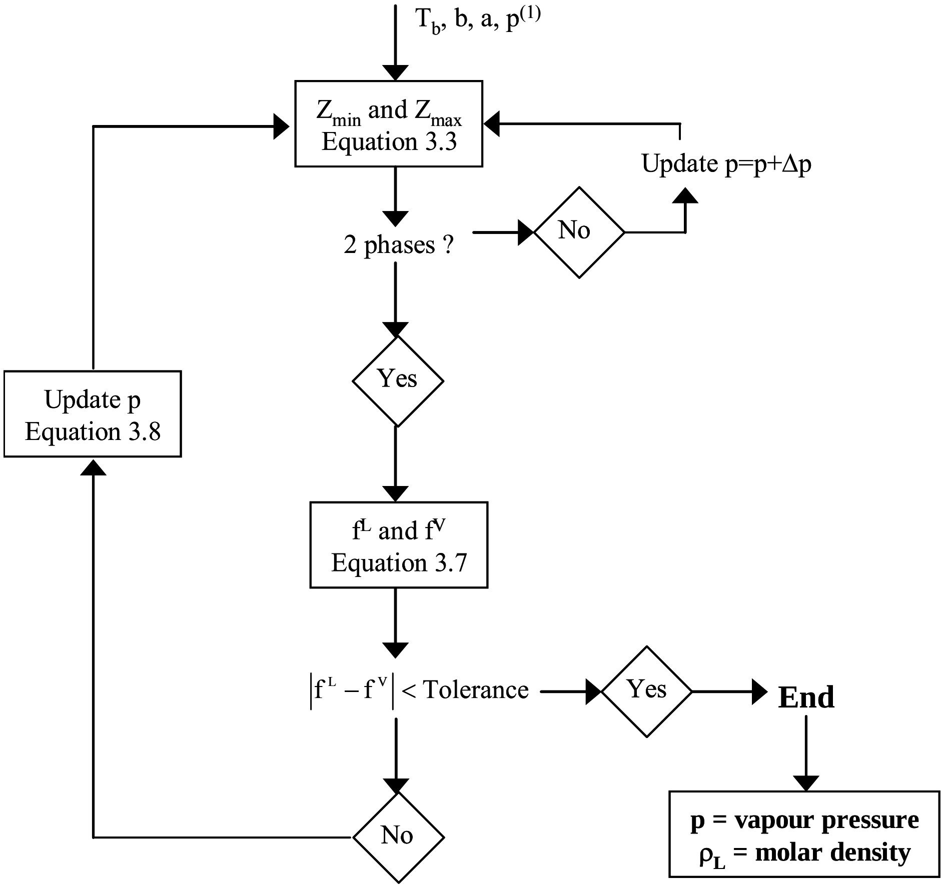


Figure 3.6- Vapour Pressure Calculation for Pure Fluids

- *Step* 7: This step transforms the molar density into mass density. In addition, the correction introduced for molar volume in the GC method (Coniglio et al., 2000) with the parameter c(T) is:

$$v_{corr} = \frac{1}{\rho_L} - c \qquad 3.3$$

$$\rho\ (g/cm^3) = \frac{MW}{v_{corr}} \qquad 3.4$$

Equations 3.3 and 3.4 are applied for each molar density (from Step 6), that means, for all temperatures of interest.

Step 7 finishes the calculation of vapour pressure and density of pure liquids.

This algorithm is then coded as a MATLAB program which will be described in Section 3.2.4.

3.2.3. Algorithm for multicomponent mixtures

The multicomponent mixture GC model requires more complicated solution algorithms compared with the single compound model. The main reason is the increment of the variables involved in the calculations. The scheme for the solution algorithm is shown in Figure 3.7 and each step is described below.

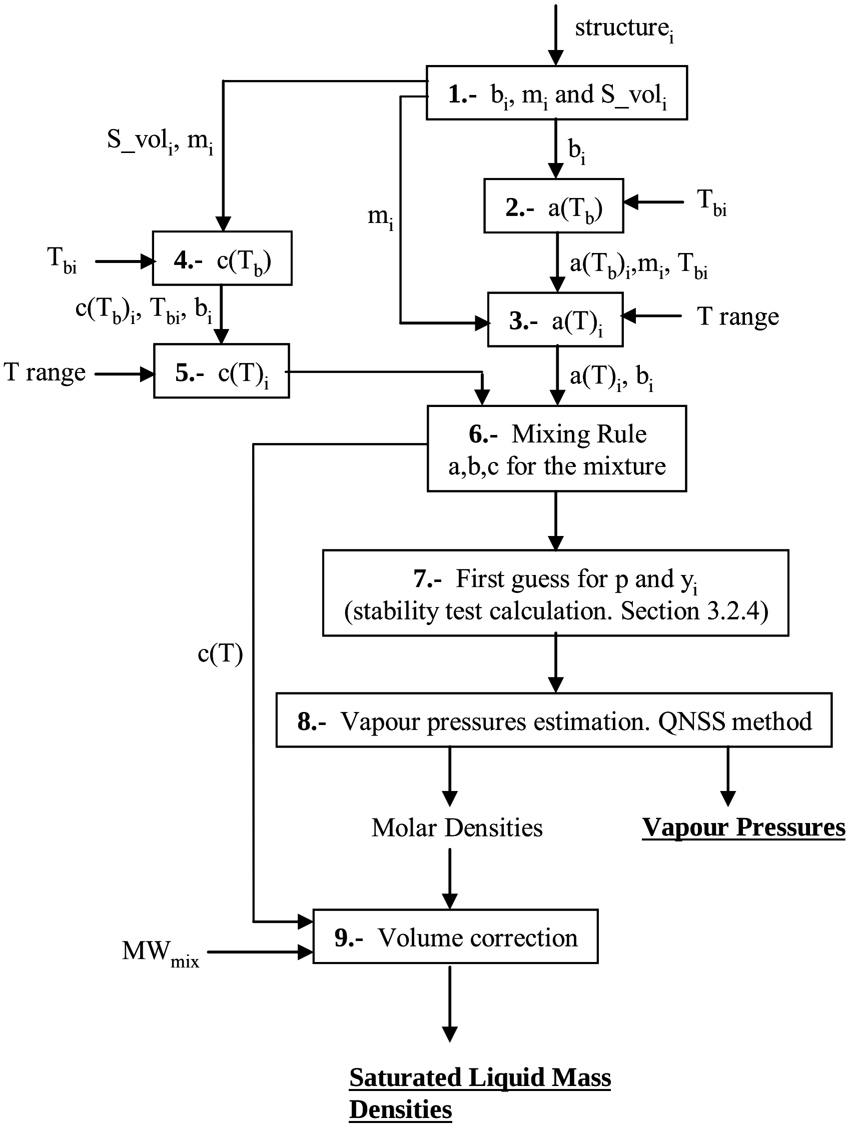


Figure 3.7- Solution Algorithm for Multicomponent Mixtures

- *Steps 1-5:*_The first five steps in the multicomponent solution algorithm estimate the parameters a, b and c for each one of the components of the mixture. The input information includes the boiling point temperature (if available) T_{bi} and the decomposed molecular structure (structure$_i$). The method to calculate the EOS parameters is the same as in the pure fluid case, described above. The only difference is that the calculation is repeated n_c

times, where n_c represents the number of components in the mixture. In addition, temperature dependent parameters c and a, must be obtained at all temperatures of interest, n. Equations 3.5-3.7 present the form of the parameters estimated after Step 5:

$$a = \begin{bmatrix} a_{11} & a_{12} \ldots a_{1nc} \\ a_{21} & a_{22} \ldots a_{2nc} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \cdots a_{nn_c} \end{bmatrix} \quad 3.5$$

$$b = [b_1, b_2, \ldots, b_{nc}] \quad 3.6$$

$$c = \begin{bmatrix} c_{11} & c_{12} \ldots c_{1nc} \\ c_{21} & c_{22} \ldots c_{2nc} \\ \vdots & \vdots \\ c_{n1} & c_{n2} \cdots c_{nn_c} \end{bmatrix} \quad 3.7$$

- *Step 6:* This step applies a mixing rule to the parameters and calculates the parameters for the liquid mixture. The input information is the value of the parameters for each component (From Step 5) and the composition of the liquid mixture (x_i). The mixing rule used in all calculations is the classical quadratic mixing rule described in Chapter 2, Equations 2.36-2.38. For parameter c, the same linear mixing rule applied for parameter b is applied.

- *Step 7:* This step is very important during the calculation process because it determines the first values for pressure and vapour compositions (or equilibrium K_i values). The solution algorithm selected was the stability test calculation (Michelsen, 1982) and it was described in Chapter 2. The general scheme is presented in Figure 3.8.

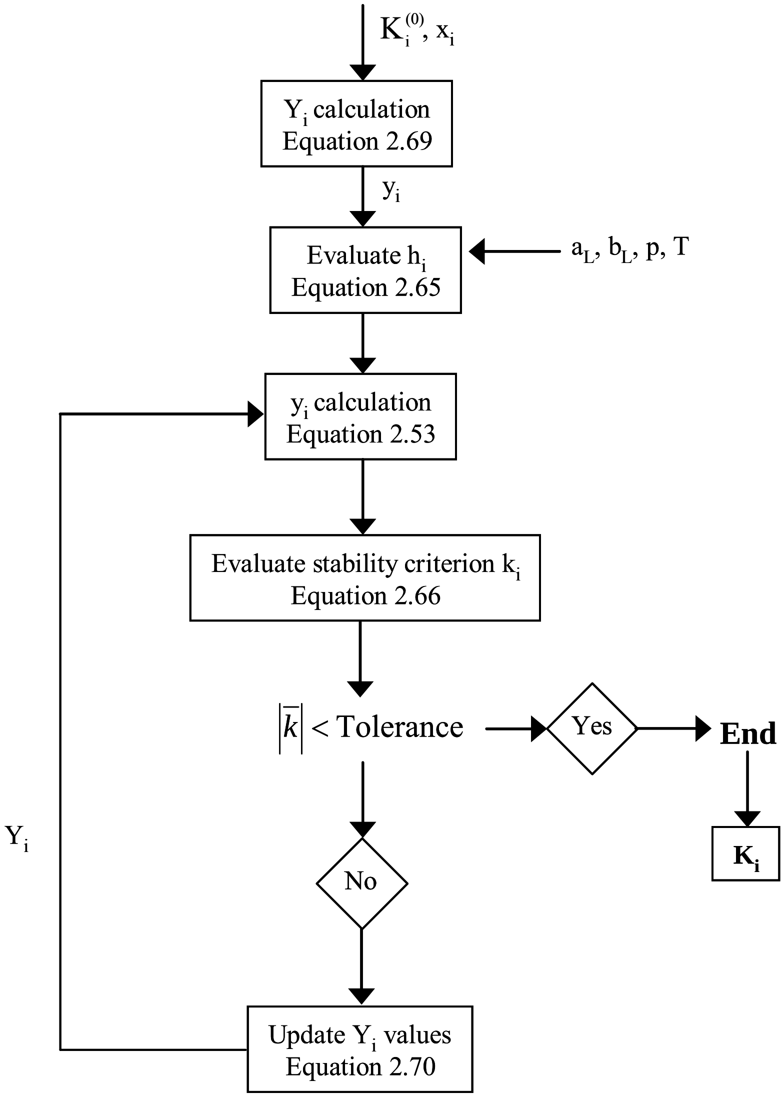


Figure 3.8- Stability Test Calculation Scheme

- *Step 8:* This step takes as input, the EOS parameters for the liquid mixture and components at all temperatures (a(T), b, a_i, b_i) and the temperature range of interest. The first guesses for pressure and equilibrium K_i values are an input as well, and are estimated in step 7. The algorithm selected to solve for pressures and vapour compositions is the QNSS method presented in Chapter 2. The results for vapour pressure, density and vapour compositions are displayed in a matrix form:

$$p = [p_1, \dots, p_n] \qquad 3.8$$

$$\rho = [\rho_1, \dots, \rho_n] \qquad 3.9$$

$$y = \begin{bmatrix} y_{11} y_{12} \cdots y_{1n_c} \\ \vdots \qquad \vdots \\ y_{n1} y_{n2} \cdots y_{nn_c} \end{bmatrix} \qquad 3.10$$

The algorithm for this step of the calculation is shown in Figure 2.3.

3.2.4. MATLAB as the programming tool

MATLAB is a software package developed by Mathworks Inc. The name MATLAB stands for matrix laboratory. It was originally written to provide easy access to matrix software. It is a high-performance language for technical computing. It integrates computation, visualization, and programming in a friendly environment. Normal applications include algorithm development, data analysis, modeling, simulation and visualization of scientific and engineering graphics. The greatest advantage of programming with MATLAB is that arrays do not require dimensioning. This feature allows solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The MATLAB system consists of five main parts (Mathworks Inc., 2004):

- Development Environment. These tools are graphical user interfaces that help use MATLAB functions and files.

- The MATLAB Mathematical Function Library. This is a vast collection of computational algorithms ranging from elementary to more sophisticated functions.
- The MATLAB Language. This is a high-level matrix/array language with control flow statements, functions, input/output, and object-oriented programming features.
- Graphics. MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs.
- The MATLAB Application Program Interface (API). This is a library that allows you to write C and Fortran programs that interact with MATLAB.

As part of MATLAB programming tool, functions can be created to form in conjunction a whole program. This functions are files that contain code in the MATLAB language and are called M-files. They are created using a text editor. The names of the M-file and of the function should be the same. Functions operate on variables within their own workspace.

In implementing the GC based EOS (Coniglio et al., 2000) functions were created for the algorithms presented in the previous sections. These functions are presented below and the complete MATLAB code comprises Appendix I.

1. *Pure fluid GC model:* the complete list of functions created in MATLAB to evaluate vapour pressure and densities of single compounds is presented

in Table 3.2. The input and output variables are listed as well. The mainprogram function evaluates the parameter $a(T_b)$ and calls the rest of the functions to estimate vapour pressures and densities.

Table 3.2- MATLAB Functions for the Single Compounds GC Model

Function Name	Input	Output
mainprogram	structure, T_b, MW	T, p, density
roots_1	T, p, R, a, b	Zmax, Zmin, nph
fugacity	T, a, b, Zmin, Zmax, p,	fugcoeff_l, fugcoeff_v
EOS_parameters	structure	S_vol, b, m
a_estimation	$a(T_b)$, m, T, T_b	a(T)
vapress	T, p, R, a(T), b	p, v
density	c_T, T, p, R, v, MW	density
CT	m, T_b, S_vol, T, b	c(T)

where,

nph = number of phases

fugcoeff_l = fugacity coefficient for the liquid phase

fugcoeff_v = fugacity coefficient for the vapour phase

2. *Multicomponent GC Model:* The complete list of variables and functions for the GC multicomponent model is presented in Table 3.3. The model follows the algorithm presented in Figure 3.7. There are two versions of the program, one that estimates the boiling point temperatures, T_{bi}, of the components using the GC method (Coniglio et al., 2001) and another one that takes this information as an input. The list of functions is the same in

both cases; therefore, only one list of functions is shown, Table 3.3 presents the version that estimates T_b. The first estimates for pressure and K_i values are estimated separately – bullet 3 below. The main function is named mixtures. It comprises the rest of the functions as a program. The functions EOS_parameters, a_tbf, a_estimation and fugacity are programs that evaluate properties of the pure components such as the pure component parameters a_i and b_i. On the other hand, fugacity_l, fugacity_v and parameter_a_b calculate properties of the mixture. These fugacity functions evaluate the fugacity coefficient of each component in the mixture while parameter_a_b evaluates the mixing rule to compute the parameters a and b of the mixture. The function Eval_f computes the stability criterion (Equation 2.57) while Eval_G evaluates the stationary point condition for each composition g_i (Equation 2.54).

Table 3.3- MATLAB Functions for the Multicomponent GC Model

Function Name	**Input**	**Output**
mixtures	structure, MW, T, x, k_{ij}	p, y, density
EOS_parameters	structure	S_vol, b, m, T_b
roots_1	T, p, R, a, b	Zmax, Zmin, nph
a_tbf	structure, T_b, b	$a(T_b)$
a_estimation	$a(T_b)$, m, T, T_b	a(T)
fugacity	T, a, b, Zmin, Zmax, p, R	fugcoeff_l, fugcoeff_v
fugacity_l	T, a_L, b_L, Zmin, p, R, x, a_i, b_i, k_{ij}	fugcoeff_l
fugacity_v	T, a_v, b_v, Zmax, p, R, x, a_i, b_i, k_{ij}	fugcoeff_v
parameter_a_b	x_vector, a_i, b_i, k_{ij}	a, b
vapress	m, T_b, T, R, n_c, x, a_i, b_i, k_{ij}	p, y, v
Eval_f	p	f
Eval_G	x, y, T, p, R, a_i, b_i, K, k_{ij}	G_
CT	m, T_b, S_vol, T, b	c(T)

where,

a_L, b_L = parameters of the GC-EOS for the liquid phase

a_v, b_v = parameters of the GC-EOS for the vapour phase

x_vector = vector of molar compositions. Could be liquid, x or vapour, y

$f = RT\sum_{i=1}^{nc} y_i \ln\left[\frac{f_i(\bar{y},p,T)}{f_i(\bar{x},p,T)}\right]$. Stability criterion (Equation 2.57)

G_ = norm ($\bar{g}$). Stationary point condition (Equation 2.55)

3. *First Estimates for p and K_i:* First guesses for pressure and equilibrium values K_i are evaluated using the stability test calculation presented in Chapter 2. The complete list of functions for this program is shown in

Table 3.4. This version of the program estimates T_b using the GC method (Coniglio et al., 2000).

Table 3.4- MATLAB Functions for the Stability Test Calculation

Function Name	Input	Output
Stability_test	structure, MW, T, p, x, k_{ij}	K, graphic_D
EOS_parameters	structure	S_vol, b, m, T_b
a_tbf	structure, T_b, b	$a(T_b)$
a_estimation	$a(T_b)$, m, T, T_b	a(T)
parameter_a_b	x_vector, a_i, b_i, k_{ij}	a, b
roots_1	T, p, R, a, b	Zmax, Zmin, nph
fugacity_l	T, a_L, b_L, Zmin, p, R, x, a_i, b_i, k_{ij}	fugcoeff_l
fugacity_v	T, a_v, b_v, Zmax, p, R, x, a_i, b_i, k_{ij}	fugcoeff_v

where,

K = vector of K_i values

Graphic_D = plot of D_x^* vs pressure (D_x^* from Equation 2.56)

Results in matrix form or as plots are easily transferable to Excel or any other data processor.

3.3. Comparison with Experimental Data

In order to test the model before applying it to AVR, test calculations were performed with sample compounds and mixtures with known properties.

The test compounds and mixtures were selected considering their affinity with the type of components present in AVR; some of the characteristics considered are listed below:

- Molecular structure (hydrocarbon compounds)
- Molar mass (high molecular weight)
- Asymmetric hydrocarbon mixtures
- The test compounds and mixtures include: toluene, cyclooctane, pyrene, eicosane and mixtures such as benzene-ethylbenzene, hexane–hexadecane and methane-cyclooctane-decane-naphthalene.

In the comparison of thermophysical properties of AVR with experimental data, only vapour pressure and saturated liquid density experimental data is available, consequently, the comparison and analysis of results focuses on these two properties.

4. RESULTS AND DISCUSSION

Section 4.1 presents a general description of the GC computational model. Sections 4.2 and 4.3 present the results of the GC model when applied to the estimation of thermophysical properties of pure fluids and mixtures respectively. Finally, results and discussion of the application of the GC model to vapour pressure and density calculations of AVR is split into two parts: the first one models AVR with a single molecule representation (Section 4.2.2); the second uses different multi-molecule representations for this hydrocarbon fraction (Section 4.4.1).

4.1. GC Computational Model

The GC computational model was implemented using the MATLAB programming language and follows the algorithms explained in Chapter 3. Four different options are available:

1. *Pure fluids:* vapour pressure and saturated liquid density estimation.
2. *Multicomponent mixtures with known T_b for the components:* bubble pressure, saturated liquid density and vapour composition estimation.
3. *Multicomponent mixtures with unknown T_b for the components:* same characteristics as the program 2, but the boiling point temperature of the components T_{bi} are estimated (Coniglio et al., 2001).

4. *Initial estimates for bubble pressure and K_i values for mixtures:* application of the stability test (Michelsen, 1982) to estimate starting values for vapour pressure and equilibrium K_i values in a mixture.

The complete code for all the programs is included in Appendix I and a digital version, which can be run using MATLAB 7, is included in the attached CD. In addition, the procedure on how to run the programs is presented in Appendix II.

4.2. Application of the GC Model to Pure Fluids

In order to test the performance of the GC model for equilibrium calculations, experimental phase equilibrium data from the literature were compared with predicted values for phase density and vapour pressure. Application of the pure fluid GC model to AVR using a single molecule representation for this hydrocarbon fraction is also included.

4.2.1. Defined hydrocarbons

Hydrocarbon compounds with available vapour pressure and density data with different type of molecular structure were selected to test the performance of the GC model as follows:

- one aromatic compound: toluene
- one naphthene: cyclooctane
- one polynuclear aromatic: pyrene
- one long chain alkane: eicosane

Figure 4.1 presents the molecular structure of all the test compounds.

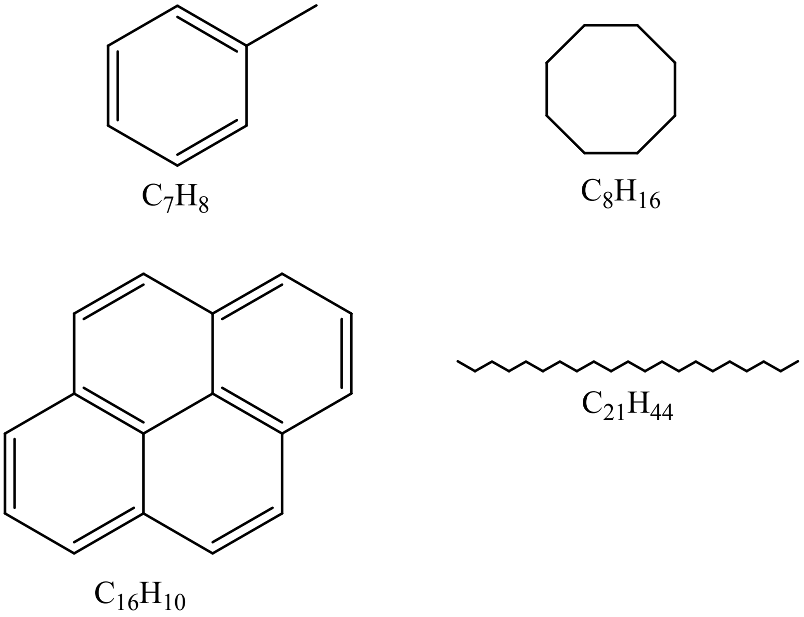


Figure 4.1- Molecular Structures of Test Compounds

The physical properties necessary as an input for the GC model are presented in Table 4.1. It can be observed that the molar mass of the test compounds ranges from 92.141-282.553. These compounds are small but possess molecular structures present in AVR. The structures for the components were obtained following the procedure presented in Chapter 3.

Table 4.1- Physical Properties of Test Single Compounds

Compound	**Formula**	**MW$^{(*)}$**	**T_b (K)$^{(*)}$**	**GC Structure**
toluene	C_7H_8	92.141	383.78	[1 0 0 0 0 0 0 0 0 5 1 0 0 0 0 0 0 0 0]
pyrene	$C_{16}H_{10}$	202.255	667.95	[0 0 0 0 0 0 0 0 0 10 0 6 0 0 0 0 0 0 0]
eicosane	$C_{20}H_{42}$	282.553	616.93	[2 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
cyclooctane	C_8H_{16}	112.215	424.30	[0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

(*) MW and T_b were obtain from Yaws (2003)

The predicted vapour pressures for the test compounds are compared with experimental data in Figure 4.2 and Figure 4.3. Good agreement between the GC model experimental data (Yaws, 2003) is observed. The GC model results were also compared with those obtained using the PR EOS where the parameters a and b were evaluated using the traditional approach (Equations 2.6-2.10). Figure 4.4 shows these results for the four test compounds. The two sets of predictions are in close agreement.

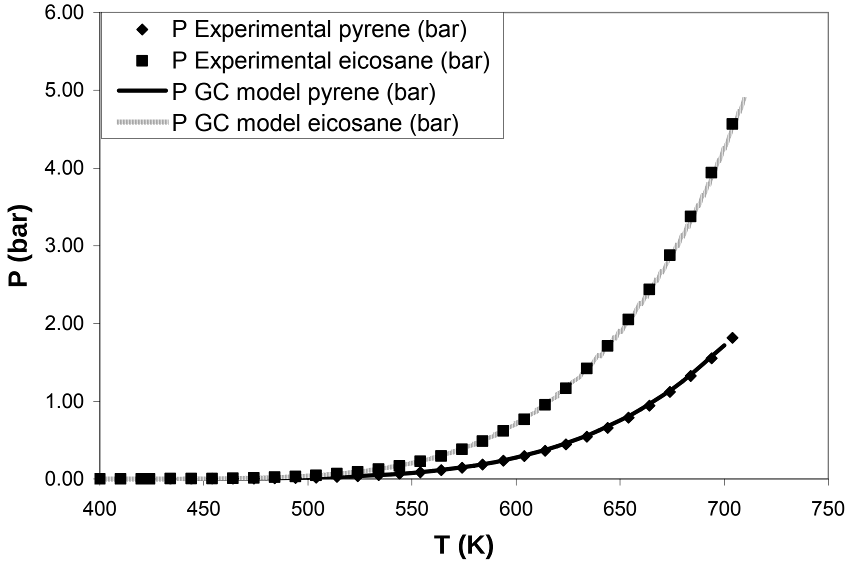


Figure 4.2- Vapour Pressure for Pyrene and Eicosane

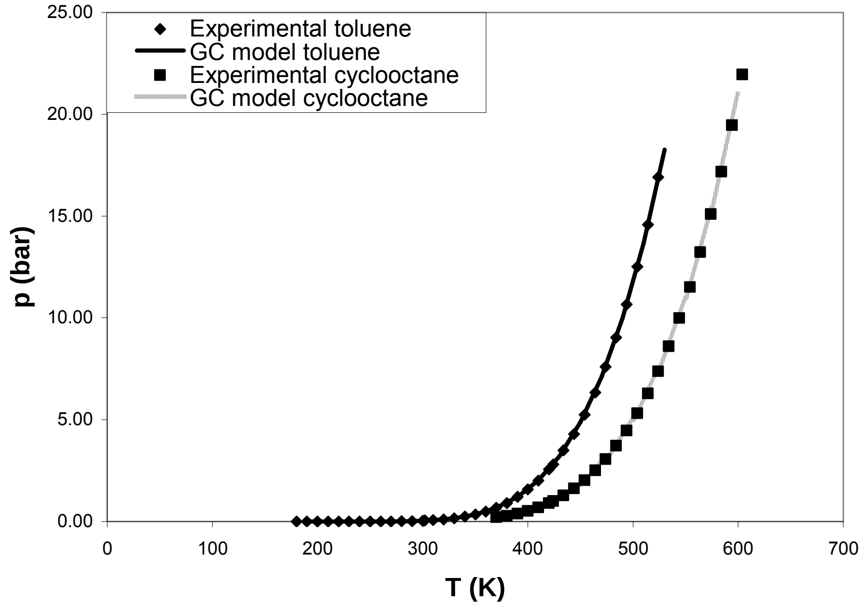


Figure 4.3- Vapour Pressure for Toluene and Cyclooctane

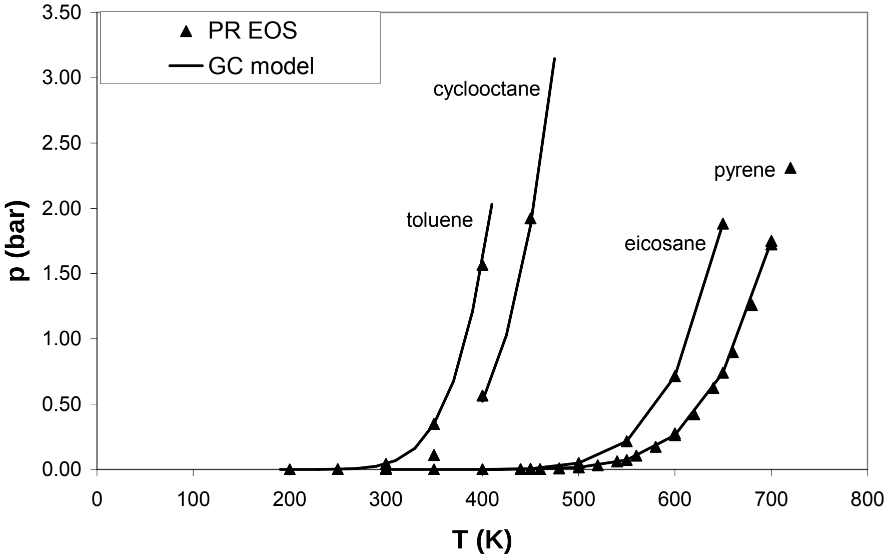


Figure 4.4- Vapour Pressure Comparison

Saturated liquid densities are compared in Figure 4.5 toFigure 4.8. The key findings are:

- For pyrene and eicosane, liquid densities from the GC model give acceptable fit with the experimental data – errors between 1.4 - 1.5 %. However, results from PR EOS differ approximately 26-30 % for pyrene and between 30-38 % for eicosane from the experimental density values.
- For toluene, both the GC model and PR EOS saturated liquid densities are very similar and have good agreement with the experimental data.
- For cyclooctane, the deviation of the GC model from the experimental data is around 5-7 %, while for PR EOS is 1-3 %. However, the GC model overestimates these values while PR EOS underestimates the densities.

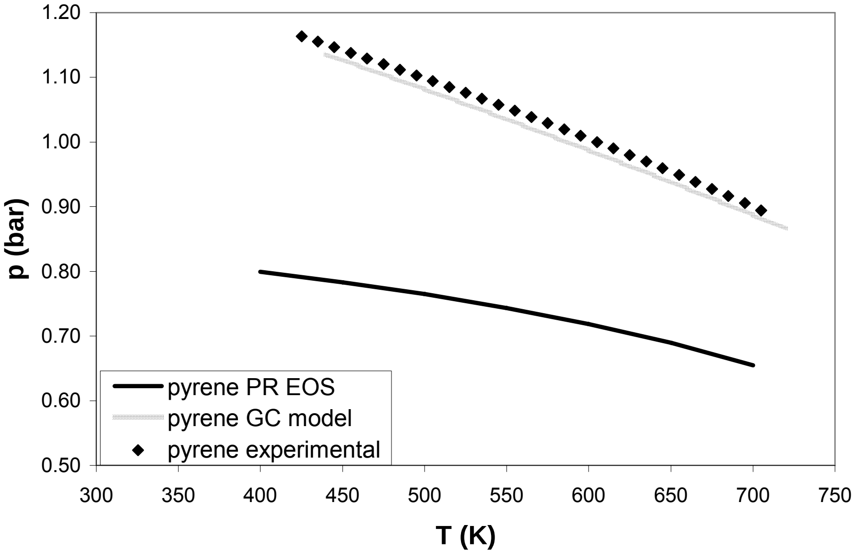


Figure 4.5- Saturated Liquid Density for Pyrene

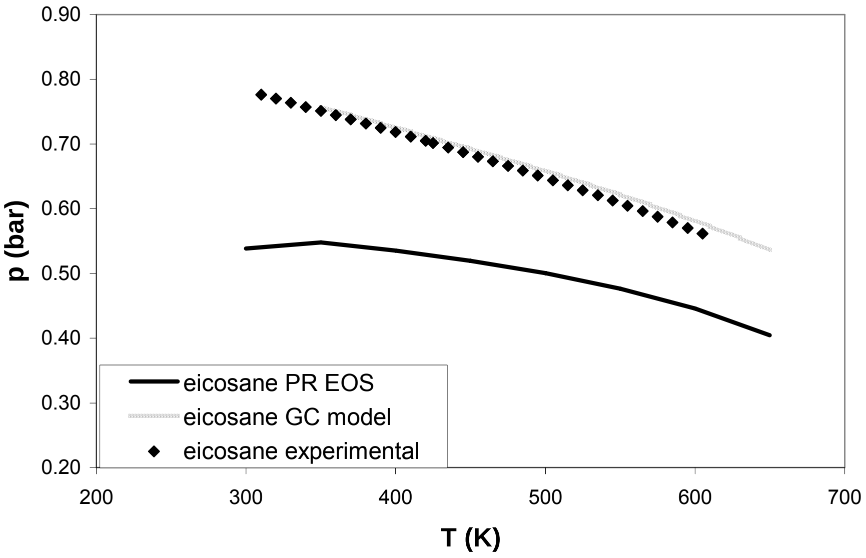


Figure 4.6- Saturated Liquid Density for Eicosane

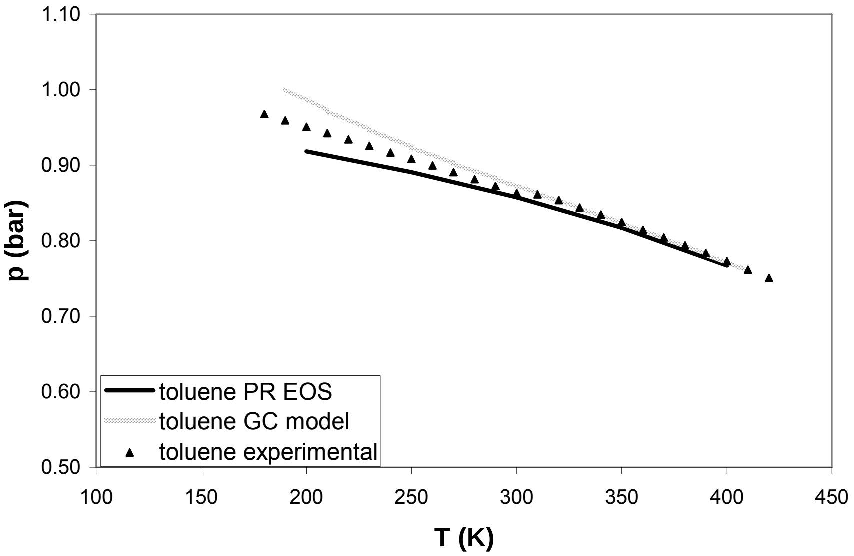


Figure 4.7- Saturated Liquid Density for Toluene

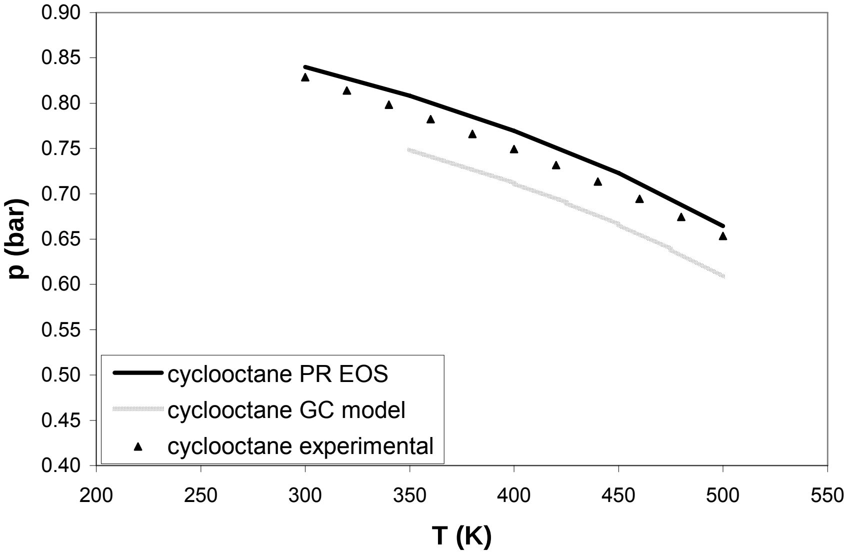


Figure 4.8- Saturated Liquid Density for Cyclooctane

The GC model and the PR EOS are equivalent for vapour pressure evaluation for pure fluids but for liquid densities the GC model is significantly better than the PR EOS.

For the pure fluid case, the values for the parameter b obtained for the PR EOS with the GC model are greater than the values from the GC model for almost all the compounds tested except for cyclooctane where the value from the GC model is slightly higher than the value from PR EOS – Table 4.2.

Table 4.2- Parameter b for Test Single Compounds

Compound	b GC model (cm^3/mol)	b PR EOS (cm^3/mol)
toluene	92.725	93.222
eicosane	329.515	477.037
pyrene	169.900	231.955
cyclooctane	127.518	117.917

Values for parameter a obtained from the GC model are similar to the values obtained for the PR EOS for cyclooctane and toluene. However, there is an appreciable difference for eicosane and pyrene – Figure 4.9, which explains the differences in density estimations between the two methods.

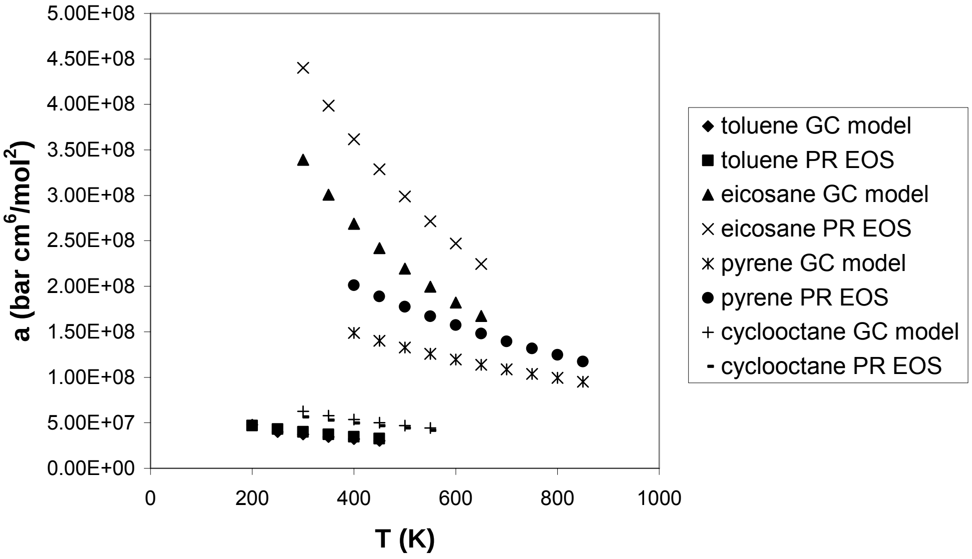


Figure 4.9- Parameter a(T) for Test Pure Fluids

4.2.2. AVR property prediction – based on pure fluid approximation

This section presents the results obtained by applying the pure fluid GC model to the calculation of the thermophysical properties of AVR. For these calculations, AVR is approximated as a pure fluid based on a molecule proposed by Sheremata (2002). The results obtained are compared with a previous application of a GC method (Coniglio et al., 2000) to AVR (Mahmoodaghdam et al., 2002). The main difference between the two calculations is that the present study uses the latest update of Coniglio's method (Crampon et al., 2003), while the previous application uses the original GC method proposed by Coniglio et al. (2000). The main differences between the two methods are the number of molecular groups used (7 Coniglio's method vs 19 Crampon update) and the expression to evaluate the shape parameter m.

The input information for the GC model was taken from Mahmoodaghdam et al. (2002) and it was based on the work done by Sheremata (2002). Table 4.3 presents the input data. An average molecule was selected for AVR with a corresponding molecular weight of 1742. The GC structure was transformed from the molecular groups definition of Sheremata (2002) to the structure applied in the present study (Crampon, 2003).

Table 4.3- Input Data of AVR for the GC Model

Compound	**MW**	**T_b (K)**	**GC Structure**
AVR pure fluid	1742	780	[12 41 9 3 18 0 0 0 0 24 14 9 0 0 0 0 0 0 0]

Vapour pressure results are presented in Figure 4.10. Experimental values were obtained from Zou (2003). Both versions of the GC model give similar results and underestimate the experimental values for vapour pressure by more than one order of magnitude at most of the temperatures considered.

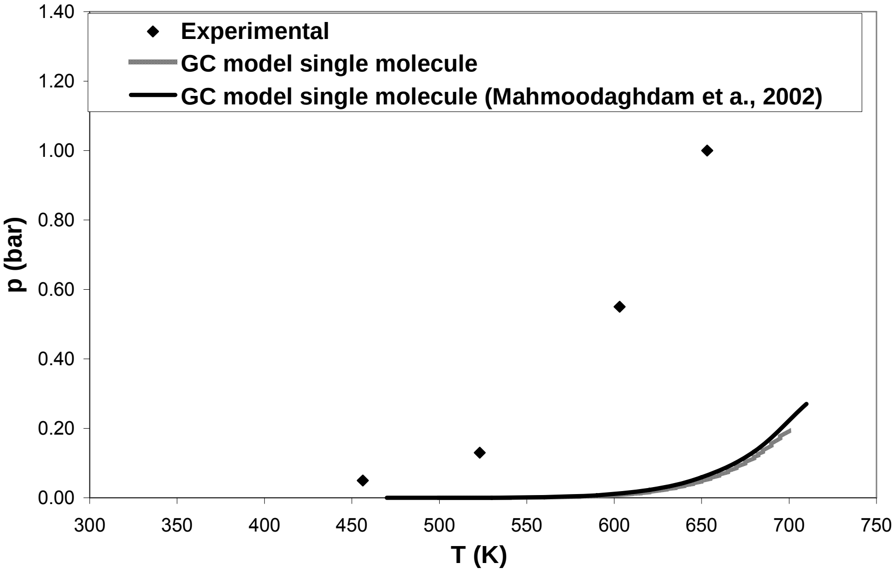


Figure 4.10- Vapour Pressure for AVR, Modeled as a Pure Fluid.

The results for saturated liquid densities are shown in Figure 4.11. Experimental values were taken from Zou (2003). The difference between GC model results and experimental data is approximately 15 % and 22 % for the current GC model and the previous application of the GC method (Mahmoodaghdam et al., 2002) respectively. However, when the volume correction introduced by the parameter c(T) is eliminated densities from the current GC model agree within the experimental error with the experimental data.

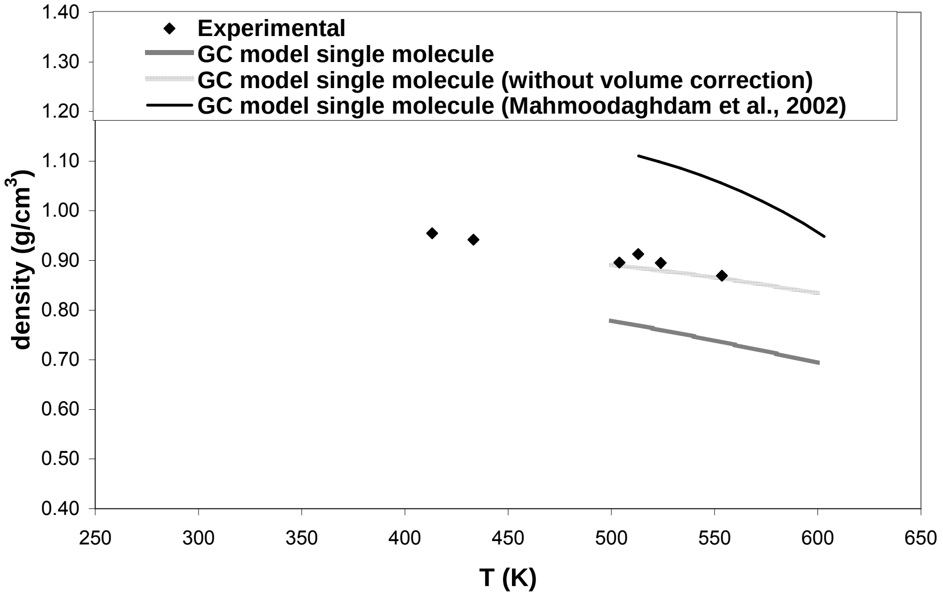


Figure 4.11- Saturated Liquid Densities for AVR, Modeled as a Pure Fluid.

Finally, the values for the parameters a and b are presented in Table 4.4 and Figure 4.12 respectively.

In both cases, the order of magnitude for these parameters is higher than the parameters calculated for the test compounds.

Table 4.4- Parameter b for AVR with a Single Molecule Representation

Compound	b GC model (cm^3/mol)
AVR pure fluid	1795.41

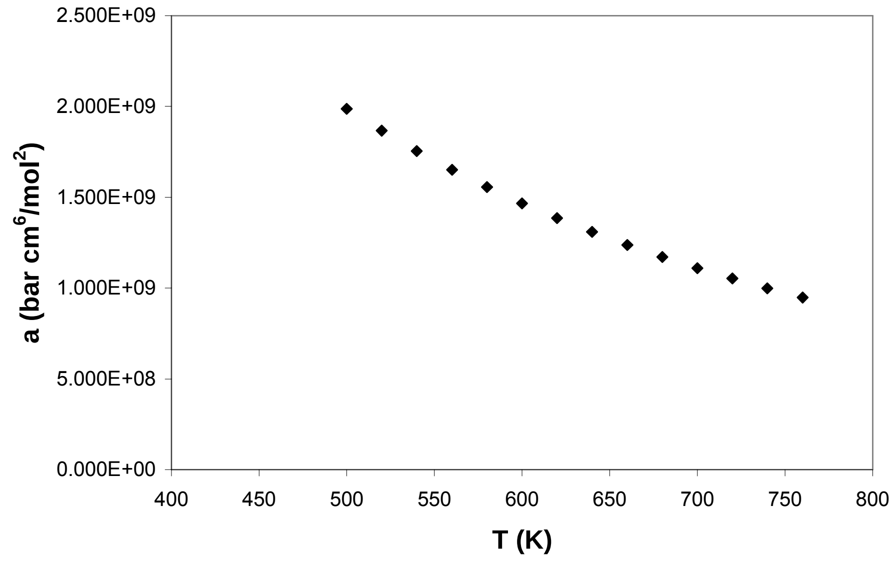


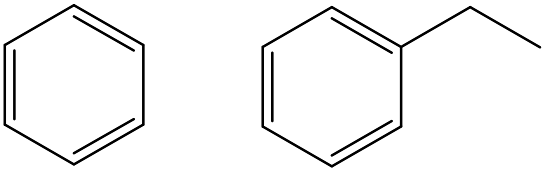
Figure 4.12- Parameter a(T) for AVR Modeled as a Pure Fluid.

Although the new version of the GC method improves the density estimation with respect to the previous version (Mahmoodaghdam et al., 2002), vapour pressures remain poorly predicted. One alternative to improve the accuracy of the GC model in the estimation of vapour pressures and densities of AVR is the application of a multicomponent GC method to AVR.

4.3. Extension of the GC Model to Mixtures

Before applying the GC multicomponent model to AVR, a set of thermophysical property calculations for test mixtures were performed and compared with available experimental data. The mixtures selected are listed below:

1. *Benzene-Ethylbenzene Mixture:* This is a binary mixture of two aromatic compounds. Their molecular structures and physical properties are presented in Figure 4.13 and Table 4.5 respectively. The MW and T_b are taken from Yaws (2003).



C_6H_6
benzene

C_8H_{10}
ethylbenzene

Figure 4.13- Molecular Structures for Benzene and Ethylbenzene

Table 4.5- Input Data for the GC Model. Benzene – Ethylbenzene Mixture

Compound	**MW**$^{(*)}$	**T_b (K)**$^{(*)}$	**GC Structure**
benzene	78.114	353.24	[0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0
ethylbenzene	106.167	409.35	[1 1 0 0 0 0 0 0 0 5 1 0 0 0 0 0 0 0

$^{(*)}$ MW and T_b were obtain from Yaws (2003)

Experimental vapour pressure data for benzene - ethylbenzene mixtures was gathered from Al-Ghamdi et al. (2001). The comparison of these data with the GC model results is illustrated in Figure 4.14. Excellent agreement with the experimental vapour pressure data is obtained. A maximum deviation of approximately 5.0 % is observed.

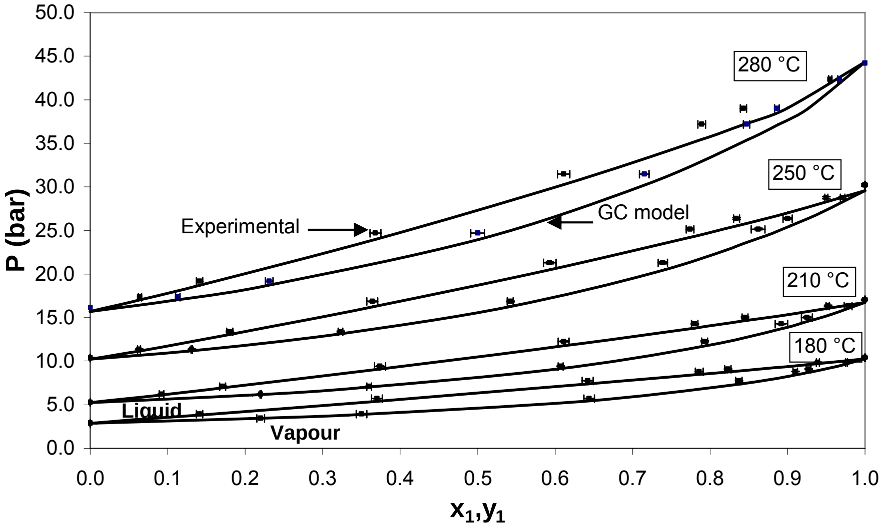


Figure 4.14- Pressure-Composition Diagrams for the Benzene (1) – Ethylbenzene (2) Mixtures. Temperature is a parameter.

The second analysis for benzene – ethylbenzene mixtures, consists of the comparison of the performance of the GC model vs. PR EOS. The PR EOS yields the same results as the GC model, both fit the experimental vapour pressure data within the experimental error, except for the vapour pressure composition diagram at 280 °C where PR EOS performs slightly better than the GC model. The pressure – composition diagram at 280 C is shown in Figure 4.15.

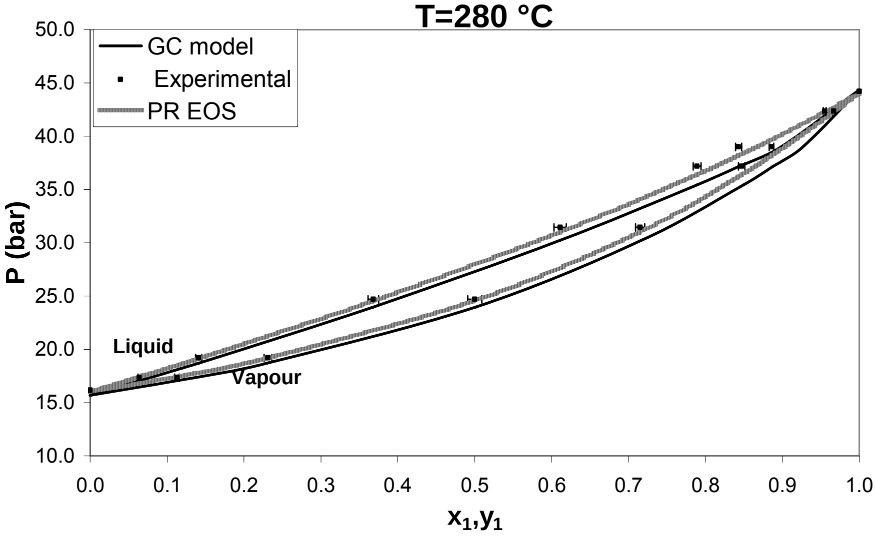


Figure 4.15- Pressure-Composition Comparison for Benzene (1) - Ethylbenzene (2) at 280 C.

Values for the parameters appearing in the GC and PR equations of state, are compared at two different temperatures. Table 4.6 presents this results, where the parameters a and b refer to the mixture and the intervals reflect the range of values arising for the composition range x_1 = 0.1 – 0.9.

Table 4.6- EOS Parameters Comparison. Benzene – Ethylbenzene Mixtures

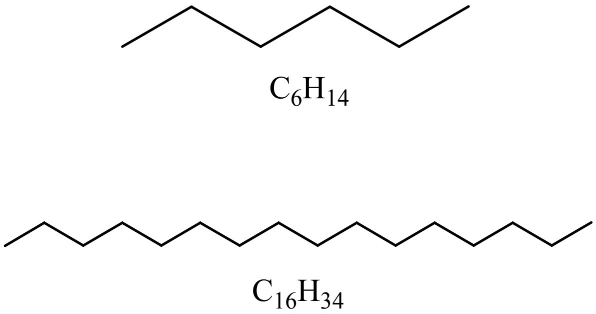
T (°C)	GC model		PR EOS	
	a (bar cm^6/mol^2)	b (cm^3/mol)	a (bar cm^6/mol^2)	b (cm^3/mol)
180	4.0 - 2.4 10^7	108.3 - 77.9	3.9 - 2.5 10^7	106.1 - 78.7
280	3.5 - 2.1 10^7		3.5 - 2.2 10^7	

Finally, two set of calculations were performed for vapour pressure, one with the standard GC model and a second one where the boiling point temperature T_b is estimated using the GC method (Coniglio et al., 2001). The estimated T_b values are compared with the experimental values (Yaws, 2003) in Table 4.7.

Table 4.7- T_b Comparison for Benzene and Ethylbenzene

Compound	**T_b estimated (K)**	**T_b experimental (K)**
benzene	362.81	353.24
ethylbenzene	429.94	409.35

2. *Hexane–Hexadecane Mixture:* this mixture is an asymmetric binary mixture comprising two alkanes. The molecular structures and physical properties are presented in Figure 4.16 and Table 4.8 respectively. The MW and T_b were taken from Yaws (2003).



C_6H_{14}

$C_{16}H_{34}$

Figure 4.16- Molecular Structure for the Hexane and Hexadecane

Table 4.8- Input Data for the GC Model. Hexane – Hexadecane Mixture

Compound	$MW^{(*)}$	T_b (K)$^{(*)}$	GC Structure
hexane	84.177	341.88	[2 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
hexadecane	226.446	560.01	[2 14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

$^{(*)}$ MW and T_b were obtain from Yaws (2003)

Vapour liquid equilibrium data for this system at two different temperatures was taken from Joyce et al. (1998) and are shown in Figure 4.17 and Figure 4.18 along with results obtained from the GC model. GC model pressure and composition values fit the equilibrium data within experimental error. A similar performance is seen with the PR EOS, Figure 4.19.

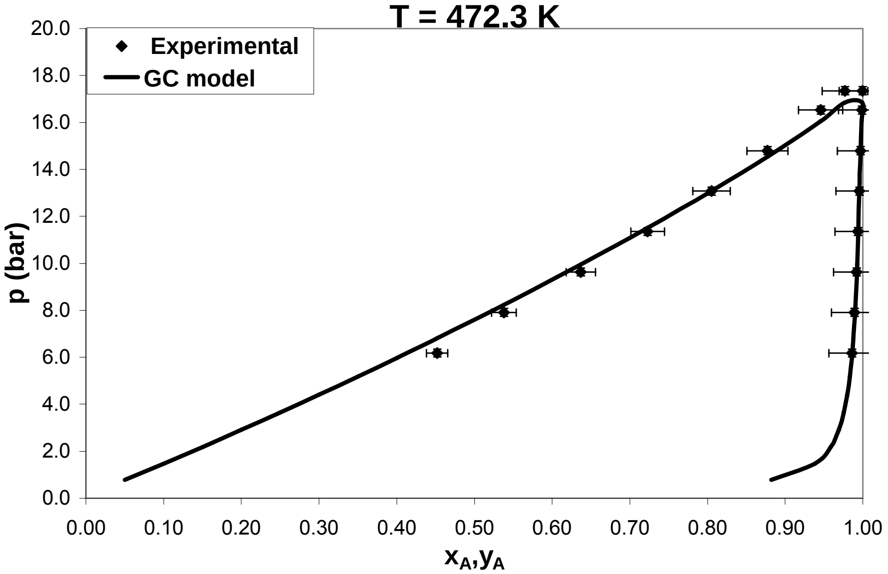


Figure 4.17- Pressure-Composition Diagram for Hexane (A)-Hexadecane (B) Mixtures. T=472.3 K

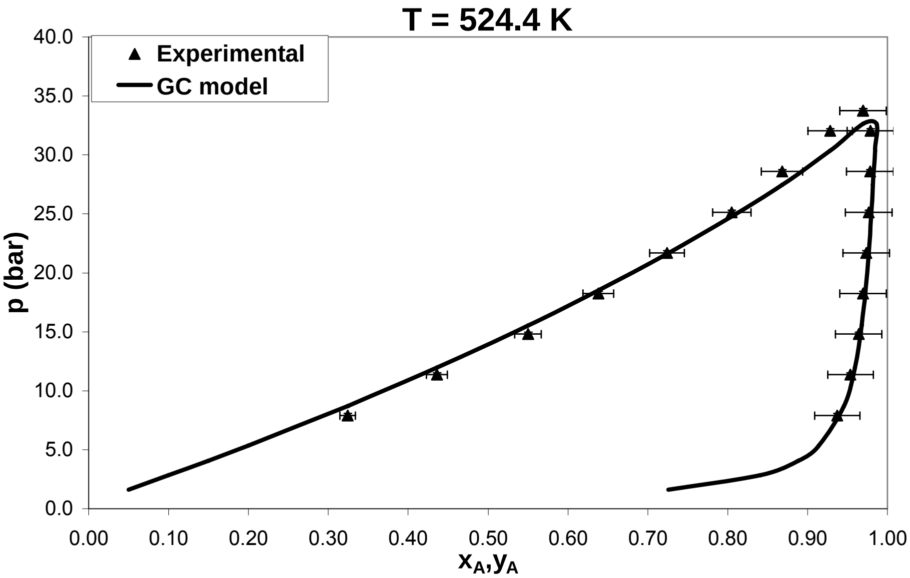


Figure 4.18- Pressure-Composition Diagram for Hexane (A)-Hexadecane (B) Mixtures. T=524.4 K

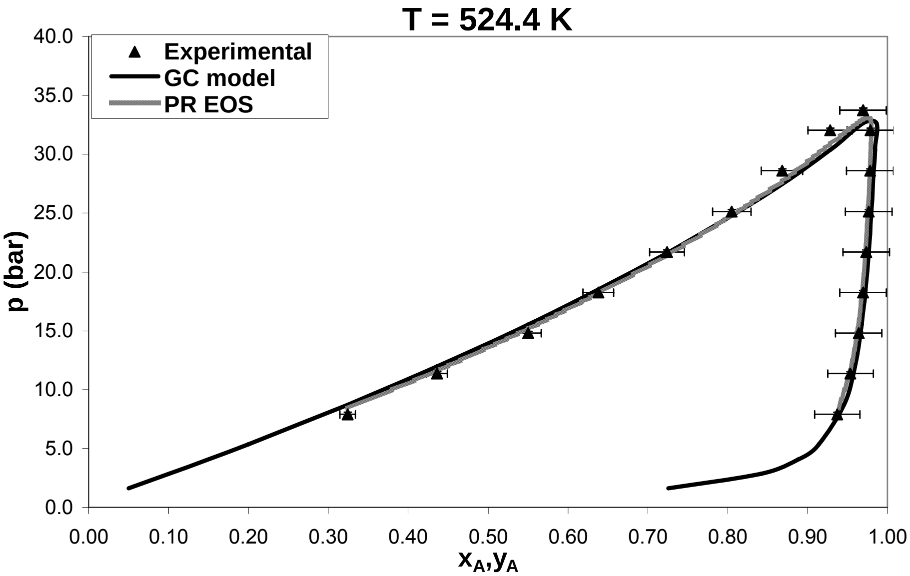


Figure 4.19- Pressure-Composition Comparison for Hexane (A)-Hexadecane (B) Mixtures. T=524.4 K

The GC and PR EOS parameters are compared in Table 4.9. Again, the intervals shown reflect variation with composition. Values for the parameter a from the GC model differ from PR EOS approximately by 2-15 % and between 4-20 % for parameter b.

Table 4.9- EOS Parameters Comparison. Hexane – Hexadecane System

T (K)	**GC model**		**PR EOS**	
	a (bar	**b (cm^3/mol)**	**a (bar**	**b (cm^3/mol)**
472.3	8.5 – 3.0 10^7	193.7 – 110.0	9.7 – 3.1 10^7	229.3 – 114.0
524.4	9.4 – 3.1 10^7		10.1 – 3.2 10^7	

T_b estimated through the GC method are compared with the experimental values in Table 4.10 There is not a significant difference between these values and the VLE results for hexane – hexadecane mixtures are unaffected.

Table 4.10- T_b Comparison for the Hexane – Hexadecane System

Compound	**T_b estimated (K)**	**T_b experimental (K)**
Hexane	347.08	341.88
Hexadecane	569.98	560.01

3. *Complex Mixture:* this system consists of a multicomponent mixture. The molecular structure and physical properties of the components are shown in Figure 4.20 and Table 4.11. In the case of decomposing methane into

the molecular structure of the GC method, an approximation to one group of $(CH_3)_{alkane}$ was made.

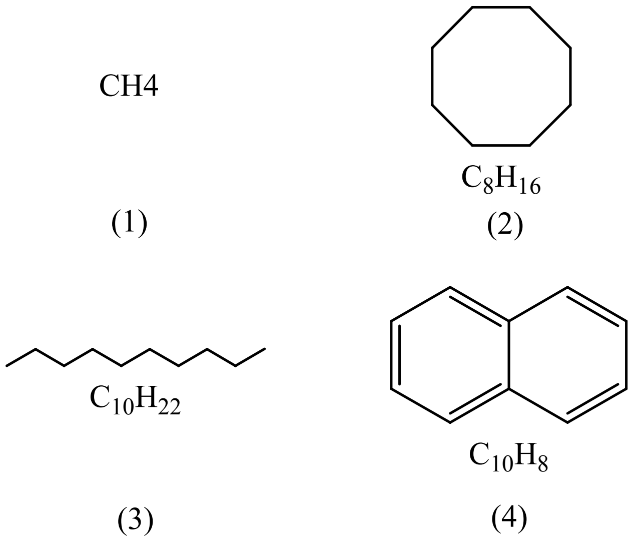


Figure 4.20- Molecular Structure for Complex Mixture

Table 4.11- Input Data for the GC Model. Complex Mixture

Compound	MW(*)	T_b (K)(*)	GC Structure
methane (1)	16.043	111.66	[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
cyclooctane (2)	112.215	424.30	[0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
decane (3)	142.285	447.30	[2 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
naphthalene (4)	128.174	491.14	[0 0 0 0 0 0 0 0 0 8 0 2 0 0 0 0 0 0 0]

(*) MW and T_b were obtain from Yaws (2003)

The molar composition of the mixture employed in calculations is presented in Table 4.12.

Table 4.12- Molar Composition for Complex Mixture

Compound	x_i
methane (1)	0.2506
cyclooctane (2)	0.0532
decane (3)	0.6475
naphthalene (4)	0.0487

A pressure-temperature diagram is shown in Figure 4.21. Two different results are shown, the first one with the interaction parameter (k_{ij}) set to zero and the second one with k_{ij} adjusted to fit the experimental data for vapour pressure of the mixture. Equivalent results obtained using the PR EOS are also presented. Experimental data was taken from Shariati et al. (1998).

With $k_{ij} = 0$, the GC model overestimates vapour pressures for this mixture, and the PR EOS underestimates them. In both cases the error with respect to the experimental data is around 12 %, which is a significant deviation. For this reason a second set of results were obtained by adjusting the interaction parameters and minimizing the error with respect to the vapour pressure experimental data. The final values for k_{ij} are presented in Table 4.13. Vapour pressures for this case agree with the experimental data within 5 % for the GC model and 3% for PR EOS.

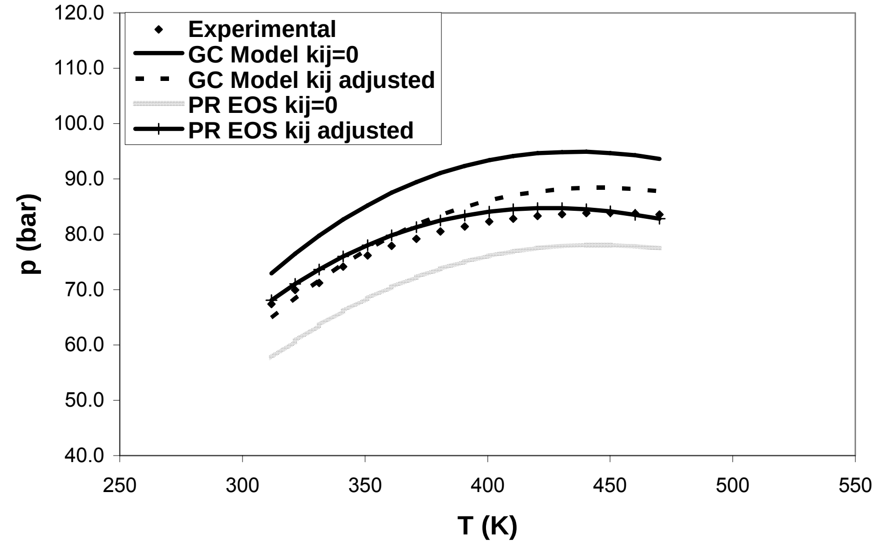


Figure 4.21- Pressure-Temperature Diagram for Complex Mixture

Table 4.13- Interaction Parameter Values for Complex Mixture

	GC model				**PR EOS**			
k_{ij} adjusted	0.00	0.00	0.00	0.00	0.00	0.03	0.05	0.03
	0.00	0.00	0.08	0.08	0.03	0.00	0.00	0.00
	0.00	0.08	0.00	0.08	0.05	0.00	0.00	0.00
	0.00	0.08	0.08	0.00	0.03	0.00	0.00	0.00

The parameters for the GC and PR EOS are compared in Table 4.14 and Figure 4.22. The differences between GC model and PR EOS parameters are 7.6-8.5 % for a(T) and 10.1 % for b.

Table 4.14- Parameter b for Complex Mixture

Mixture	b GC model	b PR EOS (cm^3/mol)
Complex Mixture	127.89	140.81

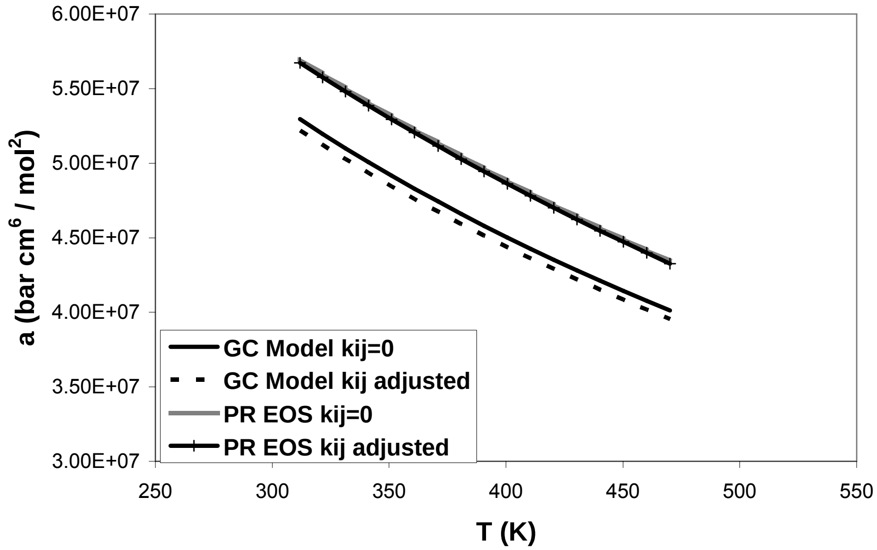


Figure 4.22- Parameter a(T) for Complex Mixture

Finally, boiling point temperatures obtained from the GC model were compared with experimental values. The deviation for methane is high (93 %). In addition, when using the estimated T_b for VLE calculations with the GC model, no convergence is obtained. This is because the T_b value used for methane is 215.56 K. Consequently, the approximation made for the molecular structure of methane as one CH_3 group is not valid.

Table 4.15- T_b Comparison for Complex Mixture

Mixture	T_b estimated (K)	T_b experimental (K)
methane (1)	215.56	111.66
cyclooctane (2)	435.27	424.30
decane (3)	455.68	447.30
naphthalene (4)	513.73	491.14

4.4. Application of the Multicomponent GC Model to AVR

Bubble pressure and liquid density results obtained by applying the multicomponent GC model with different multicomponent molecular representations to AVR are reported. The molecular structures and molar compositions for these mixtures were taken from the work of Sheremata (2005). Different scenarios are analyzed in Section 4.4.1. In the first case, a mixture of 60 molecules is taken following the recommendation from Sheremata (2005). Next, the results when using a 10 molecule approximation are analyzed. Finally, results for the mixture of AVR and decane are presented.

4.4.1. Athabasca Vacuum Residue (AVR)

In the first case AVR was divided into ten fractions and each fraction is represented by six molecules (Sheremata, 2005). The physical properties and input data for the GC model of all the sixty molecules are presented in Appendix III.

The liquid molar composition for AVR was taken from Sheremata (2005) and is presented in Table 4.16. In this case 6 molecules are considered for each fraction and the molar composition for each molecule was calculated as one sixth of the corresponding fraction composition.

Table 4.16- Molar Compositions for AVR. (Sheremata, 2005)

Fraction	**x_i**
1	0.2783
2	0.1439
3	0.1185
4	0.1469
5	0.0874
6	0.0515
7	0.0322
8	0.0238
9	0.0080
10	0.1096

The second scenario considers ten fractions with one molecule per fraction. The input data for the GC method is included in Appendix III. In this case, the molar fractions presented in Table 4.16 represent the compound mole fractions considered in the simulation.

Vapour pressure and saturated liquid density calculations were performed for both cases with the GC model and are presented in Figure 4.23 and Figure 4.24 respectively. Results are also compared with the single compound approximation for AVR presented in Section 4.2.2. For these cases the interaction parameter was set to zero.

Vapour pressures obtained with the multi-molecule representation for AVR are significantly lower than the experimental values (Zou, 2003) and deviate more from the experimental values than the pure fluid approximation for AVR. Multiple reasons for these results are considered:

- Uncertain experimental error in vapour pressure data for AVR. Due to difficulties in the handling of this very viscous material. In Addition, differences in the specific sample of AVR used for all the measurements involved in this work.
- Approximation of the GC molecular structure in cases where sulphur nitrogen or oxygen is present within the molecule. In these cases a substitution by a carbon atom was made.
- Extremely high molar mass for some molecules. These data was obtained from Sheremata (2005) and is presented in Appendix III. Molar masses for molecules representing fraction 10 range between 3348-4033. These high values of molar masses may cause worse results for vapour pressures obtained from the multicomponent GC model compared with the pure fluid approximation presented before for AVR.
- Possible multiphase behaviour within the liquid phase for AVR. New experimental calorimetric data suggest that solid-liquid equilibrium may be present at these temperatures for AVR. In this case vapour pressure would be underestimated, and GC model vapour pressures results are consistent with this hypothesis.

Predicted saturated liquid density values for AVR (Figure 4.24), follow the trend above. Densities from the pure fluid approximation have a better agreement with the experimental data than the multicomponent ones. On the other hand, results from the multi-molecule representation for AVR overestimate density values approximately by 19 % and 12% in the 10 and 60 molecules cases respectively. These deviations may be explained with the same arguments presented earlier in regarding the vapour pressure analysis.

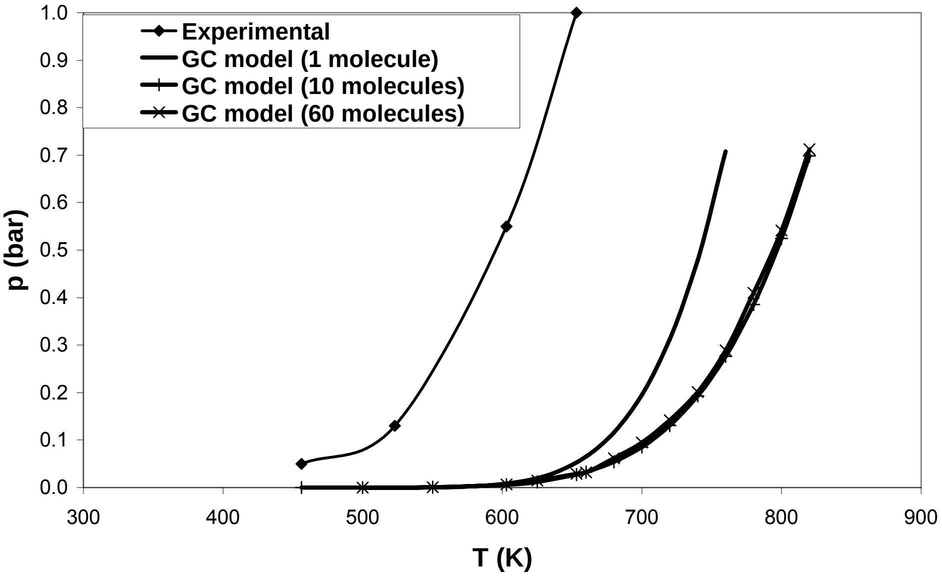


Figure 4.23- Vapour Pressures for AVR. Multi-Molecule Representations

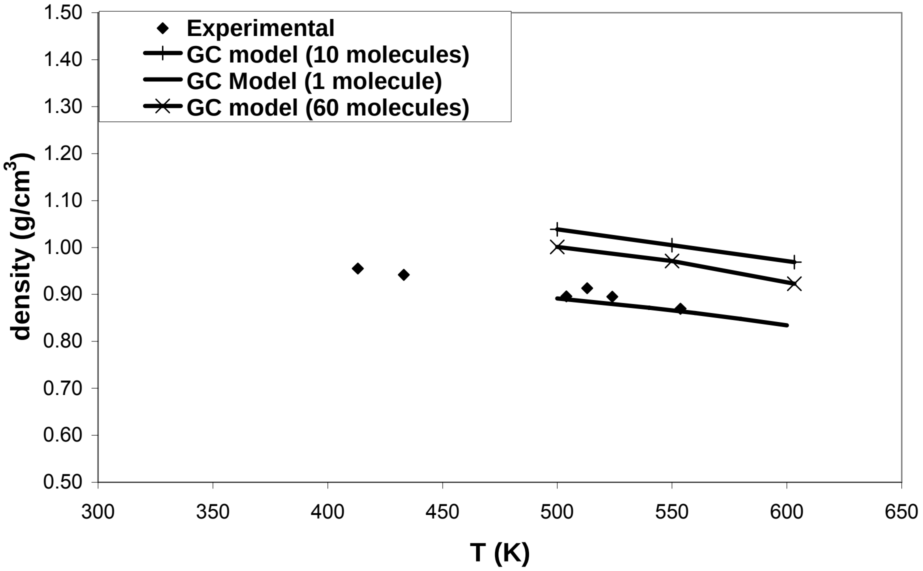


Figure 4.24- Saturated Liquid Densities for AVR. Multi-Component Representations

Values for the parameters a and b appearing in the GC and PR EOS, for the multi-componen representation for AVR are presented in Table 4.17. In both cases the values for the parameters are higher than the parameters obtained for the pure fluid approximation (Table 4.4 and Figure 4.12).

Table 4.17- EOS Parameters Comparison. AVR with Multi-Molecule Representations

T (K)	GC model. 10 molecules		GC model. 60 molecules	
	a (bar cm^6/mol^2)	b	a (bar cm^6/mol^2)	b
500.0	1.77 10^9	991.82	1.98 10^9	1132.1
820.0	9.01 10^8		1.10 10^9	

Two additional sensitivity analyses were performed with the 10 molecule representation for AVR. In the first analysis the interaction parameter (k_{ij}) was changed starting with values calculated using the expression suggested by Pedersen et al. (1988). This expression was developed for hydrocarbon-hydrocarbon systems and the correlation is based on data from North Sea reservoir fluids:

$$k_{ij} \cong 0.001\frac{MW_i}{MW_j} \quad \text{where } MW_i > MW_j \qquad 4.1$$

Using these values for k_{ij} in the GC model did not improve vapour pressure or density results for AVR. Other k_{ij} values were tested, and no noticeable improvement was obtained.

The second sensitivity analysis consisted of readjusting the molecular weight of the molecules in order to avoid extremely high molar mass values. The 10 molecule case was used and vapour pressures and densities for AVR were computed with the GC model. The best agreement with experimental data was obtained with molecular weights no greater than 500. In this case agreement with the experimental density data (Zou, 2003) was obtained. However, vapour pressures are still underestimated by the GC model. Both vapour pressure and density results are presented in Figure 4.25.

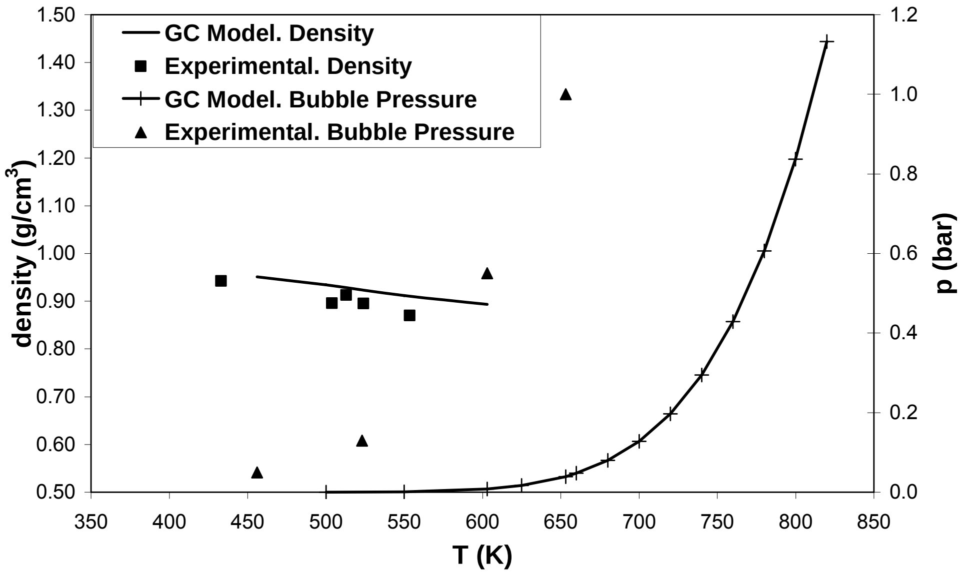


Figure 4.25- Vapour Pressures and Densities for AVR. MW ≤ 500

These MW analysis results and the fact that the single molecule representation for AVR with a MW = 1742 produced a better agreement with experimental density data suggest that molar masses obtained from the work of Sheremata (2005) for fraction 10 (3348-4033) may be too high.

In general, the GC computational model developed in the present study using multi-molecule representations obtained from Sheremata (2005) for AVR did not estimate satisfactory vapour liquid equilibria for AVR. Further research is necessary to establish the molecular structures and molar mass range for this hydrocarbon fraction. Further, there are numerous experimental challenges related to the measurement of vapour pressures of very viscous and complex substances such as AVR. The data presented for AVR have a random error of +/- 0.1 bar, at a minimum. The Petroleum Thermodynamics group in the CME Department has been developing experimental techniques in this area with a MSc project being pursued by Richard McFarlane and a

PhD project pursued by Austrich (2006). In Austrich's most recent work, for example, he reports vapour pressures of five atmospheric Mexican resids, whose vapour pressures should be higher than AVR. In all cases, the vapour pressures are lower than those reported for AVR but of the same order of magnitude. For example, Austrich reports (Austrich, 2006) vapour pressures of 0.3 +/- 0.06 bar at 570 K for the resids evaluated where as the reported value for AVR at the same temperature is 0.4 +/- 0.1 bar. Thus, vapour pressure remains under predicted by an order of magnitude, even when compared to the best available data.

The model also assumes that AVR is a single phase material. The most recent work from the Petroleum Thermodynamics research group in the CME Department suggests that this is not the case (FPE, 2005). The single phase assumption leads to under prediction of vapour pressure.

4.4.2. Mixtures of AVR and decane

This last section presents results for vapour pressures and saturated liquid densities obtained from the GC model when applied to mixtures of AVR and decane. Vapour pressure experimental data was obtained from Zhang (2005). Because the best fit with VLE experimental data for AVR was produced with the pure fluid approximation, in this case AVR was represented as a pure fluid. The input data for the GC model is presented in Table 4.18. MW and T_b for AVR and decane were taken from Mahmoodaghdam et al. (2002) and Yaws (2003) respectively.

Table 4.18- Input Data for the GC Model. AVR + Decane

Compound	**MW**	**T_b (K)**	**GC Structure**
AVR (1)	1742.0	780.0	[12 41 9 3 18 0 0 0 0 24 14 9 0 0 0 0 0 0 0]
Decane (2)	142.3	447.3	[2 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

VLE data at 50 wt % AVR and 10 wt % AVR are compared with the GC model in Figure 4.26 and Figure 4.27 respectively. The 50 wt % mixture comprises a minimum of two condensed phases while the 10 wt % mixture comprises a single condensed phase (Zhang, 2006). The corresponding molar compositions are shown in

Table 4.19. Vapour pressures obtained from the GC model are lower than the experimental data at all the temperatures tested. These results follow the trend presented in the previous section for AVR, where the GC model underestimates vapour pressures. In addition, performing calculations in the single condensed phase region, improves the agreement between the GC model and experimental data for vapour pressure - an expected result.

Density calculations are presented in Figure 4.28 and Figure 4.29. Good agreement with experimental density data is obtained in both cases. However, the mixture with the lowest content of AVR (10 wt %) gives a better fit with the experimental data, which again constitute an expected result.

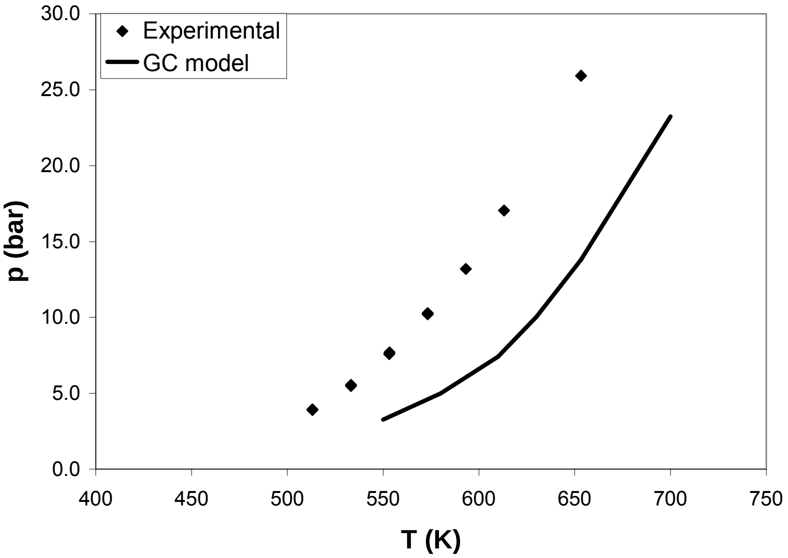


Figure 4.26- Vapour Pressures for AVR (50 wt %) + Decane

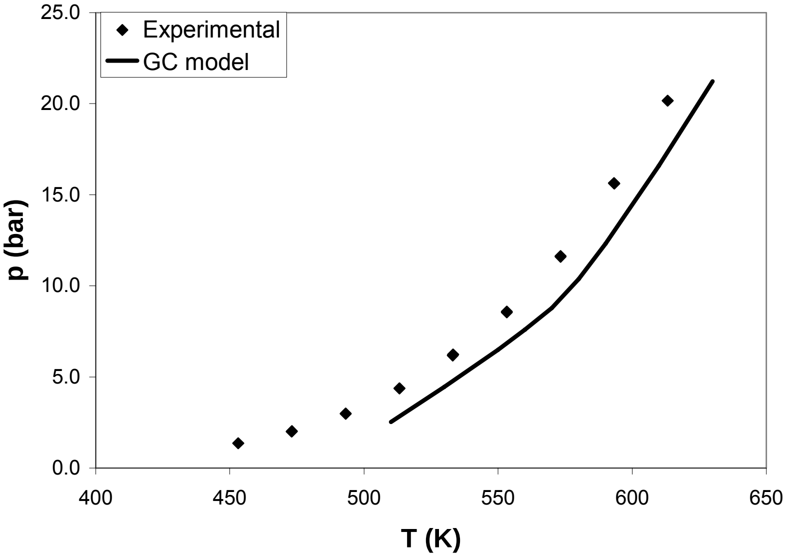


Figure 4.27- Vapour Pressures for AVR (10 wt %) + Decane

Table 4.19- Molar Compositions for AVR + Decane

	c_i (wt %)	x_i
Mixture 1:		
AVR (1)	50	0.0755
decane (2)	50	0.9245
Mixture 2:		
AVR (1)	10	0.0090
decane (2)	90	0.9910

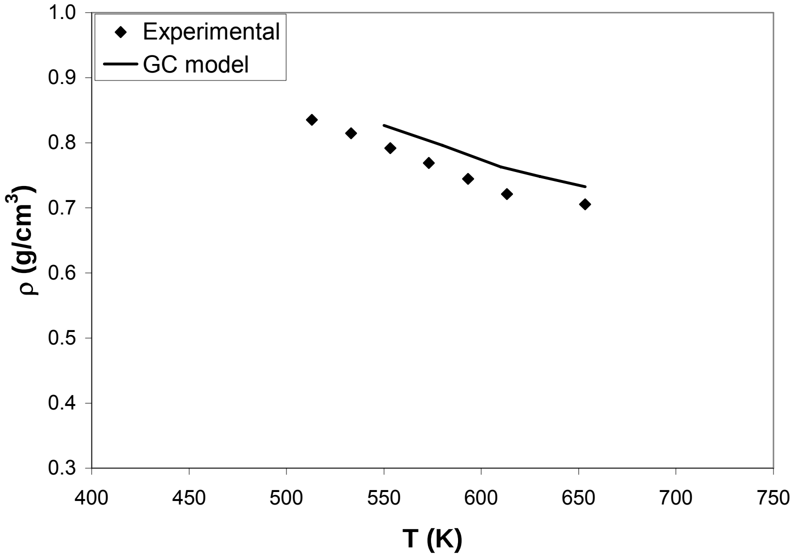


Figure 4.28.- Saturated Liquid Density for AVR (50 wt %) + Decane

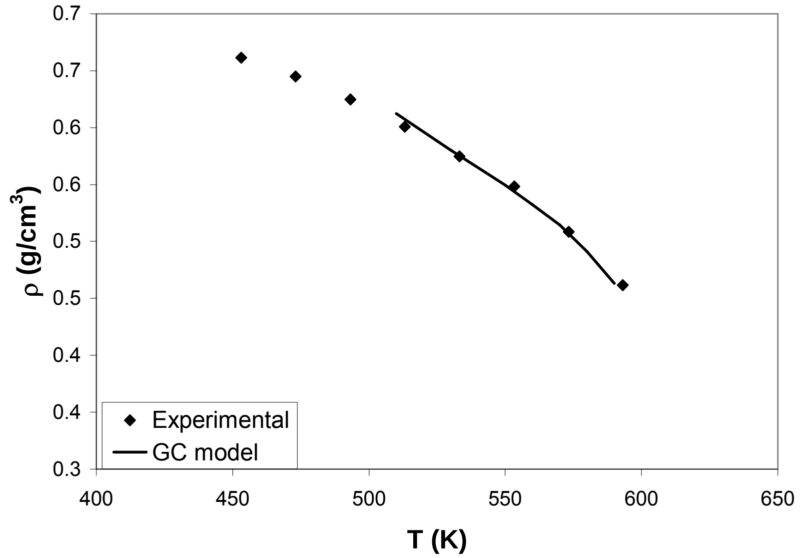


Figure 4.29.- Saturated Liquid Density for AVR (10 wt %) + Decane

The corresponding parameters a and b calculated by the GC model at two temperatures are presented in Table 4.20. It can be seen that reducing the composition of AVR from 50 wt% to 10 wt% 2 has a considerable effect in the values for both parameters.

Table 4.20- EOS Parameters. AVR + Decane

T (K)	**Mixture 1**		**Mixture 2**	
	a (bar cm^6/mol^2)	**b (cm^3/mol)**	**a (bar cm^6/mol^2)**	**b (cm^3/mol)**
550.0	1.05 10^8	292.85	6.45 10^7	184.75
700.0	7.99 10^7		5.16 10^7	

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

From the discussion presented in Chapter 4 the following conclusions can be drawn:

1) The GC method proposed by Coniglio et al (2000) can be implemented as a MATLAB computational model to estimate thermophysical properties for pure hydrocarbons as well as hydrocarbon mixtures. This well tested and well documented tool will serve as a general purpose computational platform for VLE computation and data benchmarking for the Petroleum Thermodynamics Group and will be made available more broadly to the research community.
2) The GC based EOS was tested with a variety of pure fluids and gives satisfactory results for both vapour pressure and liquid density over broad ranges of temperature and pressure.
3) For pure fluids, vapour pressures computed using the PR EOS, the standard EOS for hydrocarbon VLE calculations, are in close agreement with the values calculated using the GC model and both are in close agreement with experimental data. However, liquid densities for test compounds are better estimated using the GC model than with PR EOS.
4) The GC model underestimates vapour pressure values for AVR using a pure fluid approximation by more than one order of magnitude at almost all the temperatures considered.

5) The difference between the pure-fluid GC model density results and experimental data for AVR is approximately 15 %. However, when the volume correction introduced by the parameter c(T) is eliminated, densities from the GC model agree with the experimental data within experimental error.
6) When compared with experimental data, the multicomponent GC model gives satisfactory results for vapour pressure vs molar composition for hydrocarbon mixtures evaluated and the results obtained are in close agreement with predictions obtained using the PR EOS.
7) In the case of multicomponent asymmetric hydrocarbon mixtures, such as the Complex Mixture (methane, cyclooctane, decane and naphthalene), interaction parameters must be adjusted in order to obtain a reasonable fit of the GC model results with experimental vapour pressure data. A similar adjustment is necessary for the PR EOS.
8) Boiling point temperatures (T_b) for the components estimated through the GC method yield satisfactory agree with the experimental values, except for the case of methane, where the difference between the estimated and the experimental value is 93 %.
9) When applying the multicomponent GC model to VLE calculations for AVR no improvement was obtained in vapour pressure and density results compared with the pure fluid model.

10) Interaction parameter variation did not improve the performance of the multicomponent GC model with respect to the estimation of vapour pressure and density for AVR.
11) When the molecular weight of the fractions for AVR was decreased to a maximum value of 500 the density data calculated using the GC method fit the experimental density data within experimental error. This result suggests that the molar masses reported by Sheramata (2005) for AVR fractions are too high.
12) Predicted vapour pressures for AVR remain too low even if the assumed molar mass is reduced. This is attributed primarily to the multiphase nature of AVR and inadequacies of the molecular representation for AVR, and secondarily to experimental difficulties associated with the measurement of low volatility fluids.
13) The multicomponent GC model underestimates vapour pressures for mixtures of AVR and decane. However, the under prediction is relatively small and consistent, and the discrepancy can be tuned particularly in the single liquid phase composition region.

5.2. Recommendations

Based on the present study, the following recommendations are proposed for the extension of this research:

1) A sensitivity analysis can be conducted using the GC method to estimate the thermophysical properties of AVR by changing the mixing rules in the GC-EOS.
2) Further research is needed to better understand and represent the molecular structure of a complex mixture such as AVR, especially for the heaviest fraction that includes asphalthenes.
3) In order to decrease the error in the experimental data, more convenient methods should be designed and applied to measure vapour pressures of very viscous materials like AVR.
4) The role of density correction in the GC model should be explored further. In the mean time, the GC model should be used without density correction for heavy hydrocarbons.

REFERENCES

Al-Ghamdi A. M. and. Kabadi V. N. "High Temperature VLE for the Benzene-Ethylbenzene System". J. Chem. Eng. Data. Vol 46, 1330-1332. 2001.

Austrich, A., Gonzalez E., Lira C. and Shaw J. "Densities, Phase Diagrams and Phase Distribution in Five Mexican Atmospheric Residues Using a X-Ray View Cell". Instituto Mexicano del Petroleo. 2006.

Bondi, A. "Physical Properties of Molecular Crystals, Liquids and Glasses". John Wiley & Sons: New York, 1968.

Cartlidge, C. R. "On the Prediction of Solid Phases and K and L Points". Proposal for the Degree of Doctor in Philosophy. Graduate Department of Chemical Engineering and Applied Chemistry. University of Toronto. 1997.

Coniglio, L., Trassy, L. and Rauzy, E. "Estimation of Thermophysical Properties of Heavy Hydrocarbons through a Gropu Contribution Based Equation of State". Ind. Eng. Chem. Res. Vol. 39, 5037-5048. 2000.

Coniglio, L., and Nouviaire, A. "A Method for Estimating the Normal Boiling Point of Heavy Hydrocarbons Suitable for a Group-Contribution-Based Equation of State". Ind. Eng. Chem. Res. Vol. 40, 1781-1790. 2001.

Crampon C., Trassy L., Avaullée L., Neau E. and Coniglio L. "Simplification and extension of a predictive group contribution method for estimating heavy organic pure compound vapor pressures I. Hydrocarbons". Fluid Phase Equilibria. Vol 216, 95-109. 2003.

Gray, M. "Hydrocarbon Fluid Properties and Processing". Department of Chemical and Materials Engineering. University of Alberta: Edmonton, 2003-2004.

Joyce P.C. and Thies M.C. "Vapor-Liquid Equilibria for the Hexane + Hexadecane and Hexane + 1-Hexadecanol Systems at Elevated Temperatures and Pressures". J. Chem. Eng. Data. Vol 43, 819-822. 1998.

Mahmoodaghdam E. and Shaw J. M.. "Thermophysical Properties of Athabasca Bitumen Vacuum Bottoms Using a Group Contribution Based Equation of State". University of Alberta. 2002.

Mathworks Inc. "Learning MATLAB 7". 4th edition. 2004.

Melhem, G. A. "A modified Peng-Robinson Equation of State". Fluid Phase Equilib. Vol 47, 189-237. 1989.

Michelsen M. L. “Phase Equilibrium Calculations. What is Easy and What is Difficult?”. Computers Chem. Eng.. Vol 17, No. 5/6, 431-439. 1993.

Michelsen M. L. “The isothermal flash problem. I. Stability”. Fluid Phase Equilibria. Vol 9, 1-19. 1982.

Nghiem L. X. and Li Y. “Application of the Tangent Plane Criterion to Saturation Pressure and Temperature Computations”. Fluid Phase Equilibria. Vol 21. 39-60. 1985.

Pedersen K. S., Thomassen,P. and Fredenslund, A. “On the Dangers of Tuning Equation of State Parameters”. Chemical Engineering Science. Vol. 43, No. 2, pp. 269-278. 1988.

Péneloux, A.; Rauzy, E. “A Consistent Correction for Redlich-Kwong-Soave Volumes”. Fluid Phase Equilib. Vol 8, 7-23. 1982.

Peng, D. Y.; and D. B. Robinson. ”A New Two Constant Equation of State”. Ind. Eng. Chem. Fundam. Vol 15, 58-64. 1976.

Poling B.E., Prausnitz J.M. and O’Connell J.P. “The Properties of Gases and Liquids”. 5th edition. McGraw Hill. 2001.

Rauzy, E. “Les Me´thodes Simples de Calcul des Equilibres Liquide-Vapeur Sous Presión”. Ph.D. Dissertation. The French University of Aix-Marseille III, Marseille, France. 1982.

Redlich, O. and Kwong, J.N.S. “On the thermodynamics of solutions”. Chemical Reviews. Volume 44. 233-244. 1949.

Riazi, M. R. “Characterization and Properties of Petroleum Fractions”. ASTM. 2005.

Rogalski, M.; Carrier, B. and Péneloux, A. “Correlation and Prediction of Physical Properties of Hydrocarbons with the Modified Peng-Robinson Equation of State. 3. Thermal Properties. m - A New Significant Characterizing Parameter”. Ind. Eng. Chem. Res. Vol 29, 659-668. 1991.

Sengers, J.V., Kayser, R.F., Peters, C.J. and White H.J. “Equations of State for Fluids and Fluid Mixtures”. Part I. IUPAC. 1st edition. 2000.

Shariati, A., Peters, C. J. and Moshfeghian M. “Bubble-Point Pressures of Some Selected Methane + Synthetic C6+Mixtures”. J. Chem. Eng. Data. Vol 43, 280-282. 1998.

Sheremata J. M. “Quantitative Molecular Representation and Sequential Optimization of Athabasca Asphaltenes”. Ongoing Doctoral Thesis. University of Alberta. 2005.

Sheremata J. M. “Quantitative Molecular Representation and Sequential Optimization of Athabasca Asphaltenes”. Ongoing Doctoral Thesis. University of Alberta. 2002.

Soave, G. “Equilibrium constants from a modified Redlich-Kwong equation of state”. Chemical Engineering Science. Volume 27, number 6. 1197-1203. 1972.

Trassy L. “Une méthode predictive pour le calcul des propriétés thermodynamiques des mélanges complexes sous pression”. Doctoral Thesis. Université d’Aix-Marseille III. 1998.

Valderrama J.O. “The State of the Cubic Equations of State”. Ind. Eng. Chem. Res. Vol 42, 1603-1618. 2003.

van der Waals, J.D. “On the Continuity of the Gaseous and Liquid States”. Ph.D. Dissertation. Universiteit Leiden, Leiden, The Netherlands. 1873.

Yaws, C. L. “Yaws' handbook of thermodynamic and physical properties of chemical compounds [electronic resource]: physical, thermodynamic and transport properties for 5,000 organic chemical compounds”. UA Internet Access: http://www.knovel.com/knovel2/Toc.jsp?BookID=667 . Knovel. New York.. 2003.

Zhang, X. Ongoing Doctoral Thesis. University of Alberta. 2005.

Zou, X. Y. “Selective Removal of Inorganic Fine Solids, Heavy Metals and Sulfur from Bitumen/Heavy Oils”. PhD Thesis. Graduate Department of Chemical Engineering and Applied Chemistry. University of Toronto. 2003.

APPENDIX I. PROCEDURE TO RUN THE GC MODEL IN MATLAB

This Appendix presents the instructions required to run the different programs that are included in the GC model using the application of MATLAB. In order to avoid any inconsistency due to changes in different versions of MATLAB, the version 7 is recommended.

Four different programs were created and each one should be run independently. The four programs can be found on the attached CD inside the following folders:

1. Single Compounds
2. Multicomponent mixtures with known T_b
3. Multicomponent mixtures with unknown T_b
4. First estimates for pressure and K_i

The procedure to run each one of these programs is described next:

- **Single Compounds:** this program calculates vapour pressures and liquid densities of single compounds. To run the program the following procedure should be used:
    1. Open the MATLAB application
    2. On the main workspace window change the current directory folder to the Single Compounds folder

3. Introduce the input data for the GC model in the workspace window for the compound of interest. The molecular structure, boiling point temperature, the molecular weight and an initial guess for vapour pressure should be introduced. In addition, the temperature data points of interest for calculation must be introduced in vector form:

```
structure = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
tb = XXX
MW = XXX
po = XX
```

To = [T_1 T_2 …..T_n]

4. Run the main program as follows:

```
[a,b,m,T,pressure,densit]=mainprogram(structure,tb,MW,po,To)
```

5. The results for vapour pressure, liquid density as well as the value for the parameters a and b will be calculated. In addition, two graphs are generated plotting vapour pressure and density vs temperature

- **Multicomponent Mixtures with known T_b:** this program calculates vapour pressures and liquid densities of mixtures when the boiling point temperature of the components is known. To run the program the following procedure should be used:

  1. Open the MATLAB application
  2. On the main workspace window change the current directory folder to the Multicomponent Mixtures with known Tb folder

3. Introduce the input data for the GC model in the workspace window for the compound of interest. The molecular structure, boiling point temperature, molecular weight, interaction parameter (k_{ij}) and the liquid molar composition for the components is required. In addition, an initial guess for vapour pressure and equilibrium K_i values should be introduced (These values can be guessed using the program "First estimates for pressure and Ki"). Finally, the temperature data points of interest for calculation must be introduced in vector form:

$$\text{structure} = \begin{bmatrix} 0 \cdots 0 \\ \vdots \ \cdots \vdots \\ 0 \cdots 0 \end{bmatrix}_{nc \times 19}$$

$tb = [t_{b1}\ t_{b2}\ \ldots.\ t_{bnc}]$

$MW = [MW_1\ MW_2\ \ldots.\ MW_{nc}]$

$$k_{ij} = \begin{bmatrix} 0 \cdots 0 \\ \vdots \ \cdots \vdots \\ 0 \cdots 0 \end{bmatrix}_{nc\ x nc}$$

$x = [x_1\ x_2\ \ldots.\ x_{nc-1})$

po = XX

$K = [K_1\ K_2\ \ldots\ K_{nc}]$

$To = [T_1\ T_2\ \ldots..T_n]$

4. Run the main program as follows:

[pressure vapor_comp densit a_mix b_mix] = mixtures(structure,tb,MW,T,x,kij,po,K)

The results for vapour pressure, liquid density as well as the value for the parameters a and b will be calculated. In addition, two graphs are generated plotting vapour pressure and density vs temperature.

- **Multicomponent Mixtures with unknown T_b:** this program calculates vapour pressures and liquid densities of mixtures when the boiling point temperature of the components is unknown. To run the program the following procedure should be used:
    1. Open the MATLAB application
    2. On the main workspace window change the current directory folder to the Multicomponent Mixtures with unknown Tb folder
    3. Follow the same steps presented in the previous case (known T_b) except that the boiling point temperature for the components becomes a variable of the process and it should not be introduced.

- **First estimates for pressure and K_i:** this program estimates first estimates for equilibrium constants K_i values and vapour pressures. To run the program the following procedure should be used:
    1. Open the MATLAB application
    2. On the main workspace window change the current directory folder to the Stability Test folder
    3. Introduce the input data as follows: the molecular structure, molecular weight, interaction parameter (k_{ij}) and the liquid molar composition for the

components. In addition, a fixed value (any value, no need to be close to VLE boundary) for vapour pressure temperature should be introduced

$$\text{structure} = \begin{bmatrix} 0 \cdots 0 \\ \vdots \;\cdots \vdots \\ 0 \cdots 0 \end{bmatrix}_{\text{nc x 19}}$$

$MW = [MW_1\ MW_2\ \ldots.\ MW_{nc}]$

$$k_{ij} = \begin{bmatrix} 0 \cdots 0 \\ \vdots \;\cdots \vdots \\ 0 \cdots 0 \end{bmatrix}_{\text{nc xnc}}$$

$x = [x_1\ x_2\ \ldots.\ x_{nc-1})$

p = XX

T = XX

4. Run the main program as follows:

   `[K cont graphic_D]=Stability_test(structure,MW,T,p,x,kij)`

   The results for first estimates of K_i values will be calculated and displayed in vector form. In addition, one graph is generated presenting pressure versus D (Distance to the Gibbs Free energy surface). The points were D becomes zero will be the first estimates for pressure at the temperature specified.

APPENDIX II. MATLAB CODE FOR THE GC MODEL

This Appendix presents the MATLAB code for all the programs developed in the GC computational model.

SINGLE COMPOUNDS PROGRAM

```
function [a,b,m,T,pressure,densit]=mainprogram(structure,tb,MW,po,To)
   alpha=1;
   increment = 0;
   % This function estimates the EOS parameters
   [s_vol,b1,m,mass]= EOS_parameters(structure,MW);
   R = 83.14;  % bar(cm)^3/mol K
   p = 1;     % Guess for pressure in bar
   beta = 0.05; % Guess for Beta
   %a1=0; % Guess for a at boiling Temperature
   a1=7*tb^2*R;
   counter=0;
   c_T=0;
   counter2=1;
   counter3=0;
   Tol=1;
   x=1;
   p_old=1;
   T=tb;
   % This loop estimates "a" at boiling temperature
   while (abs(Tol)>1e-4 & counter2<100)
   % This loop estimates vapor pressure at a given "a"
      while (abs(x)>1e-6 & counter<100)
            % This function calculates the root of EOS
            [zmax,zmin,nph]=roots_1(T,p,R,a1,b1);
            if nph==1
               a1=0;
            end
            while (nph==1 & counter3<100)
                  a1 = a1 + 10000000;
                  counter3 = counter3 + 1;
                  [zmax,zmin,nph]=roots_1(T,p,R,a1,b1);
                  if counter3 ==99
                     error('max iterations while calculating vapor pressure for a given a_tb number of phases
1')
                  end
            end
             counter3=0;
            % This function calculates the fugacity of vapor and liquid gas
            [fugcoeff_l,fugcoeff_v]=fugacity(T,a1,b1,zmin,zmax,p,R,c_T);
            fug_l = exp(fugcoeff_l) * p;
```

```
        fug_v = exp(fugcoeff_v) * p;
        x = (fugcoeff_v - fugcoeff_l);
        p_new = abs(p / (zmax - zmin) * ((zmax - zmin) - (fugcoeff_v - fugcoeff_l))); % Updating pressure
        if abs(p-p_new)/p>1.0
          p_new=(p_new-p)/3+p;
        end
        p = p_new;
        if p < 0
         p = p_new + 0.1;
        end
        counter=counter+1;
        if counter ==100
          error('max iterations while calculating vapor pressure for a given a_tb')
        end
    end
  Tol= abs((p - 1.01325) / 1.01325);
    if counter2 == 1
      a_old = a1;
     %Updating temperature dependent parameter "a"
      a_new = a1 * (p / 1.01325) ^ beta;
      a1 = a_new;
      p_old=p;
     else
      beta = log(a_new / a_old) / log(p_old / p); % Updating Beta
      a_old = a1;
      a_new = a_new * (p / 1.01325) ^ beta;
      a1 = a_new;
      p_old=p;
     end
  counter2=counter2+1;
  if counter2 ==100
    error('max iterations while estimating a_tb')
  end
  x=1;
  counter=0;
  p=1;
  end
  a_tb = a1;
  b=b1;
  % Temperature data points for prediction
  T=To;
  %T(1)=500;
  %for i=2:14
  % T(i)=T(i-1)+20;
  %end
  % Function to calculate temperature dependent volume correction
  c_T=CT(m,tb,s_vol,T,b1);
  % Calculation of parameter a with Temperature
  a=a_estimation(a_tb,m,T,tb);
  p = po;
  % Function to calculate vapor pressure
  pressure=vapress(m,tb,s_vol,c_T,T,p,R,a,b);
  % Function to calculate density
  [vcorr,densit]=density(m,tb,s_vol,c_T,T,p,R,a,b,mass);
  subplot(2,2,[1 2]); plot(T,pressure);
```

```
  subplot(2,2,3); plot(T,densit);
```

```
function pressure=vapress(m,tb,s_vol,c_T,T,p,R,a,b)
n=length(T);
counter=0;
x=1;
for i=1:n
  while (abs(x)>0.0001 & counter<100)
    [zmax,zmin,nph]=roots_1(T(i),p,R,a(i),b);
      while (nph==1 & counter<300)
          p_new = p + 0.5;
          p_old = p;
          p_new = (p_old + p_new) / 2;
          p = p_new;
          [zmax,zmin,nph]=roots_1(T(i),p,R,a(i),b);
          counter=counter+1;
          if counter ==300
            error('max iterations while calculating vapor pressure 1 phase')
          end
      end
      if p < 0 Then
          p = p_new + 0.1;
      end
   [fugcoeff_l,fugcoeff_v]=fugacity(T(i),a(i),b,zmin,zmax,p,R,c_T(i));
   fug_l = exp(fugcoeff_l) * p;
   fug_v = exp(fugcoeff_v) * p;
   pressure(i)=p;
   x = (fugcoeff_v - fugcoeff_l);
   p_new = abs(p / (zmax - zmin) * ((zmax - zmin) - (fugcoeff_v - fugcoeff_l)));
   p=p_new;
       if p < 0
          p = p * -1;
       end
       if p < 0.000000001
          p = 100 * p;
       end
       counter=counter+1;
       if counter ==100
            error('max iterations while calculating vapor pressure 2 phases')
       end
  end
  x=1;
  counter=0;
end
```

```
function a=a_estimation(a_tb,m,T,tb)
% Two formulation are given by Coniglio et al. for T_dependent parameter a
    %%' a = a_tb * Exp(f1 * (1 - (T(I) / tb) ^ 0.4) - f2 * (1 - (T(I) / tb) ^ (1 / 0.4)))
n=length(T);
 for i=1:n
    a(i) = a_tb * exp(m / 0.5 * (1 - (T(i) / tb) ^ 0.5));
 end
```

```
function c_T=CT(m,tb,s_vol,T,b1)
   alpha1 = 1.89213 * 0.000001;
   alpha2 = -0.25116;
   beta1 = 2.20483 * 0.001;
   beta2 = -1.22706;
   alpha_b = 0.27468;
   beta_b = -50.9493;
   c_tb = alpha_b * tb + beta_b * m ^ 2 - s_vol;
   alpha0 = alpha1 * tb ^ 2 + alpha2;
   beta0 = beta1 * tb + beta2;
   n = length(T);
   for i=1:n
      y(i) = exp(1 - T(i) / tb);
      c_T(i) = c_tb * (1 + alpha0 * (1 - y(i)) + beta0 * (1 - y(i)) ^ 2) + (-1 + sqrt(2)) * b1;
   end
```

```
function [vcorr,densit]=density(m,tb,s_vol,c_T,T,p,R,a,b,mass)
n=length(T);
counter=0;
x=1;
R = 83.14;
for i=1:n
   while (abs(x)>0.0001 & counter<100)
     [zmax,zmin,nph]=roots_1(T(i),p,R,a(i),b);
        while nph==1
             p_new = p + 0.05;
             p_old = p;
             p_new = (p_old + p_new) / 2;
             p = p_new;
             [zmax,zmin,nph]=roots_1(T(i),p,R,a(i),b);
             counter=counter+1;
             if counter ==499
                error('max iterations while calculating density')
             end
        end
        if p < 0 Then
             p = p_new + 0.1;
        end
    [fugcoeff_l,fugcoeff_v]=fugacity(T(i),a(i),b,zmin,zmax,p,R,c_T(i));
    fug_l = exp(fugcoeff_l) * p;
    fug_v = exp(fugcoeff_v) * p;
    pressure(i)=p;
    x = (fugcoeff_v - fugcoeff_l);
    p_new = abs(p / (zmax - zmin) * ((zmax - zmin) - (fugcoeff_v - fugcoeff_l)));
    p=p_new;
         if p < 0
             p = p * -1;
         end
         if p < 0.000000001
             p = 100 * p;
         end
         counter=counter+1;
         if counter ==499
                error('max iterations while calculating density')
         end
```

```
   end
   x=1;
   counter=0;
   %vcorr = zmin * R * T(i) / pressure(i)-c_T(i);
   vcorr=zmin * R * T(i) / pressure(i);
   densit(i) = mass / vcorr;
end
```

```
function [s_vol,b1,m,mass]= EOS_parameters(structure,MW)

vand_volume=[13.67 10.23 6.78 3.33 10.23 6.78 3.33 6.78 3.33 8.06 5.54 4.74 11.94 8.47 5.01 6.96 8.47 8.05 11.55];

shape_factor=[0.085492 0.082860 0.047033 -0.028020 0.062716 0.034236 -0.010213 0.010039 0.051147 0.050476 0.071528 0.013697 0.069836 0.060085 0.112156 0.04963 0.092399 0.141491 0.138136];

c_correct=[35.9209 13.2044 -11.4445 -36.3578 13.2044 -11.4445 -36.3578 -11.4445 -4.2938 19.3874 -4.2938 7.4171 13.2044 -11.4445 -36.3578 -36.3578 -11.4445 -36.3578 -11.4445];
%c_correct=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
sum1=0;
sum2=0;
sum4=0;
for i=1:19
   sum1=sum1+vand_volume(i)*structure(i);
   sum2=sum2+shape_factor(i)*structure(i);
   sum4=sum4+c_correct(i)*structure(i);
end
s_vol=sum4;
b1=sum1*26.8/17.2;
%Coniglio original expression for parameter m:
%m=0.30048+sum2-3.03178*log(1+0.08425*sum2^2);
%Trassy new expression for parameter m:
m=0.23269*sum2^((1/(sum2^0.6))+0.5)+0.08781*log(sum2)+0.59180;
mass=MW;
```

```
function [fugcoeff_l,fugcoeff_v]=fugacity(T,a1,b1,zmin,zmax,p,R,c_T)
   aa = a1 * p / (R ^ 2 * T ^ 2);
   bb = b1 * p / (R * T);
   term1 = zmax - 1;
   term2 = log(zmax - bb);
   term3 = aa / (2 * sqrt(2) * bb);
   term4 = (zmax + 2.414 * bb) / (zmax - 0.414 * bb);
   fugcoeff_v = term1 - (term2) - term3 * log(term4) - p * c_T / (R * T);
   term1 = zmin - 1;
   term2 = zmin - bb;
   term3 = aa / (2 * sqrt(2) * bb);
   term4 = zmin + 2.414 * bb;
   term5 = zmin - 0.414 * bb;
   fugcoeff_l = term1 - log(term2) - term3 * log(term4 / term5) - p * c_T / (R * T);
```

```
function [zmax,zmin,nph]= roots_1(T,p,R,a1,b1)
   Pi = 3.14159265358979;
   aa = a1 * p / (R ^ 2 * T ^ 2);
```

```
   bb = b1 * p / (R * T);
   co(1) = 1;
   co(2) = -(1 - bb);
   co(3) = aa - 3 * bb ^ 2 - 2 * bb;
   co(4) = -(aa * bb - bb * bb - bb * bb * bb);
   x=sym('x');
   s=solve(co(1)*x^3+co(2)*x^2+co(3)*x+co(4));
   r(1)=eval(s(1));
   r(2)=eval(s(2));
   r(3)=eval(s(3));
   n=0;
   for i=1:3
      test=imag(r(i));
      test2=real(r(i));
      if (abs(test)<=1e-8 & test2>=0 & test2>=bb)
         n=n+1;
         y(n)=real(r(i));
      end
   end
   if n==1
      nph=1;
      zmax=y(n);
      zmin=zmax;
   end
   if n>1
      nph=2;
      zmax=max(y);
      zmin=min(y);
   end
```

MULTICOMPONENT MIXTURES WITH KNOWN Tb PROGRAM

```
function [pressure vapor_comp densit a_mix b_mix]=mixtures(structure,tb,MW,T,x,kij,po,K)
   [n,o]=size(structure);
   [p,q]=size(x);
   %Adding the n vlue of composition in x:
   sum=0;
   for i=1:p
      for j=1:q
         sum=sum+x(i,j);
      end
      x(i,q+1)=1-sum;
      sum=0;
   end
   R=83.14;   % bar(cm)^3/mol K
   for i=1:n
      % This function estimates the EOS parameters
      [s_vol(i),bi(i),m(i)]= EOS_parameters(structure(i,:));
   end
   % Estimation of parameter "a" at boiling point Temperature (a_tb):
   for i=1:n
      [a_tb(i)]=a_tbf(structure(i,:),tb(i),bi(i));
   end
```

```
   % Calculation of parameter a for pure components:
   for i=1:n
      a_i(:,i)=a_estimation(a_tb(i),m(i),T,tb(i));
   end
   % Function to calculate vapor pressure:
   [pressure vapor_comp]=vapress(m,tb,T,R,n,x,a_i,bi,kij,po,K);
   % Parameters of the liquid mixture:
   for i=1:p
      [a b]=parameter_a_b(x(i,:),a_i(i,:),bi,kij);
      a_mix(i)=a;
      b_mix(i)=b;
   end
   % Function to calculate density
   [vcorr,densit]=density(m,tb,s_vol,c_T,T,p,R,a,b,mass);
   %plotting:
   subplot(2,2,[1 2]); plot(T,pressure);
   subplot(2,2,[3 4]); plot(T,densit);
```

```
function [pressure vapor_comp]=vapress(m,tb,T,R,nc,x,a_i,bi,kij,p,K)
n=length(T);
[c,o]=size(x);
counter=0;
cont=0;
G=1;
%First guess of pressure:
%p=1;
% First guess of equilibrium values Ki=Yi/xi
      %for k=1:nc
         %K=[0.001          10];
      %end
      lnKi_old=log(K(:));
% Repeat vapor pressure calculation for n values of Temperature and
% compositions:
for i=1:n
   while (norm(G)>0.000001 & counter<300)
      %Calculate vapor compositions:
         sum_Y=0;
         for j=1:nc
            Y(i,j)=K(j)*x(i,j);
            sum_Y=sum_Y+Y(i,j);
         end
         for j=1:nc
            y(i,j)=Y(i,j)/sum_Y;
         end
         if counter==0
            % Evaluate function G:
            G_old=Eval_G(x(i,:),y(i,:),T(i),p,R,a_i(i,:),bi,K,kij);
         end
       % Solve for Pressure:
       % Method: Non Linear Optimization Toolbox (fsolve)
        global f_x f_y T_i f_a_i f_bi f_kij R
        f_x=x(i,:);
        f_y=y(i,:);
        T_i=T(i);
        f_a_i=a_i(i,:);
```

```
f_bi=bi;
f_kij=kij;
options=optimset('TolFun',1e-6);
p_new=fsolve(@Eval_f,p);
clear global f_x f_y T_i f_a_i f_bi f_kij
p=p_new;

% Evaluate function G:
 G=Eval_G(x(i,:),y(i,:),T(i),p,R,a_i(i,:),bi,K,kij);

% Update Ki values with QNSS algorithm:
if counter==0
    % Parameters of the liquid mixture:
    [a b]=parameter_a_b(x(i,:),a_i(i,:),bi,kij);
    [zmax,zmin,nph]=roots_1(T(i),p,R,a,b);
    if nph==2
        %Gibbs free energy:
        %Parameters of the liquid mixture:
        A = a * p / (R ^ 2 * T(i) ^ 2);
        B = b * p / (R * T(i));
        dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
        dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
        f=min(dG_RT(1),dG_RT(2));
        if f==dG_RT(1)
            z_l=zmin;
        else
            z_l=zmax;
        end
    else
        z_l=zmin;
    end
    % Estimate fugacity of Liquid
    fugcoeff_l=exp(fugacity_l(T(i),a,b,z_l,p,R,x(i,:),a_i(i,:),bi,kij));
    % Parameters of the vapor mixture:
    [a b]=parameter_a_b(y(i,:),a_i(i,:),bi,kij);
    [zmax,zmin,nph]=roots_1(T(i),p,R,a,b);
    if nph==2
        %Gibbs free energy:
        %Parameters of the vapor mixture:
        A = a * p / (R ^ 2 * T(i) ^ 2);
        B = b * p / (R * T(i));
        dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
        dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
        f=min(dG_RT(1),dG_RT(2));
        if f==dG_RT(1)
            z_v=zmin;
        else
            z_v=zmax;
        end
    else
        z_v=zmin;
    end
```

```
            % Estimate fugacity of Vapor
            fugcoeff_v=exp(fugacity_v(T(i),a,b,z_v,p,R,y(i,:),a_i(i,:),bi,kij));
            for k=1:nc
               lnKi(k)=log(fugcoeff_l(k)/fugcoeff_v(k));
               delta(k)=lnKi(k)-lnKi_old(k);
            end
         else
            if counter==1
                sigma=-delta*G_old'/(delta*(G-G_old)');
            else
               sigma=-delta*G_old'/(delta*(G-G_old)')*sigma;
            end
          if abs(sigma)>=30
             sigma=30;
          end
           delta2=-sigma*G;
           lnKi_old=lnKi;
           lnKi=delta2+lnKi;
           delta=lnKi-lnKi_old;
         end
         K(:)=exp(lnKi);
       % Store G value
        G_old=G;
        counter=counter+1;
        if counter==299
         error('max iterations while solving for Ki')
        end
   end
   G=1;
   pressure(i)=p;
   for j=1:nc
      Y(i,j)=K(j)*x(i,j);
      y(i,j)=Y(i,j)/sum(Y(i,:));
      vapor_comp(i,j)=y(i,j);
   end
   counter=0;
end
```

```
function a=a_estimation(a_tb,m,T,tb)
% Two formulation are given by Coniglio et al. for T_dependent parameter a
      %%' a = a_tb * Exp(f1 * (1 - (T(I) / tb) ^ 0.4) - f2 * (1 - (T(I) / tb) ^ (1 / 0.4)))
n=length(T);
  for i=1:n
      a(i) = a_tb * exp(m / 0.5 * (1 - (T(i) / tb) ^ 0.5));
  end
```

```
function [a_tb]=a_tbf(structure,tb,b1)
   alpha=1;
   increment = 0;
   R = 83.14;  % bar(cm)^3/mol K
   p = 1;     % Guess for pressure in bar
   beta = 0.05; % Guess for Beta
   % Guess for a at boiling Temperature
   a1=0;
```

```
  % a1=7*tb^2*R;
  counter=0;
  c_T=0;
  counter2=1;
  counter3=0;
  Tol=1;
  x=1;
  p_old=1;
  T=tb;
  % This loop estimates "a" at boiling temperature
  while (abs(Tol)>1e-4 & counter2<100)
  % This loop estimates vapor pressure at a given "a"
    while (abs(x)>1e-6 & counter<100)
        % This function calculates the root of EOS
        [zmax,zmin,nph]=roots_1(T,p,R,a1,b1);
        while (nph==1 & counter3<1000)
          if tb<300
            da1=100000;
          else
            da1=1000000;
          end
             a1 = a1 + da1;
             counter3 = counter3 + 1;
             [zmax,zmin,nph]=roots_1(T,p,R,a1,b1);
             if counter3 ==999
               error('max iterations while calculating vapor pressure for a given a_tb number of phases
1')
             end
        end
         counter3=0;
        % This function calculates the fugacity of vapor and liquid gas
        [fugcoeff_l,fugcoeff_v]=fugacity(T,a1,b1,zmin,zmax,p,R,c_T);
        fug_l = exp(fugcoeff_l) * p;
        fug_v = exp(fugcoeff_v) * p;
        x = (fugcoeff_v - fugcoeff_l);
        p_new = abs(p / (zmax - zmin) * ((zmax - zmin) - (fugcoeff_v - fugcoeff_l))); % Updating
pressure
        if abs(p-p_new)/p>1.0
          p_new=(p_new-p)/4+p;
        end
        p = p_new;
        if p < 0
         p = p_new + 0.1;
        end
        counter=counter+1;
        if counter ==100
          error('max iterations while calculating vapor pressure for a given a_tb')
        end
    end
  Tol= abs((p - 1.01325) / 1.01325);
    if counter2 == 1
      a_old = a1;
      %Updating temperature dependent parameter "a"
      a_new = a1 * (p / 1.01325) ^ beta;
      a1 = a_new;
      p_old=p;
```

```
    else
      beta = log(a_new / a_old) / log(p_old / p); % Updating Beta
      a_old = a1;
      a_new = a_new * (p / 1.01325) ^ beta;
      a1 = a_new;
      p_old=p;
    end
  counter2=counter2+1;
  if counter2 ==100
    error('max iterations while estimating a_tb')
  end
  x=1;
  counter=0;
  p=1;
  end
  a_tb = a1;
```

```
function [s_vol,b1,m]= EOS_parameters(structure)

vand_volume=[13.67 10.23 6.78 3.33 10.23 6.78 3.33 6.78 3.33 8.06 5.54 4.74 11.94 8.47 5.01 6.96 8.47
8.05 11.55];

shape_factor=[0.085492 0.082860 0.047033 -0.028020 0.062716 0.034236 -0.010213 0.010039 0.051147
0.050476 0.071528 0.013697 0.069836 0.060085 0.112156 0.04963 0.092399 0.141491 0.138136];

c_correct=[35.9209 13.2044 -11.4445 -36.3578 13.2044 -11.4445 -36.3578 -11.4445 -4.2938 19.3874 -
4.2938 7.4171 13.2044 -11.4445 -36.3578 -36.3578 -11.4445 -36.3578 -11.4445];
%c_correct=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
sum1=0;
sum2=0;
sum4=0;
for i=1:19
  sum1=sum1+vand_volume(i)*structure(i);
  sum2=sum2+shape_factor(i)*structure(i);
  sum4=sum4+c_correct(i)*structure(i);
end
s_vol=sum4;
b1=sum1*26.8/17.2;
%Coniglio original expression for parameter m:
%m=0.30048+sum2-3.03178*log(1+0.08425*sum2^2);
%Trassy new expression for parameter m:
m=0.23269*sum2^((1/(sum2^0.6))+0.5)+0.08781*log(sum2)+0.59180;
```

```
function [f]=Eval_f(p)

% This function evaluates the distance from the Gibbs free energy surface
% to the tangent plane at certain composition in order to solve for
% pressure:
global f_x f_y T_i f_a_i f_bi f_kij R
x=f_x;
y=f_y;
T=T_i;
a_i=f_a_i;
bi=f_bi;
```

```
kij=f_kij;

nc=length(bi);
% Parameters of the liquid mixture:
    [a b]=parameter_a_b(x,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the liquid mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_l=zmin;
      else
         z_l=zmax;
      end
    else
       z_l=zmin;
    end
% Estimate fugacity of Liquid
  fugcoeff_l=fugacity_l(T,a,b,z_l,p,R,x,a_i,bi,kij);
% Parameters of the vapor mixture:
    [a b]=parameter_a_b(y,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the vapor mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_v=zmin;
      else
         z_v=zmax;
      end
    else
      z_v=zmin;
    end
% Estimate fugacity of Vapor
  fugcoeff_v=fugacity_v(T,a,b,z_v,p,R,y,a_i,bi,kij);
% Calculate absolute fugacities:
  for z=1:nc
    fug_l(z) = exp(fugcoeff_l(z)) *x(z)* p;
    fug_v(z) = exp(fugcoeff_v(z)) * y(z)*p;
  end
  sum_D=0;
  for i=1:nc
     sum_D=sum_D+y(i)*log(fug_v(i)/fug_l(i));
  end
  f=sum_D;
```

```
function [G]=Eval_G(x,y,T,p,R,a_i,bi,K,kij);
% This function evaluates the G expression for all components:

nc=length(bi);
% Parameters of the liquid mixture:
    [a b]=parameter_a_b(x,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the liquid mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_l=zmin;
       else
         z_l=zmax;
      end
    else
       z_l=zmin;
    end
% Estimate fugacity of Liquid
  fugcoeff_l=exp(fugacity_l(T,a,b,z_l,p,R,x,a_i,bi,kij));
% Parameters of the vapor mixture:
    [a b]=parameter_a_b(y,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the vapor mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_v=zmin;
      else
         z_v=zmax;
      end
    else
      z_v=zmin;
    end
% Estimate fugacity of Vapor
  fugcoeff_v=exp(fugacity_v(T,a,b,z_v,p,R,y,a_i,bi,kij));
  for i=1:nc
      G(i)=log(K(i))+log(fugcoeff_v(i))-log(fugcoeff_l(i));
  End
```

```
function [fugcoeff_l,fugcoeff_v]=fugacity(T,a1,b1,zmin,zmax,p,R,c_T)
   aa = a1 * p / (R ^ 2 * T ^ 2);
   bb = b1 * p / (R * T);
   term1 = zmax - 1;
```

```
   term2 = log(zmax - bb);
   term3 = aa / (2 * sqrt(2) * bb);
   term4 = (zmax + 2.414 * bb) / (zmax - 0.414 * bb);
   fugcoeff_v = term1 - (term2) - term3 * log(term4) - p * c_T / (R * T);
   term1 = zmin - 1;
   term2 = zmin - bb;
   term3 = aa / (2 * sqrt(2) * bb);
   term4 = zmin + 2.414 * bb;
   term5 = zmin - 0.414 * bb;
   fugcoeff_l = term1 - log(term2) - term3 * log(term4 / term5) - p * c_T / (R * T);
```

```
function [fugcoeff_l]=fugacity_l(T,a,b,zmin,p,R,x,a_i,bi,kij)
   % Estimation of liquid fugacity for each component
   % Number of components:
   nc=length(bi);
   %Parameters of the mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
   %Parameters of each component:
   for i=1:nc
      Bi(i)=bi(i)*p/(R*T);
   end
   sum=0;
   for i=1:nc
      for j=1:nc
         Aij(i,j)=(a_i(i)*a_i(j))^(0.5)*(1-kij(i,j))*p/((R*T)^2);
         sum=sum+x(j)*Aij(i,j);
      end
      sum_xj_Aij(i)=sum;
      sum=0;
   end
   %Computing fugacity of each component in the liquid mixture:
   for i=1:nc
      term1(i) = Bi(i)/B*(zmin - 1);
      term2(i) = log(zmin - B);
      term3(i) = A / (2 * sqrt(2) * B)*(2*sum_xj_Aij(i)/A-Bi(i)/B);
      term4(i) = (zmin + 2.414 * B) / (zmin - 0.4142 * B);
      fugcoeff_l(i) = term1(i) - term2(i) - term3(i) * log(term4(i));
   end
nc=1;
```

```
function [fugcoeff_v]=fugacity_v(T,a,b,zmax,p,R,y,a_i,bi,kij)
   % Estimation of liquid fugacity for each component in the vapor phase
   % Number of components:
   nc=length(bi);
   %Parameters of the mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
   %Parameters of each component:
   for i=1:nc
      Bi(i)=bi(i)*p/(R*T);
   end
   sum=0;
   for i=1:nc
```

```
      for j=1:nc
         Aij(i,j)=(a_i(i)*a_i(j))^(0.5)*(1-kij(i,j))*p/(R^2*T^2);
         sum=sum+y(j)*Aij(i,j);
      end
      sum_yj_Aij(i)=sum;
      sum=0;
   end
   %Computing fugacity of each component in the liquid mixture:
   for i=1:nc
      term1(i) = Bi(i)/B*(zmax - 1);
      term2(i) = log(zmax - B);
      term3(i) = A / (2 * sqrt(2) * B)*(2*sum_yj_Aij(i)/A-Bi(i)/B);
      term4(i) = (zmax + 2.414 * B) / (zmax - 0.414 * B);
      fugcoeff_v(i) = term1(i) - term2(i) - term3(i) * log(term4(i));
   end
   nc=1;
```

```
function [J]=Jac(x,y,T,p,R,a_i,bi,kij)

% Evaluation of the Jacobian of the distance from the Gibbs free energy surface
% to the tangent plane at certain composition in order to solve for
% pressure:

% Number of components:
   nc=length(bi);
 % Parameters of the liquid mixture:
    [a b]=parameter_a_b(x,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the liquid mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_l=zmin;
       else
         z_l=zmax;
      end
    else
       z_l=zmin;
       A = a * p / (R ^ 2 * T ^ 2);
       B = b * p / (R * T);
    end
% Evaluation of constant Mi 1) Liquid phase:
for i=1:nc
   sum_xj=0;
   for j=1:nc
      sum_xj=sum_xj+x(j)*(a_i(i)*a_i(j))^0.5*(1-kij(i,j));
   end
   M_l(i)=A/(2*sqrt(2)*B)*(2/a*sum_xj-bi(i)/b);
end
a_l=a;
```

```
b_l=b;
A_l=A;
B_l=B;
% Parameters of the vapor mixture:
   [a b]=parameter_a_b(y,a_i,bi,kij);
   [zmax,zmin,nph]=roots_1(T,p,R,a,b);
   if nph==2
    %Gibbs free energy:
    %Parameters of the vapor mixture:
    A = a * p / (R ^ 2 * T ^ 2);
    B = b * p / (R * T);
    dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
    dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
    f=min(dG_RT(1),dG_RT(2));
    if f==dG_RT(1)
      z_v=zmin;
    else
      z_v=zmax;
    end
   else
    z_v=zmin;
    A = a * p / (R ^ 2 * T ^ 2);
    B = b * p / (R * T);
   end
% Evaluation of constant Mi 2) Vapor phase:
for i=1:nc
  sum_yj=0;
  for j=1:nc
    sum_yj=sum_yj+y(j)*(a_i(i)*a_i(j))^0.5*(1-kij(i,j));
  end
  M_v(i)=A/(2*sqrt(2)*B)*(2/a*sum_yj-bi(i)/b);
end
a_v=a;
b_v=b;
A_v=A;
B_v=B;

% Evaluation of the derivative dzdp:
for k=1:2
  if k==1
    z=z_l;
    a=a_l;
    b=b_l;
  else
    z=z_v;
    a=a_v;
    b=b_v;
  end
  v=R*T*z/p;
  dvdp(k)=-R*T/((v-b)^2)+2*a*(v+b)/((v^2+2*b*v-b^2)^2);
  dzdp(k)=1/(R*T)*(v+1/dvdp(k)*p);
end

%Evaluation of the Derivative dBdp:
dBdp_l=b_l/(R*T);
dBdp_v=b_v/(R*T);
```

```
%Evaluation of the derivatives of the fugacities:
for i=1:2
   %Evaluation of dCdp for each phase:
   if i==1
      z=z_l;
      M=M_l;
      c=x;
      dBdp=dBdp_l;
      a_l=a;
      b_l=b;
      A_l=A;
      B_l=B;
   else
      z=z_v;
      M=M_v;
      c=y;
      dBdp=dBdp_v;
      a=a_v;
      b=b_v;
      A=A_v;
      B=B_v;
   end
   for j=1:nc
      C(i,j)=bi(j)/b*(z-1)-log(z-B)-M(j)*log((z+(1+sqrt(2))*B)/(z+(1-sqrt(2))*B));
      dCdp(i,j)=bi(j)/b*dzdp(i)-1/(z-B)*(dzdp(i)-dBdp)-M(j)*(z+(1-sqrt(2))*B)/(z+(1+sqrt(2))*B)*(-
2*sqrt(2)*dzdp(i)*B+2*sqrt(2)*dBdp*(z+B))/((z+(1-sqrt(2))*B)^2);
      dfidp(i,j)=c(j)*exp(C(i,j))+c(j)*p*exp(C(i,j))*dCdp(i,j);
   end

end
% Evaluation of Fugacities fi_v and fi_l:
   % fugacity of Liquid
      fugcoeff_l=fugacity_l(T,a_l,b_l,z_l,p,R,x,a_i,bi,kij);
   % fugacity of Vapor
      fugcoeff_v=fugacity_v(T,a_v,b_v,z_v,p,R,y,a_i,bi,kij);
   % Calculate absolute fugacities:
   for i=1:nc
    fug_l(i) = exp(fugcoeff_l(i)) *x(i)* p;
    fug_v(i) = exp(fugcoeff_v(i)) * y(i)*p;
   end

% Evaluation of dfdp:
sum_dfdp=0;
for i=1:nc
   sum_dfdp=sum_dfdp+y(i)*(1/fug_v(i)*dfidp(2,i)-1/fug_l(i)*dfidp(1,i));
end
J=sum_dfdp;
```

```
function [a b]=parameter_a_b(x_vector,a_i,bi,kij)
% This function estimates the parameters a and b for the mixture for a given
% Temperature and composition:
[o,p]=size(a_i);
x=x_vector;
```

```
% Parameter a:
a_sum1=0;
for i=1:p
   for j=1:p
      a_sum1=x(i)*x(j)*(a_i(i)*a_i(j))^0.5*(1-kij(i,j))+a_sum1;
   end
end
a=a_sum1;
% Parameter b:
sumbi=0;
for i=1:p
      sumbi=x(i)*bi(i)+sumbi;
end
b=sumbi;
```

```
function [zmax,zmin,nph]= roots_1(T,p,R,a1,b1)
   Pi = 3.14159265358979;
   aa = a1 * p / (R ^ 2 * T ^ 2);
   bb = b1 * p / (R * T);
   co(1) = 1;
   co(2) = -(1 - bb);
   co(3) = aa - 3 * bb ^ 2 - 2 * bb;
   co(4) = -(aa * bb - bb * bb - bb * bb * bb);
   x=sym('x');
   s=solve(co(1)*x^3+co(2)*x^2+co(3)*x+co(4));
   r(1)=eval(s(1));
   r(2)=eval(s(2));
   r(3)=eval(s(3));
   n=0;
   for i=1:3
      test=imag(r(i));
      test2=real(r(i));
      if (abs(test)<=1e-8 & test2>=0 & test2>=bb)
         n=n+1;
         y(n)=real(r(i));
      end
   end
   if n==1
      nph=1;
      zmax=y(n);
      zmin=zmax;
   end
   if n>1
      nph=2;
      zmax=max(y);
      zmin=min(y);
   end
```

MULTICOMPONENT MIXTURES WITH UNKNOWN Tb PROGRAM

Only the functions that differ from the previous version are presented here.

```
function [a_mix b_mix ]=mixtures(structure,MW,T,x,kij,po,K)
```

```
[n,o]=size(structure);
[p,q]=size(x);
Tn=length(T);
%Adding the n vlue of composition in x:
sum=0;
for i=1:p
    for j=1:q
        sum=sum+x(i,j);
    end
    x(i,q+1)=1-sum;
    sum=0;
end
R=83.14;   % bar(cm)^3/mol K
for i=1:n
    % This function estimates the EOS parameters
    [s_vol(i),bi(i),m(i),tb(i)]= EOS_parameters(structure(i,:));
end
% Estimation of parameter "a" at boiling point Temperature (a_tb):
for i=1:n
    [a_tb(i)]=a_tbf(structure(i,:),tb(i),bi(i));
end
% Calculation of parameter a for pure components:
for i=1:n
    a_i(:,i)=a_estimation(a_tb(i),m(i),T,tb(i));
end
% Function to calculate vapor pressure and volume:
[pressure,vapor_comp,volume]=vapress(m,tb,T,R,n,x,a_i,bi,kij,po,K);
% Function to calculate temperature dependent volume correction
for i=1:n
c_T(:,i)=CT(m(i),tb(i),s_vol(i),T,bi(i));
end
% Calculation of Density
MW_mixture=0;
for q=1:n
    MW_mixture=MW_mixture +MW(q)*x(1,q);
end
for i=1:Tn
    c_T_mixture(i)=0;
    for j=1:n
        c_T_mixture(i)=c_T_mixture(i)+x(i,j)*c_T(i,j);
    end
end

% Parameters of the liquid mixture:
for i=1:p
    [a b]=parameter_a_b(x(i,:),a_i(i,:),bi,kij);
    a_mix(i)=a;
    b_mix(i)=b;
end
for i=1:Tn
   vcorrected(i)=volume(i)-c_T_mixture(i);
end
for i=1:Tn
density_corr(i)=MW_mixture/vcorrected(i);
density(i)=MW_mixture/volume(i);
end
```

```
    density_corr=density_corr';
    density=density';
    pressure=pressure';
    %subplot(2,2,[1 2]); plot(T,pressure);
    %subplot(2,2,3); plot(T,densit);
    %subplot(2,2,4); plot(T,delta);
```

```
function [s_vol,b1,m,tb]= EOS_parameters(structure)

vand_volume=[13.67 10.23 6.78 3.33 10.23 6.78 3.33 6.78 3.33 8.06 5.54 4.74 11.94 8.47 5.01 6.96 8.47 8.05 11.55];

shape_factor=[0.085492 0.082860 0.047033 -0.028020 0.062716 0.034236 -0.010213 0.010039 0.051147 0.050476 0.071528 0.013697 0.069836 0.060085 0.112156 0.04963 0.092399 0.141491 0.138136];

c_correct=[35.9209 13.2044 -11.4445 -36.3578 13.2044 -11.4445 -36.3578 -11.4445 -4.2938 19.3874 -4.2938 7.4171 13.2044 -11.4445 -36.3578 -36.3578 -11.4445 -36.3578 -11.4445];
%c_correct=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

t_b=[0.803931 1.77186 2.56070 3.32912 1.77186 2.56070 3.32912 2.56070 3.06025 1.39979 3.06025 3.57473 1.77186 2.56070 3.32912 3.32912 2.56070 3.32912 2.56070];
sum1=0;
sum3=0;
sum2=0;
sum4=0;
for i=1:19
    sum1=sum1+vand_volume(i)*structure(i);
    sum2=sum2+shape_factor(i)*structure(i);
    sum3=sum3+t_b(i)*structure(i);
    sum4=sum4+c_correct(i)*structure(i);
end
s_vol=sum4;
s_tb=sum3;
b1=sum1*26.8/17.2;
%Coniglio original expression for parameter m:
%m=0.30048+sum2-3.03178*log(1+0.08425*sum2^2);
%Trassy new expression for parameter m:
m=0.23269*sum2^((1/(sum2^0.6))+0.5)+0.08781*log(sum2)+0.59180;
tb=258.257+49.6530*(log(s_tb))^2-1.35746*(2-sqrt(2))*b1/m;
```

FIRST ESTIMATES FOR PRESSURE AND Ki

Only the new functions introduced are shown next.

```
function [K cont graphic_D]=Stability_test(structure,MW,T,p,x,kij)

% Determination of first guess of Ki values using the Stability Test
% Calculation (Michelsen method)

% Estimation of EOS parameters:
[n,o]=size(structure);
```

```
[r,q]=size(x);
%Adding the n vlue of composition in x:
sum=0;
for i=1:r
   for j=1:q
       sum=sum+x(i,j);
   end
   x(i,q+1)=1-sum;
   sum=0;
end
R=83.14;   % bar(cm)^3/mol K
for i=1:n
 % This function estimates the EOS parameters
 [s_vol(i),bi(i),m(i),tb(i)]= EOS_parameters(structure(i,:));
end
% Estimation of parameter "a" at boiling point Temperature (a_tb):
for i=1:n
 [a_tb(i)]=a_tbf(structure(i,:),tb(i),bi(i));
end
% Calculation of parameter a for pure components:
for i=1:n
 a_i(:,i)=a_estimation(a_tb(i),m(i),T,tb(i));
end

% Parameters of the mixture:
[a b]=parameter_a_b(x,a_i,bi,kij);

% First estimates for Ki:
nc=n;
for i=1:nc
   Ki(i)=0.5;
end

% Stability Test Calculation for Ki(o):
for i=1:nc
    Y(i)=Ki(i)*x(i);
end
[zmax,zmin,nph]=roots_1(T,p,R,a,b);
[fugcoeff_l]=exp(fugacity_l(T,a,b,zmin,p,R,x,a_i,bi,kij));
for i=1:nc
    h(i)=log(x(i))+log(fugcoeff_l(i));
end
cont=0;
sumY=0;
k=1;
while (norm(k)>= 1e-6 & cont<100)
        for i=1:nc
            sumY=sumY+Y(i);
        end
        for i=1:nc
            y(i)=Y(i)/sumY;
        end
        sumY=0;
        % Parameters of the mixture:
        [a b]=parameter_a_b(y,a_i,bi,kij);
        [zmax,zmin,nph]=roots_1(T,p,R,a,b);
```

```
      if nph==2
         cont=100;
      end
      [fugcoeff_v]=exp(fugacity_v(T,a,b,zmax,p,R,y,a_i,bi,kij));
      for i=1:nc
         k(i)=log(Y(i))+log(fugcoeff_v(i))-h(i);
         Yi_new(i)=exp(h(i)-log(fugcoeff_v(i)));
      end
      Y=Yi_new;
      cont=cont+1;
end
for i=1:nc
   K(i)=Y(i)/x(i);
end


% First Estimate evaluation for pressure:
sum_Y=[0,0];
for j=1:nc
   Y(1,j)=K(j)*x(j);
   Y(2,j)=(1/K(j))*x(j);
   sum_Y(1)=sum_Y(1)+Y(1,j);
   sum_Y(2)=sum_Y(2)+Y(2,j);
end

%Calculate vapour compositions:
for i=1:2
  for j=1:nc
     y(i,j)=Y(i,j)/sum_Y(i);
  end
end
 %Dimenssionless distance from the Gibbs free energy surface to the tangent
 %plane at x for Y1 and Y2:

sumD=0;
p=0.1;
for i=1:2
for u=1:100
   p=p+0.1;
    % Parameters of the liquid mixture:
    [a b]=parameter_a_b(x,a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the liquid mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_l=zmin;
         z_v=zmax;
      else
         z_l=zmax;
         z_v=zmin;
```

```
      end

    else
       z_l=zmin;
       z_v=zmin;
    end
    [fugcoeff_l]=exp(fugacity_l(T,a,b,z_l,p,R,x,a_i,bi,kij));
    % Parameters of the vapor mixture:
    [a b]=parameter_a_b(y(i,:),a_i,bi,kij);
    [zmax,zmin,nph]=roots_1(T,p,R,a,b);
    if nph==2
      %Gibbs free energy:
      %Parameters of the vapor mixture:
      A = a * p / (R ^ 2 * T ^ 2);
      B = b * p / (R * T);
      dG_RT(1)=zmin-1-log(zmin-B)-A/(B*sqrt(8))*log((zmin+(1+sqrt(2))*B)/(zmin+(1-sqrt(2))*B));
      dG_RT(2)=zmax-1-log(zmax-B)-A/(B*sqrt(8))*log((zmax+(1+sqrt(2))*B)/(zmax+(1-sqrt(2))*B));
      f=min(dG_RT(1),dG_RT(2));
      if f==dG_RT(1)
         z_v=zmin;
      else
         z_v=zmax;
      end

    else
      z_v=zmin;
    end
    [fugcoeff_v]=exp(fugacity_v(T,a,b,z_v,p,R,y(i,:),a_i,bi,kij));
   for j=1:nc
      sumD=sumD+y(i,j)*log(y(i,j)*fugcoeff_v(j)/(x(j)*fugcoeff_l(j)));
   end
   graphic_D(u,1)=p;
   graphic_D(u,i+1)=sumD;
   sumD=0;
end
p=0.00001;
end
```

```
function [g_i] = gi(Ki,p,T,x,nc,R,a,b,a_i,bi,kij)
% Stability Test Calculation (Michelsen Method)

% Estimate vapor compositions yi:
for j=1:nc
   Y(j)=Ki(j)*x(j);
end
sumY=sum(Y);
for i=1:nc
   y(i)=Y(i)/sumY;
end

% Estimate Fugacities:
[zmax,zmin,nph]=roots_1(T,p,R,a,b);
[fugcoeff_l]=exp(fugacity_l(T,a,b,zmin,p,R,x,a_i,bi,kij));
[fugcoeff_v]=exp(fugacity_v(T,a,b,zmax,p,R,y,a_i,bi,kij));
```

```
% Estimate gi:
for i=1:nc
    g_i(i)=log(Ki(i))+log(fugcoeff_v(i))-log(fugcoeff_l(i));
end
```

APPENDIX III. MOLECULAR STRUCTURE OF AVR

This Appendix includes the molecular structure and input data for the GC model in the case of AVR using multi-molecule representations (Table III.1 and Table III.2). A sketch presenting all the molecules is included next (Sheremata, 2005).

Table III.1- Input Data for the GC Model. AVR (60 Molecules)

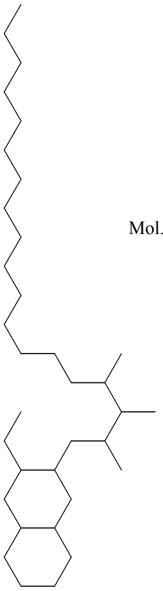
Molecule	MW	Structure																		
1	474.89	5	16	3	0	6	2	0	2	0	0	0	0	0	0	0	0	0	0	0
2	597.06	6	12	3	0	6	2	0	2	0	3	3	3	0	0	0	0	0	0	0
3	480.85	6	10	3	0	6	2	0	2	0	3	3	0	0	0	0	0	0	0	0
4	554.89	7	9	1	0	0	0	1	0	4	6	8	3	0	0	0	0	0	0	0
5	770.18	8	10	1	0	5	1	2	2	12	12	5	0	0	0	0	0	0	0	0
6	749.27	6	12	2	1	11	3	0	4	2	5	3	4	0	0	0	0	0	0	0
7	687.20	9	21	4	1	0	0	0	0	0	5	5	4	0	0	0	0	0	0	0
8	1082.76	8	19	2	0	11	3	1	4	6	9	8	6	0	0	0	0	0	0	0
9	719.28	12	12	8	0	6	2	0	2	0	2	4	3	0	0	0	0	0	0	0
10	512.83	7	10	1	0	0	0	0	0	0	6	8	5	0	0	0	0	0	0	0
11	985.81	7	25	5	0	22	10	0	2	0	0	0	0	0	0	0	0	0	0	0
12	1294.31	4	22	2	0	40	14	0	12	0	0	0	0	0	0	0	0	0	0	0
13	629.08	8	12	4	0	4	2	1	2	6	4	2	0	0	0	0	0	0	0	0
14	720.17	7	10	3	1	5	1	1	2	6	7	5	3	0	0	0	0	0	0	0
15	985.81	6	27	4	0	22	10	0	2	0	0	0	0	0	0	0	0	0	0	0
16	692.22	7	16	4	1	6	2	0	2	0	3	3	3	0	0	0	0	0	0	0
17	1075.81	8	21	2	2	13	5	0	2	2	9	7	6	0	0	0	0	0	0	0
18	561.04	4	16	2	1	10	4	0	2	0	0	0	0	0	0	0	0	0	0	0
19	878.43	5	13	2	0	14	4	2	4	8	7	3	3	0	0	0	0	0	0	0
20	925.55	7	18	3	0	9	3	1	4	8	7	3	3	0	0	0	0	0	0	0
21	701.23	8	24	3	1	0	0	0	0	0	5	5	4	0	0	0	0	0	0	0
22	641.09	6	9	1	1	8	4	0	4	4	6	2	0	0	0	0	0	0	0	0
23	891.51	7	22	2	1	4	2	1	2	6	7	5	3	0	0	0	0	0	0	0
24	703.18	6	13	1	0	9	3	0	2	2	7	5	7	0	0	0	0	0	0	0
25	800.32	10	11	7	0	5	1	1	2	6	8	4	3	0	0	0	0	0	0	0
26	799.39	8	25	3	2	0	0	0	0	0	9	9	0	0	0	0	0	0	0	0
27	1171.98	6	35	1	0	8	4	2	4	12	9	3	0	0	0	0	0	0	0	0
28	951.58	9	22	3	0	4	2	2	2	10	8	5	0	0	0	0	0	0	0	0
29	1161.88	5	18	1	0	15	5	2	4	9	12	7	3	0	0	0	0	0	0	0
30	1021.72	10	22	2	0	6	2	2	2	8	8	8	0	0	0	0	0	0	0	0
31	762.25	6	15	2	1	4	2	1	2	6	8	4	3	0	0	0	0	0	0	0
32	979.64	9	23	4	0	4	2	2	2	10	9	5	0	0	0	0	0	0	0	0
33	1063.75	9	19	3	0	13	4	0	2	2	13	11	2	0	0	0	0	0	0	0
34	1298.15	16	26	8	1	5	1	0	2	2	15	11	6	0	0	0	0	0	0	0
35	1217.97	9	17	4	1	11	1	3	4	16	12	8	0	0	0	0	0	0	0	0
36	1285.09	11	29	4	1	6	2	1	2	4	13	11	9	0	0	0	0	0	0	0
37	1347.23	10	26	6	1	12	4	2	2	12	12	6	3	0	0	0	0	0	0	0
38	1213.95	7	22	1	0	12	4	3	2	16	12	8	0	0	0	0	0	0	0	0
39	1151.86	11	24	3	0	5	1	0	2	2	18	14	4	0	0	0	0	0	0	0
40	1178.93	9	29	2	0	6	2	2	2	8	11	11	3	0	0	0	0	0	0	0
41	1298.11	9	25	2	0	11	3	3	2	14	12	10	0	0	0	0	0	0	0	0
42	1477.42	9	26	2	1	18	6	2	2	10	14	12	3	0	0	0	0	0	0	0
43	1242.01	9	20	5	0	12	2	3	2	18	12	6	0	0	0	0	0	0	0	0
44	1014.73	10	24	5	1	6	2	1	2	4	7	7	3	0	0	0	0	0	0	0
45	1499.44	10	32	3	1	12	4	1	2	8	17	13	5	0	0	0	0	0	0	0
46	1255.07	8	26	2	2	9	3	2	2	10	11	9	3	0	0	0	0	0	0	0
47	1051.68	6	21	1	0	11	3	1	4	6	10	8	6	0	0	0	0	0	0	0
48	1358.18	9	30	5	0	10	2	0	4	4	17	10	8	0	0	0	0	0	0	0
49	1620.57	14	22	5	0	15	5	0	2	6	22	18	10	0	0	0	0	0	0	0
50	1172.95	8	25	3	1	3	1	1	0	6	10	8	6	0	0	0	0	0	0	0
51	1150.86	7	18	3	1	12	3	2	2	12	11	7	3	0	0	0	0	0	0	0
52	1066.78	8	33	2	1	0	0	1	0	4	10	10	6	0	0	0	0	0	0	0
53	1475.32	10	25	2	0	10	2	2	4	12	18	14	8	0	0	0	0	0	0	0
54	1280.98	7	17	1	0	14	4	1	4	8	16	10	11	0	0	0	0	0	0	0
55	3348.25	23	51	15	1	22	4	3	6	22	46	28	21	0	0	0	0	0	0	0
56	4131.47	24	76	13	6	19	3	5	2	36	46	31	18	0	0	0	0	0	0	0
57	4172.54	23	82	7	1	26	8	5	6	26	49	35	28	0	0	0	0	0	0	0
58	4137.64	21	66	5	1	12	3	6	4	28	66	33	44	0	0	0	0	0	0	0
59	4119.39	21	65	7	2	26	5	4	4	22	62	32	25	0	0	0	0	0	0	0
60	4033.33	25	62	12	2	29	8	5	6	30	47	31	24	0	0	0	0	0	0	0

Table III.2- Input Data for the GC Model. AVR (10 Molecules)

Fraction	MW	Structure																		
1	474.89	5	16	3	0	6	2	0	2	0	0	0	0	0	0	0	0	0	0	0
2	687.20	9	21	4	1	0	0	0	0	0	5	5	4	0	0	0	0	0	0	0
3	629.08	8	12	4	0	4	2	1	2	6	4	2	0	0	0	0	0	0	0	0
4	878.43	5	13	2	0	14	4	2	4	8	7	3	3	0	0	0	0	0	0	0
5	800.32	10	11	7	0	5	1	1	2	6	8	4	3	0	0	0	0	0	0	0
6	762.25	6	15	2	1	4	2	1	2	6	8	4	3	0	0	0	0	0	0	0
7	1347.23	10	26	6	1	12	4	2	2	12	12	6	3	0	0	0	0	0	0	0
8	1242.01	9	20	5	0	12	2	3	2	18	12	6	0	0	0	0	0	0	0	0
9	1620.57	14	22	5	0	15	5	0	2	6	22	18	10	0	0	0	0	0	0	0
10	3348.25	23	51	15	1	22	4	3	6	22	46	28	21	0	0	0	0	0	0	0

Fraction 1

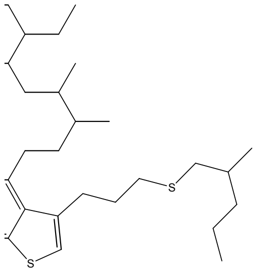
Molecule # 1



$C_{34}H_{66}$
Mol. Wt.: 474.89

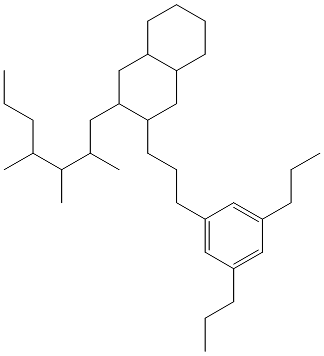
Molecule # 2

$C_{39}H_{64}S_2$
Mol. Wt.: 597.06



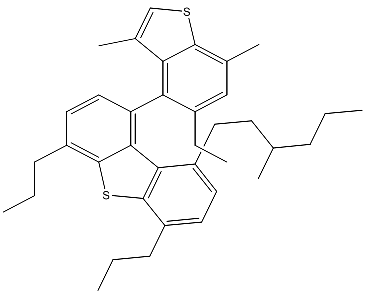
Molecule # 3

$C_{35}H_{60}$
Mol. Wt.: 480.85

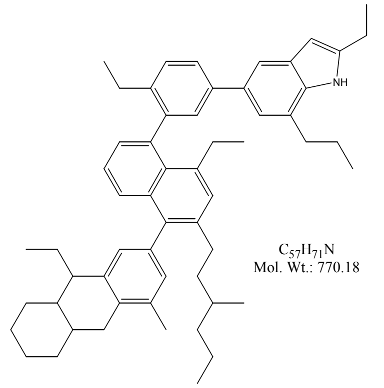


Molecule # 4

$C_{37}H_{46}S_2$
Mol. Wt.: 554.89



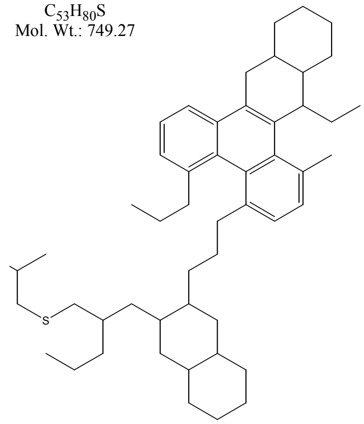
Molecule # 5



$C_{57}H_{71}N$
Mol. Wt.: 770.18

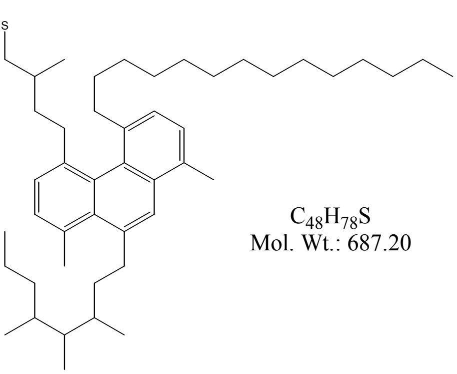
Molecule # 6

$C_{53}H_{80}S$
Mol. Wt.: 749.27



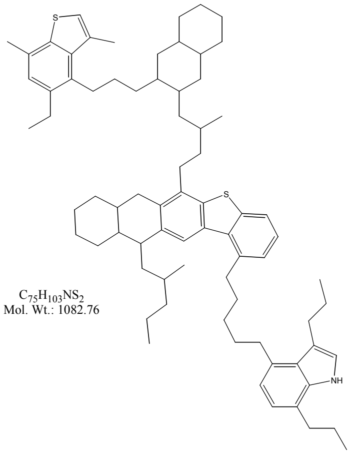
Fraction 2

Molecule # 1



$C_{48}H_{78}S$
Mol. Wt.: 687.20

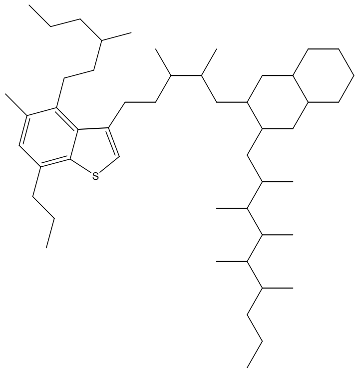
Molecule # 2



$C_{75}H_{103}NS_2$
Mol. Wt.: 1082.76

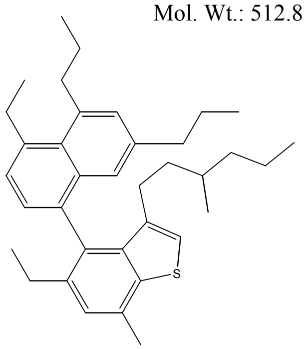
Molecule # 3

$C_{50}H_{86}S$
Mol. Wt.: 719.28



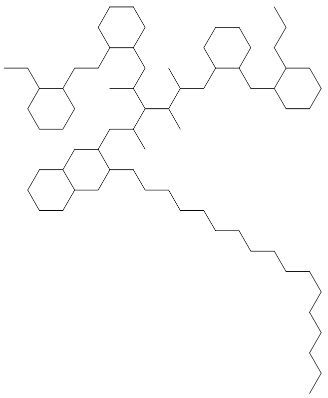
Molecule # 4

$C_{36}H_{48}S$
Mol. Wt.: 512.83

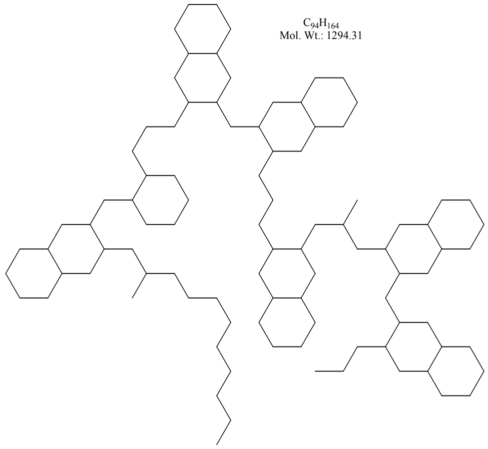


Molecule # 5

$C_{71}H_{132}$
Mol. Wt.: 985.81



Molecule # 6

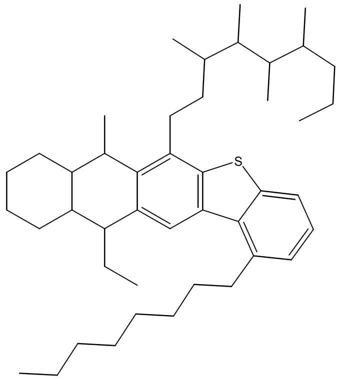


$C_{94}H_{164}$
Mol. Wt.: 1294.31

Fraction 3

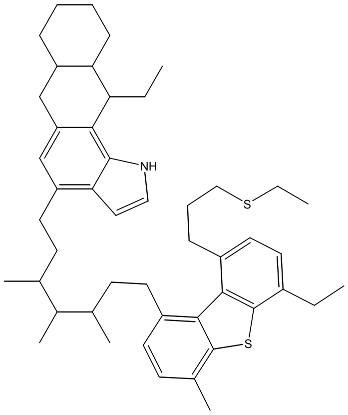
Molecule #1

$C_{44}H_{68}S$
Mol. Wt.: 629.08



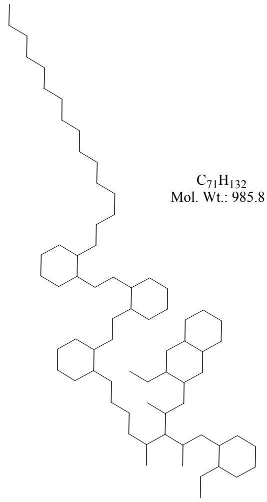
Molecule # 2

$C_{48}H_{65}NS_2$
Mol. Wt.: 720.17

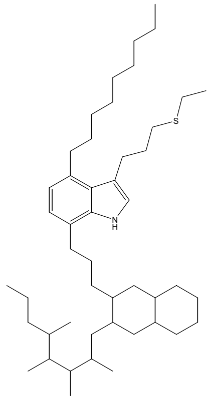


Molecule #3

$C_{71}H_{132}$
Mol. Wt.: 985.81

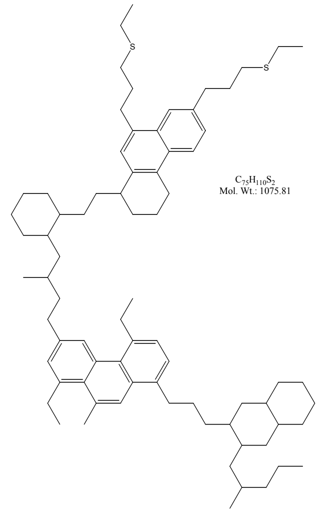


Molecule #4



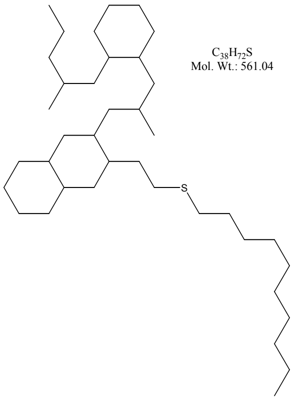
$C_{47}H_{81}NS$
Mol. Wt.: 692.22

Molecule #5



$C_{75}H_{110}S_2$
Mol. Wt.: 1075.81

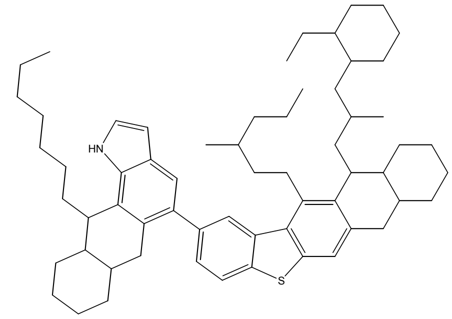
Molecule # 6



$C_{38}H_{72}S$
Mol. Wt.: 561.04

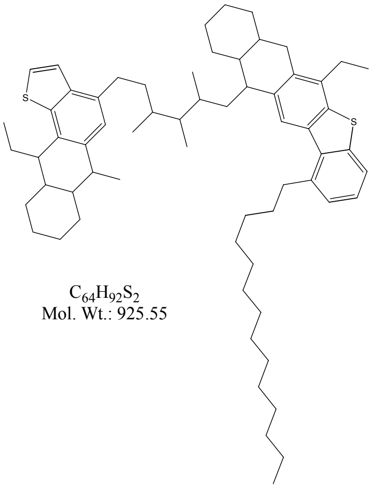
Fraction 4

Molecule #1



$C_{62}H_{87}NS$
Mol. Wt.: 878.43

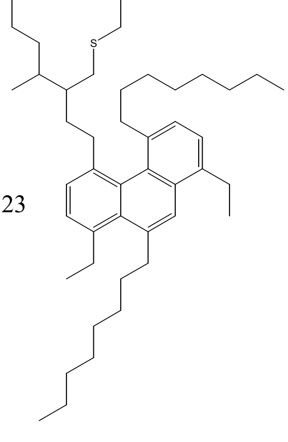
Molecule # 2



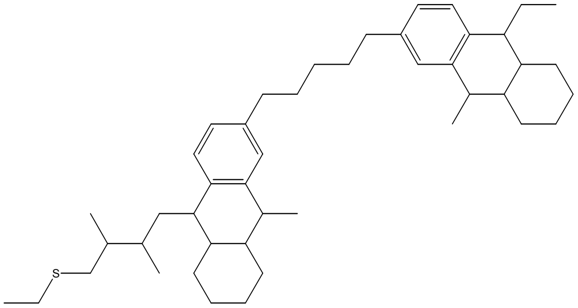
$C_{64}H_{92}S_2$
Mol. Wt.: 925.55

Molecule # 3

$C_{49}H_{80}S$
Mol. Wt.: 701.23

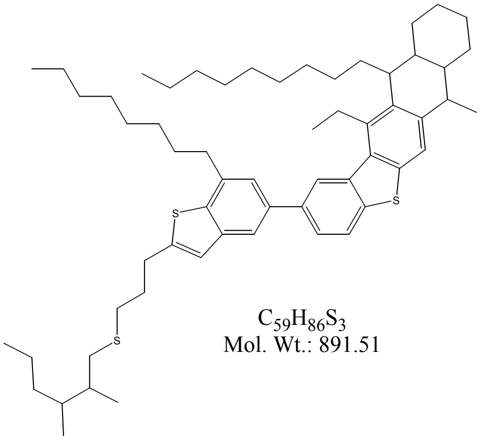


Molecule #4



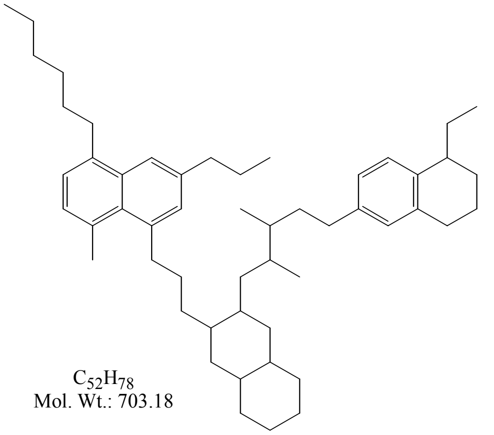
$C_{45}H_{68}S$
Mol. Wt.: 641.09

Molecule #5



$C_{59}H_{86}S_3$
Mol. Wt.: 891.51

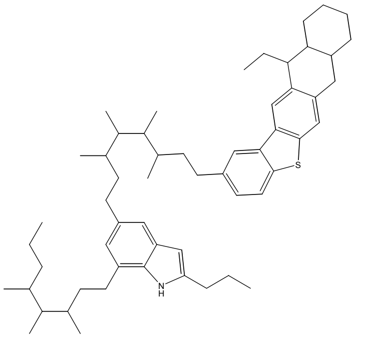
Molecule #6



$C_{52}H_{78}$
Mol. Wt.: 703.18

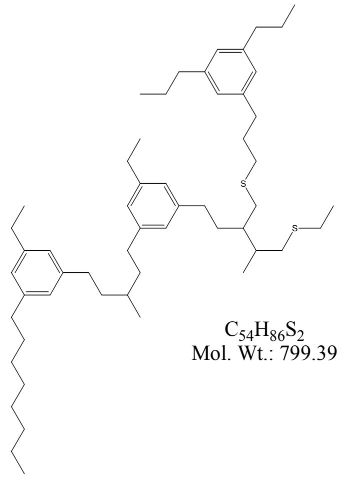
Fraction 5

Molecule # 1



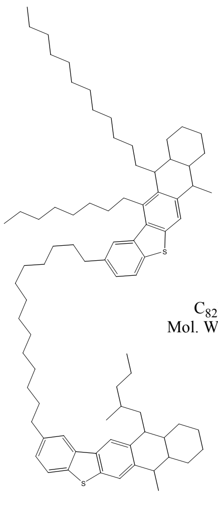
$C_{56}H_{81}NS$
Mol. Wt.: 800.32

Molecule #2



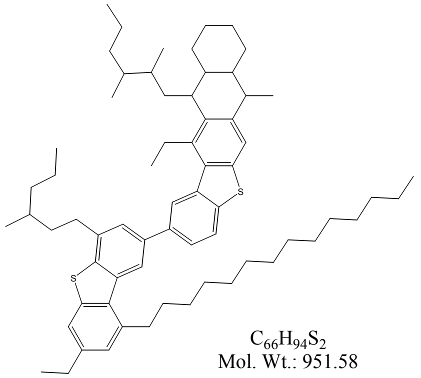
$C_{54}H_{86}S_2$
Mol. Wt.: 799.39

Molecule #3



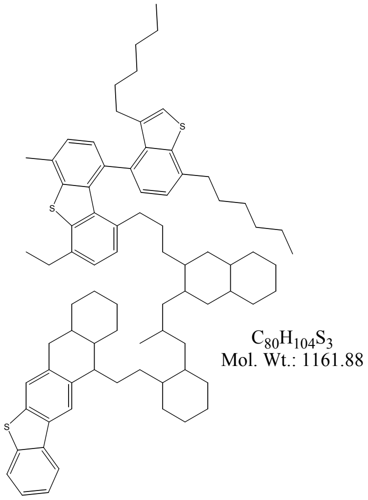
$C_{82}H_{122}S_2$
Mol. Wt.: 1171.98

Molecule # 4



$C_{66}H_{94}S_2$
Mol. Wt.: 951.58

Molecule # 5



$C_{80}H_{104}S_3$
Mol. Wt.: 1161.88

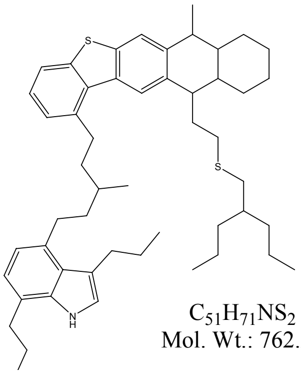
Molecule # 6

raction 6

$C_{71}H_{104}S_2$
Mol. Wt.: 1021.72

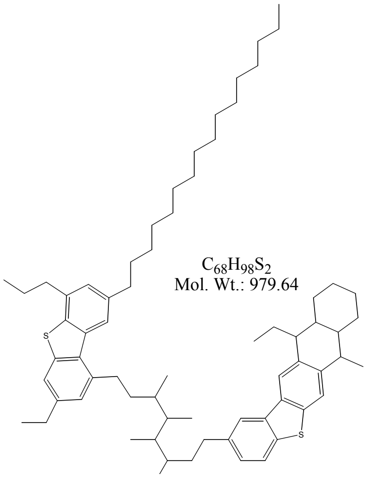
Fraction 6

Molecule # 1



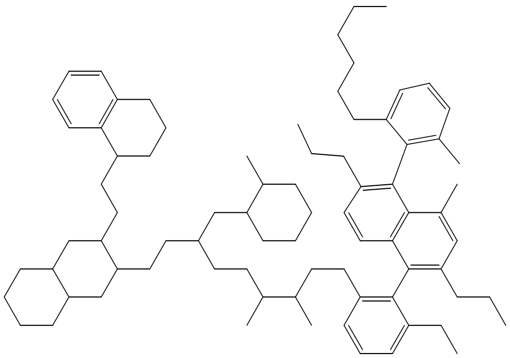
$C_{51}H_{71}NS_2$
Mol. Wt.: 762.25

Molecule # 2



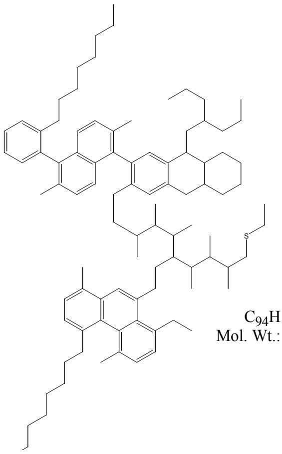
$C_{68}H_{98}S_2$
Mol. Wt.: 979.64

Molecule # 3



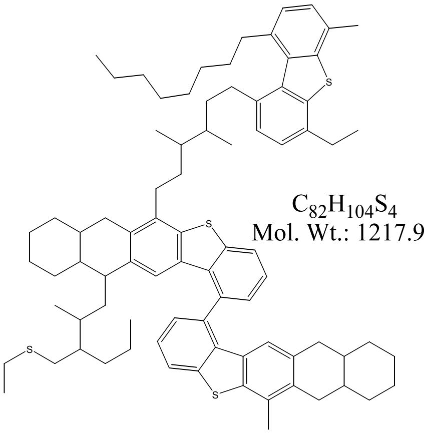
$C_{79}H_{114}$
Mol. Wt.: 1063.75

Molecule # 4



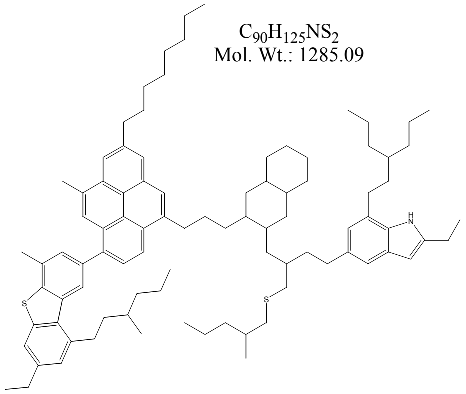
$C_{94}H_{136}S$
Mol. Wt.: 1298.15

Molecule # 5



$C_{82}H_{104}S_4$
Mol. Wt.: 1217.97

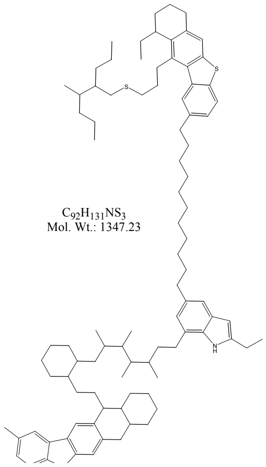
Molecule # 6



$C_{90}H_{125}NS_2$
Mol. Wt.: 1285.09

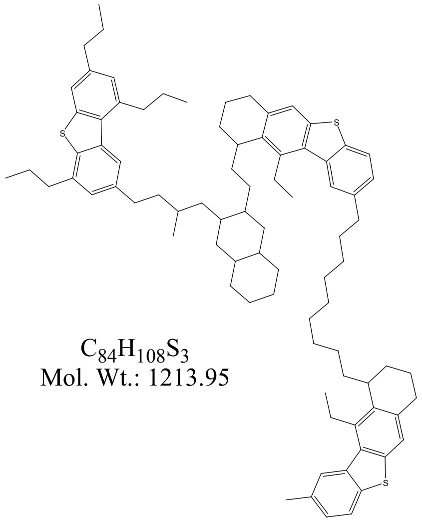
Fraction 7

Molecule #1



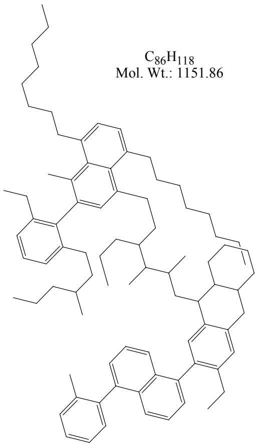
$C_{92}H_{131}NS_3$
Mol. Wt.: 1347.23

Molecule # 2



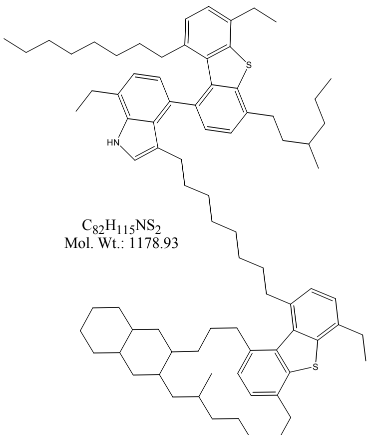
$C_{84}H_{108}S_3$
Mol. Wt.: 1213.95

Molecule #3



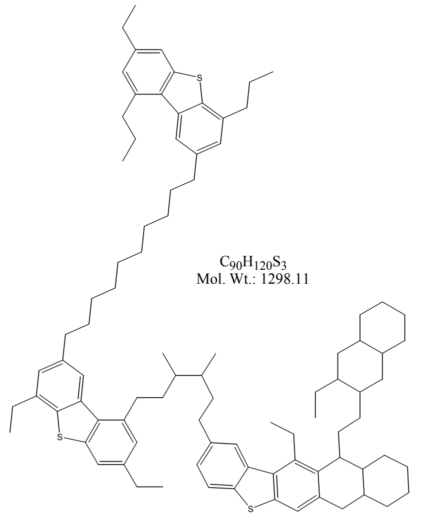
$C_{86}H_{118}$
Mol. Wt.: 1151.86

Molecule # 4



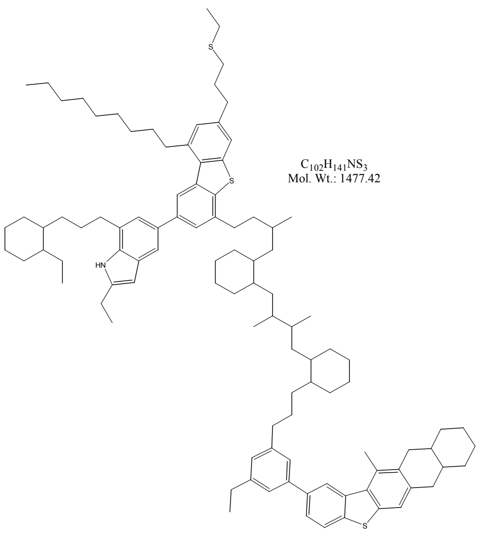
$C_{82}H_{115}NS_2$
Mol. Wt.: 1178.93

Molecule # 5



$C_{90}H_{120}S_3$
Mol. Wt.: 1298.11

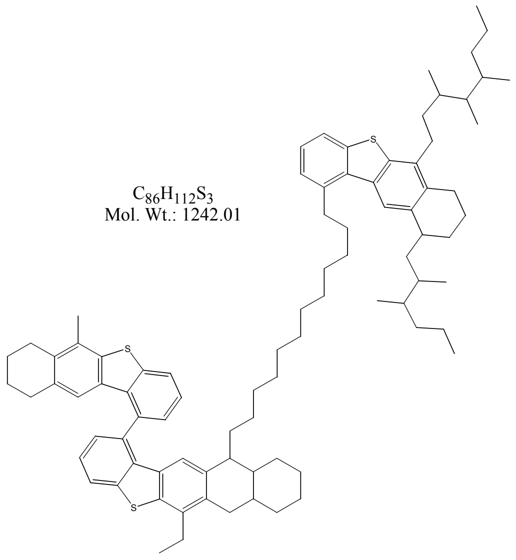
Molecule # 6



$C_{102}H_{141}NS_3$
Mol. Wt.: 1477.42

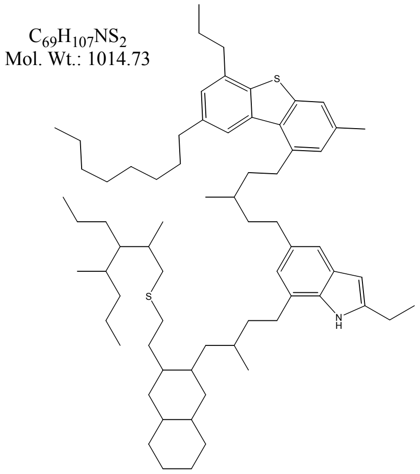
Fraction 8

Molecule #1



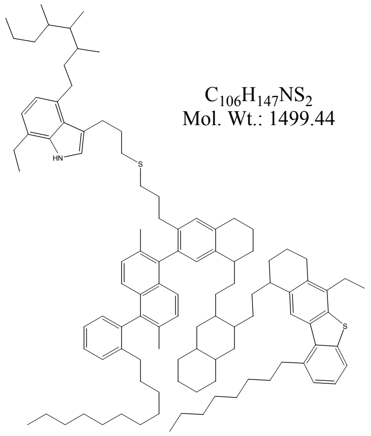
$C_{86}H_{112}S_3$
Mol. Wt.: 1242.01

Molecule #2



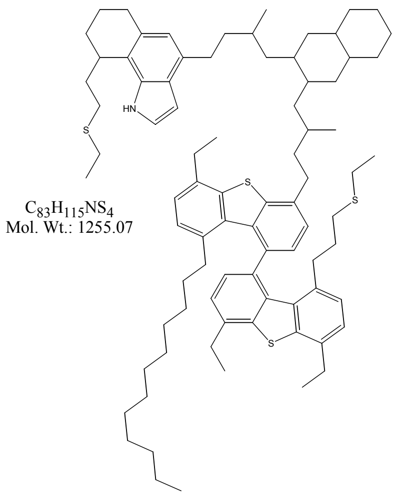
$C_{69}H_{107}NS_2$
Mol. Wt.: 1014.73

Molecule # 3



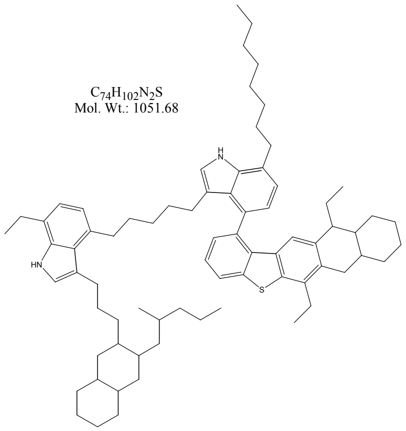
$C_{106}H_{147}NS_2$
Mol. Wt.: 1499.44

Molecule # 4



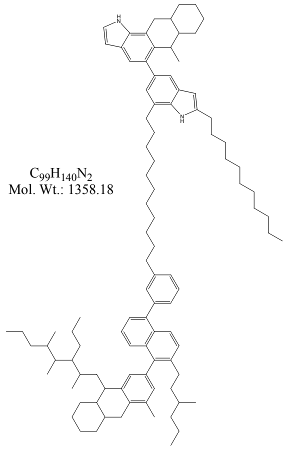
$C_{83}H_{115}NS_4$
Mol. Wt.: 1255.07

Molecule # 5



$C_{74}H_{102}N_2S$
Mol. Wt.: 1051.68

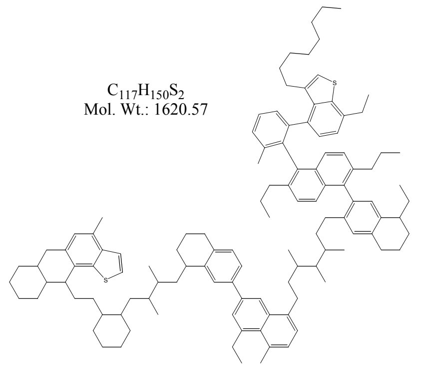
Molecule # 6



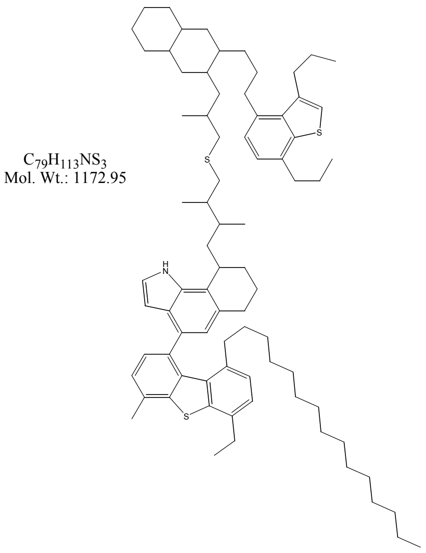
$C_{99}H_{140}N_2$
Mol. Wt.: 1358.18

Fraction 9

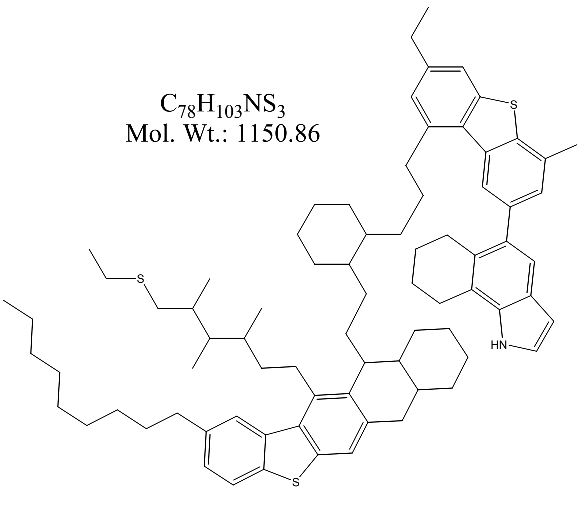
Molecule #1



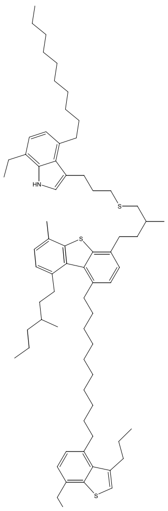
Molecule #2



Molecule #3

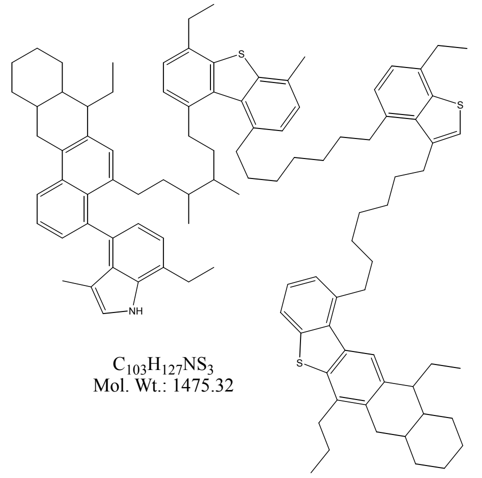


Molecule #4



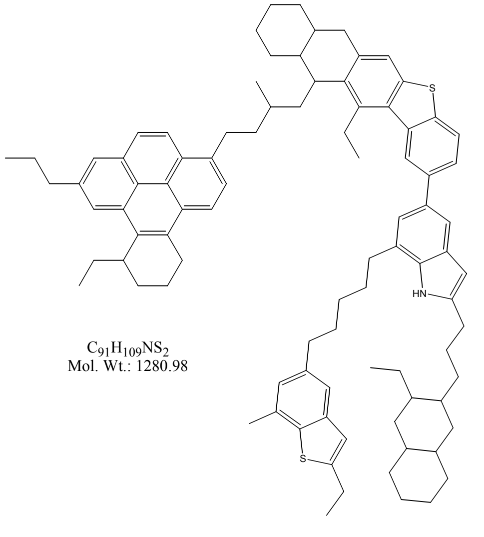
$C_{71}H_{103}NS_3$
Mol. Wt.: 1066.78

Molecule #5



$C_{103}H_{127}NS_3$
Mol. Wt.: 1475.32

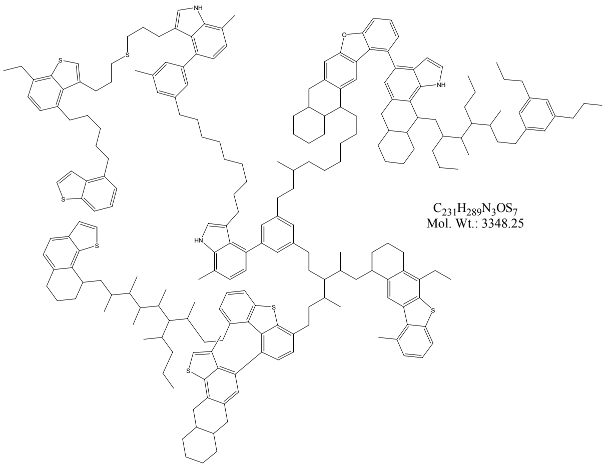
Molecule #6



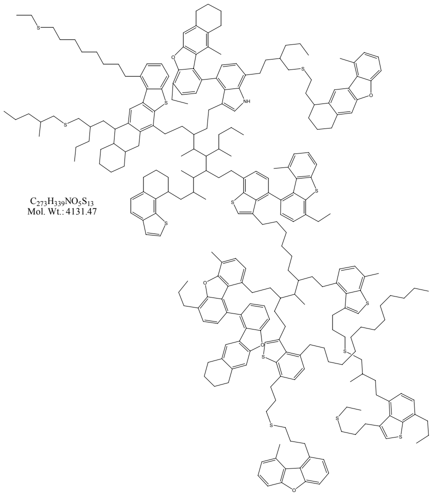
$C_{91}H_{109}NS_2$
Mol. Wt.: 1280.98

Fraction 10

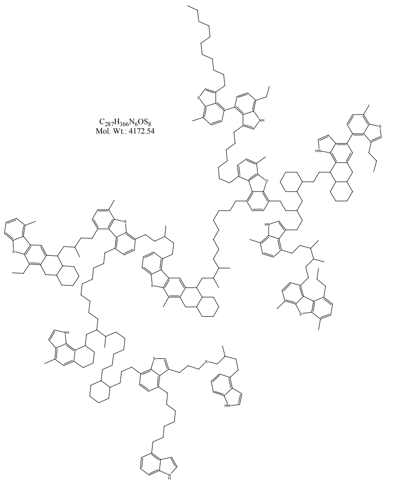
Molecule #1



Molecule # 2

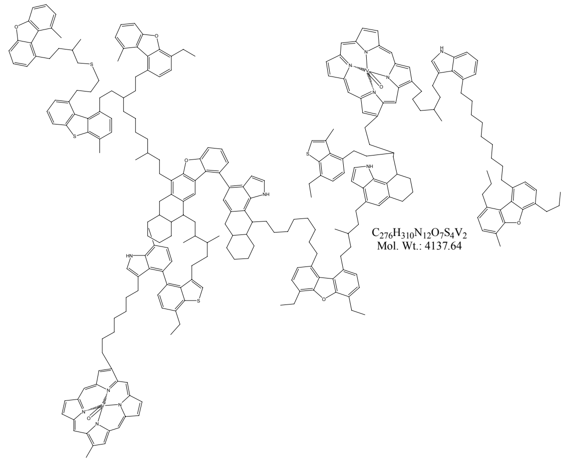


Molecule #3



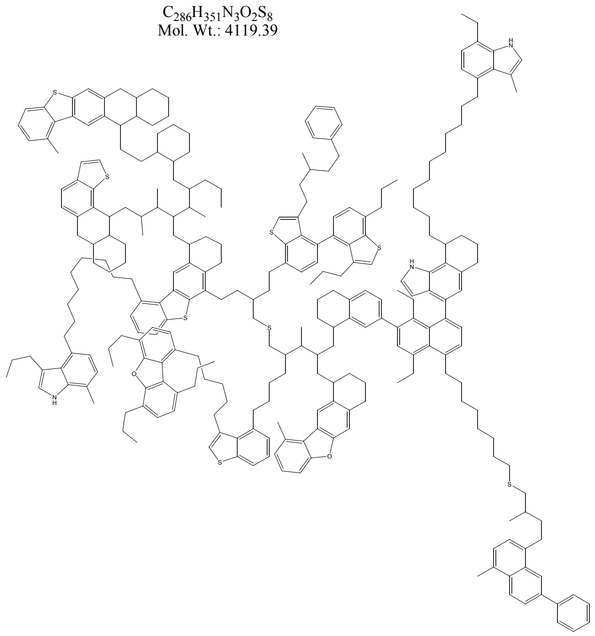
$C_{287}H_{366}N_6OS_8$
Mol. Wt.: 4172.54

Molecule # 4

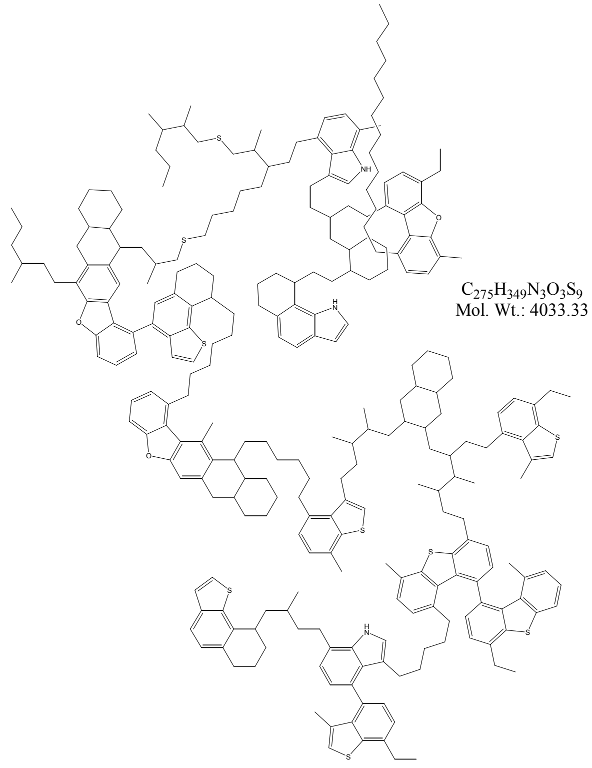


$C_{276}H_{310}N_{12}O_7S_4V_2$
Mol. Wt.: 4137.64

Molecule # 5



Molecule #6



6.

berta

Nan

Titl es Of Athabasca Vacuum Residue Using
a Gr

Deg

Year this Degree Granted: 2006

Apt. 1301 Bracewood Dr. S.W.
Calgary, AB T2W 3M4
Canada

University of Alberta

Estimation of Thermophysical Properties Of Athabasca Vacuum Residue Using a Group Contribution Based Equation of State

by

Annemi Van Waeyenberghe

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science in Chemical Engineering

Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2006

2006

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Estimation of Thermophysical Properties Of Athabasca Vacuum Residue using a Group Contribution Based Equation of State** submitted by **Annemi Van Waeyenberghe** in partial fulfillment of the requirements for the degree of **Master of Science in Chemical Engineering**.

____

Dr. John M. Shaw
(supervisor)

____

Dr. William C. McCaffrey

____

Dr. Luciane B. Cunha

Dated: June ___, 2006

To

Anthony Gabriel and Mario

7. ABSTRACT

Estimation of thermophysical properties is a key aspect of hydrocarbon process simulation. Cubic equations of state have been used widely for this purpose. The fluid parameters employed by these equations are estimated from critical properties which are not available for heavy hydrocarbons. This work deals with the estimation of thermophysical properties using the Peng-Robinson equation of state, where parameters are estimated using a group contribution method. This approach requires only information about the molecular structure of the components. The molecular groups are based on the work of Coniglio et al. (2000). Molecular representations for Athabasca Residue were obtained from Sheramata (2005). A computer program implementing the algorithm has been prepared and extended to mixtures. Agreement with experimental density and bubble pressure data and conventional equation of state calculations is discussed along with results obtained from application of the group contribution approach to Athabasca Vacuum Residue and its mixtures with decane.

8. AKNOWLEDGEMENT

I would like to thank first my supervisor Dr. John M. Shaw. His guidance, support and advice along this journey were an invaluable resource for the completion of the present study. In addition, his energy, happiness and enthusiasm transformed this experience into an important life lesson for me.

In second place I would like to thank the entire Petroleum Thermodynamics research group. Special thanks to Dr. Maham for his attention and support. Thanks to Bei, Xiaouxi, Richard, Vasek, Kan and the rest of the group for their companionship.

Thanks to Richard and Jeff for all their input and discussions.

Also, a special thank you to my friends in Edmonton. They became my extended family during the completion of this program. A special thank you, as well, to my husband Mario, for his companionship, motivation and unconditional support.

Anthony Gabriel, thanks for making the best of my days.

I acknowledge the sponsors of my project: Alberta Energy Research Institute, Albian Sands Energy Inc., Computer Modelling Group Ltd., ConocoPhillips, Imperial Oil Resources, NEXEN Inc., Natural Resources Canada, Petroleum Society of the CIMM, Oilphase-DBR, Oilphase – a Schlumberger Company, Schlumberger, Syncrude Canada Ltd. and NSERC.

Finally, thanks to the Department of Chemical and Materials Engineering and its administrative staff, for their active support.