Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

5-26-2021 10:30 AM

Thermodynamic Vapor-Liquid Equilibrium in Naphtha-Water Mixtures

Sandra Milena Lopez-Zamora, *The University of Western Ontario*

Supervisor: de Lasa, Hugo I., *The University of Western Ontario*
Co-Supervisor: Escobedo, Salvador, *The University of Western Ontario*
Co-Supervisor: Bhattacharya, Sujit, *Syncrude Canada Ltd*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Chemical and Biochemical Engineering
© Sandra Milena Lopez-Zamora 2021

Follow this and additional works at: https://ir.lib.uwo.ca/etd

 Part of the Petroleum Engineering Commons, and the Thermodynamics Commons

Recommended Citation

Lopez-Zamora, Sandra Milena, "Thermodynamic Vapor-Liquid Equilibrium in Naphtha-Water Mixtures" (2021). *Electronic Thesis and Dissertation Repository*. 7822.
https://ir.lib.uwo.ca/etd/7822

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

Abstract

Naphtha is used to dilute the froth from bitumen treatment. Naphtha is recovered using a Naphtha Recovery Unit (NRU) and sent back to the froth dilution step. To minimize the environmental and economic impact of the NRU, it is imperative to maximize the naphtha recovery. It is, in this respect, that enhanced NRU Vapour-Liquid-Liquid equilibrium data is a significant value. The prediction of phase equilibria for hydrocarbon/water blends in separators, is a subject of considerable importance for chemical processes. Despite its relevance, there are still pending questions. Among them, is the prediction of the correct number of phases. While a stability analysis using the Gibbs Free Energy of mixing and the NRTL model for n-octane/water, provide a good understanding of calculation issues when using HYSYS V9 and Aspen Plus V9 software, this shows that significant phase equilibrium uncertainties still exist. In the case of multicomponent mixtures, the Tangent Plane Distance (TPD) is evaluated as a possible criterion for calculating the number of phases. Additionally, Paraffinic Aromatic Synthetic Naphtha (PASN) with a similar True Boiling Point (TBP) as typical naphtha can be used. Runs were developed in a CREC VL Cell operated with n-octane/water and PASN/water mixtures under dynamic conditions and used to establish the two-phase (liquid-vapour) and three-phase (liquid-liquid-vapour) domains. Results obtained demonstrate that the complete solubility is larger than the predicted by simulation software or reported in the technical literature. Furthermore, and to provide an effective and accurate method for predicting the number of phases, a Classification Machine Learning (ML) technique was implemented. Finally, traditional flash split calculations are reported explaining the challenges presented for the solution of the Rachford-Rice equations. A comparison of flash calculations between water/n-octane and PASN/water mixtures using SRKKD EoS is provided. The value of an ML approach developed based on the abundant experimental data available from the CREC-VL experimental Cell experiments is presented.

Keywords

Octane, Naphtha, Thermodynamics, Machine Learning, Phases Number, Phase Split, Flash

Summary for Lay Audience

Canada has produced its Alberta oil sands for about 40 years and has become the world leader in oil sands production. The challenge is to reduce environmental impact and maximize project economics by optimizing each process step involved. One crucial process within bitumen production is the recovering of Naphtha. Naphtha is a chemical blend used as a solvent to facilitate the transportation of bitumen. Naphtha is recovered using a Naphtha Recovery Unit (NRU) and sent back to the dilution step. To minimize the environmental and economic impact of the NRU, it is imperative to maximize the naphtha recovery. To do this, it is essential to understand the interactions between water and hydrocarbon and establish Vapour-Liquid-Liquid equilibrium data is of considerable importance for chemical processes. However, there are still pending questions. Among them, is the prediction of the correct number of phases. While a stability analysis of simple mixtures such as n-octane/water, provide a good understanding of calculation issues, when using HYSYS V9 and Aspen Plus V9 software, this shows that significant phase equilibrium uncertainties still exist.

To clarify these matters, n-octane and water blends, are good surrogates of naphtha/water mixtures. Additionally, Paraffinic Aromatic Synthetic Naphtha (PASN) similar to a typical naphtha, can be used. Runs were developed in a CREC VL Cell operated with n-octane/water and PASN/water mixtures under dynamic conditions and used to establish the two-phase (liquid-vapour) and three-phase (liquid-liquid-vapour) domains. Results obtained demonstrate, that the complete solubility in the liquid phase is larger than the one reported in the technical literature. Furthermore, and to provide an effective and accurate method for predicting the number of phases, a Classification Machine Learning (ML) technique was implemented. Finally, traditional flash split calculations are reported explaining the challenges presented for the solution. A comparison of between water/n-octane and PASN/water mixtures using an ecuation of state is provided. The value of an ML approach developed based on the experimental data available from the CREC-VL experimental Cell experiments is presented.

A mis padres y hermana: José Fernando, Irene y Mariluz

Gracias por estar ahí cuando más los necesito, porque es posible alcanzar cada sueño al contar con ustedes.

To my beloved husband, Dylan:

Thank you for your support during the development of my research and this thesis, and for all the smiles you put in my face everyday

A Jorge Andrey Varón Zamora, fuiste un hermanito mayor para nosotras. Gracias por ser el primero a quien recuerdo ver programando. Fuiste un gran ejemplo y debo mucho de lo que soy a ti

A Camilo José Escobar (Md. Especialista), nuestro amigo incondicional y con gran corazón, quien nos dejó tempranamente después de una ardua lucha contra el COVID 19. Siempre estarás en nuestra memoria.

La educación es el gran motor del desarrollo personal. Es a través de la educación como las hijas de un campesino pueden convertirse en ***una médica y una ingeniera***, el hijo de un minero puede convertirse en el jefe de la mina, o el hijo de trabajadores agrícolas puede llegar a ser presidente de una gran nación

Nelson Mandela

Acknowledgments

I want to express my gratitude to my supervisor Dr. Hugo de Lasa, who gave me all his help, support and encouraged me during the development of this thesis. I would like to acknowledge my co-supervisors Dr. Salvador Escobedo, for his valuable ideas and support, and Dr. Sujit Bhattacharya from Syncrude Canada Ltd. for his valuable comments, advice and insights from the industry during the development of this research. I would also like to thank the Natural Science and Engineering Research Council of Canada (NSERC) and the industrial sponsor, Syncrude Canada Ltd., for the financial support provided to this research under the NSERC-CRD program. Additionally, I would like to acknowledge Florencia de Lasa for her assistance with the editing of this manuscript.

To my friends and colleagues: Jeonghoon Kong, Maureen Cordoba, Bianca Rusinque, Nicolas Torres, Cesar Medina, Jose Muñoz and Jansen Acosta, for all the discussion related or not with this research. To Graciela de Lasa for her wisdom and advice. To Berniece Dellow, Susan Mosley and Mira Ratkaj for being my friends and support during the hard times. To my in-law family, for all those times I was on the computer during family celebrations. Finally, I would like to say thanks to López and Zamora families, who are the most important part of my life, and to whom I keep saying: “... La fijación de metas distingue a unos hombres de otros. Y aquí lo más importante no es alcanzar dichas metas, sino luchar por ellas." (Hector Abad Gómez).

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

Symbols with Latin letter

A	Helmotz Free Energy
A’	Parameter related to PR or SRK EoS
a	Parameter related to PR or SRK EoS
B’	Parameter related to PR or SRK EoS
b	Parameter related to PR or SRK EoS
C’	Parameter related to PR or SRK EoS
d	Distance
F	Function, Feed
f	Fugacity
G	Gibbs Free Energy
H	Enthalpy, Change of variable for RR
J	Jacobian
K	Equilibrium constant
k	Binary interaction parameter
L	Liquid
N	Number of components
n	Number of data points
P	Pressure
R	Universal Gas Constant
S	Entropy, Probability for Logistic Regression
T	Temperature
t	Change of variable for RR
U	Internal Energy
V	Volumen, Vapor
W	Aqueous phase (Water)
x	Molar Fraction of Liquid Phase
Y	Variable change for TPD
y	Molar Fraction of Vapour Phase
z	Overall Molar Fraction (Feed)

Greek letters

α	Parameter for accounting local composition variations (NRTL method)
β	Phase fraction in moles
Δ	Delta
$\mathcal{F}$	Degrees of freedom
ϕ	Fugacity Coefficient
γ	Activity coefficient
g	Interaction energy (NRTL method)
Ω	SRK parameter
π	Number of phases for Gibbs phase rule
τ	Dimensionless interaction parameters (NRTL method)
μ	Chemical potential
ω	Accentric factor

Subindex and Superindex

C	Critical
E	Euclidian, Excess
feed	Refers to feed
i	Identifies component i of the solution
id	Ideal
ig	Ideal gas
igm	Ideal gas mixture
j	identifies component j or the solution or phase j
k	identifies a subgroup
L	Liquid
mix	mixing
obj	objective
p	Identifies phase *p* in multi-phase mixture
R	Residual
r	Reduced
sat	Saturation
sys	System
surr	Surroundings
V	Vapour, Volume
W	Water
0	Initial

I Phase I
II Phase II

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AUC	Area Under Curve
BIP	Binary Interaction Parameters
CREC	Chemical Reactors Engineering Center
EoS	Equation of State
FN	False Negative
FNN	Feedforward Neural Networks
FP	False Positive
FPR	False Positive Rate
KD	Kabadi-Danner
KNN	K-Nearest Neighbors
LL	Liquid-Liquid
LLE	Liquid-Liquid-Equilibrium
ML	Machine Learning
NRTL	Non-Random Two-Liquid Model
NRU	Naphtha Recovery Unit
PC	Personal Computer
PNN	Probabilistic Neural Networks
PR-EoS	Peng-Robinson - Equation of State
RBF	Radial Basis Transfer
ROC	Receiver Operating Characteristic
RR	Rachford-Rice Equation
RVM	Relevance Vector Machines
SRK-EoS	Soave-Redlich-Kong - Equation of State
SRKKD-EoS	Soave-Redlich-Kong with Kabadi-Danner modification - Equation of State
SVC	Support Vector Classification
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPD	Tangent Plane Distance
TPR	Three Phase Region
UNIQUAC	Universal Quasichemical model

USB	Universal Serial Bus
VL	Vapour-Liquid
VL Cell	Vapour-Liquid Cell
VLL	Vapour-Liquid-Liquid
VLLE	Vapour-Liquid-Liquid Equilibrium

1. Introduction

The use of natural energy resources such as bitumen (crude obtained from oil sands) and heavy oils has increased due to the growing demand for energy [1,2]. Canada has developed its Alberta oil sands for about 40 years and has become the world leader in oil sands production [1,3]. The challenge in bitumen and heavy oil processing is to maximize project economics through the optimization of each process step involved [2]. Within the bitumen upgrading process, the addition of a solvent such as naphtha is required to facilitate transportation. The naphtha is recovered in a Naphtha Recovery Unit (NRU) and recirculated to the process. This recovery step is essential from an economic and environmental point of view. Naphtha recovery should be maximized to reduce wastewater treatment costs and operating costs related to naphtha losses.

In this sense, the vapour-liquid-liquid (VLL) Equilibrium data and flash calculations for hydrocarbon/water mixtures are essential to increase the process economy. In this Chapter 1, first, the context of this research is established, including the properties of bitumen, then the objectives and a thesis overview are presented.

1.1. Context

Compared to conventional crudes, the development of oil sands is energy-intensive [2]. The intensity of oil sand processing significantly augments given the decreased quality with increased density, aromaticity, or impurities from unconventional resources [2]. A simplified process flow diagram for the bitumen process is presented in Figure 1-1.

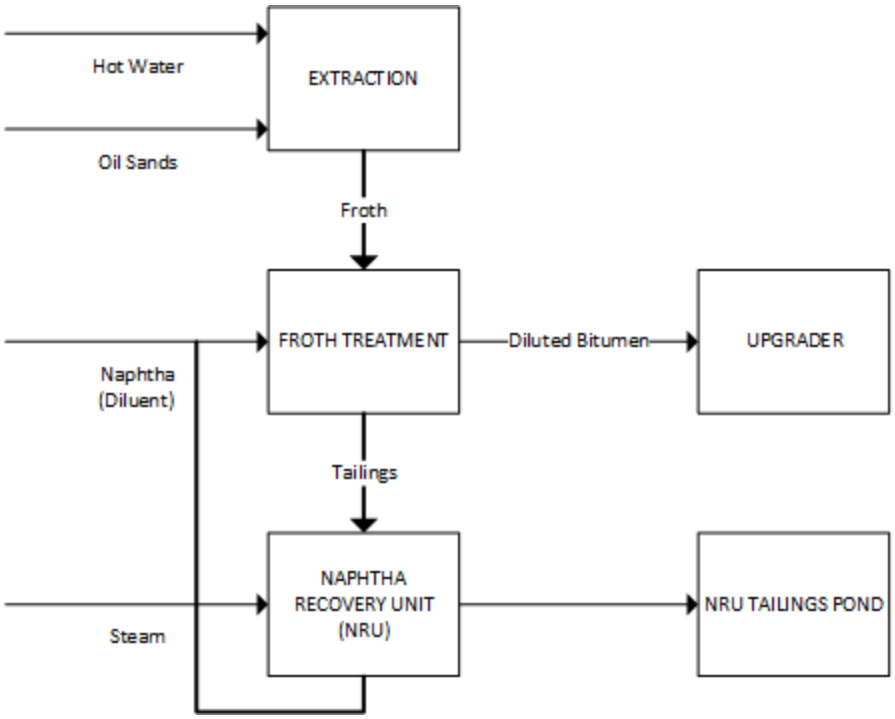


Figure 1-1. Simplified Process Flow Diagram of Bitumen Extraction. Adapted from Oil Sands [4].

The first step in bitumen production is its extraction. In northern Alberta, the bituminous oil sands are surface-mined, and then a hot-water extraction process is applied to recover the bitumen (up to 90% bitumen is recovered) [2]. In southern Alberta, the bitumen is recovered from the surface reservoir by steam injection (SAGD which refers to steam-assisted gravity drainage and with 80% bitumen recovery [2]. The froth that is produced from a hot water extraction process is a highly viscous fluid containing approximately 60wt% bitumen, 30wt% water and 10wt% solids [4]. This froth is diluted with naphtha and is sent to a centrifuge system where the bitumen is separated and sent downstream for upgrading [4]. The tailings, consisting of mostly water, sand and trace amounts of diluent and bitumen, are sent to the Naphtha Recovery Unit (NRU) [4]. The NRU unit is the one of interest for this research. Figure 1-2 presents the process flow of the NRU.

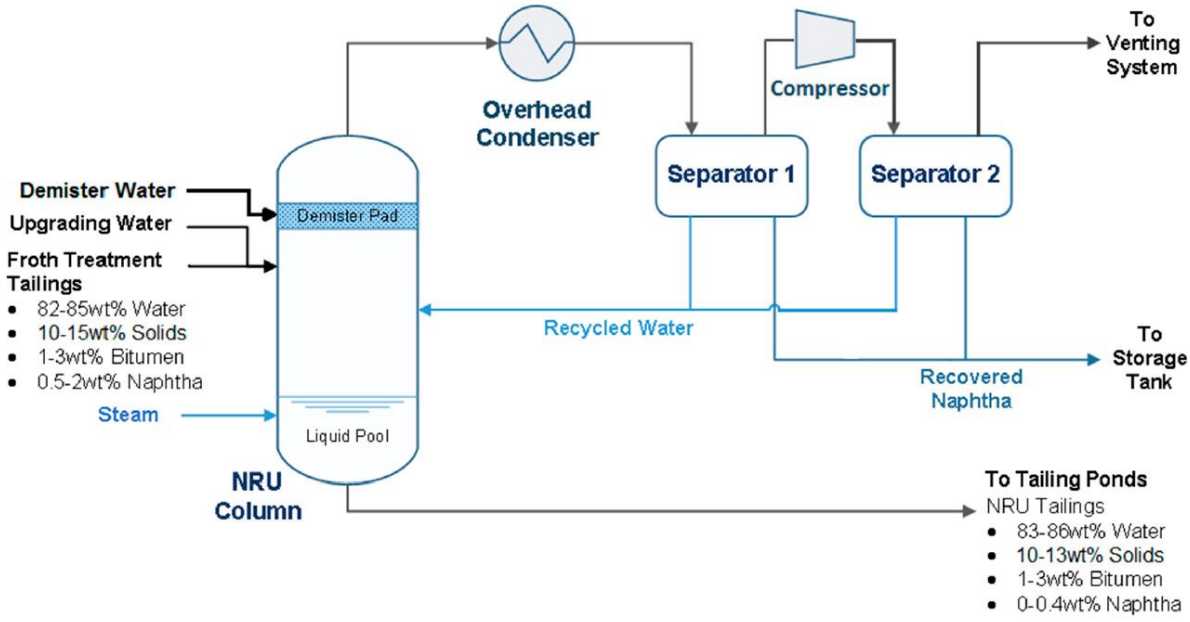


Figure 1-2. Schematic description of the NRU Process Unit. Figure from [4]

The NRU consists of a vacuum stripping tower with steam injected at its bottom. This unit recovers naphtha from the tailings for reuse, and the remnants are discharged to the tailing pond [4]. The primary role of the Naphtha Recovery Unit (NRU) is to recover naphtha from the tailings for reuse, in this process. One should note that the naphtha recovery process is a high energy consumption process where strict environmental guidelines for naphtha recovery must be met [4].

Maximizing the naphtha recovery is imperative, to minimize the environmental impact of the NRU. The analysis of the NRU requires knowledge of the thermodynamic phenomena occurring within the unit. Therefore, the main objective of this thesis is to develop an improved methodology for flash calculations for naphtha/water mixtures.

1.1.1. Properties and Chemistry of Bitumen

Bitumen is the organic fraction from oil sands, obtained following hydrocarbon recovery. Oil sands are constituted by a mixture of quartz/clay particles, water and bitumen [2]. Water, of about 10 microns thickness around the particle perimeter, makes the oil sands

water-wet, and plays an important role in the separation of bitumen from the quartz, by use of a hot-water extraction technique [2].

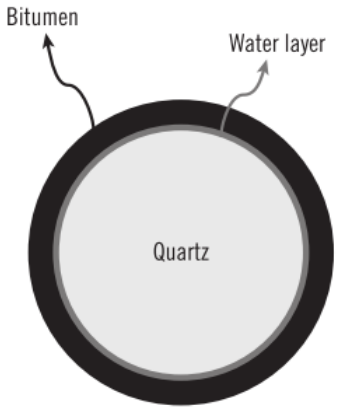


Figure 1-3. Schematic Description of the Oil Sands Particles Showing the Quarts, Bitumen and Water Components. Figure from [2].

The production and refining methods of bitumen are quite diverse. Thus, the properties of the starting bitumen and the used diluents can vary widely as well. As a result, products from this process involve highly variable behaviours [3]. In general, Canadian bitumen has an API gravity of less than 10° and a viscosity above 10,000 cP, at reservoir conditions. It is practically a solid at room temperature [2].

Bitumen content in the oil sands in Alberta varies with the types of oil sand mines. It may vary from 0 to 15 weight percent (wt%), depending on the geographical location. Oil sands are usually divided into three main classes, depending on the concentration of bitumen[2]: (i) Low-grade oil sands, which have a bitumen content of 6–8 wt%, (ii) Medium-grade oil sands, which have a bitumen content of 8–10 wt%, and (iii) Rich oil sands, which have a bitumen content of >10 wt%. Mined oil sands in the northern part of Alberta are considered to be rich oil sands [2].

The total percentage of bitumen plus water in the oil sands remains almost constant at 15 wt%. The remaining 85 wt% is composed of quartz and clay. As the bitumen content increases, the water content decreases, by the same ratio [2].

The typical composition of Alberta bitumen consists of [2]:

- Distillable gas oil product with a boiling point <535°C, which is composed of:
    - Atmospheric Gas Oil (AGO): initial boiling point (IBP) at 350°C
    - Vacuum Gas Oil (VGO): boiling point: 350-535°C
- Undistillable resid with a boiling point >535°C which is composed of :
    - Asphaltenes

Asphaltenes are found in the insoluble bitumen fraction in a normal paraffin solvent, and the soluble fraction in benzene or toluene [2]. Asphaltenes are molecular species with an exact molecular weight that can vary in an ample range from 500 to 15000, with exact quantification being influenced by the analytical technique used [2].

Bitumen is a complex mixture of hydrocarbons containing mainly carbon (82.0 – 83.0 wt%), hydrogen (10.1 – 10.2 wt%), nitrogen (3000 – 5000 pmm), sulfur (4.5 – 6.0 wt%), oxygen (<1 wt%), nickel (80 ppm), and vanadium (220 ppm) [2]. Bitumen is a high-acid crude with an average Total Acid Number (TAN) of 2.5 mg KOH/gram of sample. Concentrations of asphaltene and Conradson Carbon Residue (CCR) in the bitumen are 15.0 wt% and 13.5 wt%, respectively [2]. The atomic ratio of hydrogen to carbon (H/C ratio) in various types of bitumen remains almost constant, within 1.4–1.5, as compared to that in conventional light crude (1.7–1.8) and that in asphaltenes (1.2–1.3) [2].

Bitumen can be classified as well considering its major fractions as: i) saturates, aromatics, resins, and asphaltenes (SARA, Figure 1-4), or ii) paraffins + olefins (<10%), naphthenes (20-30%), and aromatics (60-70%) (PONA) [2].

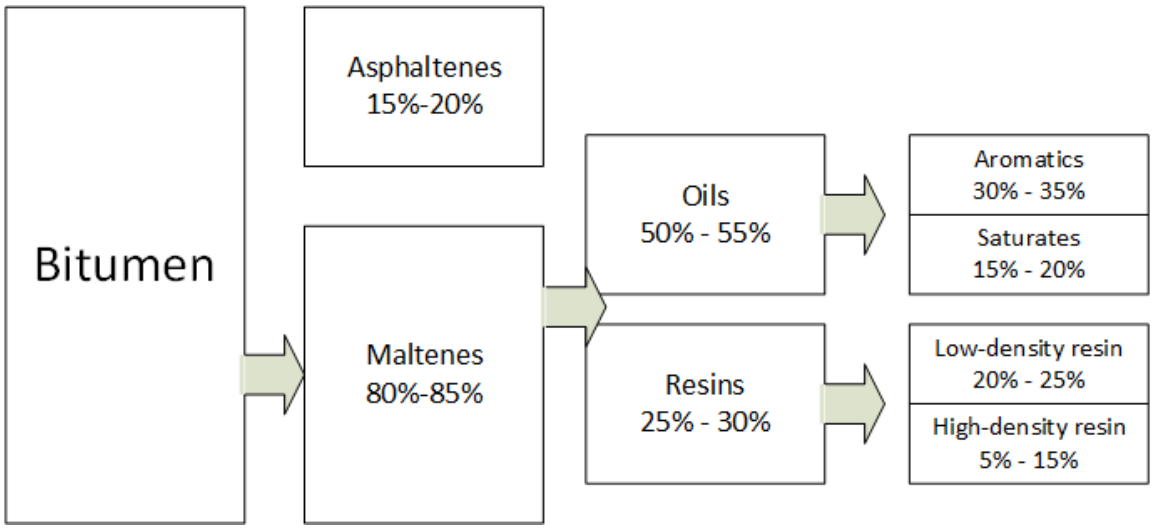


Figure 1-4. SARA Classification of Bitumen (adapted from [2])

Fractionation by distillation is the most important single operating characteristic in a refinery, used to evaluate conventional crude oil feedstocks and their manufactured products [2]. First, the distillate portion (AGO) is distilled under atmospheric pressure (760 mmHg) at about 350°C. The undistilled portion, known as atmospheric residue, is then distilled under a vacuum below 1 mmHg and at 325°C (525-545°C equivalent atmospheric temperature) [2]. In the laboratory, simulated distillation (SimDist) by gas chromatography (GC) is used as a quick method to determine the boiling-point distribution of a bitumen sample. In this process, temperatures can be as high as 640-700°C in a high-temperature SimDist (HTSD) gas chromatograph [2].

In the characterization of bitumen, one of the main challenges is the measurement of true molecular weight. Computer-modelling techniques applicable to lower-molecular weight conventional oil are widely used to predict the properties of heavy oils and bitumen [2]. However, due to the complexity of bitumen molecules, often these models fail to make correct predictions of, for example, molecular weight, viscosity or density [2].

1.1.2. Solvents Employed in Diluted Bitumen

Bitumen is so heavy, thick, and viscous that it is virtually immobile at normal conditions. To transport bitumen to refineries, it must be diluted with condensates or with synthetic crude oils [3]. In order to meet pipeline specifications (e.g. maximum of 940 kg/m3 density and 350 cSt viscosity) which are governed by pumping and pipe considerations [2,3], the viscosity is decreased, and the API gravity is increased. This is achieved by adding a solvent such as a gas condensate, synthetic crude, C4/C5, or naphtha (25-30% vol.) [2,3]. A high amount of alkanes such as C4 or C5 can cause the precipitation of asphaltenes [3]. When naphtha is added, the diluted bitumen is called DilBit. The diluent consists of hydrocarbons in the range of C5-C12. As a result given that bitumen contains hydrocarbons with a carbon number larger than C30, the resuting DilBit involve a C5-C12 and a C30+ fractions without hydrocarbons in the C13-C30 range [2].

Bitumen is usually commercialized as a Synthetic Crude Oil (SCO) or as a Dilbit [3]:

1. Diluted bitumen (Dilbit) is a mixture of bitumen and a solvent. This mixture can be transported by pipeline.
2. Synthetic crude oil (SCO) is a bitumen that has been upgraded by chemical processing and separations.

Diluted bitumens can be divided into four main classes, depending on the diluent [3]:

1. Standard Dilbit where the diluent is a gas condensate,
2. Synbit where the diluent is a synthetic crude,
3. Lightened Dilbit where the diluent is a gas condensate with added C4 and/or C5 diluents, and is also called C4/C5 Enhanced Dilbit,
4. Dilbit which is diluted with a synthetic naphtha.

Synbits are sometimes modified by the addition of a gas condensate, to meet pipeline specifications. These are alternatively called Dilsynbits [3].

1.2. Objectives

This research is aimed to develop an improved algorithm for the flash calculations of naphtha/water mixtures. While understanding that the actual blends in the NRU unit is naphtha, water and bitumen, the focus, however is on naphtha/water blends as a first step for better thermodynamic understanding of the NRU process. Further studies should in our view also include the bitumen. As a result the following two specific objectives are considered in the present Ph.D. thesis:

1. To develop a thermodynamic model for naphtha/water mixtures in the NRU process.
2. To propose an improved methodology for flash calculations of naphtha/water mixtures, accounting for the proposed thermodynamic model.

1.3. Thesis Overview

The present Ph.D. dissertation includes the research developed, as reported in articles published and articles submitted for publication.

This thesis is organized as follows:

a) Chapter 2 provides the reader with background information and a review of the literature on phase equilibrium simulations including, number of phases and phase compositions,
b) Chapter 3 presents the mathematical model and experimental setup employed to obtain the data used for the development of this thesis,
c) Chapter 4 discusses inconsistencies found when using simulation software to simulate n-Octane/Water mixtures. A phase stability analysis is developed and a machine learning classification approach to predict the number of phases, is proposed. Most of the information of Chapter 4 was published in the journal Processes [5],

d) Chapter 5, establishes a Machine Learning approach to predict the number of phases for Paraffinic Aromatic Synthetic Naphtha (PASN). A manuscript related to the results of this chapter was submitted to The Canadian Journal of Chemical Engineering [6],
e) Chapter 6, describes flash calculations for n-Octane/water and PASN/water mixture and involves a Machine Learning approach for the prediction of the mixture pressure,
f) Chapter 7, draws the main conclusions and discusses potential extensions to this research,
g) Finally, Chapter 8 summarizes the accomplishments of this research.

2. Background and Literature Review

This chapter provides background information related to the following topics: i) Thermodynamic principles, including phase equilibrium and flash calculations, ii) Machine learning fundamentals and, iii) State of the art regarding hydrocarbon/water mixture thermodynamic studies.

2.1. Thermodynamic Principles

To establish a model able to represent naphtha/water blends, it is essential to consider classical thermodynamics. In this sense, three conditions are key for VLL thermodynamic equilibria [7]: i) equality of chemical potentials, ii) conservation of mass, and iii) maximization of entropy.

In this section, a summary of the fundamental thermodynamic concepts involved in thermodynamic equilibrium and flash calculations is presented. First, the Gibb's phase rule is stated, then, thermodynamic equilibrium is considered. Finally, the isothermal phase equilibrium calculations are discussed.

2.1.1. Gibbs Phase Rule

To specify the intensive properties in a system of N components distributed among π phases, the temperature (T), the pressure (P) and the molar fractions in each one of the phases must be determined. Regarding these required data, there are $(N-1)$ unknown mole fractions in each phase. Thus, the total number of unknowns is: $2+\pi(N-1)$.

The equations used to find these unknowns are given by the equilibrium conditions for each i component as: $\mu_i^{phase\ I} = \mu_i^{phase\ II} = \cdots = \mu_i^{phase\ \pi}$ $(i = 1,2, \ldots N)$. This gives $N(\pi-1)$ available equations in total.

In this respect, the degree of freedom is then given by the Gibbs's phase rule (Equation (2-1)). This equation helps to determine the number of variables needed to fully define the thermodynamic state of the system [8]:

$$\mathcal{F} = 2 + \pi(N-1) - N(\pi - 1) = 2 + N - \pi \tag{2-1}$$

2.1.2. Thermodynamic Equilibrium

A system is considered to be in equilibrium if the driving forces for all possible internal processes are zero [9]. At the microscopical level, when a mixture splits into two phases, molecules of both components are transferred continuously from one phase to the other. However, at equilibrium, the net component transfer between phases is zero. This means that the transfer rate in one direction equals the rate of transfer in the reverse direction. Thus, the average (macroscopic) composition of each phase at equilibrium remains constant [8]. This thermodynamic equilibrium between phases, can be expressed by using the chemical potential (μ) for each component, with this chemical potential being the same in both phases:

$$\mu_i^j = \mu_i^p \tag{2-2}$$

One should note that the chemical potential represents the component molar contribution to the Gibbs energy of the mixture (G) such as [8]:

$$G = \sum_i x_i \mu_i \tag{2-3}$$

$$\mu_i|_{pure} = G_i = H_i - TS_i \tag{2-4}$$

Where G_i refers to the chemical potential of the pure component and μ_i refers to the chemical potential of a component in a mixture.

The chemical potential is a partial molar property, which complies with the Gibbs-Duhem Equation [8]:

$$\sum_i x_i d\mu_i = 0, (constant\ P, T) \tag{2-5}$$

For an ideal gas mixture (igm), the Gibbs free energy of the mixture and the chemical potential for component i in the mixture are given by:

$$G^{igm} = \sum_i x_i G_i^{ig} + RT \sum_i x_i \ln x_i \tag{2-6}$$

$$\mu_i^{igm} = G_i^{ig} + RT \ln x_i \tag{2-7}$$

where G_i^{ig} is the Gibbs free energy (chemical potential) of the pure component in the ideal-gas state. In a real mixture, the Gibbs free energy should be written in terms of the residual Gibbs free energy as follows:

$$G = G^{igm} + G^R \tag{2-8}$$

Then,

$$\mu_i = \mu_i^{igm} + \bar{G}_i^R \tag{2-9}$$

$$\bar{G}_i^R = \left(\frac{\partial n G^R}{\partial n_i}\right)_{T,P,n_j} \tag{2-10}$$

The partial molar residual Gibbs free energy ($\bar{G}_i^R$) is important for the definition of the fugacity coefficient. The fugacity of a component in a mixture is related to its ability to "escape" and is defined as:

$$f_i = x_i \phi_i P \tag{2-11}$$

where x_i is the component mole fraction in the multicomponent mixture, P is the pressure and ϕ_i is the component fugacity coefficient of the component given by:

$$\ln \phi_i = \frac{\bar{G}_i^R}{RT} \tag{2-12}$$

The fugacity of a component in a mixture is related to the chemical potential. For the two possible states (A and B) of a mixture at the same temperature, the relationship is given by Equation (2-13), with fugacity being a mathematical transformation of the chemical potential [8].

$$\mu_i^j - \mu_i^p = RT \ln \frac{f_i^j}{f_i^p} \quad (constant\ T) \tag{2-13}$$

Furthermore, the necessary criteria for phase equilibrium (Equation(2-2)) can also be expressed in terms of fugacity (Equation (2-14)):

$$f_i^j = f_i^p \tag{2-14}$$

However, chemical equilibrium is defined not only by the equality of component fugacities across phases [7], but also by the conservation of mass and maximization of entropy. For the maximization of entropy, the change of entropy between the system and its surroundings can be expressed in Table 2-1 [7]:

Table 2-1. Change of Entropy for Different Types of Systems

Irreversible process	$dS = dS_{sys} + dS_{surr} \geq 0$	(2-15)
Reversible process	$dS = dS_{sys} + dS_{surr} = 0$	(2-16)
Isolated system	$dS_{surr} = 0$, then $dS = dS_{sys} \geq 0$	(2-17)

If the system is at equilibrium, the entropy is at its maximum, $dS \geq 0$ and Equation (2-18) applies:

$$TdS - PdV - dU \geq 0 \tag{2-18}$$

The mathematical condition of stable equilibrium depends on the moles, temperature and volume (isothermal-isochoric) set, with the equilibrium state corresponds to the minimum Helmotz free energy (Equation (2-19)). If the moles, the temperature and the pressure (isothermal-isobaric) are fixed, the minimum Gibbs free energy should be at a minimum level [8]. Mathematical conditions for stable equilibrium are summarized in Table 2-2.

$$A \equiv U - TS \tag{2-19}$$

Table 2-2. Mathematical Conditions of Stable Equilibrium [8]

$(dG \leq 0)_{T,P,n}$	(2-20)
$(dA \leq 0)_{T,V,n}$	(2-21)
$(dH \leq 0)_{S,P,n}$	(2-22)
$(dU \leq 0)_{S,V,n}$	(2-23)

2.1.3. Isothermal phase equilibrium calculations

The phase equilibrium calculation problem deals with two main issues [10]: i) ***phase stability***: number of phases, and ii) ***phase-split***: composition and amount of each phase present. The challenge in phase equilibrium calculation is that the number of phases at thermodynamic equilibrium is unknown a priori [10]. To solve this problem, two approaches are typically followed [11]:

i. The number of phases is pre-assigned, and the phase-split calculation are performed, if the negative flash condition (unphysical solution) is obtained, the number of phases is decreased and the phase-split calculation is repeated
ii. Stability testing is performed in a step previous to the phase-split calculation, to determine if it is necessary to increase the number of phases.

For water/hydrocarbon mixtures these conventional approaches are computationally expensive, and there is a lack of reasonable initial estimates of equilibrium ratios [11,12].

2.1.3.1 Phase Stability Analysis

Regarding number of liquid phases in a multicomponent system, one can consider that in a closed system, as is the case in the CREC VL Cell of the present study, species molecules will tend to configure in a state that minimizes their Gibbs free energy [13].

For instance, in the case of a binary mixture, the species blend remain as a stable single-liquid phase, if the tangent line to the ΔG_{mix} curve at of the blend feeding condition point does not contact ΔG_{mix} curve anywhere else [13]. If it does contact the ΔG_{mix} curve, two or more phases will be present, as shown in Figure 2-1.

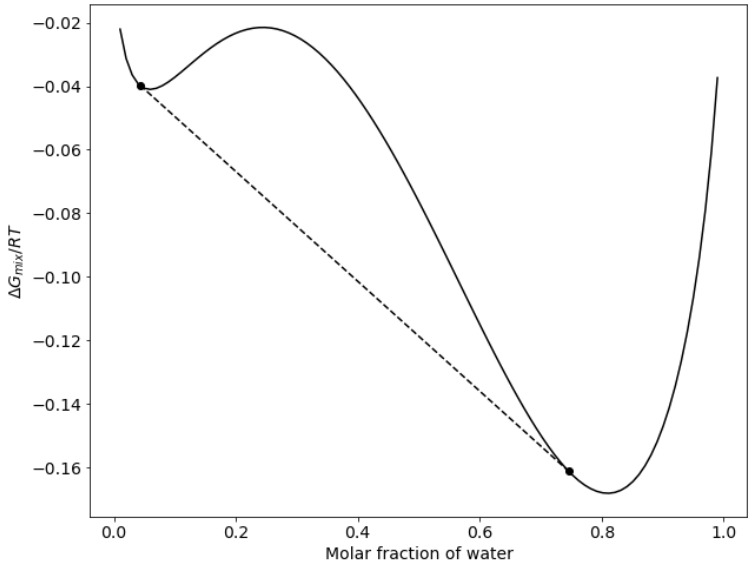


Figure 2-1. $\Delta G_{mix}/RT$ of Methanol/Carbon disulfide at 10°C. Adapted from [8] showing the tangent line contact ΔG_{mix}/RT in two points, thus predicting two phases for all feeding conditions with a water molar fraction between 0,043 and 0.745.

One should consider that for multicomponent mixtures, the ΔG_{mix} will be a surface, and the tangent line will correspond to a tangent plane. Thus, a mixture will display a single liquid phase if the tangent plane to the ΔG_{mix} surface at the point of molar composition feeding does not contact the ΔG_{mix} surface anywhere [13]. The presence of a plurality of contact points indicates one or more phases. If the distance from the plane to the ΔG mixing surface (TPD, Equation (2-24)) is evaluated, when the result is non-negative, for all the trial functions, the feed mixture form only one phase. If a trial phase gives a negative TPD value, then the mixture will split into two or more phases. Further explanation is provided in Chapter 0 and Chapter 5.

$$TPD(y_1, y_2, \dots, y_N) = \sum_{i=1}^{N} y_i \left(\mu_i^{y-phase} - \mu_i^{feed}\right) \tag{2-24}$$

2.1.3.2 Flash Calculations

Traditionally, Rachford-Rice equations [14] are used for the phase splitting analysis for a flash unit. Once this preliminary analysis is completed, the equations for two phases and three phases calculations are established.

Two phases flash equations

Figure 2-2 reports the schematic diagram for a two phases flash unit. Equations (2-25) to (2-27) represent the mole balances (no reaction is present, then the number of moles is constant). Equation (2-28) is obtained from equilibrium. Combing mass balances and equilibrium conditions, in equations (2-29) to (2-32) it is possible to arrive to Rachford-Rice equation (2-33).

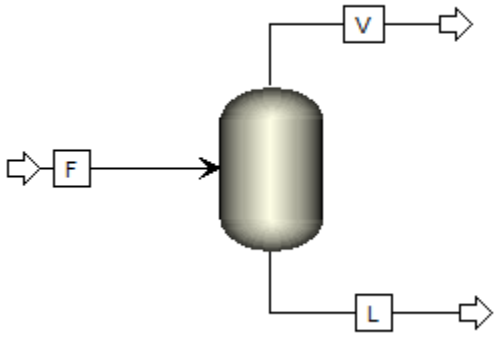


Figure 2-2. Schematic of a flash system dealing with two phases

$$F = L + V \tag{2-25}$$

$$Fz_i = Lx_i^L + Vy_i \tag{2-26}$$

$$\sum_i^N x_i^L = \sum_i^N y_i = 1 \tag{2-27}$$

From equilibrium:

$$K_i^V = \frac{y_i}{x_i^L} = \frac{\phi_i^L}{\phi_i^V} \tag{2-28}$$

$$1 = \frac{L}{F} + \frac{V}{F} \tag{2-29}$$

$$1 = \beta^L + \beta^V \tag{2-30}$$

$$x_i^L = \frac{z_i}{1 + \beta^V (K_i^V - 1)} \tag{2-31}$$

$$y_i = x_i^L K_i^V = \frac{z_i K_i^V}{1 + \beta^V (K_i^V - 1)} \tag{2-32}$$

The Rachford-Rice equation is given below:

$$RR_y = \sum_{i=1}^{N} (y_i - x_i^L) = \frac{z_i (K_i^V - 1)}{1 + \beta^V (K_i^V - 1)} = 0 \tag{2-33}$$

Three phases flash equations

Similarly, Figure 2-3 presents the schematic diagram for a three phases flash unit. Equations (2-34) to (2-36) represent the mole balances (no reaction is present, then the number of moles is constant). Equations (2-37) and (2-38) are obtained from equilibrium. Finally, by combining mass balances and equilibrium conditions, in equations (2-39) to (2-43) it is possible to arrive to Rachford-Rice equations (2-44) and (2-45).

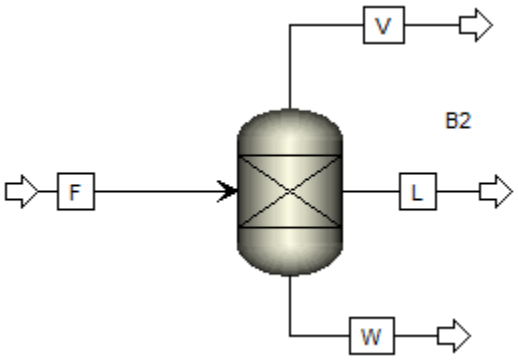


Figure 2-3. Schematic of a flash system dealing with three phases

$$F = L + W + V \tag{2-34}$$

$$F z_i = L x_i^L + W x_i^W + V y_i \tag{2-35}$$

$$\sum_i^N x_i^L = \sum_i^N x_i^W = \sum_i^N y_i = 1 \tag{2-36}$$

From equilibrium:

$$K_i^V = \frac{y_i}{x_i^L} = \frac{\phi_i^L}{\phi_i^V} \tag{2-37}$$

$$K_i^W = \frac{x_i^W}{x_i^L} = \frac{\phi_i^L}{\phi_i^W} \tag{2-38}$$

$$1 = \frac{L}{F} + \frac{W}{F} + \frac{V}{F} \tag{2-39}$$

$$1 = \beta^L + \beta^W + \beta^V \tag{2-40}$$

$$x_i^L = \frac{z_i}{1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1)} \tag{2-41}$$

$$x_i^W = x_i^L K_i^W = \frac{z_i K_i^W}{1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1)} \tag{2-42}$$

$$y_i = x_i^L K_i^V = \frac{z_i K_i^V}{1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1)} \tag{2-43}$$

The Rachford-Rice equations can be then, summarized as follows:

$$RR_y = \sum_{i=1}^{N}(y_i - x_i^L) = \frac{z_i(K_i^V - 1)}{1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1)} = 0 \tag{2-44}$$

$$RR_{x_{i,B}} = \sum_{i=1}^{N}(x_i^W - x_i^L) = \frac{z_i(K_i^W - 1)}{1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1)} = 0 \tag{2-45}$$

Furthermore, if after completion of the stability analysis, the presence of two or more phases is determined, the Rachford-Rice equations stated below have to be used [15]. Equations for these cases are summarized in Table 2-3

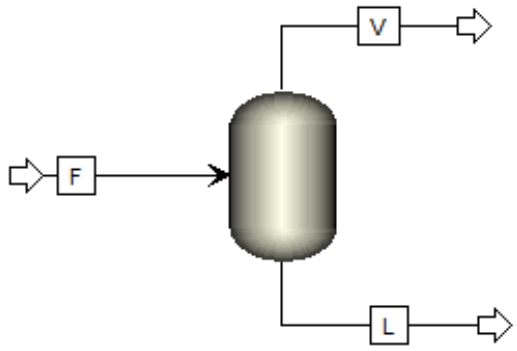
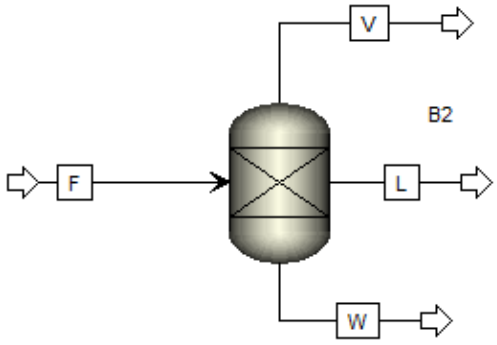
$$\sum_{i=1}^{N} \frac{z_i(K_i^m - 1)}{H_i} = 0, \quad m = 1,2 \dots, J-1 \tag{2-46}$$

$$H_i = 1 + \sum_{m=1}^{J-1} \beta^m (K_i^m - 1) \tag{2-47}$$

$$y_i^m = \frac{z_i K_i^m}{H_i} \quad i = 1,2, \dots, N \quad m = 1,2, \dots, J-1 \tag{2-48}$$

$$y_i^J = \frac{z_i}{H_i} \quad i = 1,2, \dots, N \tag{2-49}$$

Table 2-3. Rachford-Rice equation for two and three phases flash calculation

Two Phases Flash	**Three Phases Flash**
 Figure 2-4. Schematic of a flash system dealing with two phases	 Figure 2-5. Schematic of a flash system dealing with three phases
$F(\beta) = \sum_{i=1}^{N} \frac{z_i(K_i - 1)}{1 + \beta(K_i - 1)} = 0$ (2-50)	$\sum_{i=1}^{N} \frac{z_i(K_i^m - 1)}{H_i} = 0$ (2-51) $H_i = 1 + \beta^L(K_i^L - 1) + \beta^W(K_i^W - 1)$ (2-52)

Where β corresponds to the phase mole fraction.

2.2. Fundamentals of Machine Learning

Artificial Intelligence (AI) has been used in chemical engineering for more than 35 years [16]. AI is defined as “the study of how to make computers do things at which, at the moment, people are better” [17].

Among AI techniques, Machine Learning (ML) has helped to solve problems that require pattern recognition, reasoning and decision making under complex conditions [16]. ML is about modelling data [18] and combines statistics, optimization and computer science [19]. ML gives computers the ability to learn without being explicitly programmed [20]. ML algorithms can be classified into different categories [20]:

- *Supervised Learning Methods* with training data (labeled data) being considered to represent desired solutions. These methods are designated as classification and regression problems. Among the supervised ML algorithms, the following have been reported k-nearest neighbour (KNN), naïve Bayes classifier, support vector machines (SVMs), neural networks, decision trees, random forests, linear regression, and logistic regression [20].
- *Unsupervised Learning* with training “unlabeled” datasets being employed, such as in clustering and dimensionality reduction. Among the unsupervised ML algorithms, the following has been reported: k-means, fuzzy c-means, hierarchical cluster analysis, and self-organizing map, principal component analysis, locally linear embedding, and t-distributed stochastic neighbour embedding [20].
- *Semi-supervised Learning* with both training “unlabeled” datasets and some “labeled” data set being used, with the anomaly detection being an good example. One should note that most semi-supervised learning algorithms are combinations of unsupervised and supervised algorithms. This is the case of the deep belief networks (DBNs) [21] .
- *Reinforcement Learning,* the learning system algorithm (agent) accounts for system environment, and has the ability to select,perform actions, and get rewards in return [21].

This section introduces ML fundamental concepts that will help understand how to better use ML to improve flash calculations. The focus of this section is on regression and classification techniques.

2.2.1. Machine Learning tasks

- ***Regression***: A typical task in ML is to predict a target numeric value given a set of features (predictors) [21]
- ***Classification:*** It can be seen as converting a regression prediction problem, where the target variable is continuous, to a discrete representation [22]. Past data (labelled items) are used to place new instances into their respective groups or classes in classification problems [22]. Typically, a confusion matrix is used to measure the accuracy of the method (Figure 2-6). Other standard metrics are accuracy, true positive rate, true negative rate and precision.

		True/Observed Class	
		Positive	**Negative**
Predictive Class	**Positive**	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

Figure 2-6. Confusion matrix for binary classification. Adapted from [22]

2.2.2. Logistic Regression

The logistic function is an S-shaped sigmoid function with an output value between 0 and 1 (Equation 2.53). Logistic Regression estimates the probability of an instance to belong to a given class. If this probability is higher than 50%, the model predicts that the instance considered belongs to that class [21].

$$S(z) = \frac{1}{1 + e^{-z}} \tag{2-53}$$

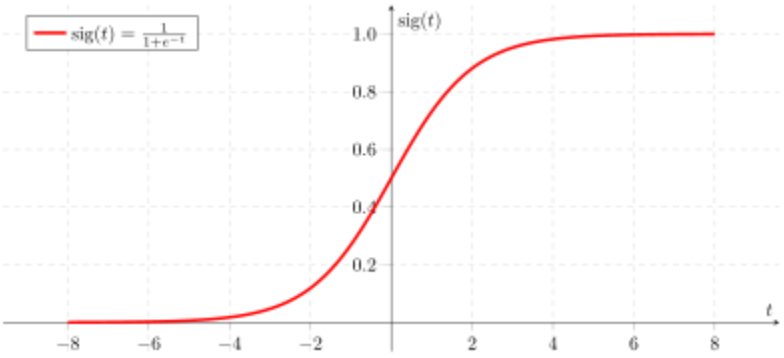


Figure 2-7. Sigma function. Note: Figure was reproduced under creative commons licence from Martin Thoma

2.2.3. Decision Tree

Decision Tree algorithms can perform regression or classification tasks and are capable of fitting complex datasets. Decision Trees build the classification or regression models based on a chain of partitions of the dataset as presented in Figure 2-8 [23]. The goal is to create a model that predicts the value of a target variable by simple decision rules inferred from the data features[24]. They are robust to noise, tolerant to missing information and have a low computational cost. The main tuning parameters in Sklearn library for these models are (a) The maximum depth of the tree, designated as max_depth, (b) The function that measures the quality of a split designated as criterion, (c) The minimum number of samples of a node that the tree must include before the split operation, named min_samples_split, and (d) Tthe minimum samples of a leaf node called min_samples_leaf [21].

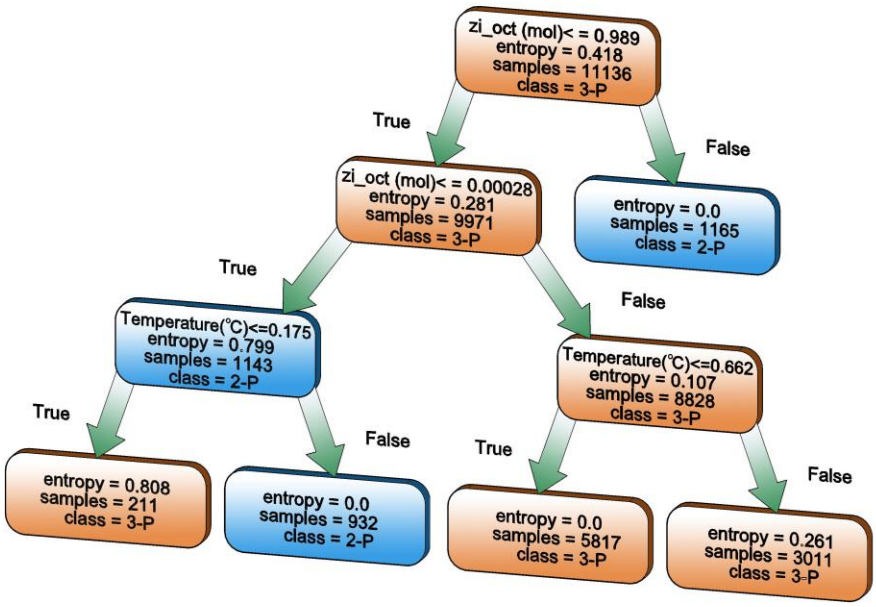


Figure 2-8. Typical Representation of a Decision Tree applied to Hydrocarbon/Water Blends

2.2.4. K-Nearest Neighbors (KNN)

The "nearest neighbour" concept was proposed in 1967 by Cover & Hart [25]. The basis of this model compares attributes of the unknown point to the K number of nearest neighbours to establish similarity (Figure 2-9). KNN is based on the Euclidian distance (Equation (2.54)) between the training and testing datasets. KNN finds the K neighbours that represent the lowest distance. The main parameter of this model is the number of K neighbours. The value of K is selected as a hyperparameter to be optimized [19]. This method compares the attributes related to the data points [26,27].

$$d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2-54}$$

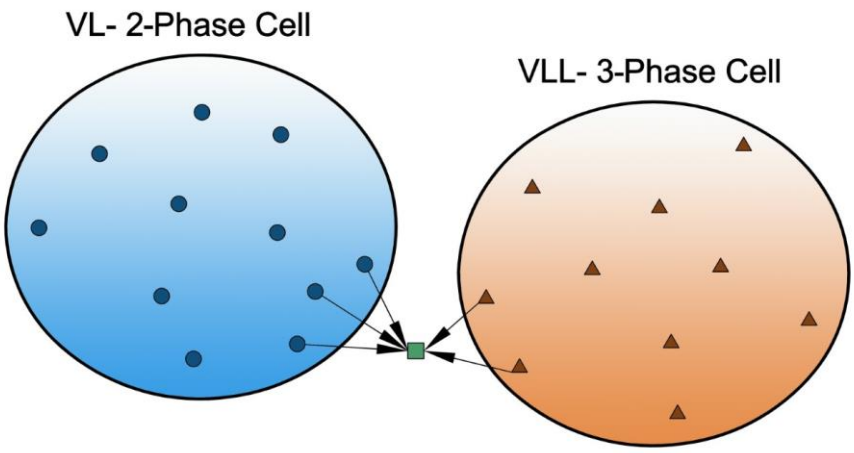


Figure 2-9. Typical representation of a KNN method applied to Hydrocarbon/Water Blends

2.2.5. Support Vector Machine (SVM)

A Support Vector Machine [28] can be used for regression or classification problems. Using a process known as the "kernel trick" there is always one higher dimension where a mapped data set is linearly separable [19,29]. In this sense, the model's objective is to map the X input vectors via a kernel function (e.g. polynomial kernel, radial basis, multilayer perceptron kernel) and to make it a linear regression (Figure 2-10). In this study, a radial basis function was used. The tuning parameters characteristics of this model are the *C* regularization parameter and the kernel scale width [30].

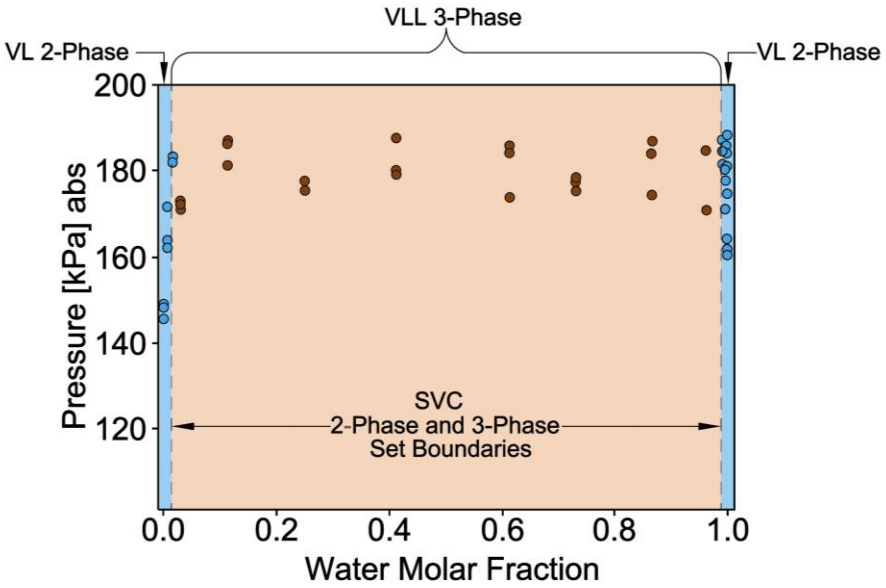


Figure 2-10. Typical Representation of a SVC Classification Algorithm applied to Pressure and Composition at a set Temperature for Hydrocarbon/Water Blends

2.3. State of the Art

2.3.1. Experimental Studies

The experimental study of hydrocarbon/water phase behaviour has been reported since 1980 as summarized in Table 2-4. Van Konynenburg and Scott [31] proposed a classification for the phase behaviour of binary mixtures based on critical points. According to van Konynenburg and Scott classification scheme, water + hydrocarbon binary mixtures present a Type II or Type III phase behaviour classes [32].

One can notice in Table 2.4 the vast technical literature available involving a diversity of water-hydrocarbon blends including paraffins, aromatics, cycloparaffins and olefins. One can also see that the reported studies consider vapour-liquid and liquid-liquid equilibrium over different temperature and pressure ranges.

Table 2-4. Literature References of Experimentally Studies hydrocarbon/water mixtures

Hydrocarbon/water	Conditions	Ref.
Propane	Vapour-liquid equilibrium Temperature: 312 – 387 °C Pressure: 15 – 200 MPa	[33]
n-Pentane	Vapour-liquid and liquid-liquid equilibrium Temperature: 21 – 280 °C Pressure: 0.1 – 14 MPa	[34]
n-Hexane	Vapour-liquid equilibrium Temperature: 337 – 402 °C Pressure: 15 – 140 MPa	[35]
n-pentane, n-heptane	Vapour-liquid equilibrium Temperature: 327 – 402 °C Pressure: 15 – 170 MPa	[36]
n-Butane, n-hexane	Vapour-liquid equilibrium Temperature: 227 – 427 °C Pressure: 15 – 170 MPa	[37]
n-Decane	Vapour-liquid and liquid-liquid equilibrium Temperature: 300, 320, 340 °C Pressure: 13 – 230 bar	[38]
n-Dodecane, squalane	Vapour-liquid and liquid-liquid equilibrium Temperature: 327- 387 °C Pressure: 9 – 310 bar	[39]
1.3-Butadiene, 1-butene, 2-butene, isobutylene, n-butane, isobutene, n-pentane, isopentane, benzene, n-hexane, cyclohexane, n-heptane, 1-heptene, 1.5-hexadiene, **n-octane**	Mutual solubilities. Temperature: 5 – 25 °C	[40]
n-Pentane, n-hexane, n-heptane, **n-octane**, n-nonane, n-decane, n-undecane, 2-methylbutane, 2,2-dimethylbutane, 2,3-dimethylbutane, 2-methylpentane, 3-methylpentane, 2,2-dimethylpentane, 2,3-dimethylpentane, 2,4-dimethylpentane, 3,3-dimethylpentane, 2-methylhexane, 3-methylhexane, 3-methylheptane, 2,2,4-trimethylpentane, 2,3,4-trimethylpentane, 2,2,5-trimethylhexane, cyclopentane, cyclohexane, methylcyclopentane, cycloheptane, methylcyclohexane, cyclooctane, cis-1,2-dymethylcyclohexane, ethylcyclohexane, 1,2-dymethylcyclohexane, ethylcyclohexane, 1-butylcyclohexane, pentylcyclopentane	Mutual solubilities. Liquid-Liquid equilibrium. Temperature: 0 – 430 °C	[41]
n-Octane	Vapour-liquid equilibrium Temperature: 5 – 75 °C	[42]
Ethylbenzene, Ethylcyclohexane, and **n-Octane**	Vapour-liquid equilibrium Temperature: 0 – 568 °C	[43]
Benzene, cyclohexane, n-hexane	Vapour-liquid equilibrium Temperature: 0 – 568 °C	[44]
n-Pentane, n-hexane, n-heptane, **n-octane**, n-nonane, decane, hexadecane	Mutual solubilities Temperature: 25 °C	[45]
n-Pentane, n-hexane, n-heptane, **n-octane**, 2-methylbutane, 2-methylpentane, 3-methylpentane, 3-methylhexane, 2,2-dimethylbutane, 2,3-dimethylbutane, 2,4-dimethylpentane, 2,2,4-trimethylpentane, 2,3,4-trimethylpentane, 2,2,5-trimethylhexane, benzene, toluene, ethylbenzene, o-xylene, m-xylene, p-xylene	Mutual solubilities Temperature: 0-25 °C	[46]
n-alkanes: C1-C12, C14, C16, C18, C20, C24, C25, C26, C28, C30, C32, C36 → **n-octane**	Liquid-Liquid-Gas equilibrium Temperature: 357 – 387 °C Pressure: 19 – 23 MPa	[47]

Some of the thermodynamic models considered by various authors to describe the phase equilibrium of hydrocarbon/water mixtures are summarized in Table 2-5. This table includes activity coefficient models such as NRTL and UNIQUAC and equations of state such as Cubic Plus Association (CPA) with statistical based thermodynamics models being highlighted.

Table 2-5. Literature References addressing Phase Equilibrium Thermodynamic modelling for hydrocarbon/water mixtures

Hydrocarbon/water	Model	Phase Equilibrium	Ref.
n-Octane	NRTL, UNIQUAC (activity coefficient)	Flash point	[48]
n-Heptane, **n-octane**, toluene, propylbenzene, cyclohexylamine (CHA)	NRTL	LLE (ternary)	[49]
Hexane, decane, butylcyclohexane, 1-hexene, 1-octene, benzene, ethylbenzene, 1,3-diethylbenzene	PC-SAFT	LLE	[50]
Ethane, propane, n-hexane, ethylene, propylene, cyclohexane, 2,2-dimethylpropane, 2,3-dimethylbutane, 2,2-dimethylpentane, benzene, toluene, ethylbenzene,	GC-PR-CPA Eos (with group contribution method)	LLE	[51]
Methane, ethane, propane, n-hexane, **n-octane**, 1-hexene, 1-octene	CPA and SAFT	LLE VLE	[52]
n-propane, n-butane, n-pentane, n-hexane, n-heptane, **n-octane**, n-nonane, n-decane, n-undecane, n-dodecane, n-tridecane, n-tetradecane, n-pentadecane	PC-SAFT	LLE VLE VLLE	[53]
n-alkanes: C5-C16, C18, C20 → **n-octane**	PC-SAFT	LLE VLE VLLE	[54]
Methane, ethane, propane, n-butane, n-pentane, n-hexane, n-heptane, **n-octane**, n-decane	CPA Eos	LLE	[55]
Cyclohexane, n-hexane, n-heptane, **n-octane**, n-decane	CPA Eos	LLE VLE	[56]

Regarding naphtha /water and bitumen/water blends, they have the intrinsic characteristic of displaying low mutual solubilities and form two liquid phases. According to the phase behaviour classification proposed by van Konynenburg and Scott [31], Athabasca bitumen/water mixture is considered as type III_m [32]. Furthermore and

regarding bitumen transport some processes use n-alkanes as a solvent leading to asphaltene bitumen component precipitation. In this sense, the modelling of bitumen/n-alkanes phase equilibria has been approached by different authors are reported in Table 2-6. One should note as well that while using a naphtha fraction as bitumen diluent, there is aromatic constituent fractions to be considered as well [4,57].

Table 2-6. Literature References Related to Experimental Studies using bitumen/hydrocarbon mixtures

Bitumen	Component	Conditions	Ref.
Athabasca	Ethane	Liquid-liquid equilibrium Temperature: 22 °C Pressure: 5 – 9 MPa	[58]
	Propane	Phase behaviour, phase composition, and phase densities Temperature: 30 – 120 °C Pressure: 1 – 6 MPa	[59]
	Propane	Phase behaviour Vapour-Liquid and Liquid-Liquid Equilibrium Temperature: 50 – 200 °C Pressure: up to 10 MPa	[60]
	n-Butane	Liquid-liquid equilibrium Saturated liquid densities and viscosities Peng-Robinson EoS Temperature: up to 200 °C Pressure: 1 – 6 MPa	[61]
Athabasca and Cold Lake	n-Butane	Phase behaviour Temperature: 100, 150, and 186 °C Pressure: up to 4 MPa	[62]
Western Canadian	n-Pentane	Vapour-liquid and liquid-liquid equilibrium Saturation pressures Temperature: 21 – 280 °C Pressure: up to 14 MPa	[34,63]
Peace River	n-Pentane	Vapour–liquid–liquid equilibria, saturation pressures and asphaltene precipitation Peng Robinson EoS Temperature: 21 – 180 °C	[64,65]
Athabasca	n-Decane	Density and dynamic viscosity Weight fractions of n-decane: 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 Temperature: 20 – 344 °C Pressure: up to 10 MPa	[66]
Athabasca	n-Tetradecane	Density and dynamic viscosity Weight fractions of n-decane: 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 Temperature: 20 – 344 °C Pressure: up to 10 MPa	[67]
	Methane, ethane, propane, n-butane, n-pentane, n-decane, n-tetradecane, toluene, xylene		[68]
Athabasca	Toluene	Density and viscosity Temperature: 22 – 190 °C Pressure: up to 10 MPa	[69]
Athabasca	Water	Vapour-liquid-liquid equilibrium 9.2 to 96.6 wt% Athabasca bitumen Temperature: 250 – 370 °C Pressure: 4.2–35.7 MPa	[32]

Regarding hydrocarbon/water blends containing bitumen, their behaviour has been studied using methane, ethane, propane, n-butane and toluene as solvents (Table 2-7).

Table 2-7. Literature related to experiments of solvent/bitumen/water mixtures

Bitumen	Solvent	Conditions	Ref.
MacKay River	Methane, ethane, propane, n-butane	Phase behaviour Temperature: up to 150 °C Pressure: up to 5 MPa	[70]
Athabasca	n-Butane	Phase behaviour Temperature: up to 160 °C Pressure: up to 10 MPa	[71]
Athabasca, Coalinga, Cat Canyon, Huntington Beach, Peace River, Mackay River	Propane	Phase boundaries Temperature: 20 - 160 °C Pressure: up to 10 MPa	[72]
Athabasca	Toluene	Phase behaviour Temperature: 200 - 300 °C Pressure: 2.5 to 12.6 MPa	[73]

However, and to the author's knowledge, there is no published data in the open literature related to more complex mixtures such as naphtha/bitumen/water, which are key stream components in a NRU (Naphtha Recovery Unit). Furthermore, there is also very limited equilibrium data on naphtha/water mixtures. Thus, in-depth analysis of naphtha-water thermodynamics provides a unique opportunity for original studies.

2.3.2. Flash Calculations

Flash calculations with water hydrocarbon blends may involve two phases (vapour-liquid) and three phases (vapour-liquid-liquid). Three-phases calculations are however, more difficult than two-phases calculations [74]. Procedures for three-phases calculations of water/hydrocarbon mixtures have been proposed previously.

In 1952, Rachford & Rice studied the vapour-liquid equilibrium for hydrocarbons and proposed to use as the objective function $\sum_i (x_i - y_i) = 0$ instead of $\sum_i x_i = \sum_i y_i = 1$. This approach eliminates the numerical problems related to multiple roots [74].

Peng & Robinson [75] proposed the so-called Peng Robinson Equation of State for systems containing water and at least two other components. This restriction is based on the degrees of freedom when three phases exist. As stated by Peng & Robinson [75], there are three types of calculations as described in Table 2-8 within this region: bubble –point, dew-point and flash calculation. Peng & Robinson proposed two successive iterations using different objective function according to the type of calculation being developed:

Table 2-8. Objective function according with the type of calculation. Taken from [75]

Type of Calculation	**Objective Function**
Liquid-liquid-vapour bubble-point prediction	$\sum_i x_i^L - \sum_i x_i^W = 0, \quad \sum_i y_i - 1 = 0$
Liquid-liquid-vapour Flash Calculation	$\sum_i x_i^L - \sum_i y_i = 0, \quad \sum_i x_i^W - 1 = 0$
Liquid-liquid-vapour Dew-point prediction	$\sum_i x_i^L - \sum_i y_i = 0, \quad \sum_i x_i^L - 1 = 0$ Or $\sum_i x_i^W - \sum_i y_i = 0, \quad \sum_i x_i^L - 1 = 0$
Liquid-liquid Flash Calculation	$\sum_i x_i^L - \sum_i x_i^W = 0$
Liquid-vapour Flash Calculation	$\sum_i x_i^L - \sum_i y_i = 0$ Or $\sum_i x_i^W - \sum_i y_i = 0$
Liquid-vapour Dew-point prediction	$\sum_i x_i^L - 1 = 0$ Or $\sum_i x_i^W - 1 = 0$

Furthermore, the methodology proposed by Mokhatab [76,77] presents two successive substitution iteration schemes as shown in Figure 2-11. Then, the algorithm for three-phases flash calculation starts assuming the equilibrium relations between the compositions of each phase. The first iterative step checks the convergence of the value of the equilibrium relations calculated using fugacity coefficients. Following this, the second iterative scheme checks the convergence of the objective function: $\sum_i x_i^L =$

$\sum_i x_i^W = \sum_i y_i = 1$. As well, other alternative is to use a successive substitution followed by a Newton Raphson method [78].

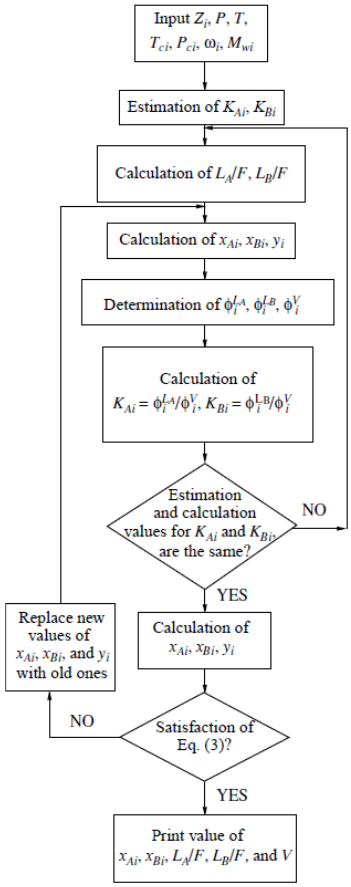


Figure 2-11. Successive substitution iteration diagram for three phases flash calculations. Reproduced with Licence Number 5037061032696 [76]

In order to proceed with calculations, Li & Nghiem proposed the results from Wilson's equation as an initial guess for the equilibrium ratios [79]. As well, Chien [74] proposed to proceed with calculations a two-dimensional search method that divides the calculations into different regions according to the number of phases present in the mixtures at given conditions. Finally, Michelsen's [80,81] approach considers a phase stability test before the phase split for two-phases mixtures and proposed the distance from the plane to the ΔG_{mix} surface (TPD) methodology presented in Chapter 5.

Traditional methods for phase stability test and phase splitting calculations are iterative schemes solving nonlinear equations [82]. These methodologies are computationally expensive and rely on the initial assumption of the equilibrium relations between the compositions of each phase. Additionally, traditional flash calculation methods present convergence and accuracy problems giving results with no physical meaning that are excluded manually [83].In this sense, in the last years, with the advance of computer power, new approaches have been implemented considering phase stability and phase splitting calculations within the same algorithm.

Nazari et al. [84] proposed a stability algorithm based on a negative flash procedure to detect the stable phases. The aqueous phase equilibrium calculations by using EOS lead to an inaccurate prediction of stable phases in the vecinity of phase change regions [84]. These authors obtained a new analytical derivative of fugacity coefficient with respect to mole fraction, improving the accuracy and convergence, and reducing the runtime compared to numerical derivatives [84].

Wang et al. [82] used artificial neural networks to assist traditional flash calculations to achieve fast and robust convergence. The artificial neural networks provided more robust initial guesses for phase splitting calculations [82].

Li et al. [83] compared Newton's method, Sparse Grid Method and a deep learning model for phase splitting problem and flash calculation procedures. These authors propose to use the deep learning model as better first estimates for more thermodynamically rigorous vapour-liquid equilibrium calculations.

Sabet et al. [85] considered a new stability algorithm with application to three-phase flash calculations of hydrocarbon/water in the presence of brine. They modelled hydrocarbon phase with an equation of state (EOS) and aqueous phase using Henry's law. They also proposed and a new initial guess for three-phase flash calculations that assure the convergence of the scheme.

Okuno et al. [86,87] developed a reduced variables method that improved convergence behaviour near the critical region. The method was adapted for hydrocarbon/water mixtures [10], where steam saturation pressure was used as a guide for the trial phase.

On the basis of the above, one can see that the development methodologies with better initial guesses, most efficient and robust calculation methods for the analysis of hydrocarbon/water systems are imperative to reduce the environmental impact of oil and gas processes. Additionally, these methodologies should be able to be extended to more hydrocarbon/water complex mixtures, such as the ones in Dilbit/water systems.

In summary and given the above, an in-depth discussion of flash calculations for water/n-octane and water/PASN is reported in Chapter 6.

2.3.3. Use of Machine Learning (ML) tools for the vapour-liquid equilibrium calculation

Machine Learning (ML) techniques have been used in Chemical Engineering for more than 35 years, helping to solve problems that require pattern recognition and reasoning, and decision making under complex conditions [16]. Machine Learning has been previously applied to thermodynamic problems, the phase stability problem and phase split calculations.

Schmitz et al. [88] proposed a classification methodology to solve the phase stability test, by determining the number and nature of the phases present in the ethanol/ ethyl acetate/ water system, which show an heterogeneous azeotrope. They used Feedforward Neural Networks (FNN) and Probabilistic Neural Networks (PNN) trained with the data obtained from the NRTL model with literature parameters. Their model was able to correctly predict the type of equilibrium for more than 99.9% of the cases.

Poort et al. [89] studied water/methanol mixtures, using classification neural networks for the phase stability and regression networks to calculate thermodynamic properties. The data for training was generated for 101 feed composition, 500 temperatures (273-700 K), and 500 pressures ($1x10^4$–$3x10^7$ Pa). Overall, phase classification showed accuracy

scores that were quite high (around 97%), although classification accuracy of the two-phase region was considerably lower than that of the pure liquid and vapour phase regions. Many property predictions showed good accuracy ($R^2 > 0.95$).

Kashinath et al. [90] studied the isothermal phase equilibria at 260 K and 370 K using a compositional model (3, 6 and 13 components). Data for reservoir conditions was generated by a phase diagram using isothermal negative flash calculations. The authors used Relevance Vector Machines (RVMs) for a classification problem in order to solve the phase stability. Following this, they solved the phase split by using Artificial Neural Networks (ANN), which predicted equilibrium K-values.

Artificial Neural Networks (ANN) for modelling vapour-liquid equilibrium of multicomponent mixtures was studied by Argatob & Kocherbitov (2019) [91]. They proposed ANN as a generalization of Wilson and NRTL models. In this way, they included previous knowledge related to VLE into ANN architecture [91]. They "introduced two ANN-based thermodynamic models for approximating the excess Gibbs energy of a multi-component mixture, which generalize the Wilson and NRTL models" [91].

Jones et al. [88] used Feedforward Neural Networks (FNN) and Probabilistic Neural Networks (PNN) trained with data obtained from NRTL model with literature parameters. They considered a classification problem to solve the phase stability test, determining the number and nature of the phases present in the system ethanol/ ethyl acetate/ water, which present a heterogeneous azeotrope. Their model was able to correctly predict the type of equilibrium for > 99.9% of the cases.

Mohanty [92] found better results for ANN compared to SRK-EoS and PR-EoS for calculating the VLE of binary mixtures of carbon dioxide/ ethyl caproate, ethyl caprylate or ethyl caprate.

Vaferi et al. [93] used ANN to derive predictive models of bubble point pressure and vapour phase composition using literature data from different binary systems containing ethanol.

Poort et al. [89] studied water/methanol mixtures, using classification neural networks for the phase stability and regression networks to calculate thermodynamic properties. The data for training was generated for 101 z_i composition, 500 temperatures (273-700 K), and 500 pressures ($1x10^4$ – $3x10^7$ Pa). Phase classification showed overall relatively high accuracy scores (around 97%), although classification accuracy of the two-phase region was considerably lower than for the pure liquid and vapour phase regions. Property predictions showed good accuracy for many properties ($R^2 > 0.95$),

Kashinath et at. [90] analyzed the isothermal phase equilibria at 260 K and 370 K using a compositional model (3, 6 and 13 components). Data was generated by phase diagram for reservoir conditions using isothermal negative flash calculations. They used a Relevance Vector Machines (RVMs) for a classification problem to solve the phase stability, with phase split via ANN predicting equilibrium K-values. In both cases, they set input blend composition and pressure.

In spite of progress, one can notice that while ML has been employed for a limited number of water-hydrocarbon blends, with its successful application to octane-water and naphtha-water blends still missing. As shown in the present Ph.D. Dissertation, ML application is successfully shown in the present Ph.D. thesis for both phases stability analysis and hydrocarbon solubility, with this being an important original contribution of the present study.

2.4. Conclusions

Chapter 2 reports the following critical topics for the development of this research:

a) Thermodynamic equilibrium fundamental concepts including phase stability and phase split calculations, needed for the development of this research.
b) Machine Learning (ML) concepts required for the data analysis of the present study
c) Hydrocarbon/water mixtures thermodynamics and ML techniques as relevant for supporting the number of phases, solubility and flash calculations findings relevant to the research developed.

3. Mathematical Model Validation and Experimental Setup

During the development of this research, two thermodynamic models were used. An activity coefficient model (NRTL) to describe the n-octane/water mixtures, and an Equation of State (Soave Redlich Kwong with Kabadi-Danner modification) to describe the more complex mixtures of Paraffinic Aromatic Synthetic Naptha/water.

This chapter provides model validation for i) the NRTL model, ii) the Soave Redlich Kwong model with a Kabadi-Danner modification. Additionally, it provides information about the CREC-VL-Cell [94,95] employed to obtain the experimental data, which is then used for this research.

3.1. NRTL Model in Python

Regarding the activity coefficient models, they can be defined in terms of the excess Gibbs free energy (Equation (3-1)), with excess variables representing deviations from the ideal behaviour.

$$\ln \gamma_i = \frac{\overline{G_i}^E}{RT} \tag{3-1}$$

The NRTL model is based on local composition theories and can be implemented using Equations (3-2) to (3-4). g_{ij} represents the interaction energy and α is set at 0.2-0.3 as recommended, while accounting for local composition variations [8,96] as follows:

$$\frac{G^E}{RT} = \sum_i x_i \frac{\sum_j \tau_{ji} G_{ji} x_j}{\sum_j G_{ji} x_j} \tag{3-2}$$

$$\tau_{ij} = \frac{g_{ij} - g_{jj}}{RT}, \tau_{ii} = 0, G_{ij} = exp\left(-\alpha \tau_{ij}\right) \tag{3-3}$$

$$\ln \gamma_i = \frac{\sum_j \tau_{ji} G_{ji} x_j}{\sum_j G_{ji} x_j} + \sum_j \frac{x_j G_{ij}}{\sum_k x_k G_{kj}} \left(\tau_{ij} - \frac{\sum_k x_k \tau_{kj} G_{kj}}{\sum_k x_k G_{kj}} \right) \tag{3-4}$$

The NRTL model parameters are obtained from the Aspen Plus V9 software and from Klauck et al. (2006) [49]. The NRTL model obtained from Aspen Plus V9 was used as a

comparison reference with the NRTL model implemented in Python. NRTL parameters from Aspen Plus, as well as the related literature references [48,49] are reported in Table 3-1.

Table 3-1. NRTL Parameters for a Water(1)/Octane(2) System.

	Aspen Plus V9	Klauck et al. [49] and Liaw [48]
Model	$\tau_{ij} = a_{ij} + \frac{b_{ij}}{T} + e_{ij} \ln T + f_{ij} T$	$\tau_{ij}\mathrm{T} = a_{ij} + b_{ij} T + c_{ij} T^2$
	Parameters	
a_{ij}	$a_{12} = 1.2166$, $a_{21} = -12.035$	$a_{12} = -169.718$, $a_{21} = 4197.06$
b_{ij}	$b_{12} = 2997.7$, $b_{21} = 5381.43$	$b_{12} = 12.5591$, $b_{21} = -7.5243$
c_{ij}	$c_{12} = c_{21} = 0.2$	$c_{12} = 0$, $c_{21} = 0$
d_{ij}	$d_{12} = d_{21} = 0$	N/A
e_{ij}	$e_{12} = e_{21} = 0$	N/A
f_{ij}	$f_{12} = f_{21} = 0$	N/A
α_{ij}	$\alpha_{ij} = c_{ij} + d_{ij}(T - 273.15K)$	0.2

Figure 3-1 and Figure 3-2 report data obtained using the NRTL model from Aspen Plus V9 and the NRTL model developed in Python. For model validation in Python, both the Aspen Plus V9 and the Klauck's et al. [49] parameters were employed. On this basis, one can conclude that NTRL model both in Aspen Plus and in Python displays reasonable agreement.

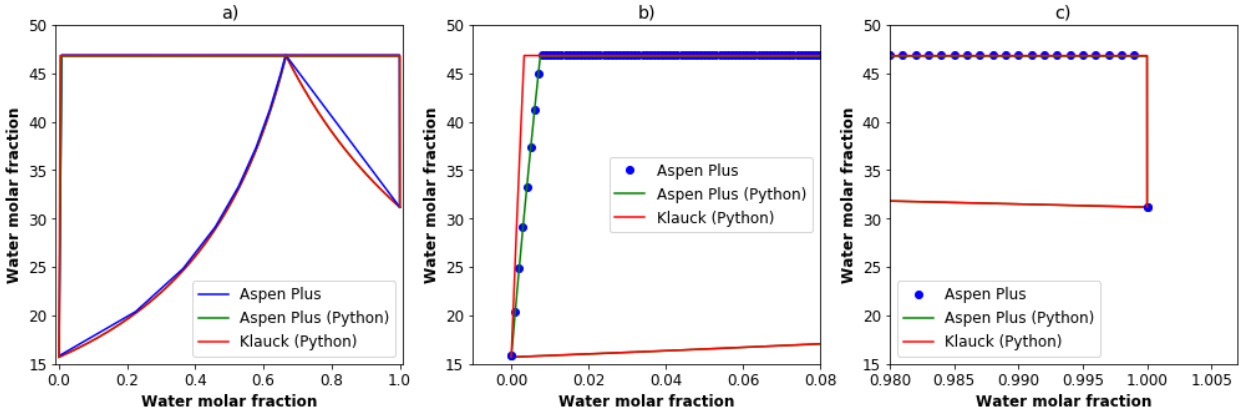


Figure 3-1. Comparison between the NRTL Model in Aspen Plus and the NRTL Model in the Python Software (using Aspen Plus and Klauck Parameters) at 70°C.

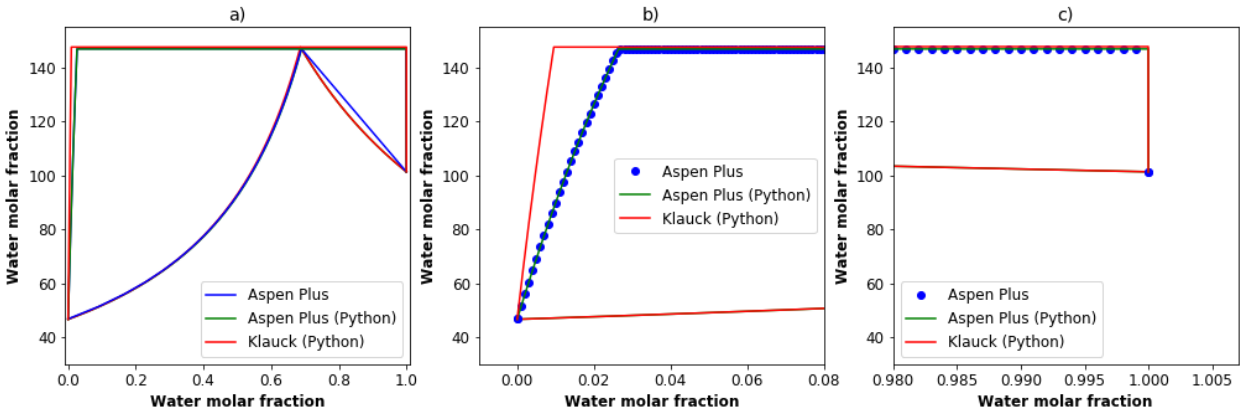


Figure 3-2. Comparison between the NRTL Model in Aspen Plus and the NRTL Model in the Python Software (using Aspen Plus and Klauck Parameters) at 100°C.

Furthermore, one can state that the NRTL algorithm developed, in the context of the present Ph.D. Dissertation, using Python, is in agreement with the Aspen Plus models and could therefore be the basis of a reliable approach for VLL hydrocarbon/water blends.

3.1.1. Comparison of Predicted Three Phase Region

Table 3-2 reports the Three Phase Region (TPR) (also known as the azeotropic point) calculations, obtained using the Python model of the present study, with the NTRL Activity Coefficients model [48].

One can notice that the predicted temperature for the TPR, at atmospheric pressure, is 89.5°C, as shown in Table 3-2. This value is close to the ones previously reported in the technical literature: 89.89 °C with a 0.6822 vapour water molar fraction [97], and 89.76 °C with a 0.6787 vapour water molar fraction [41].

Table 3-2. Predicted TPR at Different Temperatures for Water(1)/N-Octane(2) Blends Using the Klauck's Parameters [49].

Temperature (°C)	Pressure at TPR (kPa)	x_1^I	x_2^{II}	y_w^{TP}
80	70.39	0.0047998	9.1363×10^{-7}	0.6730
89.5	101.32	0.006714	9.2015×10^{-7}	0.6796
90	103.10	0.00682	9.2068×10^{-7}	0.6799
100	147.48	0.009561	9.3652×10^{-7}	0.6867

Furthermore, Figure 3-3 reports the comparison of the NRTL model using the Klauck et al. parameters, (2006) [49] with the experimental data from the technical literature, for the n-octane/water mixture at isobaric conditions. One can see, in this case, the good predictions obtained using the NRLT model.

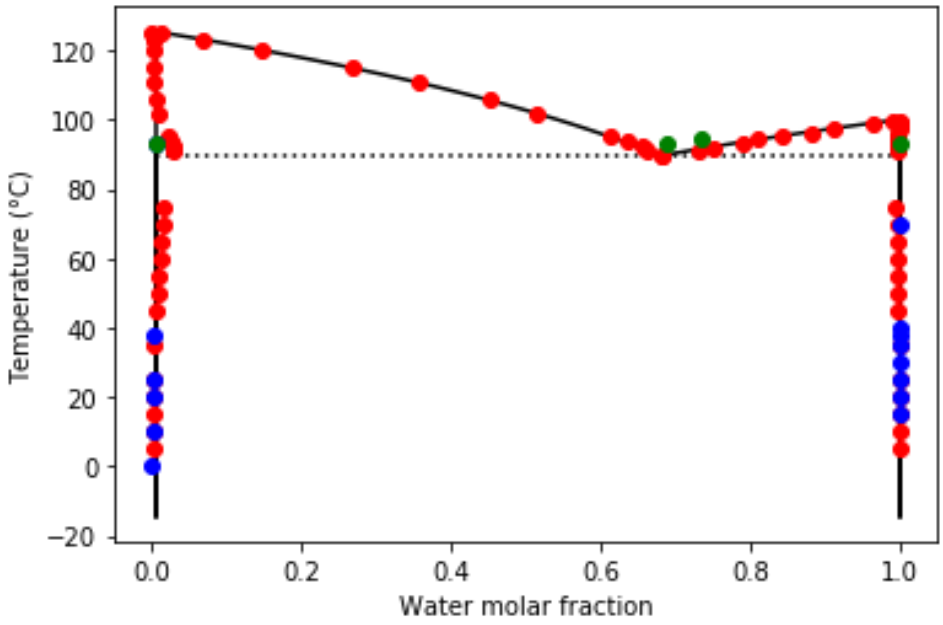


Figure 3-3. Comparison of the NRTL Model Using the Klauck et al. [49] Parameters, with the Reported Literature Data by Tu et al., (1998) [97] in Red, by Haarmann et al., (2018) [53] in Green and Mączyński et al., (2004) [41]in Blue.

Thus, and on this basis, one can further confirm the ability of the NRTL algorithm of the present study to predict TPR conditions, mutual solubilities and VL equilibrium.

3.1.2. Vapour-Liquid-Liquid Equilibrium using an NRTL Model

An NRTL activity coefficient model [96] for VLL calculations of n-octane/water mixtures, was implemented in the present study, using Python. For low pressures (close to 1 atm) and typical conditions in a NRU (Naphtha Recovery Unit), an activity coefficient model was adopted.

The proposed activity coefficient model (NRTL) involves correction factors for the chemical potential and the fugacity, accounting for non-ideal interactions between chemical species [8].

One should note as well, that the procedure for the calculation of VLL equilibrium, at isothermal and isobaric conditions, considers the coexistence of three VLL phases, as described in Equations (3-5) and (3-6).

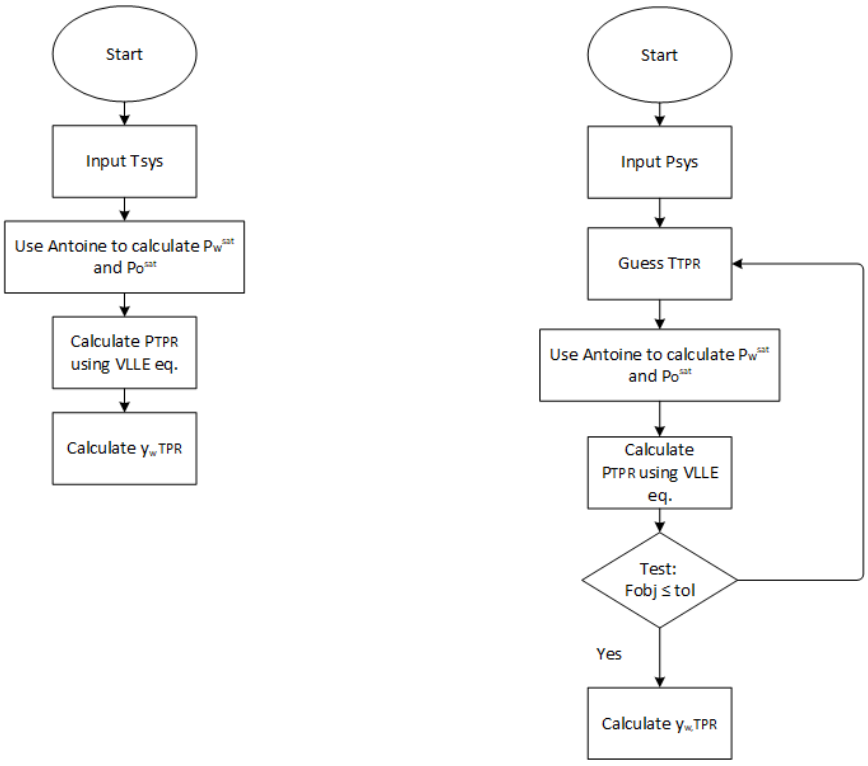
$$P_{TPR} = \sum P_i = x_1^I \gamma_1^I P_{v,1}^{sat} + x_2^{II} \gamma_2^{II} P_{v,2}^{sat} \tag{3-5}$$

$$y_{1,TPR} = \frac{x_1^I \gamma_1^I P_{v,1}^{sat}}{P_{TPR}} \tag{3-6}$$

with $P_{v,\mathrm{i}}^{sat}$ representing the vapour pressure of the *i* component (1 for water and 2 for n-octane), with γ_{i}^{I} representing the activity coefficient for phase I, and with γ_{i}^{II} representing the activity coefficient for phase II.

Regarding the cases with two liquid phases being present, as in water and hydrocarbons, the chemical species involved included partially miscible phases. As well, both liquid-liquid phases can contribute to vapour pressure, forming a three-phase system: two liquid phases and a single vapour phase (VLL) [8]. This “Three Phase Region Domain or (TPR)” can be represented using T_{TPR} and $P_{TPR,}$ with all three phases (vapour-liquid-liquid) containing different fractions of the various chemical species [74].

Figure 3-4 reports the calculation procedure for the TPR conditions, in the case of *Pxy* (fixed T) and *Txy* (fixed P) calculations. One can notice in Figure 3-4, that when the temperature is fixed, the establishment of a *Pxy* involves a direct calculation. However, in the case of calculating *Txy* at a set pressure (Figure 3-4b), the process of calculation becomes iterative, and one has to use a Newton-Raphson or a successive iteration algorithm with set objective functions.



a) b)

Figure 3-4. Algorithm for Three Phase Region (TPR): a) Calculations at a Fixed T and b) Calculations at a Fixed P.

Regarding the objective functions to be considered, they involve the mutual solubilities of the phases. This equilibrium condition is set, given the need of complying with the equality of the liquid fugacities of both phases at equilibrium, and the calculation of the Three Phase Region (TPR) pressure as follows:

$$F_{obj\,1,i} = \sqrt{(x_i^I \gamma_i^I - x_i^{II} \gamma_i^{II})^2} \tag{3-7}$$

$$F_{obj\,2} = \sqrt{\left(P - \left(x_1^I \gamma_1^I P_{v,1}^{sat} + x_2^{II} \gamma_2^{II} P_{v,2}^{sat} + P_{air}\right)\right)^2} \tag{3-8}$$

Thus, to obtain *Txy* or *Pxy* equilibrium values, the mutual solubilities at VLL equilibrium must be calculated by solving Equation (3-7), with the *fsolve* function in Python. This

function is a wrapper around MINPACK's hybrid and hybrid algorithm for solving non-linear equations [98].

Finally, the VL equilibrium can be established using Equations (3-9) and (3-10).

$$P = \sum P_i = x_1^I \gamma_1^I P_{v,1}^{sat} + x_2^I \gamma_2^I P_{v,2}^{sat} \tag{3-9}$$

$$y_1 = \frac{x_1^I \gamma_1^I P_{v,1}^{sat}}{P} \tag{3-10}$$

3.2. Soave-Redlich-Kwong-Kabadi-Danner (SRKKD) Equation of State in Python

Traditionally, the Peng-Robinson (PR) Equation of State (EoS) is one of the most popular EoS, for predicting hydrocarbon-based PVT behaviour, including vapour pressures [99]. When using simulation software, such as HYSYS V9 or Aspen Plus V9, it is considered one of the most enhanced models with an extensive binary interaction parameter database.

However, the PR EoS displays limitations when the considered blends include water or aqueous hydrocarbon mixtures [100]. In these cases, as suggested by previous research from our group [94], the PR EoS does not describe well the system under study, and a different EoS must be used. In binary systems, such as n-octane/water mixtures, an activity coefficient model can be used, as we proposed in our previous work [5].

Nevertheless, classical activity coefficient models are limited to low pressures (≤10 bar) with no C_{7+} species contained. In the context of water and heavy hydrocarbons, such as naphtha or bitumen (C_{7+}), as in the case of this study, a Cubic Equation of State is strongly suggested. In this work, the Soave-Redlich-Kwong (SRK) EoS with a Kabadi-Danner [101] modification is used. These authors suggested that it improves the VLLE calculations for water-hydrocarbon systems, particularly in dilute regions, which is of great interest for this research.

Given the reported advantages of using the Kabadi-Danner modification with the Soave-Redlich-Kwong EoS (SRKKD EoS), for VLL equilibrium calculations, in the case of

hydrocarbon-water blends [102,103], the SRKKD EoS was used to describe the hydrocarbon-water blends of this study. The SRKKD EoS can be defined by Equations (3.11) to (3.15) as follows:

$$P = \frac{RT}{V-b} - \frac{a}{V(V+b)} \tag{3-11}$$

$$a = 0.42748\frac{R^2T_C^2}{P_C}\left[1+\Omega\left(1-T_r^{1/2}\right)\right]^2 \tag{3-12}$$

$$b = 0.08664\frac{RT_C}{P_C} \tag{3-13}$$

$$\Omega = 0.480 + 1.574\omega - 0.176\omega^2 \tag{3-14}$$

$$Z^3 - Z^2 + \left(A' - B' - B'^2\right)Z - A'B' = 0 \tag{3-15}$$

Where $A' = \frac{aP}{(RT)^2}$ and $B' = \frac{bP}{RT}$

The mixing rules required to determine the SRK EoS parameters are given by Equations (3.17) and (3.18), with the Kabadi-Danner modification being reported by Equations (3.16)and (3.19) to (3.21) [101].

$$a_{mix} = a_0 + a_{KD} \tag{3-16}$$

$$a_0 = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j\left(1-k_{ij}\right)\sqrt{a_i a_j} \tag{3-17}$$

$$b_{mix} = \sum_{i=1}^{N} x_i b_i \tag{3-18}$$

$$a_{KD} = \sum_{i=1}^{N} a''_{wi} x_w^2 x_i \tag{3-19}$$

$$a''_{wi} = G_i\left[1-\left(\frac{T}{T_{cw}}\right)^{0.8}\right] \tag{3-20}$$

$$G_i = \sum_l g_l \tag{3-21}$$

With respect to the G_i , it can be calculated using a group contribution method, which accounts for the sum of the contributions of the different functional groups, included in every hydrocarbon molecule. Values from various functional groups can be obtained from the table published by Kabadi-Danner (1985) [101]. The k_{ij} parameters used for the calculations reported in this work, were taken from the HYSYS V9 software.

Furthermore, and to compute the fugacity coefficient with the SRKKD EoS, Equations (3.22) and (3.23) can be used [8] as follows:

$$\ln \phi_i = \frac{b_i}{b}(Z-1) - \ln(Z-B') - C_i{}' \ln \frac{Z+B'}{Z} \tag{3-22}$$

$$C_i' = \frac{A'}{B'}\left(-\frac{b_i}{b} + \frac{2}{a}\sum_{j=1}^{N} x_j \sqrt{a_i a_j}\left(1-k_{ij}\right) \right) \tag{3-23}$$

Finally, and to be able to validate the SRKKD model in Python, the *Pxy* diagram for water/n-octane mixtures can alternatively be obtained using the algorithm described in Figure 3-5.

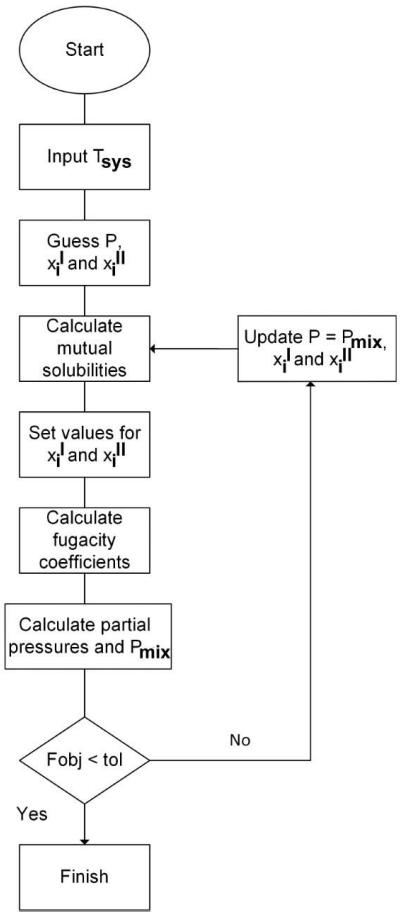


Figure 3-5. Algorithm Used in Python Software to Develop the Pxy Diagram for Water/N-Octane Systems, Using the SRKKD EoS.

One should note that the algorithm described in Figure 3-5, involves a nested calculation with a first iteration being employed to establish the mutual solubility of the liquid phases, as indicated by Equation (3.24). This is followed by a second iterative calculation used to compute the pressure of the mixture and to compare it with the assumed value, as proposed by Equation (3.25).

$$F_{obj\,1,i} = \sqrt{(x_i^I \phi_i^I - x_i^{II} \phi_i^{II})^2} \tag{3-24}$$

$$F_{obj\,2} = \sqrt{\left(P_{mix} - \left(x_1^I \phi_1^I P_{sup} + x_2^{II} \phi_2^{II} P_{sup} + P_{air}\right)\right)^2} \tag{3-25}$$

For the HYSYS V9 simulation, a 3-phase separator was specified, using a 100 kgmol/h blend consisting of a 50%mol water / 50%mol n-octane. This mixture was fed into the unit, at different temperatures, ranging from 80-110 °C, with a vapour fraction of 0.

In the case of the Aspen Plus V9 simulations, a binary analysis was performed. For the first part of the calculations, and in order to validate the program in Python, the air was not considered. The VLLE was calculated under these conditions.

The difference in the Python results with the pressure at the Three Phases Region (TPR) for HYSYS V9 and Aspen Plus V9 at 80°C was 8%. A comparison is presented in Figure 3-6. We consider that the Python version can be used for further analysis as it provides a designated White Box or fundamentally thermodynamically based methodology.

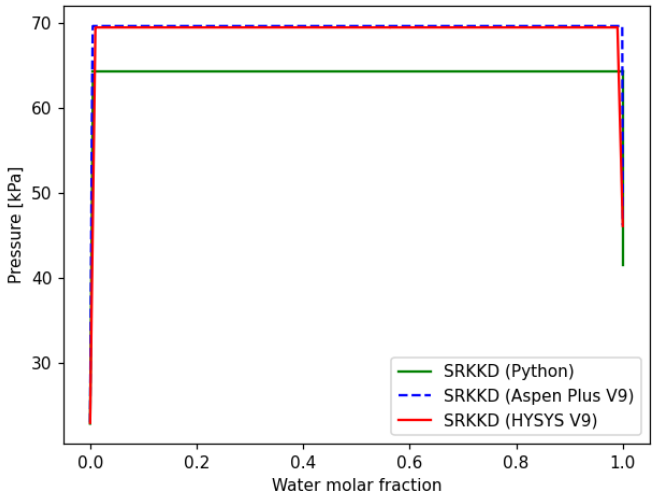


Figure 3-6. Comparison of Results for the SRKKD EoS in Python, Using HYSYS V9 and Aspen Plus V9 at 80°C, in a Water/N-Octane System.

3.3. Experimental Setup

3.3.1. Materials

Distilled water was used in all the experimental studies. The alkanes were obtained from Sigma-Aldrich. The purity of the components was as follows: n-hexane: >97%, n-heptane >96%, n-octane >99%, n-decane >99%, n-dodecane >99%. The water content of the n-

alkanes was 0% for n-octane and n-dodecane, 0.01% for n-hexane and n-decane, and 0.02% for n-heptane. Toluene was obtained from Fisher Scientific with purity >99% and 0.008% water content.

3.3.2. CREC-Vapour Liquid Equilibrium-Cell

The Chemical Reactor Engineering Center (CREC) recently developed a CREC-VL-Cell which allows the measurements of VLL equilibrium (Figure 3-7) using a "dynamic method", with the temperature of the cell increasing progressively, using a thermal ramp of 1.22°C/min. As a result, every run provides a large amount of vapour-liquid equilibrium data (10 Hz), with the vapour pressure data being recorded at various temperatures, every 0.01 seconds. Additional explanations about the cell operation are reported in [94]. Data obtained from this dynamic method has been validated with static measurements [94].

The CREC-VL-Cell uses a marine type of impeller (propeller). The unit propeller helps to ensure the homogeneous mixing of the phases, providing a good heat distribution inside the CREC-VL-Cell. This special cell design proposed by the CREC team, allows one to analyze a process sample directly, avoiding losses of light volatile components due to sample transfers.

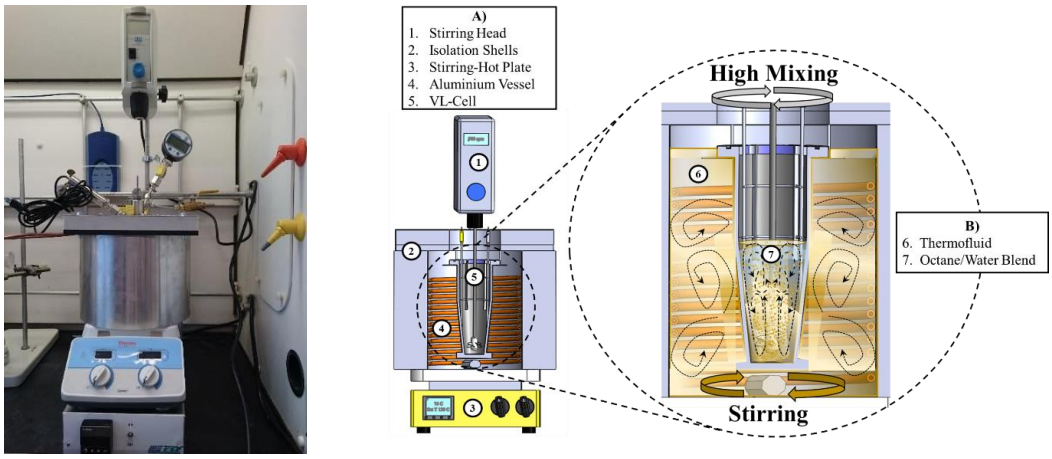


Figure 3-7. CREC-VL-Cell: a) Photography, b) Diagram 1) Stirring Head, 2) Isolation Shells, 3) Stirring Hot Plate, 4) Aluminum Vessel, 5) VL-Cell, 6) Thermofield, 7) Octane/water Blend.

4. Thermodynamics and Machine Learning Based Approaches for Vapour-Liquid-Liquid Phase Equilibria in n-Octane/Water, as a Naphtha-Water Surrogate in Water Blends

A simulation software is typically used in the oil and gas industry to provide a quick process analysis and to facilitate engineering decisions. De Tommaso et al. [104], highlighted the importance of process simulators to build digital twins, facilitating the implementation of industry 4.0 guidelines.

Usually, simulation software are used to establish the project economics, through the optimization of each process step involved [2]. For instance, to optimize the production from oil and gas fields, it is essential to have extensive knowledge of the volumetric and phase changes taking place, from the petroleum reservoir to the oil refinery [13].

When bitumen is extracted from oil sand and a naphtha based process is employed for froth treatment, the Naphtha Recovery Unit (NRU) is employed to recover naphtha from the tailings, for reuse in the process and to reduce the environmental impact of the process. This is an energy-intensive step, with environmental guidelines for naphtha recovery are required to be met [4]. Therefore, the thermodynamics for highly diluted hydrocarbon in water systems is of particular interest. While HYSYS V9 and Aspen Plus V9 software may be used with this objective in mind, the results regarding hydrocarbon/water mixtures from these simulations are not always reliable.

Hydrocarbons are separated from wastewaters before their disposal, usually by using vapour-liquid equilibrium operations. In this sense, the knowledge of the thermodynamic behaviour of hydrocarbon/water systems is of importance. It is well established that the miscibility between water and hydrocarbons is limited. However, the solubility of hydrocarbon in the aqueous phase can be an issue in terms of environmental regulations and process footprint [13].

Liquids exhibit partial miscibility only when their interactions at the molecular scale display strong positive deviations from ideality. In the case of hydrocarbon/water mixtures, these interactions do not yield full liquid-liquid miscibility [6]. However, some partially miscible systems may become fully miscible at higher temperatures, with the effect of total pressure increase being negligible [8].

Water and hydrocarbons do not intermix well. Water tends to segregate from hydrocarbons as a result of the strong polar forces acting between molecules [13]. As expressed by Carlson (1996) [105], most equilibria calculations assume two phases only: Vapour-Liquid equilibrium (VL). However, in hydrocarbon/water blends, three Vapour-Liquid-Liquid (VLL) phases may also contribute, depending on the separator operating conditions. In this respect, the accurate establishment of the number of phases (2 or 3 phases) is critical for phase equilibrium calculations.

The selection of the proper thermodynamic method to represent hydrocarbon/water mixtures is of major importance. To accomplish this, available decision trees were described by others [104,105]. For non-ideal mixtures, however, as is the case of n-octane/water systems, the NRTL model can be used.

Jia et al. (2018) [106] investigated the separation of the n-propanol/water azeotrope, using Aspen Plus, with different thermodynamic models. According to their experimental data, n-propanol/water systems form a homogeneous azeotrope, but Aspen Plus simulations miss predicting it by calculating two liquid phases [106]. On the other hand, de Tommaso et al. (2020) [104], calculated the absence of an azeotrope for the water and acetic acid blends, using available binary parameters from the PRO/II database and the UNIQUAC model.

Moreover, Marcilla et al. (2017) [107] analyzed 25 papers with 70 cases considered for the Liquid-Liquid Equilibrium (LLE) of ternary systems using a NRTL model. In the reported cases, 60% of the cases considered displayed phase inconsistencies in 52% of the papers reviewed. Regarding the number of phases discrepancies reported, they were assigned

to: i) parameters representing partial miscibility in systems that are totally miscible, ii) tie-line inconsistencies that do not satisfy the phase equilibrium criterion, showing meta-stable solutions and non-compliance with the iso-activity condition, and iii) the use of mass fractions instead of molar fractions for model definition.

In Marcilla et al.'s study [107], 12 examples using Aspen Plus for the LLE data regression were described. Three of them reported inconsistencies, with the cause being assigned to the calculation algorithm. In this regard, as Marcilla et al. [107] stated, the use of unreliable parameters can create severe uncertainty, when used in chemical process simulation software. Furthermore, given this situation, Marcilla et al. [107] recommended the use of the minimization of the Gibbs energy of mixing function $(\Delta G_{mix}/RT)$, as an additional condition to ensure the phase equilibrium prediction consistency.

In the case of flash calculations, the phase stability test and phase splitting calculations have also been studied. From an experimental point of view, phase thermodynamic equilibrium is usually measured in experimental setups that provide a limited number of data points in manageable times. Phase equilibrium measurements for dilute hydrocarbon/water systems were made in a specially designed CREC-VL Cell were reported [10]. Unlike earlier experimental techniques, the implemented CREC VL Cell is operated in the dynamic mode with a temperature ramp [94], recording up to 10 points per second of P_{mix} values. This can be considered as "big data" in the context of these experiments. Big data in ML is characterized by data volume (size or scale), variety (multitype), velocity (batch or streaming) and veracity (uncertainty, quality and accuracy) [108–110]. ML is about modelling data [18] and combines statistics, optimization and computer science [19]. ML gives computers the ability to learn without being explicitly programmed [20].

Given the above, the objectives of this work are as follows: i) to establish the problems faced with estimating the number of phases in highly diluted octane/water mixtures when using HYSYS V9 and Aspen Plus V9, and ii) to develop a methodology to predict the correct number of phases, using experimental data obtained in a new CREC VL Cell [94].

Most of the information provided in this chapter was published in a recent article in Processes [5].

4.1. Approach Adopted in the Present Study

A comparison between different thermodynamic models, using HYSYS V9 or Aspen Plus V9, was first attempted through the simulation of a flash unit. Discrepancies in simulation results were noticed depending on the software used. Then the NRTL activity coefficient model was selected and implemented in Python, with the Gibbs energy of mixing function ($\Delta \boldsymbol{G_{mix}/RT}$) being used to explain the discrepancies between HYSYS V9 and Aspen Plus V9 parameters. As well, experimental data from a CREC VL Cell and a t-test data analysis were considered to establish the two phases (VL equilibrium) and the three phases (VLL equilibrium) domains. Furthermore, machine learning methods were implemented in order to obtain an accurate classification of the phase domains in the 80-110 °C range. Accurate classification in this range is of great importance, as it is within the NRU operation conditions.

4.2. Specific Strategy

An octane/water mixture can be considered as a good surrogate for naphtha/water blends. As shown in Table 4-1, n-Octane has properties similar to those of naphtha, which is one of the primary solvents used in bitumen processing.

Table 4-1. Properties for n-Octane and Naphtha.

	n-Octane	Naphtha [111]
Carbon number	8	6-13
Molecular weight (g/gmole)	114.23	145*
Boiling point (°C)	125.6	65-230
Density (kg/m^3)	703	781

* Average molecular weight

HYSYS V9 and Aspen Plus V9 software contain VL and VLL equilibrium modules, that were used as a starting point for evaluating conventional thermodynamic models, in the present work. Water/n-octane systems have been experimentally studied in previous

works [94]. Furthermore, and regarding n-octane-water blends, there is already a significant body of data in the technical literature, as shown in Table 4-2.

Table 4-2. Results in the Technical Literature Related to Experiments with Water/N-octane Mixtures.

Conditions		Ref.
Temperature: 5 – 25 °C	Mutual solubilities.	[40]
Temperature: 0 – 430 °C	Mutual solubilities. Liquid-liquid equilibrium.	[41]
Temperature: 5 – 75 °C	Vapour-liquid equilibrium	[97]
Temperature: 0 – 568 °C	Vapour-liquid equilibrium	[112]
Temperature: 25 °C	Mutual solubilities	[113]
Temperature: 0-25 °C	Mutual solubilities	[114]
Temperature: 357 – 387 °C Pressure: 19 – 23 MPa	Liquid-liquid-vapour equilibrium	[47]

4.3. Gibbs Energy Analysis from Activity Coefficient Model

A detailed description of the thermodynamic equilibrium is essential for water/hydrocarbon mixtures. To accomplish this, three key considerations can be adopted [7]: i) equality of chemical potentials, ii) conservation of mass, and iii) maximization of entropy.

One should note that while chemical potential equality is a “necessary” condition, it is not sufficient to secure solution uniqueness in phase equilibrium calculations [115]. To achieve this, the system should display a maximum entropy. One should note that at a fixed pressure and temperature, the maximization of entropy is equivalent to the minimization of the Gibbs free energy. Thus, the Gibbs free energy of mixing analysis helps to determine this condition [8].

In the case of a binary system, where the reference state of each component is a pure liquid, the Gibbs free energy of mixing for the liquid phase can be calculated, as in Equation (4-1). Additional details of the derivation of these equations are provided in [8]:

$$\frac{\Delta G^{L}_{mix}}{RT} = \sum_{i} x_i \ln(x_i \gamma_i) = x_1 \ln(x_1 \gamma_1) + x_2 \ln(x_2 \gamma_2) \tag{4-1}$$

With x_i representing the molar fraction of each component in the liquid phase.

Thus, to establish the change of mixing Gibbs free energy for the liquid phase as per Equation (4-1), one must vary the x_i values in the 0 to 1 range. In this respect, a common tangent plane criterion can be applied to multiple liquid phases as described in [8,116,117]. Furthermore, and to compare the Gibbs free energy of mixing curve for liquid and vapour phases, it is important that both phases have a common reference state [116]. In this respect, the pure component as a liquid at the same temperature and pressure as the mixture, is selected as the reference state $\left(\frac{G_{i,o}^{L}}{RT} = 0\right)$.

On this basis, Equations (4-2) and (4-3) can thus be considered as applicable [115,116] for the vapour phase as follows:

$$\frac{\Delta G_{mix}^{V}}{RT} = \sum_{i} y_i \frac{G_{i,o}^{V}}{RT} + \sum_{i} y_i \ln(y_i) \tag{4-2}$$

$$\frac{G_{i,o}^{V}}{RT} - \frac{G_{i,o}^{L}}{RT} = \ln \frac{P}{P_i^{sat}} \tag{4-3}$$

Thus, given Equations (4-2) and (4-3), y_i can be varied, establishing as a result, the Gibbs free energy of mixing for the gas phase in the y_i 0 to 1 range. This Gibbs free energy of mixing phase can be also considered to be under the common tangent plane criterion as suggested in [116].

In practice however, it is always useful to know, before conducting the mixing calculations, whether the liquid-liquid blend considered yields a single liquid phase solution, or whether species in the liquid phase may split in more than one liquid phase [8]. This Gibbs free energy of mixing evaluation involves NRTL activity coefficients. This is based on an excess Gibbs energy model and can be applied at low total pressures (≤10 bar), as is the case of the system under study.

4.4. Results and Discussion

4.4.1. Issues with Available Models while Evaluating VLLE

The three-phase equilibrium (VLL) of n-octane/water systems was first considered in the present study, using Aspen Plus V9 and HYSYS V9. To accomplish this, activity coefficient models (NRTL and UNIQUAC), a Peng Robinson Cubic Equation of State and a COMThermo model were evaluated. One should note that activity coefficient models offer an alternative to models which consider equations of state for low pressure systems [8]. For the case of COMThermo, the vapour phase is modelled using the Antoine vapour pressure model, and the liquid phase is modelled using the Margules activity coefficient model. For a thorough comparison between models, both models (activity coefficient and fugacity coefficient) were considered in the present study using Aspen Plus V9 and HYSYS V9 and octane/water blends.

For the simulation, a 100 kgmol/h blend with a 50% mol water / 50% mol n-octane mixture was fed into a flash separator working at different temperatures from 20-120 °C. The bubble point pressure (vapour fraction = 0) and dew point pressure (vapour fraction = 1) were then calculated. A 3-phase separator was specified in HYSYS V9. In Aspen Plus V9, a flash3 separator was set, with water being selected as a key component in the liquid phase.

One should note that while using these models for VLLE, two dominant issues were found:

- Discrepancies between models when running with two different available software (e.g. HYSYS V9 and Aspen Plus V9)
- Inconsistency of the available thermodynamic model predictions (e.g. Aspen Plus V9) with available experimental data.

Figure 4-1 reports bubble point pressure calculations with Aspen Plus V9 and HYSYS V9, while the estimates for dew point pressure can be found in Figure 4-2. One can see that the results from HYSYS V9 differ by a large amount as compared to those from Aspen Plus

V9 (differences up to 104.66%) except when the Peng-Robinson EOS is employed (mean error = 7.98% for boiling point and 3.91% for dew point).

Thus, VLL results when applying HYSYS V9 software must be used with extreme caution, and this given these results are indicators of phase prediction inconsistency, as will be done in the upcoming Section 4.4.2.

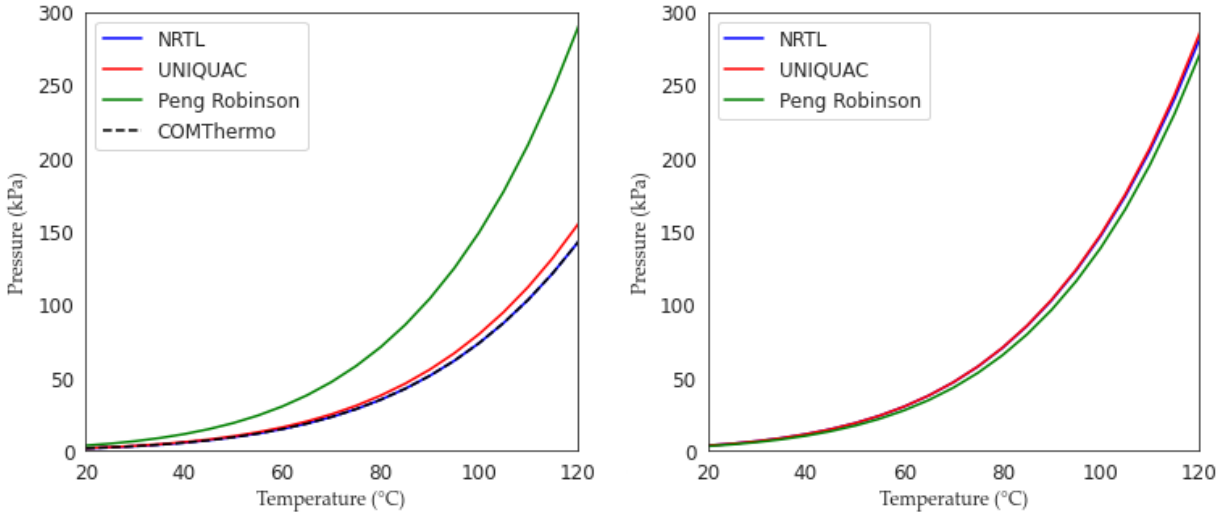


Figure 4-1. Bubble Point Pressure Calculations with Different Thermodynamic Models for a) HYSYS V9 and b) Aspen Plus V9. Note: 0.5 Octane/0.5 water molar fractions.

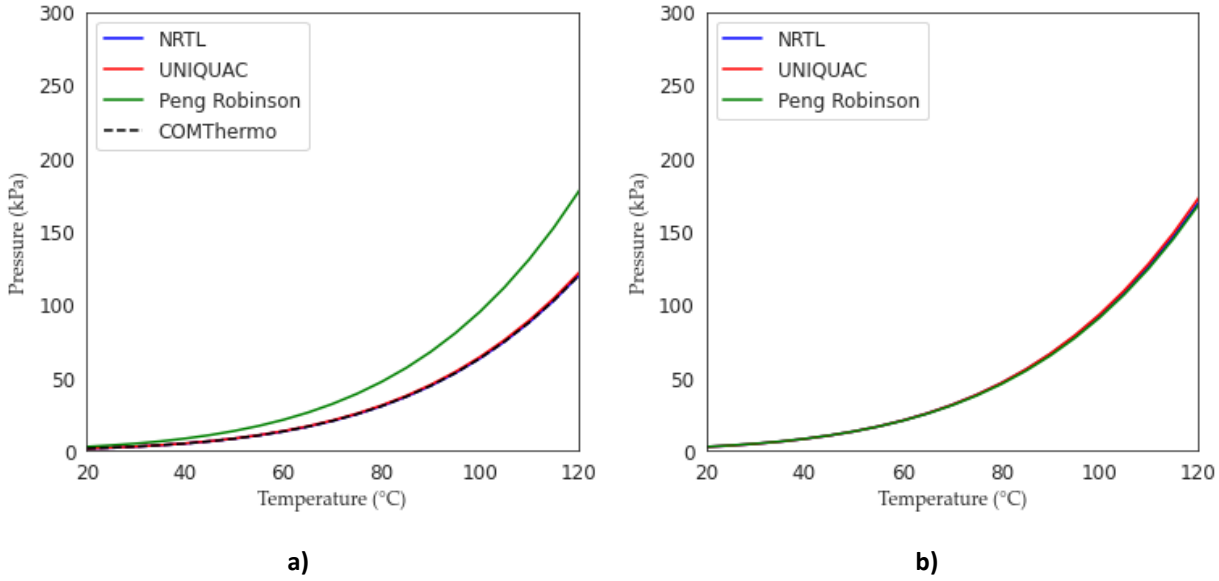


Figure 4-2. Dew Pressure Calculations with Different Thermodynamic models Using a) HYSYS V9 and b) Aspen Plus V9. Note: 0.5 Octane/0.5 water molar fractions.

Table 4-3. Comparison between HYSYS V9 and Aspen Plus V9.

Model	Boiling point Difference	Dew Point Difference
Peng Robinson	Min: 5.67% Mean: 7.98% Max: 11.58%	Min: 1.78% Mean: 3.91% Max: 6.83%
NRTL	Min: 92.88% Mean: 99.95% Max: 104.66%	Min: 34.12% Mean: 49.96% Max: 66.45%
UNIQUAC	Min: 83.20% Mean: 86.28% Max: 90.39%	Min: 37.86% Mean: 62.98% Max: 48.89%

Furthermore, and while reviewing HYSYS V9 VLL results for water/n-octane streams, it was possible to identify that only the Peng-Robinson (PR) Model accounts for two liquid phases, with the activity coefficient models (NRTL and UNIQUAC), considering the octane/water stream as two totally miscible liquids. This single liquid phase misrepresentation does not agree with experimentally observed liquid-liquid phase separations as reported by Kong (2020) [94], and shows the need for developing a reliable methodology for the prediction of the number of phases of hydrocarbon/water systems.

Given the above, the proposed methodology reported here is planned to allow the software user to develop a better than “black box” model (the user does not have access to calculations and parameters), with the user being fully aware of all equations involved. To this end, a NTRL thermodynamic model was chosen for the various calculations. This model was programmed using Python, with the Gibbs free energy analysis considered using binary interaction parameters (BIP) from HYSYS V9, Aspen Plus V9 and the technical literature [48,49]. The aim was to better models leading to two-phase simulations, predicting vapour pressures and three phases region. Finally, experimental data from the CREC VL Cell was also used, and a methodology to predict the number of phases of the n-octane/water system was proposed.

4.4.2. Theoretical Discussion of Model Discrepancy

The Gibbs Energy Analysis from the Activity Coefficient Model described in Section 4.3 was used to understand the differences between simulation software. Figure 4-3 reports the ΔG^{L}_{mix} using the NRTL model at 70°C and 100°C. One should note that the temperatures selected were one lower, and the other higher than the Three Phases Region (TPR) at 1 atm, as reported by Tu et al. [97]. Concerning the BIP (Binary Interaction Parameters), the ones from HYSYS V9, Aspen Plus V9, and Klauck et al. [49] were used. In the case of HYSYS V9, two cases were considered: a) the BIPs parameters for HYSYS V9 were set at the zero default values and b) the BIPs were estimated by HYSYS V9 assuming liquid phase immiscibility.

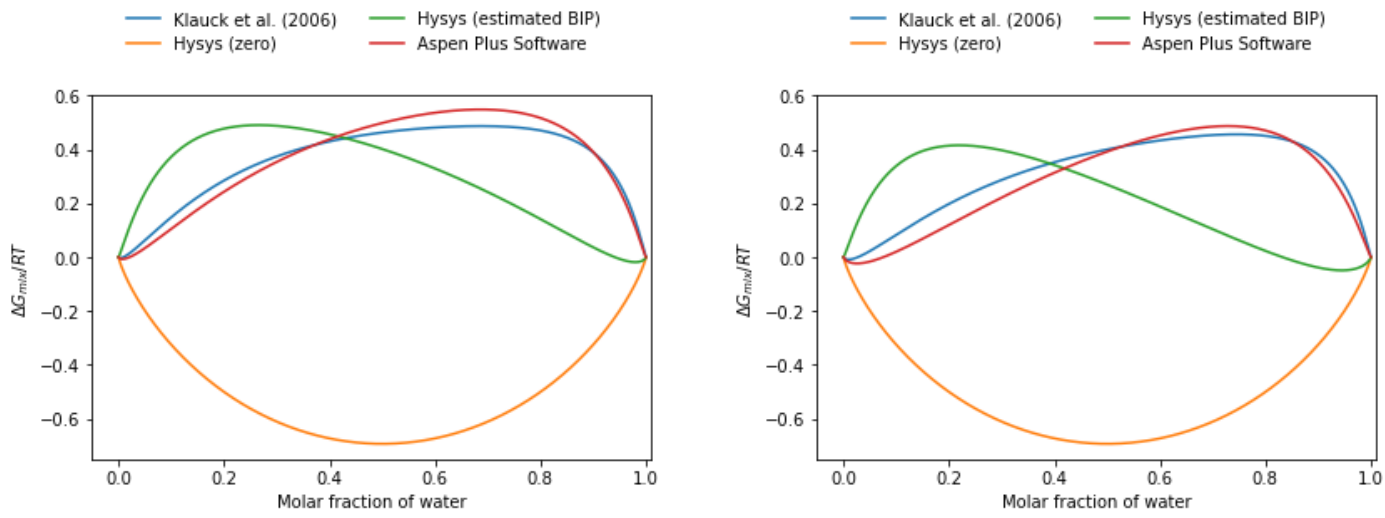


Figure 4-3. Gibbs Free Energy of Mixing for a n-Octane/Water System, at Various Water Molar Fractions and Two Thermal Levels: a) 70°C and b) 100°C.

Figure 4-3 reports that the HYSYS V9 results having the BIPs set to zero, display a catenary-shaped curve (in yellow), with only one anticipated liquid phase for the mixture at both 70°C and 100°C. One should note that the $\Delta G_{mix}/RT$ in HYSYS V9 is inconsistent with experimental observations where a liquid-liquid phase equilibrium is observed [94]. Furthermore, and when considering HYSYS V9 with the non-zero BIPs as reported in Table

4-4, it predicts a phase splitting behaviour. Nevertheless, the $\Delta G_{mix}/RT$ differs significantly from the other $\Delta G_{mix}/RT$ calculated with Aspen Plus V9 and Klauck et al. [49] BIPs.

Furthermore, the calculation of mutual water/n-octane solubilities, assuming a single liquid phase, is considered as shown in Table 4-4. One can see the significant difference of BIP parameters for the various models. As expected, BIPs from HYSYS V9, when set to zero, give a trivial single-liquid phase solution that is not in agreement with experimental results [94]. Furthermore, and when Aspen Plus V9 or Klauck et al. [49] are employed, the solubility of water in the hydrocarbon phase is, as expected, higher than the one for the hydrocarbon in a water phase. On the other hand, one can also observe that in the case of HYSYS V9, with non-zero estimated parameters, the predicted relative solubility is the reverse in magnitude. This means, that there is a discrepancy between the mutual solubilities obtained using the BIP default parameters of the NRTL, and the ones calculated with the HYSYS V9 method.

Table 4-4. Predicted Mutual Solubilities for Water(1) / n-Octane(2) Blends at 70°C and 100°C in liquid phase molar fractions.

	70°C		100°C	
BIP reference	Water in Hydrocarbon phase (x_1^I)	n-Octane in Aqueous phase (x_2^{II})	Water in Hydrocarbon phase (x_1^I)	n-Octane in Aqueous phase (x_2^{II})
Klauck et al. (2006) [6,29]	$3.32245*10^{-3}$	$9.1571*10^{-7}$	$9.56125*10^{-3}$	$9.3652*10^{-7}$
Aspen Plus (Python)	$7.50095*10^{-3}$	$8.1543*10^{-6}$	$2.644388*10^{-2}$	$2.1395*10^{-5}$
HYSYS (estimated BIP)	$8.9570*10^{-6}$	0.02069	$1.3065*10^{-5}$	0.05936
HYSYS (zero) (* one single phase)	$-6.1590*10^{-10}$	1	$-6.1590*10^{-10}$	1

To address these issues, both the “unstable” and the “metastable” regions of the liquid-liquid equilibrium for a n-octane/water system were calculated, as reported in Figure 4-4(a-d). Furthermore, and to establish the boundary between unstable and metastable regions, inflection points complying with the second derivative criteria as in (Equation(4.4)) were considered [8,118]. As explained by Soares et al. (1982) [119], a feed with a composition in the metastable region, may either present as a single liquid phase or alternatively may split and form two liquid phases under external perturbations.

$$\frac{d^2\Delta G_{mix}}{dx_1^2} = 0 \tag{4-4}$$

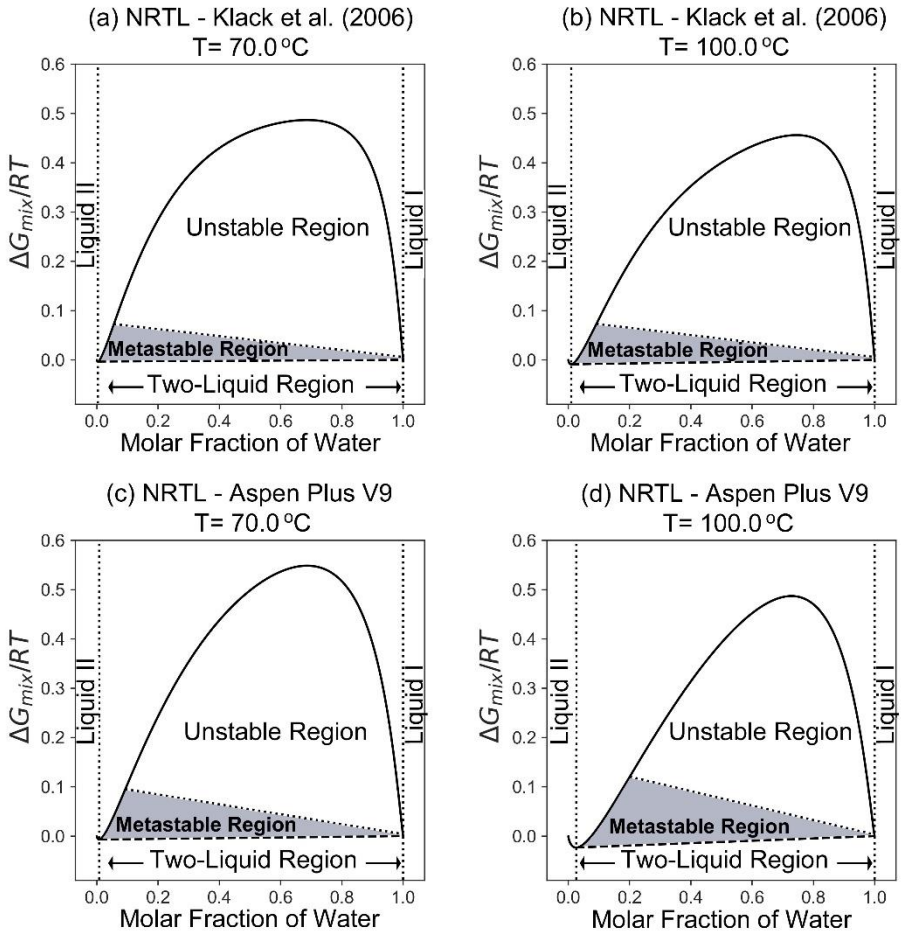


Figure 4-4. Unstable and Metastable Liquid-Liquid Equilibrium Regions for a Water/n-Octane Blends using: (a, b) Klauck et al. parameters [49] and (c, d) Aspen Plus V9 parameters. Parameters reprinted with permission from Klauck, M.; Grenner, A.; Schmelzer, J. Liquid-liquid(-liquid) equilibria in ternary systems of water + cyclohexylamine + aromatic hydrocarbon (toluene or propylbenzene) or aliphatic hydrocarbon (heptane or octane). J. Chem. Eng. Data 2006, 51, 1043–1050, doi:10.1021/je050520f. Copyright 2021 American Chemical Society.

Furthermore, it is possible to observe in Figure 4-4(a-d), that a “stable” region boundary can be established by using a “double tangent line” (black broken line) connecting two ΔG_{mix} points. These “double tangent line” shared points correspond to the mutual miscibility of both phases. The double tangent condition shows the system's stable state

[81,115]. One should note that in the case of the "metastable" region, the temperature and model parameters that are used have an influence over the region, adding uncertainty over the mixture stability.

Figure 4-5 reports the VLLE at 1 atm and 89.5°C, for the water/n-octane blends. This condition corresponds to the calculated TPR (Three Phase Region) at 1 atm. One should note that TPRs at 89.89°C [97] and 86.76°C [41] were previously reported. It is possible to observe, as is suggested in Figure 4-5(a), that the tangent line now contacts the "three" ΔG_{mix} minima points, instead of two, with this corresponding to two liquid phases and one vapour phase condition (Three Phase Region). One can notice as well, that the "three-point tangent line" is better described by Klauck et al. [49] parameters. However, and as shown in the "close up" in Figure 4-5(b), in practice, none of the available models present an exact three-point tangent line. This suggests that there are errors in this prediction and that a better model still needs to be developed.

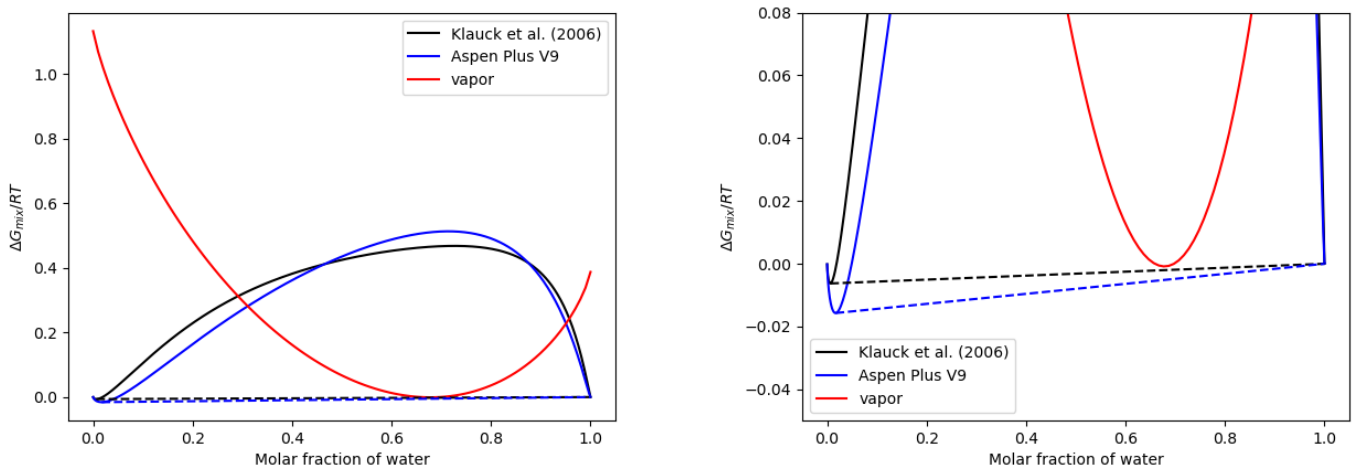
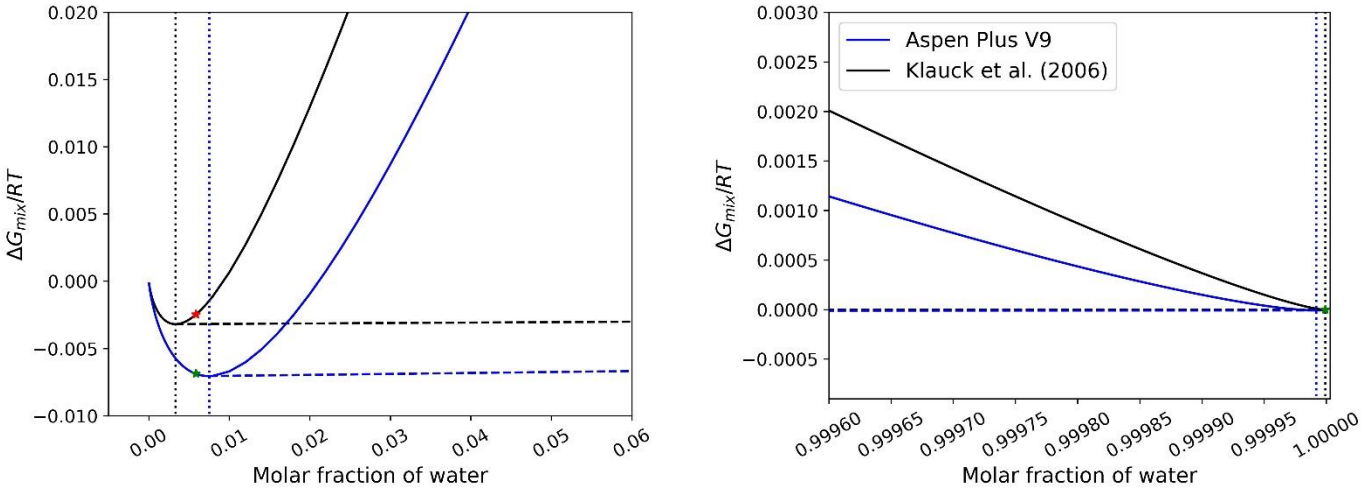


Figure 4-5. Vapour-Liquid-Liquid Equilibrium: a) **Δ**Gmix/RT including the full range of values, b) **Δ**Gmix/RT close-up for water/n-octane system at 1 atm and T = 89.5 °C.

Figure 4-6 provides a closer view of the ΔG_{mix} for VLLE at 70°C, for n-octane-water blends, at low and high n-octane concentration levels, with this showing that evaluating mutual miscibility of hydrocarbon/water systems remains a challenge.

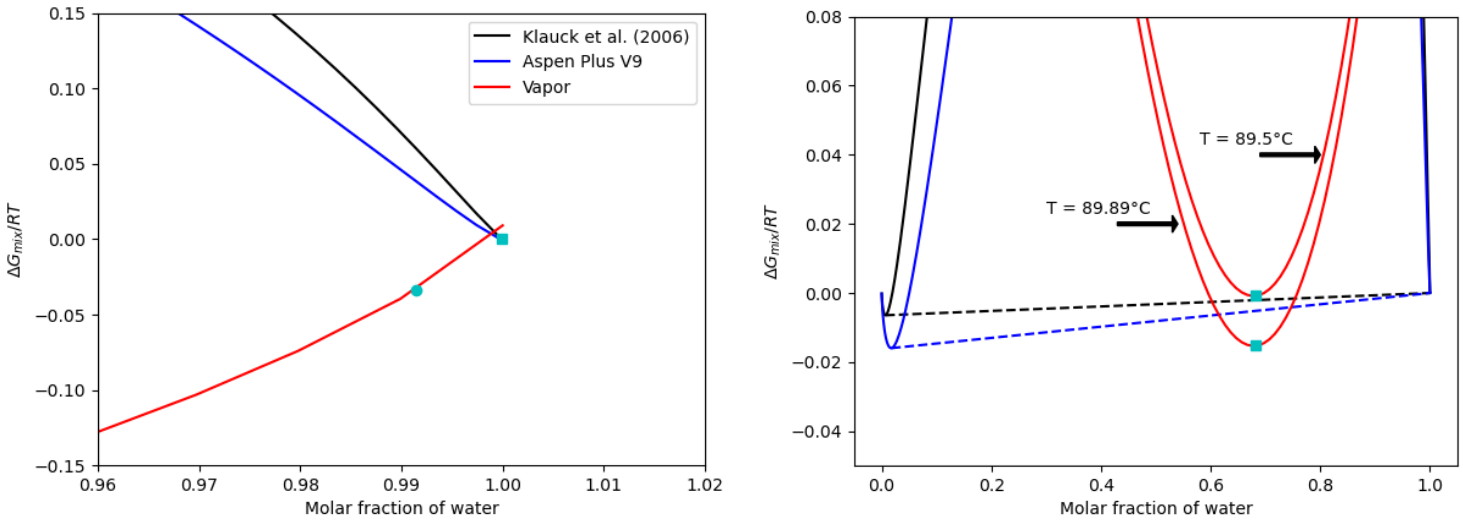
In fact, for the low water molar fractions, the solubility of water in the hydrocarbon phase is described as a change of slope in the ΔG_{mix}. In the same way, it is possible to notice that for high water fraction regions (aqueous phase) the solubility of hydrocarbon in water experiences a flattening of the Gibbs Free energy of mixing. In this sense, the n-octane in the water mixture presents partial miscibility, which is a critical condition to be identified for environmental reasons and process optimization purposes.



(**a**) (**b**)

Figure 4-6. Closer View of **Δ**Gmix/RT and Mutual Solubility Regions of Water/n-Octane system. The NRTL model implemented with win Aspen Plus V9 and Klauck et al. (2006) BIP parameters at 70°C. a) highly diluted water in n-octane region, b) highly diluted octane in water region. Note: reported data points are from [41,97]

However, as presented in Figure 4-7, when the ΔG_{mix} is calculated using published data at 1 atm, for the different phases [49], the need of a better prediction of number of phases is confirmed. This is given the fact that the reported technical literature experimental data points are not located at the minimum value of the Gibbs energy of mixing.



a) **b)**

Figure 4-7. Comparison of the Vapour **Δ**Gmix with Available Data from the Technical Literature for a) VLE at 99.76°C and 1 atm and b) VLLE at 89.5°C and 1 atm. Note: Reported data points are from [97].

Specifically, in the case of Figure 4-7(b), the vapour phase ΔG_{mix} varies significantly when calculated at 89.5°C or alternatively at 89.89°C, the experimental reported value [97]. As a result, the shared tangent line criterion does not strictly adhere in any of these two cases. In this respect, one can only agree that the availability of VLLE data at different temperatures and pressures, such as the ones provided by the CREC VL Cell, are imperative for establishing the TPR region. These data are required to obtain improved modelling of the number of phases of hydrocarbon/water mixtures.

4.4.3. Analysis of Experimental Results

Figure 4-8 reports the various phase regions that one can anticipate when using n-octane/water blends. One can notice that at a given temperature, the following is expected:

a) Three coexisting liquid-liquid-vapour (VLL) phases with the vapour pressure remaining unchanged, while the initial water composition is varied (horizontal broken line)

b) Two liquids phases at higher pressures, with every phase involving highly diluted blends,

c) Two phases, vapour and liquid, with the liquid phase encompassing completely solubilized species.

d) A mixed vapour phase at low pressures.

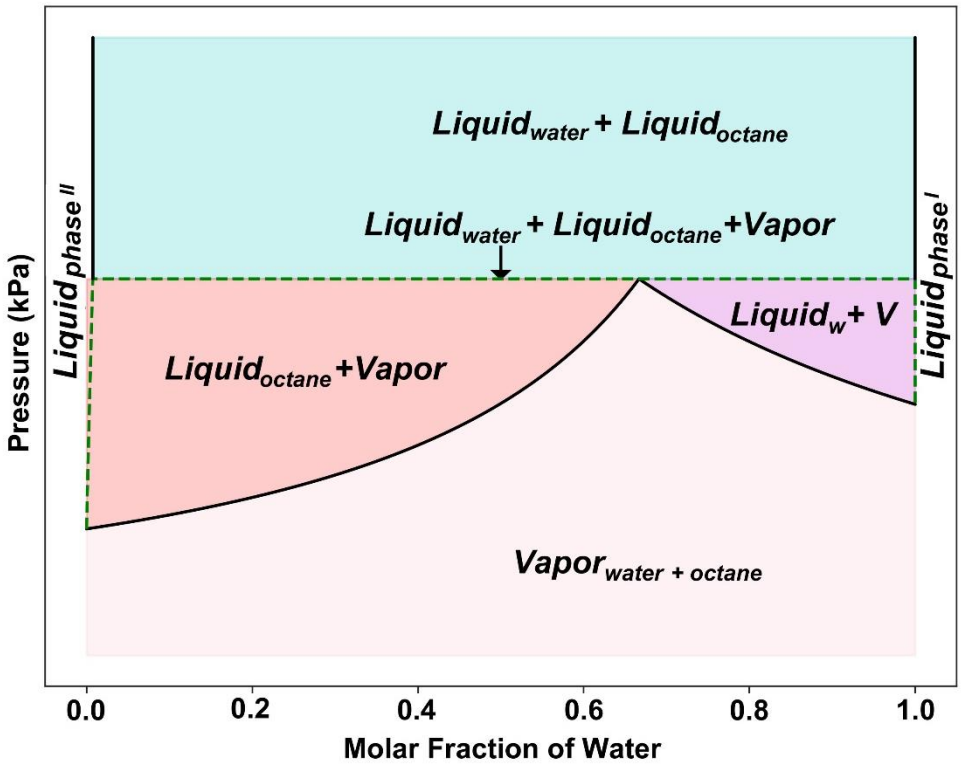


Figure 4-8. Schematic Description of the Two and Three Phase Regions for n-Octane/Water Blends Using the NRTL Model.

In the present study, however, one is especially interested in the behaviour described by the VLL dashed line in Figure 4-8, which corresponds to the Three Phase Region (TPR) and the two vapour-liquid phase domains, in the highly diluted region of a separator unit.

Regarding the experimental data considered in the present study, they are extensively described in Kong (2020) [94]. These vapour-liquid equilibrium measurements were developed at the CREC laboratory using a CREC VL Cell. These experiments were conducted using 17 different mass compositions of n-octane in water and were repeated

at least three times with good reproducibility. Standard deviations for repeats were +/- 4.85 kPa in the 80-110°C range of interest. Given the high density of the experimental data points, curves reported were obtained via linearization of data neighbours, followed by interpolation as needed for comparison of thermal levels.

Figure 4-9 reports the experimental points at different temperatures and octane concentrations, in the range of interest. The experimental setup considers the presence of air, as would occur in industrial operation. In that sense, the pressure of each experimental point and the models used for comparison, consider air.

Baselines with the mean values of pressure at 20% to 98%wt octane compositions were calculated and reported as blue lines in Figure 4-9. These baselines represent two coexisting liquid phases, as confirmed with visual observations in a Plexiglass unit [94]. In the same way, the blue band reports the 95% confidence intervals, calculated from the experimental data. In this range, data has a near-normal behaviour, the confidence intervals are calculated on this basis [120]. One should also note that the red line in Figure 4-9 describes the fully immiscible Two Liquid Model given by ($P_{oct} + P_w$). As expected, the immiscible assumption does not represent the experimental values but rather overestimates them.

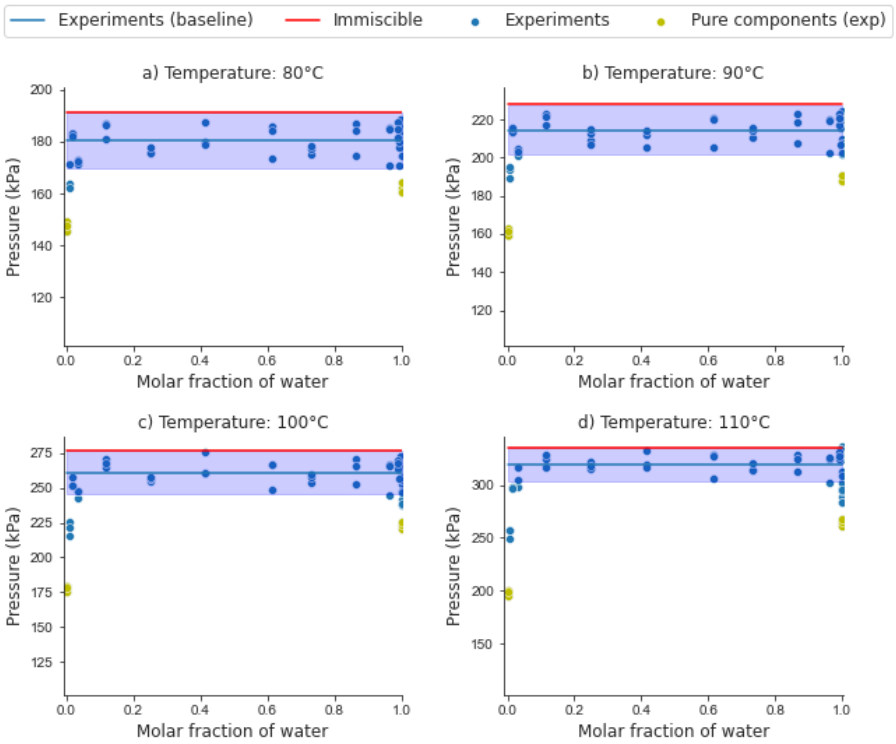


Figure 4-9. Experimental P_{mix} Results at 80°C and 100°C. Note: i) The red line describes the two-phase fully immiscible model, ii) all P_{mix} experimental and model derived points include the presence of air.

Figure 4-10 describes the P_{mix} for highly diluted octane in water (aqueous phase), while Figure 4-11 reports P_{mix} for highly diluted water in octane (hydrocarbon phase). One can notice in both cases, there are significant P_{mix} reductions, with this being attributed to the solubility of highly diluted mixtures. It is also important to notice, that at 100-110°C the highly diluted mixtures change from the TPR to the two-phase region domain. This is consistent with Figure 4-8, where a three-point straight line can be used to explain the presence of the three-phase region (TPR), with two liquid phases and a vapour phase being present.

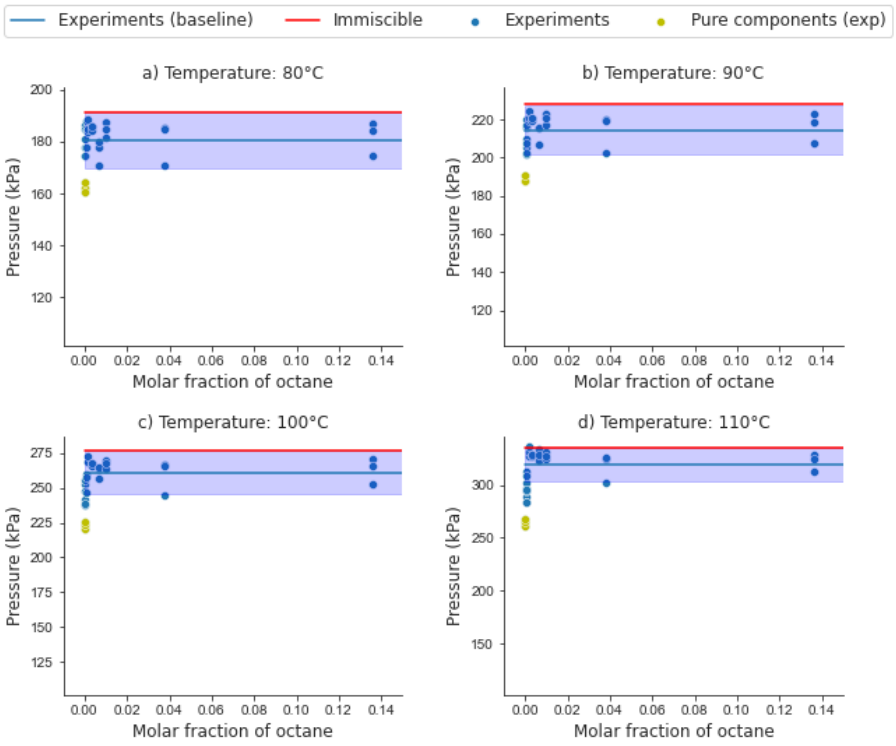


Figure 4-10. P_{mix} for Highly Diluted Octane in Water Mixtures at 80°C, 90°C, 100°C, 110°C.

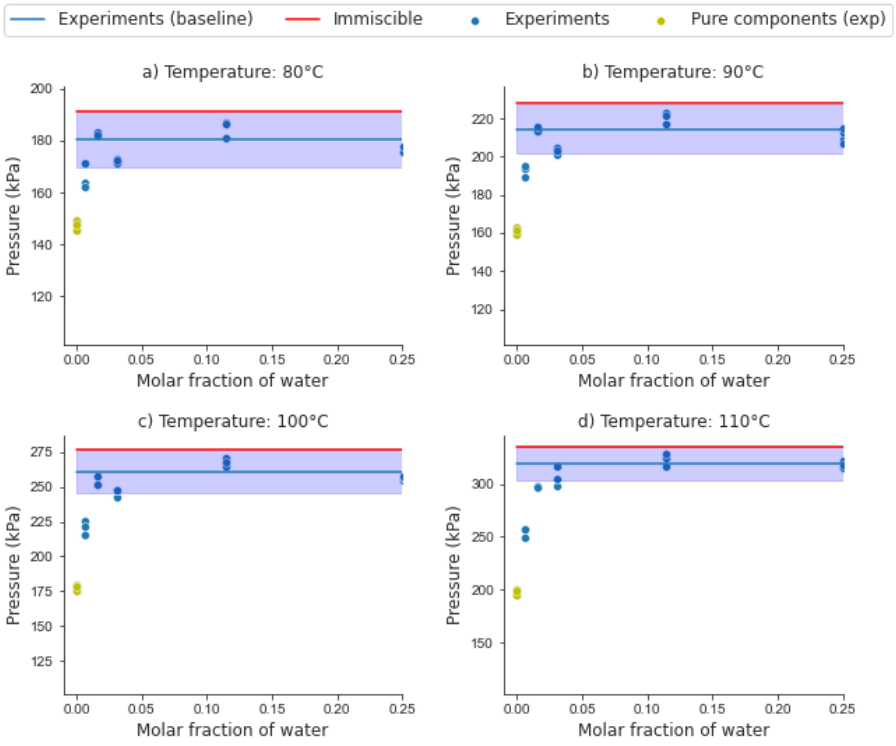


Figure 4-11. Pmix for Highly Diluted Water in Octane Mixtures at 80°C,90°C, 100°C, 110°C.

Furthermore, when the NRTL model results are plotted as in Figure 4-12, together with the experimental data points obtained in the CREC VL Cell, similar trends for the immiscible model can be observed. Here, one can see that the TPR pressure predicted by the NRTL is higher than the experimental values. Thus, better predictions are needed for *Pmix*, as is being considered in Chapter 6, with the emphasis of the present chapter being on the establishment of the right number of liquid phases.

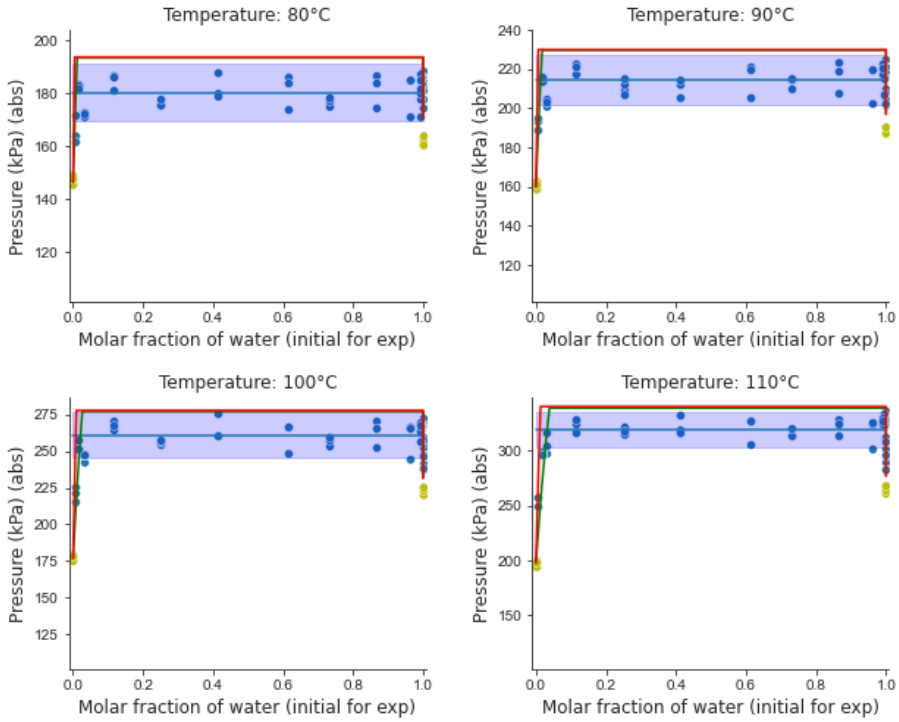


Figure 4-12. Comparison of P_{mix} for the NRTL Model with Experimental Data. Notes The *Pmix* from NRTL Aspen Plus is represented as a green horizontal line while the P_{mix} from the NRTL from Klauck [49] is represented as a red line. Parameters reprinted with permission from Klauck, M.; Grenner, A.; Schmelzer, J. Liquid-liquid(-liquid) equilibria in ternary systems of water + cyclohexylamine + aromatic hydrocarbon (toluene or propylbenzene) or aliphatic hydrocarbon (heptane or octane). *J. Chem. Eng. Data* 2006, *51*, 1043–1050, doi:10.1021/je050520f. Copyright 2021 American Chemical Society.

Figure 4-13 reports the $\Delta G_{mix}/RT$ calculated for n-octane/water blends at 80°C, with those for liquid phases represented with blue lines and those for the vapour phase with a red line. Regarding the $\Delta G_{mix}/RT$ values, one should note that the experimental P_{mix} pressures were used to calculate the vapour phase, including the uncertainty related to the 95% confidence intervals (red band) using estimates from Figure 4-9. Furthermore, the blue

bands in Figure 4-13 represents the $\Delta G_{mix}/RT$ for the liquid phases, which was calculated with the experimental temperature measurement uncertainty in the CREC VL Cell (±2°C).

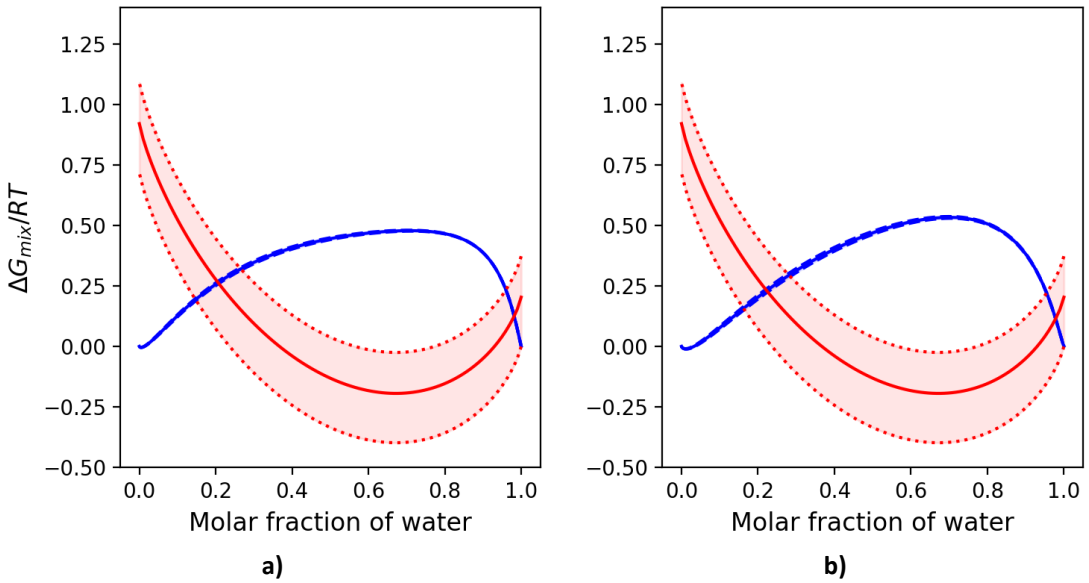


Figure 4-13. ΔGmix/RT at 80°C Using a NRTL Model and Experimental Pmix. a) NRTL Klauck et al. [49] parameters and b) NRTL with Aspen Plus V9 parameters. Notes: (i) Red bands represent the experimental ΔGmix/RT uncertainty for the vapour phase, (ii) Thick blue line includes the experimental ΔGmix/RT for the liquid phases. Parameters reprinted with permission from Klauck, M.; Grenner, A.; Schmelzer, J. Liquid-liquid(-liquid) equilibria in ternary systems of water + cyclohexylamine + aromatic hydrocarbon (toluene or propylbenzene) or aliphatic hydrocarbon (heptane or octane). J. Chem. Eng. Data 2006, 51, 1043–1050, doi:10.1021/je050520f. Copyright 2021 American Chemical Society.

Thus, and as Figure 4-13 shows, there is an important intrinsic uncertainty when the classical, three-point tangent line criteria [116] is applied to experimental data with the available models. As was reported already when discussing Figure 4-7, the data from the technical literature did not exactly match the three-point tangent criteria condition. The fact, that this condition does not precisely agree with a TPR tangent line criteria, reflects the inability of the classical stability analysis to include the experimental uncertainty, when predicting the number of phases. Thus, and to address this issue more effectively, a new Machine Learning Approach is proposed, as will be discussed in the upcoming section of this manuscript.

4.5. The Machine Learning Approach

Classification is one of the most common tasks in ML. It can be seen as converting a regression prediction problem of a target continuous variable , into a discrete function [22]. Past data (labeled items) are used to place new predictions into their respective groups or classes [22]. To establish the method reliably, standard metrics such as accuracy, true positive rate, true negative rate, precision, and a confusion matrix can be used.

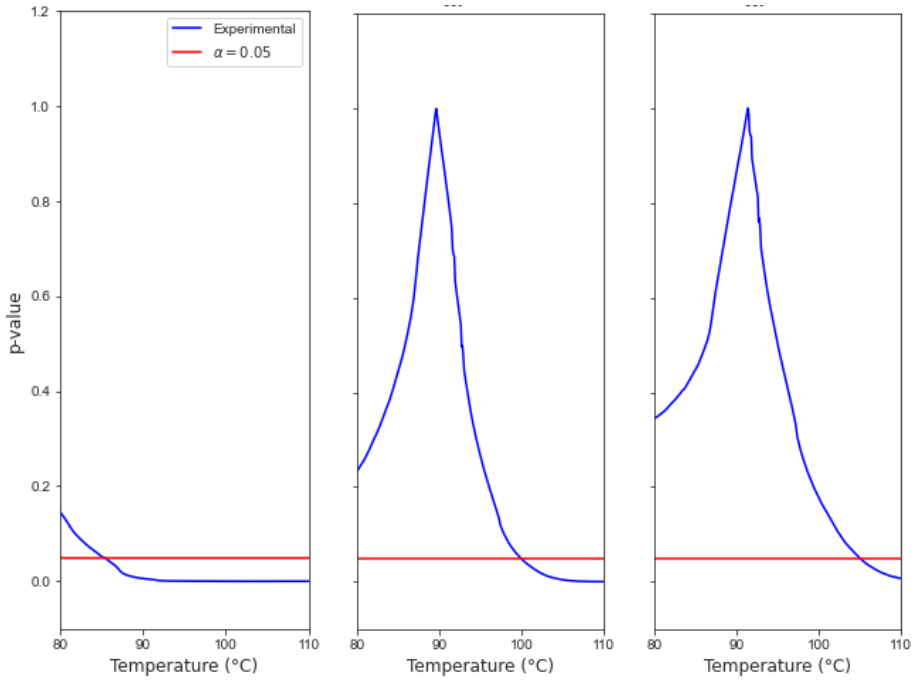
In this respect, and to predict the number of phases, a classification task is implemented in the present study, with the goal of classifying the experimental data into two different equilibrium phase regions: i) three phases (VLL) and ii) two phases (VL). One should note that the experimental data points from the CREC VL Cell are included without averaging them, with this allowing one to incorporate the typical variations of the temperature and pressure measurements within the classification task.

The first step in this classification was to determine if the mean value of the experimental measured pressures was outside the 95% confidence interval of the VLL equilibrium baseline value.

To test this hypothesis, a *t-student test* was applied. This was done using the fact that the pressure baseline for highly diluted experiments displayed a difference in some liquid fraction regions. This approach allowed us to establish that the baseline's mean was different from the experimental pressures, for highly diluted octane and highly diluted water points, with a 95% confidence interval leading to a p-value that was smaller than $\alpha = 0.05$ [120].

Figure 4-14 and Figure 4-15 describe the p-values calculated for the highly diluted mixtures at different temperatures. The red line represents the $\alpha = 0.05$ value while the blue line the p-values from experiments. When the experimental p-values were found to be higher than 0.05 $(p - value > \alpha)$, as is shown in Figure 4-14(a) for the 0.1%wt of octane in water below 85°C, the TPR assumption was considered suitable. At

temperatures above 85°C the opposite was true, with a shift occurring from 3-phases (VLL) to 2-phases (VL), with octane/water being fully soluble in each other at these conditions.

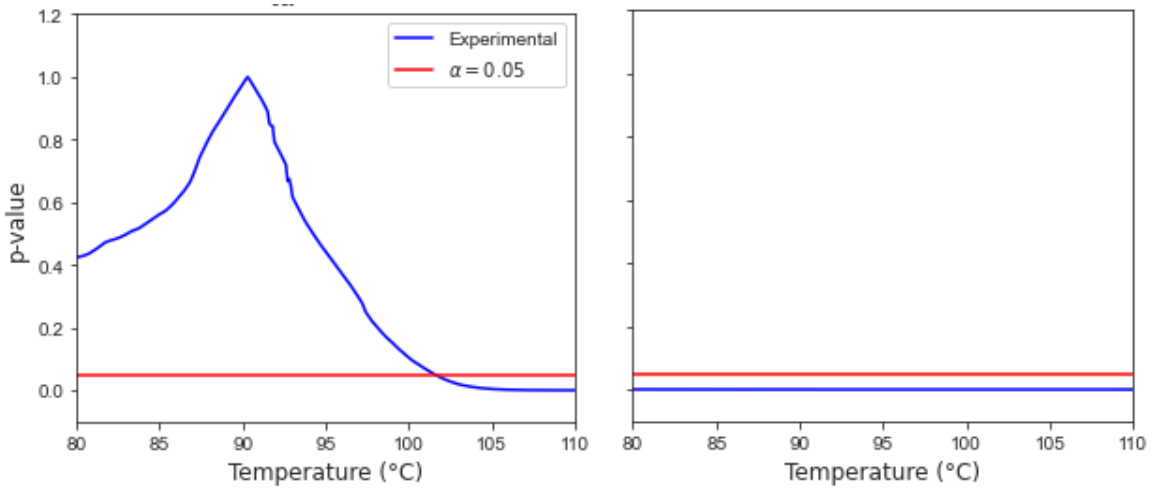


a) b) c)

Figure 4-14. t-student test for Highly Diluted Octane in Water Experiments a) 0.1%wt octane, b) 0.25%wt octane, c) 0.5%wt octane.

On the other hand, Figure 4-14(b) and Figure 4-14(c) also display the p-value for 0.25%wt and 0.5%wt of n-octane in water, with a similar transition from the TPR domain to the two-phase region, occurring at higher thermal levels of 99°C and 108°C, respectively.

Figure 4-15 considers the case of a p-value for highly diluted water in an octane blend. One can see that for 0.25%wt water in octane, there is a change from the TPR to the two-phase region at 102°C, with the 0.1%wt water in octane blend displaying complete miscibility in the entire temperature range of interest.



a) b)

Figure 4-15. t-student test for Highly Diluted Water in Octane Experiments a) 99.75%wt octane, b) 99.9%wt octane

Figure 4-16 summarizes the transition temperature for highly diluted octane in water mixtures showing a progressive increase of the transition temperature from the TPR to two-phases at initial increasing feeding separator concentrations. For instance, at concentrations in the 0.02-0.04 % molar range, the transition temperature rate seems faster than the one in the 0.04-0.08 % mol range. Demonstrating the importance of studying the phase transitions of highly diluted hydrocarbons in water.

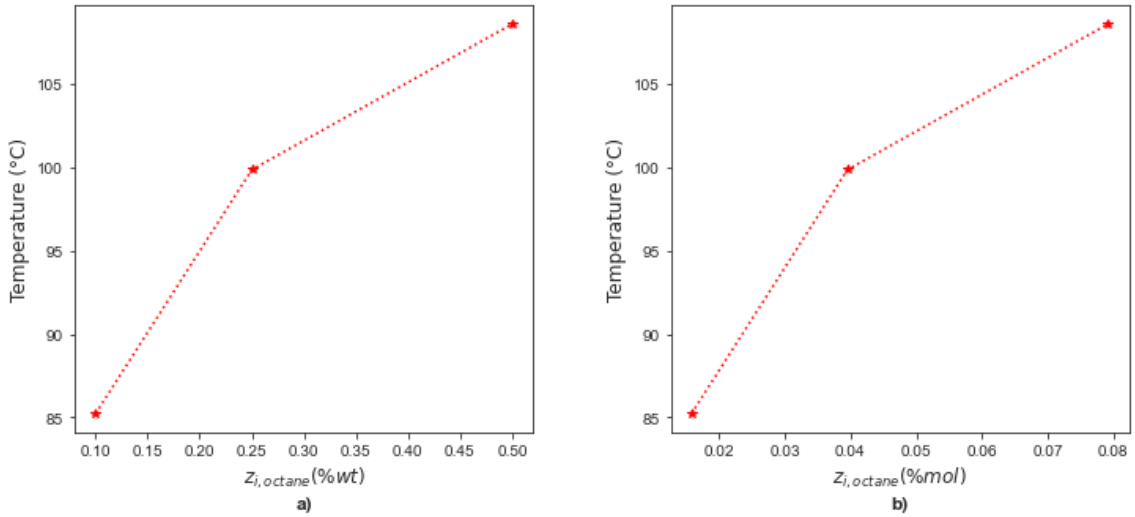


Figure 4-16. Temperature Change from the TPR to the Two-Phase Region (highly diluted octane in water).

4.5.1. Classification Methodology

Four classification methods were applied to the experimental dataset from the CREC VL Cell with the objective of establishing the value of these methods to predict the number of phases in n-octane/water mixtures. To prepare the data, an identifier label was assigned according to the *t-test*, to experimental results as two phases or three phases . The main features involved were temperature (°C), absolute pressure (kPa, including air), z_i (mol) and phase number. The first step was to apply a min-max scaler to T and P data. z_i was already in the 0 to 1 range, and no modification was required. Furthermore, and for the phases number label, the phase class was encoded as 1 for 2-Phases and 0 for 3-Phases.

In terms of the classification models, Logistic Regression, Decision Tree, K-neighbors, and Support Vector Classification from the Sklearn library in Python, were used to predict the number of phases of the experiments, available in the Temperature range of interest (80-110°C) (refer to Table 4-5 and Section 2.2). One of the main challenges of this classification problem is that it consists of an imbalanced dataset, with 4056 (approx. 23%) experimental data points for the 2-phase region and 13402 data points for 3-phase region as shown in Figure 4-17.

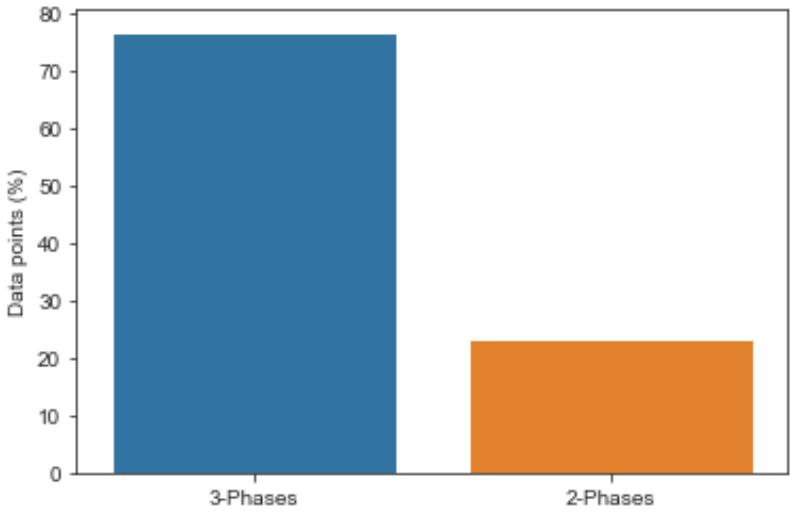


Figure 4-17. Distribution of Three Phase (liquid-liquid-vapour) and Two Phase (Liquid-vapour) Data Available.

To address the data imbalance issue, two strategies are considered: i) to undersample or downsize the 3-Phase class so that the proportion of data to train the models is the same for both phase classes, ii) to use a weighted algorithm, in the case of logistic regression and SVC, with a class weight hyper-parameter option being used. The objective of this approach is to compare the behaviour of the various models and establish which one better predicts and represents the 2-Phase region.

Table 4-5 summarizes the four models implemented with the Logistic Regression, the K-Neighbor Classifier (KNN) and the Support Vector Classifier (SVC), using default parameters [24]. Regarding the Decision Tree Classifier, a shallow tree (max depth = 3) with entropy as the classification criteria was considered to establish the split quality. The default hyper-parameters were selected as provided by the Scikit Learn library, to make of the model a predictive one and to demonstrate the applicability of the ML Classification.

Table 4-5. Classification Models Implemented.

Model #	Type	Hyper-parameters	Class Weight option
1	Logistic Regression	penalty: 12, tol: 0.0001, C: 1.0, fit_intercept: True, intercept_scaling: 1	Yes
2	Decision Tree Classifier	criterion: entropy, splitter: best, max_depth: 3, min_samples_split: 2, min_samples_leaf: 1	Yes
3	K-Neighbors Classifier	n_neighbors: 5, weights: uniform, algorithm: auto, leaf_size: 30, p: 2, metric: Minkowski	No
4	Support Vector Classifier (SVC)	C:1.0, kernel: rbf, degree: 3, gamma: scale, shrinking: True, probability: True, tol=0.001	Yes

In order to better evaluate the classification methods, 20% of the temperature data was excluded randomly from the original training dataset. This 20% of excluded data was kept aside to be included later, in the final testing dataset. After dropping these temperature data points, the remaining ones were split at 20% test data using a "train test split" function from the Sklearn library, which considers the classes ratio while performing the train splitting. Additionally, and to deal with the imbalance of the dataset, the majority class of the three phases data was randomly downsized with the idea of having two datasets with a similar class ratio.

To establish the performance of these classification models, precision, recall and F1-score were calculated as reported in the following equation.

$$\text{precision} = \frac{TP}{TP + FP} \tag{4-5}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{4-6}$$

$$\text{F}_1\text{score} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{4-7}$$

Where TP refers to a true positive, TN to a true negative, FP to a false positive and FN to false negative.

Furthermore, the Receiver Operating Characteristics (ROC) and the Areas Under the ROC Curve (AUC) were also considered in the analysis. These parameters are commonly used

in binary classifiers. ROCs plot the true positive rate (also known as Recall) versus the False Positive Rate (FPR)[21]. It is important to note that the selected models could be calibrated, with calibrated probabilities reflecting the likelihood of true events.

4.5.2. Classification Models Results

In the classification analysis developed, the more abundant class (Major Class or Class 0) was randomly downsized to match the size of the less abundant class (Minor class or Class 1). Figure 4-18 reports the resulting confusion matrix for this strategy, with the classification report being given in Table 4-6, and AUC and ROC results being shown in Figure 4-19. It can be observed that the K-Neighbors Classifier and SVC presented the best results for the 2-Phase case, which is the most valuable one in the present study. One can thus see, that the K-Neighbors Classifier and SVC can predict both 2-Phase and 3-Phase experiments with high precision, recall and F1 scores.

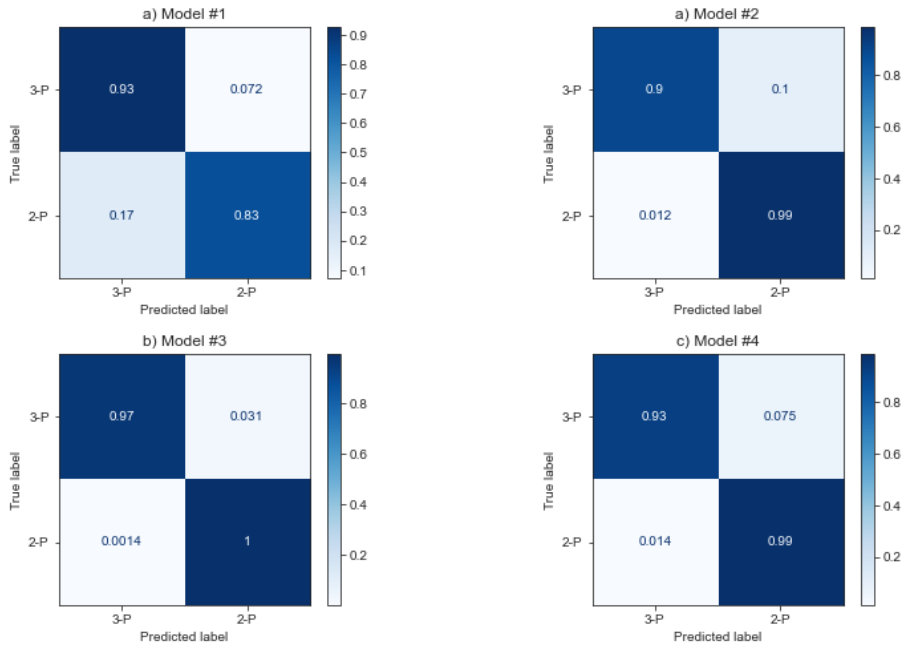


Figure 4-18. Confusion Matrix for the Tested Classification Models using Strategy 1. Note: Calculated based on test dataset.

Table 4-6. Classification Phase Report - Strategy 1. Note: Calculated based on test dataset.

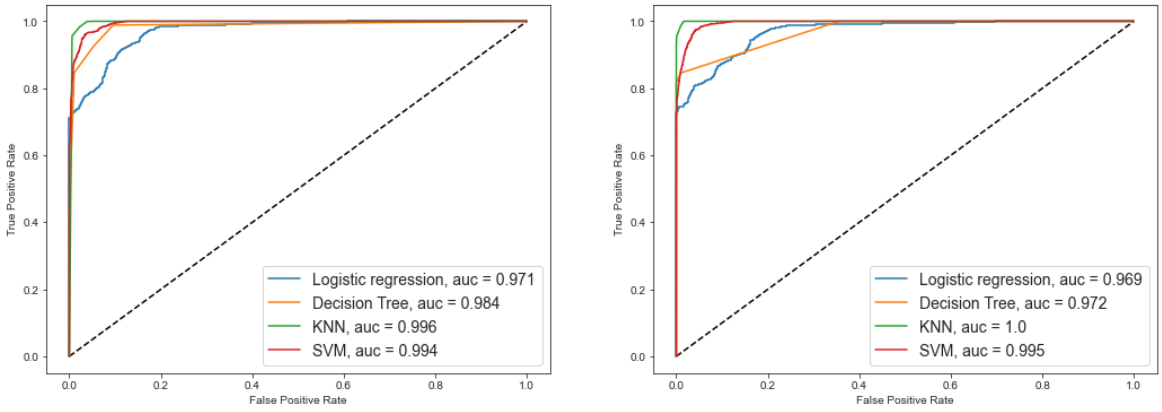
Logistic Regression			
	Precision	Recall	F1 score
3-Phases	0.92	0.93	0.94
2-Phases	0.78	0.83	0.80

Decision Tree Classifier			
	Precision	Recall	F1 score
3-Phases	1.00	0.90	0.95
2-Phases	0.75	0.99	0.85

K-Neighbors Classifier (KNN)			
	Precision	Recall	F1 score
3-Phases	1.00	0.97	0.98
2-Phases	0.91	1.00	0.95

SVC			
	Precision	Recall	F1 score
3-Phases	1.00	0.93	0.96
2-Phases	0.80	0.99	0.88

Figure 4-19(a) describes the AUC-ROC curves for the first phase classification strategy. It is possible to observe that Logistic Regression is the one with the worst performance with an AUC of 0.97, with the KNN being the best one with an AUC of 0.998.



a) **b)**

Figure 4-19. AUC and ROC Results for Strategies a) 1 and b) 2. Note: Calculated based on test dataset.

One should, however, consider that under sampling (downsized sample) one phase class can bias the posterior probabilities of the classifier [121]. To address this issue, strategy 2, which uses weighted algorithms without changing the size of the testing dataset was considered. Figure 4-20, Table 4-7 and Figure 4-19(b) report the results for the confusion matrix, classification report and AUC and ROC results using Strategy 2. One can notice an improvement using the weighted algorithms; without under sampling, the KNN and weighted SVC results were able to improve for the 2-Phase predictions, reducing the number of false positives and false negatives.

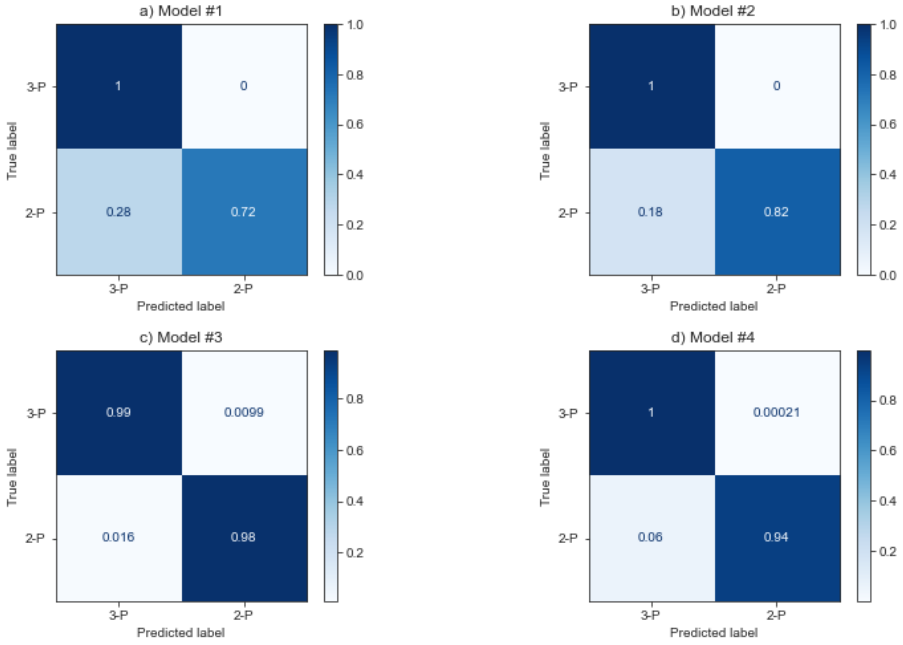


Figure 4-20. Confusion Matrix for Strategy 2. Note: Calculated based on test dataset.

Table 4-7. Classification Reports for Strategy 2. Note: Calculated based on test dataset.

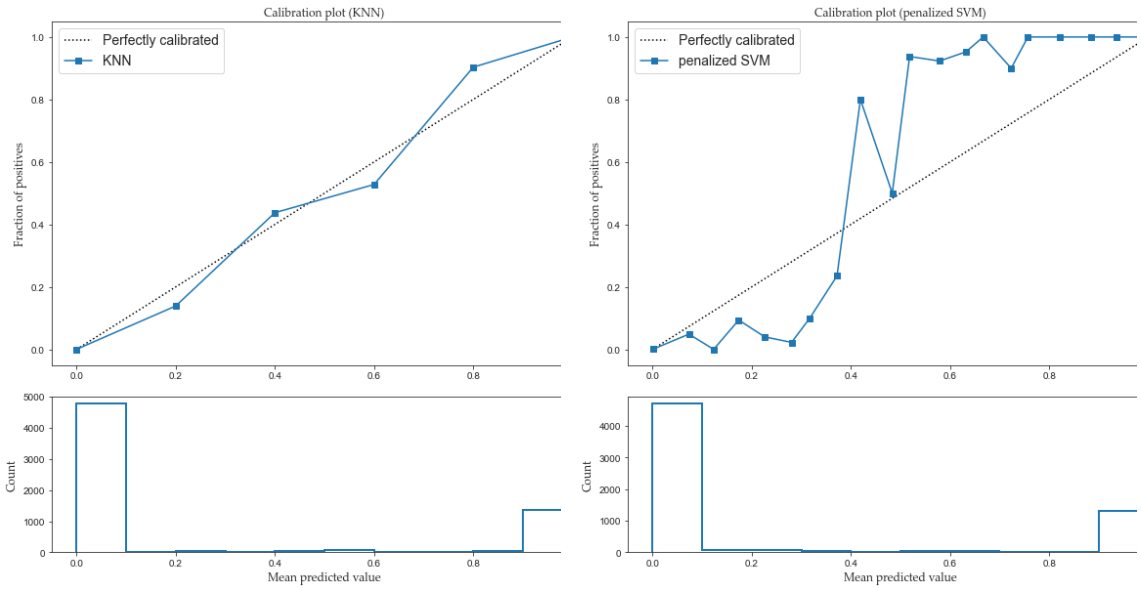
Logistic Regression (penalized)			
	Precision	Recall	F1 score
3-Phases	0.92	1	0.96
2-Phases	1	0.72	0.84

Decision Tree Classifier (penalized)			
	Precision	Recall	F1 score
3-Phases	0.95	1.00	0.97
2-Phases	1.00	0.82	0.90

K-Neighbors Classifier (KNN)			
	Precision	Recall	F1 score
3-Phases	1.00	0.99	0.99
2-Phases	0.97	0.98	0.98

SVC (penalized)			
	Precision	Recall	F1 score
3-Phases	0.98	1	0.99
2-Phases	1	0.94	0.97

Given the promise of obtaining ML results for phase classification, the calibration plot for the KNN model and the weighted SVC, which represent the best models, were further validated as reported in Figure 4-21(a) and Figure 4-21(b). As a result, one can conclude that the ML model was well calibrated with the predicted probabilities corresponding closely to the expected distribution of probabilities for each class.



a) **b)**

Figure 4-21. Calibration for a) KNN and b) SVC Models. Note: Calculated based on test dataset.

As well, one can notice that the KNN classifier shows near-perfect calibration, with this being an improvement over the weighted SVM classifier. In addition, the KNN model presents the better results overall, and is selected for our future work. Thus, it is shown in the present study, that Machine Learning provides a valuable tool to accurately discriminate between 2-Phase or 3-Phase equilibrium regions. This prediction is critical while implementing phase equilibrium calculations, where the identification of the number of phases is a critical starting point for the flash calculations. This is achieved using to train the ML classifiers the abundant CREC VL Cell experimental data, instead of available thermodynamic models or simulation software, securing the good quality of the data considered and the adequate successful application of ML techniques.

4.6. Conclusions

a) It is shown that reliable models, based on fundamentals principles, are still needed to represent the number of phases, in diluted hydrocarbon in water mixtures at phase equilibria.

b) It is proven that a phase stability analysis involving the Gibbs energy of mixing, can be used to explain calculation result discrepancies, in water/n-octane mixtures when using available simulation software.

c) It is demonstrated that runs in a CREC VL Cell employing a dynamic technique (1.22°/min temperature ramp), can provide the "big data sets" required to accurately determine the fully miscible, partially miscible, and fully immiscible octane/water blend states.

d) It is proven that ML models based on the obtained "big data sets" can be proposed for the prediction of the number of phases under the studied conditions, with the KNN model and the weighted SVC model, identified as the ones with best performance.

5. A Machine Learning Approach for Vapour-Liquid-Liquid and Vapour-Liquid Equilibrium of Paraffinic Aromatic Synthetic Naphtha/ Water Systems: Prediction of Phase Number

Canada has produced synthetic crude from Alberta oil sands for about 40 years and has become the world leader in oil sands production [1,3]. The crude obtained from the oil sands is called bitumen. The challenge for bitumen and heavy oil processing is to maximize project economics through the optimization of each process step involved [2]. When compared to conventional crudes, one can note that the processing of oil sands is energy intensive.

The NRU consists of a vacuum stripping tower with steam being injected at the bottom of a deck. This unit recovers naphtha from the bitumen process tailings for reuse, while the remaining fraction being discharged to the tailings pond [4]. This process is energy-intensive, and environmental guidelines for naphtha recovery must be met [4].

The simulation of the NRU requires knowledge of the unit thermodynamics involved. One of the main issues to clarify in naphtha-water systems is the transition from the three-phase VapourVapour-Liquid-Liquid (VLL) equilibrium to the two-phase VapourVapour-Liquid (VL) equilibrium. In this respect, this work's main objective is to develop an improved methodology, to predict the number of phases in naphtha/water systems, under the NRU operation conditions.

Traditionally, the Peng-Robinson (PR) Equation of State (EoS) is one of the most popular to predict hydrocarbon-based PVT behaviourbehaviour, including vapour pressures [99]. When using simulation software such as HYSYS V9 or Aspen Plus V9, it is considered one of the most enhanced models with an extensive binary interaction parameter database. However, the PR EoS displays limitations when the considered blends include water or aqueous hydrocarbon mixtures [100]. In this sense, as suggested by previous research from our group [94,95], the PR EoS does not describe the system under study well, and a

different EoS should be used. In the case of binary systems, such as n-octane/water mixtures, an activity coefficient model can be used, as we proposed in our previous work [5]. However, the limit of classical activity coefficient models is low operation pressures (≤10 bar) and no presence of C_{7+} in the components. Therefore, for water and heavy hydrocarbon blends such as naphtha or bitumen (C_{7+}) in water streams, as in the case of this study, a Cubic Equation of State is strongly suggested. In this work, the Soave-Redlich-Kwong (SRK) EoS with a Kabadi-Danner [101] modification is used, given that it can improve the VLLE calculations for water-hydrocarbon systems, particularly in dilute regions, as shown in the present study.

For the experimental studies, an paraffinic-aromatic synthetic naphtha (PASN) was used. As reported in Kong's Master Thesis [94], the PASN was prepared using 5 paraffinic compounds, and toluene. The True Boiling Point (TBP) Curve of the PASN used for this work, was similar to that of a typical naphtha [94]. In this work, analyses of (a) the phase stability using Soave-Redlich-Kwong Equation of State with a Kabadi-Danner modification (SRKKD EoS), (b) the experimental data reported by Kong (2020) [94] for this mixture, and (c) a machine learning approach used to predict the number of phases in PASN/water systems, are presented. The ML methodology developed in our previous work [5], for n-octane/water systems, was adapted for PASN/water mixtures.

5.1. Materials and Methods

5.1.1. Materials

Distilled water was used in all the experimental studies. As proposed by Kong's Master Thesis (2020) [94], typical naphtha can be represented by a mixture of paraffins consisting of 5 paraffinic hydrocarbons (in the 6 to 12 carbon range) and toluene (Table 5.1). The composition of the blend was selected to represent a typical naphtha, as reported in the technical literature [122]. The aromatic synthetic naphtha (PASN) presented a similar distillation curve to the one in the literature, as demonstrated by Kong (2020) [94].

Table 5-1. Aromatic Synthetic Naphtha (PASN) Composition.

Component	Brand	Purity (%)	Water content (%)	Composition (wt%)	Composition (mole%)
n-hexane	Sigma-Aldrich	>97	0.01	9.5%	11.94%
n-heptane	Sigma-Aldrich	>96	0.02	20.9%	22.59%
n-octane	Sigma-Aldrich	>99	0.00	57.0%	54.05%
n-decane	Sigma-Aldrich	>99	0.01	5.7%	4.34%
n-dodecane	Sigma-Aldrich	>99	0.00	1.9%	1.21%
Toluene	Fisher Scientific	>99	0.008	5.0%	5.88%

5.1.2. HYSYS V9 simulations

The behaviour of water/PASN mixtures such as the one described in Table 5.1 can be simulated using HYSYS V9. For this, a 3-phase separator was specified. A 100 kgmol/h stream of water was mixed with a PASN stream getting a blend with a 50%mol water / 50%mol PASN mixture. This blend was fed into the unit, with temperatures in the 80-110 °C range and a vapour fraction of 0.

It is important to notice that the expected percentual molar fraction difference related to the PASN is much lower than 0.1%, when the feed and the hydrocarbon phase are compared in 80-110°C range as reported in Table 5.2 and Table 5.3. In this way, the change on initial relative composition of hydrocarbons within the PASN is expected to be negligible.

Table 5-2.Predicted HYSYS V9 Molar Fraction Difference in VLLE between feed and outlet for the Hydrocarbon Phase at 80°C. 50wt% PASN in water

	Feed (H_20) (mole%)	**Feed (PASN) (mole%)**	**Outlet Hydrocarbon Phase (mole%)**	**Outlet Aqueous Phase (mole%)**	**Hydrocarbon Phase (mole%, base free of water)**	**Hydrocarbon Phase (% diff)**
Water	100	0.0%	6.27E-03	0.99995061		
n-Hexane		11.9%	11.87%	3.26E-06	11.94%	0.002%
n-Heptane		22.6%	22.45%	1.44E-06	22.59%	0.004%
n-Octane		54.0%	53.71%	6.53E-07	54.05%	0.005%
n-Decane		4.3%	4.31E-02	5.38E-10	4.34%	0.005%
Toluene		5.9%	5.84E-02	4.40E-05	5.87%	0.070%

n-Dodecane	1.2%	1.20E-02	8.95E-13	1.21%	0.005%

Table 5-3. Predicted HYSYS V9 Molar Fraction Difference in VLLE between feed and outlet for the Hydrocarbon Phase at 110°C. 50wt% PASN in water

	Feed (H_2O) (mole%)	**Feed (PASN) (mole%)**	**Hydrocarbon Phase (mole%)**	**Aqueous Phase (mole%)**	**Hydrocarbon Phase (mole%, base free of water)**	**Hydrocarbon Phase (% diff)**
Water	100	0.0%	$1.64*10^{-2}$	0.999932		
n-Hexane		11.9%	11.74%	$4.73*10^{-6}$	11.94%	0.006%
n-Heptane		22.6%	22.22%	$2.25*10^{-6}$	22.59%	0.004%
n-Octane		54.0%	53.16%	$1.12*10^{-6}$	54.05%	0.009%
n-Decane		4.3%	$4.27*10^{-2}$	$1.37*10^{-9}$	4.34%	0.011%
Toluene		5.9%	$5.78*10^{-2}$	$6.02*10^{-5}$	5.87%	0.094%
n-Dodecane		1.2%	$1.19*10^{-2}$	$3.62*10^{-12}$	1.21%	0.012%

5.2. Phase Stability Testing

The NRU can be represented as a flash unit. One problem when dealing with flash calculations is that the number of phases in phase equilibrium calculations is not know a priori. To address this issue, there are two main approaches [10]: (i) to assume a number of phases and proceed to phase split calculations, (ii) to perform a phases stability test. This work focuses on the second approach.

The stability analysis for binary systems can be extended to multicomponent mixtures, by minimizing the Gibbs Free energy of mixing, as reported in our previous work [5]. In the case of a multicomponent mixture, the ΔG_{mix} can be defined by a surface instead of a curve, and the tangent line expressed as a tangent plane [13]. This Tangent Plane Distance (TPD) approach was originally proposed by Baker et al. (1982) [118].

The Tangent Plane Distance (TPD) represents the separation between two Gibbs free energies, at constant temperature and pressure [123]. TPD analysis for a multicomponent fluid can be used to determine the stability of a phase, and is related to the chemical potential and fugacity coefficients as follows [124]:

$$\overline{TPD} = \sum_{i=1}^{c} y_i[\mu_i(\boldsymbol{y}) - \mu_i(\boldsymbol{z})] \tag{5-1}$$

$$TPD = \frac{\overline{TPD}}{RT} = \sum_{i=1}^{c} y_i[\ln \phi_i(\boldsymbol{y}) + \ln y_i - \ln \phi_i(\boldsymbol{z}) - \ln z_i] \tag{5-2}$$

Where $\boldsymbol{y} = y_1, y_2, \dots, y_N$ refers to the trial phase (liquid or vapour) and $\boldsymbol{z} = z_1, z_2, \dots, z_N$ refers to the testing phase (feed) [125]. The multicomponent blend is considered to be stable if the TPD is positive or zero, for any trial phase composition of $\boldsymbol{y}$ [124].

$$\begin{cases} TPD \geq 0 \ \ \forall\, y = (y_1, y_2, \dots, y_N) \in \Re \\ \sum_{i=1}^{N} y_i = 1 \end{cases} \tag{5-3}$$

Regarding the stability analysis, it is expected that if all the stationary points of the TPD function are calculated, then a ΔG_{mix} global minimum can be established [7]. However, in practice, it is not easy to find all the stationary points, as the ΔG_{mix} surface is non-convex with a local minima, with the calculation methods converging towards local minima [7]. The stationary condition is related to the derivative of Equation (5.1) as explained elsewhere [7], and is given by Equation(5.4):

$$\ln \phi_i(\boldsymbol{y}) + \ln y_i - \ln \phi_i(\boldsymbol{z}) - \ln z_i = 0 \tag{5-4}$$

In this respect, Equation (5.4) can be considered as proposed by Michelsen as an unconstrained local minimization using a variable change [80]: $Y_i = y_i e^{\mu_i(\boldsymbol{z}) - \mu_i(\boldsymbol{y})}$ with $y_i = \frac{Y_i}{\sum_{i=1}^{N} Y_i}$. Thus, the resulting stationary condition (zero derivative) is given by:

$$\ln \phi_i(\boldsymbol{Y_i}) + \ln Y_i - \ln \phi_i(\boldsymbol{z}) - \ln z_i = 0 \tag{5-5}$$

One should note that the stability condition to be satisfied by all the stationary points, given that $TPD \geq 0$, is equivalent to $\sum_{i=1}^{N} Y_i = 1$.

Thus and given the above, an equivalent equation can be obtained,

$$TPD^* = \sum_{i=1}^{c} Y_i[\ln \phi_i(\boldsymbol{Y}) + \ln Y_i - \ln \phi_i(\boldsymbol{z}) - \ln z_i] \tag{5-6}$$

With $Y_i > 0$ and the stationary points with the same sign, assessed with TPD (Equation (5.2)) and TPD* (Equation (5.6)).

5.3. Results and Discussion

5.3.1. Validation of the SRKKD EoS Python Model with N-Octane /Water Mixtures

As explained in Section 5.1.2, HYSYS simulations were completed using a 3-Phases separator, while in the case of the simulations carried out with Aspen Plus V9, a binary analysis was performed. In both cases, an air correction was applied.

Figure 5-1 reports VLLE results for water/n-octane systems using SRKKD EoS in HYSYS V9, Aspen Plus and the model developed in Python for this work. Results are compared with experimental data reported previously by our research group [5,94,126]. One can notice when comparing the HYSYS V9, Aspen Plus V9 and Python results with Experimental data at 80°C, that the pressure differences at the Three Phase Region (TPR) are 6.91%, 7.0% and 4.2%, respectively. It can be observed that the result reported by Aspen Plus V9 and HYSYS V9 are close to the pressure calculated with the immiscible assumption. As presented in our previous work [5], this assumption is however, incorrect for water/n-octane mixtures. As well, one is not able to reproduce the results from HYSYS V9 and Aspen Plus V9 using Python. Simulation software calculations, a black-box model indeed. Thus, the Python model is adequate as a *White Box* methodology, given that it is far from the immiscibility assumption, and it provides results within the 95% confidence interval provided by the blue bands as shown in Figure 5-1. Thus, as a result of this the Python-SRKKD model was considered the most adequate for the upcoming phase stability tests.

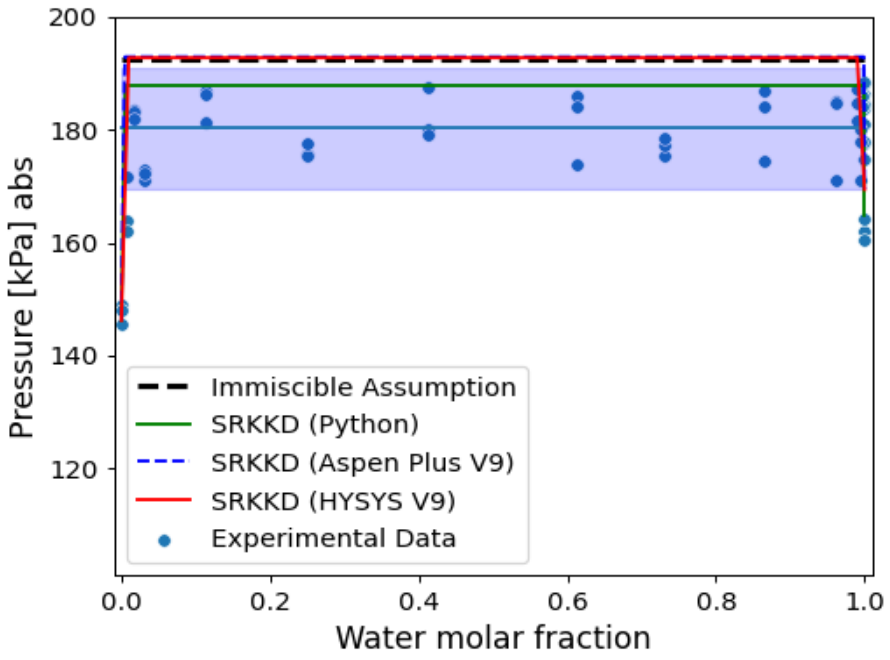


Figure 5-1. Comparison of the SRKKD EoS in Python with HYSYS V9 and Aspen Plus V9 Simulations, and experimental data for water/n-octane mixtures at 80°C.

5.4. Phase Stability Test

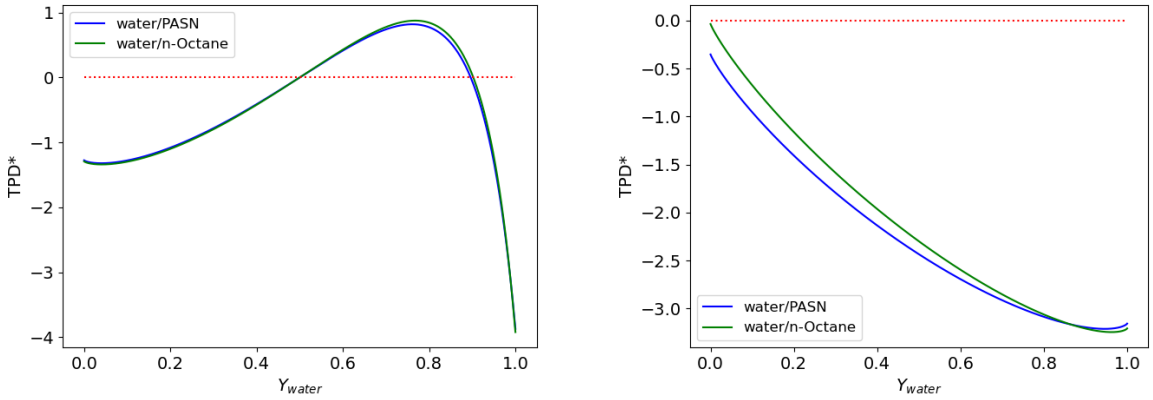
As stated previously, to perform flash calculations, it is important to be able to predict the number of phases at given operating conditions. In Chapter 4 , we used a Gibbs free energy minimization for the binary water/n-Octane mixture [5], and this to analyze the number of phases. In the case of multicomponent mixtures, as it is the case of PASN, the use of the *TPD** function, as explained in Section 5.2, was suggested instead.

Furthermore and when considering the Tangent Plane Distance (*TPD*)* calculations as can be obtained using Equation 5.6, there is a similarity between the calculated results for water/n-octane and those for water/PASN, when one considers the PASN relative compositions, which do not change within the hydrocarbon phase (pseudo-binary approach).

Figure 5-2 to Figure 5-4 report *TPD** results for the case of a mixture at 80°C, and with a P = 84.94 kPa (predicted TPR pressure for a 50 mole% water/PASN mixture as given by Python calculations). One can observe that for a water molar fraction of 0.5 when

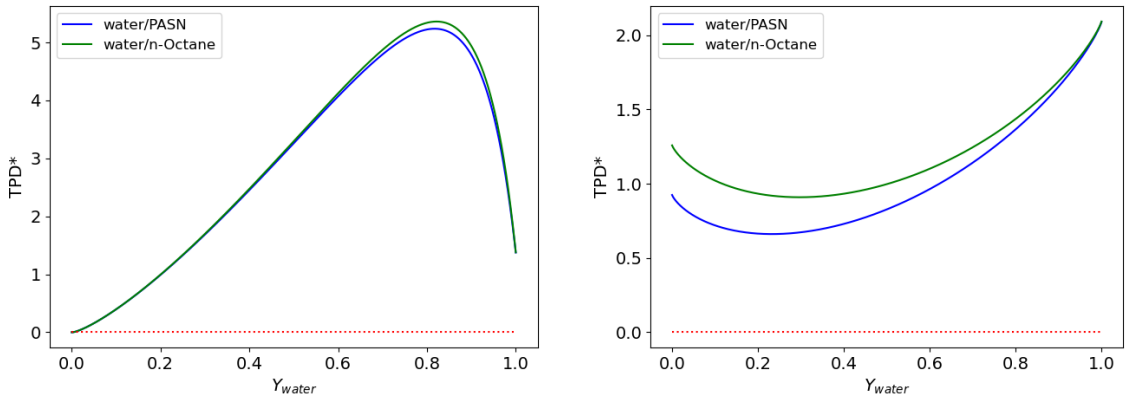
considering the liquid phase (Figure 5-2a) or the vapour phase (Figure 5-2b), there are values of the TPD* that are located below the zero dotted line, this means that 2 or more phases may co-exist.

Furthermore, in the case of Figure 5-3, for a water molar fraction of 10^{-3}, all values of TPD* function are higher than 0, which means only one liquid phase is present (complete solubility). Finally, for Figure 5-4 with a molar fraction of $1\text{-}10^{-7}$, the negative values of TPD* show the presence of two liquid phases.



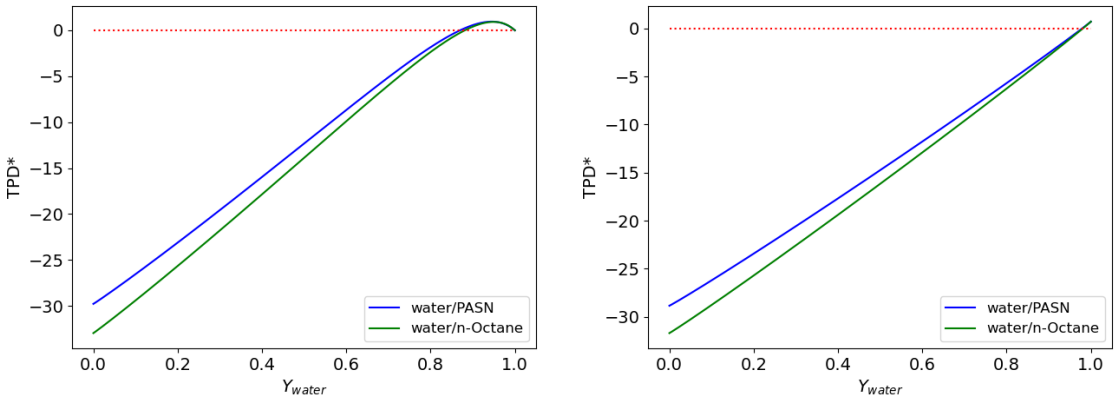
(a) (b)

Figure 5-2. TPD* at 80°C, z_w = 0.50 and 87.45 kPa – Trial Phase is a) Liquid b) vapour. Note: Dotted Red Line shows TPD* zero value



(a) (b)

Figure 5-3. TPD* at 80°C, z_w = 10^{-3} and 87.45 kPa – Trial Phase is a) Liquid b) vapour. Note: Dotted Red Line shows TPD* zero value

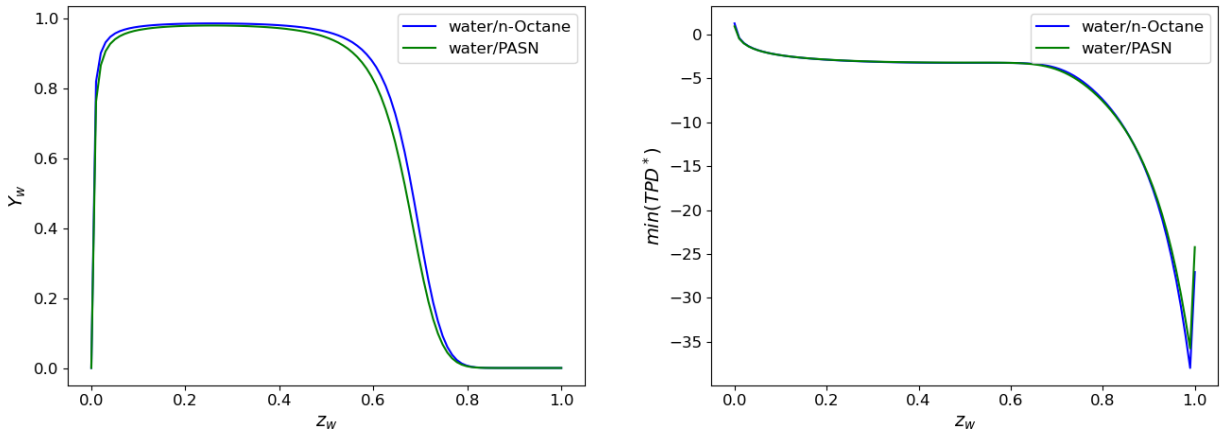


(a) (b)

Figure 5-4. TPD* at 80°C, $z_{hydrocarbon} = 10^{-7}$ and 87.45 kPa – Trial Phase is a) Liquid b) vapour.
Note: Dotted Red Line shows TPD* zero value

Then, by representing the PASN to be a pseudo-binary system, the TPD* function can be minimized using a Sequential Least Squares Programming (SLSQP) Method, available in the SciPy package from Python, and with the constraint of $Y_i > 0$.

Figure 5-5a reports the value of Y_w for which TPD* is minimum, while Figure 5-5b presents the minimum values of TPD*. Thus, for very low concentrations of water, it is possible to notice minimum TPD* values, that are higher than zero, meaning one liquid phase is present only. One should notice that for low hydrocarbon values, this behaviour was not observed, with two liquid phases being found in these cases.



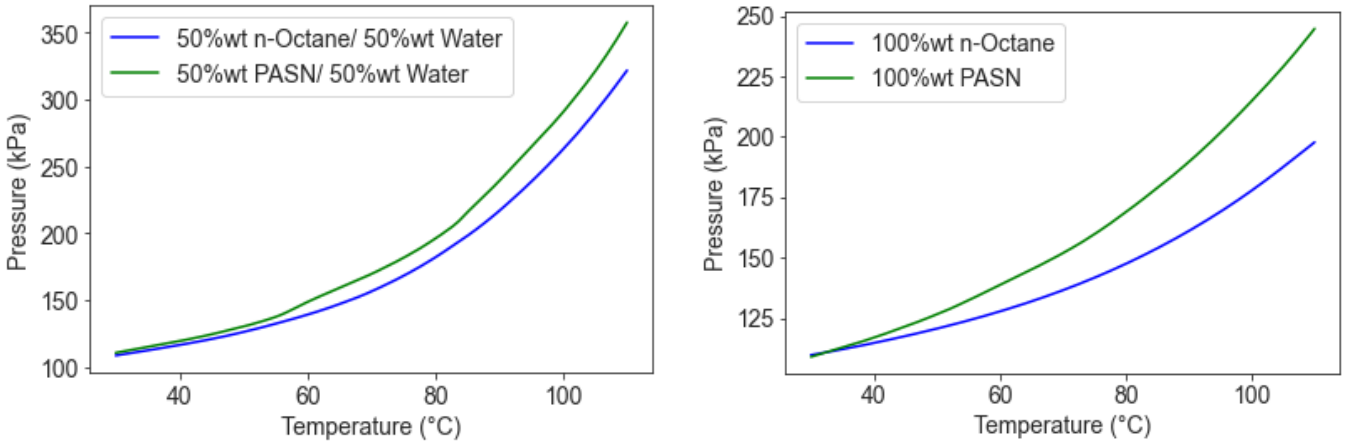
(a) (b)

Figure 5-5. Minimization of TPD* Function at 80°C and 87.45 kPa.

However, when considering multicomponent mixtures, the Y_i individual molar fractions should be varied, and the TPD should, on this basis, be calculated. For instance, in the case of the PASN in water blends of this study, this means changing the water composition, as well as the composition of 5 out of the 6 hydrocarbons present in the blend. This presents a significant computational challenge, in terms of computational time and convergence towards a global minimum. Given the above, in the upcoming sections, a Machine Learning alternate approach is proposed to identify the number of phases present in the PASN/water mixtures.

5.4.1. Analysis of Experimental Results

The obtained experimental results for water/n-octane and water/PASN mixtures were first compared as shown in Figure 5-6. This figure reports the average vapour pressure, as measured in the CREC VL Cell, for 50/50 %wt mixtures and for pure components (results include air). It can be noticed that for the 50/50 %wt mixtures, the maximum difference in pressure is 10%, while for pure octane and PASN components it is 19.2%. Thus, n-octane can be considered a good pseudo component to represent as a first approximation naphtha/water systems, as stated previously by our group [5,94].



a) b)

Figure 5-6. Comparison of Experimental Results for (a) 50%wt PASN/ 50%wt Water and 50%wt Octane/ 50%wt Water and b) 100%wt PASN and 100%wt Octane. Note: Results include air.

Figure 5-7 reports experimental data for water/n-octane and water/PASN obtained in the CREC VL Cell, as described in Section 2.2. In this figure, the “blue” and “green” horizontal lines represent a baseline for the VLL equilibrium at a given temperature, for water/n-octane and water/PASN mixtures. The blue and green bands provide the 95% confidence interval for the pressure baseline. In Figure 5-7, one can notice while comparing the VLLE pressure of water/n-octane and water/PASN mixtures, that the difference between PASN them is 7.9%, 8.9%, 9.4% and 9.9% at 80°C, 90°C, 100°C, 110°C, respectively. The observed behaviour is similar to the one described in our previous work [5], for the VLL and VL equilibrium domains.

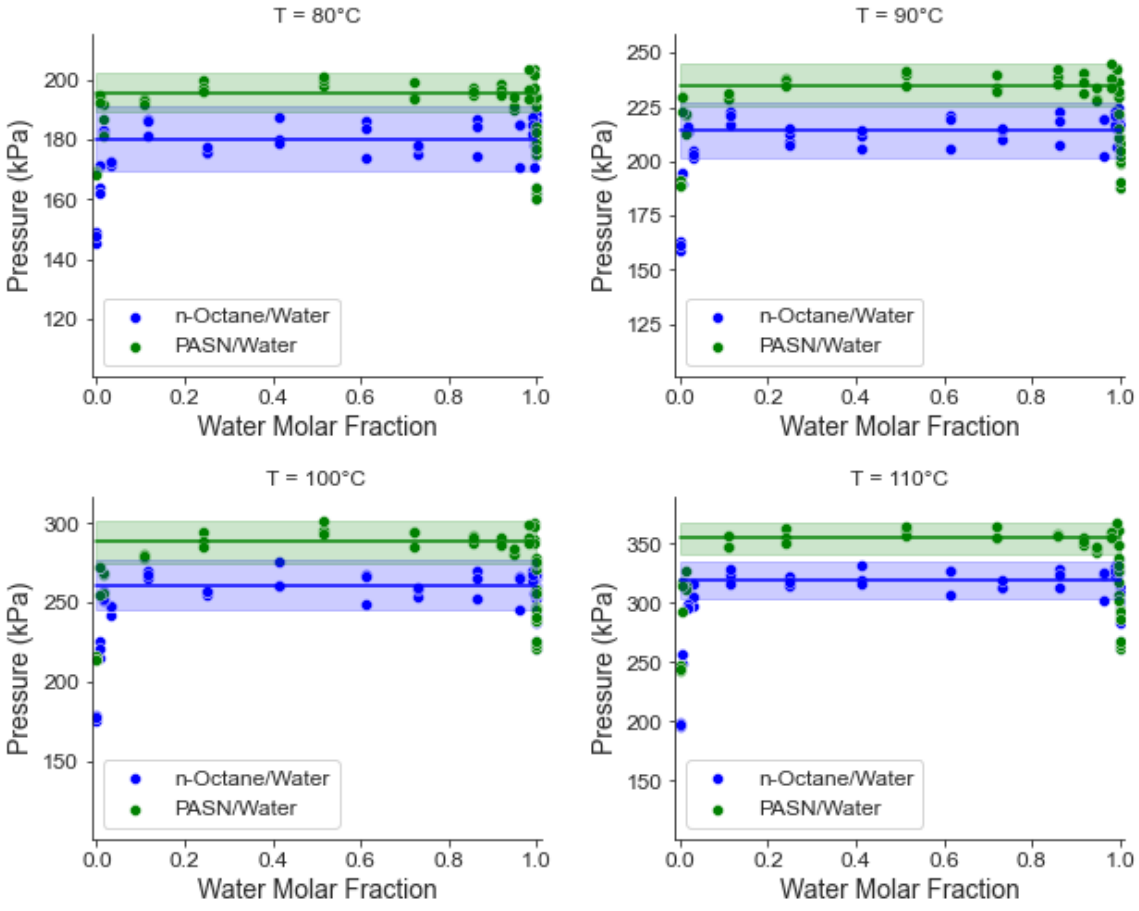


Figure 5-7. Comparison of Experimental Results with PASN/Water and Octane/Water Mixtures. Note: Experimental data for n-octane/water systems are from our previous work [5].

As well, and while reviewing Python results (including air correction), used to predict the pressure at VLLE (Figure 5-8), similar trends were observed: a close to a trapezoidal change of the total pressure with hydrocarbon molar fractions. Air was included in the calculations consistent with air presence in the CREC VL Cell. The observed difference of the CREC VL Cell data for n-octane/water and PASN/water mixtures at the TPR was of 9.87%, 10.69%, 11.17% and 11.29% at 80°C, 90°C, 100°C, 110°C, respectively.

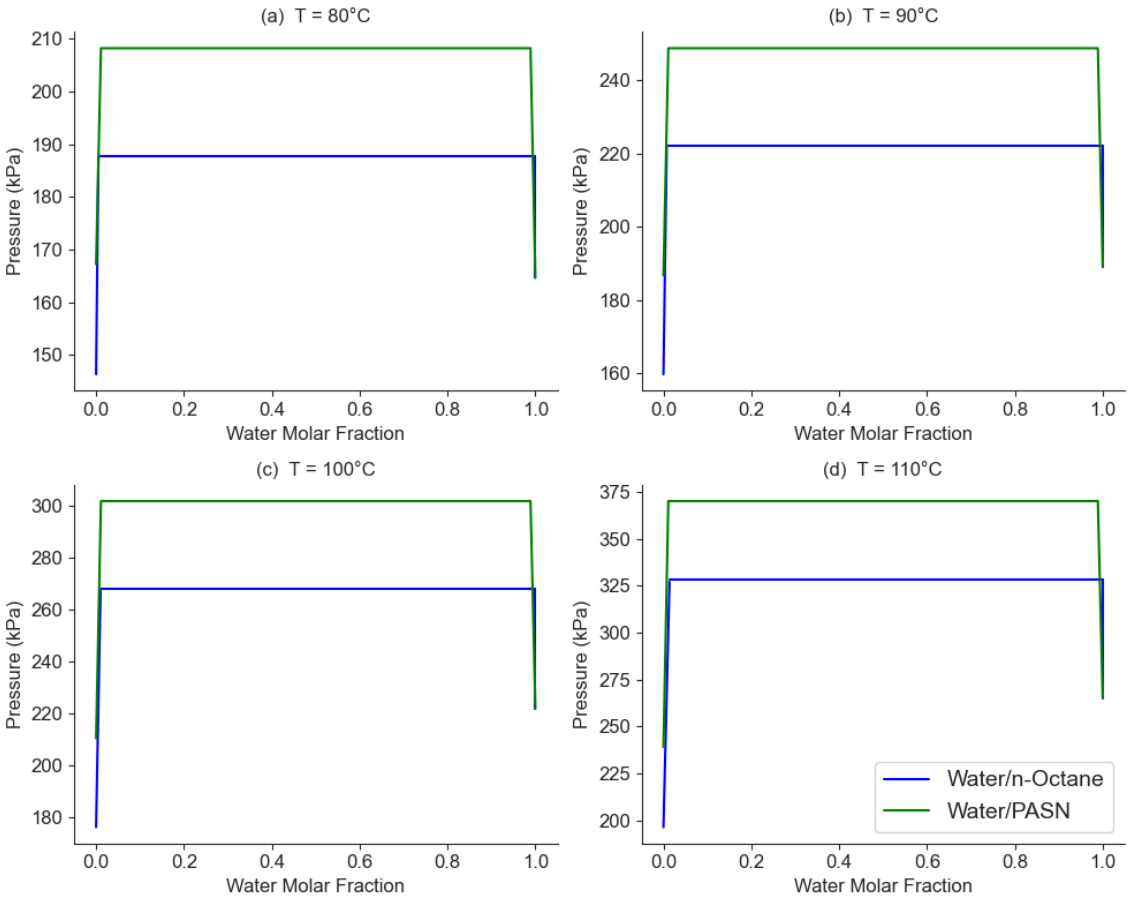


Figure 5-8. HYSYS V9 Results for Octane/Water and PASN/Water Blends.

In this respect, one can remark that traditionally, hydrocarbon/water mixtures were assumed to be completely immiscible species and this to reduce the computational effort. However, as reported in Figure 5-9, the immiscible hydrocarbon assumption (red line in Figure 5-9) does not represent PASN/water mixtures properly. The immiscible

hydrocarbon assumption overestimates in fact, the total pressure, at the three-phase region where two liquid phases and one vapour phase are present. As well, this model assumes complete two liquid phases immiscibility for both highly diluted octane in water and water in octane blends.

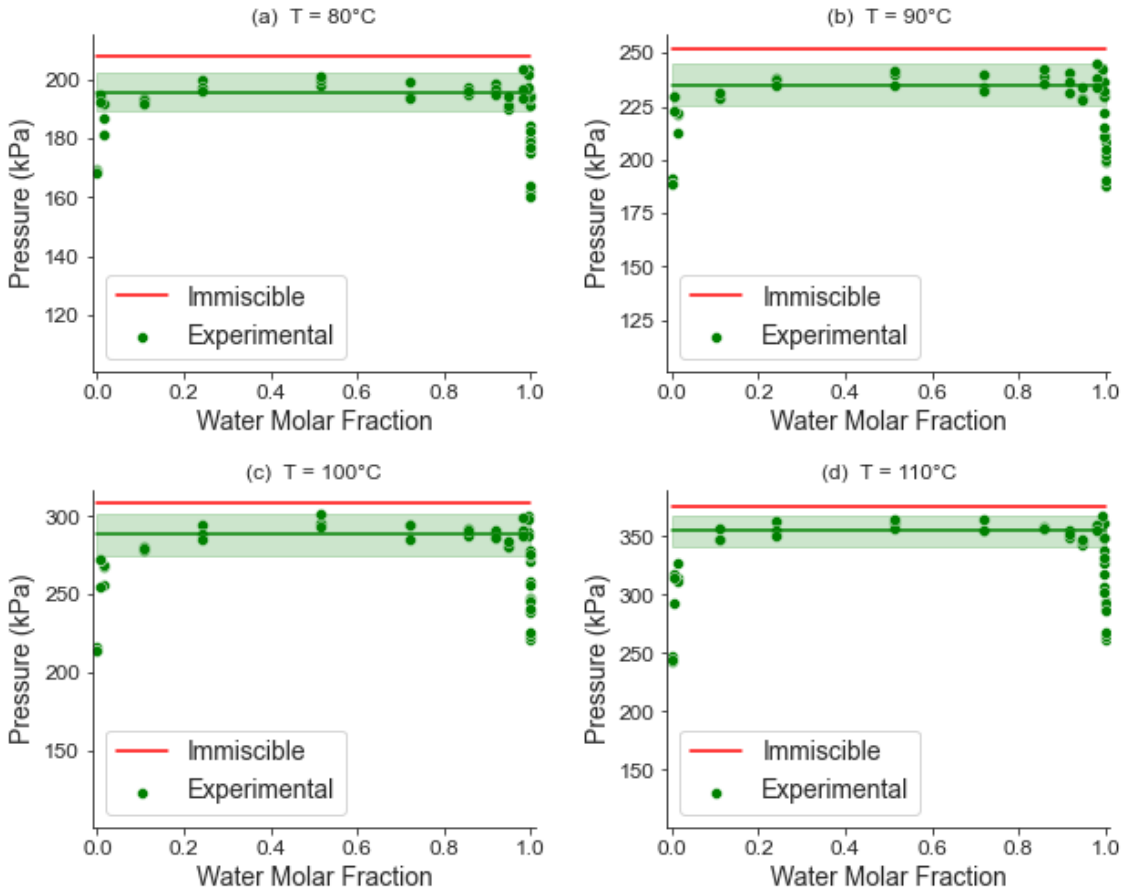


Figure 5-9. Pressure for Experiments in the CREC-VL Cell versus P_{mix} from Two Immiscible Liquid Phase Model for PASN/Water Mixtures.

Furthermore, when comparing PASN/water (Figure 5-10) experimental results with the Python predictions, it is possible to observe a similar overestimated total pressure at the three-phase region.

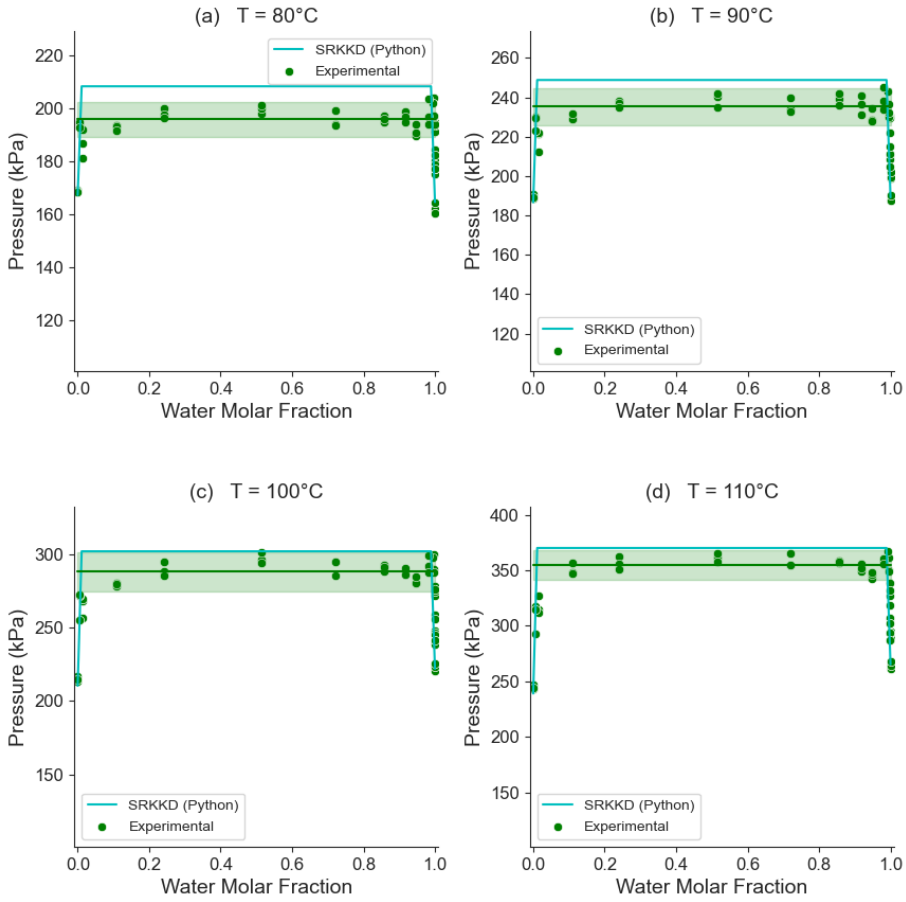


Figure 5-10. Pressure from Experimental Results in the CREC-VL Cell versus SRKKD (Python) Results for PASN/Water Mixtures.

As well, if one considers in more detail the highly diluted regions of water in PASN (Figure 5-11 and Figure 5-12), it is possible to notice significant pressure differences, as stated in our previous work [5]. This behaviour is characteristic of the two-phase region, where the two liquids are highly miscible. As in the case of the n-octane/water system [5], for the PASN/water mixtures, the horizontal baseline indicates the presence of VLLE. On the other hand, the lower than the horizontal baseline pressures for highly diluted samples show two-phase region conditions. Thus, it is also possible to observe that the SRKKD equation does not accurately represent the thermodynamic behaviour in these highly diluted blends of either aqueous or hydrocarbon liquid phases.

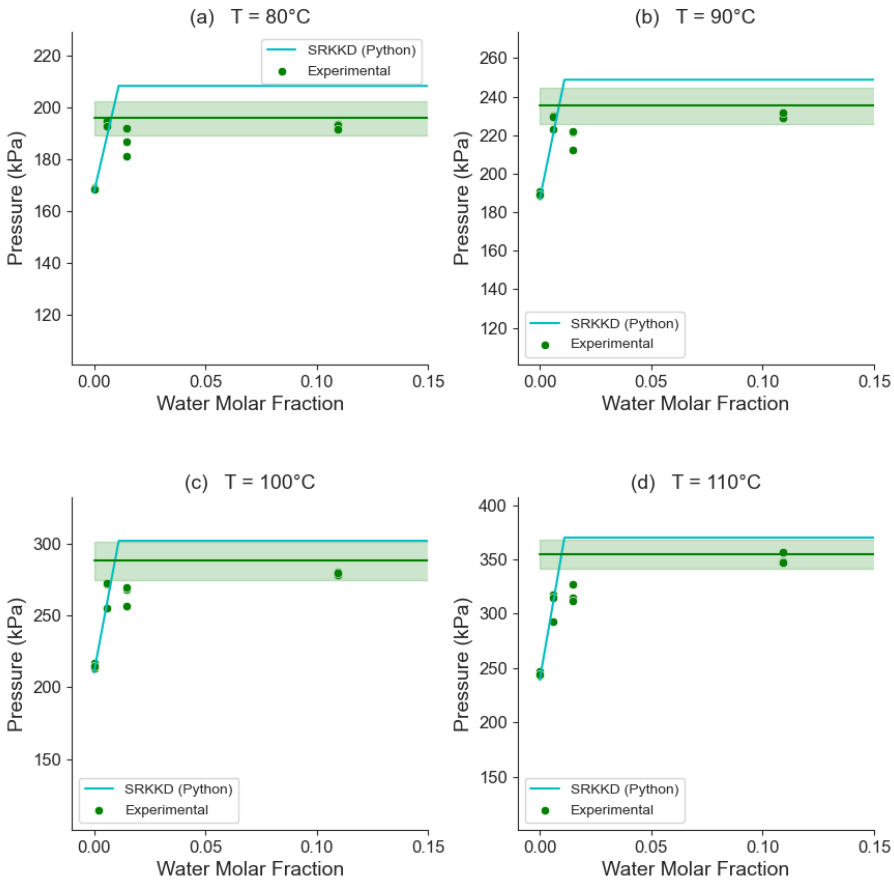


Figure 5-11. Expanded View for the Pressure for Highly Diluted Water in the PASNPASN Region using Data from the CREC-VL Cell and SRKKD (Python) results.

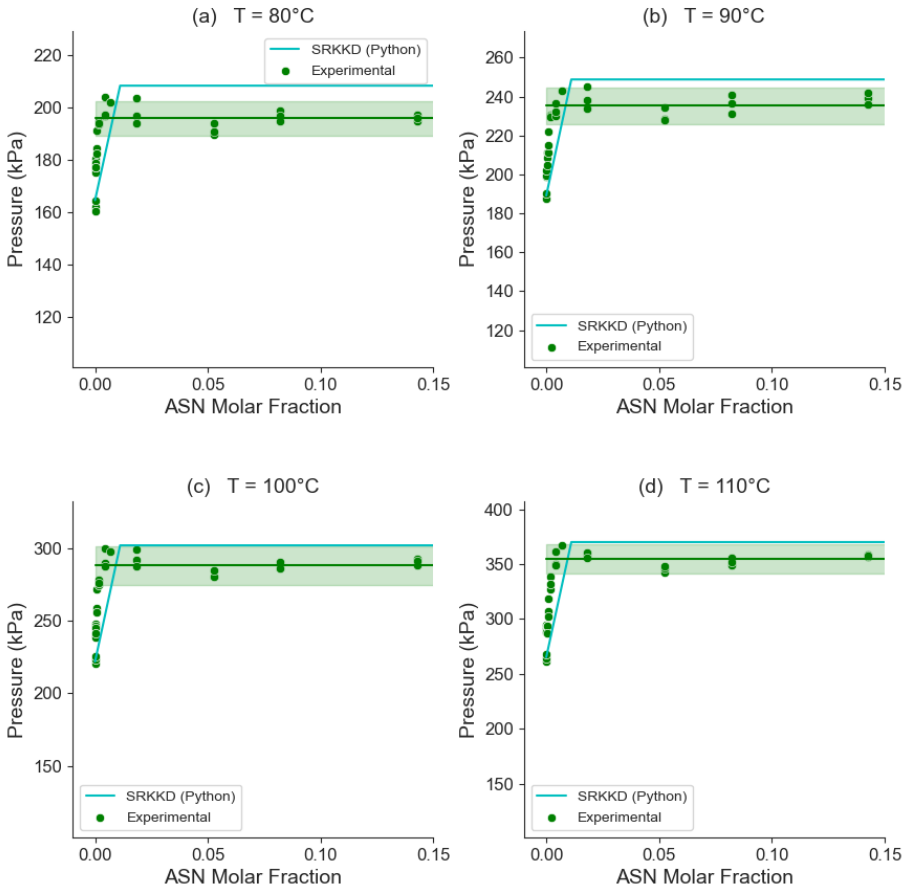


Figure 5-12. Expanded View for the Pressure for Highly Diluted PASN in the Water using data from the CREC-VL Cell and SRKKD (Python) values.

In summary and on the basis of the reported results, using the CREC VL Cell, one can argue that: a) at highly diluted hydrocarbon concentrations in water and highly diluted water concentrations in hydrocarbons, there are significant model deficiencies for assessing total pressure, b) at intermediate range molar concentrations, the fully immiscible model is deficient and requires some correction, accounting for partial solubility.

5.4.2. T-Student Test

As reported Chapter 4 and in a recent article by Lopez-Zamora et al. [5], a first step while developing the task of classifying the number of phases, is to determine if the mean value of the experimentally measured pressures is outside the 95% confidence interval of the

VLL equilibrium baseline level. To evaluate this, a *t-student test* can be considered. This approach allows one to establish under what conditions the mean of the baseline is different from the experimentally measured pressures, with a 95% confidence interval leading to a p-value smaller than $\alpha = 0.05$.

Figure 5-13 reports the p-values from the experiments, which always remain below the p-values from the t-test. This points to two completely miscible liquid phases, with this being true for 0.1wt%, 0.25wt% and 0.5wt% of PASN in water, in the 80 to 100 °C range. Additionally, when the naphtha content in PASN-water surpasses 1wt%, it displays complete miscibility at 90°C and above, p-values from the CREC VL Cell data exceed in this case, the p-values with α=0.05, with this being an indication of the two coexisting liquid phases.

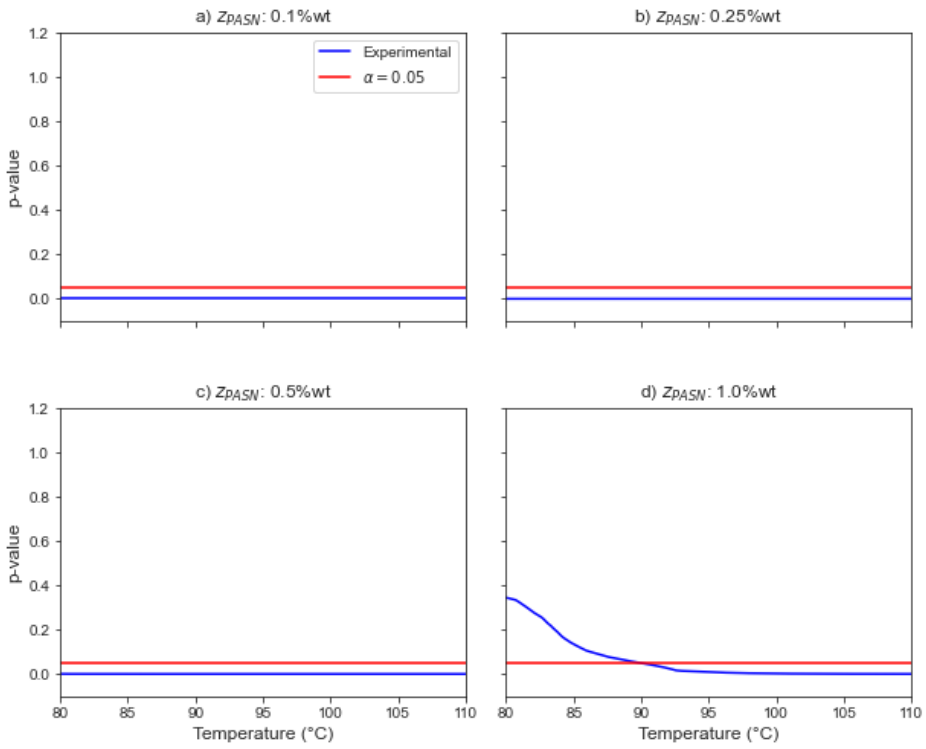
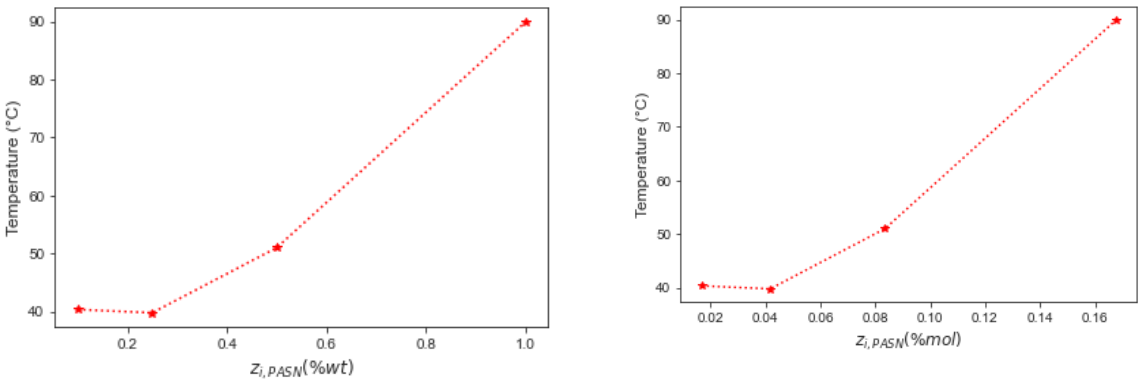


Figure 5-13. T-Student Test for Highly Diluted PASN in Water. Note: Reported lines represent the p-value from: a) the experimental data (blue line), b) the t-test with an α= 0.05.

Figure 5-14 further considers the “transition” temperatures, when the three phases become two phases, and this for PASN-water blends. One can notice that these transition temperatures show a progressive increase with higher PASN concentrations.



(a) (b)

Figure 5-14. Transition Temperatures at which the Three Phases become Two Phases, for Highly Diluted PASN in Water Mixtures, as Calculated with T-Student Test: (a) PASN Mass Percentage, (b) PASN Molar Percentage.

Figure 5-15 is obtained when calculating the TPD* representing an experimental mixture with 0.5%wt (0.0835%mole) PASN in water, at a temperature 80°C and 83.54 kPa. It is observed that the presence of negative values for TPD* suggests the presence of two liquid phases. However, and in disagreement with this, the experimental results from the CREC VL Cell show that there is only one single phase present (completely miscible mixture).

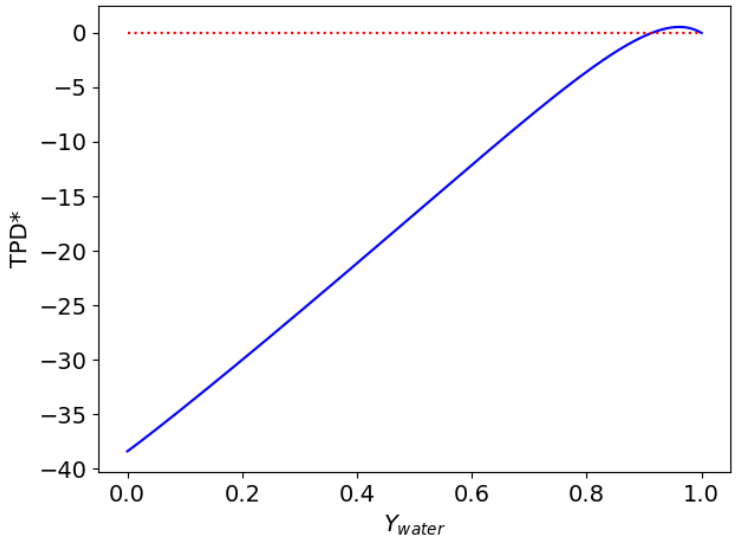


Figure 5-15. TPD* at 80°C, z_{PASN} = 0.00835 and 83.54 kPa – Trial Phase is Liquid. Note: Doted Red Line shows TPD* zero value

5.5. Classification of Phase Regions

Although a Tangent Plane Distance (TPD*) analysis could in principle be used to predict the number of phases in PASN/water mixtures, the parameters available in the technical literature do not account for the higher miscibility observed, during the experimental runs in the CREC VL Cell.

In this respect, and thanks to the large abundance of experimental data points obtained during runs with the CREC VL Cell, it is possible to establish more certain thermodynamics, and use this information as a powerful classification methodology to predict the number of phases. While this classification methodology was already proposed in our previous work, for a n-octane/water mixture [5], it is now being considered for the more complex PASN/water mixtures of this study.

In this regard, data was labeled according to the t-test results. In this case, 42% of the data correspond to the two-phase region (6462 data points) and 58% of the data correspond to the three-phase region (8889 data points). This is shown in Figure 5-16.

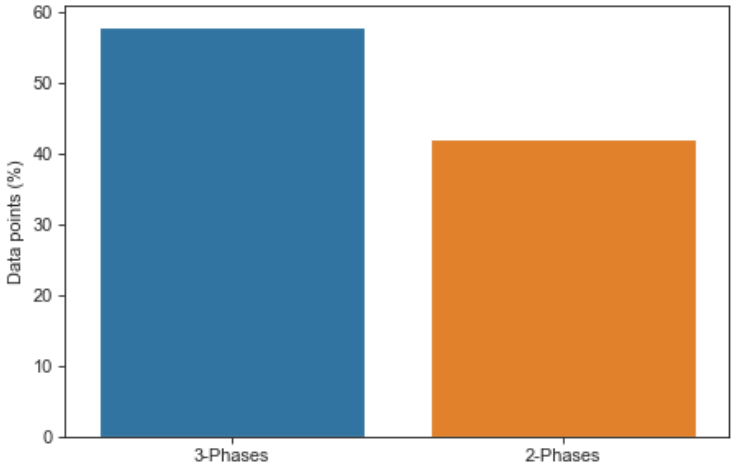


Figure 5-16. Distribution of the Three-Phase (VLLE) and Two-Phase (VL) Data Available.

This dataset can be considered balanced, as it is very near to the 50/50% perfect balanced dataset. Thus, the two best methods already proposed in Chapter 4 and Lopez-Zamora et al. (2021) [5] were applied as reported in Table 5-4. Additional descriptions of the K-Neighbors Classifier and Support Vector Classifier (SVC) are also reported in Section 2.2.

Table 5-4. Classification Models Implemented.

Model #	Type	Parameters	Class Weight option
1	K-Neighbors Classifier	Default	No
2	Support Vector Classifier (SVC)	Default	Yes

To avoid overfitting, 20% of the temperature data was dropped from the training data using a random selection. It was then added to the testing dataset. The remaining data points were split as 20% test, 80% training, using a "train test split" function from the Sklearn library. The performance of these classification models was measured using Precision, Recall and F1-score, as reported by the following equations:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5-7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5-8}$$

$$F_1 score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \qquad (5\text{-}9)$$

Figure 5-17 reports the Confusion Matrix for the KNN (Figure 5-17a) and SVC models (Figure 5-17b). In the case of KNN, only 0.46% of the 3-phase data points are mispredicted, while for the SVC, 98% of the 2-phase data are correctly predicted and only 1.7% are classified incorrectly.

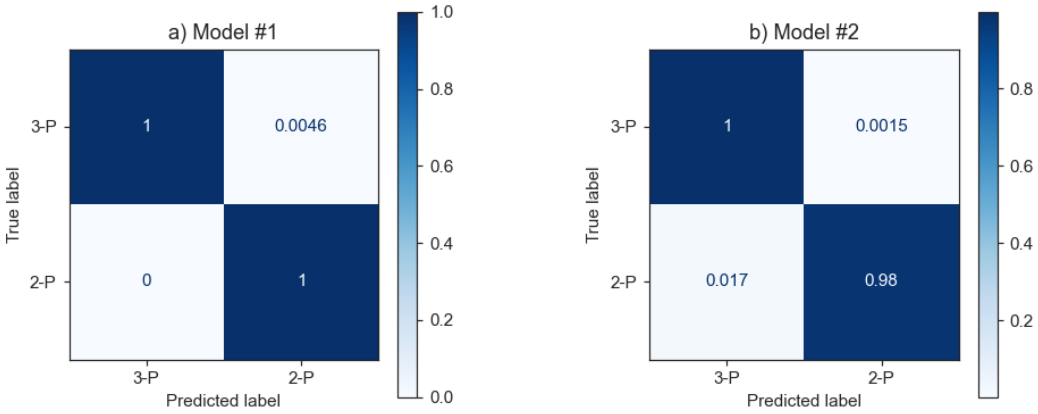


Figure 5-17. Confusion Matrix for a) KNN Model (Model 1) and b) SVC Model (Model 2). Note: Calculated based on test dataset.

Table 5-5 shows both the Precision and Recall for both the KNN and the SVC models. One can notice that the KNN model provides up to 99% Precision for the two phases, while the Recall for the SVC is up to 98% for the two-phase data.

Table 5-5. Classification Report for KNN and SVC Models.

K-Neighbors Classifier (KNN)			
	Precision	Recall	F1 score
3-Phases	1.00	1.00	1.00
2-Phases	0.99	1.00	1.00

SVC			
	Precision	Recall	F1 score
3-Phases	0.99	1.00	0.99
2-Phases	1.00	0.98	0.99

Furthermore, and when considering the model calibrations, which are established if the predicted probabilities correspond to the expected probability distribution for each class, it can be noticed that the models are not as close to the perfectly calibrated line as shown in Figure 5-18. In this case, a recalibration is suggested to increase the confidence of the predictions.

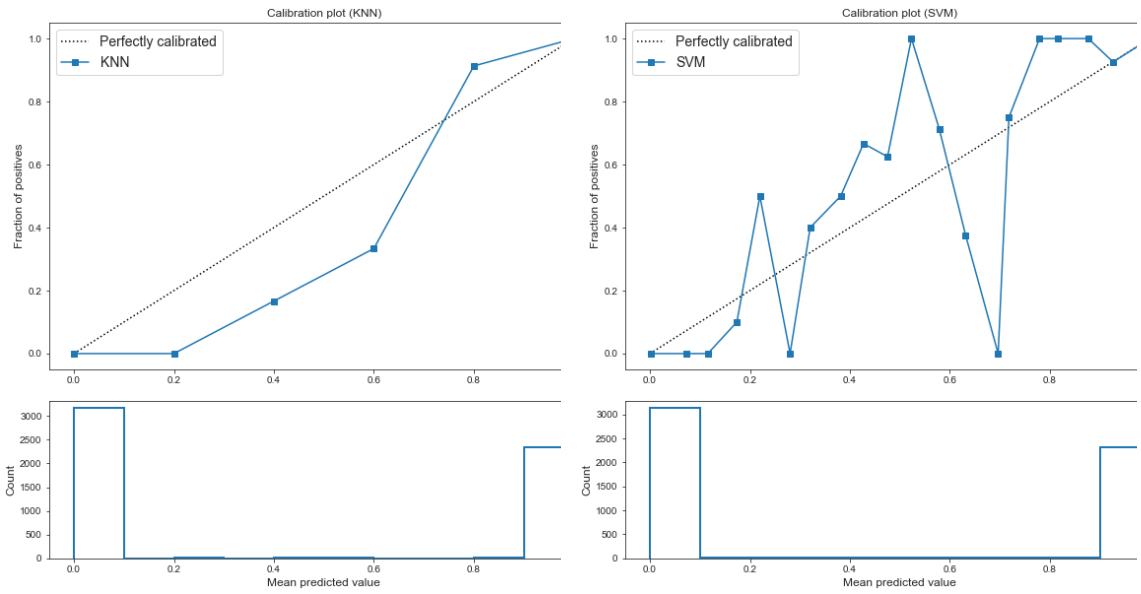


Figure 5-18. Initial Calibration of KNN and SVC Methods. Note: Calculated based on test dataset.

In order to recalibrate the models, the Calibrated Classifier CV function from Sklearn library is used. Figure 5-19 reports the recalibrated plots. It is possible to observe, especially for the SVC that the recalibrated model it is quite close to the perfectly calibrated line. As a result, this model is selected.

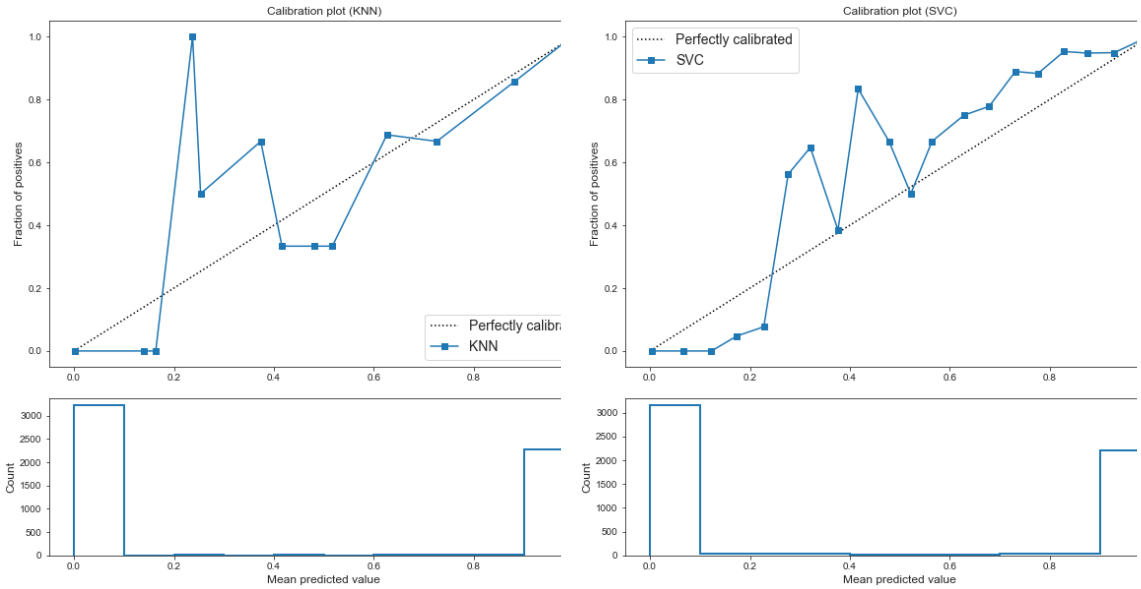


Figure 5-19. Recalibration of KNN and SVC Methods. Note: Calculated based on test dataset.

Although the measurements in the Confusion Matrix (Figure 5-20) and the Classification Report, as given in Table 5-6, are slightly lower than the ones desired, the recalibrated models give enhanced confidence. In this sense, it is considered that the SVC provides the best representation of the classification of two and three phases, under the conditions of interest of the PASN/water systems.

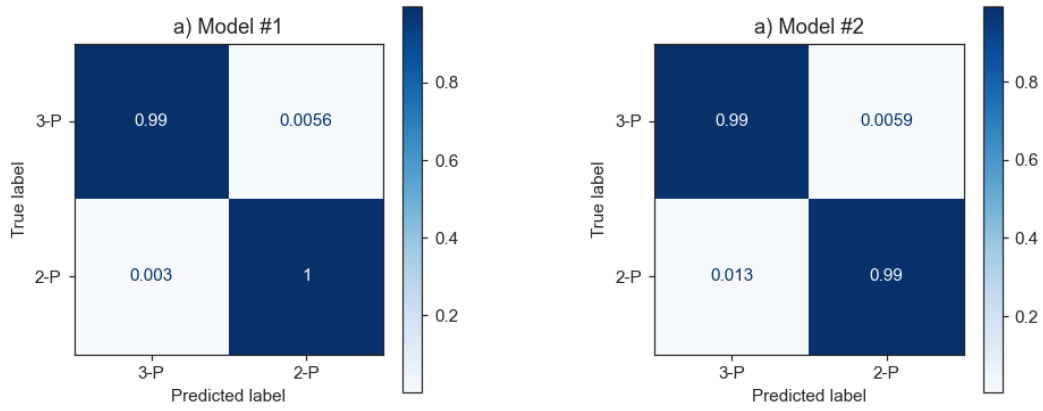


Figure 5-20. Confusion Matrix for a) KNN Model (Model 1) and b) SVC (Model 2). Note: Calculated based on test dataset.

Table 5-6. Classification Report for KNN and SVC Models after Recalibration. Note: Calculated based on test dataset.

K-Neighbors Classifier (KNN)			
	Precision	Recall	F1 score
3-Phases	1.00	0.99	1.00
2-Phases	0.99	1.00	0.99

SVC			
	Precision	Recall	F1 score
3-Phases	0.99	0.99	0.99
2-Phases	0.99	0.99	0.99

5.6. Conclusions

a) Significant differences were observed between experimentally determined total pressure obtained in a CREC VL Cell, and the VLLE results predicted using SRKKD (Python) for highly diluted PASN/water mixtures.

b) Given the similar behaviour the thermodynamics of PASN/water and n-Octane/water mixtures, it is proven that PASN/water mixtures can be treated as a pseudo-binary mixture.

c) A TPD (The Tangent Plane Distance) methodology considered for n-octane/water and PASN/water mixtures, shows intrinsic shortcomings for prediction of the two phase and three phase regions.

d) A classification methodology (Machine Learning) is proposed instead, to predict the number of phases in PASN/water blends. Several classification parameters are establish on this basis showing positive indicators, with a SVC Model being considered as the one providing better phase classification results.

6. Flash calculations for n-Octane/ Water and PASN/water Systems

As stated in section 2.1.3, the phase equilibrium calculation problem deals with two main issues [10]: i) phase stability, and ii) phase-split. While the number of phases at thermodynamic equilibrium is an unknown condition, as proposed in Chapter 4 and 5, experimental data in the CREC-VL Cell can provide the two phases and three phases region information. This allows "a priori" classification of the number of phases, reducing computational cost, and giving better initial estimates for the phase split calculation once the model is trained.

In this chapter, traditional flash calculations are compared with a Machine Learning (ML) approach. First, traditional calculations are reported explaining the challenges presented for the solution of the Rachford-Rice equations involved and the different approaches reported in the technical literature. A comparison of flash calculations between water/n-octane and PASN/water mixtures using SRKKD EoS is also provided.

Convergence calculations and their issue for water/n-octane and PASN/water systems are presented through the analysis of Rachford-Rice derived surfaces. This analysis highlights the value of an ML approach which can be developed on the basis of abundant experimental data available from the CREC-VL experimental Cell experiments.

6.1. Traditional Phase Split Calculations

Traditionally, the phase split calculations are performed by solving Rachford-Rice (RR) equations as shown in Table 2-3, involving phase equilibrium constants. Rachford-Rice equations are nonlinear functions obtained from the equal chemical potentials combined with material balances [11].

In the case of the three-phase flash, the main equations (Equations (6-1) to (6-6)) are reported below, with the hydrocarbon phase used as a as reference [11,127].

$$f_V(\beta) = RR_y = \sum_{i=1}^{N}(y_i - x_i^L) = \sum_{i=1}^{N}\frac{z_i(K_i^V - 1)}{H_i} = 0 \tag{6-1}$$

$$f_W(\beta) = RR_W = \sum_{i=1}^{N}(x_i^W - x_i^L) = \sum_{i=1}^{N}\frac{z_i(K_i^W - 1)}{H_i} = 0 \tag{6-2}$$

$$H_i = 1 + \beta^V(K_i^V - 1) + \beta^W(K_i^W - 1) \tag{6-3}$$

Or

$$f_V(\beta) = \sum_{i=1}^{N}\frac{z_i(1 - K_i^V)}{t_i} = 0 \tag{6-4}$$

$$f_W(\beta) = \sum_{i=1}^{N}\frac{z_i(1 - K_i^W)}{t_i} = 0 \tag{6-5}$$

$$t_i = 1 - \beta^V(1 - K_i^V) - \beta^W(1 - K_i^W) \tag{6-6}$$

In this respect, the root-finding solution process is complex given these equations present discontinuities at their extremes (division by zero), and may have an almost flat slope near their roots [128].

Popular methods for phase split calculations are as follows: a) successive substitution, b) quasi-Newton, Newton, c) steepest-descend and d) modifications and combinations of them [11]. To solve Rachford-Rice equations, the numerical method's choice is influenced by the independent variables that are selected, such as component mole fractions, equilibrium ratios, or the logarithm of equilibrium ratios [7].

When equilibrium constant ratios or logarithms of equilibrium constants are considered, a Newton Raphson method is typically applied. In the case of the mole fractions, either a Newton's method or a minimization of Gibbs free energy can be considered [129]. The logarithm of K-values is usually preferred given the use of mole fractions may create an

ill-defined Jacobian and, the natural logarithm stabilizes the Newton method when K-values are of different orders of magnitude [7].

Typically, multiphase flash split calculations with 3 or more phases, involve one outer loop where the equilibrium constants are solved and one inner loop where the mass balances (Rachford Rice equations) are evaluated. The end goal is to determine the phase mole fractions and the compositions for a given set of K-values [130]. This inner loop is known as “constant-K” flash [131].

Given the above, a general algorithm to solve multiphase flash calculations is described in Figure 6-1. It can be observed that within each successive substitution step, the calculation of the Rachford-Rice equations, designated as “constant-K” flash are solved independently.

One should note that the solution of the two-phases constant-K flash calculation is a relatively easy one. However, and in the case where one has to account for a three-phases flash , these calculations may become extensive and challenging. This is due to the non-linearity of the objective resulting functions [7].

The “constant-K” flash is discussed in Section 6.2. In this respect, one could mention that in phase split calculations, good initial estimates increase the probability to find the global minimum Gibbs free energy, with an initial guess from stability testing or previous simulation timestep being an option [11]. To accomplish this, constraints for the initial estimates are usually needed, as suggested previously by Okuno et al. [130] or Leibovici & Neoschil [132].

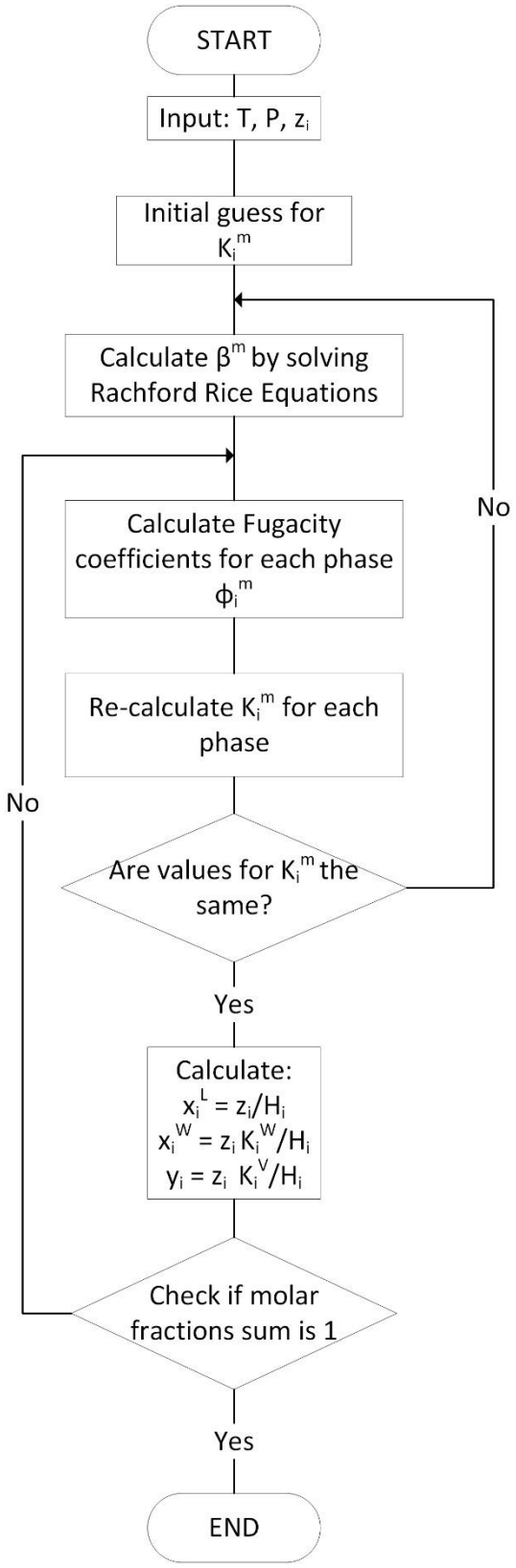


Figure 6-1. Traditional Algorithm for Three Phases Flash Calculations. Adapted from [77]

6.2. Constant-K flash solution

The solution of the "constant-K flash" has been studied previously using two different approaches [133], which include: i) minimization techniques and ii) direct solution of the RR equations. Tranenstein (1985) [129] proposed a constrained minimization of the Gibbs free energy to solve the two phases problem, while Leibovicy and Jean (1995) [132] used a Newton procedure with a relaxation parameter to solve the multiphase problem. Furthermore, Okuno et al. (2010) [130] minimized the non-monotonic convex function using the independent phases mole fractions, Haugen et al. (2011) [133] used a two-dimensional bisection method to provide good initial guesses for the Newton algorithm in the three phases case, Li & Firoozabadi (2012) [11] employed stability testing as an initial guess for phase split calculations with a two-dimensional bisection method for two and three phases; and Yan & Stenby (2014) [134] proposed Householder's high order iteration method together with a method to improve the initial estimate for the two phases problem. More recently, Fernandez-Martinez et al. (2020) [128], applied an associated polynomial to obtain all the roots of a two-phases isothermal flash.

One of the most popular approaches to solve the "constant-K" flash problem is to use a Newton Raphson to solve Rachford-Rice equations (Eqs. (6-1) and (6-2)), getting values for β^V and β^W. In thate case, the Newton Raphson method considered is given by Equations (6-7) to (6-11).

$$\beta^{m,new} = \beta^{m,old} - [\nabla f(\beta^m)^T]^{-1}[f(\beta^m)] \tag{6-7}$$

$$\beta^{m,new} = \beta^{m,old} - [J(\beta^m)]^{-1}[f(\beta^m)] \tag{6-8}$$

$$\beta^m = \begin{bmatrix} \beta^V \\ \beta^W \end{bmatrix} \tag{6-9}$$

$$f(\beta^m) = \begin{bmatrix} f_V(\beta^m) \\ f_W(\beta^m) \end{bmatrix} \tag{6-10}$$

$$J(\beta^m) = \begin{bmatrix} \sum_{i=1}^{N} \frac{-z_i(K_i^V - 1)^2}{{H_i}^2} & \sum_{i=1}^{N} \frac{-z_i(K_i^V - 1)(K_i^W - 1)}{{H_i}^2} \\ \sum_{i=1}^{N} \frac{-z_i(K_i^V - 1)(K_i^W - 1)}{{H_i}^2} & \sum_{i=1}^{N} \frac{-z_i(K_i^W - 1)^2}{{H_i}^2} \end{bmatrix} \tag{6-11}$$

Or

$$J(\beta^m) = \begin{bmatrix} \sum_{i=1}^{N} \frac{z_i(1 - K_i^V)^2}{{t_i}^2} & \sum_{i=1}^{N} \frac{z_i(1 - K_i^V)(1 - K_i^W)}{{t_i}^2} \\ \sum_{i=1}^{N} \frac{z_i(1 - K_i^V)(1 - K_i^W)}{{t_i}^2} & \sum_{i=1}^{N} \frac{z_i(1 - K_i^W)^2}{{t_i}^2} \end{bmatrix} \tag{6-12}$$

One should note that the Newton Raphson method's solution can converge to a non-desired root, with this being a function of the initial guest. It can lead as well, to numerical divergence, with this being an inherent characteristic of the non-linear equations being solved [7]. As shown by Hinojosa-Gomez et al. (2012) [135], Newton's method fails to converge near the critical point and phase boundaries. Thus, good initial guesses are required for the phase fractions (β) calculations, with poor initial estimates leading to incorrect roots or being unable to find a numerical solution [133,135].

In this respect, the initial guess for β^V, β^W should be constrained within the proper solution domain. In this sense, while approaching the numerical solution of the constant-K flash, it is advantageous to consider this as an iterative constrained minimization calculation, instead of being a root-finding problem as [130]:

$$\beta^{m,new} = \beta^{m,old} - [\nabla^2 \mathrm{F}(\beta^m)]^{-1}[\nabla F(\beta^m)] \tag{6-13}$$

One should mention that the $F(\beta)$ refers to a convex function, as proven by Okuno et al. 2010 [130], with *N* linear constrains, and with f representing the t F gradient, with the condition of having a symmetric Jacobian matrix [130,136]. If this is the case, the $F(\beta)$ scalar function involves a gradient vector which represent the RR equations [130].

By integrating f_j (Equation (6-4) and (6-5)) with respect to β^j one can can obtain Equation (6-15), having the integration constant set to zero:

$$F(\beta) = \sum_{i=1}^{N} z_i \ln|H_i| \tag{6-14}$$

Or alternatively

$$F(\beta) = \sum_{i=1}^{N} -z_i \ln|t_i| \tag{6-15}$$

While examining the possible mathematical solutions for a multiphase system at equilibrium with the number of phases being larger than 2 ($N_p > 2$), one can notice that the range of these solutions is wider than the space of the physical solutions [132]. To address this matter, Leibovicy & Neoschil (1995) [132] proposed that numerical solutions should be limited by hyperplanes defined by:

$$1 + \sum_{l=2}^{l=N_p} (K_{li} - 1)\beta^l = 0 \tag{6-16}$$

$$1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1) = 0 \tag{6-17}$$

In this respect, it is important to notice that, in Equation (6-14), if the region $t_i > 0$ is unbounded and the following applies: a) the function is monotonic, b) it does not have a minimum, and b) there is no solution to the constant-K flash with N_p phases [130].

In that sense, Okuno et al. [130] proposed a feasible solution region based on the non-negativity of phase component mole fractions in a given phase L ($0 \leq x_i^L \leq 1$ $(i = 1, 2, \dots N_c)$) such as :

$$x_i^L = \frac{z_i}{H_i} = \frac{z_i}{t_i} \tag{6-18}$$

$$x_i^W = x_i^L K_i^W \tag{6-19}$$

$$y_i = x_i^L K_i^V \tag{6-20}$$

Then, from Equation (6-18) and with positive phase component mole fractions in phase *L,* it results:

$$0 \leq \frac{z_i}{t_i} \leq 1 \tag{6-21}$$

$$0 \leq z_i \leq t_i \tag{6-22}$$

$$0 \leq z_i \leq 1 - \beta^V(1 - K_i^V) - \beta^W(1 - K_i^W) \tag{6-23}$$

$$0 \leq \beta^V(1 - K_i^V) + \beta^W(1 - K_i^W) \leq 1 - z_i \tag{6-24}$$

And from Equations (6-19) and (6-20):

$$0 \leq \frac{z_i}{t_i} K_i^W \leq 1 \tag{6-25}$$

$$0 \leq z_i K_i^W \leq t_i \tag{6-26}$$

$$0 \leq z_i K_i^W \leq 1 - \beta^V(1 - K_i^V) - \beta^W(1 - K_i^W) \tag{6-27}$$

$$0 \leq \beta^V(1 - K_i^V) + \beta^W(1 - K_i^W) \leq 1 - z_i K_i^W \tag{6-28}$$

$$0 \leq z_i K_i^V \leq t_i \tag{6-29}$$

$$0 \leq z_i K_i^V \leq 1 - \beta^V(1 - K_i^V) - \beta^W(1 - K_i^W) \tag{6-30}$$

$$0 \leq \beta^V(1 - K_i^V) + \beta^W(1 - K_i^W) \leq 1 - z_i K_i^V \tag{6-31}$$

Equations (6-24), (6-28) and (6-31) can be summarized as follows:

$$\beta^V(1 - K_i^V) + \beta^W(1 - K_i^W) \leq min(1 - z_i, 1 - z_i K_i^W, 1 - z_i K_i^V) \tag{6-32}$$

For $i = 1, 2, \ldots N_c$

Okuno et al. [130] proposed further, a general definition of these thermodynamic parameters as: $a_i^T \beta \leq b_i$ with $a_i = 1 - K_i^p, \beta = [\beta^V, \beta^W]$ and $b_i = min\left(1 - z_i, min\left(1 - z_i K_i^p\right)\right)$.

With the constant-K flash problem solved by Equation (6-33).

$$\text{Minimize: } F(\beta) = \sum_{i=1}^{N} -z_i \ln|t_i| \text{ subject to } a_i^T \beta \leq b_i \tag{6-33}$$

In that sense, Equation (6-33) accounts for the "negative flash" case. One should note that when the iterative flash procedure yields β values in the $\beta < 0$ or $\beta > 1$ ranges, this leads to a "negative flash" [137]. These "negative flashes" while "not physically acceptable" roots can be preserved for the next calculation step. This is the case given the anticipated function continuity. It is interesting to mention that the Okuno et al.'s resulting algorithm performs better with initial negative roots than when the condition $0 \leq \beta \leq 1$ is complied from the very beginning, in the first calculation step [137].

6.3. Algorithm to Solve the Flash Unit for Water/PASN mixtures

After discussing the solution of the constant-K flash problem, it is possible to complete the flash calculations as described in Figure 6-1. In this sense, the steps involved in the flash calculations are as follows:

1. Input the operating and feed conditions: T, P, z_i
2. Provide an initial guess for the K-values.
3. Solve Rachford-Rice equations as discussed in Section 6.4, minimizing Equation (6-33).
4. Calculate x_i^L, x_i^W and y_i from Equations 2.41 to 2.43
5. Calculate the fugacity coefficients from Equation 3.22
6. Calculate Objective functions and compare with tolerance

$$F_{obj}^{V} = \ln K_i^V - \ln \phi_i^L + \ln \phi_i^V \quad (6\text{-}34)$$

$$F_{obj}^{W} = \ln K_i^W - \ln \phi_i^L + \ln \phi_i^W \quad (6\text{-}35)$$

7. Update K-values

$$\ln K_i^V = \ln \phi_i^L - \ln \phi_i^V \quad (6\text{-}36)$$

$$\ln K_i^W = \ln \phi_i^L - \ln \phi_i^W \quad (6\text{-}37)$$

8. Check that $\sum_{i=1}^{N} x_i^L = 1$, $\sum_{i=1}^{N} x_i^W = 1$ and $\sum_{i=1}^{N} y_i = 1$

One should note that the main problem with the proposed algorithm is that it may be computationally very expensive [10], with this being a function of the initial guesses chosen, as well as presenting both convergence and accuracy issues .

When applying Newton Raphson Method (Equations (6-7) to (6-11)), using an initial estimate within the set boundaries (as presented in the following section) such as $\beta_{sup}^{V} = 0.4$ and $\beta_{sup}^{W} = 0.2$, for the water/PASN one gets, employing SRKKD EoS model and Python, $\beta^V = 0.10004025$ and $\beta^W = 0.44494793$ root with 4 iterations only. One should note that in this respect that HYSYS V9 result in this case were $\beta^V = 0.1$ and $\beta^W = 0.4449395$, with the difference being much lower than 0.1%. In contrast and as expected, for the water/n-octane system, the calculation reaches the 10000 maximum number of iterations with the obtained results not having physical meaning : $\beta^V = -10.9198$ and $\beta^W = 2.6645 * 10^{-15}$.

Given the above, the function $F(\beta)$ (Equation (6-33)) for the water/PASN mixture was minimized using different methods within the minimize functions available in the Python Optimize library. Tolerance was set in the 10^{-8} range, with the percentage of difference for the calculated β values being lower than 0.3% . Table 6-1 reports the various methods tested. Best results were obtained using the Constrained Optimization BY Linear Approximation (COBYLA) method.

Table 6-1. Results for water/ASN system using different methods for the minimize function of Scipy - Optimize package for Python

Initial estimate	**Method**	**Result**	**%difference**
$\beta = [0.4 \quad 0.2]$	Constrained Optimization BY Linear Approximation (COBYLA)	$\beta = [0.10004048 \quad 0.44494783]$	$[0.04\% \quad 0.0019\%]$
$\beta = [0.4 \quad 0.2]$	Sequential Least Squares Programming (SLSQP)	$\beta = [0.10004144 \quad 0.44494715]$	$[0.04\% \quad 0.0017\%]$
$\beta = [0.4 \quad 0.2]$	trust-constr	$\beta = [0.09995574 \quad 0.44541656]$	$[0.04\% \quad 0.1072\%]$
$\beta = [0.9 \quad 0.1]$	Constrained Optimization BY Linear Approximation (COBYLA)	$\beta = [0.1000405 \quad 0.44494781]$	$[0.04\% \quad 0.0019\%]$
$\beta = [0.9 \quad 0.1]$	Sequential Least Squares Programming (SLSQP)	$\beta = [0.09993294 \quad 0.44500976]$	$[0.07\% \quad 0.016\%]$
$\beta = [0.9 \quad 0.1]$	trust-constr	$\beta = [0.09976697 \quad 0.44642034]$	$[0.23\% \quad 0.32\%]$

However, and in spite of this, none of the considered methods were able to get meaningful physical solutions for water/n-octane blends, as reported in Table 6-2. Not even a genetic algorithm was able to solve this case, arriving at a result of $\beta = [-20.7 \quad 14.2]$. Reasons for this are described in Section 6.4 Figure 6-11.

Table 6-2. Results for water/n-Octane system using different methods for the minimize function of Scipy - Optimize package for Python

Initial estimate	Method	Result
$\beta = [0.4 \quad 0.2]$	Constrained Optimization BY Linear Approximation (COBYLA)	$\beta = [-0.1780 \quad 0.6150]$
$\beta = [0.4 \quad 0.2]$	Sequential Least Squares Programming (SLSQP)	$\beta = [0.4146 \quad 0.2220]$
$\beta = [0.4 \quad 0.2]$	trust-constr	$\beta = [0.4150 \quad 0.2224]$
$\beta = [0.9 \quad 0.1]$	Constrained Optimization BY Linear Approximation (COBYLA)	$\beta = [-0.0142 \quad 0.5063]$
$\beta = [0.9 \quad 0.1]$	Sequential Least Squares Programming (SLSQP)	$\beta = [0.8080 \quad -0.0388]$
$\beta = [0.9 \quad 0.1]$	trust-constr	$\beta = [0.8086 \quad -0.0377]$

In the same way, the VLLE for water/PASN can be calculated accordingly, as reported in Table 6-4. In this case, the bubble pressure ($\beta^V = 0$), which is of interest for this research, can be calculated using both Python and Hysis V9 as reported in Table 6-4,

Table 6-3. Pressure Calculation Results for water/ASN system in VLLE region (no air)

Conditions	Python Results (kPa)	HYSYS V9 results (kPa)	Difference (%)
T = 80°C Z_w = 0.5	86.10	91.53	6.31%
T = 80°C Z_w = 0.1	86.10	91.53	6.31%
T = 80°C Z_w = 0.9	86.10	91.53	6.31%
T = 110°C Z_w = 0.5	242.55	254.41	4.89%
T = 110°C Z_w = 0.1	242.55	254.41	4.89%
T = 110°C Z_w = 0.9	242.55	254.41	4.89%

One should note however, that the described Python algorithm for water/PASN works better with Python with Pressure results, as shown in Figures 6-2 and 6-3, being in better agreement with experimental data from a CRE-VL Cell .

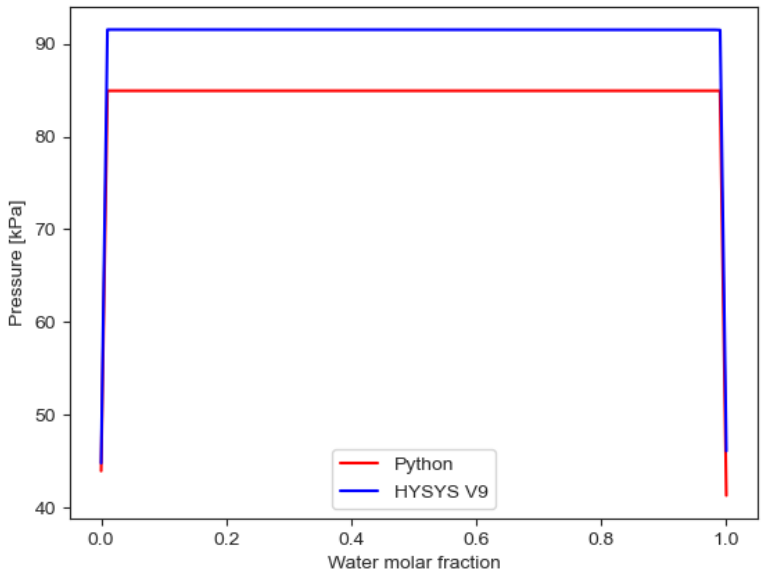


Figure 6-2. Comparison of VLLE results for water/ASN using Python and HYSYS V9 (no air)

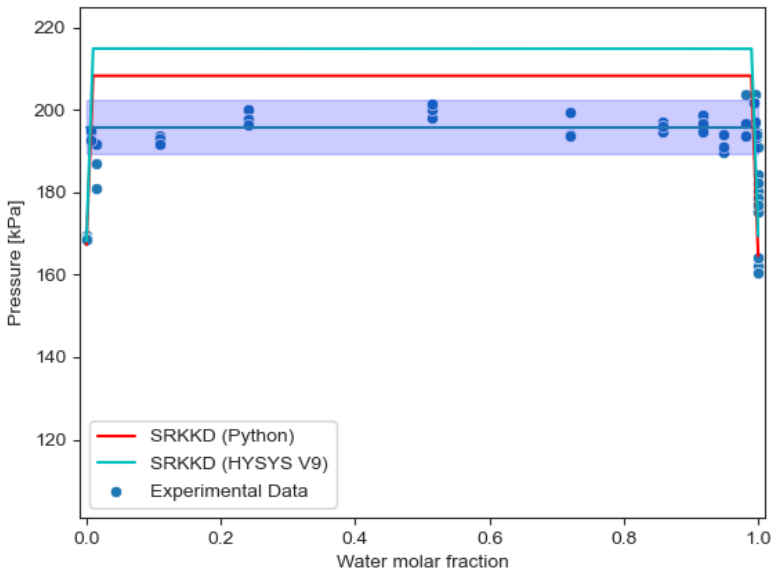


Figure 6-3. Comparison of VLLE experimental results for water/PASN using Python and HYSYS V9 at 80°C

On the other hand, in the case of water/n-octane blends, the described algorithm presents convergence problems. To describe these issues, it is important to establish how the numerical Rachford-Rice equations (Equations 6.1 to 6.3) influence these type of iterative calculations for both water/n-octane and water/PASN mixtures. To address this matter, the following section evaluates the approach proposed by Li & Firoozabadi [138] and the boundaries set by Okuno et al. [130].

6.4. Issues with Constant-K solution Calculations

Li & Firoozabadi [138] reported some examples of how RR_y and RR_w surfaces (Rachford-Rice surfaces) change while β^V and β^w are varied, with β^V and β^w parameters representing the vapor and water fraction respectively. A display of one of their examples reported by Li & Firoozabadi for a general case [138] are shown in Figure 6-4, for RR_y and RR_w intersecting the *z=0* plane .

One should note that the triangle defined in Figure 6-4 by the vertices (0,1), (0, 0) and (1, 0) represents the solution domain [138], with the solution at $RR_y = 0$ and $RR_w = 0$ plane shown with a red dot.

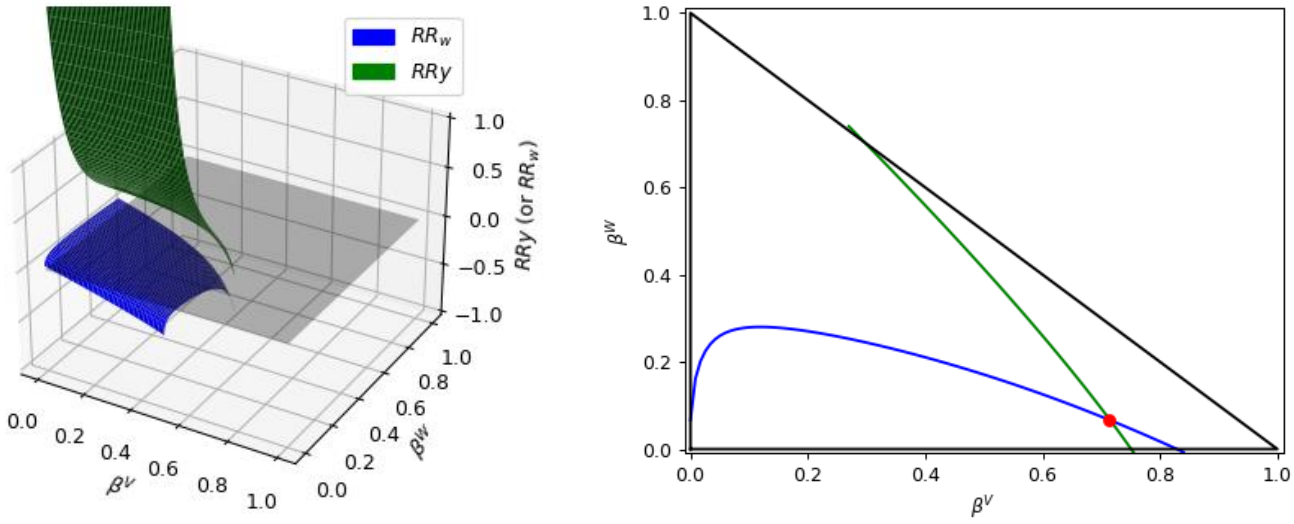


Figure 6-4. (a) $\boldsymbol{RR_y}$ and $\boldsymbol{RR_w}$ surfaces intersecting the z=0 plane highlighted in grey (Adapted from [138]), (b) The $\boldsymbol{\beta^V}$ and $\boldsymbol{\beta^w}$ lines at z=0 plane. Note: The "red dot" represents the solution at $RR_y = 0$ and $RR_w = 0$ plane

In this regard, and if one attempts to develop a similar based Python calculation for an octane/water blend "constant-K" flash one can observe that it is not possible to obtain a converging iterative solution. One can notice as well that this is true, for a wide range of octane in water concentrations in the 0.5-99.75 wt% range.

As a result, and to provide a sound explanation of the findings, the following steps were followed:

First step. It involves the SRKKD EoS model and HYSYS V9 with "constant-K" flash simulations. Equilibrium constants are approximated on this basis and used later for a thorough analysis of Rachford-Rice equations.

Second step. It considers a "constant-K" flash calculation using the equilibrium constant calculated in step 1. This helps to provide a good understanding of how the Rachford-Rice equations perform in this kind of hydrocarbon/water mixtures.

6.4.1. Octane-Water Blends

For illustrating the calculation, a 3-phase separator was specified in HYSYS V9 feeding a 100 kgmol/h blend with a 50% mol water / 50% mol n-octane mixture was considered for the first step. Working conditions for this 3-phase separator were T = 80°C and a vapour fraction of 0.1. In this case, the presence of air was not considered. Results obtained are reported in Table 6-4.

Table 6-4. HYSYS V9 results for 3 phases flash calculations at T=80°C using SRKKD

		Molar Fraction	
	Molar Flow (kgmol/h)	Water	n-Octane
Feed	100.00	0.5	0.5
Hydrocarbon Phase	46.94	$6.09*10^{-3}$	0.9939
Aqueous Phase	43.06	≈1.0	1.20E-06
Vapor	10.00	0.6651	0.3349

One could notice that while developing the second calculation step, with the K_i^V and K_i^W constants obtained from HYSYS V9 and using them now in Python, the RR_y and RR_w values were in a low level range as in Figure 6-5a . It was also possible to further notice, as shown in Figure 6-5a, that the values of β^V and β^w changed in a restricted domain. Furthermore it was also possible to see that if the β^V value was higher than 0.63, or β^w higher than 0.58, the solution for RR_y and RR_w will not converge .

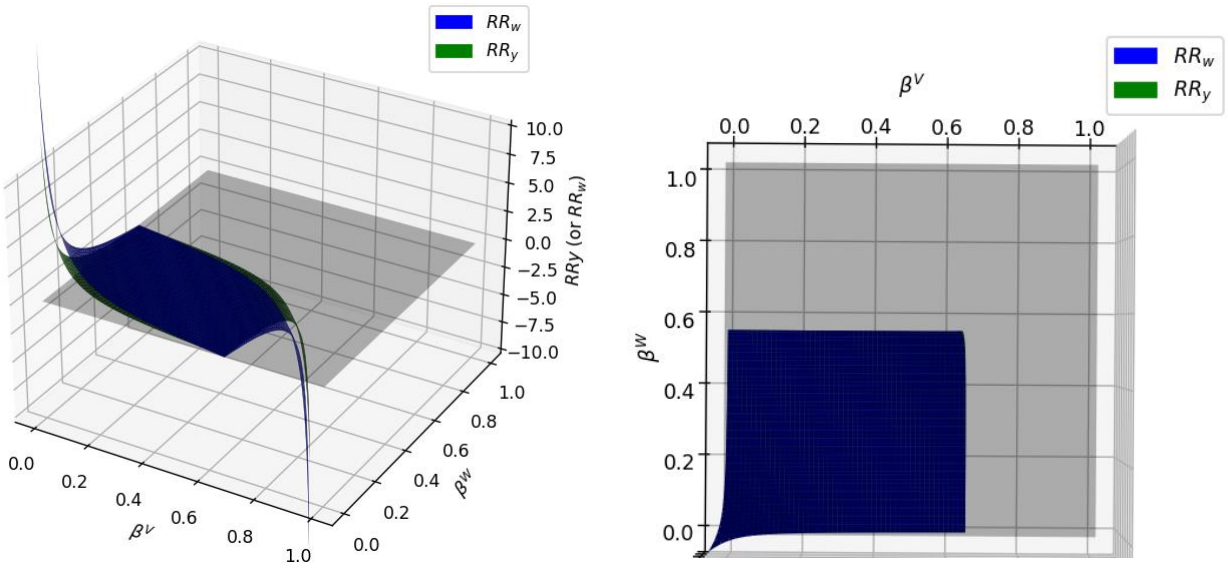


Figure 6-5. $\boldsymbol{RR_y}$ and $\boldsymbol{RR_w}$ surfaces intersecting the *z=0* plane for 50%mole water/n-octane at T = 80°C a) 3D surface, b) top view

Additionally and when RR_y and RR_w were considered at z= 0, the obtained β^W for different values of β^V led to two parallel and z=0 plane superimposed RR_y and RR_w straight lines, as shown in Figure 6-6. One should mention that, in contrast, a HYSYS V9 solution was obtained as identified with a red dot in Figure 6-6.

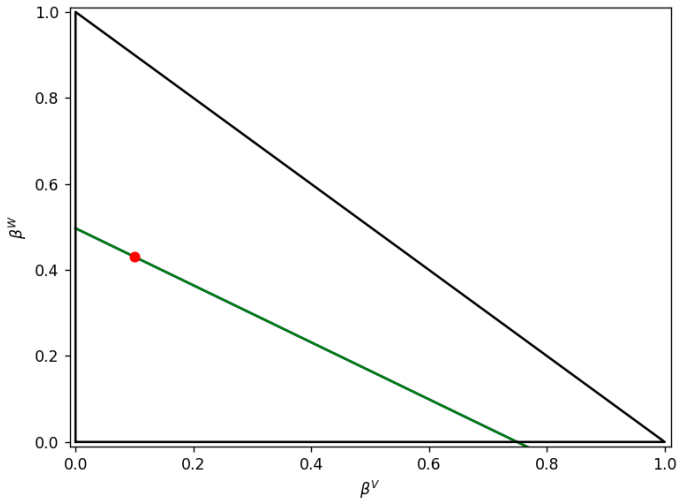


Figure 6-6. Lines for $\boldsymbol{RR_y} = 0$ and $\boldsymbol{RR_w} = 0\ at\ z = 0$ at T = 80°C. Note: The red point represents HYSYS solution

Thus, and as a result of this and under those conditions, one can understand why the iterative calculations trying to find an intersection of the RR_y and RR_w lines fail and the "constant-K " solution remain unknown.

Regarding these results, Haugen & Firoozabadi [133], advanced that algorithms of this type, solving the RR equations can fail, when the lines at $RR_y = 0$ and $RR_w = 0$ are parallel in the domain of interest. In this respect, these authors designated these conditions as the result of a "bicritical point" where two of the phases have very similar compositions. They identified three different kinds of "bicritical regions" [133]: i) $K_i^V \approx 1$, ii) $K_i^W \approx 1$or iii) $K_i^V \approx K_i^W$. In this case, the K-values for water/n-Octane mixtures are as follows: $K_i^V = [1.092510^2, 3.369210^{-1}]$ and $K_i^W = [1.642610^2, 1.210510^{-6}]$. As a result and given the conditions considered involving $K_i^V \approx K_i^W$, they could be considered in partial agreement with case iii) from Haugen & Firoozabadi criteria [133].

One should note as well that for different temperatures, in this case 110°C (Table 6-5), the behaviour of n-octane/water mixtures display similar calculation challenges, as it can be observed in Figure 6-7 and Figure 6-8.

Table 6-5. HYSYS V9 results for 3 phases flash calculations at T=100°C using SRKKD

		Molar Fraction	
	Molar Flow (kgmol/h)	Water	n-Octane
Feed	100.00	0.3	0.7
Hydrocarbon Phase	61.38	$1.60*10^{-2}$	0.9840
Aqueous Phase	8.62	≈1.0	$2.06*10^{-6}$
Vapor	30.00	0.6799	0.3201

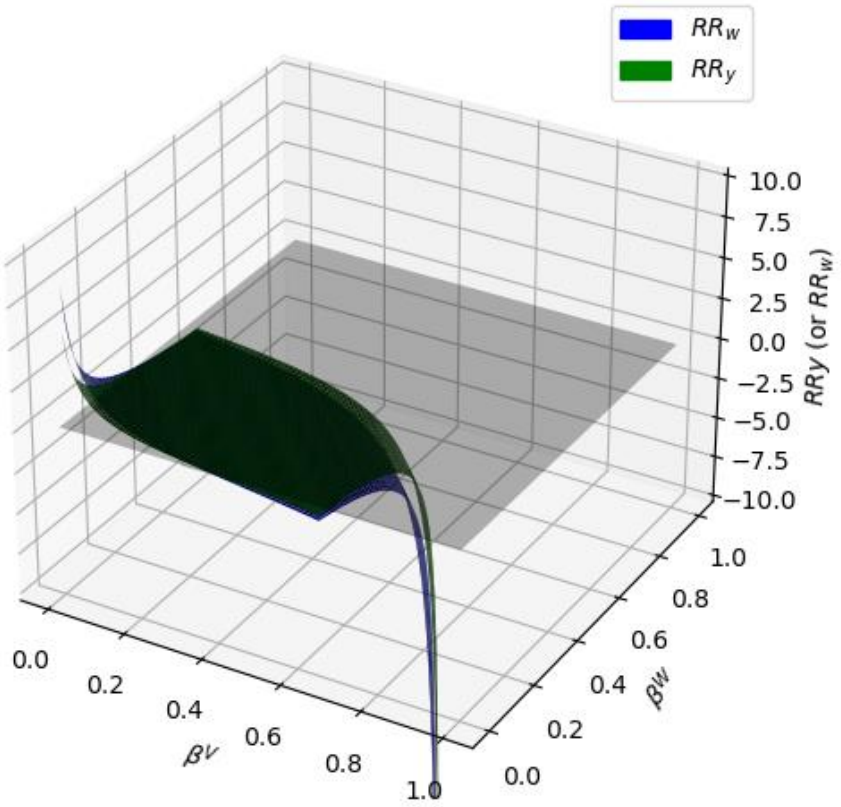


Figure 6-7. $\boldsymbol{RR_y}$ and $\boldsymbol{RR_w}$ surfaces intersecting the *z=0* plane for 30%mole water/n-octane at T = 110°C

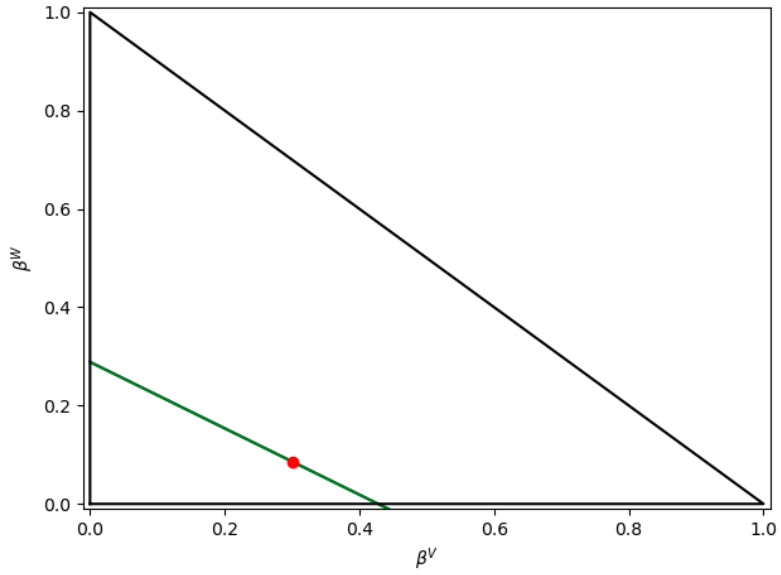


Figure 6-8. Lines for $\boldsymbol{RR_y = 0}$ and $\boldsymbol{RR_w = 0}$ at T = 110°C. Note: red point represents HYSYS solution

Haugen et al. (2011) [133] described that the non-converging lines problem leads to a very large number of iterations with the numerical solution becoming unacceptably expensive.

Thus one can conclude that the described shape of the $\mathrm{RR_y}$ and $\mathrm{RR_w}$ surfaces make it very challenging for a proposed algorithm such as the SRKKD EoS model with Python algorithm to converge towards a "Constant-K" flash solution.

6.4.2. PASN/Water Blends

In contrast with the "non-convergence" results described in Section 6.3 and 6.4.1 for n-octane/water blends, in the case of PASN/Water mixtures, the SRKKD EoS model with Python can provide instead, a consistent "constant-K" flash convergent simulations. This is the case when performing flash calculations for a 50%mole water/PASN mixture. Results after convergence are presented in Table 6.6.

Table 6-6. PASN/Water Mixture 3-phases flash calculations at T=80°C and P = 83.16 kPa using SRKKD model, the Rachford-Rice equations and the Python calculations of the present study

	Molar Flow (kgmol/h)	Molar Fraction						
		Water	n-Hexane	n-Heptane	n-Octane	n-Decane	Toluene	n-Dodecane
Feed (Water)	50.00	1	0	0	0	0	0	0
Feed (PASN)	50.00	0	0.11939808	0.2259054	0.540451	$4.34*10^{-2}$	$5.88*10^{-2}$	$1.21E*10^{-2}$
Hydrocarbon Phase	45.51	$5.48*10^{-3}$	0.0967	0.2171	0.5623	$4.75*10^{-2}$	$5.75*10^{-2}$	$1.33*10^{-2}$
Aqueous Phase	44.49	≈1.00	$5.80E*10^{-17}$	$5.11*10^{-19}$	$2.74*10^{-21}$	$8.65*10^{-28}$	$2.70*10^{-12}$	$1.47*10^{-34}$
Vapor	10.00	0.4951	0.1599	0.1481	0.1604	$2.47*10^{-3}$	$3.37*10^{-2}$	$1.32*10^{-4}$

Figure 6-9 reports the calculated RR_y and RR_w using K_i^V and K_i^W from the SRKKD model and the Python calculations. In this case, values of β^V and β^W are smaller than 0.5, providing in all cases converging numerical solutions.

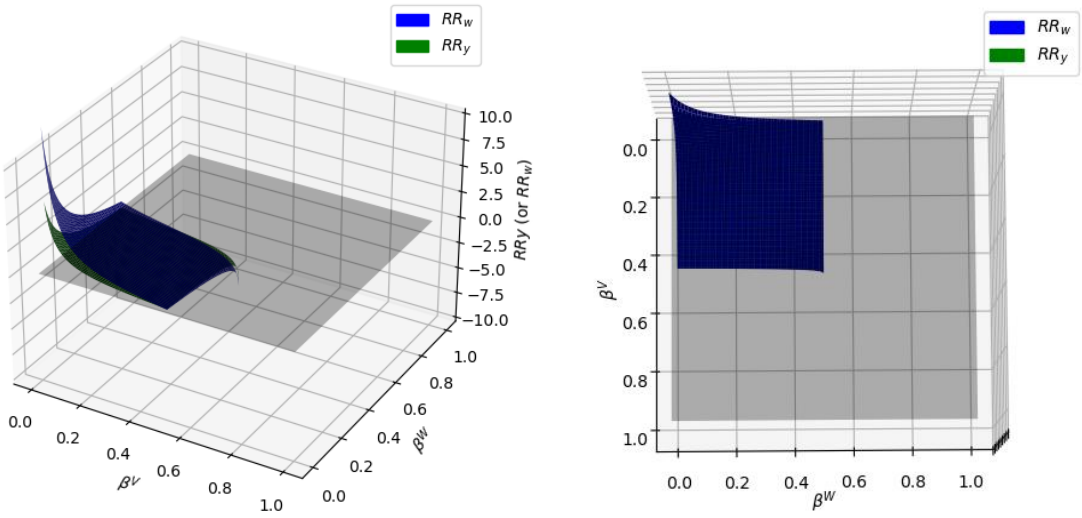


Figure 6-9. $\boldsymbol{RR_y}$ and $\boldsymbol{RR_w}$ surfaces intersecting the *z=0* plane for 50 mol% water/PASN at T = 80°C a) 3D surfaces, b) Top view

Figure 6-10 further describes the "constant-K" flash solution for the $RR_y = 0$ and $RR_w = 0$ lines displaying the numerical β^V and β^W solutions with convergence being assured.

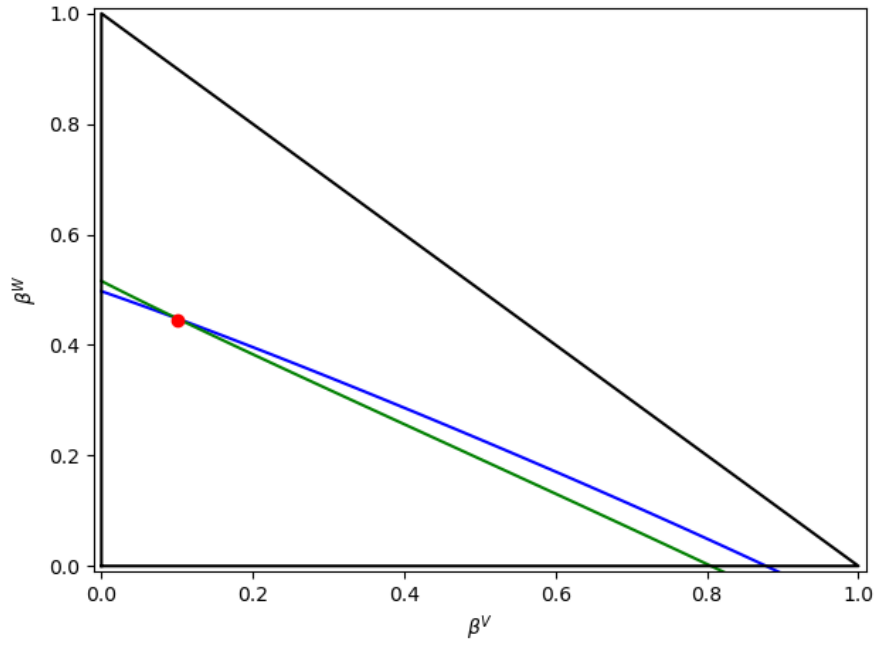


Figure 6-10. Lines for $\boldsymbol{RR_y} = 0$ and $\boldsymbol{RR_w} = 0$. Note: red point represents Python solution, blue line is related to $\boldsymbol{RR_w} = 0$ and green line $\boldsymbol{RR_y} = 0$

6.4.3. Boundary conditions

Regarding the boundary conditions, if one considers the water/n-octane mixture with K_i approximated by HYSYS V9 (Figure 6-8), the boundary conditions proposed by Okuno et al. [130] result in the Equations (6-38) and (6-39) with the hyperplanes being superimposed to the lines for $\boldsymbol{RR_y} = 0$ and $\boldsymbol{RR_w} = 0$ as presented in Figure 6-11.

$$-108.2527\beta^V - 163.2577\beta^W \leq -81.1289 \tag{6-38}$$

$$0.6631\beta^V + \beta^W \leq 0.5 \tag{6-39}$$

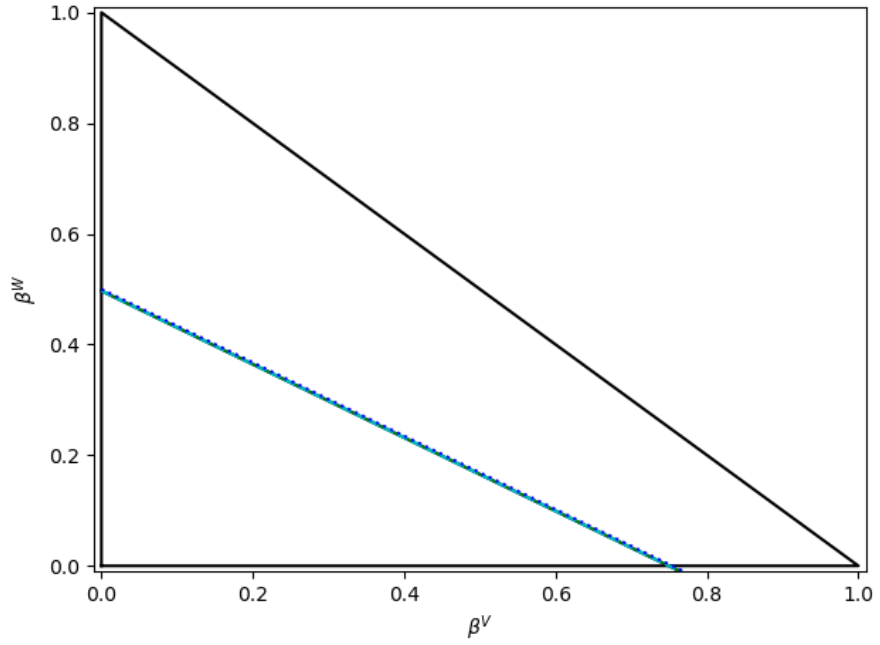


Figure 6-11. Boundaries for β^V and β^W as poposed by Okuno et al. [130] for water/n-Octane mixtures. Notes: a) Reported lines are for $\boldsymbol{RR_y} = 0$ and $\boldsymbol{RR_w} = 0$, b) The two superimposed blue lines encompass as well both Eq(6.35) and (6.36) lines

Furthermore, while applying the boundary conditions proposed by Okuno et al. [130] for the water/PASN blend, the hyperplanes related to Equations (6-40) to (6-46) can be represented as in Figure 6-12:

$$-89.43\ \beta^V - 181.62\ \beta^W \leq -90.31 \tag{6-40}$$

$$-0.6535\ \beta^V + \beta^W \leq 0.9013 \tag{6-41}$$

$$0.3178\,\beta^V + \beta^W \leq 0.8870 \tag{6-42}$$

$$0.7147\,\beta^V + \beta^W \leq 0.7297 \tag{6-43}$$

$$0.9479\,\beta^V + \beta^W \leq 0.9783 \tag{6-44}$$

$$0.9901\,\beta^V + \beta^W \leq 0.9940 \tag{6-45}$$

$$0.4134\beta^V + \beta^W \leq 0.9706 \tag{6-46}$$

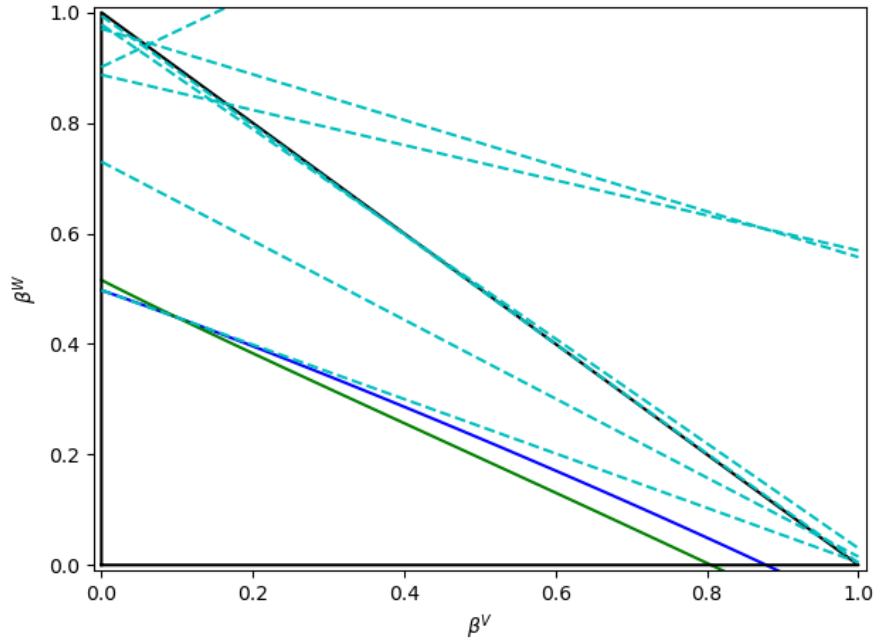


Figure 6-12. Boundaries for β^V and β^W as poposed by Okuno et al. [130] for water/PASN mixtures. Notes: a) the blue solid lines are related to $\boldsymbol{RR_w} = 0$, b) the green solid lines to $\boldsymbol{RR_y} = 0$, and c) the cyan broken lines represent the boundaries according to Equations (6.36) to (6.42)

6.4.4. Remarks

On the basis of results from "constant-K" flash calculations for n-octane/water and PASN/water blends, one can conclude that the composition of hydrocarbon/water blends is a key factor in allowing a viable numerical calculation using the Rachford-Rice equations. Thus, and to address possible calculation uncertainty and ambiguity resulting for some class of hydrocarbon/water blends, a alternate methodology to calculate the

molar fractions and mixture pressure for the water/n-octane mixture is proposed in the upcoming sections.

6.5. Liquid phase K-values from experimental data for water/n-Octane mixtures

As discussed in the previous sections, the initial guess for K-values affects the convergence of the flash calculation algorithm (Figure 6-1). Usually, Wilson correlation [139] (Equation (6.47)) is used as a first approximation for the K-value in the hydrocarbon phase. For the K-values in the water phase, Connolly et al. [10] suggested values based on the initial feed as presented in Equation (6-48).

$$\ln K_i^{Wilson} = \ln \frac{P_i^C}{P} + 5.373(1-\omega_i)\left(1-\frac{T_i^C}{T}\right) \tag{6-47}$$

$$K_i^{H_2O} = \frac{0.999}{z_i} \tag{6-48}$$

In this sense, another advantage of the CREC VL Cell developed by CREC researchers is that it allows one to determine the solubility values based on the VLLE data obtained. The transition from the three-phases and two-phases regions defines the solubility limit. Figure 6-13 presents the solubility limit regions for a water/n-octane mixture at 110°C. This region is characterized by the transition from three phases to two phases.

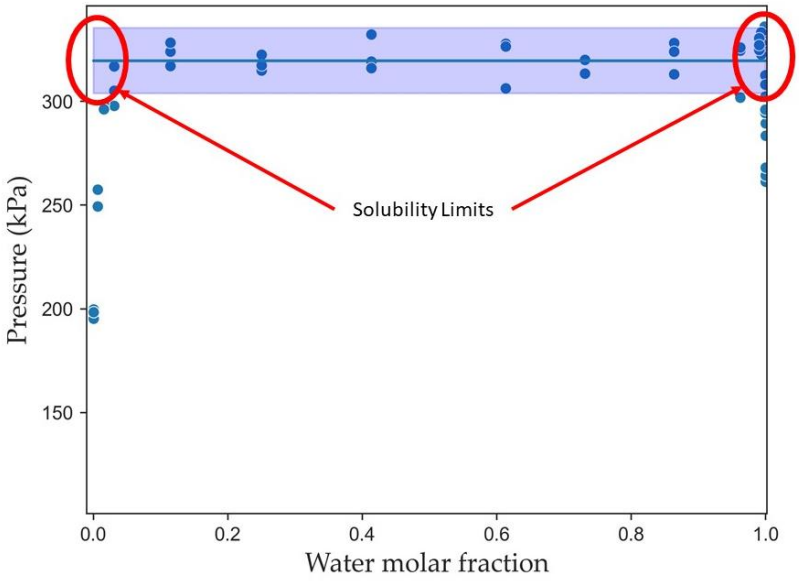


Figure 6-13. Solubility limits regions for water/Octane at 110°C

One should note that the applicability of the CREC-VL Cell for establishing solubility and solubility limits is not restricted to any type of hydrocarbon/water blend. Thus, the CREC-VL Cell can be of special value while dealing with hydrocarbon/water blends involving intrinsic convergence uncertainties, observed by octane-water mixtures, while using the Rachford-Rice equations (refer to Section 6.4).

Figure 6-14 shows that the solubility limit for an octane/water blend can be calculated from the intersection between the lines that define the three-phases and two-phases domain. The calculated solubilities at 110°C are $x_w^L = 0.01810534 \pm 2.5 * 10^{-3}$ and $x_{oct}^W = 0.00084875 \pm 2.6 * 10^{-4}$ considering the 95% confidence interval defined by the blue region. One should note that while the solubility of water in n-Octane is in close agreement with the $x_w^L = 0.016$, calculated value using HYSYS V9 with SRKKD EoS and Rachford-Rice equations, yields a solubility for n-Octane in water two orders of magnitude higher compared with the $x_{oct}^W = 2.06 * 10^{-6}$ Hysis V9 predicted value.

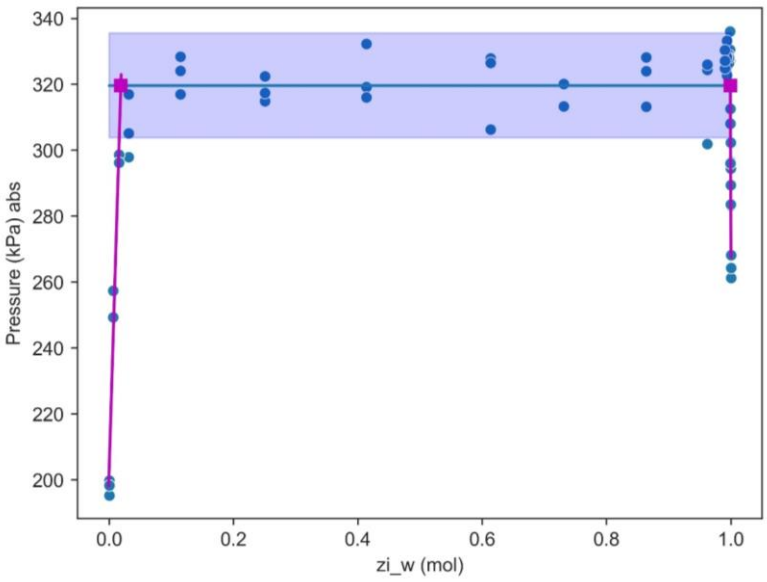


Figure 6-14. Calculated Solubility Limit for water/Octane at 110°C Note: Magenta solid lines represent the two phase region, while the blue solid line the 3 phase region. Blue bands represent the 95% confidence interval

As a result and once the solubility values are obtained from the CREC-VL Cell, it is possible to calculate $K_i^W = \frac{x_i^W}{x_i^L}$ water phase constants being $K_i^W = [55.185 \quad 8.64 * 10^{-4}]$. Furthermore and for the vapour phase species at equilibrium, one can obtain an approximative value using $y_i = \frac{\frac{x_i^L \phi_i^L}{\phi_i^V}}{\sum_i^N \frac{x_i^L \phi_i^L}{\phi_i^V}}$ and $K_i^V = \frac{y_i}{x_i^L}$. Assuming vapour phase behaves as an ideal gas, then $\phi_i^V \approx 1$ and the obtained results are: $y_w \approx 0.7347 \pm 2.38 * 10^{-2}$ and $K_i^V = [40.5772 \quad 0.2702]$.

One should note that the mean value obtained has a 16.04% difference from the value predicted by HYSYS V9 ($y_w \approx 0.6799$); with; however, the HYSYS V9 value being within the range of 95% interval confidence. Figure 6-15 and Figure 6-16 reports the observed solubilities of water in n-octane and n-octane in water, respectively, as determined in the CREC VL Cell and compared with the Maczynski et al. (2004) [41] values for both water in n-octane and n-octane in water.

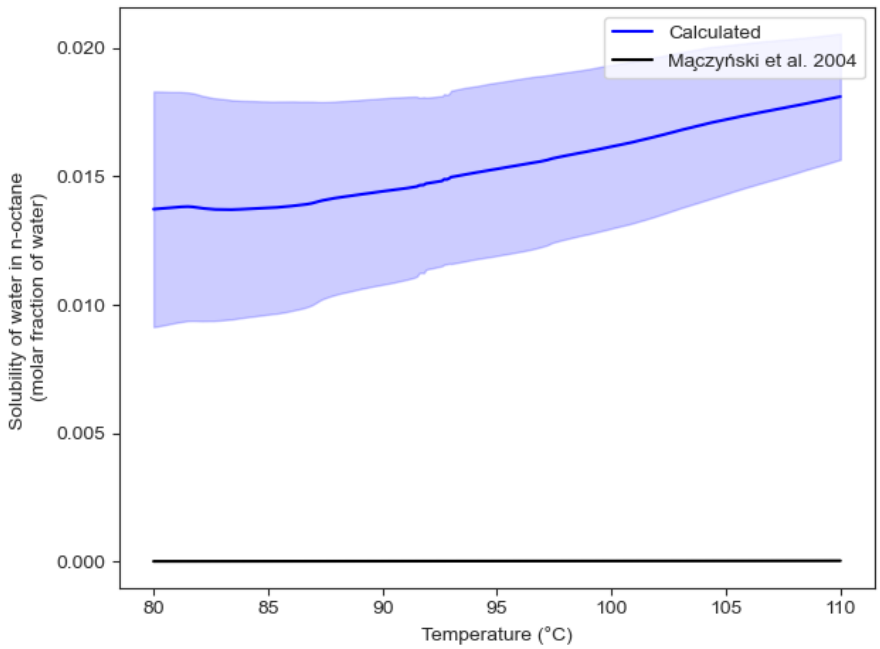


Figure 6-15. Solubility of water in n-octane in the temperature range of interest. Note: Blue bands represent the 95% confidence interval

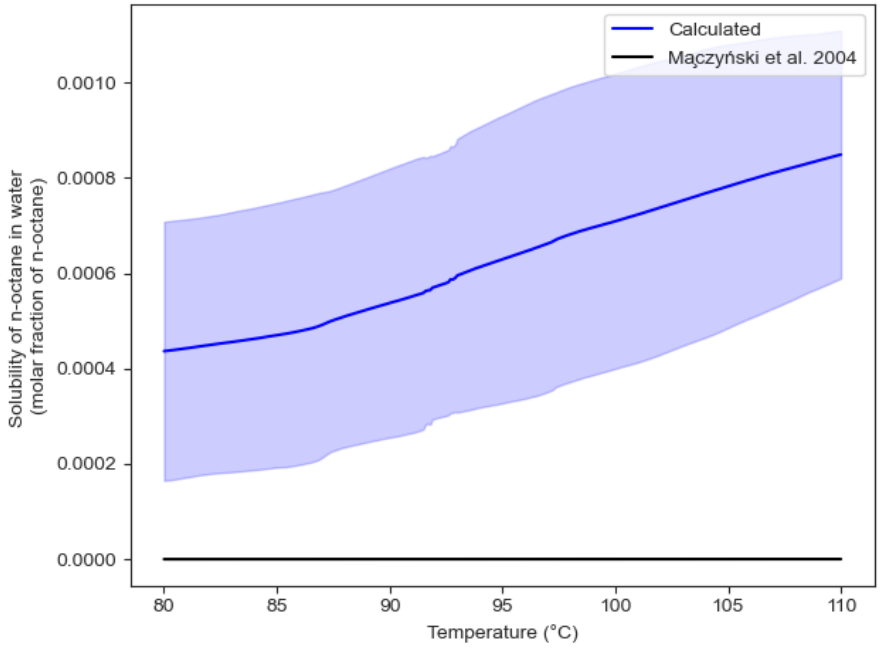


Figure 6-16. The solubility of n-octane in water in the temperature range of interest. Note: Blue bands represent the 95% confidence interval

Given the observed differences with the experimental points reported in Figure 6-3 and obtained in the CREC-VL Cell, it is important to provide a more precise methodology to calculate the VLLE values accounting for the uncertainties related to the experimental values, as will be discussed in Section 6.6.

Furthermore, and on the basis of the experimental data obtained in the CREC-VL Cell a correlation to obtain the K-values for water/octane mixtures in the temperature range of interest and low pressure (1-3 atm) is proposed. This correlation is given by the following Equation (6-49), with the constants involved reported in Table 6-7. A comparison with calculated values from experimental results is also reported in Figure 6-17, showing the adequate fitting of the experimental values by this correlation.

$$\ln K_i^p = m \ln T + b \tag{6-49}$$

Table 6-7. Constants for K-values correlation water/n-octane system

	Compound	m	b	R^2
K_i^W	Water	1.3366	-9.3022	0.9786
K_i^W	Octane	0.01390	-0.0483	0.9563
K_i^V	Water	-1.3113	9.8849	0.9851
K_i^V	Octane	1.2099	-6.9870	0.9910

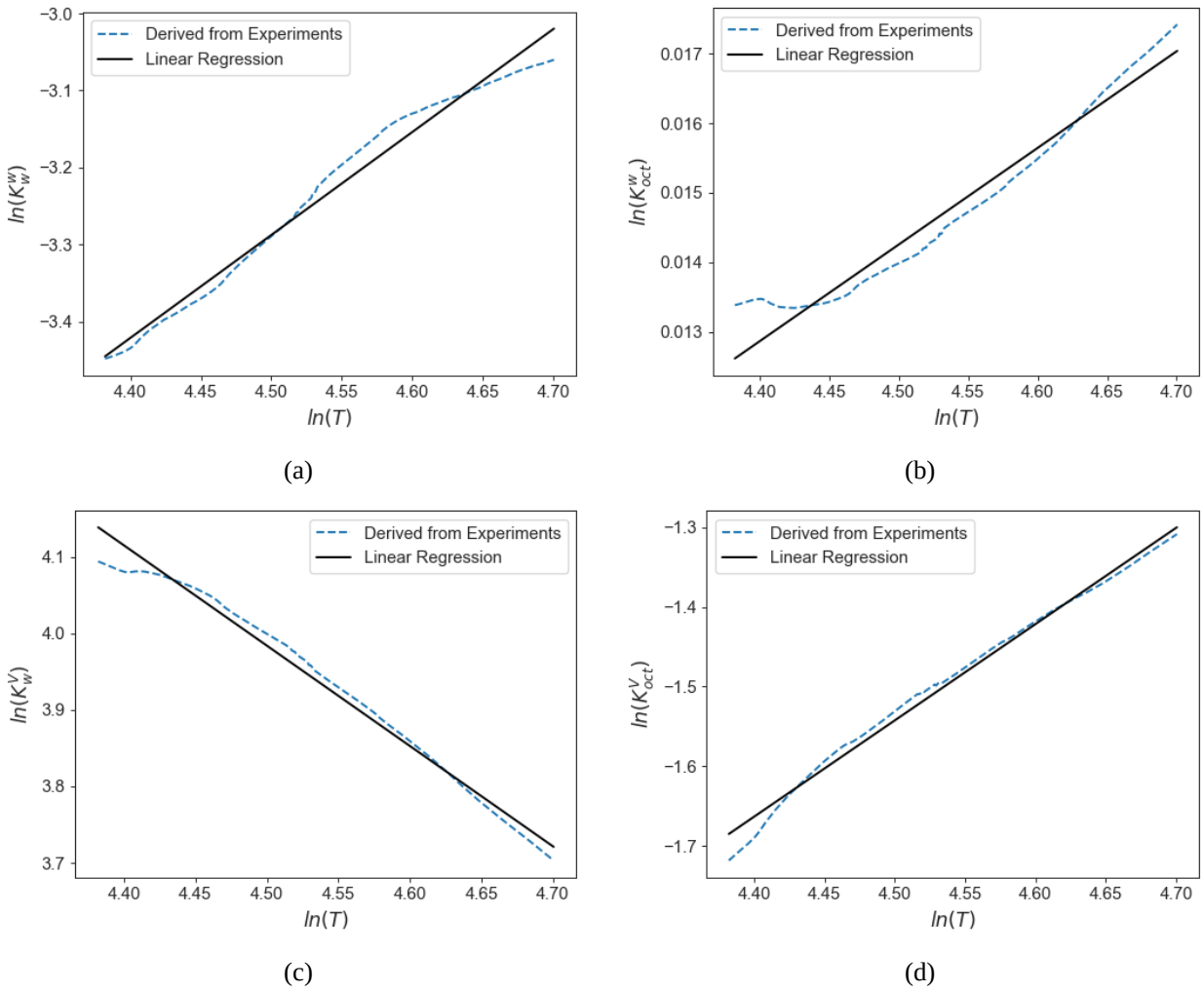


Figure 6-17. K-values calculated from experimental in the CREC VL Cell for water/n-octane mixtures: a) K-value for water in W phase, b) K-value for octane in W phase, c) K-value for water in V phase, d) K-value for octane in V phase

6.6. Machine Learning Approach

As shown in the previous sections of this chapter (Sections 6.1 to 6.5), traditional thermodynamic models for multiphase systems based on Rachford-Rice equations over-predict vapour pressure being in cases unable to provide converging and meaningful solutions.

Thus, and to address this a Machine Learning approach is considered proposed using linear regression, KNN, SVM and Decision Tree Regressor (DTR). To establish the prediction errors, coefficient of determination (R^2), Mean Squared Error (MSE) and Mean Absolute Error (MAE) are determined.

In the case of KNN, SVM and DTR the parameters were tunned using a grid search with cross-validation (GridSearchCV). This method allows finding the best set of parameters in each model based on the coefficient of determination (R^2). The training set was selected by randomly splitting the sample into 80% for training and 20% testing (train_test_split function). Results for the grid search are summarized in Table 6-8 showing the best parameters for each method calculated by the GridSearchCV function by comparing R^2.

Table 6-8. Models Selected for the Prediction of Pressure for water/n-octane

Model #	Type	Tunned Hyper-parameters	Best parameters
1	Linear Regression	N/A	
2	KNN	n_neighbors: [2, 3, 4, 5, 10, 15, 20], weights: [uniform,distance'], algorithm: [auto, ball_tree, kd_tree, brute], leaf_size: [10, 30, 50]	n_neighbors: 10 weights: distance algorithm: ball_tree leaf_size: 30
3	SVR	kernel: [linear, poly, rbf, sigmoid], degree: [2, 3], C: [1, 10, 100, 1000], epsilon:[0.1, 0.2]	kernel: rbf degree: 2 C: 1000 epsilon: 0.2
4	DTR	max_depth: [2, 3, 5, 10], min_samples_split:[2, 5, 10], min_samples_leaf: [1, 2, 5, 10]	max_depth: 10 min_samples_split: 2 min_samples_leaf: 5

Table 6-9 reports the coefficient of determination (R^2), Mean Squared Error (MSE) and Mean Absolute Error (MAE) for the selected models (best score) for octane/water using the abundant testing dataset obtained in the CREC VL cell. Figure 6-18 describes a comparison for the pressures measured and the predicted values showing that the KNN model provides the best approximation with a R^2 value of 0.99, and the lowest MSE and MSE at the 43.78 and 4.91 parameters. KNN method is selected as the best model to predict the pressure of water/n-octane mixtures in the range of interest.

Table 6-9. Metrics for the selected models. Note: Calculated based on test dataset.

	R^2	MSE	MAE

Linear Regression	0.7723	386.58	13.66
KNN	0.9911	43.78	4.91
SVR	0.8287	286.61	10.12
DTR	0.9847	29.2887	4.22

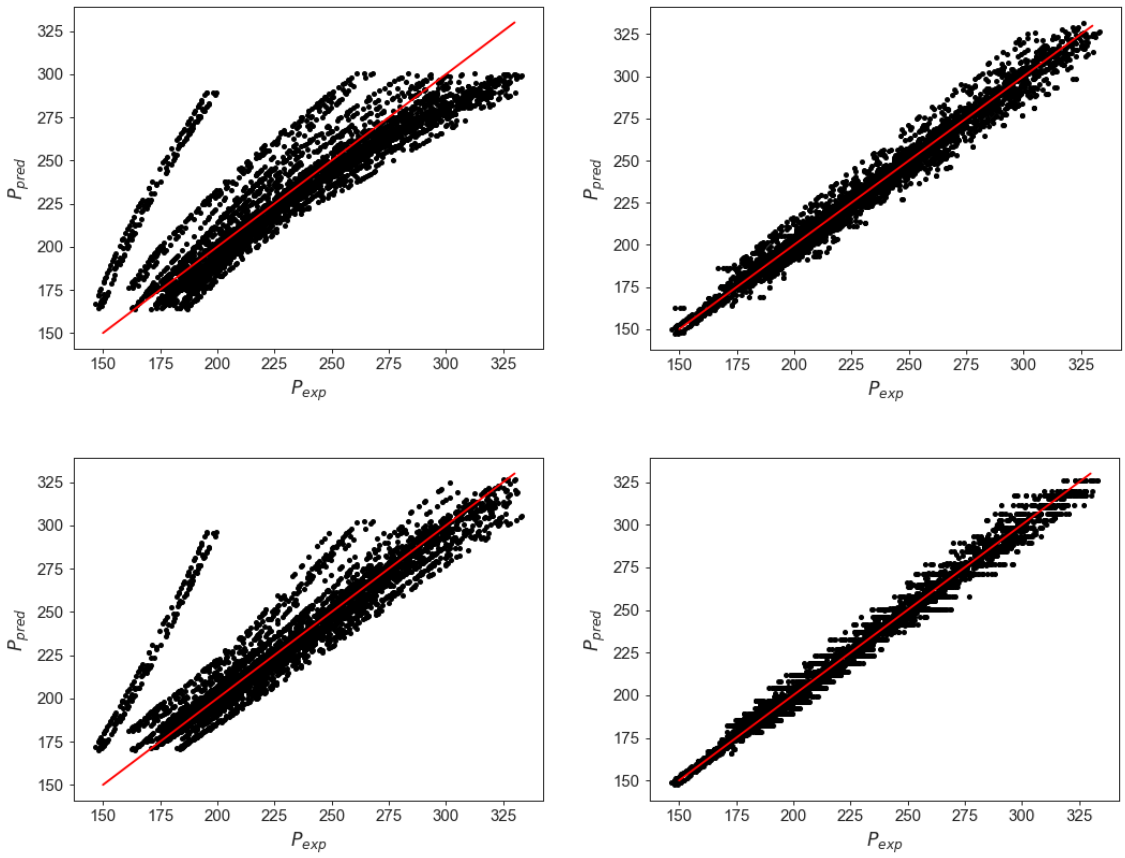


Figure 6-18. Test pressure vs. calculated for the different models a) linear regression, b) KNN, c) SVC, d) Decision Tree Regressor. Note: red line represents a perfect prediction

Thus and as a result, one can see that the best ML model to describe the behaviour of pressure for n-octane/water blends is the KNN, with this model overcoming the issues with the traditional thermodynamic models involving the Rachford-Rice equations. Furthermore, and as presented in Chapter 4, it is also possible to use ML with KNN or SVC to describe accurately the classification of the number of phases in a multiphase vapour-liquid-liquid-vapour system reducing significantly the uncertainty of any theoretically thermodynamically based algorithm.

6.7. Conclusions

a) Flash calculations for water/n-octane and water/PASN system can be described in principle, using the SRKKD EoS and Rachford-Rice equations.

b) In the case of water/n-octane blends using the SRKKD EoS and Rachford-Rice equations, it is shown they involve convergence issues. These convergence issues were clarified considering the parallelism of the RR_x and RR_y (Rachford-Rice) planes.

c) Flash calculations for naphtha/water (PASN/water) blends implemented using a Python based algorithm, showed numerical solutions free of converging issues, with however, numerical solutions showing a lack of accurate prediction of naphtha-water solubility.

d) The CREC-VL Cell data was used to propose correlations for the calculation of solubility of n-octane in water and water in n-octane, as well as the equilibrium constants values.

e) Using the developed ML approach and from a number of alternatives, the ML with KNN model showed the best performance with a $R^2 = 0.9911$, predicting the total pressure in water/n-octane mixtures accurately.

7. General Contributions and Perspectives

7.1. General Contributions

The main contribution of this thesis are summarized as follows:

- It was shown that PASN/water mixtures follow a similar thermodynamic behaviour as n-octane/water mixtures, and as a result, n-octane/water can be used as a surrogate component for naphtha/water blends.
- It was observed using phase stability analysis, based on the Gibbs Energy of Mixing, that the calculation of the number of phases involves significant intrinsic uncertainty.
- It was demonstrated on this basis, that viable determination of the number of phases in hydrocarbon/water blends, employing thermodynamic stability analysis is closely related to the hydrocarbon/water blends composition.
- It was proven that there are significant total pressure and hydrocarbon solubilities differences between the experimental data obtained in a CREC VL Cell and the VLLE results predicted using HYSYS V9 for both n-octane/water and PASN/water mixtures.
- It was also shown that Machine Learning (ML) models can be positively used to predict multiphase water/n-octane and water/PASN mixtures, without the ambiguity of stability models based on fundamentally based thermodynamic equations, such as the Rachford-Rice equations.
- It was also proven that Machine Learning (ML) models trained on big vapour-liquid-liquid data sets, obtained in a CREC-VL Cell, can be successfully employed for the prediction of total pressure and solubility both for n-octane/water and PASN/water blends, with the KNN model and the weighted SVC model, identified as the ones with the best performance.

7.2. Recommendations for Future work

The following future research is recommended on the basis of the findings of the present Ph.D. dissertation:

- To analyze in more detail, the phase stability analysis's sensitivity, based on the Gibbs Energy of Mixing and flash split including the Rachford-Rice equations to various PASN synthetic naphthas compositions in the range of potential industrial interest.
- The use of the proposed ML techniques to characterize industrial dilbit/water mixtures from the point of view of the number of phases, hydrocarbon solubility and total pressure using the CREC VL Cell.

8. Accomplishment of the Research Objectives

The accomplishments of the research objectives of the present Ph. D. dissertation are reflected in the following technical contributions:

8.1. Papers

a) **As leading author with results included in Chapters 4 and 5 of the present Ph.D. Dissertation**

- **Lopez-Zamora, S.**; Kong, J.; Escobedo, S.; de Lasa, H. Vapour-Liquid-Liquid and Vapour-Liquid Equilibrium of Paraffinic Aromatic Synthetic Naphtha/Water Blends: Prediction of The Number of Phases. **2021**. Accepted in Canadian Journal of Chemical Engineering, April 6, 2021

- **Lopez-Zamora, S.**; Kong, J.; Escobedo, S.; de Lasa, H. Thermodynamics and Machine Learning Based Approaches for Vapor-Liquid-Liquid Phase Equilibria in N-Octane/Water Blends, as a Naphtha-Water Surrogate in Water Streams. *Processes* **2021**, *9*, 1–31, https://doi.org/10.3390/pr9030413.

b) **As a co-author with results not included in the present Ph.D. Dissertation**

- Escobedo, S.; Kong, J.; **Lopez-Zamora, S.**; de Lasa, H. Synthetic Naphtha Recovery from Water Streams: Vapor-Liquid-Liquid Equilibrium (VLLE) Studies in a Dynamic VL-Cell Unit with High Intensity Mixing. *The Canadian Journal of Chemical Engineering* **2021**. https://doi.org/10.1002/cjce.24120

  My contribution to this paper was related to the data analysis and thermodynamic modelling of the blends.

- Kong, J.; Escobedo, S.; **Lopez-Zamora, S.**; de Lasa, H. Phase Equilibrium in N-Octane/Water Separation Units: Vapor Pressures, Vapor and Liquid Molar

Fractions. *International Journal of Chemical Reactor Engineering* **2021**. https://doi.org/10.1515/ijcre-2021-0031

My contribution to this paper was related to the data analysis and thermodynamic modelling of the blends.

8.2. Conferences

- **Lopez-Zamora, S.**; Kong, J.; Escobedo, S.; de Lasa, H. Thermodynamics of Hydrocarbon/Water Systems: Challenges and a Binary Interaction Parameter (BIP) Based Modelling Approach Using Experimental Data. Canadian Chemical Engineering Conference 2020. Ottawa, Canada, October 2020 (Virtual due to COVID-19)

9. Appendices

9.1. Appendix A: Gibbs Free Energy of Mixing derivation

If the binary system results in a single phase:

$$G = G^{id} + G^{E} \tag{A 1}$$

$$\frac{\bar{G}_i^E}{RT} = \ln \gamma_i \tag{A 2}$$

$$G^{id} = \sum_i x_i G_i + RT \sum_i x_i \ln x_i = x_1 G_1 + x_2 G_2 + RT(x_1 \ln x_1 + x_2 \ln x_2) \tag{A 3}$$

$$G^{E} = RT \sum_i x_i \ln \gamma_i = RT(x_1 \ln \gamma_1 + x_2 \ln \gamma_2) \tag{A 4}$$

$$\begin{aligned} G &= x_1 G_1 + x_2 G_2 + RT(x_1 \ln x_1 + x_2 \ln x_2) + RT(x_1 \ln \gamma_1 + x_2 \ln \gamma_2) \\ &= x_1 G_1 + x_2 G_2 \\ &+ RT(x_1 \ln x_1 + x_2 \ln x_2 + x_1 \ln \gamma_1 + x_2 \ln \gamma_2) \\ &= x_1 G_1 + x_2 G_2 + RT\big(x_1(\ln x_1 \ln \gamma_1) + x_2(\ln x_2 \ln \gamma_2)\big) \\ &= x_1 G_1 + x_2 G_2 + RT(x_1 \ln(x_1 \gamma_1) + x_2 \ln(x_2 \gamma_2)) \end{aligned} \tag{A 5}$$

$$\Delta G_{mix} = RT(x_1 \ln(x_1 \gamma_1) + x_2 \ln(x_2 \gamma_2)) \tag{A 6}$$

$$G = x_1 G_1 + x_2 G_2 + \Delta G_{mix} \tag{A 7}$$

If the reference state for 1 and 2 is the pure liquid, then $G_1 = G_2 = 0$

$$G = \Delta G_{mix} = RT(x_1 \ln(x_1 \gamma_1) + x_2 \ln(x_2 \gamma_2)) \tag{A 8}$$

9.2. Appendix B: Two and Three phases flash equations

9.2.1. Two phases flash equations

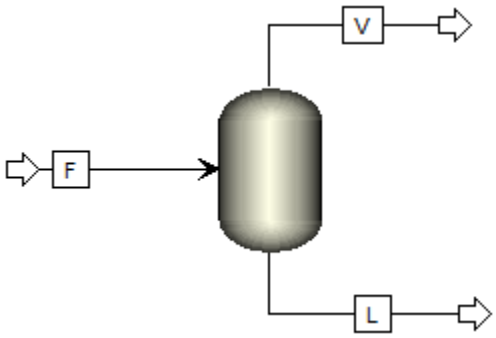


Figure 9-1. Schematic of a flash system dealing with two phases

$$F = L + V \quad \text{(B 1)}$$

$$Fz_i = Lx_i^L + Vy_i \quad \text{(B 2)}$$

$$\sum_i^n x_i^L = \sum_i^n y_i = 1 \quad \text{(B 3)}$$

From equilibrium:

$$K_i^V = \frac{y_i}{x_i^L} = \frac{\phi_i^L}{\phi_i^V} \quad \text{(B 4)}$$

$$1 = \frac{L}{F} + \frac{V}{F} \quad \text{(B 5)}$$

$$1 = \beta^L + \beta^V \quad \text{(B 6)}$$

$$\beta^L = 1 - \beta^V \quad \text{(B 7)}$$

$$z_i = \beta^L x_i^L + \beta^V y_i \quad \text{(B 8)}$$

$$z_i = (1 - \beta^V) x_i^L + \beta^V y_i \quad \text{(B 9)}$$

$$z_i = (1 - \beta^V) x_i^L + \beta^V x_i^L K_i^V \quad \text{(B 10)}$$

$$z_i = x_i^L \left((1 - \beta^V) + \beta^V K_i^V \right) \quad \text{(B 11)}$$

$$z_i = x_i^L \left(1 + \beta^V (K_i^V - 1)\right) \quad \text{(B 12)}$$

$$x_i^L = \frac{z_i}{1 + \beta^V (K_i^V - 1)} \quad \text{(B 13)}$$

$$y_i = x_i^L K_i^V = \frac{z_i K_i^V}{1 + \beta^V (K_i^V - 1)} \quad \text{(B 14)}$$

Rachford-Rice equation is given below:

$$RR_y = \sum_{i=1}^{C} (y_i - x_i^L) = \frac{z_i (K_i^V - 1)}{1 + \beta^V (K_i^V - 1)} = 0 \quad \text{(B 15)}$$

9.2.2. Three phases flash equations

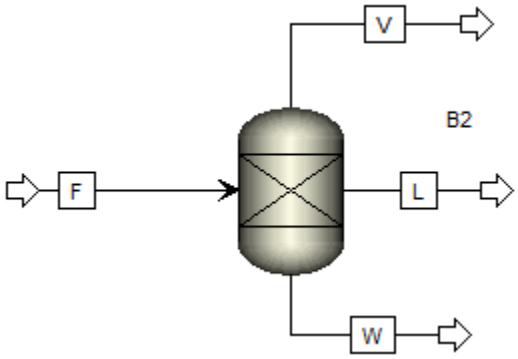


Figure 9-2. Schematic of a flash system dealing with three phases

$$F = L + W + V \quad \text{(B 16)}$$

$$F z_i = L x_i^L + W x_i^W + V y_i \quad \text{(B 17)}$$

$$\sum_i^n x_i^L = \sum_i^n x_i^W = \sum_i^n y_i = 1 \qquad \text{(B 18)}$$

From equilibrium:

$$K_i^V = \frac{y_i}{x_i^L} = \frac{\phi_i^L}{\phi_i^V} \qquad \text{(B 19)}$$

$$K_i^W = \frac{x_i^W}{x_i^L} = \frac{\phi_i^W}{\phi_i^V} \qquad \text{(B 20)}$$

$$1 = \frac{L}{F} + \frac{W}{F} + \frac{V}{F} \qquad \text{(B 21)}$$

$$1 = \beta^L + \beta^W + \beta^V \qquad \text{(B 22)}$$

$$\beta^L = 1 - \beta^W - \beta^V \qquad \text{(B 23)}$$

$$z_i = \beta^L x_i^L + \beta^W x_i^W + \beta^V y_i \qquad \text{(B 24)}$$

$$z_i = (1 - \beta^W - \beta^V) x_i^L + \beta^W x_i^W + \beta^V y_i \qquad \text{(B 25)}$$

$$z_i = (1 - \beta^W - \beta^V) x_i^L + \beta^W x_i^L K_i^W + \beta^V x_i^L K_i^V \qquad \text{(B 26)}$$

$$z_i = x_i^L \left((1 - \beta^W - \beta^V) + \beta^W K_i^W + \beta^V K_i^V \right) \qquad \text{(B 27)}$$

$$z_i = x_i^L \left(1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1) \right) \qquad \text{(B 28)}$$

$$x_i^L = \frac{z_i}{1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1)} \qquad \text{(B 29)}$$

$$x_i^W = x_i^L K_i^W = \frac{z_i K_i^W}{1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1)} \qquad \text{(B 30)}$$

$$y_i = x_i^L K_i^V = \frac{z_i K_i^V}{1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1)} \qquad \text{(B 31)}$$

Rachford-Rice equations are summarized as follows:

$$RR_y = \sum_{i=1}^{N} (y_i - x_i^L) = \sum_{i=1}^{N} \frac{z_i (K_i^V - 1)}{1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1)} = 0 \qquad \text{(B 32)}$$

$$RR_{x_i^W} = \sum_{i=1}^{N} (x_i^W - x_i^L) = \sum_{i=1}^{N} \frac{z_i (K_i^W - 1)}{1 + \beta^V (K_i^V - 1) + \beta^W (K_i^W - 1)} = 0 \qquad \text{(B 33)}$$

9.3. Appendix C: Python Code

- Activity Coefficient Models (ActivityCoefficientModels.py)

```
import numpy as np
from scipy.optimize import fsolve
from scipy.interpolate import interp1d    # For interpolation
from AntoineEquation import *

# ============================================================================
# Functions hydrocarbon/water mixtures
#
def NRTL(parametersType, T,xi, a_ij, b_ij=None, c_ij=None, d_ij=None, e_ij=None, f_ij=None, alpha_ij =
None, Rg =  None):
   """ (str, float, np.array, np.array, np.array, np.array, np.array, np.array, int, int ) -> list of int

  Returns the Gammas using NRTL model for different options:
      1. Klauck
      2. AspenPlus
```

```
    3. AspenHysys

  Units are:
    T     [K]
    xi    [Molar fraction]

  Example for Aspen Hysys: water(1)/ n-octane(2) parameters = 0
  >>> NRTL('AspenHysys', 348, np.array([0.1, 1-0.1]), np.zeros((2, 2)), np.zeros((2, 2)), '','','','', 0.2, (101325*22.414/(1000*273.15))*(1000/4184))
  array([1., 1.])

  Example for Aspen Hysys: water(1)/ n-octane(2)
  >>> NRTL('AspenHysys', 70+273.15, np.array([1e-9, 1-1e-9]), np.array([[0, 8114.72998046875], [604.739990234375, 0]]), np.array([[0, -18.2350006103516], [18.1609992980957, 0]]), '','','','', 0.2, (101325*22.414/(1000*273.15))*(1000/4184))
  array([1.09704181e+05, 1.00000000e+00])

  Example for Aspen Plus: water(1)/ n-octane(2)
  >>> NRTL('AspenPlus', 70+273.15, np.array([1e-9, 1-1e-9]), np.array([[0, 1.2166], [-12.035, 0]]), np.array([[0, 2997.7], [5381.43, 0]]), np.array([[0, 0.2], [0.2, 0]]), np.zeros((2, 2)), np.zeros((2, 2)), np.zeros((2, 2)))
  array([149.49048194,   1.        ])

  Example for Klauc: water(1)/ n-octane(2)
  >>> NRTL('Klauck', 70+273.15, np.array([1e-9, 1-1e-9]), np.array([[0, -169.718], [4197.06, 0]]), np.array([[0, 12.5591], [-7.5243, 0]]), np.zeros((2, 2)),'','','', 0.2)
  array([326.07095536,   1.        ])

  Example for Aspen Plus: water(1)/ n-octane(2) / dodecane(3)
  >>> NRTL('AspenPlus', 70+273.15, np.array([0.0203048, 0.107552, 0.872144]), np.array([[0, 1.2166, 23.4291], [-12.035, 0, 0], [-6.08871, 0, 0]]), np.array([[0, 2997.7, -2638.14], [5381.43, 0, 0], [3794.11, 0, 0]]), 0.2, np.zeros((3, 3)), np.zeros((3, 3)), np.zeros((3, 3)))
  array([156.79663005,   0.99052444,   1.00664997])

  """
  if parametersType == 'Klauck':
    A_ij = a_ij + (b_ij*T) + (c_ij*(T**2))
    tao_ij = A_ij/T
    G_ij = np.exp (-alpha_ij*tao_ij);

  elif parametersType == 'AspenPlus':
    tao_ij = a_ij + (b_ij/T) + (e_ij*np.log(T)) + (f_ij*T)
    alpha_ij=c_ij+(d_ij*(T-273.15));
    G_ij = np.exp (-alpha_ij*tao_ij);

  elif parametersType == 'AspenHysys':
    tao_ij = (a_ij + (b_ij*T) )/(Rg*T)
    G_ij = np.exp (-alpha_ij*tao_ij);


  ln_Gamma = ln_Gamma_Equation(xi, tao_ij, G_ij)
  Gamma = np.exp(ln_Gamma)
```

```
    return Gamma


def ln_Gamma_Equation(xi, tao_ij, G_ij):
    """
    Function to calculate the ln Gamma using NRTL model
    (Equations as Aspen Plus and Aspen Hysys model)
    """

    ln_Gamma = np.zeros((len(xi)))

    for ii in range(len(xi)):
        Term2_j = 0

        for jj in range(len(xi)):
            Term_k = sum(xi[:]*G_ij[:,jj])
            Term_m = sum(xi[:]*tao_ij[:, jj]*G_ij[:, jj])

            Term2_a = tao_ij[ii,jj] - (Term_m/Term_k)
            Term2_b = (xi[jj]*G_ij[ii, jj])/Term_k

            Term2_j = Term2_j + ( Term2_a * Term2_b)

        Term2_k = sum(xi[:]*G_ij[:, ii])
        Term1_j = sum(xi[:]*tao_ij[:, ii]*G_ij[:, ii])

        ln_Gamma[ii] = (Term1_j/Term2_k) + Term2_j

    return ln_Gamma



def Solubility_NRTL( x1s, parametersType,  T, a_ij, b_ij, c_ij=None, d_ij=None, e_ij=None, f_ij=None,
alpha_ij = None, Rg =  None):
    """
    This scripts uses the NRTL model proposed by ASPEN PLUS
    """

    Fobj = np.zeros(2)

    x1_I=x1s[0];
    x1_II=x1s[1];

    xi_I=[x1_I, 1-x1_I];
    Gamma_I = NRTL(parametersType, T, xi_I, a_ij, b_ij, c_ij, d_ij, e_ij, f_ij, alpha_ij, Rg)

    xi_II=[x1_II, 1-x1_II];
    Gamma_II = NRTL(parametersType, T,xi_II, a_ij, b_ij, c_ij, d_ij, e_ij, f_ij, alpha_ij, Rg)

    Fobj[0]=np.sqrt(((xi_I[0]*Gamma_I[0])-(xi_II[0]*Gamma_II[0]))**2);
    Fobj[1]=np.sqrt(((xi_I[1]*Gamma_I[1])-(xi_II[1]*Gamma_II[1]))**2);
```

```
    return Fobj

 def dGmix_NRTL(xi, Gammas):
    dGmix_RT = (xi[0]*np.log(Gammas[0]*xi[0])) + (xi[1]*np.log(Gammas[1]*xi[1]))

    return dGmix_RT
```

- Soave Redlich Kwong Equation of State (SRK_EoS.py)

```
import numpy as np
import math
from scipy.optimize import fsolve

# Soave Redlich Kwong equation of state
def  Roots_SRK(ai, bi, P, T, Rg):
   """
   Returns the roots for Soave Redlich Kwong
   Fundamentals of Chemical Engineering Thermodynamics with applications to
   Chemical Processes - Themis Matsoukas - 2013
   """

   # Parameters A' and B' definded by 2.3 for mixture
   AA = ai*P/((Rg*T)**2);
   BB = bi*P/(Rg*T);

   # SRK in terms of Z - Eq. 2.44
   CC = np.array([1, -1, (AA-BB-(BB**2)), -AA*BB ])[np.newaxis]

   r = np.array([np.roots(item) for item in CC])
   r = r[np.isreal(r)]
   r = r.real
   r = r[r >= 0]

   Roots_Z_SRK=r;
   return r, CC, AA, BB

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def Parameters_SRK(Tc, Pc, ww, Rg, Tr):
   """
      Parameters for SRK Equation - Ec. 2.41 to 2.43
      Fundamentals of Chemical Engineering Thermodynamics with applications to
      Chemical Processes - Themis Matsoukas - 2013

   """
   omega_SRK = 0.480+(1.574*ww)-(0.176*(ww**2));  # Matsoukas
   ai_SRK = (0.42748*(Rg**2)*(Tc**2)/Pc)*((1+(omega_SRK*(1-(Tr**(1/2)))))**2); # Matsoukas
   bi_SRK = 0.08664*Rg*Tc/Pc;

   return ai_SRK, bi_SRK
```

```
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def  a_b_mixture_SRK(xi, ai, bi, kij):
   """
   This function calculates the parameters for mixtures - Eq. 9.38 and 9.39
   Fundamentals of Chemical Engineering Thermodynamics with applications to
   Chemical Processes - Themis Matsoukas - 2013
   """
   b_mix = sum(xi*bi);
   a_ij = np.zeros((len(xi), len(xi)))
   ais = np.zeros(len(xi))

   for ii in range(len(xi)):
      for jj in range(len(xi)):
         a_ij[ii,jj]=xi[ii]*xi[jj]*(1-kij[ii,jj])*math.sqrt(ai[ii]*ai[jj]);
      ais[ii]=sum(a_ij[ii,:]);
   a_mix=sum(ais);

   return a_mix, b_mix

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def  a_b_mixture_SRKKD(xi, ai, bi, kij, Gi, T, Tcw):
   """
   This function calculates the parameters for mixtures - Eq. 9.38 and 9.39
   Fundamentals of Chemical Engineering Thermodynamics with applications to
   Chemical Processes - Themis Matsoukas - 2013

   [1] V.N. Kabadi, R.P. Danner, A Modified Soave-Redlich-Kwong Equation of State
   for Water-Hydrocarbon Phase Equilibria, Ind. Eng. Chem. Process Des. Dev.
   24 (1985) 537–541. https://doi.org/10.1021/i200030a004.
   """

   b_mix = sum(xi*bi);
   a_ij = np.zeros((len(xi), len(xi)))
   ais = np.zeros(len(xi))

   for ii in range(len(xi)):
      for jj in range(len(xi)):
         a_ij[ii,jj]=xi[ii]*xi[jj]*(1-kij[ii,jj])*math.sqrt(ai[ii]*ai[jj]);
      ais[ii]=sum(a_ij[ii,:]);

   a_mix=sum(ais);
   a_wi = Gi*(1-((T/Tcw)**0.8))

   xw = xi[0]     # water mole fraction
   a_KD = sum(a_wi*(xw**2)*xi)
   a_mix = a_mix + a_KD

   return a_mix, b_mix

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def FugacityCoeff_SRK(xs, ai, bi, amix, bmix, AA, BB, Z, kij = None ):
```

```
    """
    Fundamentals of Chemical Engineering Thermodynamics with applications to
    Chemical Processes - Themis Matsoukas - 2013
    Fugacity coefficient from Ec. 10.21
    """

    # Initial zeros matrix and vector
    SUMA1 = np.zeros(len(xs))
    try:
        CCi = np.zeros((len(xs), (len(AA[0]))))
    except:
        CCi = np.zeros(len(xs))

    ln_Fug_i = np.zeros(len(xs))
    Fug_i = np.zeros(len(xs))


    for ii in range(len(xs)):
        for jj in range(len(xs)):
            try:
                SUMA1[jj] = xs[jj]*np.sqrt(ai[ii]*ai[jj])*(1-kij[ii, jj])
            except:
                SUMA1[jj] = xs[jj]*np.sqrt(ai[ii]*ai[jj])

        suma = SUMA1.sum()
        CCi[ii] = (AA/BB) * (-(bi[ii]/bmix)+((2/amix)*suma))

        PartC = CCi[ii]*np.log((Z+BB)/Z)
        sandra = ((bi[ii]/bmix)*(Z-1)) - (np.log(Z-BB)) - PartC
        ln_Fug_i[ii] = sandra
        Fug_i[ii] = np.exp(ln_Fug_i[ii])

    return Fug_i, ln_Fug_i

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def Solubility_SRK_ASN_atPsup( x_sup, Psup, T, Tc, Pc, ww, kij, Rg, molFrac_ASN=None, KD = False, Gi =
None):
    """
    This scripts uses the SRK EoS to calculate solubility at a given temperature
    The order should be water as first element because Tcw = Tc[0]

    x_sup  --> Initial stimated molar fractions for solubility calculations
    Psup   --> Pressure [Pa]
    T      --> Temperature [K]
    Tc     --> Critical Temperatures [K]
    Pc     --> Critical Pressures [Pa]
    ww     --> accentric factor
    kij    --> Binary interaction parameters
    Rg     --> Universal gas constant [Pa*m2/ (mol*K)]
    """

    Fobj = np.zeros(14)
```

```
# Initial stimates
x1_I = x_sup[:7]
print(sum(x1_I))
x1_II = x_sup[7:]
print(sum(x1_II))

x1_I = x1_I/sum(x1_I)
print(x1_I)

# Calculating Peng Robinson Parameters per component
Tr=T/Tc
[ai, bi] = Parameters_SRK(Tc, Pc, ww, Rg, Tr)

# Calculations for the two phases
# Phase I
xi_I = x1_I #+ [1-sum(x1_I)]

if KD == True:
   Tcw = Tc[0]
   a_mix_I, b_mix_I = a_b_mixture_SRKKD(xi_I, ai, bi, kij, Gi, T, Tcw)
else:
   a_mix_I, b_mix_I = a_b_mixture_SRK(xi_I, ai, bi, kij)

r_I, CC_I, AA_I, BB_I = Roots_SRK(a_mix_I, b_mix_I, Psup, T, Rg)
Z_I = min(r_I)
Fug_i_I, ln_Fug_i_I = FugacityCoeff_SRK(xi_I, ai, bi, a_mix_I, b_mix_I, AA_I, BB_I, Z_I, kij  )

# Phase II
xi_II = x1_II #+ [1-sum(x1_II)]

if KD == True:
   Tcw = Tc[0]
   a_mix_II, b_mix_II = a_b_mixture_SRKKD(xi_II, ai, bi, kij, Gi, T, Tcw)
else:
   a_mix_II, b_mix_II = a_b_mixture_SRK(xi_II, ai, bi, kij)

r_II, CC_II, AA_II, BB_II = Roots_SRK(a_mix_II, b_mix_II, Psup, T, Rg)
Z_II = min(r_II)
Fug_i_II, ln_Fug_i_II = FugacityCoeff_SRK(xi_II, ai, bi, a_mix_II, b_mix_II, AA_II, BB_II, Z_II, kij  )

Fobj[0]=np.sqrt(((xi_I[0]*Fug_i_I[0])-(xi_II[0]*Fug_i_II[0]))**2);
Fobj[1]=np.sqrt(((xi_I[1]*Fug_i_I[1])-(xi_II[1]*Fug_i_II[1]))**2);
Fobj[2]=np.sqrt(((xi_I[2]*Fug_i_I[2])-(xi_II[2]*Fug_i_II[2]))**2);
Fobj[3]=np.sqrt(((xi_I[3]*Fug_i_I[3])-(xi_II[3]*Fug_i_II[3]))**2);
Fobj[4]=np.sqrt(((xi_I[4]*Fug_i_I[4])-(xi_II[4]*Fug_i_II[4]))**2);
Fobj[5]=np.sqrt(((xi_I[5]*Fug_i_I[5])-(xi_II[5]*Fug_i_II[5]))**2);
Fobj[6]=np.sqrt(((xi_I[6]*Fug_i_I[6])-(xi_II[6]*Fug_i_II[6]))**2);

Fobj[7]= np.sqrt( (1 - np.sum(xi_I))**2)
Fobj[8]= np.sqrt( (1 - np.sum(xi_II))**2 )
```

```
    Fobj[9] = sum(Fobj[:9])

    Fobj[10] = Fobj[0]
    Fobj[11] = Fobj[2]
    Fobj[12] = Fobj[4]
    Fobj[13] = Fobj[6]

    return Fobj

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def VLLE_funct_SRK(Psup, T, Pair, xi_sup, Tc, Pc, ww, kij, Rg, KD = False, Gi = None, iter_max=10000, tol=1e-
9, printSteps=False):
    """
    This scripts uses the PR EoS
    This function calculates the Pressure and solubilities based on fsolve for
    solubilities at a given P

    P_sup  --> Initial stimated pressure [Pa]
    T      --> Temperature [K]
    Pair   --> Initial pressure of air in the cell [Pa]
    ww     --> accentric factor
    kij    --> Binary interaction parameters
    Rg     --> Universal gas constant [Pa*m3/ (mol*K)]
    """
    Fobj_P = 1000
    iter_P = 0

    # Peng Robinson Parameters
    Tr=T/Tc
    [ai, bi] = Parameters_SRK(Tc, Pc, ww, Rg, Tr)

    # air correction
    factor_air = (Pair*T/(20+273.15))

    while (Fobj_P > tol) and (iter_P <= iter_max):
        iter_P += 1

        # Solubility calculation at a given P sup
        if KD == True:
            Tcw = Tc[0]
            x0_calc, info, ier, mesg  = fsolve(Solubility_SRK_atPsup, xi_sup, (Psup, T,  Tc, Pc, ww, kij, Rg, KD, Gi),
xtol = 1e-9, full_output = True)
        else:
            x0_calc, info, ier, mesg  = fsolve(Solubility_SRK_atPsup, xi_sup, (Psup, T,  Tc, Pc, ww, kij, Rg), xtol =
tol, full_output = True)

        xw_1 = x0_calc[0]
        xw_2 = x0_calc[1]

        xi_I = [xw_1, 1-xw_1]
        xi_II = [xw_2, 1-xw_2]
```

```
        # Calculate fugacity coefficient for phase 1
        if KD == True:
            Tcw = Tc[0]
            a_mix_I, b_mix_I = a_b_mixture_SRKKD(xi_I, ai, bi, kij, Gi, T, Tcw)
        else:
            a_mix_I, b_mix_I = a_b_mixture_SRK(xi_I, ai, bi, kij)

        r_I, CC_I, AA_I, BB_I = Roots_SRK(a_mix_I, b_mix_I, Psup, T, Rg)
        Z_I = min(r_I)
        Fug_i_I, ln_Fug_i_I = FugacityCoeff_SRK(xi_I, ai, bi, a_mix_I, b_mix_I, AA_I, BB_I, Z_I, kij )

        # Calculate fugacity coefficient for phase II
        if KD == True:
            Tcw = Tc[0]
            a_mix_II, b_mix_II = a_b_mixture_SRKKD(xi_II, ai, bi, kij, Gi, T, Tcw)
        else:
            a_mix_II, b_mix_II = a_b_mixture_SRK(xi_II, ai, bi, kij)

        r_II, CC_II, AA_II, BB_II = Roots_SRK(a_mix_II, b_mix_II, Psup, T, Rg)
        Z_II = min(r_II)
        Fug_i_II, ln_Fug_i_II = FugacityCoeff_SRK(xi_II, ai, bi, a_mix_II, b_mix_II, AA_II, BB_II, Z_II, kij )

        # Parcial pressures
        Pi_w = (xi_I[0]*Fug_i_I[0]*Psup)
        Pi_o = (xi_II[1]*Fug_i_II[1]*Psup)
        Pi_tot_calc = Pi_w + Pi_o + factor_air

        if printSteps:
            print(f"{iter_P}) Fobj = {Fobj_P} --> Psup = {Psup}, Pcalc={Pi_tot_calc} xi = {x0_calc}")

        Fobj_P = np.sqrt((Psup - Pi_tot_calc)**2)

        Psup = Pi_tot_calc
        xi_sup = x0_calc

    if iter_P >= iter_max:
        print("Maximum number of iterations was reached (VLLE_funct_PR)")

    return Psup, x0_calc, iter_P, Fobj_P

# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
def P_bubble_SRKKD_3P(var_sup, zi, T, Pair, Tc, Pc, ww, Rg, kij, KD, Gi):
    Psup = var_sup[0]
    print(f"Psup = {Psup}")
    betaL_sup = var_sup[1]
    print(f"betaL = {betaL_sup} ")
    xiL_sup = var_sup[2:]
    print(f"xiL_sup = {xiL_sup} ")

    xiL_sup = np.array(xiL_sup)
    print(f"sum xiL = {sum(xiL_sup)}")
```

```
nn = len(xiL_sup)
Fobj = np.zeros(nn + 2)

# Calculate compositions for aqueous phase
xiW = (zi - (xiL_sup*betaL_sup))/(1-betaL_sup)

# Calculate fugacity coefficients
# Calculating Peng Robinson Parameters per component
Tr=T/Tc
[ai, bi] = Parameters_SRK(Tc, Pc, ww, Rg, Tr)

# Calculations for the two phases
# Phase I: Hydrocarbon Phase

xi_I = xiL_sup
if KD == True:
   Tcw = Tc[0]
   a_mix_I, b_mix_I = a_b_mixture_SRKKD(xi_I, ai, bi, kij, Gi, T, Tcw)
else:
   a_mix_I, b_mix_I = a_b_mixture_SRK(xi_I, ai, bi, kij)


r_I, CC_I, AA_I, BB_I = Roots_SRK(a_mix_I, b_mix_I, Psup, T, Rg)
Z_I = min(r_I)
Fug_i_I, ln_Fug_i_I = FugacityCoeff_SRK(xi_I, ai, bi, a_mix_I, b_mix_I, AA_I, BB_I, Z_I, kij  )

# Phase II: Aquous phase
xi_II = xiW

if KD == True:
   Tcw = Tc[0]
   a_mix_II, b_mix_II = a_b_mixture_SRKKD(xi_II, ai, bi, kij, Gi, T, Tcw)
else:
   a_mix_II, b_mix_II = a_b_mixture_SRK(xi_II, ai, bi, kij)

r_II, CC_II, AA_II, BB_II = Roots_SRK(a_mix_II, b_mix_II, Psup, T, Rg)
Z_II = min(r_II)
Fug_i_II, ln_Fug_i_II = FugacityCoeff_SRK(xi_II, ai, bi, a_mix_II, b_mix_II, AA_II, BB_II, Z_II, kij  )

Fobj[:nn] = np.sqrt(((xi_I*Fug_i_I)-(xi_II*Fug_i_II))**2)

f_I = xi_I*Fug_i_I*Psup
f_II = xi_II*Fug_i_II*Psup

Pcalc1 = f_I.sum() + Pair
Pcalc2 = f_II.sum() + Pair

#Pcalc = (Pcalc1 + Pcalc2)/2
Fobj[(nn+1) - 1] = np.sqrt((Psup - Pcalc1)**2)
Fobj[(nn+2) - 1] = np.sqrt((Psup - Pcalc2)**2)

return Fobj.sum()
```

- Flash Calculatioins (FlashCalculations_SRKKD.py)

```
try:
   from Modules.SRK_EoS import *
except:
   from SRK_EoS import *

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

from scipy.optimize import fsolve
from scipy.optimize import minimize
from scipy.optimize import Bounds
from scipy.optimize import LinearConstraint

"""
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Li Firoozabadi
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
"""
def Fobj_RR_ii(beta_W, beta_V, zi, Ki, ii):
   beta_k = np.asarray([beta_V , beta_W])
   beta_k = beta_k[:, np.newaxis]
   RRs = flash_3P_RRs(zi, Ki, beta_k)
   return RRs[ii]

def Figure_liFiroozabadi_3d(zi, Kis, betaVs = [0, 1 ], betaWs = [0, 1], Z_lim = []):
   Z_lim = np.array([Z_lim])
   betas_V = np.linspace(betaVs[0], betaVs[1], 100)
   betas_W = np.linspace(betaWs[0], betaWs[1], 100)

   RRs_y = []
   RRs_w = []

   for ii, beta_V in enumerate(betas_V):
      for jj, beta_W in enumerate(betas_W):
         beta_k = np.asarray([beta_V , beta_W])
         beta_k = beta_k[:, np.newaxis]

         RRs = flash_3P_RRs(zi, Kis, beta_k)

         RRs_y.append(RRs[0, 0])
         RRs_w.append(RRs[1, 0])

   fig = plt.figure()
   ax = fig.add_subplot(111, projection='3d')

   X, Y = np.meshgrid(betas_V, betas_W)
   zs = np.array(RRs_y)
   Z = zs.reshape(X.shape)
```

```
    ax.plot_surface(X, Y, Z, color='g')

    zs2 = np.array(RRs_w)
    Z2 = zs2.reshape(X.shape)
    ax.plot_surface(X, Y, Z2, color='b')

    xx, yy = np.meshgrid([0, 1], [0, 1])
    zz = yy*0
    ax.plot_surface(xx, yy, zz, color='k', alpha=0.3)
    plt.show()
    ax.set_xlabel(r'$\beta^V$')
    ax.set_ylabel(r'$\beta^W$')
    ax.set_zlabel(r'$RRy$ (or $RR_w$)')

    if len(Z_lim) > 0:
        ax.set_zlim(-Z_lim, Z_lim)

    # Add legend with proxy artists
    col1_patch = mpatches.Patch(color='b', label=r'$RR_w$')
    col2_patch = mpatches.Patch(color='g', label=r'$RR_y$')
    plt.legend(handles=[col1_patch, col2_patch])

def Figure_liFiroozabadi_2d(zi, Ki, betasV_01 = [0, 0.84], betasV_02 = [0.27, 1.0],  betas_W_sup_in = [0.1,
0.8] ):
    betas_V_01 = np.linspace(betasV_01[0], betasV_01[1], 100)
    betas_V_02 = np.linspace(betasV_02[0], betasV_02[1], 100)
    beta_W_calc_RRy = []
    beta_W_calc_RRw = []

    betas_W_sup = betas_W_sup_in[0]
    for beta_V in betas_V_01:
        beta_k = np.asarray([beta_V , betas_W_sup])
        beta_k = beta_k[:, np.newaxis]

        root_RRw, infodict, ier, mesg  = fsolve(Fobj_RR_ii, betas_W_sup, args=(beta_V, zi, Ki, 1), xtol = 1e-9,
full_output=True, maxfev = 1000)
        beta_W_calc_RRw.append(root_RRw[0])
        betas_W_sup = root_RRw[0]

    betas_W_sup = betas_W_sup_in[1]
    for beta_V in betas_V_02:
        beta_k = np.asarray([beta_V , betas_W_sup])
        beta_k = beta_k[:, np.newaxis]

        root_RRy, infodict, ier, mesg  = fsolve(Fobj_RR_ii, betas_W_sup, args=(beta_V, zi, Ki, 0), xtol = 1e-9,
full_output=True, maxfev = 1000)
        beta_W_calc_RRy.append(root_RRy[0])
        betas_W_sup = root_RRy[0]

    plt.figure()
    plt.plot(betas_V_01, beta_W_calc_RRw, 'b')
```

```
    plt.plot(betas_V_02[:-10], beta_W_calc_RRy[:-10], 'g')
    plt.plot([0, 1, 0, 0], [0, 0, 1, 0], 'k')
    plt.xlabel(r'$\beta^V$')
    plt.ylabel(r'$\beta^W$')

"""
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Three Phases flash calculations
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
"""

def flash_3P_RachfordRice_values(beta_k, zi, Kis):

    Ki_V = Kis[0, :]
    Ki_W = Kis[1, :]

    Hi = 1 + (beta_k[0]*(Ki_V-1)) + (beta_k[1]*(Ki_W-1))

    xi_L = zi/Hi
    xi_W = xi_L*Ki_W
    yi   = xi_L*Ki_V

    return xi_L, xi_W, yi

def min_RRs(beta_k, zi, Ki):
    beta_k = np.asarray(beta_k)
    beta_k = beta_k[:, np.newaxis]

    zi = np.array(zi)
    Hi = 1 + sum(beta_k*(Ki-1))
    Fobj = sum(-zi*np.log(Hi))

    return Fobj

def flash_3P_RRs_fsolve(beta_k, zi, Ki):
    beta_k = np.asarray(beta_k)
    beta_k = beta_k[:, np.newaxis]

    Hi = 1 + sum(beta_k*(Ki-1))
    RRs = np.zeros((2, 1))

    RRs[0] = sum(zi*(Ki[0, :] - 1)/(Hi))
    RRs[1] = sum(zi*(Ki[1, :] - 1)/(Hi))

    return sum(abs(RRs[:, 0]))

def flash_3P_RRs(zi, Ki, beta_k):
    Hi = 1 + sum(beta_k*(Ki-1))
    RRs = np.zeros((2, 1))

    RRs[0] = sum(zi*(Ki[0, :] - 1)/(Hi))
    RRs[1] = sum(zi*(Ki[1, :] - 1)/(Hi))
```

```
    return RRs

def flash_3P_Jacobian(zi, Ki, beta_k):

    Hi = 1 + sum(beta_k*(Ki-1))

    Jacobian_3P = np.zeros((len(beta_k), len(beta_k)))
    Jacobian_3P[0, 0] = sum(-(zi*((Ki[0, :] - 1)**2)/(Hi**2)))
    Jacobian_3P[1, 1] = sum(-(zi*((Ki[1, :] - 1)**2)/(Hi**2)))
    Jacobian_3P[0, 1] = sum(-(zi*((Ki[0, :] - 1)*(Ki[1, :] - 1))/(Hi**2)))
    Jacobian_3P[1, 0] = Jacobian_3P[0, 1]

    return Jacobian_3P

def flash_3P_NewtonRaphson(beta_k, zi, Ki, iterMax=1000, tol=1e-6):
    beta_old = np.asarray(beta_k)
    beta_old = beta_old[:, np.newaxis]


    iter_1 = 0;
    Fobj = 1000

    while ((abs(Fobj) > tol) & (iter_1 < iterMax)):
        iter_1 = iter_1 + 1

        RRs = flash_3P_RRs(zi, Ki, beta_old)
        J = flash_3P_Jacobian(zi, Ki, beta_old)

        try:
            J_inv = np.linalg.inv(J)
        except:
            print('Convergence not reached')
            return beta_new, RRs, Fobj, iter_1
            break

        change = J_inv.dot(RRs)
        beta_new = beta_old -  change # (np.linalg.inv(J)*RRs )

        Fobj = (beta_new - beta_old).sum()
        beta_old = beta_new

    if iter_1 >= iterMax:
        print("Maximum iteration number in flash_3P'")

    return beta_new, RRs, Fobj, iter_1


"""
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Okuno 2010 - 3 Phases flash calculations using SRKKD
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
"""
def minF_Okuno(beta_k, zi, Ki):
    beta_k = np.asarray(beta_k)
    beta_k = beta_k[:, np.newaxis]

    #ti = 1 - sum(beta_k*(1-Ki))
    ti = 1 - (beta_k[0]*(1-Ki[0,:])) - (beta_k[1]*(1-Ki[1,:]))
    F_beta = (-zi*np.log(abs(ti))).sum()

    return F_beta

def Okuno_RR(beta_sup, Ki_V, Ki_W, zi):
    # Minimization method as proposed by Okuno
    ai_V = 1-Ki_V
    ai_W = 1-Ki_W

    Kis = np.asarray([Ki_V, Ki_W] )

    ai_p = np.zeros((2, len(zi)))
    ai_p[0, :] = ai_V
    ai_p[1, :] = ai_W
    aiT = ai_p.transpose()

    bi_okuno = np.zeros(len(zi))

    for jj, z_j in enumerate(zi):
        bis = [1-z_j, 1-(z_j*Ki_V[jj]), 1-(z_j*Ki_W[jj])]
        bi_okuno[jj] = min(bis)

    lb = len(zi)*[-np.inf]
    linear_constraint = LinearConstraint(aiT, lb, bi_okuno)


    results = minimize(minF_Okuno, beta_sup, args=(zi, Kis), method='COBYLA', tol=1e-8,
constraints=[linear_constraint])

    beta_new = results.x

    return beta_new

def calc_3Phases(Ki_V, Ki_W, beta_new, zi,P, T, Tc, Pc, ww, Rg, kij, Gi, Tcw):
    Kis = np.asarray([Ki_V, Ki_W] )
    xi_L, xi_W, yi = flash_3P_RachfordRice_values(beta_new, zi, Kis)

    Fug_i_W, ln_Fug_i_W, Fug_i_L, ln_Fug_i_L, Fug_i_V, ln_Fug_i_V = VLLE_fugacities(P, T, xi_W, xi_L, yi, Tc,
Pc, ww, Rg, kij, Gi, Tcw)

    return xi_L, xi_W, yi, Fug_i_L, ln_Fug_i_L, Fug_i_W, ln_Fug_i_W, Fug_i_V, ln_Fug_i_V


def VLLE_fugacities(P, T, xi_W, xi_L, yi, Tc, Pc, ww, Rg, kij, Gi, Tcw):
    Tr=T/Tc
```

```
    [ai, bi] = Parameters_SRK(Tc, Pc, ww, Rg, Tr)

    # 3. Calculate compresibiliti factors and fugacities
    # Fase W
    a_mix_W, b_mix_W = a_b_mixture_SRKKD(xi_W, ai, bi, kij, Gi, T, Tcw)
    r_W, CC_W, AA_W, BB_W = Roots_SRK(a_mix_W, b_mix_W, P, T, Rg)
    Z_W = min(r_W)
    Fug_i_W, ln_Fug_i_W = FugacityCoeff_SRK(xi_W, ai, bi, a_mix_W, b_mix_W, AA_W, BB_W, Z_W, kij  )

    # Fase L
    a_mix_L, b_mix_L = a_b_mixture_SRKKD(xi_L, ai, bi, kij, Gi, T, Tcw)
    r_L, CC_L, AA_L, BB_L = Roots_SRK(a_mix_L, b_mix_L, P, T, Rg)
    Z_L = min(r_L)
    Fug_i_L, ln_Fug_i_L = FugacityCoeff_SRK(xi_L, ai, bi, a_mix_L, b_mix_L, AA_L, BB_L, Z_L, kij  )

    # Fase Vapor
    a_mix_V, b_mix_V = a_b_mixture_SRKKD(yi, ai, bi, kij, Gi, T, Tcw)
    r_V, CC_V, AA_V, BB_V = Roots_SRK(a_mix_V, b_mix_V, P, T, Rg)
    Z_V = max(r_V)
    Fug_i_V, ln_Fug_i_V = FugacityCoeff_SRK(yi, ai, bi, a_mix_V, b_mix_V, AA_V, BB_V, Z_V, kij  )

    return Fug_i_W, ln_Fug_i_W, Fug_i_L, ln_Fug_i_L, Fug_i_V, ln_Fug_i_V

def Fobj_3Phases(Kis, beta_sup, P, T, zi, Tc, Pc, ww, Rg, kij, Gi, Tcw, iterMax = 10000, tol =  1e-6):
    Ki_V_old = Kis[0, :]
    Ki_W_old = Kis[1, :]

    Fobj_V = 1000
    Fobj_W = 1000

    iter_2 = 0

    while ((abs(Fobj_V)>tol) & (abs(Fobj_W)>tol) & (iter_2<iterMax)):
        iter_2 += 1

        beta_new = Okuno_RR(beta_sup, Ki_V_old, Ki_W_old, zi)
        beta_new = beta_new[:, np.newaxis]

        xi_L, xi_W, yi, Fug_i_L, ln_Fug_i_L, Fug_i_W, ln_Fug_i_W, Fug_i_V, ln_Fug_i_V =
calc_3Phases(Ki_V_old, Ki_W_old, beta_new, zi, P, T, Tc, Pc, ww, Rg, kij, Gi, Tc[0])

        Fobj_V = sum(np.log(Ki_V_old) - ln_Fug_i_L + ln_Fug_i_V)
        Fobj_W = sum(np.log(Ki_W_old) - ln_Fug_i_L + ln_Fug_i_W)

        Ki_V_old = np.exp(ln_Fug_i_L - ln_Fug_i_V);
        Ki_W_old = np.exp(ln_Fug_i_L - ln_Fug_i_W);

        if (abs(1-sum(xi_L)) > tol ) | (abs(1-sum(xi_W)) > tol) | ( abs(1-sum(yi)) > 1):
            Fobj_V = 1

    Kis = np.asarray([Ki_V_old, Ki_W_old] )
```

```
    beta_new = Okuno_RR(beta_sup, Ki_V_old, Ki_W_old, zi)
    beta_new = beta_new[:, np.newaxis]

    xi_L, xi_W, yi, Fug_i_L, ln_Fug_i_L, Fug_i_W, ln_Fug_i_W, Fug_i_V, ln_Fug_i_V = calc_3Phases(Ki_V_old,
Ki_W_old, beta_new, zi, P, T, Tc, Pc, ww, Rg, kij, Gi, Tc[0])
    RRs = flash_3P_RRs(zi, Kis, beta_new)

    if iter_2 > iterMax:
        print("Maximum number of Iterations reached (Fobj_3Phases)")

    return beta_new, xi_L, xi_W, yi, Fug_i_L, Fug_i_W, Fug_i_V, Ki_V_old, Ki_W_old, RRs, Fobj_V, Fobj_W,
RRs, iter_2


def Fobj_bubblePoint(Psup, betaV_required, Kis, beta_sup, T, zi, Tc, Pc, ww, Rg, kij, Gi_python, Tcw):
    # For bubble point beta_V = 0
    beta_new, xi_L, xi_W, yi, Fug_i_L, Fug_i_W, Fug_i_V, Ki_V, Ki_W, RRs, Fobj_V, Fobj_W, RRs, iter_2 =
Fobj_3Phases(Kis, beta_sup, Psup, T, zi, Tc, Pc, ww, Rg, kij, Gi_python, Tcw)
    beta_V = beta_new[0]

    Fobj = np.sqrt((betaV_required - beta_V)**2)
    return Fobj
```

References

[1] M.R. Gray, Upgrading oildsands bitumen and heavy oil, Pica Pica, Edmonton, Alberta, 2015.

[2] D.K. Banerjee, Oil sands, Heavy oil & Bitumen: From recover to refinery, PennWell Corporation, 2012.

[3] M. Fingas, Diluted Bitumen (Dilbit): A Future High Risk Spilled Material, in: Interspill, Amsterdam, Netherlands, 2015: pp. 1–24. https://doi.org/10.1017/CBO9781107415324.004.

[4] J. Du, W. Cluett, Modelling of a Naphtha Recovery Unit (NRU) with Implications for Process Optimization, Processes. 6 (2018) 74. https://doi.org/10.3390/pr6070074.

[5] S. Lopez-Zamora, J. Kong, S. Escobedo, H. de Lasa, Thermodynamics and Machine Learning Based Approaches for Vapor-Liquid-Liquid Phase Equilibria in N-Octane/Water Blends, as a Naphtha-Water Surrogate in Water Streams, Processes. 9 (2021) 1–31. https://doi.org/https://doi.org/10.3390/pr9030413.

[6] S. Lopez-Zamora, J. Kong, S. Escobedo, H. de Lasa, Vapour-Liquid-Liquid and Vapour-Liquid Equilibrium of Paraffinic Aromatic Synthetic Naphtha/Water Blends: Prediction of The Number of Phases, Can. J. Chem. Eng. (2021).

[7] M. Connolly, An isenthalpic-based compositional framework for nonlinear thermal simulation, Stanford University, 2018.

[8] T. Matsoukas, Fundamentals of Chemical Engineering Thermodynamics: with applications to chemical processes, Pearson Education, Inc., Michigan, 2013.

[9] M. Hillert, Phase Equilibria, Phase Diagrams and Phase transformations. Their Thermodynamic Basis, 2nd ed., Cambridge University Press, New York, 2008.

[10] M. Connolly, H. Pan, H. Tchelepi, Three-Phase Equilibrium Computations for Hydrocarbon–Water Mixtures Using a Reduced Variables Method, Ind. Eng. Chem. Res. 58 (2019) 14954–14974. https://doi.org/10.1021/acs.iecr.9b00695.

[11] Z. Li, A. Firoozabadi, General strategy for stability testing and phase-split calculation in two and three phases, SPE J. 17 (2012) 1096–1107. https://doi.org/10.2118/129844-PA.

[12] R. Li, H.A. Li, Improved three-phase equilibrium calculation algorithm for water/hydrocarbon mixtures, Fuel. 244 (2019) 517–527. https://doi.org/10.1016/j.fuel.2019.02.026.

[13] K. Schou Pedersen, P.L. Christensen, J. Azeem Shaikh, Phase Behavior of Petroleum Reservoir Fluids, 2nd ed., Taylor & Francis Group, LLC, 2015.

[14] H.H. Rachford, J.D. Rice, Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium, J. Pet. Technol. 4 (1952) 19–3. https://doi.org/10.2118/952327-g.

[15] A. Bahadori, Fluid Phase Behavior for Conventional and Unconventional Oil and Gas Reservoirs, 2017.

[16] V. Venkatasubramanian, The promise of artificial intelligence in chemical engineering: Is it here, finally?, AIChE J. 65 (2019) 466–478. https://doi.org/10.1002/aic.16489.

[17] E. Rich, Artificial Intelligence, McGraw-Hill book Company, New York, 1983.

[18] T.P. Trappenberg, Fundamentals of Machine Learning, Oxford University Press, Oxford, United Kingdom, 2019. https://doi.org/10.1093/oso/9780198828044.001.0001.

[19] F.C. Pereira, S.S. Borysov, Machine Learning Fundamentals, in: C. Antoniou, L. Dimitriou, F. Pereira (Eds.), Mobil. Patterns, Big Data Transp. Anal. Tools Appl. Model., Elsevier Inc., Amsterdam, Netherlands, 2019: pp. 9–29. https://doi.org/10.1016/b978-0-12-812970-8.00002-6.

[20] M.G. Pecht, Prognostics and Health Management of Electronics, John Wiley & Sons Ltd, Hoboken,USA, 2018. https://doi.org/10.1002/9780470385845.

[21] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed., O'Reilly Media, Inc., 2019.

[22] D. Delen, Predictive Analytics: Data Mining, Machine Learning and Data Science for Practitioners, 2nd ed., Pearson Education, Inc., Old Tappan, New Jersey, USA, 2021.

[23] C. Cranganu, M.E. Breaban, H. Luchian, Artificial intelligent approaches in petroleum geosciences, Springer International Publishing, Dordrecht, 2015. https://doi.org/10.1007/978-3-319-16531-8.

[24] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, C.M. Andreas, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, API design for machine learning software: Experiences From the Scikit-Learn Project, ArXivLabs. (2013) 1–15.

[25] T.M. Cover, P.E. Hart, Nearest Neighbor Pattern Classification, IEEE Trans. Inf. Theory. 1 (1967) 21–27. https://doi.org/10.1007/springerreference_62518.

[26] U. Sinha, B. Dindoruk, M. Soliman, Prediction of CO Minimum Miscibility Pressure MMP using Machine Learning Techniques, in: Soc. Pet. Eng. - SPE Improv. Oil Recover. Conf. 2020, Tulsa, OK, USA, 18 – 22 April 2020, 2020.

https://doi.org/10.2118/200326-ms.

[27] M. Kubat, An Introduction to Machine Learning, Springer International Publishing, Dordrecht, 2015.

[28] C. Cortes, V. Vapnik, Support-Vector Networks, Mach. Learn. 20 (1995) 273–297. https://doi.org/10.1109/64.163674.

[29] B.E. Boser, V.N. Vapnik, I.M. Guyon, A Training Algorithm for Optimal Classifiers, in: 5th Annu. Work. Comput. Learn. Theory, 1992: pp. 144–152. https://doi.org/https://doi-org.proxy1.lib.uwo.ca/10.1145/130385.130401.

[30] P. Kazemi, J.P. Steyer, C. Bengoa, J. Font, J. Giralt, Robust data-driven soft sensors for online monitoring of volatile fatty acids in anaerobic digestion processes, Processes. 8 (2020) 1–17. https://doi.org/10.3390/pr8010067.

[31] P.H. van Konynenburg, R.L. Scott, Critical lines and phase equilibria in binary van der Waals mixtures, Philos. Trans. R. Soc. London. 298 (1980) 495–540.

[32] M.J. Amani, M.R. Gray, J.M. Shaw, Phase behavior of Athabasca bitumen + water mixtures at high temperature and pressure, J. Supercrit. Fluids. 77 (2013) 142–152. https://doi.org/10.1016/j.supflu.2013.03.007.

[33] D. Loos, A.J.M. Wijen, G.A.M. Diepen, Phase equilibria and critical phenomena in fluid (propane + water) at high pressures and temperatures, (1980) 193–204.

[34] K.A. Johnston, F.F. Schoeggl, M.A. Satyro, S.D. Taylor, H.W. Yarranton, Phase behavior of bitumen and n-pentane, Fluid Phase Equilib. 442 (2017) 1–19. https://doi.org/10.1016/j.fluid.2017.03.001.

[35] T.W. de Loos, W.G. Penders, R.N. Lichtenthaler, Phase equilibria and critical phenomena in fluid (n-hexane + water) at high pressures and temperatures, Chem. Informationsd. 13 (1982) 83–91. https://doi.org/10.1002/chin.198217068.

[36] T.W. de Loos, J.H. van Dorp, R.N. Lichtenthaler, Phase equilibria and critical phenomena in fluid (n-alkane + water) systems at high pressures and temperatures, Fluid Phase Equilib. 10 (1983) 279–287.

[37] T. Yiling, T. Michelberger, E.U. Franck, High-pressure phase equilibria and critical curves of (water + n-butane) and (water + n-hexane) at temperatures to 700 K and pressures to 300 MPa, J. Chem. Thermodyn. 23 (1991) 105–112. https://doi.org/10.1016/S0021-9614(05)80063-1.

[38] Q. Wang, K.C. Chao, Vapor-Liquid and Liquid-Liquid Equilibria and Critical States of Water + N-Decane Mixtures, Fluid Phase Equilib. 59 (1990) 207–215. https://doi.org/10.1016/0378-3812(90)85035-9.

[39] R.L. Stevenson, D.S. LaBracio, T.A. Beaton, M.C. Thies, Fluid Phase Equilibria and critical phenomena for the dodecane-water and squalane-water systems at

elevated temperatures and pressures, Fluid Phase Equilib. 93 (1994) 317–336. https://doi.org/10.1016/0378-3812(94)87016-0.

[40] C. Black, G.G. Joris, H.S. Taylor, The solubility of water in hydrocarbons, J. Chem. Phys. 16 (1948) 537–543. https://doi.org/10.1063/1.1746932.

[41] A. Mączyński, B. Wiśniewska-Gocłowska, M. Góral, Recommended liquid-liquid equilibrium data. Part 1. Binary alkane-water systems, J. Phys. Chem. Ref. Data. 33 (2004) 549–577. https://doi.org/10.1063/1.1643922.

[42] M. Tu, D. Fei, Y. Liu, J. Wang, Phase Equilibrium for Partially Miscible System of Octane-Water, Theriogenology. 50 (1998) 507–519.

[43] C. Tsonopoulos, G.M. Wilson, High-temperature mutual solubilities of hydrocarbons and water. Part II: Ethylbenzene, Ethylcyclohexane and n-Octane, AIChE J. 31 (1985) 376–384. https://doi.org/10.1002/aic.690290618.

[44] C. Tsonopoulos, G.M. Wilson, High-temperature mutual solubilities of hydrocarbons and water. Part I: Benzene, cyclohexane andn-hexane, AIChE J. 29 (1983) 990–999. https://doi.org/10.1002/aic.690290618.

[45] C. Tsonopoulos, Thermodynamic analysis of the mutual solubilities of normal alkanes and water, Fluid Phase Equilib. 156 (1999) 21–33. https://doi.org/10.1016/S0378-3812(99)00021-7.

[46] J. Polak, B.C.-Y. Lu, Mutual Solubilities of Hydrocarbons and Water at 0 and 25 °C, Can. J. Chem. 51 (2006) 4018–4023. https://doi.org/10.1139/v73-599.

[47] B. Aktiengesellschaft, D.- Ludwigshqfen, Fluid mixtures at high pressures IX . Phase phenomena mixtures separation and critical + water), J. Chem. Thermodyn. 22 (1990) 335–353.

[48] H.J. Liaw, C.T. Chen, V. Gerbaud, Flash-point prediction for binary partially miscible aqueous-organic mixtures, Chem. Eng. Sci. 63 (2008) 4543–4554. https://doi.org/10.1016/j.ces.2008.06.005.

[49] M. Klauck, A. Grenner, J. Schmelzer, Liquid-liquid(-liquid) equilibria in ternary systems of water + cyclohexylamine + aromatic hydrocarbon (toluene or propylbenzene) or aliphatic hydrocarbon (heptane or octane), J. Chem. Eng. Data. 51 (2006) 1043–1050. https://doi.org/10.1021/je050520f.

[50] B.D. Marshall, A PC-SAFT model for hydrocarbons IV: Water-hydrocarbon phase behavior including petroleum pseudo-components, Fluid Phase Equilib. 497 (2019) 79–86. https://doi.org/10.1016/j.fluid.2019.06.007.

[51] M. Hajiw, A. Chapoy, C. Coquelet, Hydrocarbons - water phase equilibria using the CPA equation of state with a group contribution method, Can. J. Chem. Eng. 93 (2015) 432–442. https://doi.org/10.1002/cjce.22093.

[52] E.C. Voutsas, G.C. Boulougouris, I.G. Economou, D.P. Tassios, Water/hydrocarbon phase equilibria using the thermodynamic perturbation theory, Ind. Eng. Chem. Res. 39 (2000) 797–804. https://doi.org/10.1021/ie990559b.

[53] N. Haarmann, S. Enders, G. Sadowski, Modeling binary mixtures of n-alkanes and water using PC-SAFT, Fluid Phase Equilib. 470 (2018) 203–211. https://doi.org/10.1016/j.fluid.2017.11.015.

[54] D. NguyenHuynh, Accurate modeling of multiphase behavior of aqueous systems. I. Alkanes, alkenes, cycloalkanes, alcohols, aromatics, Fluid Phase Equilib. 473 (2018) 201–219. https://doi.org/10.1016/j.fluid.2018.06.019.

[55] M.B. Oliveira, J.A.P. Coutinho, A.J. Queimada, Mutual solubilities of hydrocarbons and water with the CPA EoS, Fluid Phase Equilib. 258 (2007) 58–66. https://doi.org/10.1016/j.fluid.2007.05.023.

[56] G.K. Folas, S.O. Derawi, M.L. Michelsen, E.H. Stenby, G.M. Kontogeorgis, Recent applications of the cubic-plus-association (CPA) equation of state to industrially important systems, Fluid Phase Equilib. 228–229 (2005) 121–126. https://doi.org/10.1016/j.fluid.2004.08.013.

[57] X. Yang, J. Czarnecki, The effect of naphtha to bitumen ratio on properties of water in diluted bitumen emulsions, Colloids Surfaces A Physicochem. Eng. Asp. 211 (2002) 213–222. https://doi.org/10.1016/S0927-7757(02)00279-0.

[58] H. Nourozieh, M. Kariznovi, J. Abedi, Physical properties and extraction measurements for the Athabasca bitumen + light hydrocarbon system: Evaluation of the pressure effect, solvent-to-bitumen ratio, and solvent type, J. Chem. Eng. Data. 56 (2011) 4261–4267. https://doi.org/10.1021/je2008846.

[59] Y. Dini, M. Becerra, J.M. Shaw, Phase behavior and thermophysical properties of peace river bitumen + propane mixtures from 303 K to 393 K, J. Chem. Eng. Data. 61 (2016) 2659–2668. https://doi.org/10.1021/acs.jced.6b00034.

[60] H. Nourozieh, M. Kariznovi, J. Abedi, Experimental and modeling studies of phase behavior for propane/Athabasca bitumen mixtures, Fluid Phase Equilib. 397 (2015) 37–43. https://doi.org/10.1016/j.fluid.2015.03.047.

[61] H. Nourozieh, M. Kariznovi, J. Abedi, Solubility of n-Butane in Athabasca Bitumen and Saturated Densities and Viscosities at Temperatures Up to 200°C, SPE J. (2016) 94–102. https://doi.org/10.2118/180927-PA.

[62] B. Azinfar, A. Haddadnia, M. Zirrahi, H. Hassanzadeh, J. Abedi, Phase behaviour of butane/bitumen fractions: Experimental and modeling studies, Fuel. 220 (2018) 47–59. https://doi.org/10.1016/j.fuel.2018.02.008.

[63] K.A. Johnston, Measurement and Modeling of Pentane-Diluted Bitumen Phase Behaviour, University of Calgary, 2017.

[64] P. Agrawal, F.F. Schoeggl, M.A. Satyro, S.D. Taylor, H.W. Yarranton, Measurement and modeling of the phase behavior of solvent diluted bitumens, Fluid Phase Equilib. 334 (2012) 51–64. https://doi.org/10.1016/j.fluid.2012.07.025.

[65] P. Agrawal, Measurement and modeling of the phase behavior of solvent diluted bitumens, Elsevier B.V., 2012. https://doi.org/10.1016/j.fluid.2012.07.025.

[66] H. Nourozieh, M. Kariznovi, J.G. Guan, J. Abedi, Measurement of thermophysical properties and modeling for pseudo-binary mixtures of n-decane and Athabasca bitumen, Fluid Phase Equilib. 347 (2013) 62–75. https://doi.org/10.1016/j.fluid.2013.03.010.

[67] M. Kariznovi, H. Nourozieh, J.G. Guan, J. Abedi, Measurement and modeling of density and viscosity for mixtures of Athabasca bitumen and heavy n-alkane, Fuel. 112 (2013) 83–95. https://doi.org/10.1016/j.fuel.2013.04.071.

[68] M. Ma, S. Chen, J. Abedi, Modeling the density, solubility and viscosity of bitumen/solvent systems using PC-SAFT, J. Pet. Sci. Eng. 139 (2016) 1–12. https://doi.org/10.1016/J.PETROL.2015.12.012.

[69] H. Nourozieh, M. Kariznovi, J. Abedi, Measurement and Evaluation of Bitumen / Toluene-Mixture Properties at Temperatures Up to 190 ^{o} C and Pressures Up to 10 MPa, (2016) 1705–1720.

[70] M. Zirrahi, H. Hassanzadeh, J. Abedi, Experimental and modeling studies of water, light n-alkanes and MacKay River bitumen ternary systems, Fuel. 196 (2017) 1–12. https://doi.org/10.1016/j.fuel.2017.01.078.

[71] J. Gao, R. Okuno, H.A. Li, An Experimental Study of Multiphase Behavior for n-Butane/Bitumen/Water Mixtures, SPE J. 22 (2017) 0783–0798. https://doi.org/10.2118/180736-PA.

[72] Z. Chen, D. Yang, Quantification of phase behaviour of solvents-heavy oil/bitumen systems in the presence of water at high pressures and elevated temperatures, Fuel. 232 (2018) 803–816. https://doi.org/10.1016/j.fuel.2018.05.116.

[73] M.J. Amani, M.R. Gray, J.M. Shaw, The phase behavior of Athabasca bitumen+toluene+water ternary mixtures, Fluid Phase Equilib. 370 (2014) 75–84. https://doi.org/10.1016/j.fluid.2014.02.028.

[74] H.H. Chien, Formulations for three-phase flash calculations, AIChE J. 40 (1994) 957–965. https://doi.org/10.1002/aic.690400607.

[75] D.-Y. Peng, D.B. Robinson, Two and Three Phase Equilibrium Calculations for Systems Containig Water, Can. J. Chem. Eng. 54 (1976) 595-****.

[76] S. Mokhatab, Three-phase flash calculation for hydrocarbon systems containing water, Theor. Found. Chem. Eng. 37 (2003) 291–294. https://doi.org/10.1023/A:1024043924225.

[77] S. Mokhatab, W.A. Poe, J.G. Speight, Handbook of Natural Gas Transmission and Processing, Gulf Professional Publishing, Burlington, MA, MA, 2006.

[78] J.S. Wu, P.R. Bishnoi, An algorithm for three-phase equilibrium calculations, Comput. Chem. Eng. 10 (1986) 269–276. https://doi.org/10.1016/0098-1354(86)85008-6.

[79] Y.-K. Li, L.X. Nghiem, Phase Equilibria of Oil, Gas and Water/Brine Mixtures, Can. J. Chem. Eng. 64 (1986) 486–496. https://doi.org/10.1002/cjce.5450640319.

[80] M.L. Michelsen, T. Hdjskole, The isothermal flask problem Part 1: Stability, Fluid Phase Equilib. 9 (1982).

[81] M.L. Michelsen, The isothermal flash problem. Part II. Phase-split calculation, Fluid Phase Equilib. 9 (1982) 21–40. https://doi.org/10.1016/0378-3812(82)85002-4.

[82] K. Wang, J. Luo, Y. Wei, K. Wu, J. Li, Z. Chen, Artificial neural network assisted two-phase flash calculations in isothermal and thermal compositional simulations, Fluid Phase Equilib. 486 (2019) 59–79. https://doi.org/10.1016/j.fluid.2019.01.002.

[83] Y. Li, T. Zhang, S. Sun, X. Gao, Accelerating flash calculation through deep learning methods, J. Comput. Phys. 394 (2019) 153–165. https://doi.org/10.1016/j.jcp.2019.05.028.

[84] M. Nazari, M.B. Asadi, S. Zendehboudi, A new efficient algorithm to determine three-phase equilibrium conditions in the presence of aqueous phase: Phase stability and computational cost, Fluid Phase Equilib. 486 (2019) 139–158. https://doi.org/10.1016/j.fluid.2018.12.013.

[85] N. Sabet, H.R. Erfani Gahrooei, A new robust stability algorithm for three phase flash calculations in presence of water, J. Nat. Gas Sci. Eng. 35 (2016) 382–391. https://doi.org/10.1016/j.jngse.2016.08.068.

[86] R. Okuno, R.T. Johns, K. Sepehrnoori, Application of a reduced method in compositional simulation, SPE J. 15 (2010) 39–49. https://doi.org/10.2118/119657-PA.

[87] R. Okuno, R.T. Johns, K. Sepehrnoori, Three-phase flash in compositional simulation using a reduced method, SPE J. 15 (2010) 689–703. https://doi.org/10.2118/125226-PA.

[88] J.E. Schmitz, R.J. Zemp, M.J. Mendes, Artificial neural networks for the solution of the phase stability problem, Fluid Phase Equilib. 245 (2006) 83–87. https://doi.org/10.1016/j.fluid.2006.02.013.

[89] J.P. Poort, M. Ramdin, J. van Kranendonk, T.J.H. Vlugt, Solving vapor-liquid flash problems using artificial neural networks, Fluid Phase Equilib. 490 (2019) 39–47. https://doi.org/10.1016/j.fluid.2019.02.023.

[90] A. Kashinath, M.L. Szulczewski, A.H. Dogru, A fast algorithm for calculating isothermal phase behavior using machine learning, Fluid Phase Equilib. 465 (2018) 73–82. https://doi.org/10.1016/j.fluid.2018.02.004.

[91] I. Argatov, V. Kocherbitov, A note on artificial neural network modeling of vapor-liquid equilibrium in multicomponent mixtures, Fluid Phase Equilib. 502 (2019) 112282. https://doi.org/10.1016/j.fluid.2019.112282.

[92] S. Mohanty, Estimation of vapour liquid equilibria of binary systems, carbon dioxide-ethyl caproate, ethyl caprylate and ethyl caprate using artificial neural networks, Fluid Phase Equilib. 235 (2005) 92–98. https://doi.org/10.1016/j.fluid.2005.07.003.

[93] B. Vaferi, Y. Rahnama, P. Darvishi, A. Toorani, M. Lashkarbolooki, Phase equilibria modeling of binary systems containing ethanol using optimal feedforward neural network, J. Supercrit. Fluids. 84 (2013) 80–88. https://doi.org/10.1016/j.supflu.2013.09.013.

[94] J. Kong, Multiphase Equilibrium in A Novel Batch Dynamic VL-Cell Unit with High Mixing: Apparatus Design and Process Simulation, The University of Western Ontario, 2020. https://ir.lib.uwo.ca/etd/7283/.

[95] S. Escobedo, J. Kong, S. Lopez-Zamora, H. de Lasa, Synthetic Naphtha Recovery from Water Streams: Vapor-Liquid-Liquid Equilibrium (VLLE) Studies in a Dynamic VL-Cell Unit with High Intensity Mixing, Can. J. Chem. Eng. (2021) 1–19. https://doi.org/https://doi.org/10.1002/cjce.24120.

[96] H. Renon, J.M. Prausnitz, Local compositions in thermodynamic excess functions for liquid mixtures, AIChE J. 14 (1968) 135–144. https://doi.org/10.1002/aic.690140124.

[97] M. Tu, D. Fei, Y. Liu, J. Wang, Phase Equilibrium for Partially Miscible System of Octane-Water, J. Chem. Eng. Chinese Univ. 12 (1998) 325–330.

[98] H.J. Paarsch, K. Golyaev, A Gentle Introduction to Effective Computing in Quantitative Research What Every Research Assistant Should Know, MIT Press, London, 2016.

[99] O. Castellanos Diaz, Measurement and modelling methodology for heavy oil and bitumen vapour pressure, Univ. Calgary. (2012). http://www.dt.co.kr/contents.html?article_no=2012071302010531749001.

[100] J. Wu, J.M. Prausnitz, Phase equilibria for systems containing hydrocarbons, water, and salt: an extended Peng-Robinson equation of state, Ind. Eng. Chem. Res. 37 (1998) 1634–1643. https://doi.org/10.1021/ie9706370.

[101] V.N. Kabadi, R.P. Danner, A Modified Soave-Redlich-Kwong Equation of State for Water-Hydrocarbon Phase Equilibria, Ind. Eng. Chem. Process Des. Dev. 24 (1985)

537–541. https://doi.org/10.1021/i200030a004.

[102] S. Sadjadi, M. Hosseinpour, F. Mohammadnezhad, S.J. Ahmadi, M.A. Khazayi, Hydrogen-Deuterium chemical exchange in supercritical water: Thermodynamic considerations for optimizing the synthesis of high degree deuterated benzene, J. Supercrit. Fluids. 125 (2017) 96–103. https://doi.org/10.1016/j.supflu.2017.02.010.

[103] M. Safamirzaei, H. Modarress, Modeling and predicting solubility of n-alkanes in water, Fluid Phase Equilib. 309 (2011) 53–61. https://doi.org/10.1016/j.fluid.2011.06.022.

[104] J. De Tommaso, F. Rossi, N. Moradi, C. Pirola, G.S. Patience, F. Galli, Experimental methods in chemical engineering: Process simulation, Can. J. Chem. Eng. 98 (2020) 2301–2320. https://doi.org/10.1002/cjce.23857.

[105] E.C. Carlson, Don't Gamble With Physical Properties, Chem. Eng. Prog. (1996) 35–46.

[106] H. Jia, H. Wang, K. Ma, M. Yu, Z. Zhu, Y. Wang, Effect of thermodynamic parameters on prediction of phase behavior and process design of extractive distillation, Chinese J. Chem. Eng. 26 (2018) 993–1002. https://doi.org/10.1016/j.cjche.2017.11.003.

[107] A. Marcilla, J.A. Reyes-Labarta, M.M. Olaya, Should we trust all the published LLE correlation parameters in phase equilibria? Necessity of their assessment prior to publication, Fluid Phase Equilib. 433 (2017) 243–252. https://doi.org/10.1016/j.fluid.2016.11.009.

[108] Q.P. He, J. Wang, Application of systems engineering principles and techniques in biological big data analytics: A review, Processes. 8 (2020) 1–30. https://doi.org/10.3390/PR8080951.

[109] T.A. Duever, Data science in the chemical engineering curriculum, Processes. 7 (2019) 1–7. https://doi.org/10.3390/pr7110830.

[110] L. Chiang, B. Lu, I. Castillo, Big data analytics in chemical engineering, Annu. Rev. Chem. Biomol. Eng. 8 (2017) 63–85. https://doi.org/10.1146/annurev-chembioeng-060816-101555.

[111] Naphtha (petroleum), hydrotreated heavy [MAK Value Documentation, 2010], in: MAK-Collection Occup. Heal. Saf., 2015: pp. 1–27. https://doi.org/10.1002/3527600418.mb6474248yole4815.

[112] J.L. Heidman, C. Tsonopoulos, C.J. Brady, G.M. Wilson, High-temperature mutual solubilities of hydrocarbons and water. Part II: Ethylbenzene, ethylcyclohexane, and n-octane, AIChE J. 31 (1985) 376–384. https://doi.org/10.1002/aic.690310304.

[113] C. Tsonopoulos, Thermodynamic analysis of the mutual solubilities of

hydrocarbons and water, Fluid Phase Equilib. 156 (1999) 21–33. https://doi.org/10.1016/S0378-3812(01)00520-9.

[114] J. Polak, B.C.-Y. Lu, Mutual Solubilities of Hydrocarbons and Water at 0 and 25 °C, Can. J. Chem. 51 (1973) 4018–4023. https://doi.org/10.1139/v73-599.

[115] R. Privat, J.N. Jaubert, E. Berger, L. Coniglio, C. Lemaitre, D. Meimaroglou, V. Warth, Teaching the Concept of Gibbs Energy Minimization through Its Application to Phase-Equilibrium Calculation, J. Chem. Educ. 93 (2016) 1569–1577. https://doi.org/10.1021/acs.jchemed.6b00205.

[116] M. del M. Olaya, J.A. Reyes-Labarta, M.D. Serrano, A. Marcilla, Vapor-liquid equilibria: Using the gibbs energy and the common tangent plane criterion, Chem. Eng. Educ. 44 (2010) 236–244.

[117] J.N. Jaubert, R. Privat, Application of the double-tangent construction of coexisting phases to Any Type of Phase Equilibrium For Binary Systems Modeled With the Gamma-Phi Approach, Chem. Eng. Educ. 48 (2014) 42–56.

[118] L.E. Baker, A.C. Pierce, K.D. Luks, Gibbs Energy Analysis of Phase Equilibria, Soc. Pet. Eng. J. 22 (1982) 731–742. https://doi.org/10.2118/9806-pa.

[119] M.E. Soares, A.G. Medina, C. McDermott, N. Ashton, Three phase flash calculations using free energy minimisation., Chem. Engng. Sci. 37 (1982) 521–528. https://doi.org/10.1016/0009-2509(82)80114-0.

[120] P.K. Sahu, S.R. Pal, A.K. Das, Estimation and inferential statistics, Springer, New Delhi, 2015. https://doi.org/10.1007/978-81-322-2514-0.

[121] A.D. Pozzolo, O. Caelen, R.A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification, Proc. - 2015 IEEE Symp. Ser. Comput. Intell. SSCI 2015. (2015) 159–166. https://doi.org/10.1109/SSCI.2015.33.

[122] G.W. Meindersma, Extraction of Aromatics from Naphtha with Ionic Liquids. From solvent development to pilot RDC evaluation, University of Twente, 2005.

[123] F.U. Babalola, A.A. Susu, On stability testing analysis in phase behavior calculations, Pet. Sci. Technol. 27 (2009) 1185–1195. https://doi.org/10.1080/10916460802456069.

[124] H. Hoteit, A. Firoozabadi, Simple phase stability-testing algorithm in the reduction method, AIChE J. 52 (2006) 2909–2920. https://doi.org/10.1002/aic.10908.

[125] D.V. Nichita, S. Gomez, E. Luna, Phase stability analysis with cubic equations of state by using a global optimization method, Fluid Phase Equilib. 194–197 (2002) 411–437. https://doi.org/10.1016/S0378-3812(01)00779-8.

[126] J. Kong, S. Escobedo, S. Lopez-Zamora, H. de Lasa, Phase Equilibrium in N-Octane/Water Separation Units: Vapor Pressures, Vapor and Liquid Molar

Fractions., Int. J. Chem. React. Eng. (2021) 1–11. https://doi.org/https://doi.org/10.1515/ijcre-2021-0031.

[127] Y. Li, T. Zhang, S. Sun, X. Gao, Accelerating flash calculation through deep learning methods, J. Comput. Phys. 394 (2019) 153–165. https://doi.org/10.1016/j.jcp.2019.05.028.

[128] E.H. Fernández-Martínez, E. López-López, Some theoretical results on Rachford-Rice equation for flash calculations: Multi-component systems, Comput. Chem. Eng. 140 (2020). https://doi.org/10.1016/j.compchemeng.2020.106962.

[129] J.A. Trangenstein, Minimization of Gibbs Free Energy in Compositional Reservoir Simulation., Soc. Pet. Eng. AIME, SPE. (1985) 233–246. https://doi.org/10.2523/13520-ms.

[130] R. Okuno, R.T. Johns, K. Sepehrnoori, A new algorithm for rachford-rice for multiphase compositional simulation, SPE J. 15 (2010) 313–325. https://doi.org/10.2118/117752-PA.

[131] R. Okuno, Modeling of multiphase behavior for gas flooding simulation, (2009) 371. http://proquest.umi.com/pqdweb?did=2251362301&Fmt=7&clientId=13395&RQT=309&VName=PQD.

[132] C.F. Leibovici, N. Jean, A solution of Rachford-Rice equations for multiphase systems, Fluid Phase Equilib. 112 (1995) 217–221.

[133] K.B. Haugen, A. Firoozabadi, L. Sun, Efficient and Robust Three-Phase Split Computations, AIChE J. 59 (2011) 2555–2565.

[134] W. Yan, E.H. Stenby, On solving the Rachford-Rice equation with higher order methods, Fluid Phase Equilib. 363 (2014) 290–292. https://doi.org/10.1016/j.fluid.2013.12.006.

[135] H. Hinojosa-Gomez, J. Solares-Ramırez, E.R. Bazua-Rueda, An Improved Algorithm for the Three-Fluid-Phase VLLE Flash Calculation Humberto, AIChE J. 59 (2012) 215–228. https://doi.org/10.1002/aic.

[136] D.P. Bertsekas, Nonlinear programming, 2nd ed., Athena Scientific, Belmont, Massachusetts, 1999.

[137] C.H. Whitson, M.L. Michelsen, The negative flash, Fluid Phase Equilib. 53 (1989) 51–71. https://doi.org/10.1016/0378-3812(89)80072-X.

[138] Z. Li, A. Firoozabadi, Initialization of phase fractions in Rachford-Rice equations for robust and efficient three-phase split calculation, Fluid Phase Equilib. 332 (2012) 21–27. https://doi.org/10.1016/j.fluid.2012.06.021.

[139] G.M. Wilson, A modified Redlich-Kwong equation of state, application to general physical data calculations, in: 65th Natl. AIChE Meet., Cleveland, OH, 1969.

Curriculum Vitae

Name: Sandra Milena Lopez-Zamora

Post-secondary Education and Degrees:	The University of Western Ontario London, Ontario, Canada 2018-2021 Ph.D. Chemical Engineering
	Universidad Nacional de Colombia Manizales, Caldas, Colombia 2011-2013 M.Sc. Chemical Engineering
	Universidad Nacional de Colombia Manizales, Caldas, Colombia 2005-2010 B.Eng. Chemical Engineering
Related Work Experience	Research Assistant The University of Western Ontario 2018-2021
	Teaching Assistant The University of Western Ontario 2018-2021
	Lecturer Politecnico Grancolombiano, Medellin (Colombia) 2015-2017

Publications:

Lopez-Zamora, S.; Kong, J.; Escobedo, S.; de Lasa, H. A Machine Learning Approach for Vapor-Liquid-Liquid and Vapor-Liquid Equilibrium of Aromatic Synthetic Naphtha/ Water Systems: Prediction of Phase Number. 2021. Accepted to Canadian Journal of Chemical Engineering, April 6, 2021

Kong, J.; Escobedo, S.; **Lopez-Zamora, S.**; de Lasa, H. Phase Equilibrium in N-Octane/Water Separation Units: Vapor Pressures, Vapor and Liquid Molar Fractions., *Int. J. Chem. React. Eng.* **2021**. https://doi.org/10.1515/ijcre-2021-0031

Escobedo, S.; Kong, J.; **Lopez-Zamora, S**.; de Lasa, H. Synthetic Naphtha Recovery from Water Streams: Vapor-Liquid-Liquid Equilibrium (VLLE) Studies in a Dynamic VL-Cell Unit with High Intensity Mixing. *Can. J. Chem. Eng.* **2021**. https://doi.org/10.1002/cjce.24120

Lopez-Zamora, S.; Kong, J.; Escobedo, S.; de Lasa, H. Thermodynamics and Machine Learning Based Approaches for Vapor-Liquid-Liquid Phase Equilibria in N-Octane/Water Blends, as a Naphtha-Water Surrogate in Water Streams. *Processes* **2021**.

Lopez-Zamora, S.; de Lasa, H. A Mid-Infrared Region (MIR) lumped Group Contribution based method for monitoring light gases and gasolines in Fluid Catalytic Cracking. *Chem. Eng. Sci.* **2020**, *212*, 115324, doi:10.1016/j.ces.2019.115324.

Lopez-Zamora, S.; Alkhlel, A.; de Lasa, H. Monitoring the progress of catalytic cracking for model compounds in the mid-infrared (MIR) 3200–2800 cm−1 range. *Chem. Eng. Sci.* **2018**, *192*, 788–802, doi:10.1016/j.ces.2018.08.021.

Dobrosz-Gómez, I.; Gómez-García, M.Á.; **López-Zamora, S.M.;** GilPavas, E.; Bojarska, J.; Kozanecki, M.; Rynkowski, J.M. Transition metal loaded TiO2 for phenol photo-degradation. *Comptes Rendus Chim.* **2015**, *18*, 1170–1182, doi:10.1016/j.crci.2015.03.006.

López-Zamora, S.M.; Dobrosz-Gómez, I.; Gómez-García, M.Á. Stability criteria and critical runway conditions of propylene glycol manufacture in a continuous stirred tank reactor. *Ing. e Investig.* **2015**, *35*, 56–60.

López-Zamora, S.M.; Gilpavas, E.; Gómez-garcía, M.Á.; Dobrosz-gómez, I. Phenol Photo-Degradation over TiO2 and Mo / TiO2 Catalysts. The Response Surface Methodology as Optimization Tool. *Inf. Tecnológica* **2014**, *25*, 3–12, doi:10.4067/S0718-07642014000500002.

López-Zamora, S.M.; Fontalvo, J.; Gómez-García, M.Á. Pervaporation membrane reactor design guidelines for the production of methyl acetate. Desalin. Water Treat. **2013**, 51, 2387–2393, doi:10.1080/19443994.2012.728058.