Multi-Phase Multi-Component Equilibrium Flash Calculations for *CompFlow Bio* using Modified Volume-Translated Peng-Robinson Equation of State

by

Alireza Zebarjadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Earth Sciences

Waterloo, Ontario, Canada, 2017

© Alireza Zebarjadi 2017

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Numerical modelling of fluid flow and transport in reservoir engineering problems is a challenging task, given variability and uncertainty in the physical properties of rock, the complexities of multi-fluid interaction at elevated pressure and temperature and limited computational resources. Nonetheless, this thesis seeks to provide a basis for expansion of our modeling capabilities in the context of hydrocarbon mixtures at equilibrium conditions. We briefly describe the numerical simulator *CompFlow Bio* and propose a package, with the solid thermodynamics background required for dealing with highly non-ideal mixture behavior, to amend this simulator.

Herein we present the governing equations of phase equilibrium and key expressions for calculating equilibrium mole fractions and phase properties over a broad range of pressure and temperature. We employed the traditional flash calculations model used in the petroleum industry and redesigned its procedure in order to accommodate the special design of *CompFlow Bio* and verify this modified model against a well-known commercial simulator results. In this proposed model, we use a modified Peng-Robinson equation of state improved by volume-translations for performing equilibrium flash calculations. Then, we describe three different case studies developed in order to investigate the accuracy of the proposed model, as well as describe the complexity of hydrocarbon mixture behavior at reservoir conditions.

Our findings indicate that: our model's performance is in close agreement with the commercial simulator software. Furthermore, these findings highlight various aspects of hydrocarbon behavior at high pressure and temperature, such as: decrease in non-aqueous phase mass density with increase in pressure while a gaseous phase is disappearing; a growing gas phase can dry out of the aqueous phase until it disappears; injection of carbon dioxide for enhanced oil recovery does not guarantee swelling of the non-aqueous phase and gas phase mass density increase with the injection of carbon dioxide, depending on the light hydrocarbon content of the gaseous phase.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. Andre Unger and Dr. Marios Ioannidis for their support, assistance, and mentorship. I sincerely thank them for their guidance, insight, and patience – which have made this work possible. Furthermore, I acknowledge also my committee members, Drs. Maurice Dusseault and Yuri Leonenko for their remarks and opinions throughout this process.

I gratefully acknowledge the University Consortium for Field-focused Groundwater Contamination Research for providing source of funding during this course of research.

Finally, but not at all in the least, I owe a great debt of gratitude to my parents, whose unconditional love and support made this journey possible for me.

Dedication

To my beloved mother, Shahin

In memory of my father, Ahmad.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background and Motivation

For most producing oil fields, a conventional approach to enhanced oil recovery (EOR) involves the injection of water or hydrocarbon gas. An attractive alternative to these methods which also addresses environmental considerations involves the injection of carbon dioxide (CO_2), often as a supercritical fluid. The unique properties of supercritical carbon dioxide not only are able to improve the oil production in the final (tertiary) phase of reservoir life, but also result in considerable environmental benefits through minimization of greenhouse gas (GHG) emissions, since available carbon dioxide from a point source such as a fertilizer or cement plant, can be captured and used as injection fluid for increasing oil recovery.

Understanding the physics associated with the injection of carbon dioxide into oil reservoirs requires laborious and expensive experiments. When the carbon dioxide is supercritical, the prevailing high pressure and temperature render experimentation at reservoir conditions very expensive, time-consuming, and fraught with many challenges, such as saturating the matrix and fracture fluids with the reservoir fluids and monitoring saturations in situ. To compensate for the lack of exhaustive experimentation, numerical models may be used to predict the results of these experiments without doing them. When numerical models are validated against experiments and we are confident that we understand the underlying physics, then those numerical models can be used to explore different scenarios. In this way, considerable amount of money and time can be saved.

One of these numerical models is *CompFlow Bio*, a simulator developed by Unger et al. (1995). *CompFlow Bio* is a multi-phase, multi-component, first-order accurate, finite-volume simulator and it is unique in a way that handles discrete fracture networks. The ability of *CompFlow Bio* to reproduce well-controlled laboratory experiments has been scrutinized in a number of previous works (Enouy et al., 2011; Unger et al., 1998; Unger et al., 1995; Walton et al., 2017; Yu et al., 2009). In all of these studies, however, ideal fluids and room conditions were used. To be able to use this simulator with confidence in high pressure and temperature situations, which is the case in carbon dioxide sequestration, *CompFlow Bio* needs functionality to compute *species equilibrium ratios between equilibrated phases* at high pressure and temperature. These correspond to species mole fraction ratios between phases at equilibrium and can be assumed independent of composition if mixtures can be considered ideal. This is the case for equilibrium between water, air and a low-solubility non-aqueous phase liquid (NAPL) at low pressures and temperatures, when equilibrium compositions are calculated

conveniently using Henry's, Dalton's and Raoult's laws. For complex hydrocarbon liquids, at high pressures and temperatures, this assumption is no longer valid and other approaches should be used for attaining accurate and acceptable results. Two well-known approaches for calculating equilibrium ratios are the method of *direct minimization of Gibbs free energy* and the *equal fugacity method*. In this thesis, we use the *equal fugacity method* as it provides accurate results with fewer computations.

Like any other numerical model and simulator, in *CompFlow Bio* there are some assumptions underlying the calculations. The most important assumptions are as follows:

1) Phases are in chemical equilibrium in each node: This assumption will provide the required basis for using fugacity coefficients of the components for solving for distribution coefficients.

2) Gas phase is always present: In *CompFlow Bio*, there is always a gaseous phase present, which means that the number of different combinations of phases that could be present at each node is four and we refer to each of these four combinations as a State. The states are as follows: *State 1* (gaseous, aqueous, and non-aqueous phases present); *State 2* (gaseous and aqueous phases are present); *State 3* (only gaseous phase present); *State 4* (gaseous and non-aqueous phases are present).

1.2 Objectives and Scope

The objective of this thesis is to provide *CompFlow Bio* with the required functionality to calculate equilibrium ratios accurately. This function should be adapted to *CompFlow Bio*'s terminology and of course inputs and outputs. By appending this function to *CompFlow Bio*, the simulator will expand its applicability into wider range of science and engineering problems, such as fractured reservoir engineering and carbon dioxide sequestration. As with every other model, the assumptions underlying the simulator and model restrict its applicability. The most important assumption is equilibrium of phases. This means phases are in equilibrium upon contact in every node. When appropriate, this assumption drastically reduces the computational load of *CompFlow Bio*. It should be noted, however, that even when this assumption is relaxed and rate-limited mass transfer is taken into account (*e.g.*, Enouy et al. (2011), computation of equilibrium ratios is still needed to calculate the driving force for mass transfer. The result of this work is the foundation for building a sub-function in *CompFlow Bio* that would expand its capabilities into areas of application where rigorous thermodynamic account of mixture non-ideality is necessary.

The main input to the computations carried out in this work is the current state from *CompFlow Bio*. After calculation of the equilibrium ratios based on the input given, we verify if the state assumed at first is indeed the correct one. If not, a set of tests is used to decide the correct state, which becomes an

output of the computations. This output is returned to *CompFlow Bio* alongside computed equilibrium ratios and phase properties.

1.3 Organization of this Thesis

This document is organized in five chapters and two appendices. Chapter 2 reviews the theoretical background required for finding equilibrium ratios as well as related published research. In Chapter 3, it is demonstrated how thermodynamic theory is applied to the problem at hand. Chapter 4 includes results of this work with reference to three case studies and compares the accuracy of the computations against the results of a well-known commercial simulator. Chapter 5 provides a summary of our major findings and suggests topics for future studies. Finally, the Appendices contain details on (i) improving the convergence of the flash calculations and (ii) the main parts of the computational codes used within this work.

Chapter 2
Background Theory and Literature Review

2.1 Description of *CompFlow Bio*

CompFlow Bio is a three-phase, multicomponent, deterministic numerical model for fluid flow and dissolved species transport that includes capillary pressure and equilibrium partitioning relationships in its calculations. It has been recently augmented (Walton et al., 2017) to include randomly generated, axis-aligned, discrete fracture networks (DFNs). The DFN is coupled with the porous medium (PM) to form a single continuum. The domain is discretized using a finite-volume scheme in an unstructured mesh of rectilinear control volumes (CV).

CompFlow Bio is a numerical model that includes three mobile phases: aqueous, non-aqueous and gaseous; and multiple components: water (H_2O), one or more oil species, nitrogen (N_2), and optionally additional gaseous species. Nitrogen is mentioned separately, as in *CompFlow Bio* it is required to have a gas phase always available (which is assumed to consist of, at least, N_2). Presence of a gas phase, even in infinitesimally small quantity, helps with efficient and robust calculations (Forsyth, 1993). Nitrogen is assumed insoluble in other phases. Furthermore, currently in *CompFlow Bio* water is assumed to be insoluble in the non-aqueous phase. Both of these assumptions are relaxed in the present work.

CompFlow Bio uses equilibrium partitioning to transfer components between phases. A first-order accurate finite-volume approach is used to discretize the governing three-phase flow equations. Phase pressures, saturations, and component mole fractions are solved using a fully implicit scheme. The simulator chooses time step size adaptively. *CompFlow Bio* uses a Newton-Raphson linearization method with CGStab acceleration and a block-sparse matrix data structure for the system of nonlinear differential equations (Walton, 2013). Complete sets of the governing equations for *CompFlow Bio* have been reported in the literature previously (Forsyth, 1993; Yu et al., 2009).

Here we review equations for a three-phase system with a number of components equal to NC. For cases with lower number of phases, similar equations are applicable.

The equation for conservation of moles of component p is

$$\frac{\partial}{\partial t}\left(\phi \sum_{l=n,q,g} S_l M_l x_{p,l} + \rho_b \kappa_d M_q \sum_{i=1}^{m} x_{o,q}\right) = -\sum_{l=n,q,g} \nabla.\left(M_l x_{p,l} \vec{v}_l\right) + \sum_{l=n,q,g} \nabla.\left(\phi M_l \nabla x_{p,l} S_l \boldsymbol{D}_l\right) + Q_p \quad (2\text{-}1)$$

where:

n: Non-aqueous phase,

q: Aqueous phase,

g: Gaseous phase,

o: Non-methane hydrocarbon components,

m: Number of non-methane hydrocarbon components,

t: Time [d],

ϕ: Porosity [-],

S_l: Saturation of phase l [-],

M_l: Molar density of phase l [$\frac{mol}{m^3}$],

$x_{p,l}$: Mole fraction of component p in phase l [-],

ρ_b: Balk density of the porous medium [$\frac{kg}{m^3}$],

κ_d: Sorption coefficient [$\frac{m^3}{kg}$],

Q_p: Source/sink term for species [$\frac{mol}{m^3 d}$],

$\boldsymbol{D}$: Hydrodynamic dispersion tensor [],

The Darcy velocity of phase l, $\vec{v}_l$, is given by the following equation

$$\vec{v}_l = -\boldsymbol{K}\frac{k_{r_l}}{\mu_l}(\nabla P_l - \rho_l g \nabla d) \quad (2\text{-}2)$$

in which:

ρ_l: Mass density of phase l [$\frac{kg}{m^3}$],

$\boldsymbol{K}$: Intrinsic permeability tensor [m^2],

k_{r_l}: Relative permeability of phase [-],

μ_l: Coefficient of dynamic viscosity of phase l [$\frac{kg}{m.s}$],

P_l: Fluid pressure of phase l [$\frac{kg}{ms^2}$],

g: Gravitational acceleration constant [$\frac{m}{s^2}$],

d: Depth [m],

The dispersion tensor (Bear, 1972) has the following form

$$\phi S_l \boldsymbol{D}_l = \alpha_L^l \vec{v}_l \delta_{IJ} + \left(\alpha_L^l - \alpha_T^l\right) \frac{\vec{v}_{lI} \vec{v}_{lJ}}{|\vec{v}_l|} + \phi S_l \tau \boldsymbol{D}_l^* \delta_{IJ} \tag{2-3}$$

where:

α_L^l,: Longitudinal and transversal dispersivities of phase l [m^2],

α_T^l: Transversal dispersivities of phase l [m^2],

δ_{IJ}: Kroneker delta function at matrix indices I and J,

τ: Tortuosity [-],

$\boldsymbol{D}_l^*$: Molecular diffusion coefficient [$\frac{m^2}{s}$],

$\vec{v}_{lI}, \vec{v}_{lJ}$: Covariant components of phase l velocity in terms of considered coordinate system [$\frac{m}{s}$].

Note that the intrinsic permeability tensor is assumed to be aligned with the principal spatial axes and thus, for the porous medium, it has the following form:

$$\boldsymbol{K} = \begin{bmatrix} k_{x_1x_1} & 0 & 0 \\ 0 & k_{x_2x_2} & 0 \\ 0 & 0 & k_{x_3x_3} \end{bmatrix} \tag{2-4}$$

For fractures with effective hydraulic aperture $2b$, planar directions I and J, and with normal direction K, the intrinsic permeability components are given by:

$$k_{II} = k_{JJ} = \frac{(2b)^2}{12} \tag{2-5}$$

The above formulation leaves $3 \times NC + 6$ unknowns per CV: three saturations, S_l; $3 \times NC$ mole fractions, $x_{p,l}$; and three pressures P_l. Various constraints (or simplifying assumptions) exist among the

unknowns that eventually reduce the number of unknowns to NC *primary variables* per control volume. Unknown values calculated via constraints are termed *secondary variables*.

This way of calculations practically decouples governing equations from other equations such as those used for equilibrium ratios. This gives *CompFlow Bio* significant advantage over other simulators, because it renders the calculations more robust and time efficient. By separating governing equations such as conservation of moles equations, first *CompFlow Bio* breaks a huge system of equations to a Newton iteration loop, which instead converges more rapidly and is more robust.

The following applies to a CV in which all three phases are present. If a phase is not present, the numerical model undergoes primary variable switching for the affected CVs. This, as well as a description of system closure in all primary variable configurations, is given in (Forsyth and Shao, 1991). A first constraint concerns the summation of phase saturations:

$$\sum_{l=n,q,g} S_l = 1 \tag{2-6}$$

After solving for two saturations, the third one is computed by this constraint and is thus a secondary variable. The above constraint reduces the number of unknowns in *CompFlow Bio* by one.

The sum of mole fractions of all components in a phase must equal unity. The system is reduced by three primary variables (one per phase) by using the following additional constraints:

$$\sum_{i=1}^{NC} x_{i,l} = 1 \quad l = n, q \; or \; g \tag{2-7}$$

Equilibrium partitioning is employed to transfer components between phases. Partitioning relationships reduce the number of unknowns by $2 \times NC$. The general forms of the partitioning relationships are

$$K_{i,gn} = \frac{x_{i,g}}{x_{i,n}} \frac{P_n}{P_g} \tag{2-8}$$

$$K_{i,gq} = \frac{x_{i,g}}{x_{i,q}} \frac{P_q}{P_g} \tag{2-9}$$

where $K_{i,pl}$ is equilibrium ratio of component i between phases p and l.

Finally, pressures are related to each other via capillary pressure relationships. With experimentally determined data to parameterize the two-phase capillary pressures, $P_{c_{nq}}$, $P_{c_{ng}}$ and $P_{c_{gq}}$, the constraint equations are

$$P_q = P_n - \hat{\alpha} P_{c_{nq}}(S_q) - (1 - \hat{\alpha}) P_{c_{nq}}(S_q = 1)$$

$$P_g = P_n + \hat{\alpha} P_{c_{gn}}(S_g) - (1 - \hat{\alpha}) \left[P_{c_{gq}}(S_g) - P_{c_{nq}}(S_q = 1) \right] \tag{2-10}$$

$$\hat{\alpha} = min\langle 1, S_n / S_n^* \rangle$$

where $P_{c,pl}$ is capillary pressure between phases p and l, capillary pressures are functions of saturation and S_n^* [-] is an empirical curve blending parameter.

The blending function $\hat{\alpha}$ and parameter S_n^* provide a linear transition from two-phase capillary pressure data to three phase system of gas, non-aqueous and aqueous (Forsyth, 1991).

In conclusion, for the number of primary variables per CV we can write:

$$\underbrace{(3 \times NC + 6)}_{total\ unknowns} - \overbrace{1}^{Eq.(2\text{-}6)} - \underbrace{3}_{Eq.(2\text{-}7)} - \overbrace{2 \times NC}^{Eqs.(2\text{-}8)\&(2\text{-}9)} - \underbrace{2}_{Eq.(2\text{-}10)} = \underbrace{NC}_{primary\ variables} \tag{2-11}$$

It should be noted that some physical quantities (phase molar and mass densities) are functions of temperature, pressure, and mole fraction of components. These add to the non-linearity of the problem.

It is evident from the aforementioned discussion that relationships for viscosity, equilibrium ratios, capillary pressure, and relative permeability are required by the *CompFlow Bio* model.

Phase densities are required for the calculation of viscosities. In addition, the saturation of each phase directly affects its capillary pressure and relative permeability. In return, the composition of each phase is required for obtaining its saturation and density. There is no need to emphasize that the composition of each phase in a multi-phase system is governed by equilibrium ratios. Therefore, equilibrium ratios are a critical piece of information, as they directly and indirectly affect viscosity, capillary pressure, and relative permeability. It should be noted these ratios vary significantly with pressure, temperature, and composition of the phases.

Currently in *CompFlow Bio*, equilibrium ratios are calculated based on Henry's law, Dalton's law of partial pressures and Raoult's law (Unger et al., 1995). Both Raoult's and Dalton's laws are based on ideal mixture behavior. This means that in cases of non-ideal phase behavior, application of these laws for calculating equilibrium ratios would result in considerable error. The same is true of Henry's law, which is applicable to ideally dilute solutions for which there are exist no interactions between solution components. To broaden the range of applicability of *CompFlow Bio*, it is essential to improve the manner in which equilibrium ratios are calculated. In this thesis, we will present an improved and robust

package, which calculates equilibrium ratios across a broad range of pressure and temperature, from ambient conditions to the conditions prevailing in petroleum reservoirs. At high pressure and temperature, equilibrium ratios are strong functions of pressure, temperature and phase composition and ignoring these dependencies results in considerable error. The proposed package also corrects the current sub-functions in *CompFlow Bio* that calculate mass and molar densities as well as the sub-function that determines phase appearance or disappearance.

2.1.1 List of Primary and Secondary Variables in CompFlow Bio

As mentioned previously, *CompFlow Bio* always considers a gas phase present. Therefore, four different combinations of phases are possible in each *CompFlow Bio* node. Each of these combinations is called a *state*. These states are as follows:

State 1: Gaseous, aqueous, and non-aqueous phases are present,

State 2: Gaseous and aqueous phases are present,

State 3: Only gaseous phase is present,

State 4: Gaseous and non-aqueous phases are present.

Table 2-1 summarizes the primary variables chosen for each state. Unknowns not mentioned in Table 2-1 are calculated by using constraints and thus are secondary variables. In this table, the number of non-methane hydrocarbons components, denoted by Greek letter Omicron, o, present in the system is m. Therefore, total number of hydrocarbon components is $m + 1$ and total number of components available in the system is equal to $m + 4$. In addition, subscript *w* stands for water component.

The first column in Table 2-1 indicates the governing equation, i.e., mole conservation equation (see Equation (2-1)), associated with each primary variable. In other words, in order to solve for a certain primary variable, *CompFlow Bio* differentiates the associated governing equation with respect to that primary variable and uses a Newton-Raphson iteration procedure to find the value of that primary variable. Since each governing equation is a function of the phase pressures, saturations and the components mole fractions, all governing equations are solved simultaneously using a Newton-Raphson iteration scheme. Note that based on the state, the primary variable associated with each equation may differ, but the order of the equations mentioned in the first column is always the same, as we can differentiate a function with respect to different independent variables.

In each Newton-Raphson iteration, *CompFlow Bio* adopts associated primary variables based on the *state* and passes these primary variables as input to our package, which then calculates all secondary variables, as well as phase properties. It also signals to *CompFlow Bio* if a phase is about to appear or

disappear. Figure 2-1 describes a simplified and schematic logic diagram of the current configuration of *CompFlow Bio*. In this figure, a closed envelope named "*Current Thesis*" outlines sections of *CompFlow Bio* that this work intends to replace.

Mole Conservation Equation	State 1	State 2	State 3	State 4
H_2O	P_n	P_q	P_g	P_n
N_2	S_q	S_q	$x_{w,g}$	S_n
o_1	S_n	$x_{o_1,g}$	$x_{o_1,g}$	$x_{w,g}$
o_2	$x_{o_1,n}$	$x_{o_2,g}$	$x_{o_2,g}$	$x_{o_1,n}$
o_3	$x_{o_2,n}$	:	:	$x_{o_2,n}$
:	:	$x_{o_{m-1},g}$	$x_{o_{m-1},g}$	:
o_m	$x_{o_{m-1},n}$	$x_{o_m,g}$	$x_{o_m,g}$	$x_{o_{m-1},n}$
CO_2	$x_{CO_2,g}$	$x_{CO_2,g}$	$x_{CO_2,g}$	$x_{CO_2,g}$
CH_4	$x_{CH_4,g}$	$x_{CH_4,g}$	$x_{CH_4,g}$	$x_{CH_4,g}$

Table 2-1: List of primary variables and their associated governing equation in each state.

2.2 Quantitative Description of Phase Equilibrium

In this section, we review the thermodynamic basis and general relationships governing a system in chemical equilibrium, as well as different approaches to calculate equilibrium ratios. Unless otherwise stated, the formulations presented in this section are adopted from Smith et al. (2001).

For closed thermodynamic systems with constant composition, which can only exchange heat and work, we can write

$$dU = \delta Q + \delta W \tag{2-12}$$

in which, U is the system internal energy, Q is heat and W is work done by the system. This equation constitutes the first law of thermodynamics.

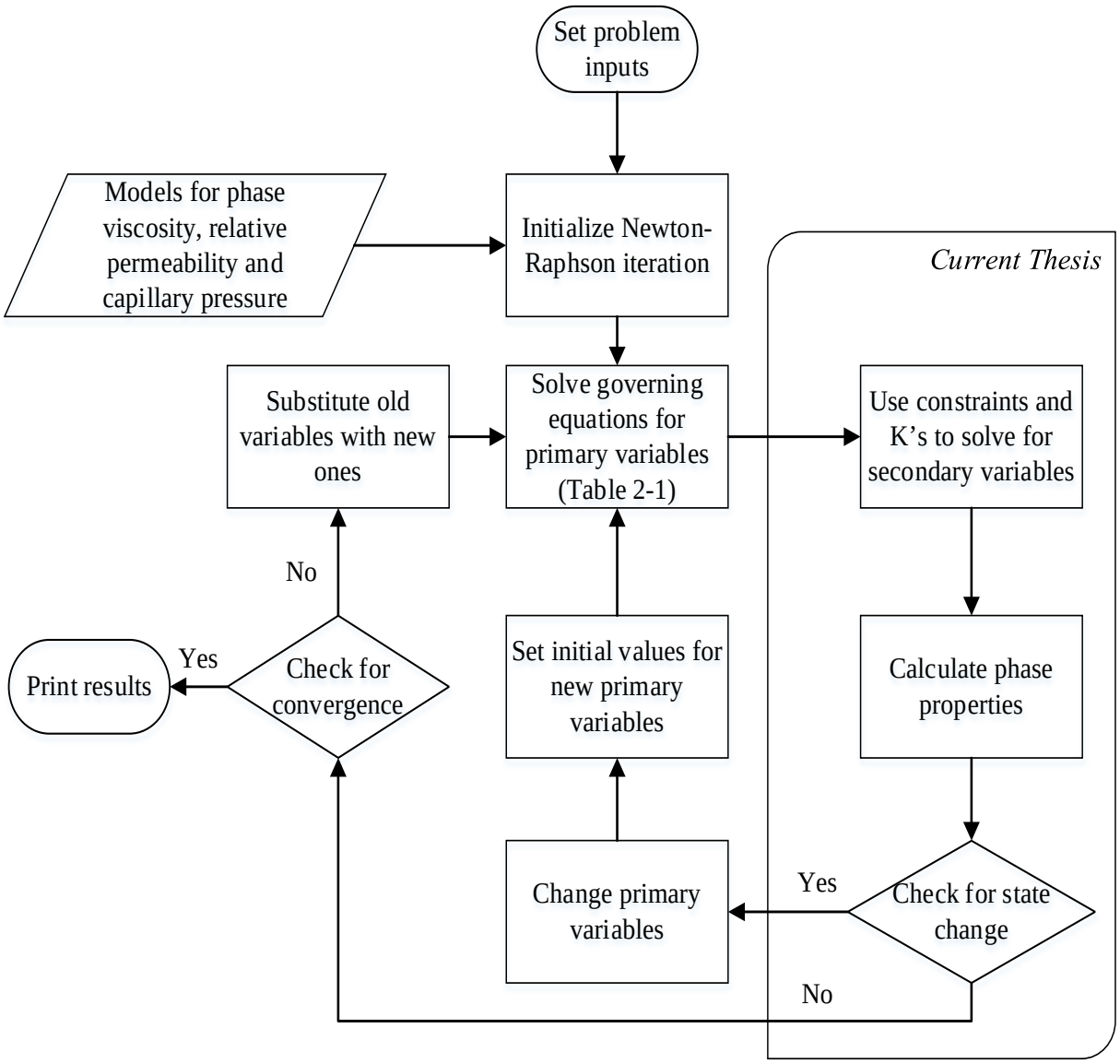

Figure 2-1: A schematic flowchart of *CompFlow Bio* including role of current thesis.

Furthermore, we may write the following expressions for work and heat, considering reversible process and non-viscous fluid (ideal gas)

$$\delta W = -PdV \tag{2-13}$$

$$\delta Q = TdS \tag{2-14}$$

where S is the system entropy. The negative sign in work equation, defines W to be energy flow from system to surroundings; while for heat, direction of Q is considered to be into the system.

Therefore, we can write:

$$dU = TdS - PdV \tag{2-15}$$

Now we recall definitions of three common thermodynamics potentials; Enthalpy (H), Helmholtz Free Energy (A) and Gibbs Free Energy (G):

$$H \equiv U + PV \tag{2-16}$$

$$A \equiv U - TS \tag{2-17}$$

$$G \equiv H - TS \tag{2-18}$$

Differentiating these three equations and substituting dU with using Equation (2-15), yields:

$$dH = TdS + VdP \tag{2-19}$$

$$dA = -SdT - PdV \tag{2-20}$$

$$dG = -SdT - VdP \tag{2-21}$$

For mixtures, these thermodynamic potentials (U, H, A, G) are also function of composition. For systems of changing composition, we can write the total differential for enthalpy as follows:

$$dH = \left(\frac{\partial H}{\partial S}\right)_{P,n} dS + \left(\frac{\partial H}{\partial P}\right)_{S,n} dP + \sum_{i=1}^{NC} \left(\frac{\partial H}{\partial n_i}\right)_{S,P,n_{j\neq i}} dn_i \tag{2-22}$$

From Equation (2-19), we have:

$$\left(\frac{\partial H}{\partial S}\right)_{P,n} = T, \;\; and \;\; \left(\frac{\partial H}{\partial P}\right)_{S,n} = V \tag{2-23}$$

Substituting in Equation (2-22) gives:

$$dH = TdS + VdP + \sum_{i=1}^{NC} \left(\frac{\partial H}{\partial n_i}\right)_{S,P,n_{j\neq i}} dn_i \tag{2-24}$$

Similarly, we can write:

$$dU = TdS - PdV + \sum_{i=1}^{NC} \left(\frac{\partial U}{\partial n_i}\right)_{S,V,n_{j\neq i}} dn_i \tag{2-25}$$

$$dA = -SdT - PdV + \sum_{i=1}^{NC} \left(\frac{\partial A}{\partial n_i}\right)_{T,V,n_{j\neq i}} dn_i \tag{2-26}$$

$$dG = -SdT + VdP + \sum_{i=1}^{NC} \left(\frac{\partial G}{\partial n_i}\right)_{P,T,n_{j\neq i}} dn_i \tag{2-27}$$

Next, we define:

$$\left(\frac{\partial H}{\partial n_i}\right)_{S,P,n_{j\neq i}} \equiv \bar{H}_i \tag{2-28}$$

$$\left(\frac{\partial U}{\partial n_i}\right)_{S,V,n_{j\neq i}} \equiv \bar{U}_i \tag{2-29}$$

$$\left(\frac{\partial A}{\partial n_i}\right)_{T,V,n_{j\neq i}} \equiv \bar{A}_i \tag{2-30}$$

$$\left(\frac{\partial G}{\partial n_i}\right)_{P,T,n_{j\neq i}} \equiv \bar{G}_i \tag{2-31}$$

From Equations (2-15) and (2-21), the following result is obtained:

$$dG = dU - TdS - SdT + PdV + VdP \tag{2-32}$$

Substituting Equations (2-25) and (2-27) into Equation (2-32) results in

$$\sum_{i=1}^{NC} \bar{G}_i dn_i = \sum_{i=1}^{NC} \bar{U}_i dn_i \tag{2-33}$$

From which it follows

$$\bar{G}_i = \bar{U}_i \tag{2-34}$$

Similarly, from Equations (2-19) and (2-21) one can write:

$$dG + SdT - VdP = dH - TdS - VdP \tag{2-35}$$

Which in combination with Equations (2-24) and (2-27) results:

$$\bar{G}_i = \bar{H}_i \tag{2-36}$$

Similarly, an equation for $\bar{A}_i$ can be derived.

Therefore, we have:

$$\bar{G}_i = \bar{H}_i = \bar{U}_i = \bar{A}_i \tag{2-37}$$

These quantities are defined as the *chemical potential* (Edmister, 1961):

$$\mu_{c_i} \equiv \bar{G}_i = \bar{H}_i = \bar{U}_i = \bar{A}_i \tag{2-38}$$

In this thesis, we denote chemical potential with μ_c to avoid confusion with dynamic viscosity coefficient.

The chemical potential of a species in a mixture is defined as the rate of change of free energy of the thermodynamic system with respect to change in the number of atoms or molecules of the species in the mixture.

For a system at equilibrium, Gibbs free energy is minimum, so Equation (2-27) becomes:

$$dG = -SdT + VdP + \sum_{i=1}^{NC} \mu_{c_i} dn_i = 0 \tag{2-39}$$

Since at equilibrium temperature and pressure are constant, the previous equation leads to:

$$\sum_{i=1}^{NC} \mu_{c_i} dn_i = 0 \tag{2-40}$$

Now we assume that the system consists of NP phases. Equation (2-40) becomes:

$$\sum_{j=1}^{NP} \sum_{i=1}^{NC} \mu_{c_{i,j}} dn_{i,j} = 0 \tag{2-41}$$

Since at equilibrium a change in moles of each component is only due to mass transfer between phases and therefore obeys the conservation principle, we can write

$$dn_{i,j} = \sum_{\substack{l=1 \\ l \neq j}}^{NP} dn_{i,jl} \tag{2-42}$$

in which $dn_{i,jl}$ denotes mole transfer of component i from phase j to l.

In addition, at equilibrium for each two phases, we can write:

$$dn_{i,jl} = dn_{i,lj} \tag{2-43}$$

Therefore, Equation (2-41), after factoring, becomes:

$$\sum_{j=1}^{NP} \sum_{\substack{l=1 \\ l \neq j}}^{NP} \sum_{i=1}^{NC} \left(\mu_{c_{i,j}} - \mu_{c_{i,l}} \right) dn_{i,jl} = 0 \tag{2-44}$$

As $dn_{i,jl}$ are independent and arbitrary, the only way the left side of this equation can in general be zero is for each term in parentheses separately to be zero. Therefore, for each two phases:

$$\mu_{c_{i,j}} = \mu_{c_{i,l}} \quad i = 1, \dots, NC; \quad j, l = 1, \dots, NP \tag{2-45}$$

While fundamental, chemical potential is not very convenient for use in solving problems. For this reason, *fugacity* is introduced.

The fugacity of a real gas is an effective partial pressure, which replaces the mechanical partial pressure in computations related to chemical equilibrium. For an ideal gas, fugacity is exactly equal to gas partial pressure. The ratio of fugacity of a gas to its partial pressure is called the *fugacity coefficient*. A mathematical definition of fugacity is given next.

From Equation (2-21), for an isothermal process, we can write:

$$\left(\frac{\partial G}{\partial P}\right)_T = V \tag{2-46}$$

Integrating gives:

$$G_2 - G_1 = \int_{P_1}^{P_2} V \, dp \tag{2-47}$$

For an ideal gas, where $V = RT/P$, we have:

$$G_2 - G_1 = RT \ln \frac{P_2}{P_1} \tag{2-48}$$

Fugacity is defined as the replacement for pressure to make this expression applicable to real gases:

$$G_2 - G_1 = RT \ln \frac{f_2}{f_1} \tag{2-49}$$

The fugacity of a component in a mixture is defined by a similar expression:

$$\bar{G}_{i_2} - \bar{G}_{i_1} = RT \ln \frac{f_{i_2}}{f_{i_1}} \tag{2-50}$$

By using Equation (2-38), we have

$$\mu_{c_{i_2}} - \mu_{c_{i_1}} = RT \ln \frac{f_{i_2}}{f_{i_1}} \tag{2-51}$$

in which, i designates any component in the mixture and subscripts 1 and 2 indicate two points in an isotherm.

At equilibrium we may choose points 1 and 2 to be phases j and l, such that

$$\mu_{c_{i,j}} - \mu_{c_{i,l}} = RT \ln \frac{f_{i,j}}{f_{i,l}} \tag{2-52}$$

From Equation (2-45), we readily obtain at equilibrium:

$$\mu_{c_{i,j}} - \mu_{c_{i,l}} = RT \ln \frac{f_{i,j}}{f_{i,l}} = 0 \tag{2-53}$$

Since neither the temperature nor the gas constant is zero:

$$\ln \frac{f_{i,j}}{f_{i,l}} = 0 \rightarrow f_{i,j} = f_{i,l} \tag{2-54}$$

If we repeat this procedure for each two phases in a system with NP phases, we can write:

$$f_{i,1} = f_{i,2} = \cdots = f_{i,NP} \tag{2-55}$$

This proves that at equilibrium the fugacities of each specific component are equal between any two phases.

2.2.1 Fugacity Method in Calculating Equilibrium Ratios

In this sub-section, we derive an equation for obtaining equilibrium ratios from fugacity coefficients. Then, we develop an expression for obtaining fugacity coefficients based on phase and components properties.

If we write the definition of fugacity coefficients at equilibrium for component i in phase j, we have

$$\phi_{i,j} = \frac{f_{i,j}}{x_{i,j} P_j} \tag{2-56}$$

where $\phi_{i,p}$ is the fugacity coefficient of component i in phase p.

Writing a similar equation for component i in phase l and dividing the two fugacity coefficients, yields:

$$\frac{\phi_{i,j}}{\phi_{i,l}} = \frac{\left(\frac{f_{i,j}}{x_{i,j} P_j}\right)}{\left(\frac{f_{i,l}}{x_{i,l} P_l}\right)} \tag{2-57}$$

Since at equilibrium fugacities are equal, we obtain:

$$\frac{\phi_{i,j}}{\phi_{i,l}} = \frac{x_{i,l}}{x_{i,j}} \frac{P_l}{P_j} \tag{2-58}$$

Now we can rearrange the above equation to obtain an expression for the equilibrium ratio:

$$K_{i,jl} = \frac{x_{i,j}}{x_{i,l}} = \frac{\phi_{i,l}}{\phi_{i,j}} \frac{P_l}{P_j} \tag{2-59}$$

Therefore, given knowledge of phase pressures and component fugacity coefficients, the distribution coefficients or equilibrium ratios can be found from Equation (2-59). Note that phase pressures are connected via capillary pressure relationships (Equation (2-10)). Note also that one phase pressure and one or two saturations, depending on *state*, are primary variables (see Table 2-1), available from *CompFlow Bio* as input. Therefore, to complete our calculations, we require only a means to calculate fugacity coefficients.

For our purpose, we divide G by RT and differentiate to obtain:

$$d\left(\frac{G}{RT}\right) = \frac{1}{RT} dG - \frac{G}{RT^2} dT \tag{2-60}$$

In the above equation, we substitute dG from Equation (2-21) and G from Equation (2-18), and after algebraic reduction; we arrive at the following equation for phase j:

$$d\left(\frac{G_j}{RT}\right) = \frac{V_j}{RT} dP_j - \frac{H_j}{RT^2} dT + \sum_{i=1}^{NC} \frac{\bar{G}_{i,j}}{RT} dn_{i,j} \tag{2-61}$$

G_j is the Gibbs free energy of phase j and $\bar{G}_{i,j}$ is the molar Gibbs free energy of component i in phase j.

Before we continue our derivation, we need to define an important property in thermodynamics, namely the *residual Gibbs free energy*, by the following equation

$$G^R \equiv G - G^{ig} \tag{2-62}$$

in which the superscripts R and ig stand for residual and ideal gas, respectively.

G and G^{ig} are the actual and the ideal-gas values of the Gibbs energy at the same temperature, pressure, and composition.

Defining residual thermodynamic properties is very helpful in solving thermodynamic problems. In general, residual properties are defined as follows

$$M^R \equiv M - M^{ig} \tag{2-63}$$

where M can be any thermodynamic property.

As Equation (2-61) is general, it may be written for the special case of an ideal gas:

$$d\left(\frac{G_j^{ig}}{RT}\right) = \frac{V_j^{ig}}{RT}dP_j - \frac{{H_j}^{ig}}{RT^2}dT + \sum_{i=1}^{NC}\frac{\bar{G}_{i,j}^{ig}}{RT}dn_{i,j} \tag{2-64}$$

If we subtract this equation from Equation (2-61), we have:

$$d\left(\frac{G_j^{R}}{RT}\right) = \frac{V_j^{R}}{RT}dP_j - \frac{H_j^{R}}{RT^2}dT + \sum_{i=1}^{NC}\frac{\bar{G}_{i,j}^{R}}{RT}dn_{i,j} \tag{2-65}$$

Equation (2-65) is the *fundamental residual property relation*.

If we apply conditions of constant pressure and temperature to Equation (2-65), we can write:

$$\left[\frac{\partial\left(G_j^R/RT\right)}{\partial n_j}\right]_{P_j,T,n_{l\neq i}} = \frac{G_{i,j}^R}{RT} \tag{2-66}$$

In Equation (2-65) for residual volume, we can write:

$$V_j^R = \mathrm{V_j} - V_j^{ig} \tag{2-67}$$

For an ideal gas, we can write the ideal gas law

$$V_j^{ig} = \frac{n_j RT}{P_j} \tag{2-68}$$

whereas for a real gas

$$V_j = \frac{n_j z_j RT}{P_j} \tag{2-69}$$

where z is the *compressibility factor*. For an ideal gas z is equal to one. Furthermore, $n_j = \sum_{i=1}^{NC} n_{i,j}$.

Therefore, for residual volume, we have:

$$V_j^R = \frac{n_j z_j RT}{P_j} - \frac{n_j RT}{P_j} \tag{2-70}$$

If we substitute Equation (2-70) in Equation (2-65) and consider constant temperature and composition, we can write

$$\mathrm{d}\left(\frac{G_j^R}{RT}\right) = \left(n_j z_j - n_j\right)\frac{dP_j}{P_j} \tag{2-71}$$

from which we derive by integration:

$$\frac{G_j^R}{RT} = \int_0^{P_j} (n_j z_j - n_j) \frac{dP_j}{P_j} \tag{2-72}$$

Now we recall Equation (2-49):

$$G_2 - G_1 = RT \ln \frac{f_2}{f_1}$$

If we choose point 1 to be an ideal gas and point 2 to be a real gas with same composition and temperature, for component i in phase j, we have:

$$G_{i,j} - G_{i,j}^{ig} = RT \ln \frac{f_{i,j}}{f_{i,j}^{ig}} \tag{2-73}$$

Since for an ideal gas fugacity equals partial pressure, it follows that

$$G_{i,j} - G_{i,j}^{ig} = RT \ln \frac{f_{i,j}}{x_{i,j} P_j} \tag{2-74}$$

where x is the mole fraction of component i in phase j. The left hand side of the above equation is the definition of residual Gibbs free energy and the argument of the logarithm is the fugacity coefficient. Thus, we can write:

$$\frac{G_{i,j}^R}{RT} = \ln \phi_{i,j} \tag{2-75}$$

In combination with Equation (2-66), we have:

$$\left[\frac{\partial \left(G_j^R / RT\right)}{\partial n_j}\right]_{P_j, T, n_{l \neq i}} = \ln \phi_{i,j} \tag{2-76}$$

If we differentiate Equation (2-72) with respect to $n_{i,j}$:

$$\left[\frac{\partial \left(G_j^R / RT\right)}{\partial n_j}\right]_{P_j, T, n_{l \neq i}} = \int_0^{P_j} \left[\frac{\partial \left(n_j z_j - n_j\right)}{\partial n_{i,j}}\right]_{P_j, T, n_{l \neq i}} \frac{dP_j}{P_j} \tag{2-77}$$

If we define $\overline{z}_j \equiv \left[\frac{\partial (n_j z_j)}{\partial n_{i,j}}\right]_{P_j, T, n_{l \neq i}}$, and use Equation (2-76), then we can write:

$$\ln \phi_{i,j} = \int_0^{P_j} \left(\overline{z}_j - 1\right) \frac{dP_j}{P_j} \tag{2-78}$$

Equation (2-78) is a key equation in our procedure. In this equation, $\overline{z}_j$ is calculated from an equation of state (EOS) (McCain, 1990). As there are numerous equations of state (McCain, 1990), selecting a suitable equation can be application-specific. The Soave-Redlich-Kwong (Soave, 1972) and Peng-Robinson (Peng and Robinson, 1976a) EOS, which are cubic equations with two empirical constants, are two popularly accepted equations of state in the petroleum industry. While these cubic equations have been used widely to calculate physical properties and vapor-liquid equilibria of hydrocarbon mixtures (McCain, 1990; Tarek, 2009) and both provide accurate equilibrium mole fractions, the Peng-Robinson EOS provides better liquid phase density, particularly near the critical region (Danesh, 1998; Peng and Robinson, 1976a; Tarek, 2007). In this work, we adopted a modified Peng-Robinson equation of state, in which a third parameter is introduced for volumetric correction (Jhaveri and Youngren, 1988; Péneloux et al., 1982; Peng and Robinson, 1976a).

With compressibility factor calculated from the selected equation of state, fugacity coefficients can be obtained from Equation (2-78). Then, equilibrium ratios are calculated by Equation (2-59).

Once all equilibrium ratios are calculated, secondary mole fractions and subsequently phase densities can be obtained.

In the next chapter, detailed description of the selected EOS along with related equations is provided.

2.2.2 Free Gibbs Energy Minimization Method in Calculating Equilibrium Ratios

In the previous sub-section, we discussed the fugacity method for obtain equilibrium ratios. The fugacity method is based on the thermodynamic fact that Gibbs free energy is minimized at equilibrium. This fact may be used directly, giving rise to a method which directly minimizes Gibbs free energy of a mixture to obtain equilibrium ratios, as explained below.

For a system of NP phases with NC components at equilibrium (constant pressure and temperature), we have from Equation (2-27):

$$dG = \sum_{j=1}^{NP}\sum_{i=1}^{NC}\left(\frac{\partial G}{\partial n_{i,j}}\right)_{P,T,n_{l\neq i}} dn_{i,j} \tag{2-79}$$

Combining with Equations (2-31) and (2-38), results in the following equation

$$dG = \sum_{j=1}^{NP}\sum_{i=1}^{NC}\mu_{c_{i,j}} dn_{i,j} \tag{2-80}$$

which after integration gives:

$$G = \sum_{j=1}^{NP}\sum_{i=1}^{NC} \mu_{c_{i,j}} n_{i,j} \tag{2-81}$$

In this equation, G is the function to be minimized.

Now we find an expression for $\mu_{c_{i,j}}$ to be substituted in the above equation. If we rewrite the Equation (2-51) for component i in phase j for current and standard conditions, we have

$$\mu_{c\,i} = \mu_{c_i}^{0}(T) + RTln\left(\frac{f_i}{f_i^{0}(T)}\right) \tag{2-82}$$

where μ_i^0 and f_i^0 are the chemical potential and fugacity at standard conditions, respectively. These standard properties are only functions of temperature. Therefore, the equation, which should be minimized, becomes

$$G = \sum_{i=1}^{NC} n_i\left[\mu_{c_i}^{0}(T) - RTln\left(f_i^{0}\right)\right] + RT\sum_{j=1}^{NP}\sum_{i=1}^{NC} n_{i,j}\ln(f_{i,j}) \tag{2-83}$$

in which n_i is total mole number of component i in the mixture. The first term on the right hand side of the above equation is a constant at equilibrium and the second term is a function of $n_{i,j}$. Therefore, there are $NC \times NP$ variables.

Minimizing Equation (2-83) is a constrained optimization problem with following constraints:

$$\Gamma_i = n_i - \sum_{j=1}^{NP} n_{i,j} = 0 \quad i = 1,\ \ldots, NC \tag{2-84}$$

$$0 \le n_{i,j} \le n_i \qquad i = 1, \ldots, NC; j = 1, \ldots, NP \tag{2-85}$$

One method to solve this constrained optimization problem is to use a penalty function method (Izadpanah et al., 2006). The idea is to convert the constrained problem into a sequence of unconstrained problems. Therefore the new objective function is

$$\begin{aligned}\Omega\left(n_{i,j}, r\right) &= G\left(n_{i,j}\right) + \frac{1}{\sqrt{r}}\sum_{i=1}^{NC}\Gamma_i^2 \\ &= \sum_{i=1}^{NC} n_i\left[\mu_{c_i}^{0}(T) - RTln(f_I^{0})\right] \\ &+ RT\sum_{j=1}^{NP}\sum_{i=1}^{NC} n_{i,j}\ln\left(f_{i,j}\right) + \frac{1}{\sqrt{r}}\sum_{i=1}^{NC}\Gamma_i^2\end{aligned} \tag{2-86}$$

where r is a positive real number, which forms a monotonically decreasing sequence during the process of minimization. It can be proven (Rao, 1996) that

$$\lim_{r\to 0} \underline{n}(r) = \underline{n}^* \tag{2-87}$$

in which $\underline{n}$ is the minimum of the new function Ω for a given r and $\underline{n}^*$ is the solution of the initial problem (Equation (2-83)).

Minimizing the Gibbs free energy is particularly useful when there is chemical reaction in the mixture, but in comparison to the fugacity method mentioned in the previous sub-section, is computationally more demanding.

For the purposes of this research, performance is a key issue and since chemical reactions are absent, we choose the fugacity method for equilibrium ratios calculations.

2.3 Three Phase Compositional Modeling of Oil Recovery using CO_2

Practical reservoir engineering problems are associated with high pressures and temperatures. Since it is common for oil reservoirs to be found at depths of one kilometer or more, dealing with pressures in excess of 100 bar (or 10 MPa) and temperatures greater than 100 ℃ cannot be avoided.

A wide range of reservoir engineering problems involves using carbon dioxide for secondary or enhanced oil recovery. CO_2 can be injected as a supercritical fluid above minimum miscibility pressure to reduce water-flood residual oil saturation (enhanced oil recovery) (Lake, 1989) or injected dissolved in water to improve water-flood performance (secondary oil recovery) (Seyyedi and Sohrabi, 2017) or for enhanced oil recovery of water-flood residual (Alizadeh et al., 2014).

In addition to dealing with high pressure and temperature, robust handling of fractures is often needed. *CompFlow Bio* is a unique simulator with regard to the handling of fractures, which distinguishes it from other well-known compositional simulators such as *Eclipse*® or *WinProp*®. By introducing a package, which improves compositional calculations of *CompFlow Bio*, this simulator will have all tools required for robust and accurate simulation of reservoir problems. The kind of reservoir engineering problems this research aims to ultimately address has already come under investigation by a number of authors in recent years.

Darvish (2007) has performed three sets of experiments to investigate the role of initial water saturation and temperature in CO_2 injection, as well as to demonstrate key mechanisms in secondary and tertiary recovery processes in fractured rocks. In their research, they injected supercritical CO_2 into a North Sea chalk core. Chalk reservoirs are known for their high porosity and low permeability. They

used an annulus space surrounding the core as fracture and performed CO_2 injection with zero and 26.3% initial water saturation at 130 ℃. They also performed a tertiary CO_2 injection at 60 ℃. As a result of their work, they suggested that strong compositional behavior of produced oil in all experiments must be considered in any EOR assessment that considers CO_2 as injection fluid. To consider strong compositional behavior of hydrocarbons in reservoir condition proper calculation of the equilibrium ratios is vital. Attempts by Darvish (2007) to model their experiments using commercial software (Eclipse®) were unsuccessful.

Karimaie et al. (2007) investigated the effect of secondary and tertiary CO_2 injection at high pressure and temperature. They used an outcrop chalk as porous medium in a setup similar to the one used by Darvish (2007). Their results show high efficiency of CO_2 injection both in secondary and tertiary injection. In another work, Karimaie and Torsaeter (2010) investigated the effect of tertiary methane and CO_2 injection at two different porous media, an outcrop chalk and a limestone. They found CO_2 injection to be more efficient. In addition, water injection prior to tertiary gas injection was found to be very efficient in water-wet systems, while in mixed-wet systems this sequence of injection is not an efficient recovery process.

Moortgat et al. (2009) used an improved model to simulate results obtained by Darvish (2007). They simulated one of Darvish's experiments, which has no initial water saturation and also did not include capillary pressure in their model. These limitations highlight the potential significance of our work: currently, in its current state *CompFlow Bio* includes capillary pressure in its calculations and can handle water presence as well, while efficiently handling discrete fracture networks. In its current state, however, *CompFlow Bio* lacks proper thermodynamics to deal with high pressure and high temperature problems. This thesis provides *CompFlow Bio* with means to handle such problems accurately.

In a multi-scale three-phase study, Alizadeh et al. (2014) performed macro and micro scale experiments to investigate the effect of carbon dioxide exsolution during tertiary carbonated water injection. In macro scale setup, first a secondary water-flood injection was performed on a 25.4 cm length Berea sandstone at pressure of 90 $psig$ and ambient temperature. Then carbonated water injection was carried along with reduction of system pressure. System pressure was reduced in small increments until 88 psi pressure drop across the core was observed. In micro scale experiments, a similar procedure was followed on smaller 9.5 cm length Berea sandstone, as well as on a series of flow experiments in two-dimensional etched-glass micromodels. During both micro and macro scale experiments, the porosity and in situ fluid saturations were determined using X-ray techniques. Tertiary carbonated water injection, under described configuration, resulted in 34.6% and 40.7% additional oil recovery in macro and micro scale experiments, respectively. The increase in pressure drop led to exsolution of carbon

dioxide and internal gas drive, causing mobilization of oil ganglia, and reduction in water-flood residual oil saturation. In addition to improving oil recovery, Alizadeh et al. (2014) proposed entrapment of carbon dioxide as free gas as well as dissolved gas in the aqueous phase at the end of carbonated water injection process, as a potentially effective scheme for geological carbon dioxide sequestration in petroleum reservoirs.

In a recent work, Seyyedi and Sohrabi (2017) used a glass micromodel setup to investigate the effect of carbonated water on oil recovery during secondary and tertiary injection at pressure of 2500 $psia$ and temperature equal to 100 °F. Using a high-resolution camera, authors visually monitored the flow of each phase as well as their appearance or disappearance. Based on their results, Seyyedi and Sohrabi (2017) concluded that carbonated water injection as a secondary process leads to better oil recovery in comparison to post water-flood injection of carbonated water. The main mechanisms for improving oil recovery by carbonated water injection suggested by the authors are: (i) reconnection of the trapped oil and oil displacement, (ii) creating a favorable three-phase flow region with less residual oil saturation, and (iii) restricting the flow path of carbonated water and diverting it toward un-swept areas of the porous medium.

The above-mentioned studies highlight the importance of injecting carbon dioxide, as a pure phase or dissolved in water, at high pressures and temperatures for the purpose of secondary or tertiary oil recovery. In simulating these complex processes, a first step for any simulator is to have robust thermodynamics and efficient equilibrium calculations. Even in non-equilibrium processes with finite mass transfer rate, equilibrium calculations are required. In this work, we provide *CompFlow Bio* with a package that performs equilibrium calculations in a wide range of pressure and temperature.

Chapter 3
Methodology

3.1 Introduction

CompFlow Bio is a multi-phase multi-component simulator implementing a control volume (CV) approach. To each CV, there corresponds a node, with which all the characteristics of the CV are associated. For each node, sets of governing conservation equations are solved iteratively. In each iteration, the mole fraction of each component in each phase is required. As mentioned earlier, *CompFlow Bio* divides the unknown variables to the primary and secondary variables. Primary variables are set at the beginning of each iteration based on *state* of the node, passed by previous iteration, and secondary variables are found based on the given primary variables throughout the iteration (see Figure 2-1). In this thesis, our goal is to provide *CompFlow Bio* with a more precise function to calculate these secondary variables at conditions of chemical equilibrium condition over a wide range of pressures and temperatures.

In this chapter, we first list primary variables in different cases and then expose in detail the equations and procedure for calculating secondary variables based on primary variables.

In *CompFlow Bio,* it is given that the gas phase is always present even in very small quantity. Furthermore, the maximum number of phases present is three and all phases are fluids. These phases are called a gaseous phase, an aqueous phase, and a non-aqueous phase. Based on these assumptions, we can have four different combinations of phases. We call each case a State. These states are described below:

State 1: Gaseous, Aqueous, and Non-Aqueous phases are present,

State 2: Gaseous Aqueous phases are present,

State 3: Only Gaseous is present,

State 4: Gaseous and Non-Aqueous phases are present.

Each state has its own set of primary variables as listed in Table 2-1.

These primary variables are input to our code and all other unknowns, which are other mole fractions, pressures and saturations that are not mentioned in the Table 2-1 are secondary variables and should be calculated in our code.

In each state, the number of given saturations is one less than the number of present phases. Thus, the unknown saturation, which is a secondary variable, could be calculated using Equation (2-6):

$$\sum_{l=n,q,g} S_l = 1$$

Pressures that are secondary variables are calculated using capillary pressure functions from knowledge of phase saturations.

The most important task concerns the calculation of unknown mole fractions, i.e. secondary mole fractions.

In order to calculate secondary mole fractions, we should recall Equation (2-59):

$$K_{i,jl} = \frac{x_{i,j}}{x_{i,l}} = \frac{\phi_{i,l}}{\phi_{i,j}} \frac{P_l}{P_j}$$

We know the pressure and some of the mole fractions in each phase. Therefore, if we know the equilibrium ratios, we can calculate mole fraction of each component in other phases. In other words, knowing one mole fraction and the partition coefficients for each component, we can calculate the mole fraction of that component in other phases.

For calculating equilibrium ratios, we implement a fixed-point iteration procedure. However, before that procedure is explained further, it is important to note the number of primary mole fractions we get as input from *CompFlow Bio*.

The total number of components in each phase is $m + 4$, with $m + 1$ of them being hydrocarbons, the other three components being carbon dioxide, water and nitrogen.

There are, however, components for which the mole fractions in any phase are not known as input from *CompFlow Bio*. These components are listed below:

State 1: water, nitrogen and the heaviest non-methane hydrocarbon component (o_m),

State 2: water and nitrogen,

State 3: nitrogen,

State 4: nitrogen and the heaviest non-methane hydrocarbon component (o_m).

We note that the number of these unknowns in each case is equal to the number of phases. Also from Equation (2-7), we have:

$$\sum_{i=1}^{NC} x_{i,l} = 1 \quad l = n, q \; or \; g$$

This means that, once all other mole fractions have been calculated using partition coefficients, by writing Equation (2-7) for each phase, we can find the other component mole fractions as well. Depending on the number of phases, we may solve a linear algebraic equation (*state 3*) or a system of linear algebraic equations (in other cases), as explained below.

For the case of *state 1*, we can write the system of linear algebraic equations as follows:

$$\begin{bmatrix} x_{w,n} \\ x_{N_2,n} \\ x_{o_m,n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ K_{w,ng} & K_{N_2,ng} & K_{o_m,ng} \\ \dfrac{K_{w,ng}}{K_{w,qg}} & \dfrac{K_{N_2,ng}}{K_{N_2,qg}} & \dfrac{K_{o_m,ng}}{K_{o_m,qg}} \end{bmatrix} \begin{bmatrix} 1 - \sum\limits_{\substack{i=1 \\ i \neq w,N_2,o_m}}^{NC} x_{i,n} \\ 1 - \sum\limits_{\substack{i=1 \\ i \neq w,N_2,o_m}}^{NC} x_{i,g} \\ 1 - \sum\limits_{\substack{i=1 \\ i \neq w,N_2,o_m}}^{NC} x_{i,q} \end{bmatrix} \tag{3-1}$$

If only gaseous and aqueous phases are present (*state 2*), the following equations should be solved simultaneously

$$\begin{bmatrix} x_{w,q} \\ x_{N_2,q} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ K_{w,qg} & K_{N_2,qg} \end{bmatrix} \begin{bmatrix} 1 - \sum\limits_{\substack{i=1 \\ i \neq w,N_2}}^{NC} x_{i,q} \\ 1 - \sum\limits_{\substack{i=1 \\ i \neq w,N_2}}^{NC} x_{i,g} \end{bmatrix} \tag{3-2}$$

If only the gaseous phase is present (*state 3*), solving the following linear equation gives the only unknown mole fraction.

$$x_{N_2,g} = 1 - \sum_{\substack{i=1 \\ i \neq N_2}}^{NC} x_{i,g} \tag{3-3}$$

Finally, for the *state 4*, the following system of equations should be solved:

$$\begin{bmatrix} x_{N_2,n} \\ \\ x_{o_m,n} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \\ K_{N_2,ng} & K_{o_m,ng} \end{bmatrix} \begin{bmatrix} 1 - \sum_{\substack{i=1 \\ i \neq N_2,o_m}}^{NC} x_{i,n} \\ \\ 1 - \sum_{\substack{i=1 \\ i \neq N_2,o_m}}^{NC} x_{i,g} \end{bmatrix} \tag{3-4}$$

We are now in a position to describe the fixed-point iteration procedure used for calculating partition coefficients.

3.2 Fixed-Point Iteration Procedure

As a first step, we use appropriate initial guess for partition coefficients to start the iteration.

In this work, we used Wilson equation (Wilson, 1969) for $K_{i,gn}$

$$K_{i,gn} = \frac{1}{P_{r_i}} exp\left[5.37(1+\omega_i)\left(1-\frac{1}{T_{r_i}}\right)\right] \tag{3-5}$$

where P_{r_i} and T_{r_i} are reduced pressure and temperature of component i, respectively and ω_i is the acentric factor of that component (dimensionless). Reduced parameters are defined as follows:

$$P_{r_i} \equiv \frac{P}{P_{c_i}} \tag{3-6}$$

$$T_{r_i} \equiv \frac{T}{T_{c_i}} \tag{3-7}$$

To obtain initial values for $K_{i,gq}$, the following equation from Peng and Robinson (1976b) is used:

$$K_{i,gq} = 1 \times 10^6 \left(\frac{P_{r_i}}{T_{r_i}}\right) \tag{3-8}$$

Once we have all the initial values of partition coefficients, we can calculate secondary mole fractions from Equations (2-59) and (2-7).

Now with knowing all mole fractions for all components in all phases, we use modified Peng-Robinson equation of state to calculate compressibility factors and thereafter fugacity coefficients using Equation (2-78). After obtaining the fugacity coefficients, we can calculate new sets of equilibrium ratios using Equation (2-59). Then we compare these new sets with old ones. If the difference is less than a pre-set tolerance, our secondary variables are considered accurate and are passed to *CompFlow Bio*. Otherwise, we replace the old set of equilibrium ratios with the new one and re-iterate the loop with

calculating new mole fractions based on new equilibrium ratios. This procedure is described in the flowchart of Figure 3-1.

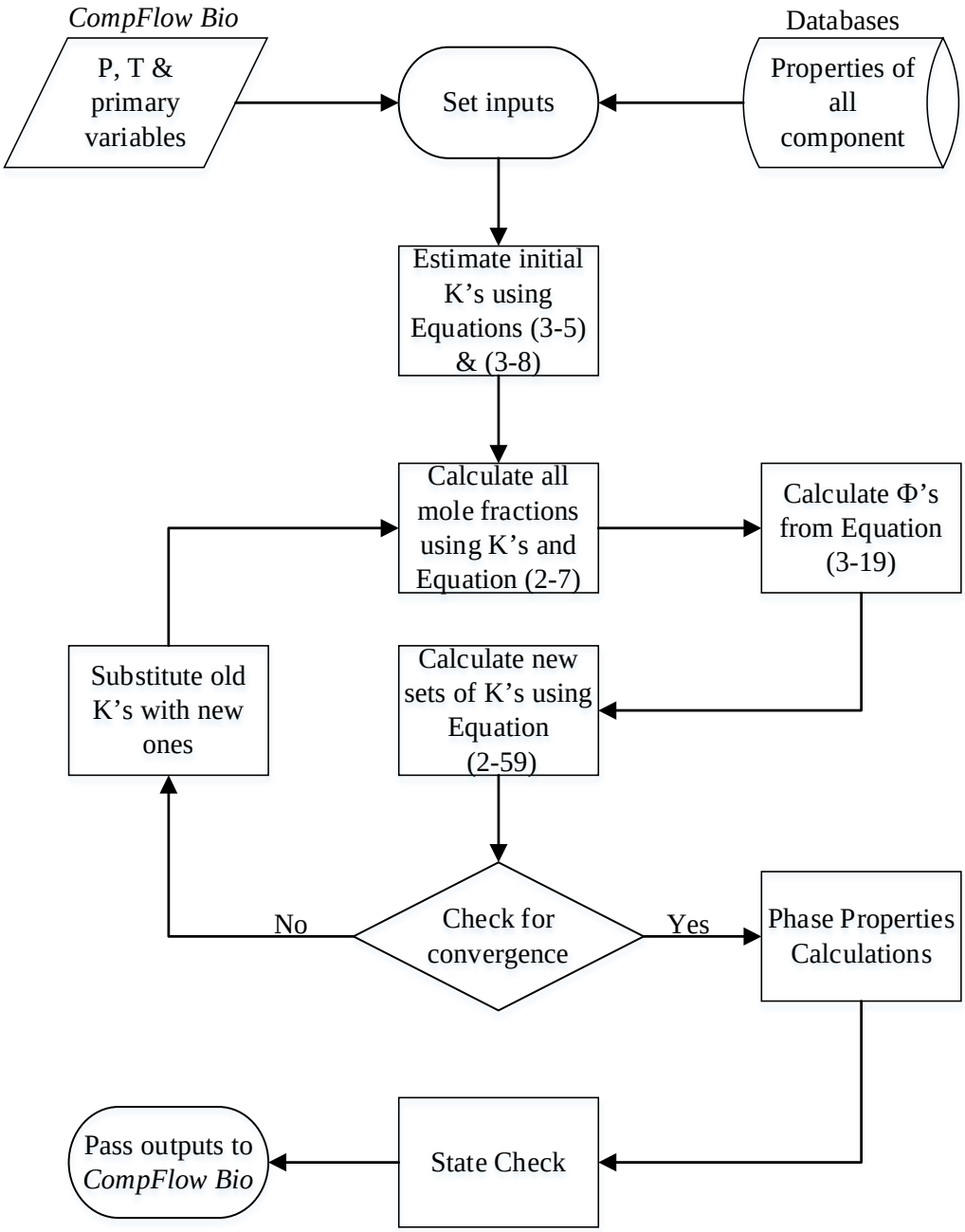

Figure 3-1: Flowchart of different steps in flash calculations using Peng-Robinson EOS.

In this thesis, the following equation is used for estimating error, e:

$$e = max\left[\begin{array}{c}\dfrac{\left(K_{i,gn}^{new} - K_{i,gn}\right)^2}{K_{i,gn}^{new} \times K_{i,gn}} \\ \\ \dfrac{\left(K_{i,gq}^{new} - K_{i,gq}\right)^2}{K_{i,gq}^{new} \times K_{i,gq}}\end{array}\right] \quad i = 1,2,\dots,NC \tag{3-9}$$

This error is then ensured to be less than a pre-set tolerance. Value of the tolerance used in this work is equal to 1×10^{-16}.

As mentioned before, a single point iteration procedure is employed to solve coupled nonlinear algebraic equations. In this method, convergence is not guaranteed and depends sensitively on initial guesses, which should be selected carefully. Equations (3-5) and (3-8), are appropriate guesses for hydrocarbon systems (Danesh, 1998) and help the iteration procedure converge quickly.

3.3 Calculating Fugacity Coefficients using Peng-Robinson EOS

In this section, the Peng-Robinson equation of state and its associated variables is explained. First, we review two-parameter Peng-Robinson EOS and then in subsequent sections, we introduce modifications suggested to improve the performance of this cubic equation.

The Peng-Robinson equation of state (1976a) was developed at the University of Alberta by Ding-Yu Peng and Donald Robinson in order to satisfy the following goals:

- The parameters should be expressible in terms of the critical properties and the acentric factor.
- The model should provide reasonable accuracy near the critical point, particularly for calculations of the compressibility factor and liquid density.
- The mixing rules should not employ more than a single binary interaction parameter, which should be independent of temperature, pressure, and composition.
- The equation should be applicable to all calculations of all fluid properties in natural gas processes.

The original formulation of Peng-Robinson EOS was later modified by them (1976b; 1980; 1978) and others (Jhaveri and Youngren, 1988; Péneloux et al., 1982) to improve its prediction and range of applicability.

The two-parameter Peng-Robinson equation of state (1976a) has the following formulation

$$P = \frac{RT}{V_M - b} - \frac{a\alpha}{V_M^2 + 2bV_M - b^2} \tag{3-10}$$

in which V_M is molar volume (SI unit: $\frac{m^3}{mol}$) and:

$$a = 0.45724 \frac{R^2 T_c^2}{P_c} \tag{3-11}$$

$$b = 0.07780 \frac{RT_c}{P_c} \tag{3-12}$$

$$\alpha = \left(1 + \kappa\left(1 - \sqrt{T_r}\right)\right)^2 \tag{3-13}$$

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \tag{3-14}$$

a and b are two empirical constant of this equation of state, obtained from fluid properties.

From real gas law we have (Smith et al., 2001):

$$P = \frac{zRT}{V_M} \tag{3-15}$$

Combining this equation with Equation (3-10) and solving for compressibility factor yields

$$z = \frac{1}{1 - \frac{b}{V_M}} + \frac{a'\alpha}{V_M + 2b - \frac{b^2}{V_M}} \tag{3-16}$$

in which we defined

$$a' = \frac{a}{RT} \tag{3-17}$$

Now we recall Equation (2-78):

$$\ln \phi_{i,j} = \int_0^{P_j} \left(\overline{z}_j - 1\right) \frac{dP_j}{P_j}$$

If we substitute $\overline{z}_j$ in this equation from Equation (3-16), we will have an expression for calculation of fugacity coefficient based on Peng-Robinson EOS. We note, however, that before this substitution we have to change the integration variable in Equation (2-78) from P_j to V_j. For this purpose, we use Equation (3-15) to obtain:

$$\ln \phi_{i,j} = \int_\infty^{V_j} \left(1 - \overline{z}_j\right) \frac{dV_j}{V_j} \tag{3-18}$$

Now if we rewrite Equation (3-16) for phase j and combine the resulting equation with Equation (3-18), we obtain after algebraic manipulation an expression for the fugacity coefficient of component i in phase j (McCain, 1990):

$$\ln(\phi_{i,j}) = -\ln(z_j - B_j) + (z_j - 1)B'_{i,j} - \frac{A_j}{2\sqrt{2}B_j}(A'_{i,j} - B'_{i,j}) * ln\left[\frac{z_j + (\sqrt{2}+1)B_j}{z_j - (\sqrt{2}-1)B_j}\right] \quad (3\text{-}19)$$

In Equation (3-19), phase-component variables, $A'_{i,j}$ and $B'_{i,j}$, are defined as follows

$$A'_{i,j} = \frac{1}{a_j}\left[2\sqrt{a_i}\sum_{l=1}^{NC} x_{l,j}\sqrt{a_l}(1-\delta_{li})\right] \quad (3\text{-}20)$$

$$B'_{i,j} = \frac{b_i}{b_j} \quad (3\text{-}21)$$

where the following mixing rules are used for phase j:

$$a_j = \sum_{i=1}^{NC}\sum_{l=1}^{NC} x_{i,j}x_{l,j}\sqrt{a_i a_l}(1-\delta_{il}) \quad (3\text{-}22)$$

$$b_j = \sum_{l=1}^{NC} x_{l,j}\, b_l \quad (3\text{-}23)$$

δ_{ij} is the binary interaction coefficient between component i and j. As a first approximation, it may be assumed constant for each two components and is calculated for each pair of components in their binary mixture in order to match experimental data. The databases that we used for critical properties, as well as acentric factors and binary interaction coefficients are given in Chapter 4.

Component-dependent variables used in the above formulations are given below, where each equation is written for component i:

$$a_i = a_{c_i}\alpha_i \quad (3\text{-}24)$$

$$a_{c_i} = 0.45724\frac{R^2 T_{c_i}^{\,2}}{P_{ci}} \quad (3\text{-}25)$$

$$\alpha_i = \left(1 + \kappa_i\left(1 - \sqrt{T_{r_i}}\right)\right)^2 \quad (3\text{-}26)$$

$$\kappa_i = 0.37464 + 1.54226\omega_i - 0.26992\omega_i^2 \tag{3-27}$$

$$b_i = 0.07780\frac{RT_{c_i}}{P_{c_i}} \tag{3-28}$$

The phase variables, A_j and B_j, are calculated by:

$$A_j = \frac{a_j P_j}{R^2 T^2} \tag{3-29}$$

$$B_j = \frac{b_j P_j}{RT} \tag{3-30}$$

The compressibility factor required in Equation (3-19) is obtained as follows. First substitute V_M in Equation (3-16) by real gas law and then solve the resulting equation for z. Applied to phase *j*, the equation to solve is (McCain, 1990):

$$z_j^3 - \left(1 - B_j\right)z_j^2 + \left(A_j - 2B_j - 3B_j^2\right)z - \left(A_j B_j - B_j^2 - B_j^3\right) = 0 \tag{3-31}$$

Equation (3-31) is a cubic function and has exact analytical solutions (Abramowitz and Stegun, 1964), therefore for phase *j* we can write

$$r_m = -\frac{1}{3}\left(B_j - 1 + \zeta^m\eta + \frac{\Delta_0}{\zeta^m\eta}\right), \quad m\epsilon\{0,1,2\} \tag{3-32}$$

where

$$\zeta = -\frac{1}{2} + \frac{1}{2}\sqrt{3}i \tag{3-33}$$

$$\eta = \sqrt[3]{\frac{\Delta_1 \pm \sqrt{{\Delta_1}^2 - 4{\Delta_0}^3}}{2}} \tag{3-34}$$

$$\Delta_{0=} 10{B_j}^2 + 4B_j - 3A_j + 1 \tag{3-35}$$

$$\Delta_1 = 56{B_j}^3 + 12{B_j}^2 - 12B_j + 9A_j - 36A_jB_j - 2 \tag{3-36}$$

In the above, r_m denotes the *m*-th root of the cubic function. It should be noted that in Equation (3-33), i is the unit imaginary number which is equal to $\sqrt{-1}$.

As Equation (3-31) is a third degree polynomial, it has three roots. Based on the value of the discriminant Δ, we can identify these roots (Abramowitz and Stegun, 1964):

$$\Delta = \frac{4{\Delta_0}^3 - {\Delta_1}^2}{27} \tag{3-37}$$

If $\Delta > 0$, then the polynomial has three distinct real roots,

If $\Delta = 0$, then the polynomial has real roots with following description:

- If $\Delta_0 = 0$, then the polynomial has one triple root equal to r_1:

$$r_1 = \frac{1 - B_j}{3} \tag{3-38}$$

- If $\Delta_0 \neq 0$, then the polynomial has one double root equal to r_2 and one simple root equal to r_3:

$$r_2 = \frac{10A_jB_j - A_j - 12B_j^3 - 8B_j^2 + 2B_j}{6A_j - 20B_j^2 - 8B_j - 2} \tag{3-39}$$

$$r_3 = \frac{13A_jB_j - 4A_j - 22B_j^3 - 2B_j^2 + 5B_j + 1}{-3A_j + 10B_j^2 + 4B_j + 1} \tag{3-40}$$

If $\Delta < 0$, then the equation has one real root and two non-real complex conjugate roots.

If in Equation (3-31) $\Delta = \Delta_0 = 0$, we have one root and that root is the compressibility factor of the phase j. In all other cases, in which we have more than one roots, we need to decide which root is the compressibility factor. In order to obtain the correct compressibility factor, we apply these criteria:

Since the compressibility factor is always a real number, in case of $\Delta < 0$, we only calculate the real route and that is the compressibility factor,

The compressibility factor is always positive. Therefore, we neglect zero or negative roots,

Since logarithms cannot admit negative arguments (see Equation (3-19)), we neglect those compressibility factors that do not satisfy the inequality $z_j > B_j$

If after applying these criteria still more than one root remains, we select the compressibility factor based on the nature of the phase in question: If we are solving Equation (3-31) for a gaseous phase, we choose the highest value; if we are dealing with a liquid phase, we choose the lowest value.

After obtaining compressibility factors, we can calculate fugacity coefficient using Equation (3-19). Subsequently, equilibrium ratios can be calculated from knowledge of the fugacity coefficients from Equation (2-59). Flowchart of the flash calculations proposed in this thesis is described in Figure 3-1.

3.4 Modifications of Peng-Robinson Equation of State

In order to obtain better results using the Peng-Robinson equation of state for hydrocarbon mixtures containing water, three modifications have been proposed as follows.

3.4.1 Modification for Heavy Hydrocarbon Components

To improve predications for heavier hydrocarbons, for which $\omega > 0.49$, Robinson and Peng (1980) suggested that the following equation be used instead of Equation (3-27):

$$\kappa_i = 0.379642 + 1.4853\omega_i - 0.164423\omega_i^2 + 0.01666\omega_i^3 \qquad (3\text{-}41)$$

3.4.2 Modification for Water Component

When developing the original correlations for α and κ, Equations (3-26) and (3-27), Peng and Robinson (1976a) did not include water as one of the components and consequently the predicted vapor pressures for water were not as good as expected. In a later publication (1980), they proposed that for $\sqrt{T_{r_w}} < 0.85$, the following equation should be used instead of Equation (3-26):

$$\alpha_w = \left(1.0085677 + 0.82154\left(1 - \sqrt{T_{r_w}}\right)\right)^2 \qquad (3\text{-}42)$$

At temperatures where $\sqrt{T_{r_w}} \geq 0.85$, Equation (3-26) still applies. In the above notation, subscript w denotes water component.

3.4.3 Modification for Aqueous Phase

For the aqueous phase, Peng and Robinson (1980) also suggested that the following equations should be used instead of Equations (3-20) and (3-22):

$$A'_{i,q} = \frac{1}{a_q}\left[2\sqrt{a_i}\sum_{l=1}^{NC} x_{l,q}\sqrt{a_l}(1 - \tau_{li})\right] \qquad (3\text{-}43)$$

$$a_q = \sum_{i=1}^{NC}\sum_{l=1}^{NC} x_{i,q}x_{l,q}\sqrt{a_i a_l}(1 - \tau_{il}) \qquad (3\text{-}44)$$

In the above equation, τ_{il} is a temperature-dependent binary interaction parameter between components i and l in the aqueous phase. Introduction of this parameter for each aqueous binary pair means that the interaction between the water molecule and the non-water molecule is much different from that in the non-aqueous phases. Although this assumption lacks a solid theoretical basis, in practice provides more accurate results. Peng and Robinson (1980) investigated the temperature-dependent

binary interaction parameter for various binary mixtures, including hydrocarbons and non-hydrocarbons with water. Results of their work, which illustrates temperature-dependent binary interaction parameter as a function of experimental conditions and critical properties are given in Figure 3-2 and Figure 3-3.

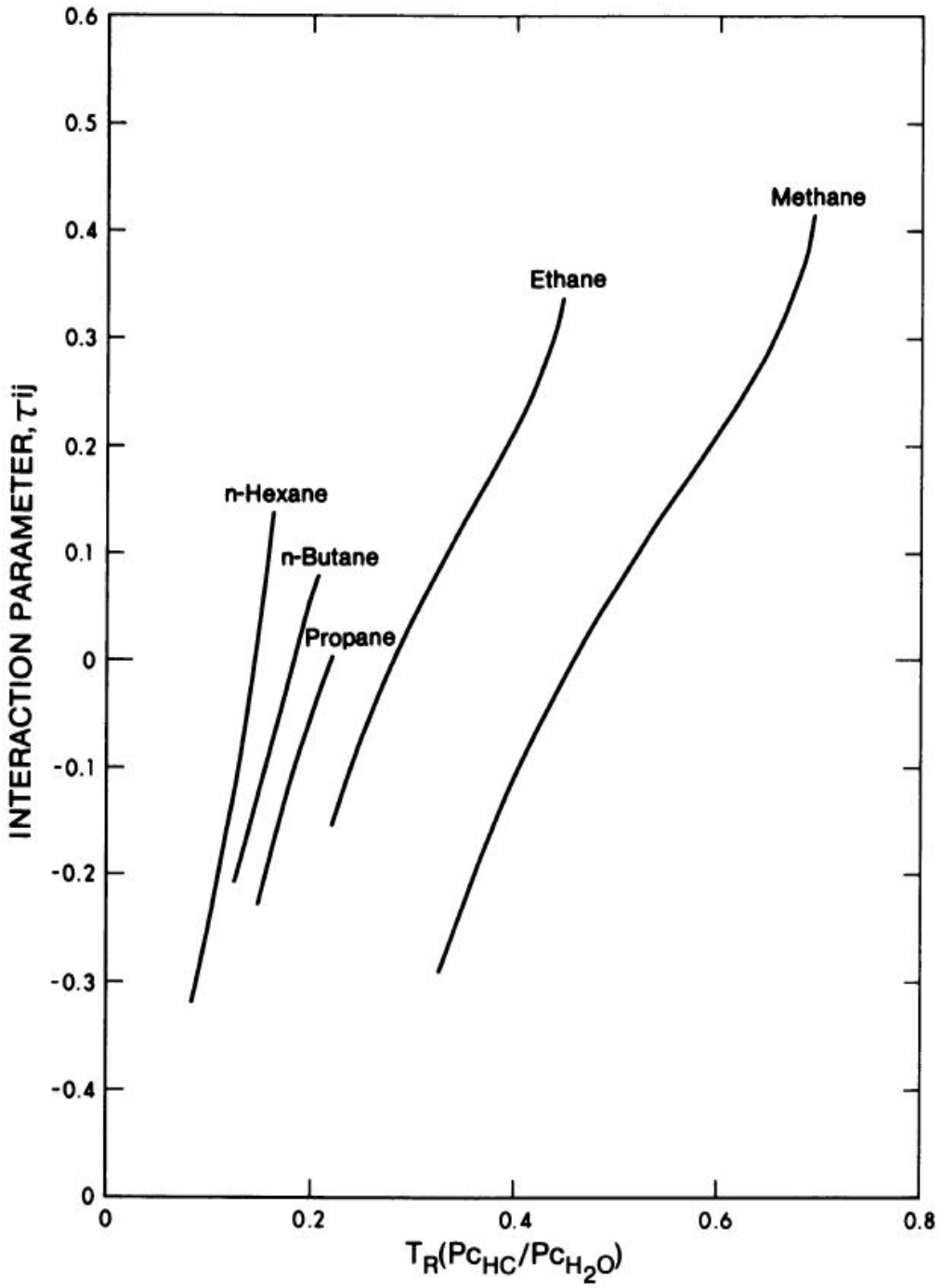

Figure 3-2: Temperature-dependent binary interaction parameter for selected paraffin-water binary systems (Peng and Robinson, 1980).

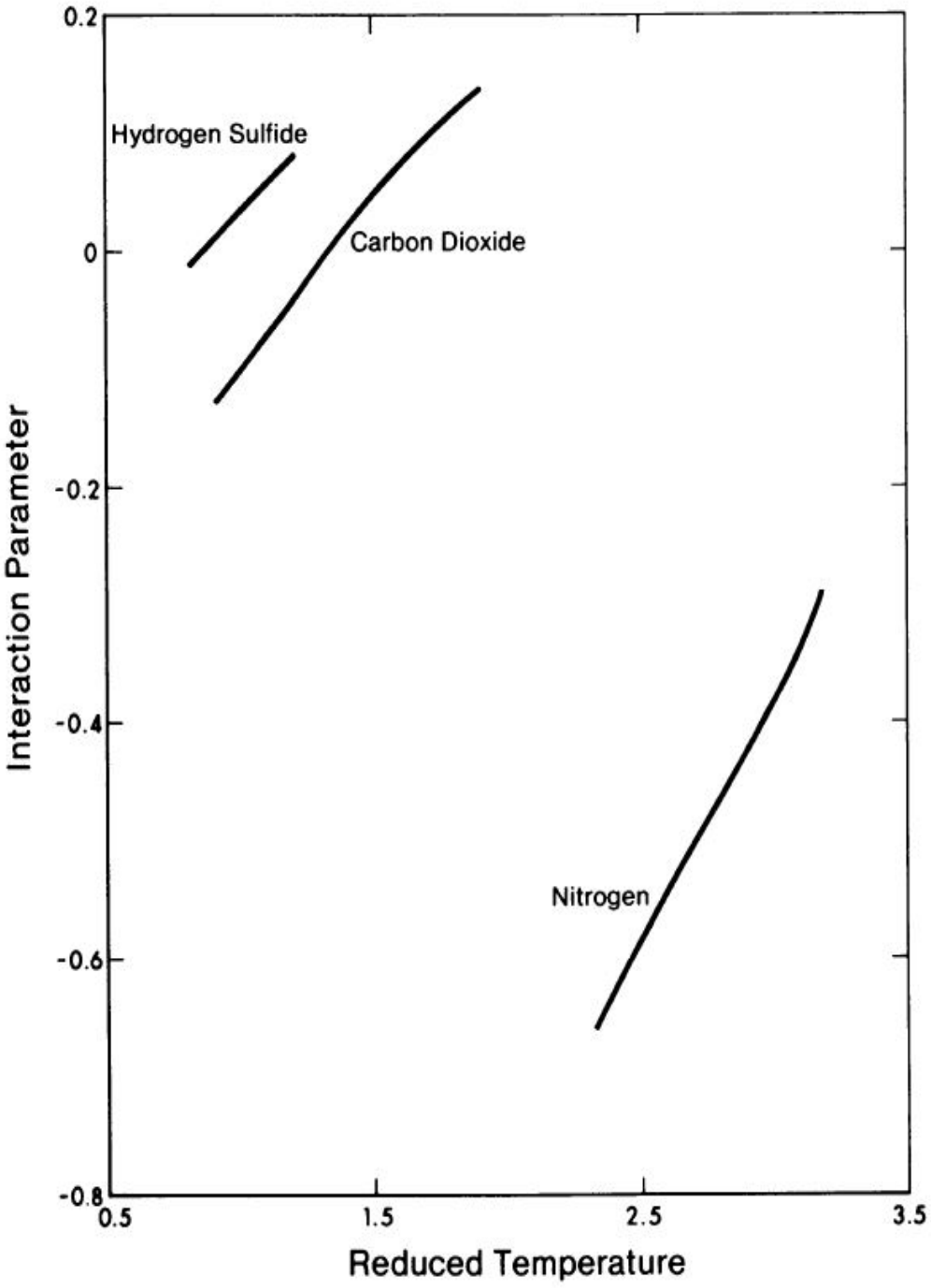

Figure 3-3: Temperature-dependent binary interaction parameter for selected non-hydrocarbon and water binary systems (Peng and Robinson, 1980).

3.5 Volume Translations

Two parameter cubic equations of state, such as Soave-Redlich-Kwong and Peng-Robinson, provide accurate equilibrium mole fractions. However, their ability to estimate volumetric properties has room for improvement (Jhaveri and Youngren, 1988). Péneloux et al. (1982), introduced a third parameter in the Soave-Redlich-Kwong equation of state to improve its volumetric estimations. Their proposed correction is particularly attractive because this third parameter does not change equilibrium mole fractions obtained from two-parameter equation of state, but only modifies the phase volume by affecting certain translations along the volume axis. After Péneloux et al. (1982), Jhaveri and Youngren (1988) introduced a similar parameter into the Peng-Robinson equation of state. In the current work, we use the third parameter as proposed by Jhaveri and Youngren (1988). A description of this third parameter with associated equations is given in the following paragraphs.

The third parameter, called *shift parameter* or *volume shift* or *volume translation* is denoted in literature by c, analogous to the first two parameters, a and b. This parameter, like the two other parameters, is empirical and for each component, it is calculated by using experimental data and statistical approaches.

Since the third parameter, c, has the same units as the second parameter, b, the dimensionless shift parameter, s, is defined as

$$s \equiv \frac{c}{b} \tag{3-45}$$

where c is the volumetric shift parameter and b is given by Equation (3-12).

Given the dimensionless shift parameter, volumetric shift parameters are calculated for each component using Equations (3-45) and (3-28). Then the molar volume of phase j is corrected by the following equation

$$\tilde{V}_{M_j} = V_{M_j} - \sum_{i=1}^{NC} x_{i,j} \times c_i \tag{3-46}$$

where V_{M_j} is calculated by Equation (3-15)

$$V_{M_j} = \frac{z_j RT}{P_j}$$

and $\tilde{V}_{M_j}$ is the corrected molar volume of phase j (SI unit: $\frac{m^3}{mol}$).

3.6 Calculating Phase Properties

In previous sections, we described the equations and procedure for calculating equilibrium ratios as well as proposed corrections for more accurate phase and components estimations. Now, we are in a position to calculate phase properties.

Molar density of phase j (SI unit: $\frac{mol}{m^3}$) is obtained from the following equation:

$$M_j = \frac{1}{\tilde{V}_{M_j}} \tag{3-47}$$

Phase fraction of phase j, denoted by Ψ_j, is given by:

$$\Psi_j = \frac{M_j \times S_j}{\sum_{j=1}^{NP} M_j \times S_j} \tag{3-48}$$

Phase molecular weight (SI unit: $\frac{kg}{mol}$) can be calculated by the following equation

$$MW_j = \sum_{i=1}^{NC} MW_i \times x_{i,j} \tag{3-49}$$

where MW_i is the molecular weight of component i.

Given molar density of the phase and phase molecular weight, phase mass density (SI unit: $\frac{kg}{m^3}$) is obtained from the following equation:

$$\rho_j = M_j \times MW_j \tag{3-50}$$

3.7 *State* Change Signaling

In addition to calculating equilibrium ratios and phase properties, this work intends to substitute another part of *CompFlow Bio* that detects phase appearance or disappearance. For this purpose, first we need to calculate the total mole fraction of each component:

$$x_{i,total} = \sum_{j=1}^{NP} \Psi_j x_{i,j} \qquad i = 1,2,\dots,NC \tag{3-51}$$

Once total mole fractions are calculated, we use tests introduced by Bünz et al. (1991) to determine which phases are present. Given that gas phase is always present as an assumption by default in *CompFlow Bio*, we need to check only three tests.

In order to minimize the calculations, the single-phase behavior is checked first and if not confirmed, two-phase tests are performed. If none of the single or two-phase tests is confirmed, three-phase behavior is assumed.

In the calculations for verifying two-phase behavior, introducing the following functions is necessary:

$$Q_1(\Psi_n, \Psi_q) = \sum_{i=1}^{NC} x_{i,n} - \sum_{i=1}^{NC} x_{i,g} = \sum_{i=1}^{NC} \frac{x_{i,total} K_{i,gq}(1 - K_{i,gn})}{K_{i,gn}K_{i,gq} + \Psi_n K_{i,gq}(1 - K_{i,gn}) + \Psi_q K_{i,gn}(1 - K_{i,gq})} \tag{3-52}$$

$$Q_2(\Psi_n, \Psi_q) = \sum_{i=1}^{NC} x_{i,q} - \sum_{i=1}^{NC} x_{i,g} = \sum_{i=1}^{NC} \frac{x_{i,total} K_{i,gn}(1 - K_{i,gq})}{K_{i,gn}K_{i,gq} + \Psi_n K_{i,gq}(1 - K_{i,gn}) + \Psi_q K_{i,gn}(1 - K_{i,gq})} \tag{3-53}$$

The tests suggested by Bünz et al. (1991) may be presented in the following forms:

Single gaseous phase (*state 3*):

$$\sum_{i=1}^{NC} \frac{x_{i,total}}{K_{i,gq}} < 1 \tag{3-54}$$

$$\sum_{i=1}^{NC} \frac{x_{i,total}}{K_{i,gn}} < 1 \tag{3-55}$$

Gaseous phase and non-aqueous phase (*state 4*):

$$\sum_{i=1}^{NC} \frac{x_{i,total}}{K_{i,gn}} > 1 \tag{3-56}$$

$$\sum_{i=1}^{NC} x_{i,total} K_{i,gn} > 1 \tag{3-57}$$

with $Q_2(\Psi_n, 0) < 0$ at the point $Q_1(\Psi_n, 0) = 0$.

Gaseous phase and aqueous phase (*state 2*):

$$\sum_{i=1}^{NC} \frac{x_{i,total}}{K_{i,gq}} > 1 \tag{3-58}$$

$$\sum_{i=1}^{NC} x_{i,total} K_{i,gq} > 1 \tag{3-59}$$

with $Q_1(0, \Psi_q) < 0$ at the point $Q_2(0, \Psi_q) = 0$.

If none of the *sates 2*, *3*, or *4* is verified, then all three phases are present, i.e. *state 1* is the correct *state*.

For checking these tests, two sets of equilibrium ratios are required, even if fewer than three phase are present. In order to obtain equilibrium ratios between two phases when one of them is not present, we assume a hypothetical phase and equilibrate it with the present phase. Then, equilibrium ratios between those two phases are used in the above-mentioned procedure. Since in this work, gas phase presence is a principle assumption, we equilibrate the hypothetical phase with gas phase and therefore, the hypothetical phase could be either aqueous or non-aqueous phase.

The results of the above procedure are output to *CompFlow Bio*. If the state that *CompFlow Bio* reported as input is different from what is obtained through Equations (3-54) to (3-59), *CompFlow Bio* adopts the new state, selects the appropriate set of primary variables (see Table 2-1) and starts over the Newton iteration. In the new iteration, our package finds associated secondary variables and this process is repeated until Newton iteration converges (see Figure 2-1).

Chapter 4
Results and Discussion

4.1 Introduction

In this chapter, we demonstrate the results of our package under different scenarios. For validating our results, we compare this work's results with a well-known commercial simulator's results under the same conditions. However, before going through the detailed results, first we review the assumption under which our package's calculations are valid and then describe the procedure of validating our results.

4.2 Assumptions

In this section, we describe the assumptions, by which our model is restricted.

The most important basis on which our calculations are structured is *chemical equilibrium*. By chemical equilibrium, we mean all components available in the mixture are in same temperature; mixture pressure remains unchanged and total molar composition of each component is constant. Furthermore, there is no ongoing chemical reaction in the mixture.

Presence of gas phase is a fundamental assumption of *CompFlow Bio*. In this work, we assume the gas phase is always present, even in infinitesimally small quantity. This assumption make calculations for *CompFlow Bio* more efficient and robust as well as with gas phase amount to be very small, it does not result in significant error since total mass balance equations are still valid. It means in case of aqueous and non-aqueous two-phase mixture, we consider it to be *state 1* (three phase) with a threshold amount of gas phase. In case of single aqueous phase, we consider the mixture to be at s*tate 2* containing threshold amount of gas phase and when there is only single non-aqueous phase, it is assumed that mixture is at *state 4* with the threshold amount of gas phase. In each of the mentioned situations, composition of the gas phase is calculated based on equilibrium with other phases. In this study, we assume the fraction associated with threshold amount to be 1×10^{-5}. It means we assume that always at least 1×10^{-5} of total moles of the mixture is in the gas phase.

Peng-Robinson equation of state with volume translation is mostly designed for dealing with hydrocarbon-dominated problems, particularly in high pressures and temperatures. Although results of applying this package to other problems are expected to be within a reasonable margin of error, such conditions are not verified in this thesis. In addition, Peng-Robinson EOS requires several sets of physical properties as inputs. Although most of these inputs are constant and identical, such as critical properties and acentric factors, binary interaction parameters for aqueous phase are temperature-

dependent. Binary interaction parameters used in this work are valid for the scope of demonstrated cases as well as for mixtures with lower pressure and temperature. Application of these interaction parameters to considerably higher temperature cases should be verified before attempting to perform flash calculations in those conditions.

Fixed-point iteration procedure implemented in the current thesis uses initial guesses that are widely considered appropriate estimates for hydrocarbon mixtures (see Equations (3-5) and (3-8)). For other mixtures with significant difference in consisting components, other appropriate initial guesses should be considered.

Furthermore, in *CompFlow Bio*, water is assumed insoluble in non-aqueous phase and nitrogen is only supposed to be in gas phase, these assumptions are not used in structuring calculations of this work. As a result, before assembling this package into *CompFlow Bio*, certain changes have to be made within the current version of *CompFlow Bio* to accommodate partitioning of water and nitrogen in all three phases.

4.3 Results Verifying Procedure

In order to demonstrate the accuracy of our results, we compare our results to a well-known commercial simulator, *Eclipse*®. To attain this goal, we used the *PVTi* module of *Eclipse*® to perform flash calculations under the same conditions as our package. *PVTi* is a module of *Eclipse*® designed to perform fluid characterizations and flash calculations. To this end, *PVTi* uses direct Gibbs free energy minimization method to obtain equilibrium mole fractions. Although the outputs of *PVTi* and our package include almost identical variables (the only exception is phase saturation, which is not calculated by *PVTi*), inputs required by them are different. As *CompFlow Bio* requires a specific and unique configuration in inputs, a special procedure should be used so we can compare the results of *PVTi* module and our package. A description of this procedure, which also clarifies its necessity, is detailed in following paragraphs.

The *PVTi* module of *Eclipse*®, as well as other simulators, requires the total mole fraction of each component as well as pressure and temperature as inputs. Component properties such as critical temperature and pressure, acentric factor and binary interaction parameters can be provided by user as well. On the other hand, *CompFlow Bio* passes primary variables as well as pressure and temperature as inputs to our package. As a result of this particular configuration, our package requires primary variables as inputs while *PVTi* calculates them as part of outputs.

To make *PVTi* results comparable to our results, first we set all of the components properties in *PVTi* to the same values we used in our package. These properties include critical pressures and temperatures

for components, pseudo-critical pressures and temperatures for pseudo-components, as well as acentric factors and binary interaction parameters (BIPs). It should be mentioned that within the *PVTi*, we are only able to set constant binary interaction parameters, i.e. δ_{ij} in Equations (3-20) and (3-22), while temperature-dependent binary interaction parameters, i.e. τ_{ij} in Equations (3-43) and (3-44), are embedded within the *PVTi* and are not adjustable by user. After setting properties of each component and temperature and pressure, an arbitrary mixture with certain total composition is adopted and entered in *PVTi* as input. Then flash calculations are performed by *PVTi* and the results are obtained. From *PVTi* outputs *state* of the problem is determined. Depending on the *state* of the problem, an associated set of primary variables are chosen from the *PVTi* results and with these primary variables, our package performs flash calculations. Afterward, we can compare our results to *PVTi* results and verify our calculations. Results of this verification procedure for different cases are given in following sections of this chapter.

It should be mentioned that *PVTi* does not calculate phase saturation (S). However, phase saturation is an input in *CompFlow Bio*. Therefore, we use the following equation to calculate the saturation of phase j from *PVTi* results:

$$S_j = \frac{\Psi_j \times MW_j}{\rho_j} \tag{4-1}$$

A schematic logic diagram, describing various steps of the verification procedure, is given in Figure 4-1.

4.4 Inputs and Data

In previous sections, procedures for comparing the results of our work with a well-known commercial simulator (*PVTi* of *Eclipse*®) were described in general terms. In the current section, we go through the data and inputs that are adopted from databases and literature. For the sake of comparability, these data are set to be the same, both in our package and *PVTi*. These data include critical or pseudo-critical pressure, critical or pseudo-critical temperature, acentric factor, dimensionless shift parameter, and molecular weight for each component and binary interaction parameters for each pair of components. The data provided in this section are associated with composition of the mixture for which flash calculations were performed. To perform flash calculations for mixtures containing additional components or pseudo-components that are not part of the mixtures in our case studies, the properties of these components have to be provided to the package.

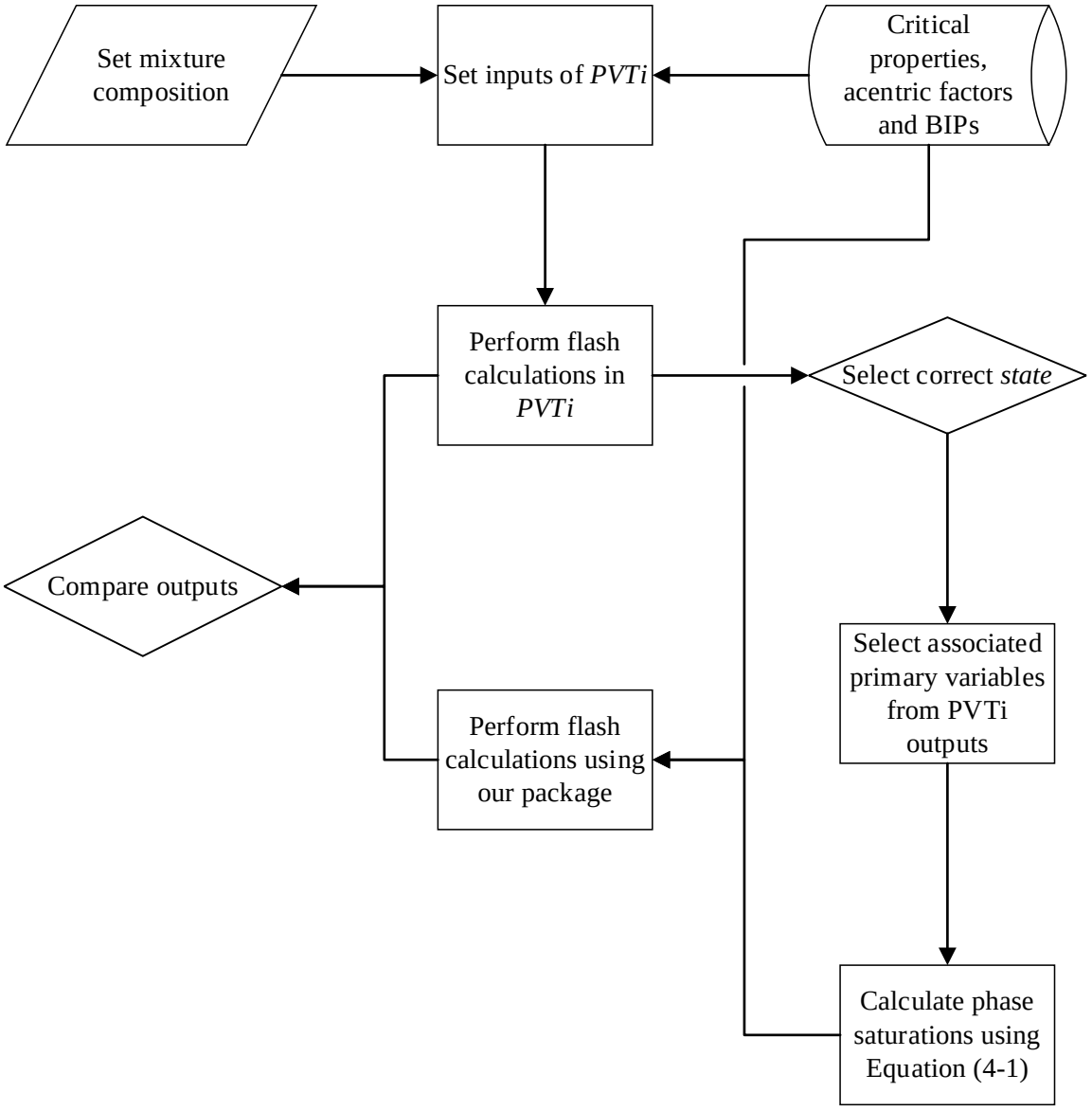

Figure 4-1: Flowchart of verifying procedure.

In this study, our focus is on hydrocarbon mixtures; therefore, in our case studies we considered typical reservoir oil samples as mixtures. Reservoir oil samples consist of numerous hydrocarbon fractions, but for simulation purposes, it is not practical to include all fractions in calculations. Thus, regrouping individual components in pseudo-components is considered as an alternative. For replicating experimental data, it is vital to precisely regroup the experimental sample fractions. Since, however, in our study we are interested in validating our results against another simulator, the definition of pseudo-components is not essential, as long it is consistent in both simulators. Therefore, we chose a primary sample of reservoir oil, which is close to our case study mixtures and estimated pseudo-components properties for that sample and the estimated properties are used for all case studies in this thesis.

4.4.1 Sample Oil Composition

The primary oil composition is adopted from work done by Darvish (2007). Table 4-1 shows the composition of their reservoir sample before regrouping. This oil may be described as volatile oil and originally the sample is taken from a North Sea chalk reservoir.

In the Table 4-1, other forms of each paraffin are lumped with the normal one. For example, iso-butane is lumped with n-butane.

i	$x_{i,total}(\%)$	$MW_i\ (g/mole)$
N_2	0.12	28.0
CO_2	0.83	44.0
CH_4	44.15	16.0
C_2H_6	7.56	30.1
C_3H_8	4.21	44.1
C_4H_{10}	3.15	58.1
C_5H_{12}	2.19	72.2
C_6H_{14}	2.07	86.2
$C_7H_{16} - C_9H_{20}$	8.21	108.3
$C_{10}H_{22} - C_{15}H_{32}$	11.58	166.0
$C_{16}H_{34} - C_{22}H_{46}$	5.51	247.1
$C_{23}H_{48} - C_{34}H_{70}$	4.65	336.2
$C_{35}H_{72} - C_{41}H_{84}$	3.34	484.0
$C_{42}H_{86} - C_{80}H_{162}$	2.43	659.2

Table 4-1: Oil sample composition chosen for estimating pseudo-components properties.

For the sake of brevity, from now on, we denote paraffin groups by their carbon number only; e.g. C_1 instead of CH_4.

Moortgat et al. (2009) used the same oil composition for performing their simulations. We adopted their hydrocarbon pseudo-components; however, they regrouped nitrogen with methane, whereas *CompFlow Bio* requires nitrogen to be a separate component. The regrouped composition, which we used for estimating properties of each pseudo-component, is given in Table 4-2.

For estimating pseudo-component properties, a weighted average approach is used. For example, pseudo-critical pressure of pseudo-component $C_2 - C_3$ is weighted average of ethane and propane

critical pressure. Weights of this average are relative mole percentage of ethane and propane to each other in the initial mixture.

i	$x_{i,total}(\%)$
N_2	0.12
CO_2	0.83
C_1	44.15
$C_2 - C_3$	7.56
$C_4 - C_6$	4.21
$C_7 - C_9$	3.15
$C_{10} - C_{15}$	2.19
C_{16+}	2.07

Table 4-2: Regrouped oil composition used for estimating pseudo-components properties.

4.4.2 Properties of Components

In the following table, critical (or pseudo-critical) pressure and temperature along with acentric factor, molecular weight and dimensionless volume-translation, for each component (or pseudo-component), which we used as input to our package as well as *PVTi*, is given:

i	$T_{ci}\ (K)$	$P_{ci}\ (bar)$	ω_i	MW_i $(g/mole)$	s_i
N_2	126.1	33.94	0.0403	28.0	-0.1927
CO_2	304.2	73.76	0.239	44.0	0.02
H_2O	647.1	220.55	0.345	18.0	0.0147
C_1	190.6	45.99	0.0115	16.0	-0.1595
$C_2 - C_3$	328.5	46.56	0.118	35.0	-0.095
$C_4 - C_6$	458.1	34.24	0.234	70.1	-0.047
$C_7 - C_9$	566.0	25.80	0.370	108.3	0.038
$C_{10} - C_{15}$	651.0	18.60	0.595	166.0	0.155
C_{16+}	824.1	9.61	1.427	385.6	0.277

Table 4-3: Physical properties of regrouped oil sample, taken from Moortgat et al. (2009), Danesh (1998) and Søreide (1989).

In the composition of the mixture used for simulations by Darvish (2007) and adopted by Moortgat et al. (2009), a water component is absent; however, in real reservoir engineering problems, water presence is inevitable. As such, in this study we added water component to our case studies in order to simulate problems that are more realistic.

In the above table, all properties for all pseudo-components and carbon dioxide are taken from Moortgat et al. (2009). Pseudo-component properties are calculated from constituent component properties using a weighted-average formula. Nitrogen, methane and water properties, except dimensionless volume-translation parameters, are adopted from Danesh (1998). Dimensionless volume-translation parameters for nitrogen and methane are taken from Søreide (1989). For water, the following correlation proposed by Søreide (1989) is used to calculate the dimensionless volume-translation parameters.

$$s = |T_r - a_1|^{a_2} + a_3 + a_4\omega + a_5\, exp[a_6(T_r - 1)] \qquad (4\text{-}2)$$

In Equation (4-2), constants a_1 to a_6 for the Peng-Robinson equation of state are given in the table below:

Constant	Value for PR EOS
a_1	0.74145
a_2	1.35489
a_3	-0.16410
a_4	0.47894
a_5	0.42829
a_6	25.3301

Table 4-4: Constants used in Equation (4-2), proposed by Søreide (1989).

Equation (4-2) is a temperature-dependent correlation, which can be used in case of using temperature-dependent dimensionless volume-translation parameter. In case of using temperature-independent dimensionless volume translation parameter, $T_r = 0.7$ should be used. In this work, we adopted a temperature-independent dimensionless volume translation parameter and the value of water dimensionless volume-translation in Table 4-3 is calculated by setting $T_r = 0.7$.

It should be noted that Equation (4-2) is primarily proposed for light hydrocarbon components, but in the absence of work on evaluating the volume-translation parameter for water, the above-mentioned

equation is used for water as well. Note that other commercial simulators, like *PVTi*, are using the same correlation for water.

4.4.3 Binary Interaction Parameters

The only input parameter, which is not covered in the previous sub-section, is the binary interaction coefficient (BIP). As mentioned in Chapter 3, Peng and Robinson (1980) in modification of their original equation of state, proposed two different sets of binary interaction parameters for aqueous phase and non-aqueous liquid and gaseous phases. Binary interaction parameters for non-aqueous liquid and gaseous phase are constants while aqueous phase BIPs are temperature-dependent. This temperature-dependency improves the equilibrium molar distribution estimates in compare to experiments (Peng and Robinson, 1980). In the following paragraphs, values, which in this work we used for these two different sets of parameters, are provided.

4.4.3.1 BIP for Gaseous and Non-Aqueous Phases

Binary interaction parameters for components in non-aqueous liquid and gaseous phases are divided to two parts: BIP between pair of hydrocarbon components and BIP between non-hydrocarbon and hydrocarbon components. In simple models using Peng-Robinson equation of state, as well as other equations of state, it was assumed that BIP between hydrocarbons are zero (Moortgat et al., 2009; Peng and Robinson, 1976b). Nevertheless, precise investigations suggested that BIP between light hydrocarbons and heavy ones are not zero. Specifically, it was found that non-zero BIPs between methane and heptane-plus fractions results in more accurate outputs (Peng and Robinson, 1976a). In this work, the binary interaction parameters between hydrocarbons are non-zero. Values of BIPs used in his work for different pairs of components, except water, are given in Table 4-5.

In the above mentioned table, data for binary interaction parameters for pure components are taken from (Danesh, 1998) and pseudo-components BIPs are calculated as the average of constituent component BIPs.

Binary interaction parameters between water and other components are given in the Table 4-6.

Binary interaction parameters between water and hydrocarbons are taken from Peng and Robinson (1976b) and BIP between water and nitrogen and carbon dioxide are adopted from Søreide and Whitson (1992). Similar to other properties of pseudo-components, BIPs between pseudo-components and water are the average of BIPs between their constituent components and water. For heptane-plus and water binaries, the value of BIP between n-octane and water is used.

Component	N_2	CO_2	C_1	$C_2 - C_3$	$C_4 - C_6$	$C_4 - C_6$	$C_{10} - C_{15}$	C_{16+}
N_2	0	0	0.031	0.068	0.335	0.150	0.155	0.155
CO_2	0	0	0.107	0.128	0.422	0.100	0.015	0.015
C_1	0.031	0.107	0	0.008	0.066	0.043	0.052	0.066
$C_2 - C_3$	0.068	0.128	0.008	0	0.030	0.029	0.026	0.037
$C_4 - C_6$	0.335	0.422	0.066	0.030	0	0.016	0.006	0.010
$C_7 - C_9$	0.150	0.100	0.043	0.029	0.016	0	0	0
$C_{10} - C_{15}$	0.155	0.015	0.052	0.026	0.006	0	0	0
C_{16+}	0.155	0.015	0.066	0.037	0.010	0	0	0

Table 4-5: Binary interaction parameters between components, except water, in non-aqueous liquid and gaseous phases. Original data are taken from Danesh (1998).

Binary Components	BIP
$H_2O - N_2$	0.478
$H_2O - CO_2$	0.190
$H_2O - C_1$	0.500
$H_2O - (C_2 - C_3)$	0.490
$H_2O - (C_4 - C_6)$	0.480
$H_2O - (C_7 - C_9)$	0.480
$H_2O - (C_{10} - C_{15})$	0.480
$H_2O - C_{16+}$	0.480

Table 4-6: Value of BIPs between water and other components. Original data are taken from Peng and Robinson (1976b) and Søreide and Whitson (1992).

4.4.3.2 BIP for Aqueous Phase

Peng and Robinson (1980) suggested that a different mixing rule should be used for the aqueous phase for better estimations of equilibrium mole fractions in mixtures consisting a water-rich phase. The suggested mixing rule for aqueous phase is given in Equations (3-43) and (3-44). In these modified mixing rules, they suggested for the aqueous phase that a temperature-dependent binary interaction coefficient, τ, replace the constant binary interaction coefficient, σ. In their modification, Peng and

Robinson (1980) demonstrated the temperature-dependent binary interaction coefficient, τ, as function of the mixture temperature and components critical properties. Figure 3-2 and Figure 3-3 are adopted from their paper and contain plots of τ for different hydrocarbons and non-hydrocarbons. In order to use values plotted in these two figures, we used linear regression approach to quantify their results. Least square method was used to fit the linear equation to each set of data.

A general form of fitted linear equation for hydrocarbon components (Figure 3-1) is given below:

$$\tau_{iw} = a_{iw} \times T_{r_i}\left(\frac{P_{c_i}}{P_{c_w}}\right) + b_{iw} \quad i \in \{HC\} \tag{4-3}$$

The following is a general form of fitted linear equation for non-hydrocarbon components (Figure 3-2)

$$\tau_{iw} = a_{iw} \times T_{r_i} + b_{iw} \quad i \in \{nonHC\} \tag{4-4}$$

where a_{iw} and b_{iw} are slope and intercept of fitted linear equation to BIP data between component i and water.

Slope and intercept of each fitted equation as well as its coefficient of determination (R-Squared) is given in Table 4-7.

Peng and Robinson (1980) did not provide data for n-pentane or hydrocarbons heavier than hexane. In this study, we generalized and extended the fitted equation for n-hexane to all hydrocarbons heavier than butane. The generalized equation for BIP between water and $C_n (n > 4)$ has the form of:

$$\tau_{C_n w} = 0.333\left(\frac{T_{c_{C_n}}}{P_{c_{C_n}}}\right) \times T_{r_{c_n}}\left(\frac{P_{c_{C_n}}}{P_{c_w}}\right) - 0.812 \quad n > 4 \tag{4-5}$$

Critical temperature and pressure in Equation (4-5) should be entered in $Kelvin$ and bar, respectively.

From Figure 3-2, it can be observed that with increasing number of the hydrocarbon, the slope of the linear equation increases. Therefore, in Equation (4-5), the slope of the fitted equation is correlated to the ratio of hydrocarbon critical (pseudo-critical) temperature to pressure, as this ratio increases monotonically with increasing hydrocarbon carbon number.

i	a_{iw}	b_{iw}	R^2
Methane	1.66	-0.759	0.9931
Ethane	1.991	-0.576	0.9955
Propane	3.124	-0.687	0.9962
n-Butane	3.537	-0.655	0.9997
n-Hexane	5.608	-0.812	0.9884
Carbon Dioxide	0.273	-0.371	0.9955
Nitrogen	0.417	-1.631	0.9988

Table 4-7: Results of linear regression models applied to data of Peng and Robinson (1980).

If we substitute the definition of reduced temperature in Equation (4-5) and cancel the same terms from enumerator and denominator, we obtain the following result

$$\tau_{C_n w} = \left(\frac{0.333}{P_{c_w}}\right) T - 0.812 \qquad (4\text{-}6)$$

in which T is the mixture temperature and has the unit of $Kelvin$ and unit of P_{c_w} is bar. As P_{c_w} is a constant value, the binary interaction parameters between water and hydrocarbons heavier than butane are practically correlated to the mixture temperature and are independent of hydrocarbon critical properties. This can be confirmed by comparing values of BIP for non-aqueous phases in Table 4-6. In Table 4-6, it can be seen that for pseudo-components heavier than propane, values of BIP between water and hydrocarbon are the same.

4.5 Results

In this study, we designed three different case studies, to both validate our package and to demonstrate thermodynamic aspects of multiphase hydrocarbon mixtures at elevated pressure and temperature. In performing calculations for these case studies, the methodology described in Chapter 3 along with inputs provided previously in this chapter is used. Results of these calculations along with description of each scenario are given in the next three sub-sections.

In the following figures, N, A and G stand for non-aqueous, aqueous and gaseous phases, respectively. Since primary variables are input to our package, they are exactly the same in our package and *PVTi* results. Therefore, in most of the charts, only secondary variables are plotted when comparing our package to *PVTi*. In few cases demonstrating the phase behavior of the mixture, primary variables are plotted as well. While *PVTi* results are shown using solid lines in the following charts, results of our

package are represented with dots to provide readable charts. The solid lines in these figures are only connecting *PVTi* results and have no other physical meaning.

4.5.1 Case Study 1

In the first case study, we selected a volatile oil sample. This sample is composed by adding about 10 percent of CO_2 to the primary oil sample, used for estimating pseudo-components and described in sub-section 4.4.1 of this chapter (see Table 4-2). In this case study, which concerns a mixture of *constant total composition* (see Table 4-8), we increase the pressure of the mixture from 1 to 300 bar at constant temperature of 180 ℃. Increasing mixture pressure from ambient condition to high value of 300 bar covers a wide range of problems, from groundwater flow to real reservoir problems. Pressures higher than 300 bar are rare in natural systems.

Figure 4-2 to Figure 4-7 compare the results of our package for Case Study 1 to PVTi results. Flash calculations are performed for whole range of 1 to 300 bar, however equilibrium mole fractions in each phase are plotted within the range of phase presence.

i	$x_{i,total}$(%)
CO_2	11.13
H_2O	9.70
N_2	0.12
C_1	34.15
$C_2 - C_3$	11.77
$C_4 - C_6$	7.41
$C_7 - C_9$	8.21
$C_{10} - C_{15}$	6.58
C_{16+}	10.93

Table 4-8: Composition of mixture used in Case Study 1.

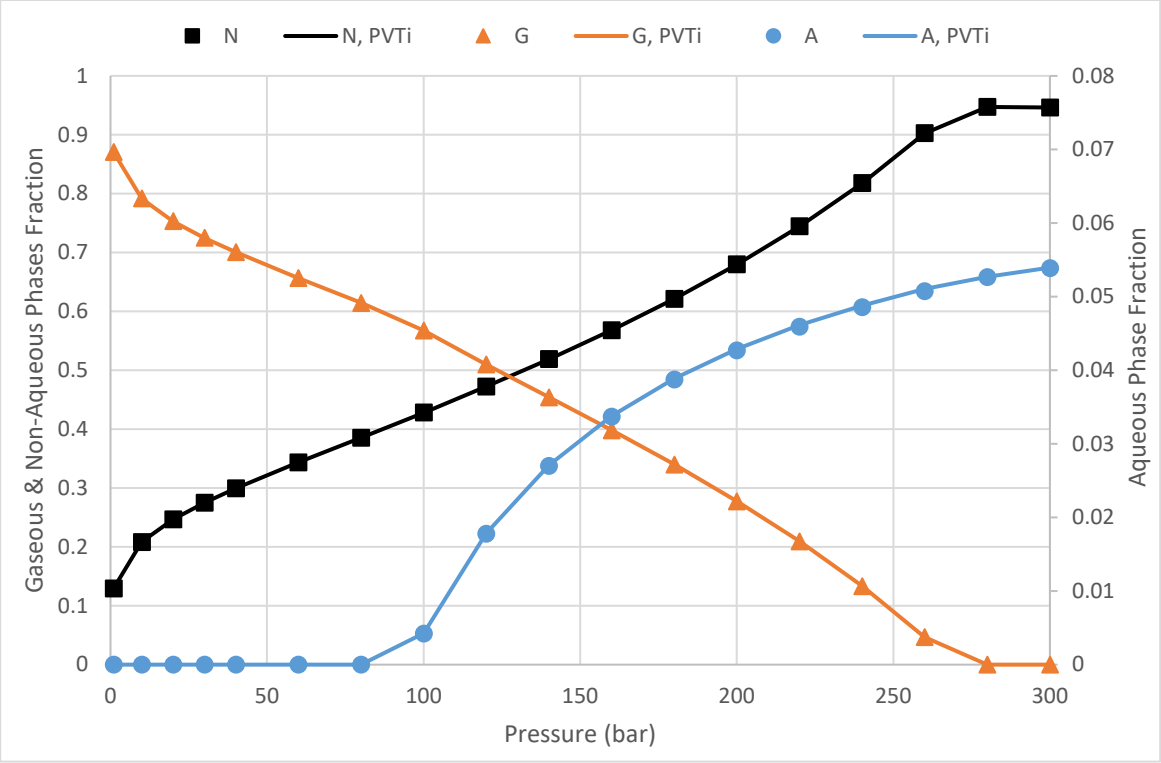

Figure 4-2: Equilibrium phase fractions vs. pressure in Case Study 1.

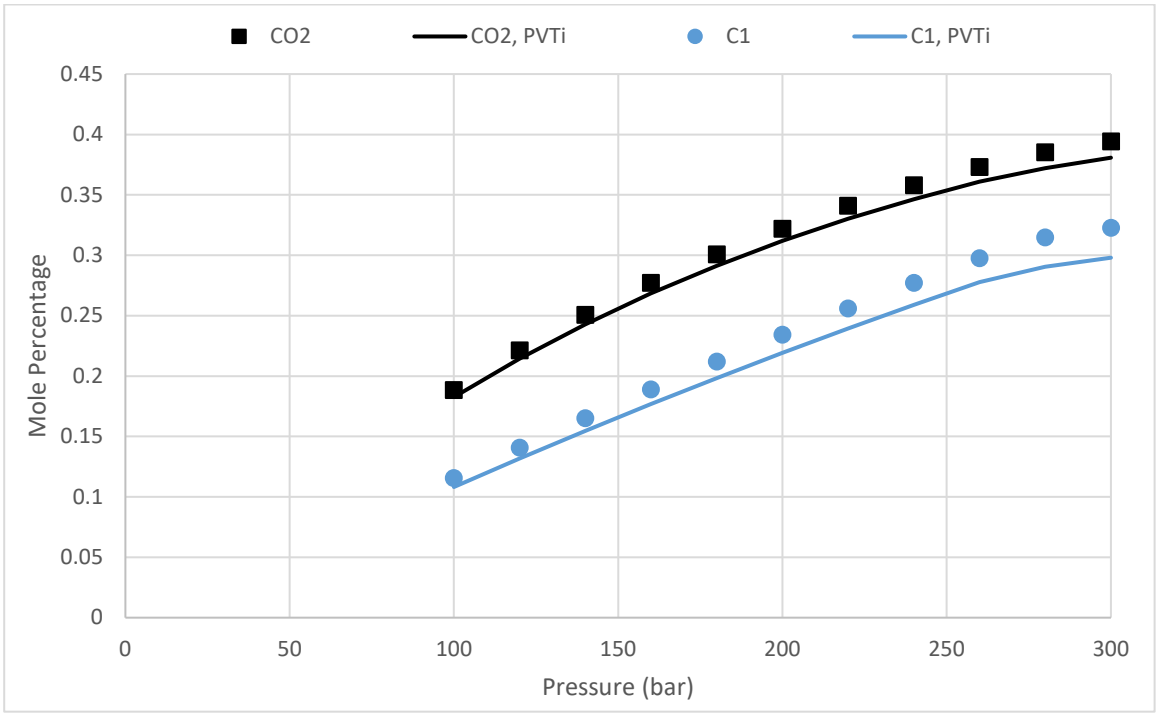

Figure 4-3: Equilibrium mole percent of C_1 and CO_2 in aqueous phase vs. pressure in Case Study 1.

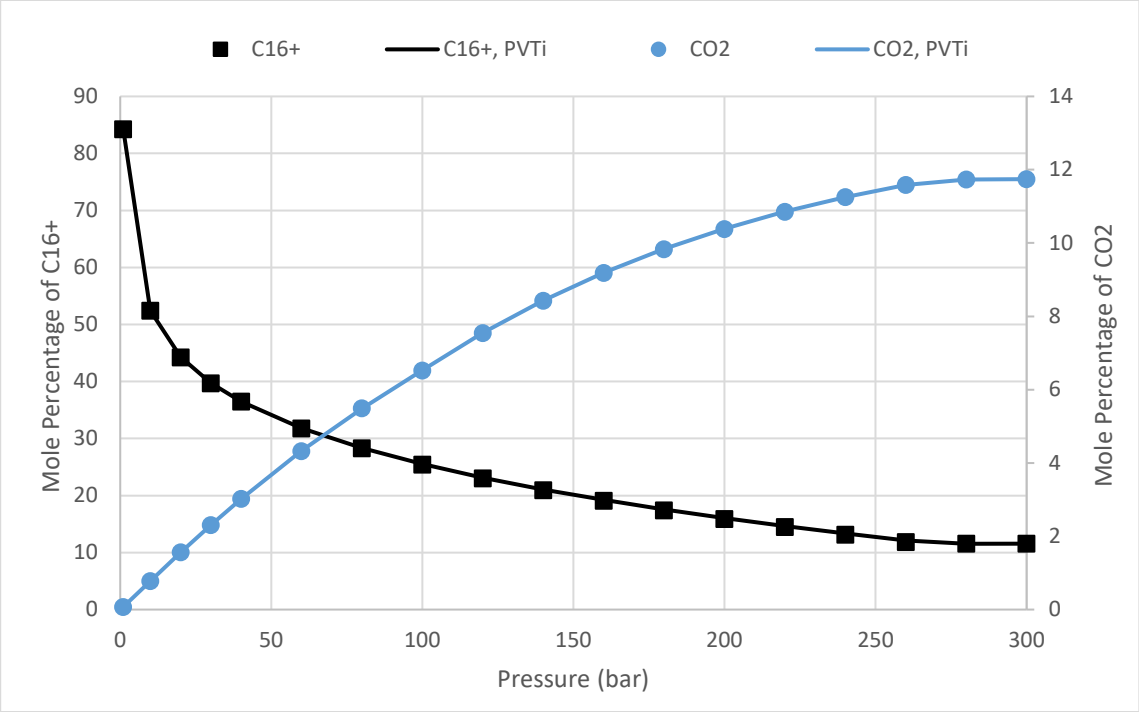

Figure 4-4: Equilibrium mole percent of CO_2 and C_{16+} in non-aqueous phase vs. pressure in Case Study 1.

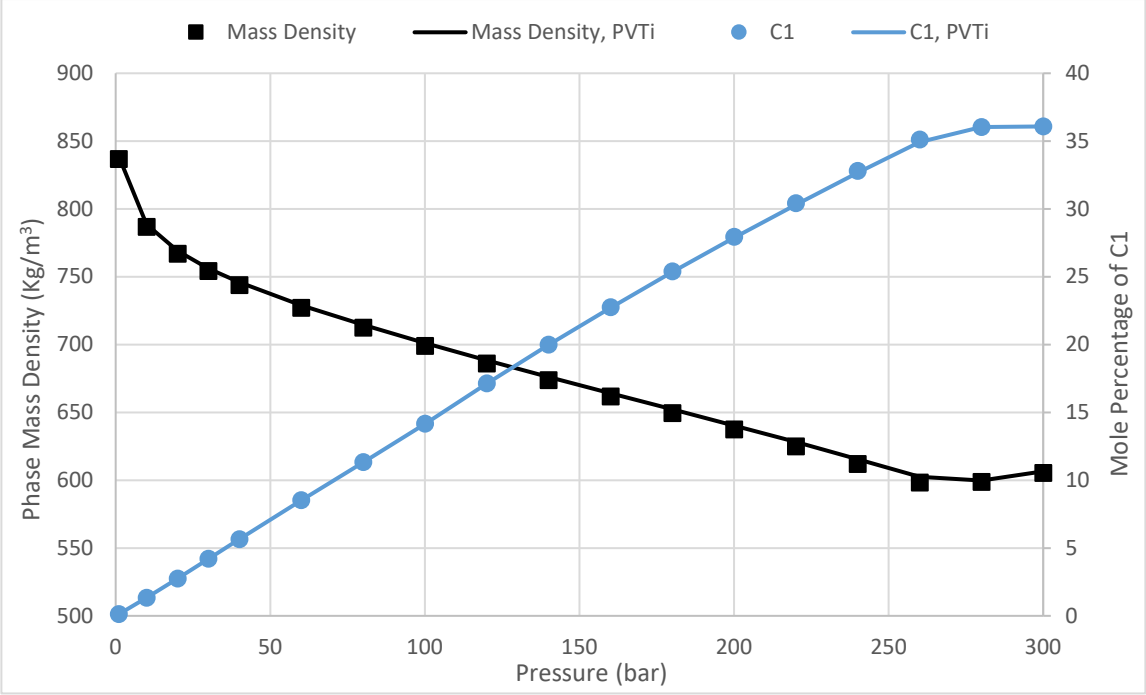

Figure 4-5: Equilibrium mass density of non-aqueous phase and C_1 mole percent in that phase vs. pressure in Case Study 1.

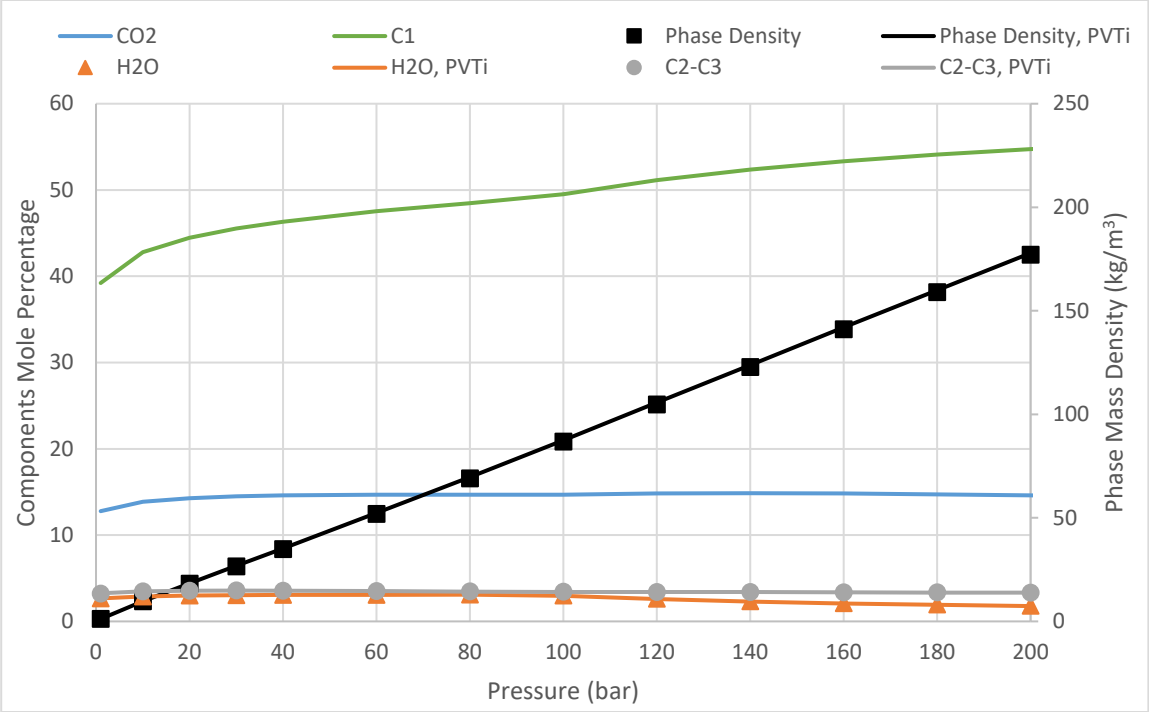

Figure 4-6: Equilibrium composition and mass density of gaseous phase vs. pressure in Case Study 1.

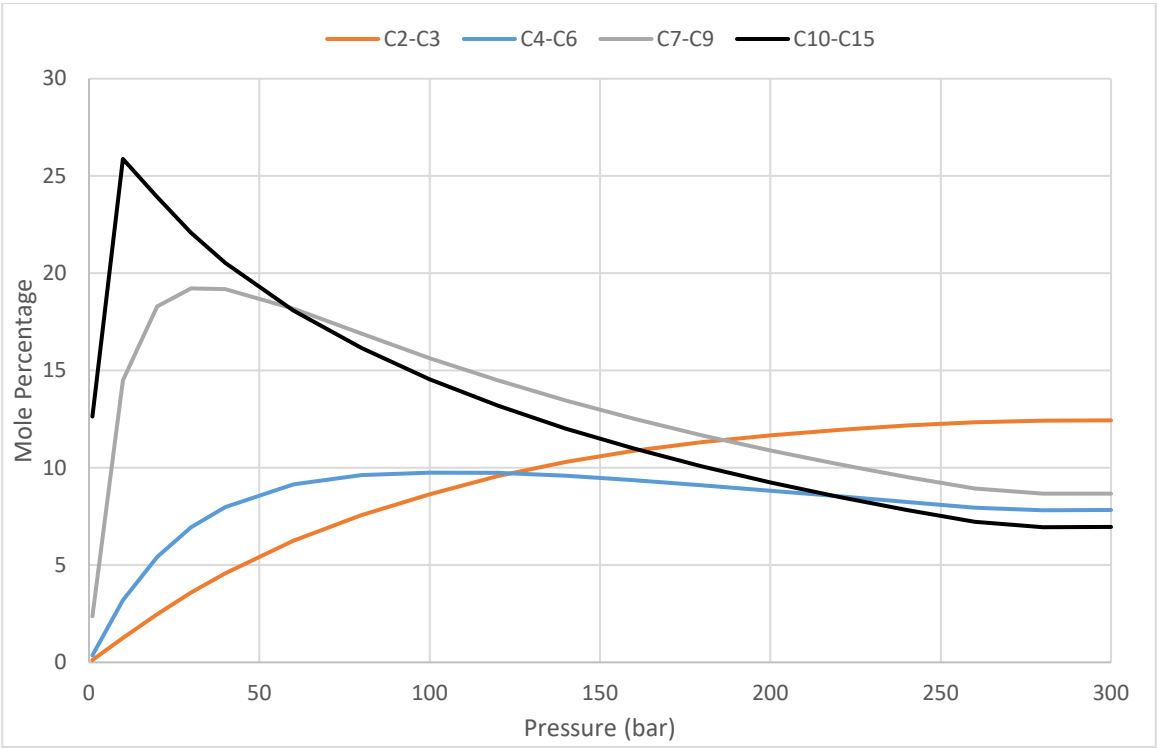

Figure 4-7: Equilibrium composition of non-aqueous phase vs. pressure in Case Study 1.

4.5.2 Case Study 2

In the second case study, we investigated the effect of increasing the CO_2 content of a mixture at constant temperature and pressure. This scenario represents injecting CO_2 into a high pressure and temperature reservoir for enhanced oil recovery or as a part of a CO_2 sequestration project. The mixture pressure during the process is set to be 280 bar and mixture temperature, as before, is 180 ℃. The initial composition of the mixture is similar to the initial oil composition described in sub-section 4.4.1. The difference here is that in the initial mixture composition no CO_2 is present. The composition of the initial mixture used in this case study is given in Table 4-9.

Figure 4-8 to Figure 4-11 demonstrate results of flash calculations for Case Study 2. Calculations were performed until the total mole percent of CO_2 was about 54%. Beyond that point, no further phase appearance or disappearance is expected. It should be noted that, same as results of Case Study 1, phase mole fractions are plotted in the range of phase presence.

i	$x_{i,total}(\%)$
CO_2	0
H_2O	10.91
N_2	0.14
C_1	38.43
$C_2 - C_3$	13.24
$C_4 - C_6$	8.34
$C_7 - C_9$	9.24
$C_{10} - C_{15}$	7.40
C_{16+}	12.30

Table 4-9: Initial oil composition used in Case Study 2.

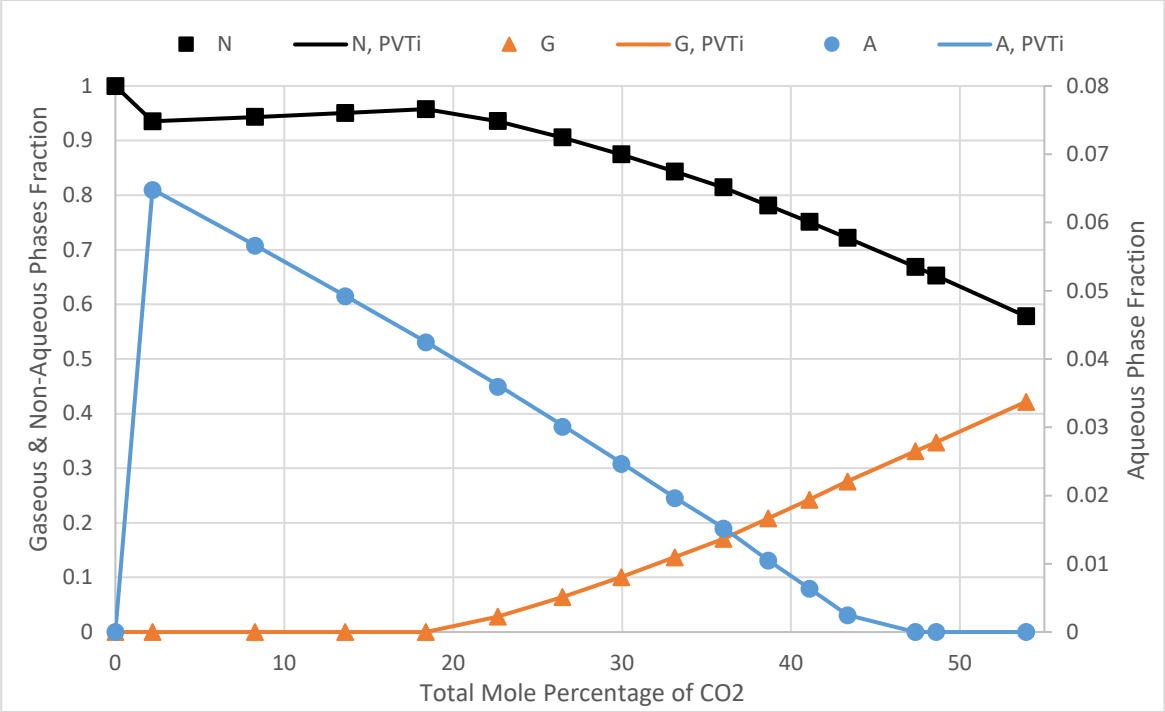

Figure 4-8: Equilibrium phase fractions vs. injected CO_2 mole percent in Case Study 2. Case Study 2.

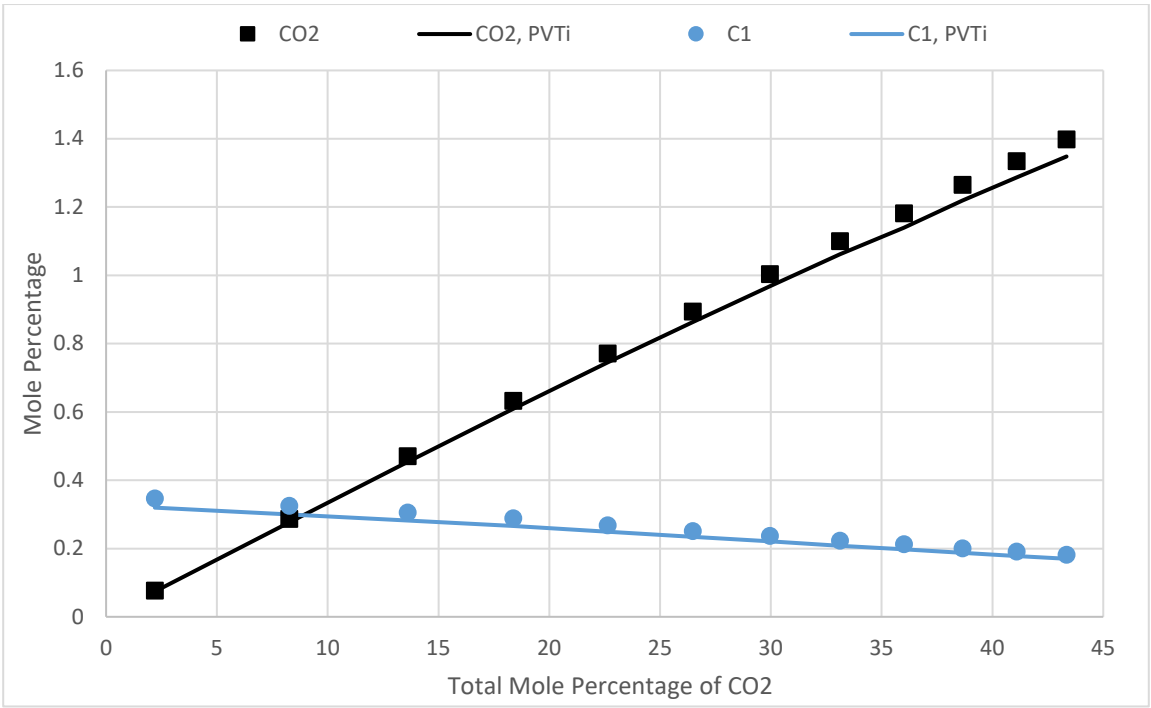

Figure 4-9: Equilibrium mole percent of C_1 and CO_2 in the aqueous phase vs. injected mole percent of CO_2 in Case Study 2.

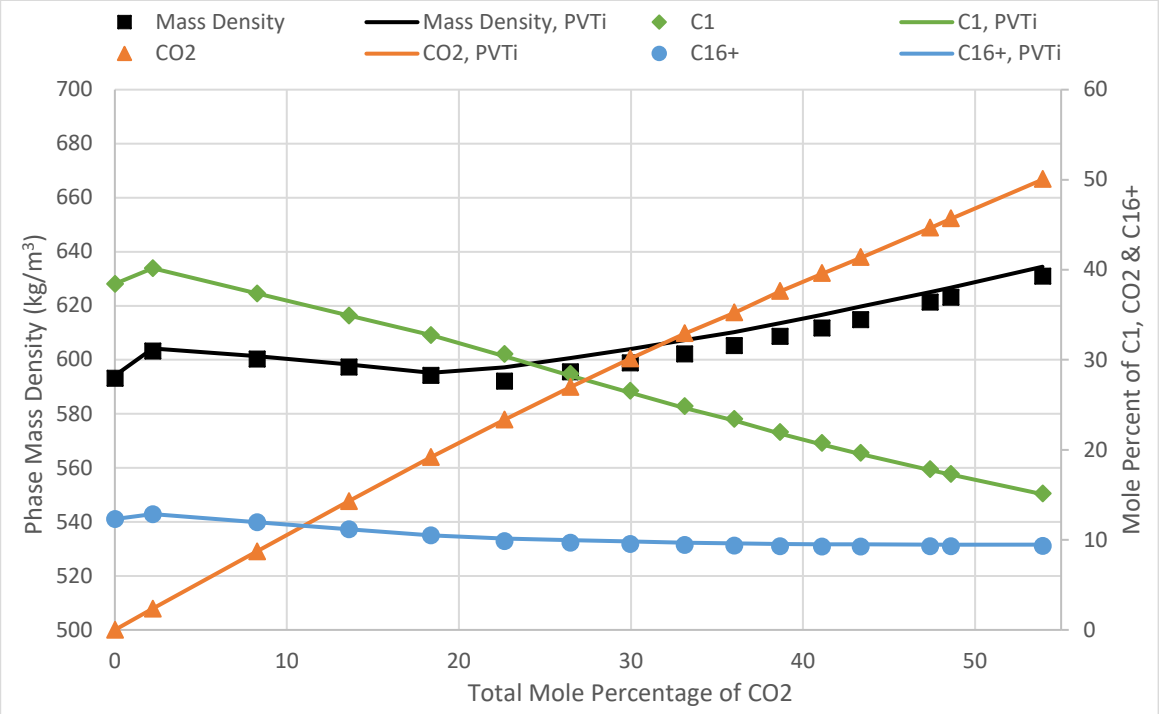

Figure 4-10: Equilibrium composition and mass density of non-aqueous phase vs. injected mole percent of CO_2 in Case Study 2.

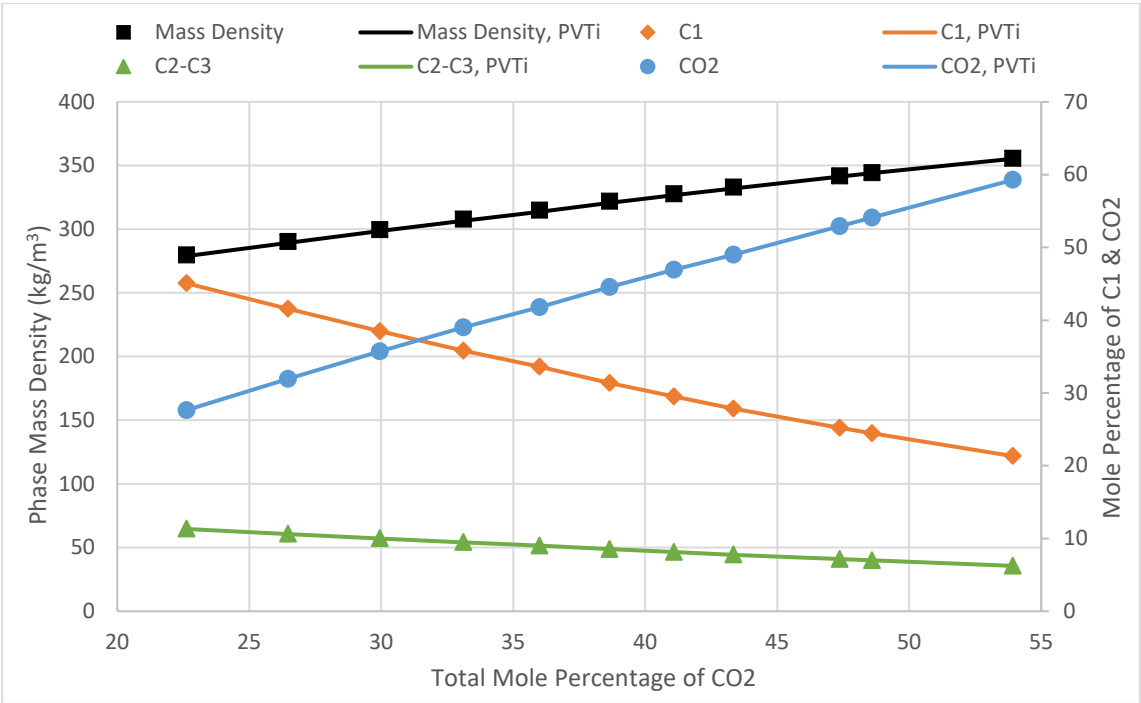

Figure 4-11: Equilibrium composition and mass density of gaseous phase vs. injected mole percent of CO_2 in Case Study 2.

4.5.3 Case Study 3

In the last case study of this thesis, we investigated the effect of CO_2 injection on swelling of an oil sample. For this purpose, we selected a black oil sample and performed the calculation at same pressure as in Case Study 2. Initial composition of the selected oil is given in Table 4-10. In order to observe the effect of CO_2 injection into the oil sample separate from the effect of CO_2 solution into the aqueous phase, we assumed there is no water in the mixture. The pressure and temperature of the mixture are constant and equal to 280 bar and 180 ℃, respectively. The high pressure in this case study, results in delayed formation of a gaseous phase, which would strip the liquid phase out its volatile components, resulting in denser non-aqueous phase.

i	$x_{i,total}(\%)$
CO_2	0.02
N_2	0.34
C_1	34.62
$C_2 - C_3$	5.12
$C_4 - C_6$	3.50
$C_7 - C_9$	12.00
$C_{10} - C_{15}$	17.00
C_{16+}	27.40

Table 4-10: Initial composition of black oil sample used in Case Study 3.

Results of flash calculations for this case study are given in Figure 4-12 to Figure 4-14. Similar to previous sets of charts, equilibrium mole fractions of each phase plotted in the range of phase presence.

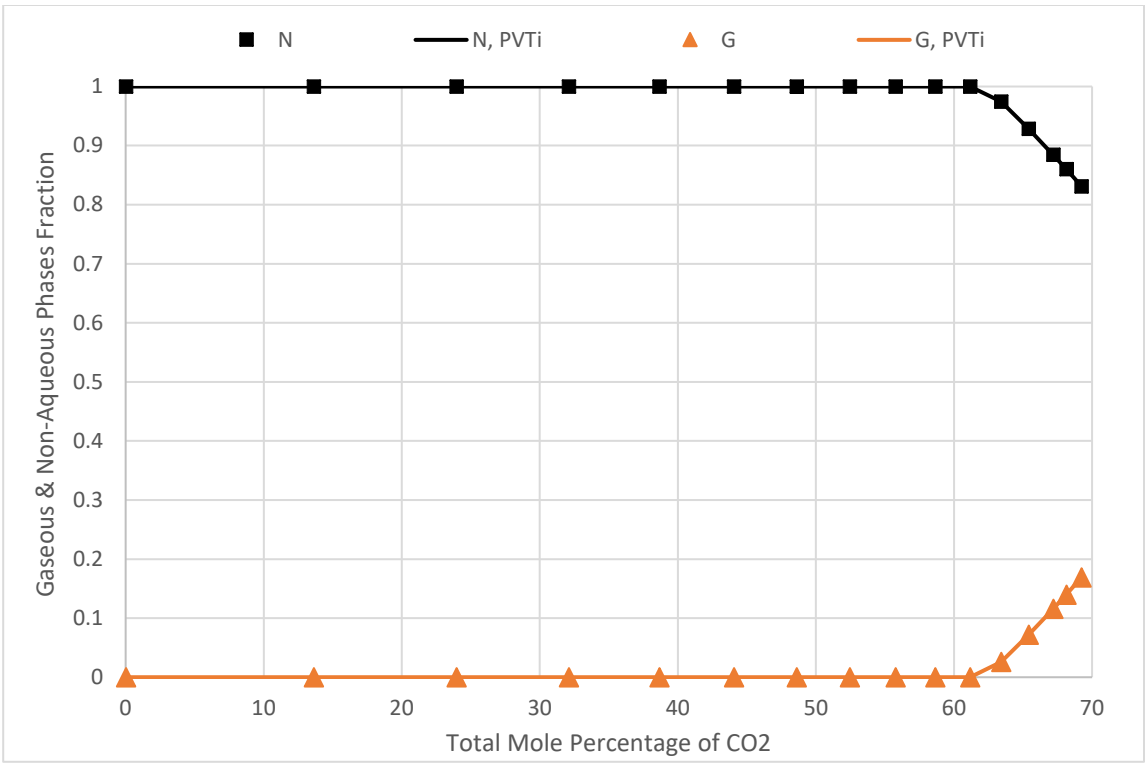

Figure 4-12: Equilibrium phase fractions vs. injected mole percent of CO_2 in Case Study 3.

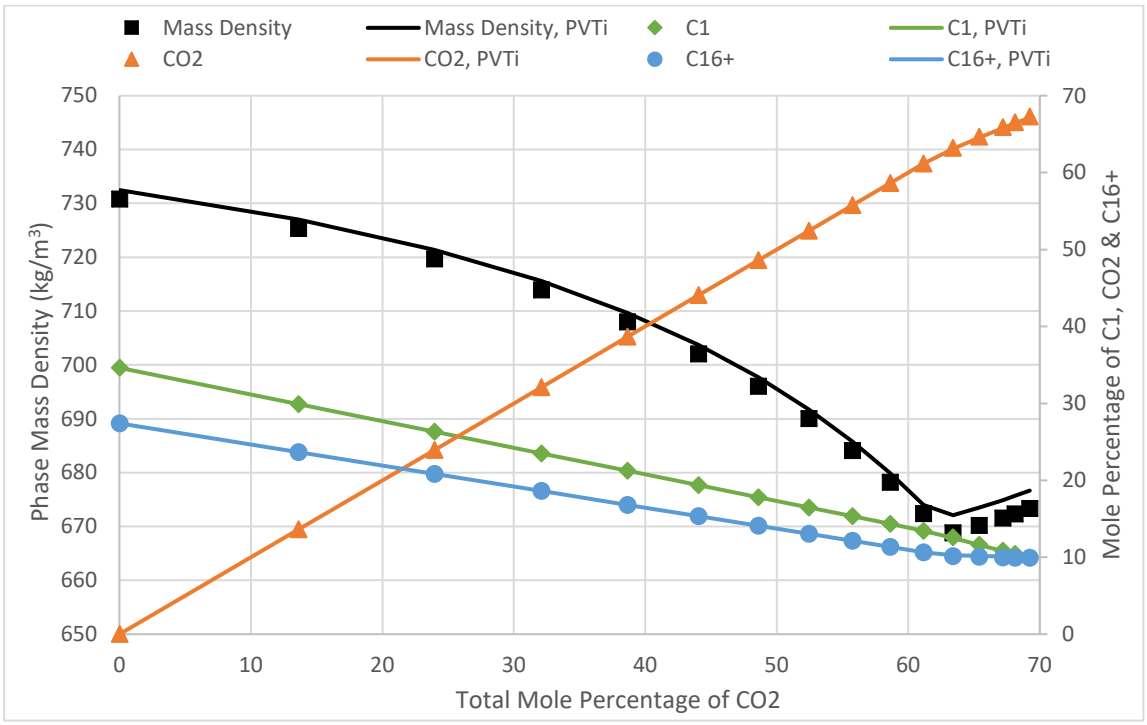

Figure 4-13: Equilibrium composition and mass density of non-aqueous phase vs. injected mole percent of CO_2 in Case Study 3.

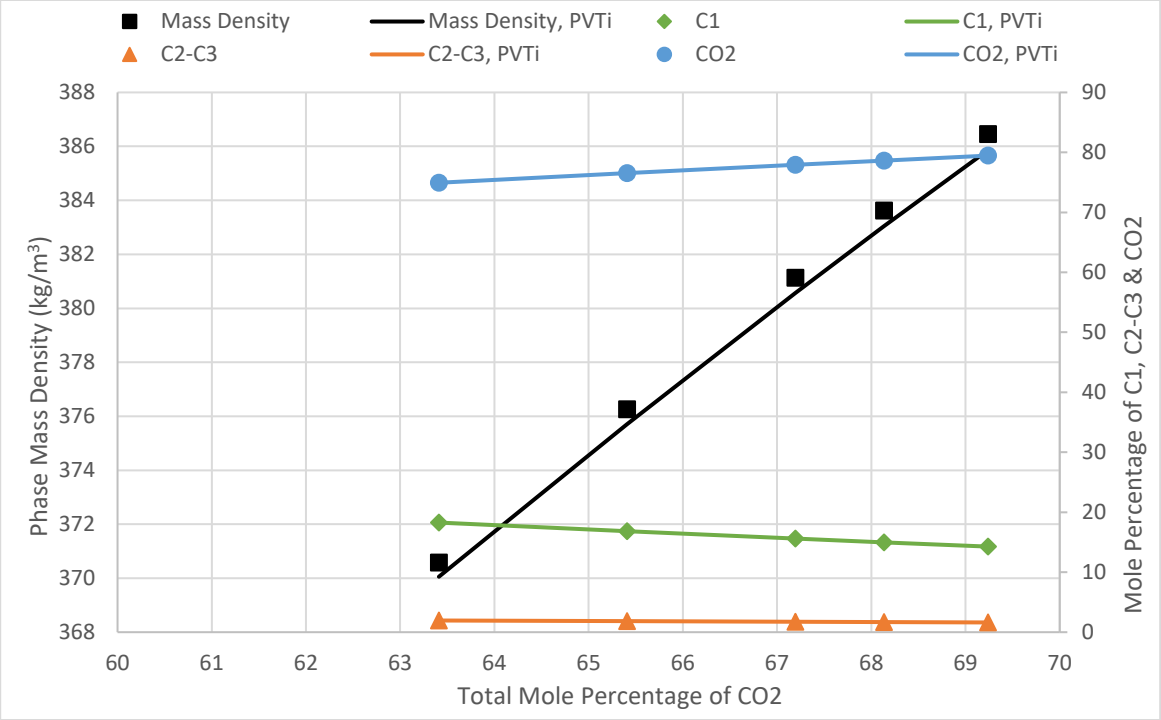

Figure 4-14: Equilibrium composition and mass density of gaseous phase vs. injected mole percent of CO_2 in Case Study 3.

4.6 Discussion

In all of the cases studied in the previous sections, the results of our model show very good agreement with the results from a well-known commercial simulator. Considering the wide range of conditions and different scenarios covered in the case studies of this thesis, our package exhibits the robust configuration and solid thermodynamic background needed by *CompFlow Bio* to deal with complex systems of hydrocarbons. This package can be applied with confidence to the investigation of various aspects of reservoir engineering problems.

While the presented graphs confirm our model's validity and accuracy, there are differences between results of our model and *PVTi*. Sources of these differences are described in the following paragraphs.

The commercial model used in this study uses direct minimization of Gibbs free energy to calculate equilibrium mole fractions, whereas our package uses the equal fugacity method. Difference in numerical approximation errors between the two methods explains, in part, the small differences between the outputs.

Numerous constants and conversion factors used in flash calculations are another source for differences between our results and *PVTi*. The exact value for all of the constants and conversion factors

or the number of digits used for each constant, used in these calculations in *PVTi*, are not known to the writer. While we set the properties for each component the same in *PVTi* as we set in our model, some of these constants are not adjustable in *PVTi*.

Special treatments used for special cases are an important source of difference in the results. Particularly for the aqueous phase as described in sub-section 3.4.3, treatments in our model are adopted from a modification proposed by Peng and Robinson (1980). The treatment used in *PVTi* is not exactly the same. Original data used for this case are taken from the same source but fitted equations (detailed in sub-section 4.4.3) are different. Therefore, the main reason for differences between our package's and *PVTi*'s aqueous phase composition and properties is this difference in treating water-rich phase.

Another source of difference in the results is due to the special structure of our verification procedure. Since input variables for both models are not the same (total mole fraction for each component in *PVTi* vs. primary variables in our model) and we have to use part of the *PVTi* results as inputs to our model, error propagation in these two models is not the same and therefore difference in results is inevitable.

Now that various sources of difference between this thesis' and *PVTi's* results are described, we move on to discuss the results presented in each case study.

In Case Study 1, equilibrium phase fractions are given in Figure 4-2. This figure shows that the gas phase fraction decreases with increase of pressure, as one may expected. In the beginning of the Case Study 1, when pressure is low and temperature is above the normal boiling point of water, there is no aqueous phase. With increasing pressure, an increasing amount of light hydrocarbons partition into the non-aqueous phase. When pressure is about 100 bar, an aqueous phase appears, as mixture pressure is high enough for water condensation. With further increase of mixture pressure, more carbon dioxide and light hydrocarbons partition into the non-aqueous and aqueous phases. At about 280 bar, disappearance of the gas phase occurs and after that, the aqueous phase fraction slightly increases due to solution of more carbon dioxide and light hydrocarbons in water (Figure 4-3). As the mole fraction of carbon dioxide and light hydrocarbons in the non-aqueous phase monotonically increases prior to the disappearance of gas phase, the mole fraction of heavy hydrocarbons decreases until the gas phase disappears. After that, the mole fraction of heavy hydrocarbons reaches a plateau state (Figure 4-4).

One of the most notable aspects of Case Study 1 is shown in Figure 4-5. The mass density of the non-aqueous phase decreases with increasing pressure until gas phase disappears; then it starts to increase slightly. The reason for significant decrease in non-aqueous phase mass density is dissolution of increasing amounts of carbon dioxide and light hydrocarbons in the non-aqueous phase due to pressure increase. Following the disappearance of the gas phase, when no more carbon dioxide or light

hydrocarbons are available to dissolve in the non-aqueous phase, the mass density shows a small linear increase with pressure due to the small, but finite compressibility of the oil phase.

The mass density of the gas phase in Case Study 1, increases with pressure as expected (Figure 4-6). Since components heavier than methane and carbon dioxide preferentially partition out of the gaseous phase with increasing pressure, the mole fractions of methane and carbon dioxide in the gaseous phase increase, as the volume fraction of the entire gaseous phase decreases.

In Figure 4-7, we observe two different trends for medium and heavy hydrocarbon constituents of the non-aqueous phase, depending on their initial amount in the gaseous phase. First, for lighter hydrocarbons such as $C_2 - C_3$, we see a monotonically increasing mole fraction, which is due to dissolution of more $C_2 - C_3$ into non-aqueous phase from gaseous phase. For medium range hydrocarbons, $C_4 - C_6$ and $C_7 - C_9$, an initial increase in mole fraction is due to dissolution of these hydrocarbons in non-aqueous phase; but as the amount of these components in the gaseous phase is depleted, with further increase of pressure and solution of more carbon dioxide and methane in non-aqueous phase, their mole fraction decreases. For heavy hydrocarbons, a monotonic decrease of mole fraction is seen, as their amount in the gas phase, even in the initial conditions of the case study, is negligible.

In Case Study 2, the initial composition involves no carbon dioxide and no aqueous phase is present in the mixture. With the injection of even a tiny amount of carbon dioxide, most of the water in non-aqueous phase partitions out of the non-aqueous phase and forms a separate aqueous phase. In reality, carbon dioxide is always present in a hydrocarbon system, and for this reason, an aqueous phase should be expected to exist in real reservoirs. With additional injection of carbon dioxide (Figure 4-8), increasing amounts of carbon dioxide is dissolved at equilibrium in both phases, until the amount of injected carbon dioxide becomes sufficient to sustain a gaseous phase with other light hydrocarbons (this occurs when about 20 mole percent of the mixture is carbon dioxide). Prior to the appearance of the gaseous phase, a decrease in aqueous phase fraction is due to the fact that most of the injected carbon dioxide partitions into the non-aqueous phase and increases its total mole fraction. After appearance of the gaseous phase, the main reason for decreasing fraction of water-rich phase is being dried out by the gaseous phase. This dry out continues until the aqueous phase disappears (when about of 45 mole percent of the mixture is carbon dioxide). As more carbon dioxide is injected, most of it partitions into the gaseous phase, the phase fraction of the gaseous phase increases and that of the non-aqueous phase decreases.

In the range where an aqueous phase is present, its carbon dioxide content increases (Figure 4-9) as more carbon dioxide is added to the mixture, whereas the mole fractions of methane and other components concomitantly decrease.

Figure 4-10 plots the mass density and composition of the non-aqueous phase for Case Study 2. Injection of carbon dioxide at first step causes increase in the mass density of non-aqueous phase along with separation of the aqueous phase from non-aqueous phase. Formation of the aqueous phase strips non-aqueous phase out of most of its dissolved water and therefore, considerable decrease in the water mole fraction results in increase in other components mole fractions including heavy fractions. The effect of increase in the heavy pseudo-components mole fractions outweighs effect of increase in the light hydrocarbons mole fractions, such as methane, and as a result, mass density of the non-aqueous phase increases. Further injection of more carbon dioxide, decreases mass density of the non-aqueous phase very slightly due to dissolution of this light component, but immediately after formation of the gaseous phase it begins to increase significantly. This increase is due to the gas phase stripping light hydrocarbons out of the non-aqueous phase. As expected, during injection of the carbon dioxide to the mixture its mole fraction in the non-aqueous phase, as well as other phases, increases and the mole fraction of methane decreases due to addition of carbon dioxide and partitioning into the gaseous phase, as soon as this phase is formed. Prior to the formation of a gaseous phase, the mole fractions of all components, expect carbon dioxide, decrease due to the addition of carbon dioxide. After formation of a gaseous phase, however, an increase in carbon dioxide content of the non-aqueous phase is almost compensated by a decrease in methane mole fraction. Thus, mole fractions of other hydrocarbons remain constant.

As in Case Study 1, the mass density of the gaseous phase increases monotonically during Case Study 2 (Figure 4-11). The reason, however, is quite different. In Case Study 2 mole fraction of carbon dioxide increases and as such other component mole fractions decrease. Increasing carbon dioxide mole fraction causes mass density decline, whereas decreasing methane mole fraction causes increase of the mass density. As methane is lighter than carbon dioxide, its mole fraction reduction has a stronger effect in increasing gaseous phase density than increase of carbon dioxide mole fraction has on decreasing density.

Case Study 3 is designed to further illustrate the effect of carbon dioxide injection on non-aqueous phase. Here, the pressure is chosen high enough to prevent formation of a gaseous phase until a large portion of the mixture consists of carbon dioxide. Furthermore, no water is assumed present in the mixture to remove the dry-out effect and thus magnify the effect of carbon dioxide on non-aqueous phase. Lastly, the oil sample chosen for this case study is black oil, whilst the oil chosen for the first

two case studies was a volatile oil. The effect of carbon dioxide on black oil is more considerable than other lighter type of the oil. A mixture, like the one chosen for Case Study 3, remains in a single-phase region until about 63 mole percent of the mixture is carbon dioxide (Figure 4-12).

The most important result of the Case Study 3 is indicated in Figure 4-13. The equilibrium mass density of oil declines by 8.1% before emergence of gas phase, due to the injection of carbon dioxide. After gaseous phase appearance, the non-aqueous phase mass density starts to increase as its light components are being stripped out by gaseous phase. These results illustrate that the addition of carbon dioxide to an oil phase does not always result in swelling of the oil phase.

While the mole fraction of carbon dioxide in the non-aqueous phase increase consistently, following the emergence of the gaseous phase the rate of its increase declines. This is due to competition from the newly formed gaseous phase for partitioning of the injected carbon dioxide. The mole fractions of other components in the non-aqueous phase decrease prior to the formation of a gas phase because of carbon dioxide addition. After the appearance of the gas phase, as the mole fraction of methane decreases further due to non-aqueous phase being stripped out of its light hydrocarbons by gaseous phase, the mole fraction of C_{16+} increases.

The behavior of the gaseous phase in the third case (Figure 4-14) is the same as in Case Study 2. The mole fraction of carbon dioxide increases, while methane and other components mole fractions decrease. The mass density of the oil phase increases, as a decline in the mole fraction of methane has a more significant effect than the increasing mole fraction of carbon dioxide mole fraction.

Chapter 5
Conclusions and Recommendations

5.1 Summary

In support of numerical modelling of multiphase flow in complex hydrocarbon systems, this thesis covered background thermodynamics and three testing scenarios of enhancements proposed for the simulator *CompFlow Bio*. Chapter 2 briefly reviewed *CompFlow Bio* structure and principal equations, then provided the reader with basic thermodynamic rules leading to governing phase equilibria equations, along with past research highlighting the need for robust and efficient account of both the physics of flow in discretely fractured networks and the physics of three-phase equilibria involving complex hydrocarbon mixtures, water and carbon dioxide. Chapter 3 of the thesis explained the specific methodology used to calculate equilibrium mole fractions based on a Peng-Robinson EOS, in a manner that accommodates *CompFlow Bio*'s underlying assumptions and special requirements. Then, three different case studies were developed in order to investigate the results of this thesis against the phase characterization module of a well-known simulator in the petroleum industry. The case study results presented in Chapter 4 are the culmination of this thesis. In Chapter 4, along with the results, a procedure designed for investigating the accuracy of the results and the data used in models was given. The three cases confirmed the accuracy of our results and demonstrated the complex behavior that should be expected of hydrocarbon mixtures under high pressure and temperature. This complex behavior reaffirms the need for a robust and efficient package to calculate equilibrium partitioning of mixtures components between phases. The current version of *CompFlow Bio* lacked a sophisticated model for non-ideal mixtures and this thesis provides it with a much-needed tool. Integration of the proposed package in this thesis with the current version of *CompFlow Bio*, is expected to form a sophisticated simulator which can model fractured reservoirs more efficiently and precisely than other well-known simulators available in the petroleum industry.

We conclude this document by summarizing some specific findings and making recommendations about how this package may be further improved.

5.2 Conclusions

Main conclusions of this research may be listed as follows:

- Reservoir problems include highly non-ideal hydrocarbon mixtures, and for modelling such systems effectively and accurately, a robust model with solid thermodynamic background is in need.

- *CompFlow Bio* is a unique simulator with regard to its ability to handle the physics of flow in fractured porous media (Walton et al., 2017), but has simplistic thermodynamics which limits its ability to handle realistic problems in petroleum reservoir engineering. This thesis provides *CompFlow Bio* with essential thermodynamics needed for solving highly non-ideal equilibrium problems.
- Peng and Robinson (1976a) and Soave-Redlich-Kwong (1972) equations of state are widely accepted for modelling hydrocarbon mixtures (McCain, 1990; Tarek, 2009). While both equations of state provide accurate equilibrium mole fractions, Peng-Robinson EOS provides better liquid phase density, particularly near the critical region (Danesh, 1998; Peng and Robinson, 1976a; Tarek, 2007). In this thesis, a modified Peng-Robinson EOS has been adopted for performing equilibrium flash calculations.
- Results of case scenarios developed for verifying the proposed model of this thesis are in close agreement with results of a well-known commercial simulator. These results also provide important insights into phase behavior of the hydrocarbon mixtures.
- The mass density of the non-aqueous phase can be decreased by increasing pressure, provided a gaseous phase with light hydrocarbons is available.
- An expanding gaseous phase can dry out the existing aqueous phase until the water-rich phase completely disappears, even in a highly pressurized mixture.
- Injection of carbon dioxide at constant pressure and temperature does not guarantee swelling and consequent decrease in non-aqueous phase density. Swelling phenomenon is a strong function of pressure, temperature, and mixture composition. Injection of carbon dioxide without sufficient knowledge of the mixture may not result in the desired outcome and even increase non-aqueous phase density.
- In order to have accurate estimates for the swelling effect of carbon dioxide in high-pressure mixtures, it is highly recommended to include non-equilibrium mass transfer in the model. Since all of the calculations done in this research are restricted to equilibrium conditions, no generalized comment has been made about the magnitude of carbon dioxide swelling effect on black oil mass density.
- Injection of carbon dioxide at constant pressure and temperature does not guarantee reducing gas phase density as well. It strongly depends on light hydrocarbon content of the gaseous phase, particularly methane.

5.3 Recommendations for Future Research

Recommendations for future researches fall into two categories: enhancements in *CompFlow Bio* code, and improvements for the proposed model in this thesis.

At this point following enhancements are recommended for *CompFlow Bio* code to enable calculating accurate results for highly non-ideal situations:

- Eliminating restriction on partitioning water in three phases. Currently, water component is not allowed to partition in the non-aqueous phase, which imposes a significant barrier on integration of this work to *CompFlow Bio* code.
- Improving models used for predicting phase viscosity. Currently, *CompFlow Bio* uses simplistic relationships for the phase viscosity estimations that are only functions of temperature (Forsyth, 1993). In order to predict phase viscosity in highly non-ideal hydrocarbon mixtures at high pressure and temperature accurately, these relationships need to be replaced with more sophisticated viscosity models, such as models proposed by Pedersen et al. (1984); Lohrenz et al. (1964); or Aasberg-Petersen et al. (1991).
- Addition of non-equilibrium calculation to the *CompFlow Bio* code. With amendment of the proposed package in this thesis to *CompFlow Bio*, this simulator would be able to perform equilibrium conditions calculations. While equilibrium calculations are important in estimating mixtures properties, they are the first step to perform non-equilibrium calculations. The most important non-equilibrium calculations in hydrocarbon mixtures are finite-rate mass transfers between different phases. Addition of such calculations to the *CompFlow Bio* is a substantial and significant step toward accurate modelling of reservoir engineering problems.

The following improvements are recommended for the proposed model in this thesis:

- Replacing fixed-point iteration method with other iterative methods, like Newton-Raphson that guarantee convergence of the iteration.
- Using stability analysis and phase-split calculations for determination of number of available phases instead of using tests proposed by Bünz et al. (1991). Stability analysis and subsequent phase-split calculations proposed by Michelsen (1982a; 1982b) are highly recommended for the hydrocarbon mixtures.

References

Aasberg-Petersen, K., Knudsen, K., and Fredenslund, A. 1991. Prediction of Viscosities of Hydrocarbon Mixtures. Fluid phase equilibria 70(2-3): 293-308.

Abramowitz, M., and Stegun, I.A. 1964. Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables. Courier Corporation.

Alizadeh, A.H., Khishvand, M., Ioannidis, M.A., and Piri, M. 2014. Multi-Scale Experimental Study of Carbonated Water Injection: An Effective Process for Mobilization and Recovery of Trapped Oil. Fuel 132: 219-235.

Bear, J. 1972. Dynamics of Fluids in Porous Media. Courier Corporation.

Bünz, A.P., Dohrn, R., and Prausnitz, J.M. 1991. Three-Phase Flash Calculations for Multicomponent Systems. Computers & chemical engineering 15(1): 47-51.

Danesh, A. 1998. PVT and Phase Behaviour of Petroleum Reservoir Fluids. Elsevier.

Darvish, G.R. 2007. Physical Effects Controlling Mass Transfer in Matrix Fracture System During CO2 Injection into Chalk Fractured Reservoirs, Fakultet for Ingeniørvitenskap og Teknologi.

Edmister, W.C. 1961. Applied Hydrocarbon Thermodynamics. Gulf Publishing Company, Houstan, Texas.

Enouy, R., Li, M., Ioannidis, M.A., and Unger, A.J.A. 2011. Gas Exsolution and Flow During Supersaturated Water Injection in Porous Media: II. Column Experiments and Continuum Modeling. Advances in Water Resources 34(1): 15-25.

Forsyth, P.A. 1991. A Control Volume Finite Element Approach to NAPL Groundwater Contamination. SIAM Journal on Scientific and Statistical Computing 12(5): 1029-1057.

Forsyth, P.A. 1993. A Positivity Preserving Method for Simulation of Steam Injection for NAPL Site Remediation. Advances in Water Resources 16(6): 351-370.

Forsyth, P.A., and Shao, B.Y. 1991. Numerical Simulation of Gas Venting for NAPL Site Remediation. Advances in water resources 14(6): 354-367.

Izadpanah, A.A., Vafaie Sefti, M., and Varaminian, F. 2006. Multi-Component-Multiphase Flash Calculations for Systems Containing Gas Hydrates by Direct Minimization of Gibbs Free Energy. Iranian Journal of Chemistry and Chemical Engineering (IJCCE) 25(3): 27-34.

Jhaveri, B.S., and Youngren, G.K. 1988. Three-Parameter Modification of the Peng-Robinson Equation of State to Improve Volumetric Predictions. SPE reservoir engineering 3(03): 1,033-031,040.

Karimaie, H., Lindeberg, E.G.B., Torsaeter, O., and Darvish, G.R. 2007. Experimental Investigation of Secondary and Tertiary Gas Injection in Fractured Carbonate Rock, Society of Petroleum Engineers.

Karimaie, H., and Torsaeter, O. 2010. CO2 and C1 Gas Injection for Enhanced Oil Recovery in Fractured Reservoirs, Society of Petroleum Engineers.

Lake, L.W. 1989. Enhanced Oil Recovery. Prentice Hall, Englewood Cliffs, New Jersey.

Lohrenz, J., Bray, B.G., and Clark, C.R. 1964. Calculating Viscosities of Reservoir Fluids From Their Compositions. Journal of Petroleum Technology 16(10): 1,171-171,176.

McCain, W.D. 1990. The Properties of Petroleum Fluids. PennWell Books.

Michelsen, M.L. 1982a. The Isothermal Flash Problem. Part I. Stability. Fluid phase equilibria 9(1): 1-19.

Michelsen, M.L. 1982b. The Isothermal Flash Problem. Part II. Phase-Split Calculation. Fluid Phase Equilibria 9(1): 21-40.

Moortgat, J., Firoozabadi, A., and Farshi, M.M. 2009. A New Approach to Compositional Modeling of CO2 Injection in Fractured Media Compared to Experimental Data, Society of Petroleum Engineers.

Pedersen, K.S., Fredenslund, A., Christensen, P.L., and Thomassen, P. 1984. Viscosity of Crude Oils. Chemical Engineering Science 39(6): 1011-1016.

Péneloux, A., Rauzy, E., and Fréze, R. 1982. A Consistent Correction for Redlich-Kwong-Soave Volumes. Fluid Phase Equilibria 8(1): 7-23.

Peng, D.Y., and Robinson, D.B. 1976a. A New Two-Constant Equation of State. Ind. Eng. Chem. Fundam 15(1): 59-64.

Peng, D.Y., and Robinson, D.B. 1976b. Two and Three Phase Equilibrium Calculations for Systems Containing Water. The Canadian Journal of Chemical Engineering 54(5): 595-599.

Peng, D.Y., and Robinson, D.B. 1980. Two- and Three-Phase Equilibrium Calculations for Coal Gasification and Related Processes. ACS Publications.

Rao, S.S. 1996. Engineering Optimization: Theory and Practice. New Age International.

Robinson, D.B., and Peng, D.-Y. 1978. The Characterization of the Heptanes and Heavier Fractions for the GPA Peng-Robinson Programs. Gas Processors Association.

Seyyedi, M., and Sohrabi, M. 2017. Pore-Scale Investigation of Crude Oil/CO2 Compositional Effects on Oil Recovery by Carbonated Water Injection. Industrial & Engineering Chemistry Research 56(6): 1671-1681.

Smith, J.M., Van Ness, H.C., and Abbott, M.M. 2001. Introduction to Chemical Engineering Thermodynamics. McGraw Hill, Singapore.

Soave, G. 1972. Equilibrium Constants from a Modified Redlich-Kwong Equation of State. Chemical Engineering Science 27(6): 1197-1203.

Søreide, I. 1989. Improved Phase Behavior Predictions of Petroleum Reservoir Fluids from a Cubic Equation of State. Dr. Ing. dissertation, Norwegian Inst. of Technology, Trondheim, Norway.

Søreide, I., and Whitson, C.H. 1992. Peng-Robinson Predictions for Hydrocarbons, CO2, N2, and H2 S with Pure Water and NaCI Brine. Fluid Phase Equilibria 77: 217-240.

Tarek, A. 2007. Equations of State and PVT Analysis. Golf Publishing Company, Houston, Texas.

Tarek, A. 2009. Working Guide to Vapor-liquid Phase Equilibria Calculations. Gulf Professional Publishing.

Unger, A.J.A., Forsyth, P.A., and Sudicky, E.A. 1998. Influence of Alternative Dissolution Models and Subsurface Heterogeneity on DNAPL Disappearance Times. Journal of contaminant hydrology 30(3): 217-242.

Unger, A.J.A., Sudicky, E.A., and Forsyth, P.A. 1995. Mechanisms Controlling Vacuum Extraction Coupled with Air Sparging for Remediation of Heterogeneous Formations Contaminated by Dense Nonaqueous Phase Liquids. Water Resources Research 31(8): 1913-1925.

Walton, K.M. 2013. On Modeling Three-Phase Flow in Discretely Fractured Porous Rock, University of Waterloo.

Walton, K.M., Unger, A.J.A., Ioannidis, M.A., and Parker, B.L. 2017. Impact of Eliminating Fracture Intersection Nodes in Multiphase Compositional Flow Simulation. Water Resources Research 53.

Wilson, G.M. 1969. A Modified Redlich-Kwong Equation of State, Application to General Physical Data Calculations.

Yu, S., Unger, A.J.A., and Parker, B. 2009. Simulating the Fate and Transport of TCE from Groundwater to Indoor Air. Journal of contaminant hydrology **107**(3): 140-161.

Appendices

Appendix A Modifications for Better Convergence of Fixed-Point Iteration

Fixed-point iteration scheme does not guarantee convergence of the iterations procedure. Starting the iteration scheme with good initial guesses is a key step in using this iteration scheme, which significantly improves convergence of the iteration. For the case of hydrocarbon mixture, Equations (2-5) and (2-8) are suggested to provide good initial guesses. However, considering the highly non-ideal behavior of hydrocarbon mixtures in elevated pressure and temperature, even with using these initial guesses, negative mole fractions can be seen during early stages of iteration. These anomalies in first steps of iteration procedure, could lead to non-convergent iteration or delay the convergence of the iteration scheme. In order to prevent such conditions, in each step of the iteration after evaluating the components mole fractions in all phases, additional conditions applied to the code to check and replace any possible negative value for mole fractions. Since primary variables are always positive and so are equilibrium ratios, negative mole fractions could only be a results of imposing unity for summation of phase mole fractions (see Equations (3-1) to (3-4)). Therefore, in each iteration after solving those equations for obtaining associated mole fractions, values of those mole fractions should be checked and replaced, if needed.

Nevertheless, substitution of any of the above-mentioned mole fractions leads to non-unity of summation of mole fractions. As a treatment for these situations, in the case of required mole fraction substitution, the mole fractions of that phase are normalized to ensure their summation remains equal to one. It should be noted that, to prevent any unwanted change to primary variables, primary variables of the phases are excluded from the normalization.

Appendix B Main *MATLAB*® Codes used for Performing Flash Calculations

In this work, four sets of *MATLAB*® functions developed in order to perform flash calculations in four different *states* of *CompFlow Bio*, using the formulations presented in Chapter 3. Inputs and components properties used in these functions are listed in section 4.4 of the current document.

Four functions are presented in this appendix, each of which is associated with one *state* of *CompFlow Bio.* In each function, temperature is first input and the second input is an array of $m + 1$ rows (see Table 2-1). Sequence of inputs for each *state* in that array is the same as mentioned in Table 2-1.

Outputs of each function includes: (i) equilibrium phase properties: fractions, compressibility factors, mass densities and molecular weights; and (ii) equilibrium mole percent for each component in each phase.

In addition to the four main functions, two auxiliary functions are provided as well. These auxiliary functions are used to equilibrate the hypothetical non-aqueous and aqueous phases with the present gas phase in order to obtain equilibrium ratios needed for *state* change signaling (see section 3.7).

```
function [ output ] = Flash_PR_State1( T_C, input )
%State 1
%     Gaseous, Non-Aqueous and Aqueous phases are present.
% Inputs: T_C: temperature in Centrigrade
%          input: 1st row: Non-aqueous phase pressure
%                 2nd row: Non-aqueous phase saturation
%                  3rd row: aqueous phase saturation
%                  4th row to end: primary mole percents
%  Outputs:  1st  row:  phase fractions
%              2nd row: phases' compresibility factors
%          3rd row: phases' mass density (kg/m^3)
%             4th row: molecular weight of phases
%              5th row to end: equilibrium mole fractions of components
%           Columns: 1st column: gaseous
%                    2nd column: non-aqueous
%                    3rd column: aqueous

format long;

%% Input form CompFlow

% pressure for aqueous, non-aqueous and gaseous phases (bar)
global p_q; global p_n; global p_g;
% no capillary pressure data currently availbale
p_n = input(1); p_q = p_n; p_g = p_n;
p_q = p_q*14.5037738; p_n = p_n*14.5037738;      p_g      = p_g*14.5037738;

% saturations (volume fractions) of each phase
global S_q; global S_n; global S_g;
S_n = input(2); S_q = input(3); S_g = 1 - S_n - S_q;

N_row = 9;
N_phase = 3;

% converting mole percents to fractions
input(4:N_row)                 = input(4:N_row)/100;

% known molar fractions in gaseous phaseh
global X_CH4_g; global X_CO2_g;
X_CO2_g = input(8); X_CH4_g = input(9);

% known molar fractions in non-aqueous phase
global X_C2C3_n; global X_C4C6_n; global     X_C7C9_n;     global X_C10C15_n;
X_C2C3_n = input(4); X_C4C6_n = input(5); X_C7C9_n = input(6);
X_C10C15_n = input(7);

%% Setting physical properties & calculating dependent variables

T    =    1.8*T_C    +    491.67; %temperature conversion to Rankin
T_SI = T_C + 273.15; %temperature conversion to Kelvin

% critical pressures
% component   CO2     H2O      N2    CH4    C2C3   C4C6   C7C9   C10C15   C16+
Pc =        [73.76  220.55  33.94  45.99  46.56  34.24  25.80  18.60  9.61];
Pc = Pc*14.5037738; % conversion of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2    CH4    C2C3   C4C6   C7C9   C10C15   C16+
```

```
Tc =         [304.2  647.1  126.2
190.6   328.5   458.1   566.0   651
824.1];
Tc = Tc*1.8; % conversoion of
critical temperatures from Kelvin
to Rankin

% acentric factors
% component   CO2      H2O       N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
w =         [0.239  0.345   0.0403
0.0115  0.118  0.234  0.370  0.595
1.427];

% molecular weight
% component   CO2      H2O       N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
MW =         [44.0   18.0    28.0
16.0    35.0   70.1   108.3  166.0
385.6];

% dimensionless volume shifts
% component   CO2       H2O      N2
CH4       C2C3      C4C6      C7C9
C10C15  C16+
s =       [  0.02   .0147   -.1927
-.1595    -.095     -.047     .038
.155    .277];

% calculating reduced parameters
Pr = p_n./Pc;
Tr = T./Tc;

% binary interaction coefficients
for non-aueous and gaseous phases
% component    CO2       H2O     N2
CH4       C2C3       C4C6      C7C9
C10C15   C16+
sigma =       [0         .19     0
.107      .128       .422      .1
.015     .015
    .19        0        .478    .5
.49        .48       .48       .48
.48
    0         .478    0       .031
.068       .335     .15       .155
.155
    .107       .5       .031    0
.008       .066      .043      .052
.066
    .128      .49     .068    .008
0          .03       .029     .026
.037
    .422      .48     .335    .066
.03        0         .016     .006
.01
    .1        .48     .15     .043
.029     .016    0       0       0
    .015      .48     .155    .052
.026     .006    0       0       0
    .015      .48     .155    .066
.037     .01     0       0       0];

% binary interaction coefficients
for aqueous phase
tau = sigma;
% slope and intercepts of linear
regression
slint = [0.273   -0.371
    0       0
    0.417  -1.631
    1.66   -0.759
    1.991  -0.576
    3.124  -0.687
    .00151  -.812];
% water-CO2
tau(2,1)  =  slint(1,1)*Tr(1)  +
slint(1,2);
% water-N2
tau(2,3)  =  slint(3,1)*Tr(3)  +
slint(3,2);
% water-methane
tau(2,4)                        =
slint(4,1)*Tr(4)*Pc(4)/Pc(2)   +
slint(4,2);
% water-C2C3
tau(2,5)           =           (
.5*slint(5,1)+.5*slint(6,1)
)*Tr(5)*Pc(5)/Pc(2) + ...
    (
.5*slint(5,2)+.5*slint(6,2) );
% water-C4C6
tau(2,6)   =   slint(7,1)*T_SI   +
slint(7,2);
% water-C7C9
tau(2,7) = tau(6,1);
% water-C10C15
tau(2,8) = tau(6,1);
```

```
% water-C16+
tau(2,9) = tau(6,1);

tau(:,2) = tau(2,:);

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)                -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units
R_SI = 8.3144598; % gas cosntants
in SI units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;
b_i = 0.07796074*R*Tc./Pc;
c = s.*b_i; % volume shifts
n = size(sigma,2);

%  molar  fraction  in  phases
(primary variables)
% component  CO2        H2O     N2
CH4      C2C3      C4C6      C7C9
C10C15       C16+
x_n =       [0          0       0
0            X_C2C3_n   X_C4C6_n
X_C7C9_n   X_C10C15_n   0];
x_q =       [0          0       0
0         0          0          0
0            0];
x_g =       [X_CO2_g    0       0
X_CH4_g  0          0          0
0            0];

%%    Solving    for    secondary
variables

K_n = (1./Pr).*exp(5.37.*( 1 + w
).*( 1 - 1./Tr ));
K_q = 10^6*(Pr./Tr);
precision = 1e-16;
e = [1 1];

while ( max(e) > precision )
    x_n(1) = x_g(1)/K_n(1);
    x_q(1) = x_g(1)/K_q(1);
    x_n(4) = x_g(4)/K_n(4);
    x_q(4) = x_g(4)/K_q(4);
    x_g(5:8)                    =
x_n(5:8).*K_n(5:8);
    x_q(5:8)                    =
x_g(5:8)./K_q(5:8);
    A  =  [1  1  1;  K_n(2)  K_n(3)
K_n(9);             K_n(2)/K_q(2)
K_n(3)/K_q(3) K_n(9)/K_q(9)];
    B = [1-x_n(1)-sum(x_n(4:8));
1-x_g(1)-sum(x_g(4:8));       1-
x_q(1)-sum(x_q(4:8))];
    result = (A\B)';
    if ( result(1)/K_q(2)*K_n(2)
< .9 )
        result(1)               =
0.98*K_q(2)/K_n(2);
    end
    if ( result(2) < 0 )
        result(2)               =
0.001/100/K_n(3);
    end
    if ( result(3) < 0)
        result(3) = 0.2;
    end
    x_n(2:3) = result(1:2);
    x_n(9) = result(3);
    x_g(2:3)                    =
K_n(2:3).*x_n(2:3);
    x_g(9) =  K_n(9)*x_n(9);
    x_q(2:3)                    =
x_g(2:3)./K_q(2:3);
    x_q(9) =  x_g(9)/K_q(9);
    % normalizing gaseous phase
    excess = sum(x_g)-1;
    exc_app                     =
excess/(sum(x_g(2:3))+sum(x_g(5:
9)));
    x_g(2:3)    =    x_g(2:3)*(1-
exc_app);
    x_g(5:9)    =    x_g(5:9)*(1-
exc_app);
```

```
    % normalizing non-aqueous
phase
    excess = sum(x_n)-1;
    exc_app                    =
excess/(sum(x_n(1:4))+x_n(9));
    x_n(1:4)    =    x_n(1:4)*(1-
exc_app);
    x_n(9) = x_n(9)*(1-exc_app);
    % normalizing aqueous phase
    x_q = x_q./sum(x_q);

    a_T_g     =     sum(     sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_g = sum(x_g.*b_i);
    a_T_n     =     sum(     sum(
(x_n'*x_n).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_n = sum(x_n.*b_i);
    a_T_q     =     sum(     sum(
(x_q'*x_q).*sqrt(a_Ti'*a_Ti).*(1
-tau) ) );
    b_q = sum(x_q.*b_i);
    A_g = a_T_g*p_g/(R^2*T^2);
    B_g = b_g*p_g/(R*T);
    A_n = a_T_n*p_n/(R^2*T^2);
    B_n = b_n*p_n/(R*T);
    A_q = a_T_q*p_q/(R^2*T^2);
    B_q = b_q*p_q/(R*T);
    r  =  roots([1  B_g-1  (A_g  -
2*B_g - 3*B_g^2) (B_g^3 + B_g^2 -
A_g*B_g)]);
    u  =  roots([1  B_n-1  (A_n  -
2*B_n - 3*B_n^2) (B_n^3 + B_n^2 -
A_n*B_n)]);
    t  =  roots([1  B_q-1  (A_q  -
2*B_q - 3*B_q^2) (B_q^3 + B_q^2 -
A_q*B_q)]);
    r1  =  r(imag(r)==0);  u1  =
u(imag(u)==0);        t1        =
t(imag(t)==0);
    u2   =   u1(u1>B_n);   t2   =
t1(t1>B_q);
    Z_g = max(r1);
    Z_n = min(u2);
    Z_q = min(t2);
    A_prime_g      =      2*sum(
(ones(n,1)*x_g).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_g;
    A_prime_n      =      2*sum(
(ones(n,1)*x_n).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_n;
    A_prime_q      =      2*sum(
(ones(n,1)*x_q).*sqrt(a_Ti'*a_Ti
).*(1-tau), 2)'/a_T_q;
    B_prime_g = b_i/b_g;
    B_prime_n = b_i/b_n;
    B_prime_q = b_i/b_q;
    phi_g = exp( -log( Z_g - B_g
) + ( Z_g - 1 )*B_prime_g -
A_g/B_g/...
        2^1.5*(    A_prime_g    -
B_prime_g   )*log(   (   Z_g   +
B_g*(sqrt(2)+1) )/...
        ( Z_g - B_g*(sqrt(2)-1) )
) );
    phi_n = exp( -log( Z_n - B_n
) + ( Z_n - 1 )*B_prime_n -
A_n/B_n/...
        2^1.5*(    A_prime_n    -
B_prime_n   )*log(   (   Z_n   +
B_n*(sqrt(2)+1) )/...
        ( Z_n - B_n*(sqrt(2)-1) )
) );
    phi_q = exp( -log( Z_q - B_q
) + ( Z_q - 1 )*B_prime_q -
A_q/B_q/...
        2^1.5*(    A_prime_q    -
B_prime_q   )*log(   (   Z_q   +
B_q*(sqrt(2)+1) )/...
        ( Z_q - B_q*(sqrt(2)-1) )
) );
    K_n_new                    =
(p_n/p_g)*phi_n./phi_g;
    K_q_new                    =
(p_q/p_g)*phi_q./phi_g;
    e = [ max( ( K_n_new - K_n
).^2 ./ ( K_n.*K_n_new ) )
        max( ( K_q_new - K_q ).^2
./ ( K_q.*K_q_new ) )];
    K_n = K_n_new;
    K_q = K_q_new;
end

Z_n_corr       =       Z_n       -
p_n*sum(x_n.*c)/(R*T);
Z_q_corr       =       Z_q       -
p_q*sum(x_q.*c)/(R*T);
Z_g_corr       =       Z_g       -
p_g*sum(x_g.*c)/(R*T);
```

```
%% Calculating outputs

output = zeros(N_row+4, N_phase);

% calculating total molar
concentration for each phase
(mole/m^3)
MC_n                          =
p_n*6894.76/(Z_n_corr*R_SI*T_SI)
;
MC_q                          =
p_q*6894.76/(Z_q_corr*R_SI*T_SI)
;
MC_g                          =
p_g*6894.76/(Z_g_corr*R_SI*T_SI)
;

% phase fractions
denom = MC_n*S_n + MC_q*S_q +
MC_g*S_g;
psi_n = MC_n*S_n/denom;
psi_q = MC_q*S_q/denom;
psi_g = MC_g*S_g/denom;
output(1,:) = [psi_n, psi_q,
psi_g];

% calculating molecular weight of
phases
MW_n = sum( MW.*x_n );
MW_q = sum( MW.*x_q );
MW_g = sum( MW.*x_g );
output(2,:) = [MW_n, MW_q, MW_g];

% compresibility factors
output(3,:)     =     [Z_n_corr,
Z_q_corr, Z_g_corr];

% calculating mass density for
each phase (kg/m^3)
rho_g = MC_g * MW_g/1000;
rho_n = MC_n * MW_n/1000;
rho_q = MC_q * MW_q/1000;
output(4,:) = [rho_n, rho_q,
rho_g];

% mole percents
output(5:N_row+4,:) = [x_n' x_q'
x_g']*100;

%% State change signaling

% calculating total mole
fractions
x_t = psi_n.*x_n + psi_q.*x_q +
psi_g.*x_g;

% forming Q-functions
Q1    =    @(psi_n,    psi_q)sum(
x_t.*K_q.*(1-K_n)./(  K_n.*K_q  +
...
    psi_n*K_q.*(1-K_n)         +
psi_q*K_n.*(1-K_q) ) );
Q2    =    @(psi_n,    psi_q)sum(
x_t.*K_n.*(1-K_q)./(  K_n.*K_q  +
...
    psi_n*K_q.*(1-K_n)         +
psi_q*K_n.*(1-K_q) ) );

current_state = 1;
correct_state = 1;

% single-phase test
% checking for single gaseous
phase
if ( sum(x_t./K_n) < 1 ) && (
sum(x_t./K_q) < 1 )
    correct_state = 3;

% two-phase tests
% checking for gaseous phase and
non-aqueous phase
else
    auxfunc = @(psi_n) Q1(psi_n,
0);
    psi_1 = fzero(auxfunc, 0.5);
    if ( Q2(psi_1, 0) < 1e-5 ) &&
(  sum(x_t./K_n)  >  1  )  &&  (
sum(x_t.*K_n) > 1 )
        if ( psi_1 > 1e-5 )
            correct_state = 4;
        else
            correct_state = 3;
        end

    % checking for gaseous phase
and aqueous phase
    else
        auxfunc = @(psi_q) Q2(0,
psi_q);
        psi_2  =  fzero(auxfunc,
0.5);
```

```
        if ( Q1(0, psi_2) < 1e-5
) && ( sum(x_t./K_q) > 1 ) && (
sum(x_t.*K_q) > 1 )
            if ( psi_2 > 1e-5 )
                correct_state   =
2;
            else
                correct_state   =
3;
            end
        end
    end
end

if     (     correct_state     ==
current_state )
    fprintf('\nCurrent  state  is
confirmed\n');
else
    fprintf('\nCurrent  state  is
not correct\n');
    fprintf('Switch   to   state
%d\n', correct_state);
end

end

*******************************
***

function    [    output    ]    =
Flash_PR_State2( T_C, input )
%State 2
%  Gaseous and Aqueous phases are
present.
% Inputs: T_C: temperature in
Centrigrade
%        input: 1st row: Aqueous
phase pressure
%                      2nd:
Aqueous phase saturation
%                3rd row to end:
primary mole percents
%   Outputs:   1st   row:   phase
fractions
%               2nd  row:  phases'
compresibility factors
%           3rd row: phases' mass
density (kg/m^3)
%             4th row: molecular
weight of phases
%               5th row to end:
equilibrium  mole  fractions  of
components
%           Columns: 1st column:
gaseous
%                    2nd column:
aqueous

format long;

%% Input form CompFlow

% pressure for aqueous, non-
aqueous and gaseous phases (bar)
global p_q; global p_g;
p_q = input(1); p_g = p_q; % no
capillary pressure data currently
available
p_q  =  p_q*14.5037738;  p_g  =
p_g*14.5037738; % converting bar
to psi

% saturations (volume fractions)
of each phase
global S_q; global S_g;
S_q = input(2); S_g = 1 - S_q;

N_row = 9;
N_phase = 2;

% converting mole percents to
fractions
input(3:N_row)                 =
input(3:N_row)/100;

%  known  molar  fractions  in
gaseous phase
global X_CH4_g; global X_CO2_g;
global X_C2C3_g; global X_C4C6_g;
global      X_C7C9_g;      global
X_C10C15_g; global X_C16p_g;
X_CO2_g  =  input(8);  X_CH4_g  =
input(9);  X_C2C3_g  =  input(3);
X_C4C6_g = input(4);
X_C7C9_g = input(5); X_C10C15_g =
input(6); X_C16p_g = input(7);

%% Setting physical properties &
calculating dependent variables
```

```
T = 1.8*T_C + 491.67; % temperature conversion to Rankin
T_SI = T_C + 273.15; % temperature conversion to Kelvin

% critical pressures
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Pc =        [73.76  220.55  33.94
45.99   46.56  34.24  25.80  18.60
9.61];
Pc = Pc*14.5037738; % conversion
of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Tc =         [304.2  647.1  126.2
190.6   328.5   458.1   566.0   651
824.1];
Tc = Tc*1.8; % conversoion of
critical temperatures from Kelvin
to Rankin

% acentric factors
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
w =         [0.239  0.345   0.0403
0.0115  0.118  0.234  0.370  0.595
1.427];

% molecular weight
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
MW =        [44.0   18.0    28.0
16.0    35.0   70.1   108.3  166.0
385.6];

% dimensionless volume shifts
% component   CO2      H2O     N2
CH4       C2C3      C4C6      C7C9
C10C15  C16+
s =       [  0.02   .0147   -.1927
-.1595    -.095     -.047     .038
.155    .277];

% calculating reduced parameters
Pr = p_q./Pc;
Tr = T./Tc;

% binary interaction coefficients
for non-aueous and gaseous phases
% component    CO2       H2O     N2
CH4       C2C3       C4C6      C7C9
C10C15   C16+
sigma =       [0         .19     0
.107      .128       .422      .1
.015     .015
                .19              0
.478     .5        .49        .48
.48     .48       .48
                0         .478    0
.031      .068       .335      .15
.155     .155
                .107             .5
.031     0         .008       .066
.043    .052      .066
                .128             .49
.068     .008      0              .03
.029    .026      .037
                .422             .48
.335    .066    .03     0      .016
.006     .01
                .1               .48
.15     .043    .029     .016    0
0        0
                .015             .48
.155    .052    .026     .006    0
0        0
                .015             .48
.155    .066    .037     .01     0
0        0];

% binary interaction coefficients
for aqueous phase
tau = sigma;
% slope and intercepts of linear
regression
slint = [0.273   -0.371
         0       0
         0.417   -1.631
         1.66    -0.759
         1.991   -0.576
         3.124   -0.687
         .00151 -.812];
% water-CO2
tau(2,1)  =  slint(1,1)*Tr(1)  +
slint(1,2);
```

```
% water-N2
tau(2,3)  =  slint(3,1)*Tr(3)  +
slint(3,2);
% water-methane
tau(2,4)                       =
slint(4,1)*Tr(4)*Pc(4)/Pc(2)   +
slint(4,2);
% water-C2C3
tau(2,5)           =           (
.5*slint(5,1)+.5*slint(6,1)
)*Tr(5)*Pc(5)/Pc(2) + ...
    (
.5*slint(5,2)+.5*slint(6,2) );
% water-C4C6
tau(2,6)  =  slint(7,1)*T_SI  +
slint(7,2);
% water-C7C9
tau(2,7) = tau(6,1);
% water-C10C15
tau(2,8) = tau(6,1);
% water-C16+
tau(2,9) = tau(6,1);

tau(:,2) = tau(2,:);

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)               -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units
R_SI = 8.3144598; % gas cosntants
in SI units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;
b_i = 0.07796074*R*Tc./Pc;
c = s.*b_i; % volume shifts
n = size(sigma,2);
number = 0;

% molar fraction in phases
% component  CO2        H2O     N2
CH4      C2C3      C4C6      C7C9
C10C15       C16+
x_q =       [0          0       0
0         0          0          0
0            0];
x_g =       [X_CO2_g    0       0
X_CH4_g     X_C2C3_g     X_C4C6_g
X_C7C9_g   X_C10C15_g   X_C16p_g];

%%    Solving    for    secondary
variables

K_q = 10^6*(Pr./Tr);
precision = 1e-16;
e = 1;

while ( e > precision )
    number = number + 1;
    x_q = x_g./K_q;
    A = [1 1; K_q(2) K_q(3)];
    B = [1-sum(x_q(4:9))-x_q(1);
1-sum(x_g(4:9))-x_g(1)];
    result = (A\B)';
    if ( result(1) < 0 )
        result(1) = 0.98;
    end
    if ( result(2) < 0 )
        result(2)              =
0.15/100/K_q(3);
    end
    x_q(2:3) = result;
    x_g(2:3)                   =
x_q(2:3).*K_q(2:3);
    % normalizing gaseous phase
    excess = sum(x_g)-1;
    exc_app                    =
excess/sum(x_g(2:3));
    x_g(2:3)    =    x_g(2:3)*(1-
exc_app);
    % normalizing aqueous phase
    x_q = x_q./sum(x_q);

    a_T_g     =     sum(     sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_g = sum(x_g.*b_i);
```

```
    a_T_q     =     sum(     sum(
(x_q'*x_q).*sqrt(a_Ti'*a_Ti).*(1
-tau) ) );
    b_q = sum(x_q.*b_i);
    A_g = a_T_g*p_g/(R^2*T^2);
    B_g = b_g*p_g/(R*T);
    A_q = a_T_q*p_q/(R^2*T^2);
    B_q = b_q*p_q/(R*T);
    r  =  roots([1  B_g-1  (A_g  -
2*B_g - 3*B_g^2) (B_g^3 + B_g^2 -
A_g*B_g)]);
    t  =  roots([1  B_q-1  (A_q  -
2*B_q - 3*B_q^2) (B_q^3 + B_q^2 -
A_q*B_q)]);
    r1  =  r(imag(r)==0);  t1  =
t(imag(t)==0);
    t2 = t1(t1>B_q);
    Z_g = max(r1);
    Z_q = min(t2);
    A_prime_g      =      2*sum(
(ones(n,1)*x_g).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_g;
    A_prime_q      =      2*sum(
(ones(n,1)*x_q).*sqrt(a_Ti'*a_Ti
).*(1-tau), 2)'/a_T_q;
    B_prime_g = b_i/b_g;
    B_prime_q = b_i/b_q;
    phi_g = exp( -log( Z_g - B_g
) + (  Z_g  -  1  )*B_prime_g  -
A_g/B_g/...
        2^1.5*(    A_prime_g    -
B_prime_g   )*log(   (   Z_g   +
B_g*(sqrt(2)+1) )/...
        ( Z_g - B_g*(sqrt(2)-1) )
) );
    phi_q = exp( -log( Z_q - B_q
) + (  Z_q  -  1  )*B_prime_q  -
A_q/B_q/...
        2^1.5*(    A_prime_q    -
B_prime_q   )*log(   (   Z_q   +
B_q*(sqrt(2)+1) )/...
        ( Z_q - B_q*(sqrt(2)-1) )
) );
    K_q_new                      =
(p_q/p_g)*phi_q./phi_g;
    e = max( ( K_q_new - K_q ).^2
./ ( K_q.*K_q_new ) );
    K_q = K_q_new;
end
```

```
Z_q_corr      =      Z_q      -
p_q*sum(x_q.*c)/(R*T);
Z_g_corr      =      Z_g      -
p_g*sum(x_g.*c)/(R*T);

%% Calculating outputs

output = zeros(N_row+4, N_phase);

%   calculating   total   molar
concentration   for   each   phase
(mole/m^3)
MC_q                           =
p_q*6894.76/(Z_q_corr*R_SI*T_SI)
;
MC_g                           =
p_g*6894.76/(Z_g_corr*R_SI*T_SI)
;

% phase fractions
denom = MC_q*S_q + MC_g*S_g;
psi_q = MC_q*S_q/denom;
psi_g = MC_g*S_g/denom;
output(1,:) = [psi_q, psi_g];

% calculating molecular weight of
phases
MW_q = sum( MW.*x_q );
MW_g = sum( MW.*x_g );
output(2,:) = [MW_q, MW_g];

% compresibility factors
output(3,:)     =     [Z_q_corr,
Z_g_corr];

% calculating mass density for
each phase (kg/m^3)
rho_q = MC_q * MW_q/1000;
rho_g = MC_g * MW_g/1000;
output(4,:) = [rho_q, rho_g];

% mole percents
output(5:N_row+4,:)    =    [x_q'
x_g']*100;

%% State change signaling

%    calculating    total    mole
fractions
x_t = psi_q.*x_q + psi_g.*x_g;
```

```
% using AuxiliaryN.m function to calculate equilibrium ratios between
% hypothetical non-aqeuous phase and available gaseous phase
K_n = AuxiliaryN(p_g, T_C, x_g);

% forming Q-functions
Q1    =    @(psi_n,    psi_q)sum( x_t.*K_q.*(1-K_n)./(  K_n.*K_q  + ...
    psi_n*K_q.*(1-K_n)          + psi_q*K_n.*(1-K_q) ) );
Q2    =    @(psi_n,    psi_q)sum( x_t.*K_n.*(1-K_q)./(  K_n.*K_q  + ...
    psi_n*K_q.*(1-K_n)          + psi_q*K_n.*(1-K_q) ) );

current_state = 2;
correct_state = 1;

% single-phase test
%  checking  for  single  gaseous phase
if  ( sum(x_t./K_n) < 1 ) && ( sum(x_t./K_q) < 1 )
    correct_state = 3;

% two-phase tests
% checking for gaseous phase and non-aqueous phase
else
    auxfunc = @(psi_n) Q1(psi_n, 0);
    psi_1 = fzero(auxfunc, 0.5);
    if ( Q2(psi_1, 0) < 1e-5 ) &&  (  sum(x_t./K_n)  >  1  )  &&  ( sum(x_t.*K_n) > 1 )
        if ( psi_1 > 1e-5 )
            correct_state = 4;
        else
            correct_state = 3;
        end

    % checking for gaseous phase and aqueous phase
    else
        auxfunc = @(psi_q) Q2(0, psi_q);
        psi_2  =  fzero(auxfunc, 0.5);
        if ( Q1(0, psi_2) < 1e-5 ) && ( sum(x_t./K_q) > 1 ) && ( sum(x_t.*K_q) > 1 )
            if ( psi_2 > 1e-5 )
                correct_state   = 2;
            else
                correct_state   = 3;
            end
        end
    end
end

if     (     correct_state     == current_state )
    fprintf('\nCurrent  state  is confirmed\n');
else
    fprintf('\nCurrent  state  is not correct\n');
    fprintf('Switch   to   state %d\n', correct_state);
end

end

********************************
***

function    [    output    ]    = Flash_PR_State3( T_C, input )
%State 4
%  Only Gaseous phase is present.
%  Inputs:  T_C:  temperature  in Centrigrade
%          input: Rows: 1st row: gaseous phase pressure
%
2nd  row  to  end:  primary  mole percents
%   Outputs:   1st   row:   phase fraction
%                2nd  row:  phase compresibility factor
%            3rd row: phase mass density (kg/m^3)
%             4th row: molecular weight of phases
```

```
%               5th row to end: equilibrium mole fractions of components

format long;

%% Input form CompFlow

% pressure for non-aqueous and gaseous phases (bar)
global p_g;
p_g = input(1);
p_g    =    p_g*14.5037738;    % converting bar to psi

N_row = 9;
N_phase = 1;

% converting mole percents to fractions
input(2:N_row)                 = input(2:N_row)/100;

% known molar fractions in gaseous phase
global X_CO2_g; global X_H2O_g; global X_CH4_g; global X_C2C3_g; global X_C4C6_g; global X_C7C9_g; global     X_C10C15_g;     global X_C16p_g;
X_CO2_g  =  input(8);  X_H2O_g  =  input(2); X_CH4_g = input(9);     X_C2C3_g  =  input(3);  X_C4C6_g  =  input(4); X_C7C9_g = input(5);
X_C10C15_g = input(6); X_C16p_g = input(7);

%% Setting physical properties & calculating dependent variables

T    =    1.8*T_C    +    491.67;    %temperature conversion to Rankin
T_SI = T_C + 273.15; %temperature conversion to Kelvin

% critical pressures
% component   CO2     H2O      N2      CH4    C2C3   C4C6   C7C9   C10C15  C16+
Pc =        [73.76  220.55  33.94  45.99   46.56  34.24  25.80  18.60  9.61];
Pc = Pc*14.5037738; % conversion of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2      CH4    C2C3   C4C6   C7C9   C10C15  C16+
Tc =         [304.2  647.1  126.2  190.6   328.5   458.1   566.0   651  824.1];
Tc = Tc*1.8; % conversoion of critical temperatures from Kelvin to Rankin

% acentric factors
% component   CO2     H2O      N2      CH4    C2C3   C4C6   C7C9   C10C15  C16+
w =         [0.239  0.345   0.0403  0.0115  0.118  0.234  0.370  0.595  1.427];

% molecular weight
% component   CO2     H2O      N2      CH4    C2C3   C4C6   C7C9   C10C15  C16+
MW =         [44.0   18.0    28.0  16.0    35.0   70.1   108.3  166.0  385.6];

% dimensionless volume shifts
% component   CO2      H2O     N2      CH4       C2C3      C4C6      C7C9  C10C15  C16+
s =       [  0.02   .0147   -.1927  -.1595    -.095     -.047     .038  .155    .277];

% calculating reduced parameters
Tr = T./Tc;

% binary interaction coefficients for non-aueous and gaseous phases
% component    CO2      H2O     N2      CH4       C2C3       C4C6     C7C9  C10C15   C16+
```

```
sigma =       [0         .19     0
.107      .128        .422      .1
.015     .015
                .19              0
.478     .5         .49         .48
.48     .48       .48
                 0         .478    0
.031      .068       .335      .15
.155     .155
                .107             .5
.031     0          .008       .066
.043    .052     .066
                .128             .49
.068     .008     0              .03
.029    .026     .037
                .422             .48
.335    .066    .03     0       .016
.006     .01
                .1               .48
.15     .043    .029     .016    0
0        0
                .015             .48
.155    .052    .026     .006    0
0        0
                .015             .48
.155    .066    .037     .01     0
0        0];

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)                -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units
R_SI = 8.3144598; % gas cosntants
in SI units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;

b_i = 0.07796074*R*Tc./Pc;
c = s.*b_i; % volume shifts

%  molar  fraction  in  phases
(primary variables)
% component  CO2       H2O     N2
CH4      C2C3      C4C6      C7C9
C10C15       C16+
x_g =        [X_CO2_g     X_H2O_g
0     X_CH4_g  X_C2C3_g  X_C4C6_g
X_C7C9_g   X_C10C15_g   X_C16p_g];

x_g(3) = 1 - sum(x_g);
a_T_g      =      sum(      sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
b_g = sum(x_g.*b_i);
A_g = a_T_g*p_g/(R^2*T^2);
B_g = b_g*p_g/(R*T);
r = roots([1 B_g-1 (A_g - 2*B_g -
3*B_g^2)   (B_g^3   +   B_g^2   -
A_g*B_g)]);
r1 = r(imag(r)==0);
Z_g = max(r1);
Z_g_corr      =      Z_g      -
p_g*sum(x_g.*c)/(R*T);

%% Calculating outputs

output = zeros(N_row+4, N_phase);

%   calculating   total   molar
concentration (mole/m^3)
MC_g                            =
p_g*6894.76/(Z_g_corr*R_SI*T_SI)
;

psi_g   =   1;   %setting   phase
fraction
output(1,:) = psi_g;

%  calculating  phase  molecular
weight
MW_g = sum( MW.*x_g );
output(2,:) = MW_g;

% compresibility factors
output(3,:) = Z_g_corr;

% calculating phase mass density
(kg/m^3)
```

```
rho_g = MC_g * MW_g/1000;
output(4,:) = rho_g;

% mole percents
output(5:N_row+4,:) = x_g'*100;

%% State change signaling

% calculating total mole
fractions
x_t = psi_g.*x_g;

% using auxiliary functions to
calculate equilibrium ratios
between
% hypothetical non-aqeuous and
aqueous phases and available
gaseous phase
K_n = AuxiliaryN(p_g, T_C, x_g);
K_q = AuxiliaryQ(p_g, T_C, x_g);

% forming Q-functions
Q1 = @(psi_n, psi_q)sum(
x_t.*K_q.*(1-K_n)./( K_n.*K_q +
...
    psi_n*K_q.*(1-K_n)          +
psi_q*K_n.*(1-K_q) ) );
Q2 = @(psi_n, psi_q)sum(
x_t.*K_n.*(1-K_q)./( K_n.*K_q +
...
    psi_n*K_q.*(1-K_n)          +
psi_q*K_n.*(1-K_q) ) );

current_state = 3;
correct_state = 1;

% single-phase test
% checking for single gaseous
phase
if ( sum(x_t./K_n) < 1 ) && (
sum(x_t./K_q) < 1 )
    correct_state = 3;

% two-phase tests
% checking for gaseous phase and
non-aqueous phase
else
    auxfunc = @(psi_n) Q1(psi_n,
0);
    psi_1 = fzero(auxfunc, 0.5);
    if ( Q2(psi_1, 0) < 1e-5 ) &&
(  sum(x_t./K_n)  >  1  )  &&  (
sum(x_t.*K_n) > 1 )
        if ( psi_1 > 1e-5 )
            correct_state = 4;
        else
            correct_state = 3;
        end

    % checking for gaseous phase
and aqueous phase
    else
        auxfunc = @(psi_q) Q2(0,
psi_q);
        psi_2  =  fzero(auxfunc,
0.5);
        if ( Q1(0, psi_2) < 1e-5
) && ( sum(x_t./K_q) > 1 ) && (
sum(x_t.*K_q) > 1 )
            if ( psi_2 > 1e-5 )
                correct_state  =
2;
            else
                correct_state  =
3;
            end
        end
    end
end

if    (    correct_state    ==
current_state )
    fprintf('\nCurrent state is
confirmed\n');
else
    fprintf('\nCurrent state is
not correct\n');
    fprintf('Switch  to  state
%d\n', correct_state);
end

end

*******************************
***

function   [   output   ]   =
Flash_PR_State4( T_C, input )
%State 4
%  Gaseous and Non-Aqueous phases
are present.
```

```
% Inputs: T_C: temperature in Centrigrade
%          input: 1st row: Non-aqueous phase pressure
%                 2nd row: Non-aqueous saturation
%                  3rd row to end: primary mole percents
%  Outputs:  1st  row:  phase fractions
%             2nd row: phases' compresibility factors
%          3rd row: phases' mass density (kg/m^3)
%            4th row: molecular weight of phases
%              5th row to end: equilibrium mole fractions of components
%           Columns: 1st column: non-aqueous
%                    2nd column: gaseous

format long;

%% Input form CompFlow

% pressure for non-aqueous and gaseous phases (bar)
global p_n; global p_g;
p_n = input(1); p_g = p_n; % no capillary pressure data currently available
p_n = p_n*14.5037738; p_g = p_g*14.5037738; % converting bar to psi

% saturations (volume fractions) of each phase
global S_n; global S_g;
S_n = input(2); S_g = 1 - S_n;

N_row = 9;
N_phase = 2;

% converting mole percents to fractions
input(3:N_row)                  = input(3:N_row)/100;

% known molar fractions in gaseous phase
global X_CH4_g; global X_CO2_g; global X_H2O_g;
X_CO2_g = input(8); X_H2O_g = input(3); X_CH4_g = input(9);

% known molar fractions in non-aqueous phaseh
global X_C2C3_n; global X_C4C6_n; global X_C7C9_n; global X_C10C15_n;
X_C2C3_n = input(4); X_C4C6_n = input(5); X_C7C9_n = input(6);
X_C10C15_n = input(7);

%% Setting physical properties & calculating dependent variables

T    =    1.8*T_C    +    491.67; %temperature conversion to Rankin
T_SI = T_C + 273.15; %temperature conversion to Kelvin

% critical pressures
% component   CO2     H2O      N2    CH4    C2C3   C4C6   C7C9   C10C15   C16+
Pc =        [73.76  220.55  33.94  45.99  46.56  34.24  25.80  18.60  9.61];
Pc = Pc*14.5037738; % conversion of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2    CH4    C2C3   C4C6   C7C9   C10C15   C16+
Tc =         [304.2  647.1  126.2  190.6  328.5  458.1  566.0  651  824.1];
Tc = Tc*1.8; % conversoion of critical temperatures from Kelvin to Rankin

% acentric factors
% component   CO2     H2O      N2    CH4    C2C3   C4C6   C7C9   C10C15   C16+
```

```
w =         [0.239  0.345   0.0403
0.0115  0.118  0.234  0.370  0.595
1.427];

% molecular weight
% component   CO2      H2O       N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
MW =         [44.0   18.0    28.0
16.0    35.0   70.1   108.3  166.0
385.6];

% dimensionless volume shifts
% component   CO2       H2O      N2
CH4       C2C3      C4C6      C7C9
C10C15  C16+
s =       [  0.02   .0147   -.1927
-.1595    -.095     -.047     .038
.155    .277];

% calculating reduced parameters
Pr = p_n./Pc;
Tr = T./Tc;

% binary interaction coefficients
for non-aueous and gaseous phases
% component    CO2       H2O     N2
CH4       C2C3       C4C6      C7C9
C10C15   C16+
sigma =       [0         .19     0
.107      .128       .422      .1
.015     .015
                .19              0
.478     .5         .49         .48
.48     .48       .48
                0         .478    0
.031      .068       .335      .15
.155     .155
                .107             .5
.031     0          .008       .066
.043    .052      .066
                .128             .49
.068     .008      0              .03
.029    .026      .037
                .422             .48
.335   .066    .03     0      .016
.006     .01
                .1               .48
.15     .043    .029     .016    0
0        0
                .015             .48
.155    .052    .026     .006    0
0        0
                .015             .48
.155    .066    .037     .01     0
0        0];

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)                -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units
R_SI = 8.3144598; % gas cosntants
in SI units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;
b_i = 0.07796074*R*Tc./Pc;
c = s.*b_i; % volume shifts
n = size(sigma,2);
number = 0;

%  molar  fraction  in  phases
(primary variables)
% component  CO2        H2O     N2
CH4      C2C3      C4C6      C7C9
C10C15       C16+
x_n =       [0          0       0
0            X_C2C3_n   X_C4C6_n
X_C7C9_n   X_C10C15_n   0];
x_g =       [X_CO2_g    X_H2O_g 0
X_CH4_g  0          0          0
0            0];

%%   Solving   for   secondary
variables
```

```
K_n = (1./Pr).*exp(5.37.*( 1 + w
).*( 1 - 1./Tr ));
precision = 1e-16;
e = 1;

while ( e > precision )
    number = number + 1;
    x_n(1:2)                    =
x_g(1:2)./K_n(1:2);
    x_n(4) = x_g(4)/K_n(4);
    x_g(5:8)                    =
x_n(5:8).*K_n(5:8);
    A = [1 1; K_n(3) K_n(9)];
    B     =     [1-sum(x_n(1:2))-
sum(x_n(4:8));   1-sum(x_g(1:2))-
sum(x_g(4:8))];
    result = (A\B)';
    if ( result(1) < 0 )
        result(1) = 0.05/100;
    end
    if ( result(2) < 0 )
        result(2) = 0.1;
    end
    x_n(3) = result(1);
    x_n(9) = result(2);
    x_g(3) = K_n(3)*x_n(3);
    x_g(9) =  K_n(9)*x_n(9);
    % normalizing gaseous phase
    excess = sum(x_g)-1;
    exc_app                     =
excess/(x_g(3)+sum(x_g(5:9)));
    x_g(3) = x_g(3)*(1-exc_app);
    x_g(5:9)    =    x_g(5:9)*(1-
exc_app);
    %   normalizing   non-aqueous
phase
    excess = sum(x_n)-1;
    exc_app                     =
excess/(sum(x_n(1:4))+x_n(9));
    x_n(1:4)    =    x_n(1:4)*(1-
exc_app);
    x_n(9) = x_n(9)*(1-exc_app);

    a_T_g     =     sum(     sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_g = sum(x_g.*b_i);
    a_T_n     =     sum(     sum(
(x_n'*x_n).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_n = sum(x_n.*b_i);
    A_g = a_T_g*p_g/(R^2*T^2);
    B_g = b_g*p_g/(R*T);
    A_n = a_T_n*p_n/(R^2*T^2);
    B_n = b_n*p_n/(R*T);
    r  =  roots([1  B_g-1  (A_g  -
2*B_g - 3*B_g^2) (B_g^3 + B_g^2 -
A_g*B_g)]);
    u  =  roots([1  B_n-1  (A_n  -
2*B_n - 3*B_n^2) (B_n^3 + B_n^2 -
A_n*B_n)]);
    r1  =  r(imag(r)==0);  u1  =
u(imag(u)==0);
    u2 = u1(u1>B_n);
    Z_g = max(r1);
    Z_n = min(u2);
    A_prime_g      =      2*sum(
(ones(n,1)*x_g).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_g;
    A_prime_n      =      2*sum(
(ones(n,1)*x_n).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_n;
    B_prime_g = b_i/b_g;
    B_prime_n = b_i/b_n;
    phi_g = exp( -log( Z_g - B_g
) + ( Z_g - 1 )*B_prime_g -
A_g/B_g/...
        2^1.5*(   A_prime_g   -
B_prime_g   )*log(   (   Z_g   +
B_g*(sqrt(2)+1) )/...
        ( Z_g - B_g*(sqrt(2)-1) )
) );
    phi_n = exp( -log( Z_n - B_n
) + ( Z_n - 1 )*B_prime_n -
A_n/B_n/...
        2^1.5*(   A_prime_n   -
B_prime_n   )*log(   (   Z_n   +
B_n*(sqrt(2)+1) )/...
        ( Z_n - B_n*(sqrt(2)-1) )
) );
    K_n_new                     =
(p_n/p_g)*phi_n./phi_g;
    e = max( ( K_n_new - K_n ).^2
./ ( K_n.*K_n_new ) );
    K_n = K_n_new;
end

Z_n_corr      =      Z_n      -
p_n*sum(x_n.*c)/(R*T);
Z_g_corr      =      Z_g      -
p_g*sum(x_g.*c)/(R*T);
```

```
%% Calculating outputs

output = zeros(N_row+4, N_phase);

% calculating total molar concentration for each phase (mole/m^3)
MC_n                              = p_n*6894.76/(Z_n_corr*R_SI*T_SI) ;
MC_g                              = p_g*6894.76/(Z_g_corr*R_SI*T_SI) ;

% phase fractions
denom = MC_n*S_n + MC_g*S_g;
psi_n = MC_n*S_n/denom;
psi_g = MC_g*S_g/denom;
output(1,:) = [psi_n, psi_g];

% calculating molecular weight of phases
MW_n = sum( MW.*x_n );
MW_g = sum( MW.*x_g );
output(2,:) = [MW_n, MW_g];

% compresibility factors
output(3,:)     =     [Z_n_corr, Z_g_corr];

% calculating mass density for each phase (kg/m^3)
rho_n = MC_n * MW_n/1000;
rho_g = MC_g * MW_g/1000;
output(4,:) = [rho_n, rho_g];

% mole percents
output(5:N_row+4,:)    =    [x_n' x_g']*100;

%% State change signaling

% calculating total mole fractions
x_t = psi_n.*x_n + psi_g.*x_g;

% using auxiliary functions to calculate equilibrium ratios between
% hypothetical aqueous phase and available gaseous phase
K_q = AuxiliaryQ(p_g, T_C, x_g);

% forming Q-functions
Q1    =    @(psi_n,    psi_q)sum( x_t.*K_q.*(1-K_n)./(  K_n.*K_q  + ...
    psi_n*K_q.*(1-K_n)          + psi_q*K_n.*(1-K_q) ) );
Q2    =    @(psi_n,    psi_q)sum( x_t.*K_n.*(1-K_q)./(  K_n.*K_q  + ...
    psi_n*K_q.*(1-K_n)          + psi_q*K_n.*(1-K_q) ) );

current_state = 4;
correct_state = 1;

% single-phase test
% checking for single gaseous phase
if ( sum(x_t./K_n) < 1 ) && ( sum(x_t./K_q) < 1 )
    correct_state = 3;

% two-phase tests
% checking for gaseous phase and non-aqueous phase
else
    auxfunc = @(psi_n) Q1(psi_n, 0);
    psi_1 = fzero(auxfunc, 0.5);
    if ( Q2(psi_1, 0) < 1e-5 ) && (  sum(x_t./K_n)  >  1  )  &&  ( sum(x_t.*K_n) > 1 )
        if ( psi_1 > 1e-5 )
            correct_state = 4;
        else
            correct_state = 3;
        end

    % checking for gaseous phase and aqueous phase
    else
        auxfunc = @(psi_q) Q2(0, psi_q);
        psi_2  =  fzero(auxfunc, 0.5);
        if ( Q1(0, psi_2) < 1e-5 ) && ( sum(x_t./K_q) > 1 ) && ( sum(x_t.*K_q) > 1 )
            if ( psi_2 > 1e-5 )
```

```
                correct_state   =
2;
            else
                correct_state   =
3;
            end
        end
    end
end

if     (     correct_state     ==
current_state )
    fprintf('\nCurrent  state  is
confirmed\n');
else
    fprintf('\nCurrent  state  is
not correct\n');
    fprintf('Switch   to   state
%d\n', correct_state);
end

end

*******************************
***

function [ K_n ] = AuxiliaryN(
pressure, T_C, x_g  )
%AuxiliaryN
%    Equilibrate present gaseous
phase  with  hypothetical  non-
aqueous phase
% Inputs: pressure: pressure in
bar
%            T_C: temperature in
Centrigrade
%          input: composition of
present   gaseous   phase   (mole
fraction)
%  Outputs:  equilibrium  ratios
between  present  gaseous  phase
with
%        hypothetical non-aqueous
phase

format long;

%% Inputs

% pressure for non-aqueous and
gaseous phases (bar)
global p_n; global p_g;
p_g = pressure; p_n = p_g;
p_n   =   p_n*14.5037738;   p_g   =
p_g*14.5037738; % converting bar
to psi

T    =    1.8*T_C    +    491.67;
%temperature conversion to Rankin

%% Setting physical properties &
calculating dependent variables

% critical pressures
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Pc =        [73.76  220.55  33.94
45.99   46.56  34.24  25.80  18.60
9.61];
Pc = Pc*14.5037738; % conversion
of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Tc =         [304.2  647.1  126.2
190.6   328.5   458.1   566.0   651
824.1];
Tc  =  Tc*1.8;  %  conversoion  of
critical temperatures from Kelvin
to Rankin

% acentric factors
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
w =         [0.239  0.345   0.0403
0.0115  0.118  0.234  0.370  0.595
1.427];

% calculating reduced parameters
Pr = p_n./Pc;
Tr = T./Tc;

% binary interaction coefficients
for non-aueous and gaseous phases
% component    CO2      H2O     N2
CH4      C2C3      C4C6     C7C9
C10C15   C16+
```

```
sigma =       [0         .19     0
.107      .128        .422      .1
.015     .015
                .19              0
.478     .5         .49         .48
.48     .48       .48
                0         .478    0
.031      .068       .335      .15
.155     .155
                .107             .5
.031     0          .008       .066
.043    .052      .066
                .128             .49
.068     .008      0              .03
.029    .026      .037
                .422             .48
.335    .066    .03     0       .016
.006     .01
                .1               .48
.15     .043    .029     .016    0
0        0
                .015             .48
.155    .052    .026     .006    0
0        0
                .015             .48
.155    .066    .037     .01     0
0        0];

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)                -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;
b_i = 0.07796074*R*Tc./Pc;
n = size(sigma,2);
number = 0;

%%    Solving    for    secondary
variables

K_n = (1./Pr).*exp(5.37.*( 1 + w
).*( 1 - 1./Tr ));
precision = 1e-16;
e = 1;

while ( e > precision )
    number = number + 1;
    x_n = x_g./K_n;
    %   normalizing   non-aqueous
phase
    x_n = x_n/sum(x_n);
    a_T_g     =     sum(     sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_g = sum(x_g.*b_i);
    a_T_n     =     sum(     sum(
(x_n'*x_n).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_n = sum(x_n.*b_i);
    A_g = a_T_g*p_g/(R^2*T^2);
    B_g = b_g*p_g/(R*T);
    A_n = a_T_n*p_n/(R^2*T^2);
    B_n = b_n*p_n/(R*T);
    r  =  roots([1  B_g-1  (A_g  -
2*B_g - 3*B_g^2) (B_g^3 + B_g^2 -
A_g*B_g)]);
    u  =  roots([1  B_n-1  (A_n  -
2*B_n - 3*B_n^2) (B_n^3 + B_n^2 -
A_n*B_n)]);
    r1  =  r(imag(r)==0);  u1  =
u(imag(u)==0);
    u2 = u1(u1>B_n);
    Z_g = max(r1);
    Z_n = min(u2);
    A_prime_g      =      2*sum(
(ones(n,1)*x_g).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_g;
    A_prime_n      =      2*sum(
(ones(n,1)*x_n).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_n;
    B_prime_g = b_i/b_g;
    B_prime_n = b_i/b_n;
    phi_g = exp( -log( Z_g - B_g
) + ( Z_g - 1 )*B_prime_g -
A_g/B_g/...
```

```
        2^1.5*(    A_prime_g    -
B_prime_g   )*log(   (   Z_g   +
B_g*(sqrt(2)+1) )/...
        ( Z_g - B_g*(sqrt(2)-1) )
) );
    phi_n = exp( -log( Z_n - B_n
) + ( Z_n - 1 )*B_prime_n -
A_n/B_n/...
        2^1.5*(    A_prime_n    -
B_prime_n   )*log(   (   Z_n   +
B_n*(sqrt(2)+1) )/...
        ( Z_n - B_n*(sqrt(2)-1) )
) );
    K_n_new                      =
(p_n/p_g)*phi_n./phi_g;
    e = max( ( K_n_new - K_n ).^2
./ ( K_n.*K_n_new ) );
    K_n = K_n_new;
end

end

*******************************
***

function [ K_q ] = AuxiliaryQ(
pressure, T_C, x_g  )
%AuxiliaryQ
%    Equilibrate present gaseous
phase with hypothetical aqueous
phase
% Inputs: pressure: pressure in
bar
%            T_C: temperature in
Centrigrade
%          input: composition of
present   gaseous   phase   (mole
fraction)
%  Outputs:  equilibrium  ratios
between  present  gaseous  phase
with
%           hypothetical aqueous
phase

format long;

%% Inputs

% pressure for non-aqueous and
gaseous phases (bar)
global p_q; global p_g;
p_g = pressure; p_q = p_g;
p_q  =  p_q*14.5037738;  p_g  =
p_g*14.5037738; % converting bar
to psi

T    =    1.8*T_C    +    491.67;
%temperature conversion to Rankin
T_SI = T_C + 273.15; % temperature
conversion to Kelvin

%% Setting physical properties &
calculating dependent variables

% critical pressures
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Pc =        [73.76  220.55  33.94
45.99   46.56  34.24  25.80  18.60
9.61];
Pc = Pc*14.5037738; % conversion
of Pc from bar to psi

% critical temperatures
% component   CO2      H2O     N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
Tc =         [304.2  647.1  126.2
190.6   328.5   458.1   566.0   651
824.1];
Tc = Tc*1.8; % conversoion of
critical temperatures from Kelvin
to Rankin

% acentric factors
% component   CO2     H2O      N2
CH4    C2C3   C4C6   C7C9   C10C15
C16+
w =        [0.239  0.345   0.0403
0.0115  0.118  0.234  0.370  0.595
1.427];

% calculating reduced parameters
Pr = p_q./Pc;
Tr = T./Tc;

% binary interaction coefficients
for non-aueous and gaseous phases
% component    CO2      H2O     N2
CH4      C2C3      C4C6     C7C9
C10C15   C16+
```

```
sigma =       [0         .19     0
.107      .128        .422      .1
.015     .015
                .19              0
.478     .5         .49         .48
.48     .48       .48
                 0         .478    0
.031      .068        .335      .15
.155     .155
                .107              .5
.031     0          .008       .066
.043    .052      .066
                .128             .49
.068     .008      0              .03
.029    .026      .037
                .422             .48
.335    .066    .03     0       .016
.006     .01
                .1               .48
.15     .043    .029     .016    0
0        0
                .015             .48
.155    .052    .026     .006    0
0        0
                .015             .48
.155    .066    .037     .01     0
0        0];

% binary interaction coefficients
for aqueous phase
tau = sigma;
% slope and intercepts of linear
regression
slint = [0.273   -0.371
    0       0
    0.417  -1.631
    1.66   -0.759
    1.991  -0.576
    3.124  -0.687
    .00151  -.812];
% water-CO2
tau(2,1)  =  slint(1,1)*Tr(1)  +
slint(1,2);
% water-N2
tau(2,3)  =  slint(3,1)*Tr(3)  +
slint(3,2);
% water-methane
tau(2,4)                        =
slint(4,1)*Tr(4)*Pc(4)/Pc(2)    +
slint(4,2);
% water-C2C3
tau(2,5)           =           (
.5*slint(5,1)+.5*slint(6,1)
)*Tr(5)*Pc(5)/Pc(2) + ...
    (
.5*slint(5,2)+.5*slint(6,2) );
% water-C4C6
tau(2,6)   =   slint(7,1)*T_SI   +
slint(7,2);
% water-C7C9
tau(2,7) = tau(6,1);
% water-C10C15
tau(2,8) = tau(6,1);
% water-C16+
tau(2,9) = tau(6,1);

tau(:,2) = tau(2,:);

alpha = ( 1 + ( .37464 + 1.54226*w
- .26992*w.^2 ).*( 1 - sqrt(Tr) )
).^2;
% adjusment for big accentric
factors bigger thatn 0.49
alpha(w>.49) = ( 1 + ( .379642 +
1.48503*w(w>.49)                -
.164423*w(w>.49).^2 + ...
    .016666*w(w>.49).^3 ).*( 1 -
sqrt(Tr(w>.49) ) ) ).^2;
% adjustment for water component
if ( Tr(2) < 0.7225 )
    alpha(2)  =  (  1.0085677  +
0.82154*( 1 - sqrt(Tr(2)) ) )^2;
end

R = 10.732159; % gas cosntants in
Field units

a = 0.457235529*R^2*Tc.^2./Pc;
a_Ti = a.*alpha;
b_i = 0.07796074*R*Tc./Pc;
n = size(sigma,2);
number = 0;

%%   Solving   for   secondary
variables

K_q = 10^6*(Pr./Tr);
precision = 1e-16;
e = 1;

while ( e > precision )
    number = number + 1;
```

```
    x_q = x_g./K_q;
    %   normalizing   non-aqueous
phase
    x_q = x_q/sum(x_q);
    a_T_g     =     sum(     sum(
(x_g'*x_g).*sqrt(a_Ti'*a_Ti).*(1
-sigma) ) );
    b_g = sum(x_g.*b_i);
    a_T_q     =     sum(     sum(
(x_q'*x_q).*sqrt(a_Ti'*a_Ti).*(1
-tau) ) );
    b_q = sum(x_q.*b_i);
    A_g = a_T_g*p_g/(R^2*T^2);
    B_g = b_g*p_g/(R*T);
    A_q = a_T_q*p_q/(R^2*T^2);
    B_q = b_q*p_q/(R*T);
    r  =  roots([1  B_g-1  (A_g  -
2*B_g - 3*B_g^2) (B_g^3 + B_g^2 -
A_g*B_g)]);
    t  =  roots([1  B_q-1  (A_q  -
2*B_q - 3*B_q^2) (B_q^3 + B_q^2 -
A_q*B_q)]);
    r1  =  r(imag(r)==0);  t1  =
t(imag(t)==0);
    t2 = t1(t1>B_q);
    Z_g = max(r1);
    Z_q = min(t2);
    A_prime_g      =      2*sum(
(ones(n,1)*x_g).*sqrt(a_Ti'*a_Ti
).*(1-sigma), 2)'/a_T_g;
    A_prime_q      =      2*sum(
(ones(n,1)*x_q).*sqrt(a_Ti'*a_Ti
).*(1-tau), 2)'/a_T_q;
    B_prime_g = b_i/b_g;
    B_prime_q = b_i/b_q;
    phi_g = exp( -log( Z_g - B_g
)  +  (  Z_g  -  1  )*B_prime_g  -
A_g/B_g/...
        2^1.5*(    A_prime_g    -
B_prime_g   )*log(   (   Z_g   +
B_g*(sqrt(2)+1) )/...
        ( Z_g - B_g*(sqrt(2)-1) )
) );
    phi_q = exp( -log( Z_q - B_q
)  +  (  Z_q  -  1  )*B_prime_q  -
A_q/B_q/...
        2^1.5*(    A_prime_q    -
B_prime_q   )*log(   (   Z_q   +
B_q*(sqrt(2)+1) )/...
        ( Z_q - B_q*(sqrt(2)-1) )
) );
    K_q_new                     =
(p_q/p_g)*phi_q./phi_g;
    e = max( ( K_q_new - K_q ).^2
./ ( K_q.*K_q_new ) );
    K_q = K_q_new;
end

end

********************************
***
```