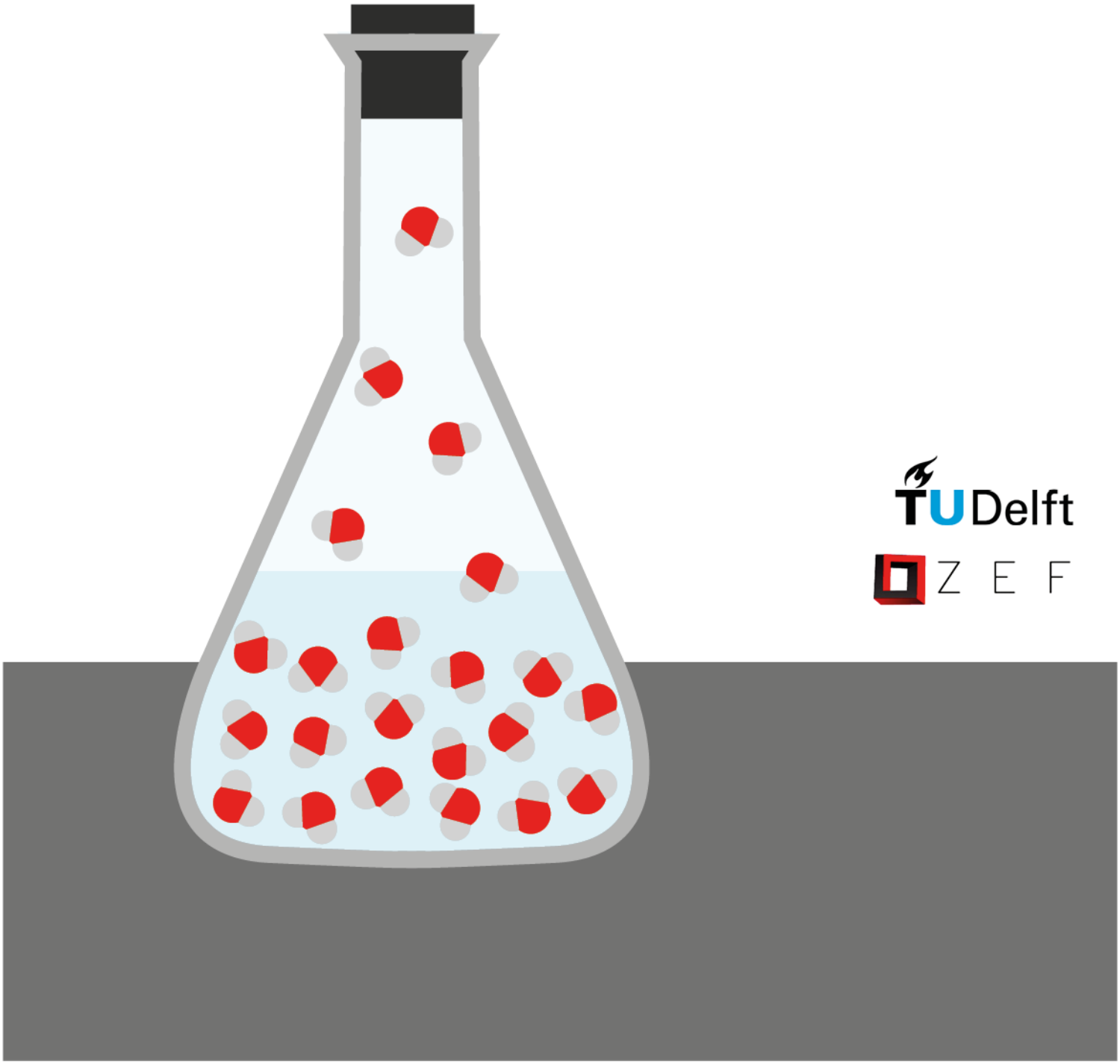
The Thermodynamic Stability Problem

from a Mathematical Optimisation Perspective

Lennart Kerckhoffs

September 2020



TU Delft

Z E F

M. Sc. Thesis report TECHNISCHE WISKUNDE

"The Thermodynamic Stability Problem, from a Mathematical Optimisation Perspective"

("Het thermodynamische stabiliteitsprobleem, vanuit een wiskundig optimaliseringsperspectief")

LENNART KERCKHOFFS (4309197)

Technische Universiteit Delft & ZEF B.V.

Supervisors

Dr.ir. J.H. Weber
Ir. J. van Kranendonk

Committee Members

Dr. J.L.A. Dubbeldam
Dr. ir. M. Ramdin

September, 2020 Delft

Preface & Acknowledgements

This master thesis is written as part of the curriculum for master students of Applied Mathematics at the Technical University Delft, and carried out at Zero Emission Fuels B.V., located on the TU Delft campus. It was spawned as an answer to the question: "Can this problem, improving the speed and stability of flash calculations and dynamic vapor-liquid-equilibrium simulations, which is rooted mostly in physics, be solved using mostly mathematics or by someone with very little background in physics?". Maybe as a result, this project has been a bit of a rabbit hole. The formulation at the start was about as detailed as "apply optimisation to improve the performance of flash calculations, which contain a state function that should be minimised". Which part of the flash employed this state function, what the state function looked like, or even what a state function or an equation of state is, were foreign to me at that point in time (just like all other thermodynamical terms mentioned in the work below).
The structure of this work thus loosely follows the path through the project: first figuring out the context and necessary terms, then mapping what a flash calculation looks like, figuring out which parts contain the "hard" mathematics and what causes the problems, and finally zoning in on a potential avenue for solving the problem (stability tests), and different methods to walk down this avenue.

I would like to thank Jan van Kranendonk, who was always ready to help with technical problems or interests, and whose enthusiasm always managed to (re-)spark mine. He is also the one that came up with the initial problem that spawned this thesis, and who helped me shape the objectives as the research evolved.
Next, I would like to thank my supervisor from the university, Jos Weber, for sticking with me through the sometimes confusing thermodynamics, all the while also being very clear on the administrative parts of a graduation thesis, as well as helping me create structure in my research and providing valuable feedback on my work, not even mentioning his patience in proofreading my thesis multiple times.
I also extend my gratitude to Mahinder Ramdin for his help whenever I had questions and for his MatLab code, which helped immensely with starting up the project and understanding how flash calculations can be implemented.

Finally, thank you to the people who have proofread my thesis, you know who you are.

Abstract

Flash calculations are used for the dynamic simulation of vapor-liquid phase equilibria (VLE) in many chemical processes. When used to predict the composition of mixtures in VLE, the repeated applicatino of these flash calculations, particularly in dynamic cases can cause small inaccuracies to cascade. One way to prevent this, is to have strong predictions and initial guesses for what a mixture in equilibrium will look like. For this, stability tests can be employed. Stability tests require the minimisation of the tangent plane distance, which is derived from the excess Gibbs Energy. The VLE problem and stability tests have been approached from a viewpoint based in mathematical optimisation, not one based in thermodynamics. As such, this work provides a slightly different description of flash calculations and stability tests, and can serve as both an introduction to the problems, as well as an approach to the problem in a different way than is traditionally done.

In this report, the tangent plane distance function has been studied and applied to an example to gain intuition on the problem, and different optimisation methods were considered for minimising this function. A stochastic- and a deterministic method were applied to the stability problem to gain a better understanding of which type of method will work better for minimising the tangent plane distance. Both methods seem sufficient, but the scalability to higher dimensions is lacking, especially in the deterministic case.

Contents

Chapter 1

Introduction

Many practical problems in the world can be translated to mathematics. This can often be surprising to people not as ingrained in the fields of mathematics, physics or other engineering subjects. For example, one could model cars accelerating when a traffic light switches from red to green using a differential equation, or a mailman trying to deliver mail to different houses on his route in the shortest time frame using Linear Programming. Similarly, many processes and situations in the design of a chemical plant can be modelled and predicted using formulas and algorithms. Simply said, in this thesis we will consider the following:

> Consider a box which perfectly isolates, and the temperature and pressure of which we can regulate (set at specific constants). We put a liquid mixture of two (or more) components into this box, close it, and set the temperature and pressure. We would like to be able to predict what happens to the mixture[1] if we wait a while (until the mixture does not change anymore, i.e. is in *equilibrium*); will the contents of the box only be a liquid, or will a gas be formed? If so, how much gas? How much of each component is in each phase (liquid, gas)?

This is the problem we will be considering in this work, generally called the *vapor-liquid equilibrium* (VLE) problem, generally solved with flash calculations, algorithms that iteratively converge to a solution that predicts what the components in the box look like. They will be explained in more detail later in this work, in Section 2.1.

1.1 Zero Emission Fuels

The idea for this project came from, and the research was performed at, Zero Emission Fuels B.V. (ZEF). ZEF is a company located on the campus of the TU Delft, and it works together with TU Delft and several other companies and institutes to design and produce a small methanol production plant. Using the energy from two solar panels, the plant will convert (carbon dioxide and water in the) air into methanol. This methanol can then be used as fuel for vehicles or power generators, or as a part of industrial chemical processes, such as in the production of hydrocarbons [32]. ZEF aims to reduce the costs of synthetic methanol production by reducing

[1] Here, we will not consider cases where different components of the mixture react with each other.

the overall size, weight and number of parts of the plant, and by integrating different parts of a typical methanol production chain.

Because the plant is small, and its power source naturally fluctuates in output, a dynamic model is required to simulate the behavior of the (contents of the) plant. Existing simulation tools like Aspen fail to properly model the behavior, thus finding new dynamic tools and improving existing tools for the modelling of these dynamic situations are quite interesting to ZEF.

1.2 Research Goals

Flash calculations have three problems: they take a lot of time to complete, they do not accurately predict equilibria, especially in the critical region[2], and they often cause breakdowns and return nonphysical outputs when used in dynamic simulations (where they are run many times in succession)[3]. Solving all three of these problems is not within the scope of this project, and the accuracy poses the largest problems for ZEF, and thus will be the main focus here. There are two concepts that spawned this project:

1. the observation that much of the (still used and referenced) research into flash calculations (for VLEs) are relatively old: most of the state-of-the-art is from papers published until 1990, or builds increasingly complicated models that do not necessarily improve the calculations for general VLE flash calculations, but for specific cases, compounds, circumstances or ranges of pressure and temperature.

2. the hypothesis that few researchers with knowledge of mathematical optimisation, computational science or algorithm design contributed to the field of thermodynamics and vice versa. This would mean that most attempts at improving flash calculations have been done with a purely thermodynamics background and focus in mind, leaving different avenues of experimentation and research largely unused. This is an opportunity to approach the subject from a different side, in this case mathematics and optimisation.

Keeping that in mind, we define the research goals to be as follows:

1. Frame the VLE and the problems surrounding it in a mathematics focused way, as opposed to a more thermodynamics focused one.

2. Provide an overview for the prominent optimisation techniques used in relation to flash calculations.

3. Compare different optimisation techniques on their merit for the modelling of VLEs.

Where the "holy grail" for the research is a clear mathematical outline and definition of the problem, complete with an algorithm or other method to improve on the current state-of-the-art for flash calculations.

[2]The *critical region* is the range of P and T close to the *critical point*, the combination of the P and T where a substance can exist in both the liquid- and the vapor phase at the same time. See also Appendix B.

[3]By "nonphysical" we here mean a theoretical result that cannot happen in practice, such as negative or even amounts of substance in a mixture that are complex numbers.

1.3 Research Questions

Based on these goals, we will try to answer the following questions:

1. How up to date is the field of thermodynamics on the use of (mathematical) optimisation techniques, specifically in the case of flash calculations?
2. Can the stability- and accuracy problems of flash calculations be improved using state-of-the-art (NLP-)optimisation techniques?
3. Can the stability- and accuracy problems of flash calculations be improved when approached from an almost purely mathematical point of view, as opposed to a more thermodynamics-oriented one?

Questions 2 and 3 seem very close in their definition, but are defined separately as their might be a subtle but relevant difference between being able to apply optimisation techniques to the problem, and approaching the problem from a purely mathematical point of view.

1.4 Assumptions and Parameters

In this work, the mixture of substances that will mostly be used as a "direct object" is the [0.4 0.4 0.2] water-methanol-carbon dioxide. We will use this notation in square brackets to denote the composition of a mixture: in this case, the mixture consists of 40% water, 40% methanol and 20% carbon dioxide. This is because this is a less complicated version of the mixture used by ZEF, and helps us in defining some important boundaries for the scope of this project, e.g. only vapor-liquid equilibria will be considered in this work, and no reactions between the substances will take place. VLEs are still very relevant in the industry [42], and are the only equilibria expected for the ZEF case. Thus, other equilibria like liquid-liquid or solid-liquid will not be considered.

1.5 Overview

We will define and describe all the moving parts of the problem in Chapter 2.
In Chapter 3, we will go deeper into formulating the problem as minimising the tangent plane distance, and take a look at what the tangent plane distance looks like for a [0.4 0.4 0.2] H_2O-CH_4O-CO_2[4] mixture at different pressures and temperatures.
Chapter 4 will focus on the optimisation part of the problem, discussing the objective function, different methods that have been applied in literature, and choosing which methods will be tested, also going deeper into explaining each of these methods.
Chapter 5 contains the results of our tests, with some small preliminary conclusions and observations.
Finally, we finish with Chapter 6, where we draw conclusions from our findings, evaluate our research goals and -questions, give annotations to the data in the work, and give recommendations for the future.

[4] This is the chemical formula for a water-methanol-carbon dioxide mixture.

Appendix A contains a list of symbols, and Appendix B includes a list of terms to clarify some thermodynamic terms. It is recommended to keep these two appendices ready for readers that do not have a background in thermodynamics. Appendix C contains a block diagram of a flash calculation. Appendix D talks about the fugacity coefficient, specifically an expression for it for the Soave-Redlich-Kwong equation of state (see Section 2.2). Appendix E contains notes and calculations for the (MatLab) implementations of each of the methods applied in Chapter 5. In Appendix F, a mathematical description of the problem without any of the physical context has been included.

Chapter 2

Problem Description

In this chapter, we will formally introduce the problem(s) we are dealing with: we will explain what a flash calculation is, talk about equations of state and which one will be employed in this work. Finally, we will look at two existing methods that are used to improve on the modelling of VLEs.

2.1 Flash Calculations

Flash calculations are used to calculate the (thermodynamic) properties of mixtures. Consider a liquid mixture with two components C_1 and C_2 that are mixed in a tank at some pressure P and temperature T. While this liquid is in the tank, particles in liquid phase are moving to the gas phase and vice versa (liquid is "evaporating" and gas is "condensating"). After some time, the number of particles going from the gas phase to the liquid phase are equal to the number of particles going to the gas phase from the liquid phase: an *equilibrium* between the gas (vapor) phase and liquid phase is reached. In the end, we obtain two substances that are in a VLE. The known quantities now are P, T, and the two substances and their properties: critical pressure P_c, critical temperature T_c and acentric factor ω (see below).

Intermezzo: Critical and Acentric Factors

The acentric factor ω is a conceptual number used in defining properties and equations for substances, it represents the "non-sphericity" of a substance's molecules. The higher ω is, the higher the boiling point. It is different for each substance. For examples of different values of ω for different substances, see [20].
To understand the concept of critical temperature and pressure, we will look at the phase diagram of water, a visual representation of what phase water is in for what P and V or T. See Figure 2.1.

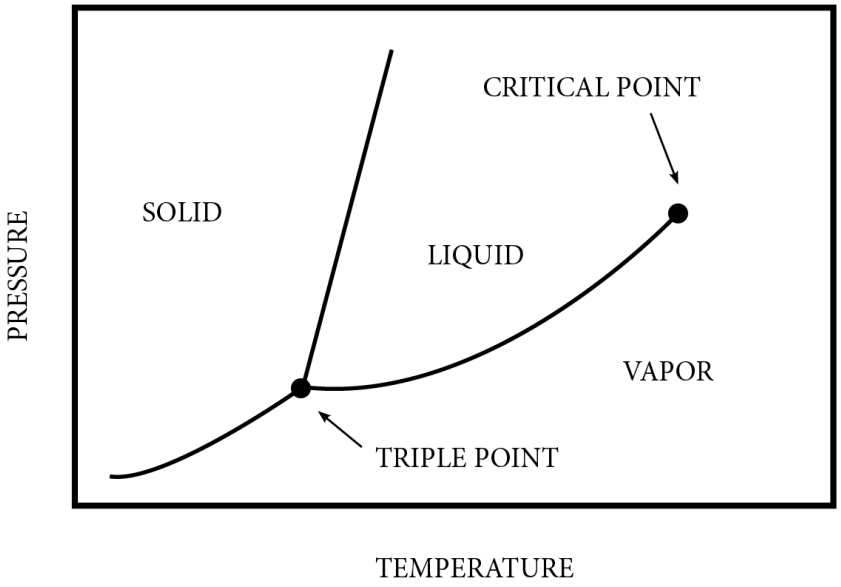


Figure 2.1: Schematic of the phase diagram of water (H_2O). At the triple point solid, liquid and vapor-phase water can coexist. At (and beyond) the critical point, liquid- and vapor-phase water can coexist.

Most people know that, at athmospheric pressure, water boils at 100 ^{o}C, or about 373 K. Looking at Figure 2.1, it becomes clear why the addendum "at athmospheric pressure" is necessary: at different pressures, the boiling point (the line separating liquid and gas) of water is at a different temperature. The critical point is the combination of the highest temperature (critical temperature T_c) and pressure (critical pressure P_c) for which steam and liquid water can coexist, like on the rest of the line separating vapor and liquid.

Flash Calculations: Continued

From P, T, $\mathbf{z}$ and substance properties such as ω and P_c, we want to compute the volume of the mixture in the tank, how much of it is in the liquid phase, how much is in the vapor phase (β, the vapor fraction) and the composition of the liquid- and vapor phases themselves[1]. To find these unknown quantities, flash calculations are used. Flash calculations are the basic building blocks of simulation software.

A flash calculation takes P, T (other inputs are possible), and the properties of the substances (more than two substances and more than two phases is possible) as inputs. Then, if there is uncertainty about the number of phases in equilibrium, or to refine initial guesses, a stability test is performed to see how many phases will be present when the mixture reaches equilibrium. Here, a global optimisation problem is solved (see Section 2.4.1). Then, an initial guess for the liquid

[1]The difference between β the vapor fraction and $\mathbf{y}$ the vapor composition is as follows: β describes how much (what fraction) of the contents of the box are a vapor, and $\mathbf{y}$ describes what this vapor then looks like. For example, $\beta = 0.4$ and $\mathbf{y}$ =[0.2 0.8] means that 40% of the contents of the box are a vapor (and 60% a liquid), and that the vapor then consists of 20% component 1 and 80% component 2.

composition $\mathbf{x}$ and vapor composition $\mathbf{y}$ is made (e.g. "I estimate that the vapor will consist for 30% of substance C_1 and 70% of substance C_2", or $y = [0.3\ 0.7]$). Both the number of phases and the initial guess can be determined in different ways, such as using existing modelling software, intuition, very little care (choosing a 50/50 split) or algorithms. Using these initial guesses we iteratively do the following: using an equation of state (see Section 2.2) the compressibility factor Z[2] for each phase is calculated, from which the fugacities[3] for each phase can be calculated from a Z-polynomial associated with the equation of state. From this, we can calculate our vapor fraction β by solving the Rachford-Rice equation [21]:

$$\sum_{i=1}^{C} \frac{z_i(K_i - 1)}{1 + \beta(K_i - 1)} = 0 \tag{2.1}$$

where C is the number of components in the mixture, z_i is the fraction of component i in the feed mixture (i.e. the mixture that was originally put into the box), and $K_i = \frac{y_i}{x_i}$ is the distribution ratio of component i. Now, we can update our estimates for the liquid- and vapor compositions $\mathbf{x}$ and $\mathbf{y}$. If these stop converging (the change in $\mathbf{x}$ and $\mathbf{y}$ from one iteration to the next is very small), we have obtained $\mathbf{x}$ and $\mathbf{y}$, and we terminate the flash calculations.
A general description of the flash procedure can be found in literature such as [21] or [58], and Appendix C includes a block diagram of the flash calculation.

2.2 Equations of State

An equation of state are one of the core parts of flash calculations. An equation of state is an algebraic relation between the pressure P, the temperature T, and the volume V of a substance [36]. This is of course quite generic, writing $P + V = T$ would technically also be an "algebraic relation between P, V and T", but not many people would recognize it as a legitimate equation of state. Equations of state are usually employed to model the behavior of a chemical substance(s) that is (are) in equilibrium in the form of $f(P, V, T) = 0$. One example is the Ideal Gas Law, one of the first equations of state:

$$PV = nRT \tag{2.2}$$

where R is the ideal gas constant ($R \approx 8.31 m^3 \cdot Pa \cdot K^{-1} \cdot \text{mol}$), and n the number of moles of gas. There is, as of yet, no one surefire way to theoretically define an equation of state for all species and mixtures, so many equations of state are based on empirical findings. Their accuracy is often judged based on their results compared to experimental data [39]. As such, most equations of state work quite well for a specific range of T, V and/or P and specific substances, but fail to give accurate results outside of these parameters (without additional empirical corrections).
The purpose of equations of state is essentially threefold [50]:

1. Representation of pressure-volume-temperature (PVT) data for smoothing and interpolation, and PVT data integration for calculating derived properties such as the fugacity.

2. Prediction of gas phase properties of pure (i.e. unmixed) substances and their mixtures from only a small amount of experimental data.

[2]The compressibility factor is a factor which describes how strongly some liquid or gas deviates from an ideal gas, $Z = \frac{PV}{RT}$, where P the pressure, V the volume, T the temperature and R the ideal gas constant

[3]Fugacity is an "effective pressure", the pressure of an ideal gas with the same T and P as the non-ideal substance. See also Appendix B.

3. Prediction of vapour-liquid equilibria of mixtures, especially at high pressures.

Though understanding equations of state is contextually important, these are not the qualities of equations of state that we will focus on. Instead, we focus on equations of state as a part of the algorithms. Because the focus of this work is not equations of state, it is convenient to choose a specific one as a base for our case, and for the definition of the benchmarks. First, we will introduce the equations of state that are considered. Then we will define criteria on which we base the choice of equation of state we will be working with on, and finally choose an equation of state.

Below is a qualitative description of the equations of state that were considered for this work. While this is not an exhaustive list of all possible equations of state, we consider these specific few based on recommendations by experts and literature research. Most of the equations of state mentioned below are also useful because from them, other more accurate ones were developed (see also Section 2.3).

2.2.1 Van der Waals Equation

One of the first equations of state to perform better than the Ideal Gas Law, this equation of state is formulated as follows:

$$(P + \frac{a}{V^2})(V - b) = RT \tag{2.3}$$

$$a = 3P_cV_c^2 \tag{2.4}$$

$$b = \frac{V_c}{3} \tag{2.5}$$

where a is the attraction parameter representing the attraction between the molecules, b is a repulsion parameter representing the size of the molecules, T_c, P_c are the temperature and pressure at the substance's critical point, and P, V, T and R are as defined above. This equation was based on the Ideal Gas Law and was modified to account for the fact that real gasses do not behave ideally [53]. This equation no longer sees much use, because its accuracy is now trumped by other equations of state, such as the ones mentioned below[4]. It is discussed here mostly for its historic significance and context.

2.2.2 Redlich-Kwong (RK)

The Redlich-Kwong equation of state is defined as follows:

$$P = \frac{RT}{V - b} - \frac{a}{\sqrt{T}V(V + b)} \tag{2.6}$$

$$a \approx 0.427\frac{R^2T_c^{\frac{5}{2}}}{P_c} \tag{2.7}$$

$$b \approx 0.087\frac{RT_c}{P_c} \tag{2.8}$$

[4]It is still used to find bounds on the values sometimes as well.

This method was constructed to conform to the corresponding states principle (in such a way that $b = 0.26V_c$), which relates the compressibility factors of different pure substances [39]. While this equation of state performed well for the time that it was developed, these days it is less relevant due to the success of other methods, such as the two mentioned below. This equation of state is mainly discussed because it was a basis for the next two equations of state mentioned in this chapter, which are both more accurate and more used in practice.

2.2.3 Soave-Redlich-Kwong (SRK)

A modification and improvement of Redlich Kwong, defined as follows:

$$P = \frac{RT}{V-b} - \frac{a\alpha}{V(V+b)} \tag{2.9}$$

$$a \approx 0.427\frac{R^2T_c^2}{P_c} \tag{2.10}$$

$$b \approx 0.087\frac{RT_c}{P_c} \tag{2.11}$$

$$\alpha = (1 + (0.485 + 1.551\omega - 0.156\omega^2)(1 - T_r^{0.5}))^2 \tag{2.12}$$

$$T_r = \frac{T}{T_c} \tag{2.13}$$

where ω is the acentric factor. This equation of state assumes temperature-dependency for the attraction parameter a (instead of letting it be a constant, as in Redlich-Kwong). Mixing rules can be used to extend this equation for multicomponent vapor-liquid-equilibrium calculations. This modification of Redlich-Kwong by Soave proved to be a significant improvement over the original Redlich-Kwong, and is widely used in the industry to this day because of its relative simplicity and accuracy [42]. It was designed to accurately reproduce the vapour pressures of pure polar substances [44][5]. SRK has been modified in many ways, such as the volume translation by Peneloux [34] and predictive-SRK [15].

2.2.4 Peng-Robinson

Unsatisfied with the performance of SRK, Peng and Robinson set out to formulate a (situationally) better equation of state.

$$P = \frac{RT}{V-b} - \frac{a\alpha}{V^2 + 2bV - b^2} \tag{2.14}$$

$$a \approx 0.457\frac{R^2T_c^2}{P_c} \tag{2.15}$$

$$b \approx 0.078\frac{RT_c}{P_c} \tag{2.16}$$

$$\alpha = (1 + \kappa(1 - T_r^{\frac{1}{2}}))^2 \tag{2.17}$$

$$\kappa \approx 0.375 + 1.54\omega - 0.270\omega^2 \tag{2.18}$$

$$T_r = \frac{T}{T_c} \tag{2.19}$$

[5] "Polar" is a description of the separation of the electric charge of the molecules, with a negatively- and a positively charged part.

It was designed to accurately reproduce the vapour pressures of pure polar substances [35]. It did not outperfrom SRK in all cases (SRK works better for polar mixtures, and Peng-Robinson works better for non-polar ones), and nowadays Peng-Robinson and SRK are applied in different areas of chemical engineering [51].

2.2.5 Statistical Associating Fluid Theory (SAFT)

The formulation for the SAFT model is based on the molecules and the interactions between them. It uses an association term for multi-component mixtures. It models molecules as spheres with associating mixtures as chains [8]. SAFT is a nonanalytical equation of state consisting of a few different moving parts that more accurately represent the real behaviour of chemical species in a compound, such as perturbation models. The complete formulation is quite complex and extensive, and thus will not be fully included here. Its formulation can be found in literature [7, 8].

2.3 Model Selection

We will now evaluate the equations of state above on the following criteria:

1 The equation of state should be useful (and used) for modelling fluid properties.

2 The equation can be applied to the ZEF test case.

3 The model needs to allow for the application of mathematical (optimisation) techniques when implemented.

4 The equation of state should not be rooted too deeply in thermodynamics, to prevent problems with scope.

We decided that SRK and Peng-Robinson were the two that best met this Criterium **1**. While others are still relevant (SAFT was recommended because of its literary significance and accuracy), none of them can boast the popularity of SRK and Peng-Robinson. Criterium **2** was fairly easy to evaluate. If the algorithmics could be improved, relevance for ZEF would be all but guaranteed. Also, ZEF mainly uses SRK and Peng-Robinson for their VLE simulations. Criterium **3** is implied for every equation of state, as the equations can get quite complicated for real-life mixtures, such as in the ZEF case. Thus, they cannot easily be solved analytically, and employ optimisation and numerical methods to be solved. Finally, for Criterium **4**, Van der Waals, (S)RK and Peng-Robinson are, because of their analytical nature, quite close in their definition and conception to mathematics. This makes it so that most papers related to these equations of state are readily readable for people coming from mathematics. The implementation of the methods is also not too difficult thanks to their simple algebraic formulation. SAFT is more strongly rooted in the fields of chemistry and statistical mechanics, and was due to its nonanalytical nature less explicitly defined. Thus, it would be a lot harder for a layman in thermodynamics to work with SAFT than it would be to work with (S)RK and Peng-Robinson.

Taking all of the considerations above into account, the decision not to use SAFT could be made fairly quickly. The Ideal Gas Law, van der Waals equation, and Redlich-Kwong are disregarded

because of lack of literary significance (Criterium **1**). It should be noted that for the purposes of this work, we speculate there will not be a big difference between SRK and Peng-Robinson. Both are cubic equations of state with fairly comparable applications and very similar algebraic formulations. Thus, we hypothesise that any mathematical solution for improving the stability, accuracy and/or solving speed of (algorithms employing) one would also be applicable to (algorithms employing) the other with minimal modification This hypothesis is also supported by literature [22, 29]. Considering their similarities, the choice falls to SRK, as it works better for polar mixtures such as water-methanol. Further comparison between SRK and Peng-Robinson can be found in [51].

2.4 State-of-the-Art

Now that we have generally defined our model, we take a look at how the problems for this model are resolved in literature. Commonly one of two methods is used to improve on the performance of flash calculations: stability tests and initial guess methods.

2.4.1 Stability Tests

In Section 2.1 it was mentioned that before we perform a flash calculation, we need to know into how many phases the mixture will split. If the (liquid) feed mixture is stable for example, meaning that the phase of the mixture in equilibrium is the same as the input (feed) phase, no $\mathbf{x}$, $\mathbf{y}$ or β have to be calculated, as we already know the overall composition $\mathbf{z}$, and then trivially $\beta = 0$, $\mathbf{x} = \mathbf{z}$ and $\mathbf{y}$ is irrelevant .

Tangent Plane Analysis

Almost all stability tests are based on (Michelsen's) tangent plane analysis [25]. Tangent plane analysis uses Gibbs's Tangent Plane Criterion [1] to determine whether Gibbs energy is at a global minimum, which would mean that the mixture is in equilibrium. We give a derivation of the tangent plane condition below for sake of completeness. Consider a C-component mixture in a perfectly isolated box as introduced in Chapter 1, with a pressure P and temperature T. Assume P and T are chosen in such a way that the mixture eventually reaches a VLE: there are both a vapor- and a liquid phase present inside the box and the number of particles going from vapor- to liquid phase at any moment is equal to the number of particles going from liquid- to vapor phase. For the C-component mixture of feed composition $\mathbf{z}$, a necessary condition for the mixture to be in equilibrium is for the chemical potentials for each component to be the same for every phase:

$$\mu_i^{(L)} = \mu_i^{(V)} \tag{2.20}$$

or equivalently, that the fugacities of individual components are equal:

$$f_i^{(L)} = f_i^{(V)} \tag{2.21}$$

We can also define an equation in terms of the molar fractions [6]:

$$\beta y_i + (1 - \beta) x_i = z_i \tag{2.22}$$

[6] The molar fraction is the concentration of each component in each phase

and the mole fractions must sum to unity, which also gives us:

$$\sum_{i=1}^{C} x_i = \sum_{i=1}^{C} y_i = 1 \tag{2.23}$$

$$\Rightarrow \sum_{i=1}^{C} (y_i - x_i) = 0 \tag{2.24}$$

From this, we have obtained a system of $2C + 1$ equations ((2.21), (2.22), (2.24)) and $2C + 2$ variables ($\mathbf{x}$, $\mathbf{y}$, T, P)). While dealing with equations of state and phase equilibrium calculations, we define a trivial solution to be a solution where the liquid composition ($\mathbf{x}$) and vapor composition ($\mathbf{y}$) are identical (such that the equations (2.21), (2.22), (2.24) are satisfied automatically). Sometimes the trivial solution could be the only valid one (for example for a (P, T) flash in the single-phase region), but iterative solution processes often get stuck in a trivial solution even when it is not the optimum. Many thermodynamic (flash-)calculations encounter (convergence-)problems when solved in the critical region, i.e. near the critical point. This is because at the critical point, we have $x_i = y_i = z_i$, thus in the critical region around this point we are close to the trivial solution, i.e. $x_i \approx y_i \approx z_i$. In these cases, calculations often converge to the trivial solution unless initial guesses are sufficiently close to the desired solution (the correct combination of $\mathbf{x}$, $\mathbf{y}$ and β in equilibrium) [28].

The conditions mentioned above are only necessary, and not sufficient. A necessary and sufficient condition is for the total Gibbs Energy[7], to be at a global minimum (for specified (T, P), other energies can be minimised for different flashes, see also [25]). The reduced global Gibbs Energy is equal to:

$$\begin{aligned} \frac{g^E(T,P,\mathbf{z})}{RT} =& \frac{P}{RT}\left(v - \sum_i z_i v_i\right) - \left(\ln(v-b) - \sum_i z_i \ln(v_i - b_i)\right) \\ & - \frac{1}{\delta_2 - \delta_1}\left(\alpha \ln \frac{v + \delta_2 b}{v + \delta_1 b} - \sum_i z_i \alpha_i \left(\frac{v_i + \delta_2 b_i}{v_i + \delta_1 b_i}\right)\right) \end{aligned} \tag{2.25}$$

where v_i is the molar volume of component i, $v = \frac{\sum_{i=1}^{N} x_i M_i}{\rho_{mixture}}$ with M_i the molar mass of component i, and

$$\frac{P}{RT} = \frac{1}{v-b} - \alpha \frac{b}{(v + \delta_1 b)(v + \delta_2 b)} \tag{2.26}$$

is determined by the cubic equation of state (if we take $\delta_1 = 0$ and $\delta_2 = 1$, and set $b = \frac{0.08664 RT_c}{P_c}$ we get the SRK equation of state [28]). It can also be written as:

$$\frac{g^E}{RT} = \ln f_{mix}(T, P, \mathbf{z}) - \sum_i z_i \ln f_i(T, P) \tag{2.27}$$

where

$$\ln\left(\frac{f_{mix} b}{RT}\right) = \frac{Pv}{RT} - 1 - \ln\left(\frac{v-b}{b}\right) - \frac{\alpha}{\delta_2 - \delta_1} \ln\left(\frac{v + \delta_2 b}{v + \delta_1 b}\right) \tag{2.28}$$

$$\ln\left(\frac{f_i b_i}{RT}\right) = \frac{P v_i}{RT} - 1 - \ln\left(\frac{v_i - b_i}{b_i}\right) - \frac{\alpha_i}{\delta_2 - \delta_1} \ln\left(\frac{v_i + \delta_2 b_i}{v_i + \delta_1 b_i}\right) \tag{2.29}$$

[7] A kind of potential energy, the amount of reversible work that may be performed at constant P and T, see also Appendix B.

are the fugacities of the mixture and component i, respectively. The change in Gibbs Energy $\Delta G = (\mu_i^{(v)} - \mu_i^{(L)})\Delta n_i$ (which corresponds to transferring Δn_i moles of component i from liquid to vapor phase) must be zero at the global minimum for the Gibbs Energy, which would imply equation (2.20) (and equation (2.21)). From this, we find that the change in Gibbs Energy associated with a new phase with composition $\mathbf{w}$ being formed is equal to

$$\Delta G = \Delta\varepsilon \sum_{i=1}^{C} w_i(\mu_i(\mathbf{w}) - \mu_i(\mathbf{z})) \tag{2.30}$$

where $\Delta\varepsilon$ is equal to the (infinitesimal) amount of the phase formed, from which we can define the Gibbs tangent plane condition:

$$\begin{gathered} \text{The mixture is stable iff:} \\ \sum_{i=1}^{C} w_i(\mu_i(\mathbf{w}) - \mu_i(\mathbf{z})) \geq 0 \\ \text{for all } \mathbf{w} \end{gathered} \tag{2.31}$$

From these calculations, Michelsen [28] ends up with the following formulation for the tangent plane distance (tpd), the distance from the tangent line at a point to the Gibbs Energy curve at $\mathbf{w}$:

$$TPD(\mathbf{w}) = g(\mathbf{w}) - t(\mathbf{w}) = \sum_i w_i(\mu_i(\mathbf{w}) - \mu_i(\mathbf{z})) \tag{2.32}$$

If TPD at a composition $\mathbf{w}$ is negative for some feed composition $\mathbf{z}$, we can choose the $\mathbf{y}$-phase composition $\mathbf{y} = \mathbf{w}$ and find an $\mathbf{x}-$phase composition $\mathbf{x} < \mathbf{z}$ such that the phase split yields a decrease in the Gibbs Energy. Equation (2.32) can be rewritten using the physical relation between ϕ_i and μ_i into the reduced tangent plane distance

$$\begin{aligned} tpd(\mathbf{w}) = \frac{TPD(\mathbf{w})}{RT} &= \sum_i w_i(\ln w_i + \ln\phi_i(\mathbf{w}) - d_i) \\ &\text{where } d_i = \ln z_i + \ln\phi_i(\mathbf{z}) \end{aligned} \tag{2.33}$$

where $\phi_i(\mathbf{w})$ is the fugacity coefficient of component i at composition $\mathbf{w}$. $TPD \geq 0$ is a necessary and sufficient condition for phase equilibrium stability, and for our (P,T) flash even specifies that the solution is unique. However, verifying that the condition is satisfied requires a search over the entire composition space, requiring extensive computations [25]. Using the tangent plane condition not only answers the yes/no question of phase stability, but also helps in finding a good starting point for subsequent calculation of the correct phase distribution from the trial phase composition $\mathbf{w}$ that results in a negative TPD. Using the TPD instead of the Gibbs Energy has as an advantage that it is in general easier to work with fugacity coefficients than chemical potentials, and that minimising the TPD is more reliable and robust than minimising the Gibbs Energy [16].

The problem we are now dealing with is this: we want to find a global minimum of $tpd(w)$, and

see if it is negative[8]. We can formulate this as a minimisation problem:

$$\min\ tpd(\mathbf{w}) = \sum_i w_i(\ln w_i + \ln \phi_i(\mathbf{w}) - \ln z_i - \ln \phi_i(\mathbf{z})) \tag{2.34}$$

$$\text{subject to } w_i > 0,\ i = 1,2,\ldots,\text{k}, \tag{2.35}$$

$$w_i = 0,\ i = \text{k+1},\ldots,\text{C}, \tag{2.36}$$

$$\sum_i^k w_i - 1 = 0 \tag{2.37}$$

where (2.35) represents the components that are (allowed) in the phase model, and (2.36) represent those that are not. This is a nonlinear, nonconvex problem [13]. Usually $k = C$, e.g. all components can be present in all phases, except in the case of some substances, like gas hydrates [28]. For the cases we are dealing with, we can assume that $k = C$, which allows us to rewrite the minimisation problem:

$$\min\ tpd(\mathbf{w}) = \sum_i w_i(\ln w_i + \ln \phi_i(\mathbf{w}) - \ln z_i - \ln \phi_i(\mathbf{z})) \tag{2.38}$$

$$\text{subject to } w_i > 0,\ i = 1,2,\ldots,\text{C}, \tag{2.39}$$

$$\left(\sum_i w_i\right) - 1 = 0 \tag{2.40}$$

Thus, tangent plane analysis tells us whether the mix has a single phase (is stable) or has two phases (is unstable), and gives us a composition $\mathbf{w}$ for the new phase being formed, which can help us in determining initial guesses for the flash.

2.4.2 Initial Guesses

As mentioned before, when in the critical region, solutions have a tendency to converge to a trivial- or nonphysical solution. To combat this, methods and algorithms have been developed for the generation of more accurate initial guesses, which allow for a better chance of converging to a solution that better represents the reality. The Wilson Equation is one of the more popular ways of determining initial guesses.

The Wilson Equation

Initial guesses for x_i and y_i are often estimated using the Wilson Equation [55] [33], via an expression for $K_i = \frac{\mathbf{y_i}}{\mathbf{x_i}}$:

$$K_i = \frac{x_i}{y_i} = \frac{P_{ci}}{P} \exp\left\{5.37(1+\omega_i)(1-\frac{T_{ci}}{T}\right\} \tag{2.41}$$

[8] Michelsen [28] recommends finding all minima of $tpd(w)$ and then checking if any of them are negative

From this, we can obtain the liquid and vapor decompositions $(\mathbf{x}, \mathbf{y})$ via the general rules:

$$\beta \Rightarrow \sum_{i=1}^{C} \frac{z_i(K_i - 1)}{1 + \beta(K_i - 1)} = 0 \qquad \text{(Rachford-Rice)} \tag{2.42}$$

$$x_i = \frac{K_i z_i}{1 + \beta(K_i - 1)} \tag{2.43}$$

$$y_i = \frac{z_i}{1 + \beta(K_i - 1)} \tag{2.44}$$

the derivations of which can be found in [33].

2.5 The Reduced Tangent Plane Distance Function

Consider the fugacity coefficient $\phi_i(\mathbf{w})$. This is a function calculated using the equation of state, e.g. for SRK we have:

$$\ln \phi_i(\mathbf{w}) = \frac{b_i}{b^{(m)}}(Z - 1) - \log(Z - B) - \frac{A}{B}\left(2\frac{\sum_j w_j a_{ij}}{a^{(m)}} - \frac{b_i}{b^{(m)}} \log \frac{1 + B}{Z}\right) \tag{2.45}$$

where $A, B, Z, a^{(m)}, b^{(m)}$ are all dependent on $\mathbf{w}$. Expressions for these $\mathbf{w}$-dependent terms for the SRK equation of state can be found in Appendix D. Calculating Z involves finding the root of a polynomial of degree at most 3. ϕ is a nonlinear, nonconvex function.

Problem (2.38) can be transformed into an unconstrained optimisation problem using new decision variables $\gamma_i \in (0, 1)$ [57]:

$$n_{i\mathbf{w}} = \gamma_i z_i n_{\mathbf{z}},\ i = 1, \ldots, C \tag{2.46}$$

$$w_i = \frac{n_{i\mathbf{w}}}{\sum_{j=1}^{c} n_{j\mathbf{w}}},\ i = 1, \ldots, C \tag{2.47}$$

$$n_F = \sum_{i=1}^{C} n_{i\mathbf{z}}, \text{ the total number of moles in the (feed) mixture,} \tag{2.48}$$

giving us the following optimisation problem:

$$\min\ tpd(\gamma) \tag{2.49}$$

$$0 \leq \gamma_i \leq 1,\ i = 1, \ldots, C \tag{2.50}$$

$$\tag{2.51}$$

Having the problem be an unconstrained one can be useful for the purposes of some solution methods.

Another reformulation for the objective function is:

$$TPDF(\mathbf{w}) = \sum_{i=1}^{C} (k_{i+1}(\mathbf{w}) - k_i(\mathbf{w})) \tag{2.52}$$

where $k_i(\mathbf{w}) = \ln(\phi_i(\mathbf{w})) + \ln(w_i) - \ln(z_i) - \ln(\phi_i(\mathbf{z}))$, and we choose $k_{C+1} = k_1$. The summand for component i is the reduced tangent plane distance $tpd(\mathbf{y})$. The advantage of this formulation

is that the minima lie at $TPDF = 0$, and knowing the value of minima a priori can be advantageous. The zeros of $TPDF$ correspond with points on the Gibbs Energy surface where the tangent plane at $\mathbf{w}$ is parallel to the one at $\mathbf{z}$, $k_i(\mathbf{w})$. Thus, when we find $\mathbf{w}^*$ such that $TPDF$ is minimised, we have found the minimum *tpd*: $tpd(\mathbf{w}^*)$ [46, 47, 48].
Another reformulation, turning the function into a dimensionless one [30]:

$$\bar{D}(\mathbf{x}) = \frac{D(\mathbf{x})}{RT} = \sum x_i(\ln \phi_i(\mathbf{x}) - \ln \phi_i(\mathbf{z})) \tag{2.53}$$

In general, the "standard" definition for the *tpd* suffices for the practical applications discussed in this work, but the reformulations were discussed for completeness' sake. Other ways of using the stability problems in phase equilibrium and their difficulties can be found in [27].

Chapter 3

The Tangent Plane Distance

Flash calculations are usually not run in isolation, especially in the ZEF case. In dynamic simulations, flash calculations are but one part, and they are run multiple times. Output from one iteration is often fed back into the next iteration. Because of this, small errors or inconsistencies returned by a flash calculation during the simulation often "explode" into larger errors, physically impossible outputs or cause the algorithm to crash. This means that technically, we are working to improve on issues with stability of the flash calculations. However, because we are working with single isolated flash calculations in this research, we will focus on improving the accuracy of the flash algorithms, which then would improve the general stability of the dynamic simulations. In this chapter, we will look deeper into the tangent plane distance function and its properties. This analysis will be aided by looking at tangent plane distance plots for a [0.4 0.4 0.2] H_2O-CH_4O-CO_2 [1] mixture at different pressures and temperatures.

3.1 Stability Tests

In Section 2.4, we discussed initial guesses and stability tests as ways to improve VLE simulations. The main focus of this thesis will be on the minimisation of the stability tests, for the following reasons:

- Stability tests have a more rigorous theoretical proof of why they work, which lends itself better to applying mathematical optimisation techniques to it.
- They more explicitly contain optimisation problems, which allows for a more direct exploration of the hypotheses mentioned in Chapter 1.2.
- Solutions to stability tests can give a trial phase composition for the new phase being formed, which naturally leads to better initial guesses for the flash calculations.
- Stability tests are generally applicable to different flash calculations (with different equations of state), and also useful outside of improving flash calculations and dynamic simulations (just for the purpose of predicting whether a mixture under some P and T is stable).

[1] water-methanol-carbon dioxide.

We aim to improve the problems in the critical region, which are also illustrated in Figure 3.1

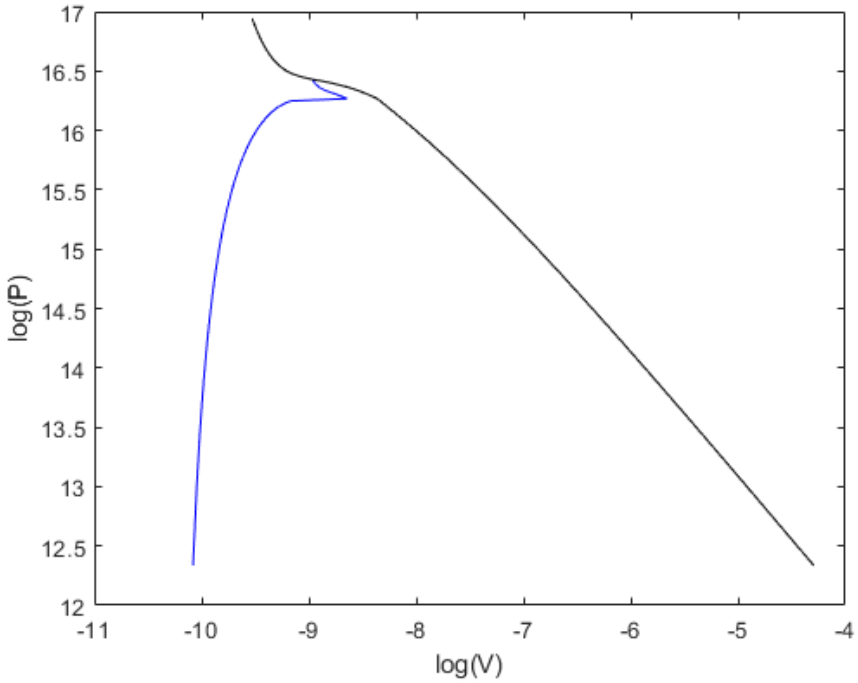


Figure 3.1: Plot of a two-phase region using the Wilson equation and the SRK equation of state, formed by the bubble point curve (blue) and dew point curve (black). They come together at the critical point, where the liquid- and vapor phase can coexist.

Here, the Wilson equation in combination with the SRK equation of state were used to plot something called the two-phase region: for a range of pressures, the bubble- and dew point temperatures of [0.4 0.4 0.2] H_2O-CH_4O-CO_2 mixtures were calculated. These were converted to volumes via $V = \frac{ZRT}{P}$ and log plotted, to make the results easier to visually identify. In theory, Figure 3.1 should look like a "neat" parabola, but problems arise near the critical point (at the top of the would-be parabola). Near the critical point (where the dew- and bubble curves coincide), estimates for the bubble- and dew point volume (for a given pressure) are often off. This is generally believed to be caused due to having solutions to the equilibrium equations be very close to a trivial solution, and thus tending to converge to an incorrect solution [28]. SRK and Wilson are however stable in the sense that, given a P, a V will be almost always be provided, even if no bubble- or dew- point exists (in the so-called supercritical region for example, above the critical point, $\log(P) > 16.25$ in Figure 3.1). These two factors cause the sudden dropoff and peculiar angle for the bubble point, the dew point not reaching a stationary point correctly and both the bubble- and dew point curve continuing into the supercritical region (where they coincide because the mixture is no longer 2-phase).

3.2 TPD plots

To check the performance of tangent plane analysis and how to handle calculating the *tpd*, we use tangent plane analysis to judge whether the feed was stable or unstable for a liquid- and a

vapor feed. This was done for T=330,...,380 at $P = 0.96$ bar, This was then compared to data from the Dortmünd Data Bank [10] to evaluate the ability of tangent plane analysis to correctly predict the stability of a mixture, see Figure 3.2.

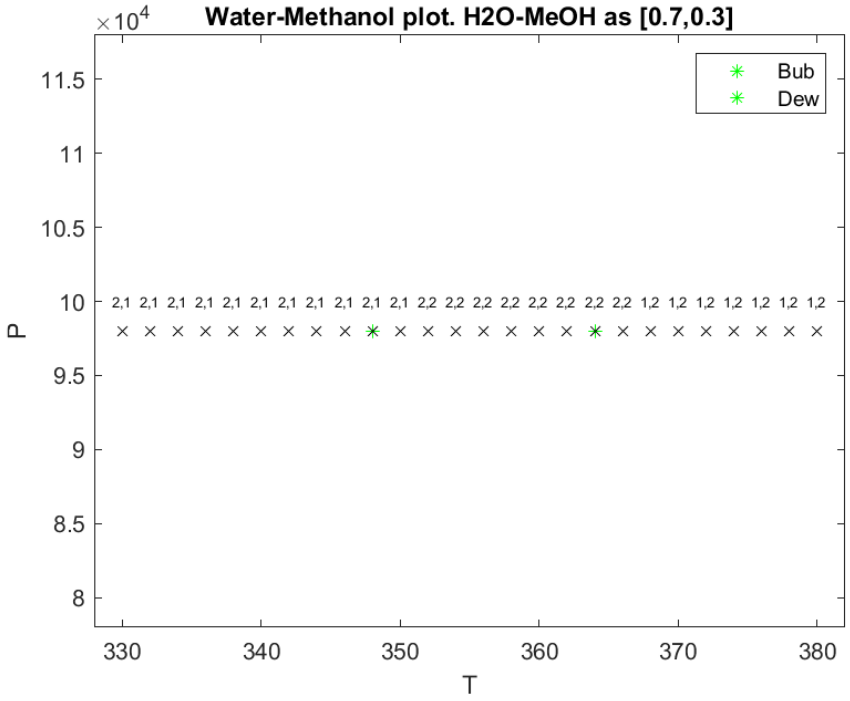


*Figure 3.2: Results of tangent plane analysis, compared with data from the Dortmünd Data Bank (DDB). Green * represent the bubble- and dew point of the mxiture according to the DDB. Black × represent a tested data point. Numbers above a × represent whether tangent plane analysis judged a point to be stable (1) or unstable (2) for a vapor- or liquid feed, respectively.*

In Figure 3.2, × represent a point on which we performed tangent plane analysis. This was done for both a vapor- and a liquid feed. The results are represented by the numbers above the ×. A 1 means that the minimum tangent plane distance was nonnegative (i.e., tangent plane analysis predicts the mixture is stable, i.e. single-phase). A 2 means that the minimum tangent plane distance was negative (i.e., tangent plane analysis predicts the mixture is unstable, i.e. two-phase in this case). The first number of the two represents the result when assuming a vapor feed, and the second number for a liquid feed.
For example, "2,1" above one of the × means that the mixture was judged to be unstable for a vapor feed, and stable for a liquid feed. The green ∗ represent the bubble- and dew point, respectively, for the mixture at this P according to the data from the Dortmünd Data Bank. This plot was made for $\mathbf{z} = [0.1\ 0.9], [0.2\ 0.8], \ldots\ [0.9\ 0.1]$. In general tangent plane analysis performs very close to reality for this P, with tangent plane analysis having a similarly sized two-phase region, shifted up by about $2K$. This is assuming that we "intelligently" choose the phase for our input mixture, i.e. if our (P, T) combination lies to the left of the bubble line we input a liquid feed and we input a vapor feed when we are to the right of the dew line (see also Figures 2.1 and 3.1). We can make this assumption because if we chose the feed phase "unintelligently", the resulting phases and their components in equilibrium will be the same as for the intelligently chosen feed in reality. The problem could also be solved by always performing tangent plane analysis assuming both a liquid- and a vapor feed, and then choosing the most appropriate one.

This will inform how we use tangent plane analysis later on: we will calculate the minimum *tpd* for both a liquid- and a vapor feed, and if any one of them is stable, we judge the mixture to be stable (as we assume we chose our feed phase logically)[2].

To further gain intuition about the Tangent Plane Distance, of which we aim to find the global minimum later, tangent plane distance plots were made for different values of P and T for a [0.4 0.4 0.2] H_2O-CH_4O-CO_2 mixture. These values were chosen from the liquid, two-phase and vapor regions of the mixture, which were estimated using the Wilson equation (see Figure 3.3).

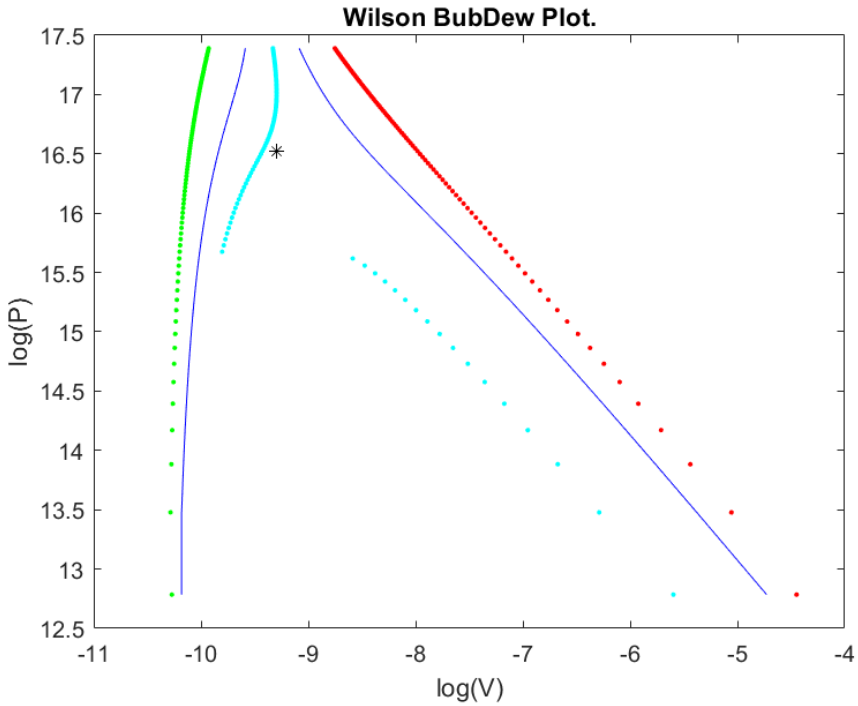


*Figure 3.3: Plot of the P, V for which the tpd surface was plot, with the bubble- and dew point curves. The P, T data points (Figure 3.4) was converted to log P, log V for easier visual identification. The critical point, *, was (visually) estimated to be around* $P \pm 1.5 \cdot 10^7 Pa$, $T \pm 523K$.

Then, for every other point along each of these three lines, the tangent plane distance for each possible trial phase composition $\mathbf{w}$ was calculated, and this was visualised using a ternary diagram[3], resulting in a picture such as Figure 3.5.

[2]This problem of not knowing the number and kinds of phases a priori is solved in literature in two ways: i) by adjusting the number of phases and conditions simultaneously and ii) by calculating the minimum *tpd* for each possible resulting combination of phases. Here, we have chosen option ii) but we will not go deeper into this specific problem. See also [49].

[3]Plotted using the formulae $x = -0.5w_1 + 0.5w_2$ and $y = w_3$.

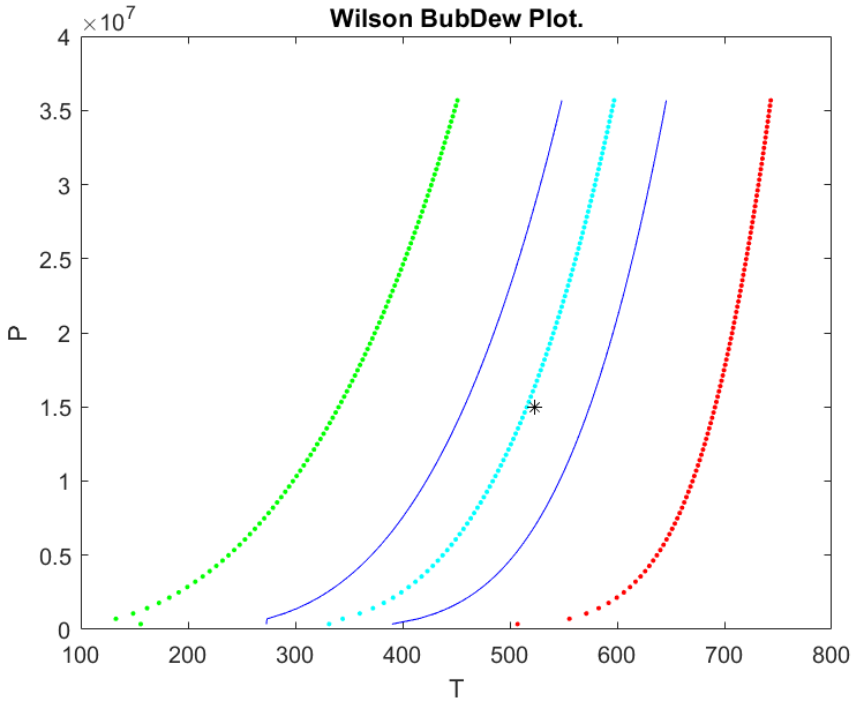


Figure 3.4: The P, T plot of the data (see also Figure 3.3).

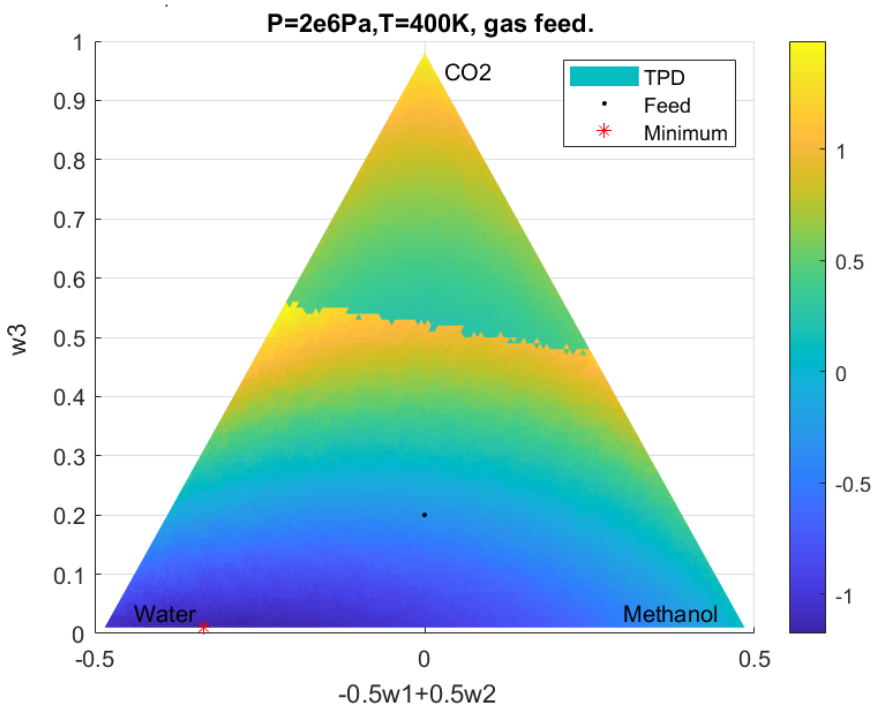


Figure 3.5: A 3D plot of the tpd, with the value of tpd represented by color. A gas feed means that the new phase that is formed is assumed to be liquid (and vice cersa, for some of the figures below).

Immediately visible from this first plot is a "line" (of a sudden shift in height) in the middle of the triangle, showing a discontinuity, see also Figure 3.6.

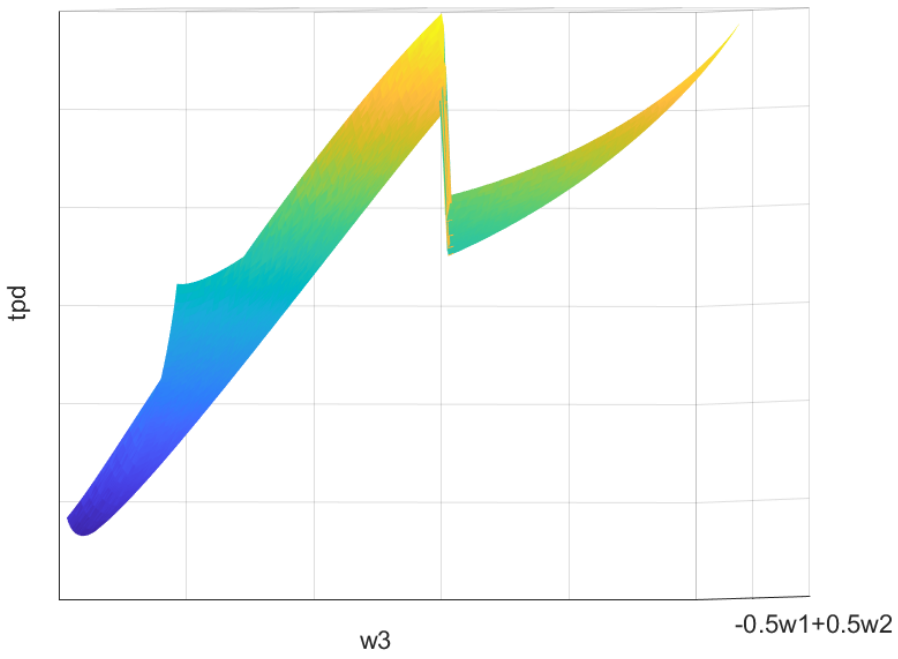


Figure 3.6: Side view to have a better look at the discontinuity, which can be seen from top view in Figure 3.5.

Keeping in mind that we find these tangent plane distances assuming both a liquid- and a vapor feed, and then if either one is stable, we judge the mixture to be stable. In the case that both feed phases result in a stable mixture or both result in an unstable mixture, we choose the "least stable" one, i.e. the feed phase for which the minimum tangent plane distance is the lowest. When both minima are equal, the one belonging to a vapor feed is chosen. This last decision was made fairly arbitrarily, as the chosen phase does not matter much when both minima are equal in size and position.

3.2.1 Liquid Region/Left (Green)

28 out of the 50 samples were judged by tangent plane analysis to be single-phase (stable). All of these samples had, in general, the same properties. The global *tpd* minimum was at the feed composition $\mathbf{z}$, with peaks at pure methanol. These samples that were judged to be single phase were all found at the highest pressures (the last 28 cases checked, $P \approx 1.6 \cdot 10^7 \ldots 3.6 \cdot 10^7$).

The other 22 samples were judged to be two-phase (unstable), with the minimum being located at 100% CO_2 (with at most a 0.1 concentration of methanol at higher pressures in this range $P \approx 0.07 \cdot 10^7 \ldots 1.5 \cdot 10^7$). Using Figures 3.7 and 3.8 we can see that these cases that were wrongly judged to be unstable are at subcritical pressures.
Remarkably, all these *tpd* plane plots looked the same regardless of the assumed feed phase.

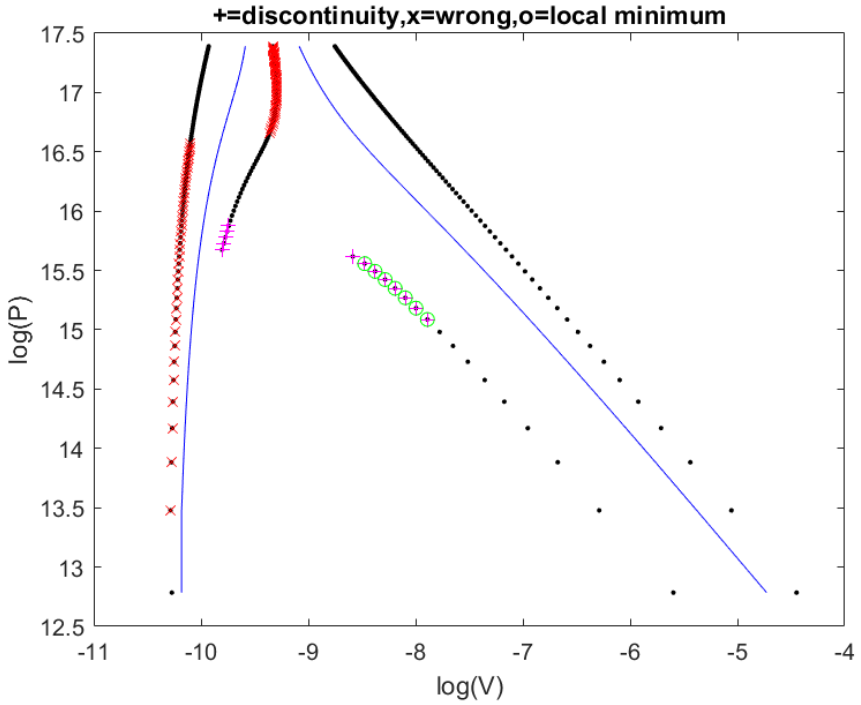


Figure 3.7: log(P), log(V) plot. In this picture, a magenta + is used to signify cases that have a discontinuity in the tpd (e.g. Figure 3.9). Red × denote a point that was misjudged, e.g. when a point left of the bubble point curve was decided by the algorithm to be unstable. Finally, green o are for cases that have at least one local minimum.

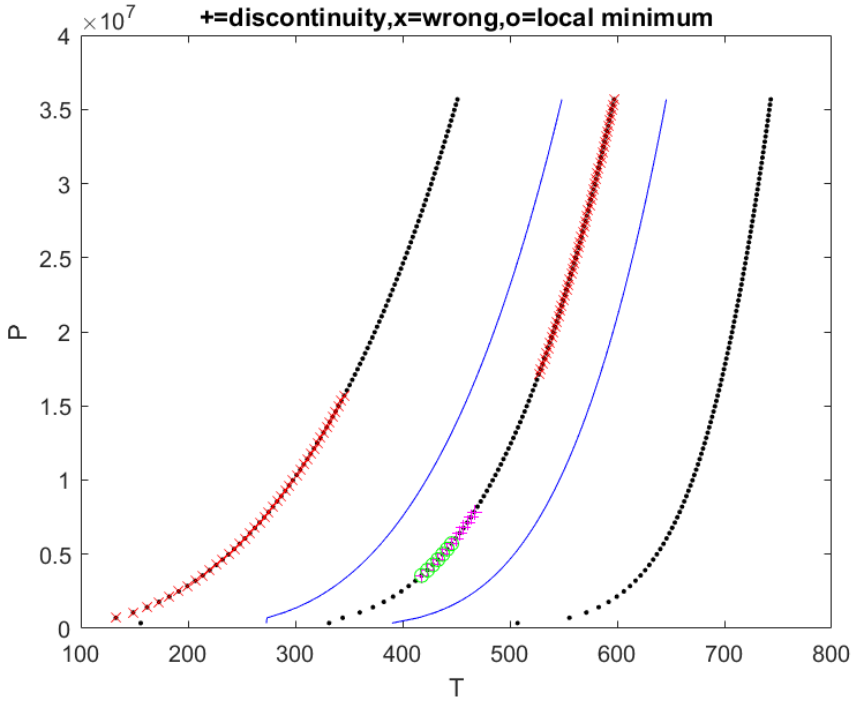


Figure 3.8: P, T plot of Figure 3.7.

3.2.2 2-phase Region/Mid (Yellow)

In the 2-phase region, 26 of the samples were judged to be stable (single phase). All of these had their global minimum at $\mathbf{w} = \mathbf{z}$, and assumed a vapor feed, and peaks for trial phases consisting of a single component. These occurred for the highest pressures (last 26 samples, $P \approx 1.7 \cdot 10^7 \ldots 3.6 \cdot 10^7$), in the supercritical region. Thus, these points should indeed be stable, even though they lie in the middle region.

The other 24 unstable (2-phase) samples exhibited differing behaviors. For the lowest pressures (until around $P \approx 2.9 \cdot 10^6$Pa), a liquid feed was assumed and the global minimum was found at 100% CO_2, with a maximum at the H_2O-CH_4O base of the ternary diagram, assuming a liquid feed.
For $P \approx 3 \cdot 10^6 \ldots 5.7 \cdot 10^6$, the global minimum was found at $[0.8 \pm 0.1,\ 0.2 \pm 0.1\ 0]$[4]. These samples also had a local minimum with a higher concentration of carbon dioxide, see Figure 3.9 as an example. They assumed a vapor feed.
Finally, for $P \approx 6.4 \cdot 10^6 \ldots 17.1 \cdot 10^6$ a mix consisting of chiefly CO_2 and a small portion of 60-40 divided methanol-water formed the global minimum. As the pressure increased, the amount of CO_2 at this global minimum decreased, with a clear minimum around this global minimum, see Figure 3.10. These last few samples assumed a vapor feed.
For the last two cases that were 2-phase, some apparent discontinuities can be seen in the *tpd* plot (see again Figure 3.9).

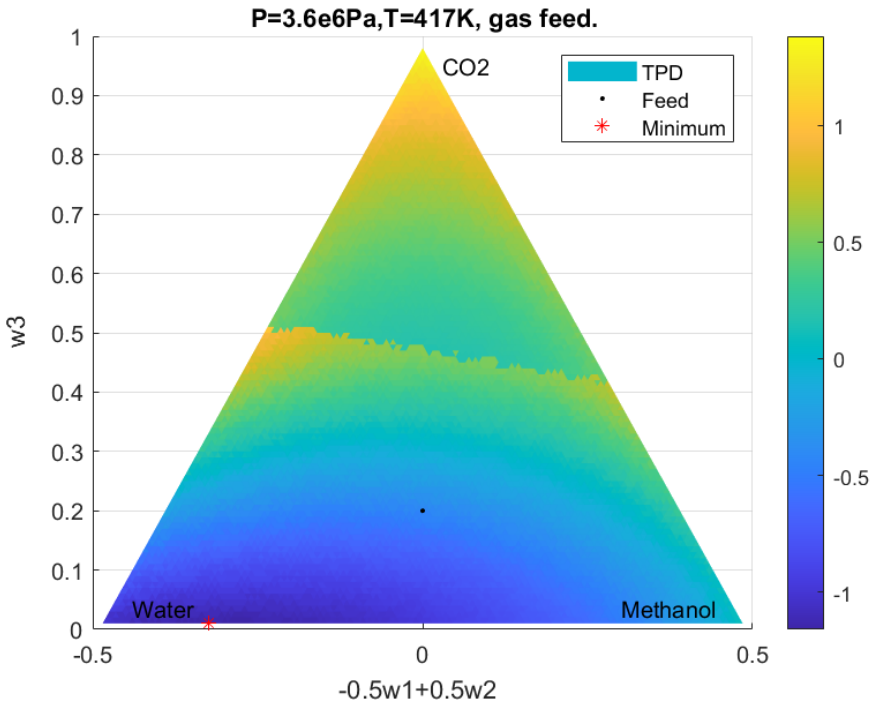


Figure 3.9: Example of the second case ($P \approx 3 \cdot 10^6 \ldots 5.7 \cdot 10^6$) of the unstable points in the middle region. The line in the middle shows a discontinuity in the value of the tangent plane distance.

[4]None of the components have a composition equal to 0 of course, but in these cases the CO_2 composition was negligible.

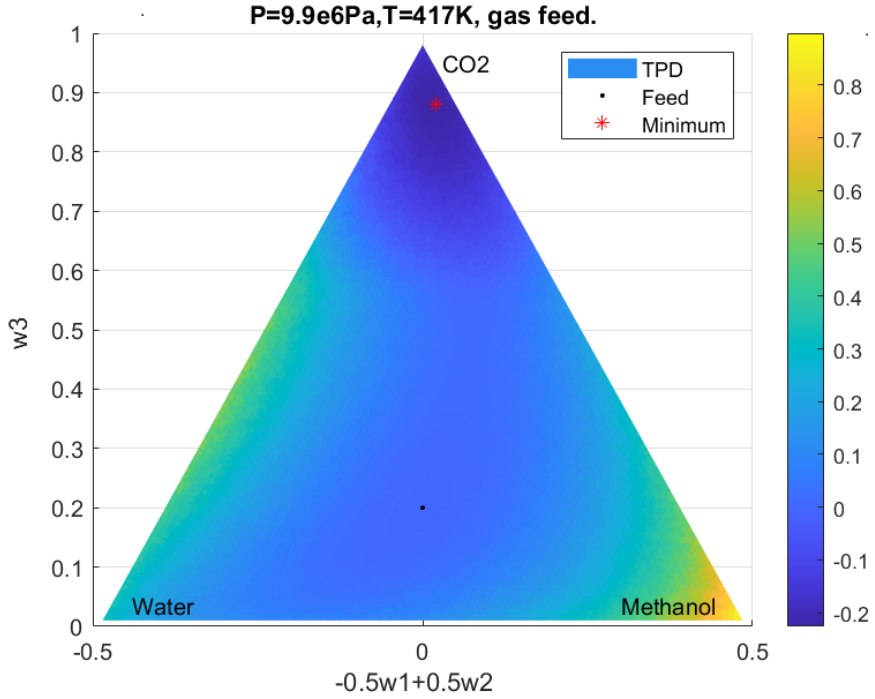


Figure 3.10: Example of the third case ($P \approx 6.4 \cdot 10^6 \ldots 17.1 \cdot 10^6$) of the unstable points in the middle region.

3.2.3 Vapor Region/Right (Red)

50 out of 50 points were (correctly) judged to be stable (single-phase) inputs by tangent plane analysis, and all assumed a vapor feed with a global minimum at $\mathbf{w} = \mathbf{z}$. Aside from very few samples at low pressures, $P \approx 0.7 \cdot 10^6$, the peaks were found near 100% CO_2. This first sample has a very strong and localised peak at 100% H_2O, see Figure 3.11.

3.2.4 Conclusions and Assumptions

From the above, we see that for all tested stable samples, the minimum could be found at $\mathbf{w} = \mathbf{z}$, with no other local minima. This is not surprising, as when a mixture is stable, there is no possibility for the mixture's composition to change in the isolated box. Unstable samples sometimes do have a local minimum. Also, it seems for this feed composition, unstable liquid feeds have a global minimum at (almost) pure CO_2 trial compositions for the new gas phase being formed.
For this fairly simple 3-component mixture a lot of information can be derived from these plots. From the points we tested for above, we could minimise 3 times with a straightforward greedy or local minimisation approach, each time using a pure composition consisting of only one of the three components as an initial guess to find the global minimum. This idea was considered because for each of the P, T combinations a purely descending path from at least one of the pure components can be traced towards the global minimum. In this way information for optimisation- and algorithmic strategies can be visually derived.
More importantly, we now know for sure that we are tasked with finding the minimum in a set which might include local minima and discontinuities. We can also use these plots of the *tpd* to

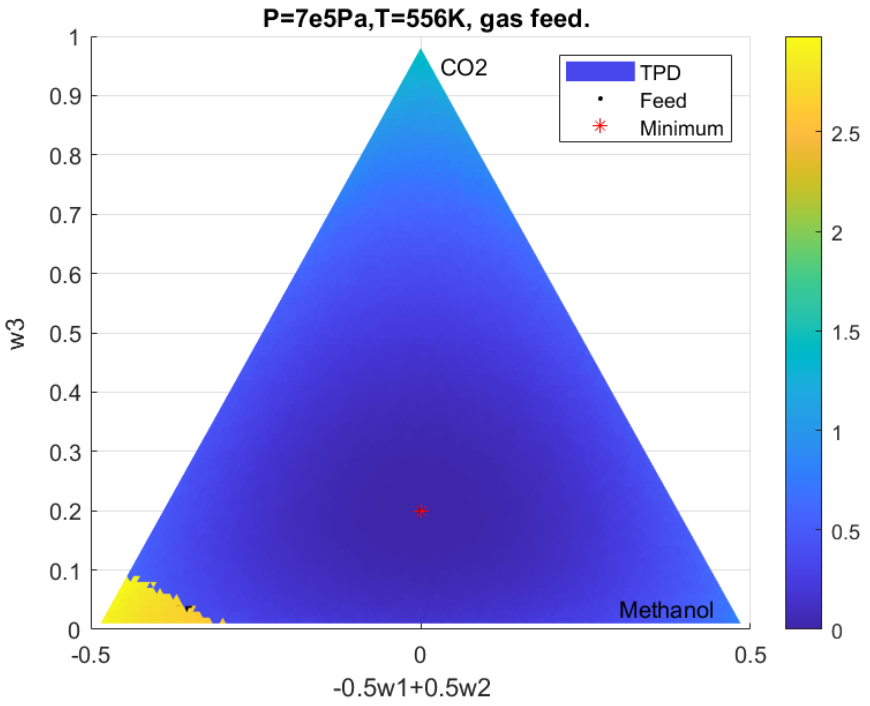


Figure 3.11: A strong, localised peak at pure H_2O

give us information and form hypotheses on peculiarities. For example, Figure 3.1 and the P, V and P, T plots in this chapter show problems estimating the bubble- and dew point in the critical region. As we can see in Figure 3.12, the global minimum is not very "deep", i.e. the value is close to surrounding values (or the gradient is small). Additionally, the global minimum and local minimum share a collective "valley" close to (but below) 0. This might lead an optimisation algorithm to converge to different solutions and even ones that result in barely non-negative values.
Another phenomenon can be observed in the P, V plot of Figure 3.3: the curve of data points in the show a sudden jump halfway. We look to *tpd* plots before and after this discontinuity for an explanation.

Looking at and comparing Figures 3.13, 3.14, 3.15 and 3.16, we hypothesise that the cause of the jump is the mixture going from a more liquid-like state to a more vapor-like state. We draw this conclusion from the observation that before the jump, a liquid feed has the least negative global minimum, and afterwards the vapor feed has the least negative global minimum. Both are unstable, so the phase with the least negative global minimum is the "right" one, as that means that less drastic phase splitting is required to find the (vapor-liquid-)equilibrium. Another interesting observation that supports this is that for the liquid feed, the feed composition lies on the liquid-like H_2O-CH_4O side of the discontinuity before the jump, and on the vapor-like CO_2 side of the discontinuity after the jump. This change from a more liquid-like to a more vapor-like mixture causes a change in the compressibility factor Z, which is used to compute V.

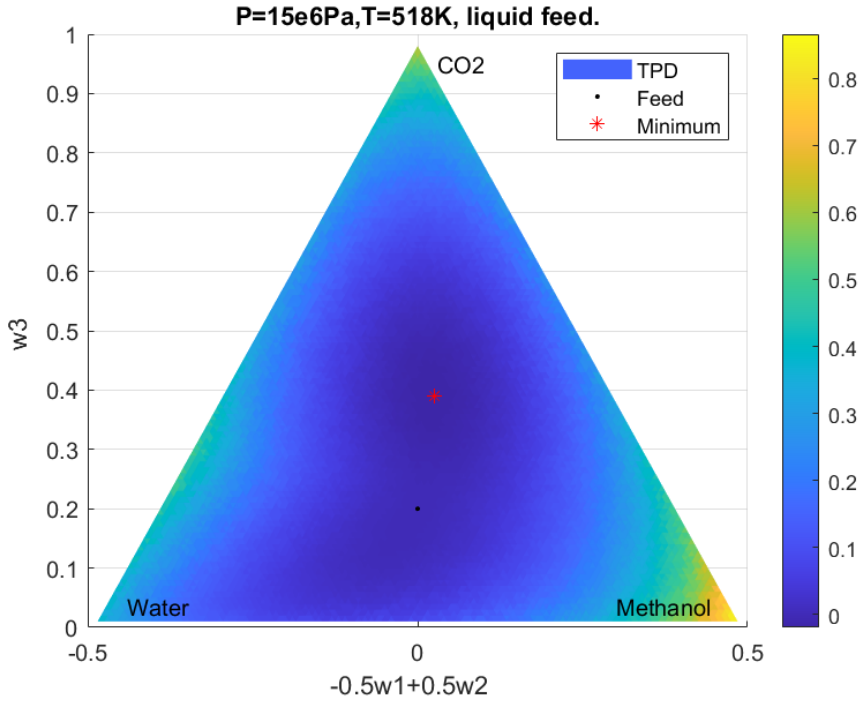


Figure 3.12: Plot of the tpd near the critical point. The plot looked the same for both a liquid- and vapor-assumed feed.

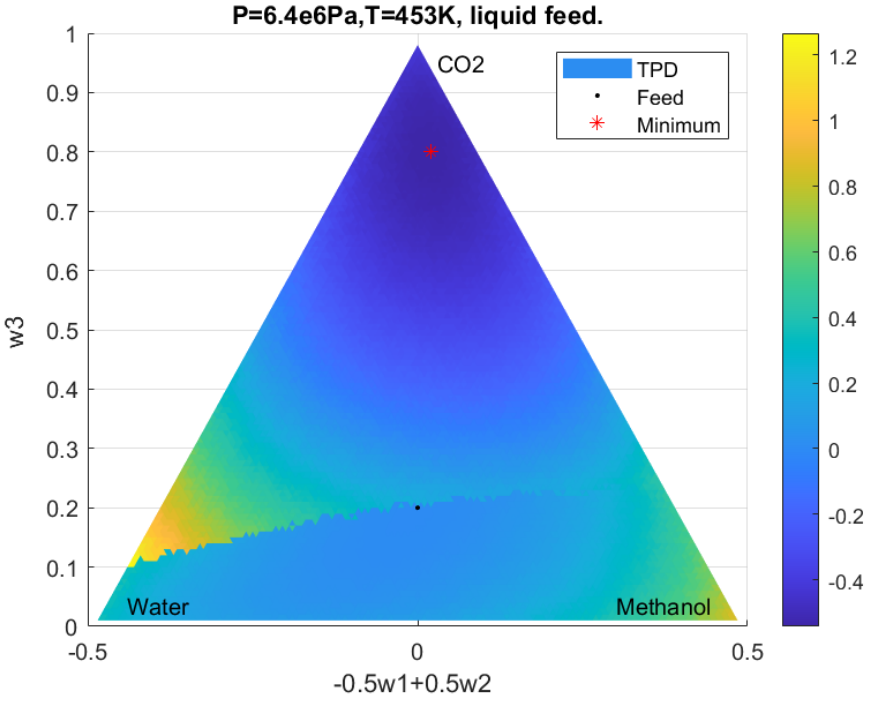


Figure 3.13: Plot of the tpd before the jump in logV (see Figure 3.7), assuming a liquid feed.

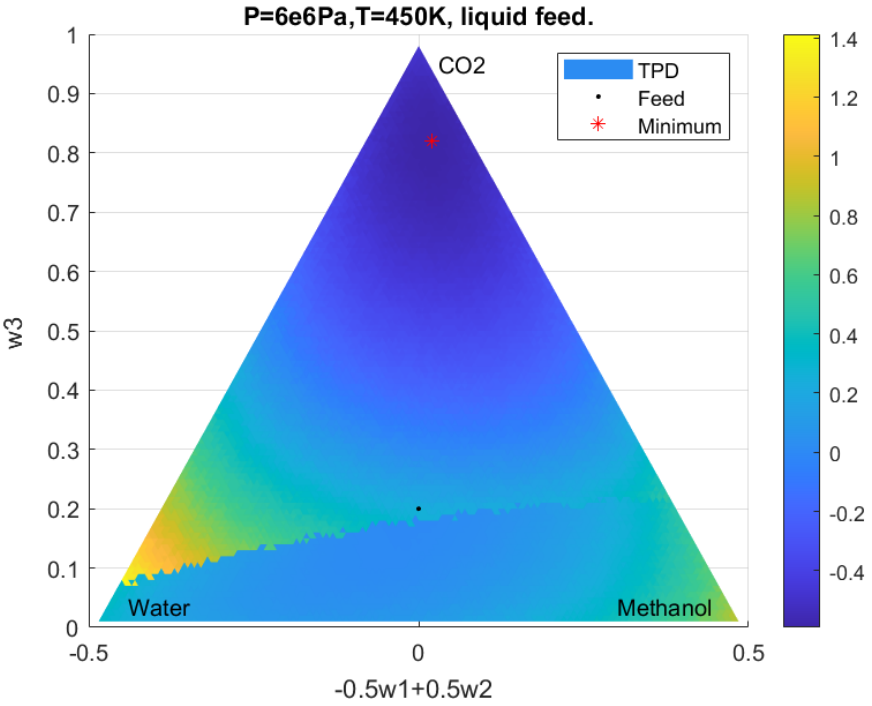


Figure 3.14: Plot of the tpd after the jump, assuming a liquid feed. Notes how the discontinuity has shifted to above the feed from before the jump.

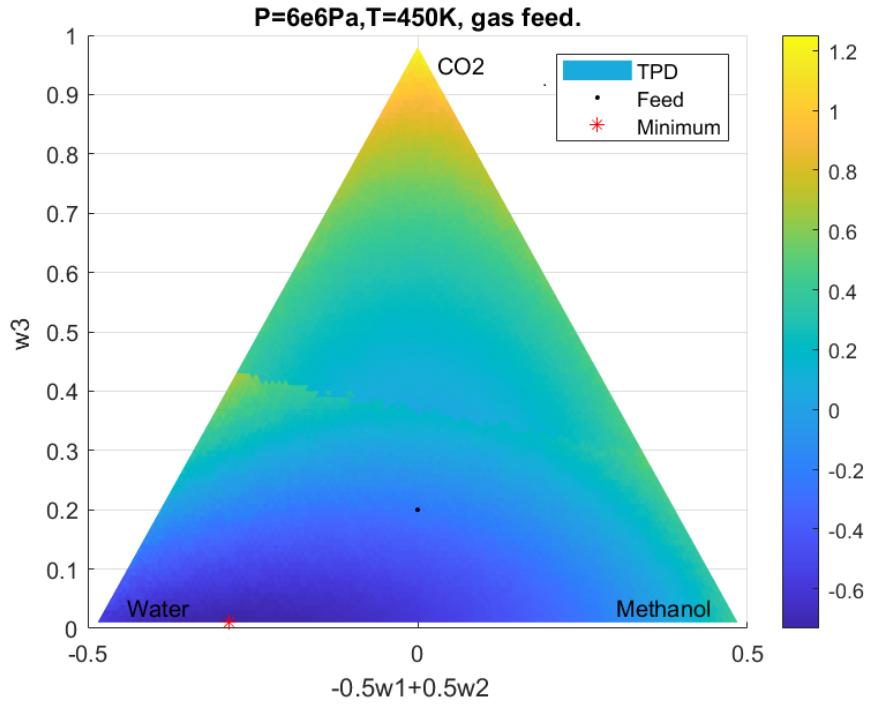


Figure 3.15: Plot of the tpd before the jump, assuming a vapor feed.

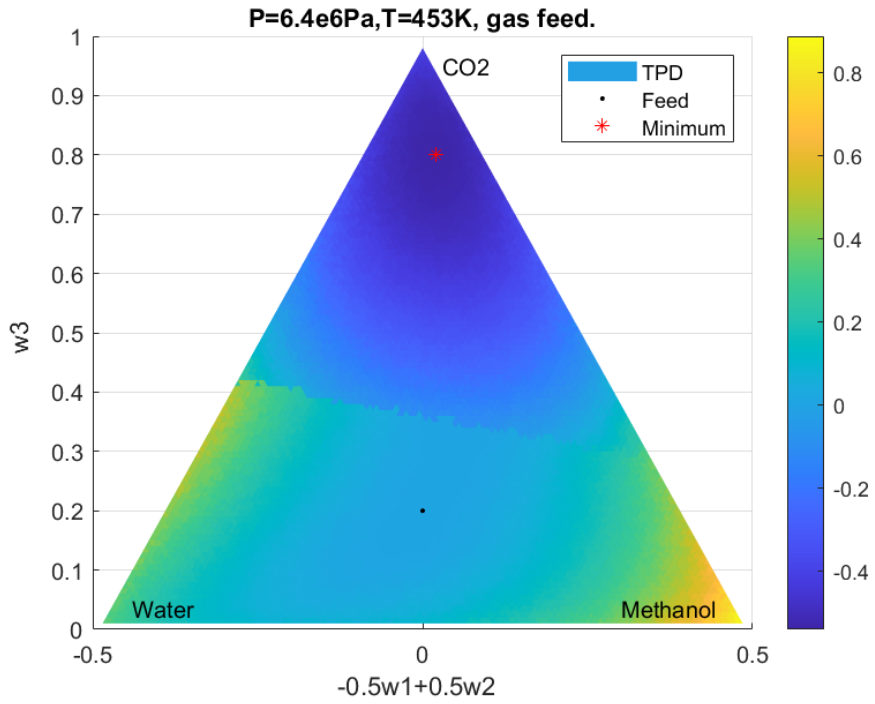


Figure 3.16: Plot of the tpd after the jump, assuming a vapor feed.

Chapter 4

Optimisation Methods

In this chapter, we will look into the optimisation problem associated with tangent plane analysis (see equation (2.38)). We will consider some prominent global optimisation methods applied to problem (2.38) in literature. Then, we will discuss the merits of these methods, and choose which we will implement, going into more detail of the chosen methods.

4.1 Global Optimisation in Literature

Michelsen [25] originally proposed estimating the distribution ratio **K** using the Wilson equation

$$K_i = \frac{P_{c_i}}{P} \exp\left(5.42\left(1 - \frac{T_{c_i}}{T}\right)\right)$$

for low pressures in hydrocarbon systems[1], which could then be used to obtain the following as initial guesses:

$$y_i = K_i z_i$$
$$y_i = \frac{z_i}{K_i}$$

Corresponding to the Wilson Equation's estimations for the bubble point and dew point of the pure components, respectively [58]. From these initial guesses, minima would then be found via direct substitution. Michelsen stated that one of these initial guesses will converge to the trivial solution, and the other to the desired minimum for a specification on the two-phase region. This method, however, gives no guarantee on the probability that the global minimum will be found, or how it deals with peculiarities in the domain. The concept of predicting whether a given mixture would be stable using tangent plane analysis was a great opportunity to more reliably work with stability analysis, but the method for finding the global minimum seemed lacking. Since then, many different ways of finding the global minimum of the tangent plane distance were implemented in literature, as will be shown in this chapter.

[1] meaning systems containing Hydrogen (H) and Carbon (C) in their components.

A nice overview for the global optimisation methods applied so far to the phase stability problem (and other thermodynamic problems) in literature can be found in [57]. This includes both *deterministic* and *stochastic* methods, and below some of these methods that have been successfully applied using cubic equations of state are listed:

- Deterministic methods, which find (a sequence of points converging to) a minimum that is provably the global optimum, often exploiting properties of the objective function (such as continuity or convexity, possibly local).
  - *Homotopy Continuation*: these methods are able to find all roots of a set of nonlinear equations. A relaxed problem is solved first, and a homotopy path and -parameter t are used to gradually increase nonlinearity [17, 49].
  - *Branch and Bound*: provable upper- and lower bounds to the problem are progressively refined until $\varepsilon-$convergence is guaranteed and achieved [13].
  - *Tunneling Method*: a sequence of 2-phase cycles is used. The problem is (locally) minimised in the first phase, and in the second phase (the tunneling phase) a zero not equal to the local minimum is found. This is repeated until convergence [31, 30].
  - *Dividing Rectangles*: for this method, the search space is divided into a hypercube, which is sequentially divided into smaller hyperrectangles [40].
- Stochastic methods, which require little to no assumptions on the characteristics of the problem and converge to a global optimum with a high probability. Though they offer no guarantee of finding the global optimum, they are often less computationally intensive.
  - *Stochastic Sampling and Clustering Method*: starting from an initially generated random set of points in the domain, the region of attraction, defined using clustering, of local minima is searched. Local optimisation is used to cluster different points closer together [2].
  - *Simulated Annealing*: Starting from an initial point, the minimum is found by randomly selecting a point in its neighbourhood, and selecting that point if it has a lower objective value, or with a probability p if it has a higher objective value. The probability p goes down over time [14, 6, 38].
  - *(Adaptive) Random Search*: (Pure) Random Search generates uniformly distributed random points and keeps track of the minimum. Adaptive Random Search generates random points around an initial point, keeping track of the best point. Then, new points are generated around this best point, reducing region size every iteration [19].
  - *Particle Swarm Optimisation*: this is a population-based stochastic optimisation technique, comparable to a genetic algorithm. The points, or "particles" in this context, form a swarm, each particle changing position based on velocity and previous position. From the randomly generated initial population, each particle then adjusts its position according to the personal- and global best iteration, converging to an optimum [5, 37].
  - *Differential Evolution*: Comparable to genetic algorithms. Starting from some point (or "individual", in this context), two other individuals are used to generate ("mutate into") a fourth individual. The best individuals from the target and trial individuals are chosen, converging to better (closer to optimal) individuals [45].
  - *Tabu Search*: While searching for the global minimum, points already looked at are placed in a taboo list and not looked at again. New points will be compared to the

ones in the taboo list. After a while, the taboo list can be used to find promising areas for the optimum, which can then be intensively searched [45].

A more thorough overview of these methods can be found in [57] and their respective papers.

It should be noted that it is not always completely necessary to keep looking until a global minimum is found. Whenever a tangent plane distance with negative value is found, any optimisation strategy could be allowed to terminate, as we have determined the mixture to be unstable [57]. So far, none of the methods explicitly take advantage of this fact in their methodology, aside from sometimes using it as a stopping criterion.
We will not specifically use this knowledge when implementing solution methdos, as this work is focused more on the mathematical (research-) side of the problem. Also, finding the actual global optimum is still useful in determining a better initial guess for the composition (as an input for the flash).

Because the structure of the TPDplane can differ a lot for different numbers and types of components in mixtures, and because we expect them to work better for higher dimensional cases[2] we hypothesise that a stochastic method, or a method that in some way transforms or approximates the *tpd* objective function, would be better suited to minimising the *tpd* than a deterministic method. To explore this hypothesis, we will choose one deterministic method and one stochastic method to implement and compare them to each other.
We compare the methods outlined above to each other and come to the following decision:

- For our deterministic method, we choose to go with Dividing Rectangles (DIRECT). We have chosen this method for its low number of parameters, its stability and its initialisation independence.

- For our stochastic method, we choose to go with Stochastic Sampling and Clustering (SSC), because it has been reported to be a reliable method, and SSC being tweakable via the χ-parameter and UNIRANDI *tol* value can also be attractive in practice. Additionally, the clustering part of the algorithm which finds the local minima can provide valuable insights on what the surface of *tpd* looks like. Problems could arise thanks to the *tpd* function's nonlinearities with the local optimisation part of the algorithm depending on the procedure selected, but that problem can be solved by choosing a local optimisation method that is not affected too much by that (see also Section 4.3).

Also, we discuss an approximation method that could transform/approximate the *tpd* function using Taylor series, and solve it using standard Semi-Definite Programming (SDP) methods. Unfortunately, this proved to be infeasible (see Section 4.4). Finally, we look into a few more "primitive" (stochastic) methods, as these are claimed and accepted to be inferior to more rigorous global methods, but a more direct comparison with the more elaborate methods could be useful. This way, we have different types of methods to compare to each other.
In the sections below, we will discuss the methods that will be implemented more deeply. Specific notes on the final MatLab implementations of the methods can be found in Appendix E.

[2] i.e., mixtures with higher numbers of components.

4.2 Dividing Rectangles (DIRECT)

DIRECT is based on Lipschitzian methods [18]. The name comes from being a direct optimisation method, and because DIviding RECTangles is a core component of the algorithm. In Lipschitzian optimisation, a Lipschitz constant K is used to balance a method between a more global- or local-based search. The larger K is, the more globally a method is focussed. In this context, a method being focused "globally" can be seen as the method not trying to exclude any areas of search, which might happen when using a more "local" focus, i.e. going for the most direct way of minimising the objective function. Advantages of these methods are that, assuming a Lipschitz constant K has been found, convergence of the methods is theoretically provable: they are deterministic methods. Additionally, the methods require no additional parameters to work aside from the Lipschitz constant K, and they are able to estimate how close they are to the global minimum via bounds, allowing for specialised stopping criteria [43]. However, these methods have three main problems [18]:

1. The Lipschitz constant cannot always be determined, and even if it can, it might be computationally too intensive to do so.

2. The method should never be "too local": to guarantee that a global optimum is found, K is often quite large (it must exceed the maximum rate of change of the objective function). This puts a heavy global emphasis on the method, causing a slow convergence.

3. Third, these methods partition the search space into many (hyper)rectangles and squares, and evaluate on all 2^n vertices of each of these hyperrectangles, causing the complexity of the problem to grow very large with the dimension n.

DIRECT tries to benefit from the pros of Lipschitzian optimisation while getting rid of the cons by not using one specific Lipschitz constant K, but all of them (this will be more specifically in Section 4.2.1). Thus, it solves Problem 1 by not requiring to find a K and Problem 2 because it now operates both locally and globally, allowing it to quickly focus on the more local part of the algorithm as soon as an area of optimality is found, enabling faster convergence. Problem 3 is solved by not evaluating all the vertices of the hyperrectangles, but instead looking at the center of the rectangles, a singular point per rectangle.

4.2.1 Univariate DIRECT

The method will be explained by first outlining a univariate version of DIRECT (**Algorithm** 1), and then extending this to higher dimensions (**Algorithm** 3).
One way DIRECT is more efficient than the original Lipschitzian methods because it samples centers of rectangles instead of their vertices. This means that in order to keep center points in the set of all rectangles from partition to partition, rectangles should be divided into thirds, and evaluated at the center of each interval, the original center becoming the center of a smaller interval (but at the same position), see also Figure 4.1.

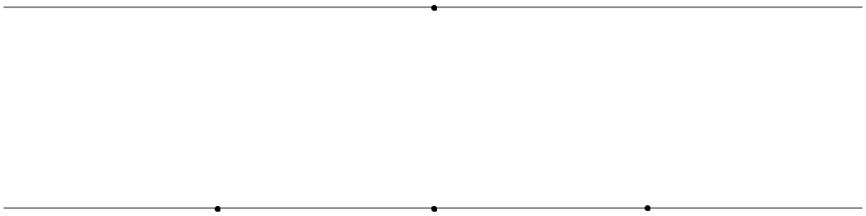


Figure 4.1: Partitioning an interval with center points. The top line is before subdivision, the bottom line is after subdivision.

But how do we now select which intervals are worth looking in to? Suppose the (1-dimensional) search space has been partitioned into intervals $[a_i, b_i]$, $i = 1, \ldots, m$. We can (construct a visual aid to) evaluate the value of each point by calculating $(b_i - a_i)/2$, the distance from the center to the edges of the interval and $f(c_i)$, the objective value at the interval's center point, and using these as horizontal- and vertical coordinates, respectively. See also Figure 4.2.

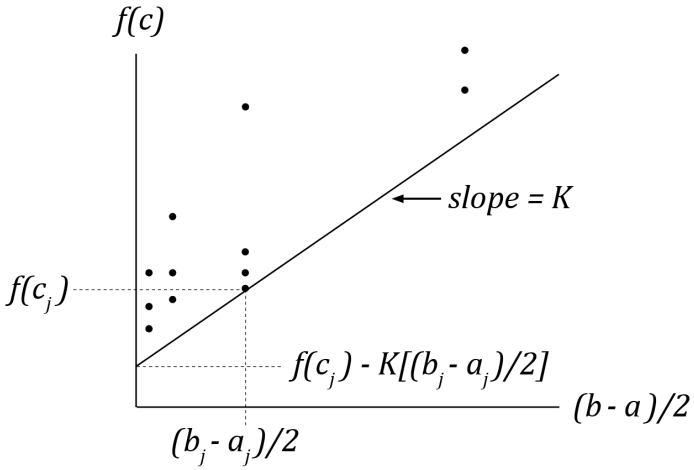


Figure 4.2: In this plot, the potential optimality of each rectangle is represented as a point in $\mathbb{R}^2$. A rectangle j is potentially optimal if $\exists K > 0$ s.t. its associated point in the plot can be reach by moving the line with slope K up until it touches any point in the space.

The horizontal coordinate represents the "global goodness" of the interval, based on the amount of unexplored space it contains. The vertical coordinate represents the "local goodness" of the interval, based on its objective value. The most promising interval can now be found by positioning a line with slope K under the cloud of dots, and moving this line up until it intersects

with a dot (see Figure 4.2). This K then corresponds to a Lipschitz constant, determining how globally/locally focused the method will be. As mentioned before, DIRECT tries to operate both locally and globally, in essence using all possible values for K at the same time, which translates to finding the set of intervals that could be detected using a line with slope $K > 0$, see Figure 4.3. DIRECT then selects (and samples within) all of these intervals during one iteration. More formally, this leads to the following definition of a **potentially optimal** interval:

Definition. (Potentially Optimal)

Suppose we have partitioned the interval $[l, u]$ into intervals $[a_i, b_i]$ with midpoints c_i for $i = 1, \ldots, m$. Let $\varepsilon > 0$ a constant, and f_{min} the current best function value. Interval j is said to be **potentially optimal** if there exists some rate-of-change constant $\tilde{K} > 0$ such that

$$f(c_j) - \tilde{K}\left[\frac{b_j - a_j}{2}\right] \leq f(c_i) - \tilde{K}\left[\frac{b_i - a_i}{2}\right] \text{ for all } i = 1, \ldots, m \tag{4.1}$$

$$f(c_j) - \tilde{K}\left[\frac{b_j - a_j}{2}\right] \leq f_{min} - \varepsilon|f_{min}| \tag{4.2}$$

Condition (4.1) forces the interval to be on the lower right of the convex hull of the points, and (4.2) makes sure that the lower bound for the interval is nontrivially better than the current best solution. In practice, the performance of the algorithm is insensitive to the value of ε [18].

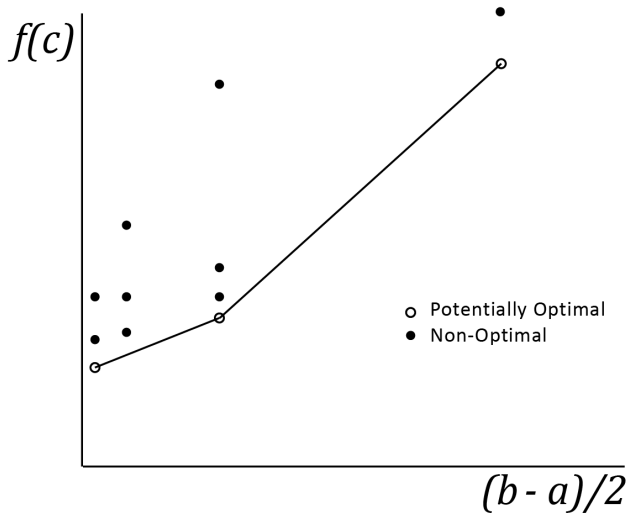


Figure 4.3: Combining different lines with slopes $K > 0$ into S_{OPT}. See also Figure 4.2

This process of identifying potentially optimal intervals, and partitioning each of them until eventually an optimum is reached leads us to DIRECT in one dimension, i.e. **Algorithm 1**.

Algorithm 1 Univariate DIRECT

1: **procedure** UNIVARIATE DIRECT
2: INPUT: an interval $[l, u]$, the objective function f
3: OUTPUT: A global optimum f_{min}, and its input c_j, the center of an interval
4: INITIALISE: $m = 1$, $[a_1, b_1] = [l, u]$, $c_1 = \frac{a_1+b_1}{2}$, $f_{min} = f(c_1)$, $iter = 0$
5: Determine S_{OPT} the set of potentially optimal intervals
6: Select any interval $j \in S_{OPT}$
7: Let $\delta = \frac{b_j-a_j}{3}$, $c_{m+1} = c_j - \delta$, $c_{m+2} = c_j + \delta$, f_{min} =min$\{f(c_m), f(c_{m+1}), f(c_{m+2})\}$
8: Add intervals $[a_{m+1}, b_{m+1}] = [a_j, a_j + \delta]$ with center c_{m+1} and $[a_{m+2}, b_{m+2}] = [a_j + 2\delta, b_j]$ with center c_{m+2}
9: Modify j by $[a_j, b_j] = [a_j + \delta, a_j + 2\delta]$, $m = m + 2$
10: $S_{OPT} = S_{OPT} \setminus \{j\}$. If $S_{OPT} \neq \emptyset$, go to Step 6
11: $iter = iter + 1$. If $iter = iter_{max}$ STOP. Else, go to Step 5

In **Algorithm** 1, the order in which potentially optimal intervals j are chosen is not important, as long as each of them is eventually chosen. Identifying the set of potentially optimal intervals S_{OPT} (Step 5) can be done with algorithms such as Graham's Scan (see Section 4.2.3 for more details) or solving the system of equations from the definition. In the next section, we will extend **Algorithm** 1 to the general $n-$dimensional case.

4.2.2 Extending to Higher Dimensions

For the eventual algorithm, the n-dimensional unit hypercube is used as a search space: all variables are assumed to be within the interval[3] [0,1], which will be divided into hyperrectangles with a sampled point at their center. In higher dimensions, dividing hyperrectangles is not as straightforward as in the univariate case, because now the order of dimensions along which to divide matters, as can be seen in Figure 4.4.

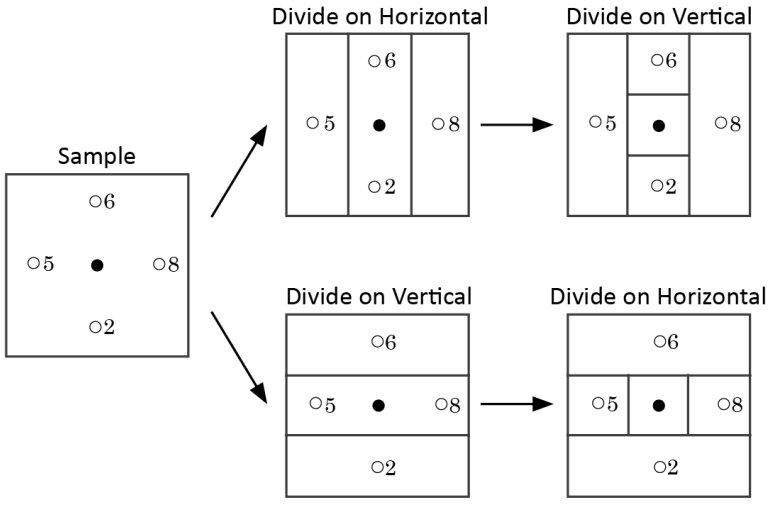


Figure 4.4: Different ways to subdivide the intervals. The numbers represent function values. Notice how the order of division matters for the positions of the larger rectangles.

[3]For the stability problem this is the case, but for other problems this could always be obtained via scaling.

Instead of choosing this order arbitrarily, it will be done in such a way the best centers lie in the largest rectangles, as this strengthens the emphasis on local search, and speeds up convergence without sacrificing consistency [18]. In order to achieve this, it is first noted that if we split first on dimension i, then the points $\mathbf{c} + \delta\mathbf{e}_i$, will be at the centers of the biggest subrectangles (see also Figure 4.4). This leads us to a rule for subdividing hypercubes:
Let $p_i = \min\{f(\mathbf{c} + \delta\mathbf{e}_i), f(\mathbf{c} - \delta\mathbf{e}_i)\}$ the best function values sampled along dimension i. Split along the dimension with the smallest $p-$value first, and then split the rectangle containing $\mathbf{c}$ into thirds along the dimension with the next smallest p-value and so forth. So in Figure 4.4, the bottom order (dividing on vertical first) would be chosen. This first subdivision will result in both hyperrectangles and hypercubes. For subdividing the rectangles, only the long dimensions will be subdivided, ensuring that the rectangles shrink in every dimension.

Algorithm 2 contains a more formal description on how to divide the rectangles.

Algorithm 2 Dividing Rectangles

1: **procedure** PROCEDURE FOR DIVIDING RECTANGLES
2: INPUT: a rectangle j with center $\mathbf{c}$
3: OUTPUT: a division of the rectangle, consisting of multiple other rectangles.
4: Identify the set I of dimensions with the maximum side length. Let δ equal one-third of this maximum side length.
5: Sample the function at the points $\mathbf{c} \pm \delta\mathbf{e}_i \ \forall i \in I$
6: Divide the rectangle containing $\mathbf{c}$ into thirds along the dimensions in I, starting with the dimension with the lowest value of $p = \min\{f(\mathbf{c} - \delta\mathbf{e}_i), f(\mathbf{c} + \delta\mathbf{e}_i)\}$, and continuing in order along the other dimensions in I.

For each rectangle, $f(\mathbf{c}_i)$ and d_i, the distance from $\mathbf{c_i}$ to the vertices, are known. This allows us to form a similar figure to Figure 4.2, using d for the horizontal coordinates, and forming a definition for potentially optimal intervals:

Definition. (Potentially Optimal)

Suppose that we have a partition of the unit hypercube into m hyperrectangles. Let $\mathbf{c}_i$ denote the center point of the ith hyperrectangle, and let d_i denote the distance from the center point to the vertices. Let $\varepsilon > 0$ a positive constant. A hyperrectangle j is said to be potentially optimal if there exists some $\tilde{K} > 0$ such that

$$f(\mathbf{c}_j) - \tilde{K}d_j \leq f(\mathbf{c}_i) - \tilde{K}d_i \text{ for all } i = 1, \ldots, m \tag{4.3}$$

$$f(\mathbf{c}_j) - \tilde{K}d_j \leq f_{min} - \varepsilon|f_{min}| \tag{4.4}$$

We can now combine all of this theory to extend DIRECT to higher dimensions. After initialising, by sampling at the center of the unit hypercube, we iteratively identify the set of potentially optimal hyperrectangles and sample and subdivide those as described, until a prespecified iteration limit is reached[4]. A formal description of the algorithm can be found in **Algorithm** 3.

[4]If a Lipschitzian constant has been found, this can be used to form a stopping criterion, but we will assume that we do not have such a constant here, and thus will not use this fact.

Algorithm 3 DIRECT

1: **procedure** DIRECT
2: INPUT: a search space S, an objective function f
3: OUTPUT: a global optimum $x^* = \mathbf{c}_j$ with value f_{min}
4: INITIALISE: Normalise the search space to be the unit hypercube, with center $\mathbf{c}_1$, set $f_{min} = f(\mathbf{c}_1)$, $m = 1$, $iter = 0$
5: Identify the set S_{OPT} of potentially optimal rectangles
6: Select any rectangle $j \in S_{OPT}$
7: Using **Algorithm** 2, determine where to sample in rectangle j and how to subdivide it. Update f_{min} and set $m = m + \Delta m$, where Δm is the number of new points sampled.
8: $S_{OPT} = S_{OPT} \setminus \{j\}$. $S_{OPT} \neq \emptyset$, go to Step 6
9: Set $iter = iter + 1$. If $iter = iter_{max}$, stop. Otherwise, go to Step 5

Identifying the set of potentially optimal rectangles is done in the same way as in one dimension (see Section 4.2.3). DIRECT is guaranteed to converge when the objective function is continuous in the neighbourhood of the global minimum [40].

4.2.3 Potentially Optimal Intervals

As mentioned above, DIRECT requires a way to identify which of the rectangles is potentially optimal. One way to do this is Graham's Scan [12]. Graham's Scan is an algorithm that can be used to determine the convex hull of a set of points. This can be used to determine the set S_{OPT} of potentially optimal intervals by representing the intervals in a two-coordinate system as mentioned in Sections 4.2.1 and 4.2.2, and shown inFigure 4.2: the underside of the convex hull of these points now is all the points that can be reached by placing a line with positive slope K underneath the cloud of points and moving it up.
Graham's Scan is formally described in **Algorithm** 4.

Note that in Step 6 of **Algorithm** 4, thanks to the choice of P, we have that $\phi_{k-1} - \phi_k \leq \pi$ (indices module n). Each time Step 8 is performed, either the number of possible points in the convex hull of S is reduced by one, or the total number of points considered in S' reduces by one. Because of this, we can stop the procedure after Step 8 has been performed $2n'$ times [12].

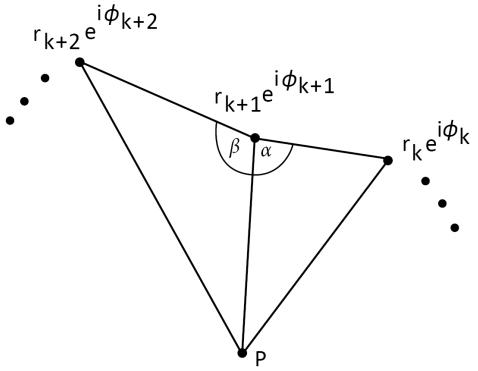


Figure 4.5: The triangles determining α and β, used to check whether point $k+1$ is part of the interior of the convex hull of S.

Algorithm 4 Graham's Scan

1: **procedure** GRAHAM'S SCAN
2: INPUT: a set S to find the convex hull of
3: OUTPUT: conv(S)
4: find a point $P \in \text{int}(\text{conv}(S))$. This can be done by finding $\{s_1, s_2, s_3\} \in S$ a set of three points, and testing them for colinearity. If they are colinear, discard the middle point and find a new point in S to get a three point set. Repeat until we have a set of three points that are not colinear, e.g. $\{s_1, s_2, s_3\}$. P can then be chosen as the middle of the triangle between x, y, z
5: Express each point in S in polar coordinates with origin P and $\theta = 0$ in the direction of an arbitrary half-line L from P
6: Order the elements $\rho_k \exp(i\theta_k)$ of S in terms of increasing θ_k. Now, $S = \{r_1 \exp(i\phi_1), \ldots, r_n \exp(i\phi_n)$, where $0 \leq \phi_1 \leq \cdots \leq \phi_n \leq 2\pi$, $r_i \geq 0, i = 1, \ldots, n\}$.
7: If $\phi_i = \phi_{i+1}$ then the point with smaller amplitude is not part of the convex hull and can be deleted, just like any points with $r_i = 0$. This gives us a set $S' = \{r_1 \exp(i\phi_1), \ldots, r_n \exp(i\phi_{n'})\} \subseteq S$, $n' \leq n$.
8: Start with three consecutive points in S', $r_k \exp(i\phi_k)$, $r_{k+1} \exp(i\phi_{k+1})$, $r_{k+2} \exp(i\phi_{k+2})$. There are two possibilities for $\alpha + \beta$, the sum of the angles that are at the base of the triangle formed by $r_k \exp(i\phi_k), r_{k+1} \exp(i\phi_{k+1})$ and P and the triangle formed by $r_{k+2} \exp(i\phi_{k+2}), r_{k+1} \exp(i\phi_{k+1})$ and P, respectively (see also Figure 4.5):
9: (i) $\alpha + \beta \geq \pi$: in this case, delete $r_{k+1} \exp(i\phi_{k+1})$ from S', and return to the start of Step 8, now with points $r_{k-1} \exp(i\phi_{k-1}), r_k \exp(i\phi_k), r_{k+2} \exp(i\phi_{k+2})$, indices module n'
10: (ii) $\alpha + \beta < \pi$: in this case, return to the start of Step 8, now with points $r_{k+1} \exp(i\phi_{k+1}), r_{k+2} \exp(i\phi_k), r_{k+3} \exp(i\phi_{k+3})$, indices modulo n'

Thus, what Graham's Scan effectively does, is use polar coordinates to order the points in S, and check for each set of three points whether the middle point lies to the left of the line between the first and last point, i.e. whether the middle point lies in the interior of the convex hull of S. If so (if $\alpha + \beta \geq \pi$), the point is deleted out of the set of points that potentially form the convex hull of S, and a previous point is added to the start to check again. If not (if $\alpha + \beta < \pi$), then the check is again performed for the next pair of three points.

4.3 Stochastic Sampling and Clustering

Like most stochastic methods, SSC starts from a sample of points drawn randomly from S, the space containing all possible inputs $\mathbf{w}$:

$$S = \{(w_1, \ldots, w_C) | w_i \in (0, 1), \sum_{i=1}^{C} w_i = 1\}$$

In addition to this sampling phase, or global phase, it also contains a local phase, where a local search procedure is used, starting from sampled points. We want to avoid finding the same local minimum multiple times, which would cause unnecessary computation and could prevent other local minima from being found. Thus a *region of attraction* of a local minimum x^* is defined as the set of all $x \in S$ that converge to the local minimum x^* using the local search procedure. The aim is to start the local search procedure no more than once in each region of attraction. This method is modifiable in the sense that a user can specify the parameter γ (see also below). A γ

closer to 1 means a higher computation time as more samples have to be considered (assuming the same n as for a lower γ), but also means a higher likelihood of finding the correct global minimum.
A pseudoalgorithm of the method is described in **Algorithm** 5.

Algorithm 5 SSC

1: **procedure** STOCHASTIC SAMPLING AND CLUSTERING
2: INPUT: n (the number of points), S, S_K (the current sample), S_T (the transformed sample), γ.
3: OUTPUT: a global minimum y
4: INITIALISE: $S_K = \emptyset$, $S_T = \emptyset$
5: Draw n points uniformly at random from S and add them to S_K.
6: Obtain S_T by taking the $\gamma\%$ points with lowest function value.
7: Use the clustering procedure on each known local minimum $\mathbf{x}^* \in S_T$ to cluster S_T. If all points in S_T are clustered, go to Step 10.
8: **while** $\exists \mathbf{x} \in S_T$, where $\mathbf{x}$ unclustered. **do**
9: Apply local search to $\mathbf{x}$ to assign it to a cluster. If a new local minimiser is found, go to Step 5
10: Determine the smallest local minimum, this is the returned global minimum.

Algorithm 5 includes both a local search and clustering a procedure. We will discuss these procedures in Sections 4.3.1 and 4.3.2.

4.3.1 Local Search

Different local search procedures are possible to quickly and reliably find a local minimum, for example a quasi-Newton method with Davidon–Fletcher–Powell update formula and UNIRANDI, a random walk type direct search method [9]. Analysis in literature has shown that UNIRANDI is well-suited for SSC as a robust method that is derivative-free (as it only relies on function evaluations), among other things. This local search method will also inform our clustering method later, see Section 4.3.2. The method operates in two steps: first, a trial direction to search in is generated, and then a line search is performed, both performed iteratively until a stopping condition is met. When, starting in some function value, the function value cannot be improved in some search direction $\mathbf{d}$, the direction is reversed. If both of these directions are unsuccessful, the step size h is halved, until eventually a local minimum is found. A pseudoalgorithm for UNIRANDI is described in **Algorithm** 6 [3].

In **Algorithm** 6, h is the step size, $\mathbf{d}$ is the search direction and x_0 is a point in S_T that is unclustered. The LineSearch method mentioned in Steps 9 and 15 of **Algorithm** 6 is described as a pseudoalgorithm in **Algorithm** 7.

Thus, the LineSearch procedure starts if the line in direction $\mathbf{d}$ has improvement, and searches along the line in the direction $\mathbf{d}$ through x_{trial}, until a local minimum is found.

4.3.2 Clustering

The clustering procedure tries to exclude points from which the local search would find a local minimum that has been found before. Clusters are formed around so-called seed points. Multiple

Algorithm 6 UNIRANDI local search

procedure UNIRANDI
INPUT: objective function f ($tpd(\mathbf{w})$, in this case), starting point $\mathbf{x}_0$, *tol* the threshold value of the step length
OUTPUT: $f_{best}, \mathbf{x}_{best}$, the local minimum value and its location.
INITIALISE: $h = 0.001$, fails=0, $\mathbf{x}_{best} = \mathbf{x}_0$
    while Convergence criterion has not been met **do**
        $\mathbf{d} \sim N(0, 1)$
        $\mathbf{x}_{trial} = \mathbf{x}_{best} + h \cdot \mathbf{d}$
        if $f(\mathbf{x}_{trial} < f(\mathbf{x}_{best})$ **then**
            $[\mathbf{x}_{best}, f_{best}, h]$ =LineSearch$(f, \mathbf{x}_{trial}, \mathbf{x}_{best}, \mathbf{d}, h)$
            $h = h/2$
        else
            $\mathbf{d} = -\mathbf{d}$
            $\mathbf{x}_{trial} = \mathbf{x}_{best} + h \cdot \mathbf{d}$
            if $f(\mathbf{x}_{trial}) < f(\mathbf{x}_{best})$ **then**
                $[\mathbf{x}_{best}, f_{best}, h]$ =LineSearch$(f, \mathbf{x}_{trial}, \mathbf{x}_{best}, \mathbf{d}, h)$
                $h = h/2$
            else
                fails = fails+1
                if fails≥ 2 **then**
                    $h = h/2$
                    fails = 0
                    if $h < tol$ **then**
                        RETURN $f_{best}, \mathbf{x}_{best}$

possible clustering procedures are possible, but considering arguments from literature we choose to use single-linkage clustering [4, 9]. Single linkage clustering partitions the set into single element subsets, and then it repeatedly fuses two subsets with minimal distance. A definition of distance is required, and defined as follows:

$$d(\mathbf{x}, \mathbf{x}') = ((\mathbf{x} - \mathbf{x}')^T H(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}'))^{1/2} \tag{4.5}$$

where $H(\mathbf{x}^*)$ is the Hessian of the objective function. For UNIRANDI, the identity matrix I can be used instead of $H(\mathbf{x}^*)$ [4][5]. A point $\mathbf{x}$ is only added to a cluster G if:

$$\exists \mathbf{x}' \in G \text{ s.t.} \tag{4.6}$$

$$d(\mathbf{x}, \mathbf{x}') \leq \frac{\Gamma(1 + \frac{1}{2}C)|H(\mathbf{x}^*)|^{1/2} m(S)}{\pi^{C/2}} (1 - \alpha^{1/(n'-1)})]^{1/C} \tag{4.7}$$

where $m(S)$ is a measure of S, n' is the total number of sampled points, Γ is the gamma-function

$$\Gamma(z) = \int_0^\infty \xi^{z-1} e^{-\xi} d\xi \tag{4.8}$$

and $0 < \alpha < 1$ is a parameter of the clustering procedure [4].
For this specific case, we will let $m(S)$ be the Lebesgue measure of S. A pseudoalgorithm of the method can be found at **Algorithm** 8.

[5]For most other local search methods, other approximations of the Hessian exist

Algorithm 7 Line Search

1: **procedure** LINESEARCH
2: INPUT: objective function f ($tpd(\mathbf{w})$, in this case), the current point $\mathbf{x}_{trial}$, the best point found so far $\mathbf{x}_{best}$, and $\mathbf{d}$ and h, the current direction and step length
3: OUTPUT: $f_{best}, \mathbf{x}_{best}$, h the local minimum value, its location and the step length.
4: **while** $f(\mathbf{x}_{trial}) < f(\mathbf{x}_{best})$ **do**
5: $\mathbf{x}_{best} = \mathbf{x}_{trial}$
6: $f_{best} = f(\mathbf{x}_{best})$
7: $h = 2h$
8: $\mathbf{x}_{trial} = \mathbf{x}_{best} + h\mathbf{d}$

Algorithm 8 Single Linkage

1: **procedure** SINGLE LINKAGE CLUSTERING
2: INPUT: $\mathbf{x}^*$ the seed point.
3: OUTPUT: a clustering around $\mathbf{x}^*$, cluster K
4: INITIALISE: Cluster $K = \{\mathbf{x}^*\} \cup \mathbf{x}'$, where $\mathbf{x}' = \text{argmin}_{\mathbf{x} \in S_T}\{d(\mathbf{x}^*, \mathbf{x})\}$
5: find the $\mathbf{x} \in S_T$ closest to K.
6: **while** $D(\mathbf{x}, K) = \min_{\mathbf{x} \in K}\{d(\mathbf{x}, \mathbf{x}')\} \leq$threshold and $\mathbf{x}$ conforms to gradient criterion (Equation (4.9)) **do**
7: add $\mathbf{x}$ to K
8: find a new $\mathbf{x}$

Algorithm 8 is applied until all points in S_T have been sampled. During this procedure, a set X^* is maintained, containing all local minima found so far. The seed points mentioned in Line 2 are used to grow clusters around, in this case we use the local minima as seed points.
Line 6 of **Algorithm** 8 references a "gradient criterion". This is used to make sure that local search from a point $\mathbf{x}$ points towards the correct local minimum in the cluster [6], making sure clusters are correctly build around the correct local minimum. A condition for a point $\mathbf{x}$ to be added to the cluster around $\mathbf{x}^*$ is

$$\frac{tpd(\mathbf{x} + \varepsilon(\mathbf{x}^* - \mathbf{x})) - tpd(\mathbf{x})}{\varepsilon||\mathbf{x}^* - \mathbf{x}||} < 0 \tag{4.9}$$

for ε a small value. The threshold value in line 6 of **Algorithm** 8 is equal to the value in equation (4.7).

4.4 Transforming Using Taylor Series

The idea behind this solution method is to first transform (approximate) the *tpd* function using Taylor (MacLaurin) series. This then simplifies the objective function to a polynomial, after which the problem can be solved using standard Semidefinite Programming methods. This will also demonstrate a "third" class of methods, next to the deterministic and stochastic classes: one where the objective function is transformed. It is interesting to see if the solutions to a transformed method are useful for the solution for the overall optimisation problem. Transforming the function into an easier to identify form might also help with researching how to best solve it.

[6] It is a check to verify that the negative gradient at $\mathbf{x}$ is verified to point in the direction of $\mathbf{x}^*$.

After some initial testing, this method turned out to not be viable. The objective function (2.32) is too complicated to easily be approximated. The logarithms and the quadratic w_i terms (e.g. $a^{(m)} = \sum w_i w_j a_{ij}$) make approximating it into a quadratic function not feasible. The terms for the fugacity function were attempted to be approximated. Writing Z, the solution to $Z^3 - Z^2 - (A - B - B^2)Z - AB = 0$, explicitly for the two-component (two-dimensional) case, and using the matlab *taylor* function to generate the MacLaurin series for Z resulted in an expression that, when written out, was about 14.000 characters long. Additionally, it might be hard to formulate a positive semidefinite matrix for the constraints for this function, because for example the b_{ij} would form a matrix that is not positive semidefinite. Using some kind of approximating function to write the problem as a semidefinite programming one might still be useful for finding lower bounds and obtaining a relaxation, or to solve the problem using branch and bound [13]. Approximating $\log \phi_i$ using a polynomial in w_i and w_j could further improve on the accuracy of such lower bounds. Due to time constraints and the hypothesis that an approximation would not prove fruitful, no other way of transforming the objective function was found.

4.5 Stochastic Local Methods

Most of the methods discussed in this chapter are quite elaborate in their conception. This is logical seeing as the problem, as has been shown, can get quite complicated and each case can be quite different from the next. Still, it might be interesting to test the effect of more "primitive" methods. It is hypothesised that these primitive methods offer the advantage of easy implementation, while having the possibility of being sufficiently robust and fast, especially for low-dimension cases. Some possible methods are outlined below. Each of these methods in essence comes down to "naively" applying local optimisation from a set of starting points, chosen in particular (random) way.

4.5.1 Shotgun Search

The idea here is to sample n points uniformly at random points in the domain, and then apply local search starting from each of the n points to converge to local minima in the domain. After this has been done, a list of local minima has been obtained, and the smallest one is then chosen as the global minimum. This is, in essence, comparable to what SSC does but with a less intelligent way of assigning points to "clusters".

4.5.2 Grid Search

The idea for this initialisation of the local search method is partially derived from the DIRECT method. What if, instead of going through the process of deciding which rectangles are optimal, we just divide the domain (which has been transformed into the unit hypercube) in n parts along each dimension, creating a grid, and then perform local search from the center of each of the rectangles? At least one of these rectangles should contain the global minimum, if it exists. If this square does not contain a too strong local minimum (or one close to the center of the triangle, the global minimum should be found. Local search can also be unrestricted, allowing the local search method to go outside a grid square. This method then becomes a more structured approach to Shotgun Search.

4.5.3 Pure Component Search

In this method, local optimisation is started from each pure component. The example studied in Chapter 3 spawned this idea, as the global minimum was in the same continuous area as at least one of the pure components, local search should find the global minimum when started from at least one of the local minima. When a local minimum is situated "in between" the pure component and the global minimum, the global minimum might not be reached, as the local search likely gets stuck inside the local minima.

4.5.4 Hypothesis for Results

Looking over the theory, it is hypothesised that between SSC and DIRECT, SSC will perform better in general, with SSC being expected to also always find the global minimum but requiring less computations (see Section 4.1). As for the Stochastic Local Methods, Grid- and Shotgun Search are expected to be able to find the global minimum in all cases, but to be a bit slower than DIRECT and SSC. Pure Search is expected to be the fastest out of all the methods, but to not be completely robust, likely failing to find the global minimum in some cases.

Chapter 5

Testing and results

In this chapter, we will go over the methods we have implemented and their results. All algorithms were implemented using Matlab 2020a, and run on a HP ProBook 400 G6 laptop with Intel Core I5-8265U. Fugacities were calculated according to the SRK equation of state using the Matlab library by Martín et al. [23]. In the following sections, we will first discuss the mixtures the methods will be tested on, then discuss some notes on the implementation of each of the methods chosen in Chapter 4. Then, the results will be listed, and observations that came from these results will be discussed. Appendix E contains some additional information for the implementation of the methods.
To more accurately represent the computation time, we will not only measure the computation time in seconds, but in number of function calls that were made to calculate the objective function *tpd*. Calculating $tpd(\mathbf{w})$ of some composition vector $\mathbf{w}$ involves calculating the fugacity coefficient ϕ, which is usually heaviest calculation done in these algorithms (see also Appendix D). This time measure more uniformly represents the calculation time across different devices, programs, etc.
Calculations were made to a precision of order of magnitude 10^{-4}. This was judged to be precise enough for most practical cases, while allowing to circumvent rounding errors present in MatLab (which for long calculations routinely estimates values equal to 0 to a value in order of magnitude 10^{-17}), and was large enough to save time.

5.1 The Test Mixtures

Recall that an $[x_1 \ \dots \ x_C]$ $C_1 - \cdots - C_n$ mixture means an input mixture (or feed mixture) consisting of x_1 substance C_1, x_2 substance C_2 and so forth, and that the "global minima at $\mathbf{w}$" for these mixtures are the lowest value for their associated tangent plane distance if a new phase with composition $\mathbf{w}$ would be formed, i.e. what the composition of a hypothetical new phase would look like when the mixture reaches equilibrium. When the tangent plane distance is negative for that composition, a new phase is formed because the mixture is unstable, and otherwise it is stable and no new phase is formed. We have decided to consider the following three mixtures as test cases in this work:

1. The [0.4 0.4 0.2] H_2O-CH_4O-CO_2 mixture. This mixture was chosen because it is specifically used in ZEF's case, and was used to test much of the theory and algorithms on. It is

also interesting because we know it has some interesting properties we want our algorithms to overcome (see also Chapter 3). A downside of this mixture is that it does not have much precedent in literature like our next two mixtures. In this case we chose $T = 400$K and $P = 2 \cdot 10^6$Pa, for its discontinuity and two local minima (including the global minimum, see Figure 2.32). The global minimum is at [0.8520, 0.1476, $4 \cdot 10^{-4}$] with a minimum value of -1.2007.

2. The [0.68, 0.32] CH_4O-C_3H_8 mixture at $T = 277.6$K and $P = 10^7$. This is a low-component mixture that is near-critical, and has been tested on before in literature [40]. It has a global minimum at [0.7657, 0.2343] with a value of -0.5332·10^{-3}, barely unstable.

3. Next, the six component mixture [0.3, 0.55, 0.07, 0.03, 0.03, 0.02] N_2-CH_4-C_2H_6-C_3H_8-C_4H_{10}-C_5H_{12} at $T = 151$K and $P = 40 \cdot 10^5$Pa will be used for testing. This mixture also has precedent in literature [2, 40]. It is also interesting because of the large number of components, while still being compatible with the SRK equation of state [40]. This mixture was also studied by Michelsen [26]. The global minimum is at [0.2909,0.5564,0.0730,0.0379,0.0215,0.0204], with a value of 0 (thus, this mixture is stable).

Binary interaction parameters for these mixtures were found using [20] and [52].

5.2 Dividing Rectangles

In accordance with literature, we set $\varepsilon = 10^{-4}$ [40], though the specific value of ε should not have much effect on the performance of the algorithm. As a convergence criterion, we used a tolerance for the change in objective value, *tol* (see also Appendix E). In this case, we set $tol = 10^{-8}$.

5.2.1 Notes on Implementation

In this case, Graham's Scan was used to find potentially optimal rectangles, as solving the system of equations to find potentially optimal rectangles turned out to be very slow time wise (Graham's Scan was 8 to 9 times faster, though the number of *tpd* function evaluations was the same). The procedure is as follows:
First, a picture comparable to Figure 4.3 was made. If only one rectangle was considered (in the first iteration), that rectangle was automatically chosen to be potentially optimal. Then, the convex hull was determined using Graham's scan (**Algorithm** 4). This gave us a set of points larger than the set of potentially optimal ones, as points that lie above and to the left of another point could still not be touched first by a line with slope $K > 0$. Thus, after determining the convex hull, all points in the hull lying above and to the left of at least one other point where deleted, obtaining the set of potentially optimal rectangles.
Choosing the iteration limit $iter_{max}$ is not completely straightforward either. Jones et al. [18] gave a proof that for a number of iterations T_{iter} and a constant $\delta > 0$, then for every number of iterations $t > T_{iter}$, the center-to-vertex distance is less than δ, such that every point in the hypercube lies within a distance δ of some sampled point, meaning the global minimum will be found. What the value of T_{iter} is however, is not fully clear. Thus in our implementation, we stop the algorithm not after a certain number of iterations, but when a sufficiently small change in objective function is observed. This also gives a tunable parameter *tol* by which focus on the speed or precision of the algorithm can be tuned, if required.

For DIRECT to work, the domain must be equal to the unit hypercube. Transforming the domain to the unit hypercube required the problem to be "unconstrained", or else large sections of the eventual hypercube would not be viable solutions, which would require the algorithm to be adjusted in order to still work. Saber and Shaw [40] proposed working in one dimension lower, effectively only using $w_1, \ldots, w_{C-1}$ as variables, and defining w_C as $w_C = 1 - \sum_{i=1}^{C-1} w_i$. However, it is still possible to now find a point $\mathbf{w}' = (w'_1, \ldots, w'_{C-1})$ such that $\sum_{i=1}^{C-1} w'_i \geq 1$, which would then not give a valid value for w_C and by extension $\mathbf{w}$. We solve these problems in this case by searching in the unit hypercube, and normalising each test solution $\mathbf{w}$. Experiments showed that the flash algorithm used to calculate ϕ gives the same result for $\mathbf{w}$ with $\sum w_i \neq 1$ and for the vector $\frac{\mathbf{w}}{\sum w_i}$, i.e. the vector $\mathbf{w}$ normalised. This allows us to search the full hypercube. DIRECT is a deterministic method, meaning that, if started from the same point, it will always take the same path towards the global optimum. Thus, DIRECT was only tested once (as opposed to the 100 times for the SSC method, see Section 5.3 below). For mixture 2, a convergence tolerance of 10^{-6} was used. This proved insufficient for mixture 1, where a tolerance of 10^{-8} was applied instead.

Figure 5.1 shows how DIRECT progressed for Mixture 1.

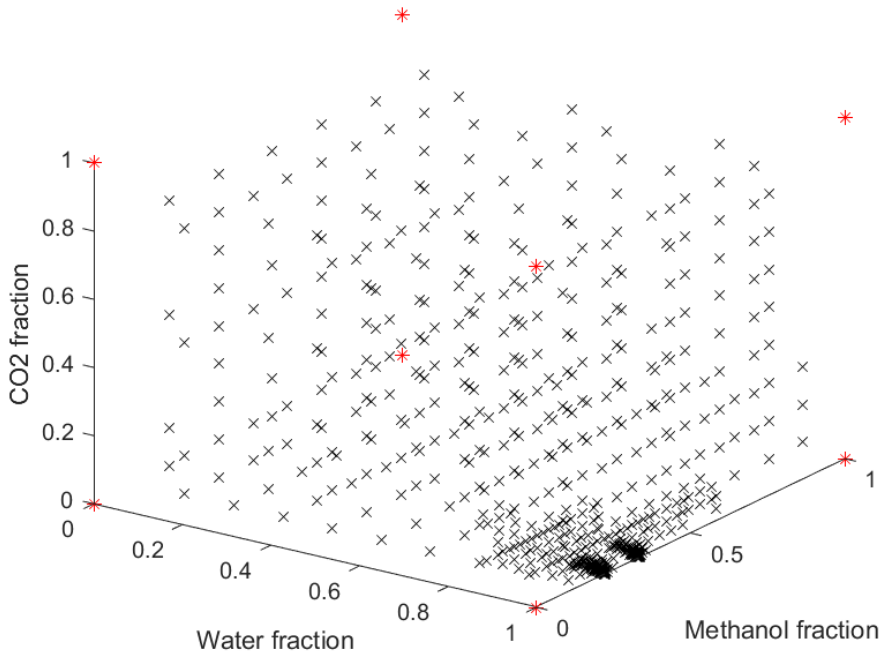


Figure 5.1: Plot showcasing how DIRECT progressed for Mixture 1. Each × shows a point that DIRECT sampled.

5.3 Stochastic Sampling and Clustering

It should be noted that during the clustering procedure (**Algorithm** 8), when tested, the gradient criterion seemed a lot stronger than the threshold (when checking to see if a point could be added to a cluster): when testing for $n = 100$ and $\gamma = 1$, 100 out of 100 clustering procedures were

terminated because of the gradient criterion preventing a point from being added. We discuss the values of the different parameters below:

- ε in the gradient criterion, equation (4.9), is chosen to be $10^{-}5$.
- h, the (starting) step size, $h = 10^{-6}$.
- *tol*, the tolerance of the stepsize h UNIRANDI local search method, was chosen to be 10^{-8}. Testing showed that this combination of *tol* and h achieved the desired consistency without causing unnecessary calculations.
- n, the number of points sampled, was chosen according to the mixture, based on literature or testing to find out what number was the most appropriate. For Mixture 1., we chose $n = 50$, for Mixture 2., $n = 100$. For Mixture 3., we chose $n = 100$.
- γ, the percentage of sampled points used in the algorithm, was set to 1 for each mixture, to put the emphasis of the algorithm of trying to find the global minimum.

5.3.1 Notes on Implementation

UNIRANDI

UNIRANDI is derivative free, and running it from the same x_0 multiple times might yield different results every time because of the randomness involved with finding the search direction $\mathbf{d}$.
UNIRANDI (and its LineSearch method) find the next point on the line with direction d and a step size of h by permuting $\mathbf{w} = \mathbf{w} + \mathbf{d}h$. Here, this is achieved by first generating a random vector $\mathbf{d} \in \mathbb{R}^C$ such that $d_i \sim N(0, 1)$. Now applying $\mathbf{w} = \mathbf{w} + \mathbf{d}h$ would usually not achieve a new vector $\mathbf{w}$ on the line in S, because often one or both of the criteria that $w_i > 0$ or $\sum w_i = 1$ no longer holds. Drawing a random $\mathbf{d}$ with $||\mathbf{d}|| = 1$, adding it to $\mathbf{w}$ and then forcing $\mathbf{w}$ to again be in S via $w_i = \frac{w_i}{\sum w_i}$ for each component of $\mathbf{w}$ causes the effective step size to become smaller each time it is applied, see Figure 5.2. Instead, if we generate $\mathbf{d}$ as a vector such that $||\mathbf{d}|| = 1$ and $\sum d_i = 0$, we make sure that $\sum(w_i + d_i h) = 1$. This can also be seen by realising that the plane of all these $\mathbf{d}$ and the plane of all $\mathbf{w}$ such that $\sum w_i = 1$ are parallel (see Figure 5.3). Now, it is still possible for $\mathbf{w} = \mathbf{w} + \mathbf{d}h$ to result in a $\mathbf{w} \notin S$, by one of the w_i becoming greater or equal to 1, or smaller or equal to 0. When this would happen, we terminate the LineSearch or treat the direction as unviable in the case of UNIRANDI, see also Figure 5.4.

SSC

SSC is slightly tricky to implement. The high degree of variability in the different parameters that can be set (h, γ, tol, n) allows for tuning the algorithm with a focus on speed or accuracy, but as a subsequence, this allows for "wrong" choices. If h and *tol* are not small enough, in such away that UNIRANDI might terminate close to a local minimum instead of directly on the local minimum for example, then SSC might get stuck in a loop were less points in S_T get clustered than n, a "new" local minimum (one close to but not equal to an already existing one) is found, n new points are drawn, and so forth. This means that implementing SSC requires some testing beforehand to determine the correct range for these parameters, making SSC less flexible than it would seem at first glance. For calculating the threshold value (see Section 4.3.2), a definition

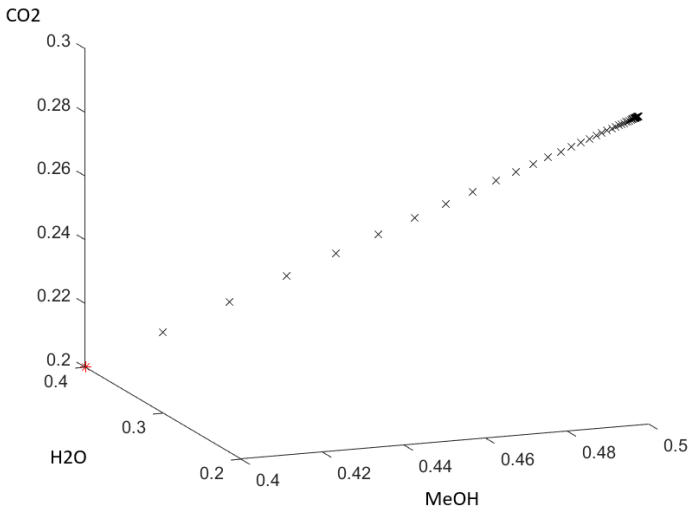


Figure 5.2: Plot showing the progression of $\mathbf{x} = \mathbf{x} + \mathbf{d}h$*, when normalised to enforce* $\sum x_i = 1$*.. *$h = 0.1$ *was set in this case to more easily visually identify the change in step size.*

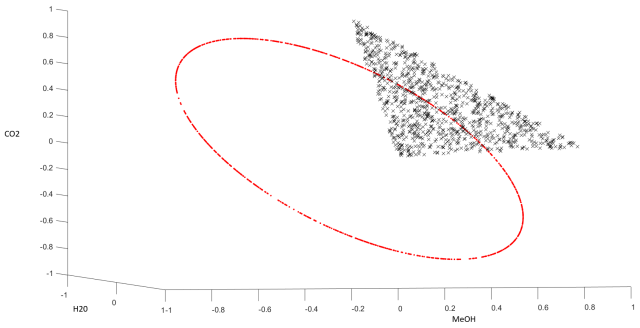


Figure 5.3: Plot showing that the plane of all $\mathbf{d}$ *such that* $||\mathbf{d}|| = 1$ *(the red circle) and the plane of all* $\mathbf{w}$ *such that* $0 < w_i < 1$ *and* $\sum(w_i) = 1$ *(the black triangle) are parallel.*

of the measure $m(S)$ is required. $m(S)$ is chosen to be the Lebesgue measure as suggested by Boender et al. [4]. In our case, S is a $C-$simplex in $\mathbb{R}^{C+1}$, for which the volume can be used as the Lebesgue measure:

$$m(S) = (-1)^{C-1} 2^{C-1} C \tag{5.1}$$

A derivation for $m(S)$ can be found in Appendix E. α is also required to calculate the threshold value, and by convention is chosen to be equal to 0.025 (see also Appendix E)

5.4 Stochastic Local Methods

Each of the methods below came down to generating a set of initial points, and performing UNIRANDI from each of these methods. The parameters were set to $h = 1 \cdot 10^{-6}$ and $tol = 10^{-8}$, for an honest comparison to DIRECT and SSC. The number of points we start the searches, n, from for each of the methods are chosen as follows:

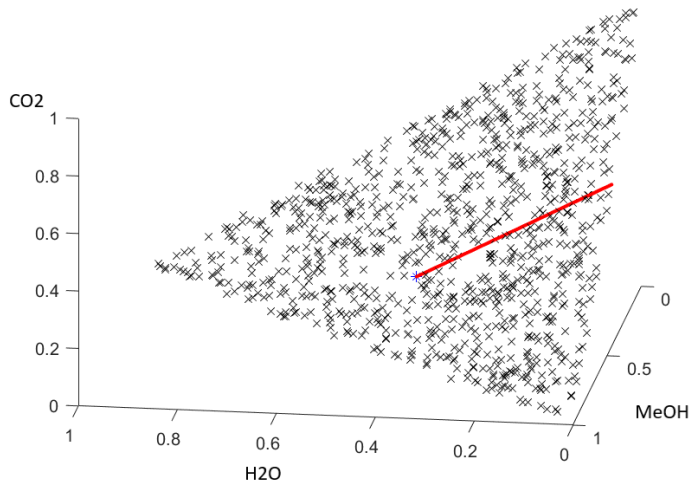


Figure 5.4: A plot showing a line with search direction $\mathbf{d}$ *(red), and said line terminating at the edge of the space of possible vectors* $\mathbf{w}$ *(the black* $\times$*). The red line shows the effect of repeatedly applying* $\mathbf{w} = \mathbf{w} + dh$*, starting from the blue* $*$ $\mathbf{w} = [0.4\ 0.4\ 0.2]$

- **Shotgun Search**: The cloud of $n = 100$ points were generated uniformly at random in the same way as with SSC.

- **Grid Search**: $n = 5$ in this case, meaning that for Mixture 1 we perform UNIRANDI $3^5 = 125$ times for Mixture 3, and 25 times for Mixture 2.

- **Pure Search**: Because our algorithm (and formulation of the *tpd*, see equation (2.38)) assumes none of the components has a composition equal to either 0 or 1, the composition of each component other than the "pure" one was set to 10^{-4}, and consequently the "pure" component had a composition of $1 - (C - 1) \cdot 10^{-4}$.

5.4.1 Notes

Because they are expected to not be scalable to six dimensions, Grid- and Shotgun Search were not applied to Mixture 3[1]. The notes on UNIRANDI in Section 5.3.1 are applicable to these methods as well.

5.5 Results and Observations

In Tables 5.5 and 5.6, the results for each of the methods on each of the mixtures is documented. Figure 5.8 presents the results more visually using bar graphs.

[1]This was tried, and acquiring results for these methods took longer than was ultimately deemed feasible.

Mixture	Method	Average Steps	Maximum Steps	Minimum Steps	Global Min Found
1.	SSC	1861	3053	1145	100%
	DIRECT	3762	3762	3762	100%
	Pure	3880	5260	2222	100%
	Shotgun	115219	118993	110031	100%
	Grid	150990	154625	146847	100%
2.	SSC	1802	145559	622	100%
	DIRECT	308	308	308	100%
	Pure	587	599	575	100%
	Shotgun	30044	31027	29789	100%
	Grid	7575	7612	7528	100%
3.	SSC	3794	7727	2337	100%
	DIRECT	39566	39566	39566	100%
	Pure	20910	22984	19668	100%
	Shotgun	×	×	×	×
	Grid	×	×	×	×

Figure 5.5: The number of steps, i.e. number of function calls to the objective function tpd, of each of the methods and mixtures, and their accuracy.

Mixture	Method	Average Time (s)	Max Time (s)	Min Time (s)
1.	SSC	405	902	102
	DIRECT	596.8	596.8	596.8
	Pure	499	676	282
	Shotgun	14564	15033	13872
	Grid	19191	19665	18680
2.	SSC	252	10890	23
	DIRECT	17.62	17.62	17.62
	Pure	30	49	28
	Shotgun	1826	2602	995
	Grid	363	751	249
3.	SSC	2758	5677	1672
	DIRECT	21672	21672	21672
	Pure	14971	16534	14059
	Shotgun	×	×	×
	Grid	×	×	×

Figure 5.6: The amount of time each method took. The time is mostly included for completeness' sake, as this might vary from device to device.

Local Minima Found	1	2	3	4	5	6	7	12	27
Times out of 100	32	4	16	33	11	1	1	1	1

Figure 5.7: Table depicting how many local minima were detected by SSC for Mixture 1.

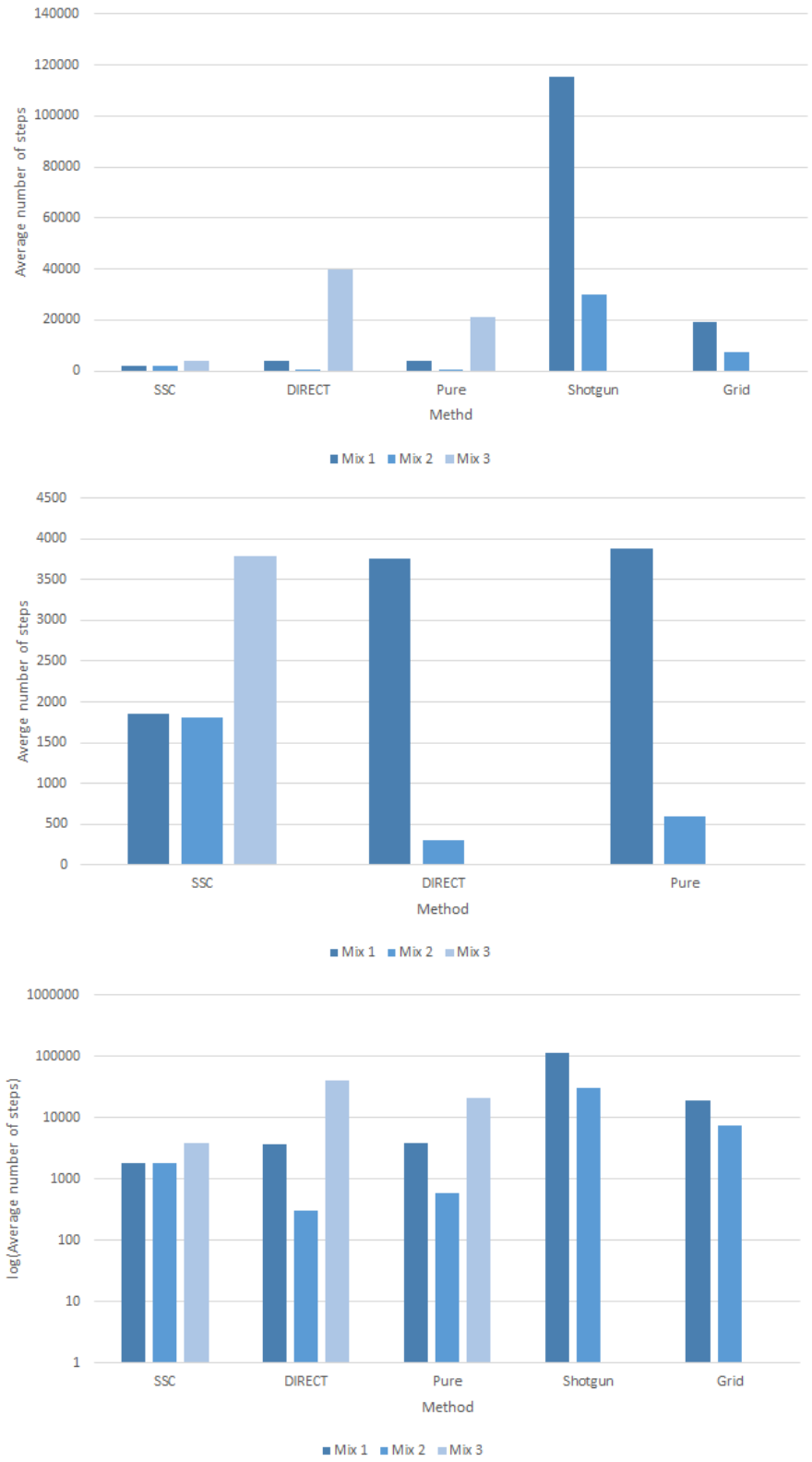


Figure 5.8: Bar graphs representing the results. Bar graphs excluding the higher numbers of evaluations and using a logarithmic scale were included for an easier visual comparison of the methods.

Looking over these results, DIRECT seems to not scale well. This is not completely unexpected, as the number of rectangles per division grows quickly with the number of dimensions.
SSC, while it in theory "promises" to find all of the local minima and form clusters around them, it is not always able to, see Table 5.7. In the case of Mixture 1. for example, almost every point $\mathbf{w} \in S$ satisfied the gradient criterion for the global minimum because of having a relatively "deep" global minimum, sometimes causing the first n points to all be clustered around the global minimum. Additionally, if the local minimum was hard to locate or particularly deep, such as in the case of Mixture 2., it is possible for more clusters to be formed than the number of local minima that exist as can be seen in Table 5.7. This is directly related to the precision of UNIRANDI. For our examples, we optimised to about 10^{-4}, which caused local minima (i.e., seed points for clusters) to not always technically be equal, causing more clusters to be formed. In rare cases, this could cause a large spike in the number of clusters formed, and thus the computation times, hence the single case of a very large number of steps and clusters for Mixture 2. This problem could be resolved by forcing a higher precision in UNIRANDI, however. On the other hand, some stability could be sacrificed for speed by using a different local optimisation method (though it would still have to be derivative free).
Finally, the stochastic local methods scale not too well, and frequently use a lot more starting points than needed, as Pure Search was able to find the global minimum by performing UNIRANDI only C times (as opposed to the large number of initial points used for Grid- and Shotgun Search).

Comparing the different methods to each other, it is immediately evident that, every method is sufficiently stable to find the global minimum 100% of the time, even the Stochastic Local methods. Thus, the quality of the methods will be compared on the basis of speed (number of function evaluations).

5.5.1 Comparison with Literature

Saber and Shaw [40] reported a number of function evaluations for DIRECT for Mixture 2 that was 6 to 10 times lower than the numbers found in this work.

This difference is could be due to Saber and Shaw having their code more optimised than the one that was used for this work, and MatLab likely being less efficient than FORTRAN, which they used. Also, their method of working in one dimension lower (see Section 5.2.1) than the number of components might also have drastically reduced the total number of function evaluations. However, their number of function evaluations for DIRECT for Mixture 3 is in the same order of size as reported here. The cause of this is unknown. Additionally, Saber and Shaw compared DIRECT to SSC for the hydrogen sulfide-methane system at $T = 190K$ and $P = 40.5 \cdot 10^6$Pa, where they showed DIRECT being 31 to 75 times faster than SSC (for different feed compositions). Sadly, this mixture proved too complicated for the flash calculation library that was used in this work [23], and thus could not be implemented or reproduced. Their SSC generally reported higher numbers of function evaluations than were found here, up to 10 times more.
The original paper on SSC [2] was applied to only two mixtures, neither of which has been included in this work.
Thus, the amount of data on DIRECT and SSC has been expanded.

We can summarise our findings as follows:

SSC is the fastest method, because of its scalability: it is about as fast for the 6-dimensional

case as DIRECT is for the 3-dimensional one.
For mixture 2., DIRECT is about 6 times faster than SSC, and Pure Search is about 3 times faster.
However, SSC is twice as fast as DIRECT and Pure Search (which are about equal) for mixture 1.
For mixture 3., SSC is about 7 times faster than Pure Search, and more than ten times faster than DIRECT.
Grid- and Shotgun Search are not competitive.
Pure Search is stable enough to always find the global minimum despite the low number of initial points.
It scales better than DIRECT, breaking even on performance with DIRECT for the mixture 1. and being about 1.5 times as fast as DIRECT for mixture 3.

Chapter 6

Conclusions and Discussion

In this chapter, conclusions are drawn from the findings in this work using the research goals and -questions from Chapter 1. It is checked whether the former have been reached, and the latter can be answered. Finally, some notes on the research and conclusions are discussed, and some possibilities for continuing this research are shared.

6.1 Conclusions

The results and findings in this work will now be compared with the research goals and -questions of Chapter 1.

6.1.1 Research Goals

> *1. Frame the VLE and the problems surrounding it in a mathematics focused way, as opposed to a more thermodynamics focused one.*

This goal was more a matter of finding and collecting, as opposed reframing what was already there. With the definitions in Chapters 2 and 3, as well as Appendices D and E, the problem has been reframed in a way that is purely about the mathematics of the problem, as opposed to one that requires the thermodynamical context and -definitions. Because this has not been clearly collected in this work thus far, a purely mathematical formulation without any of the thermodynamical context has been included in Appendix F.

> *2. Provide an overview for the more prominent optimisation techniques used in relation to flash calculations.*

This goal was achieved, and this outline can be found in Chapter 4.

> *3. Compare different optimisation techniques on their value for the modelling of VLEs.*

This goals was achieved and the comparison can be found in Chapter 5. In short, it seems that stochastic methods are better suited for solving the stability problem than deterministic ones, which was found by comparing Stochastic Sampling and Clustering to Dividing Rectangles. The choice for these methods as "representatives" for stochastic or deterministic methods was made by considering arguments in literature. See also Chapters 4 and 5.

While we did find a clear mathematical definition, the "holy grail" in the form of a new method improving on the state-of-the-art could sadly not be found. This was partially due to time constraints. One other side of this problem is that the stability problem has a clear mathematical definition. This is very nice for having a clear path to solving the problem, as can be seen in this work, but also means that solutions to this problem are "constrained" by the solution brought forth by the field of nonlinear programming, and as such a method specifically taylored to stability problems can be hard to find.

6.1.2 Research Questions

> *1. How up to date is the field of thermodynamics on the use of (mathematical) optimisation techniques, specifically in the case of flash calculations?*

Throughout this project, the focus has shifted from the overarching flash calculations to stability problems specifically as that is more suited to our mathematics-focused approach (see also Chapter 3). Contrary to the hypotheses at the start of this project (see Chapter 1), literature on the stability problem seems to be fairly up to date. The literature on the flash problem specifically is still relatively old as mentioned in Chapter 1, perhaps because others have also decided that improving the problem with flash calculations can be done through the improving of other methods, such as solving the stability problem. Aside from the methods mentioned in Chapter 4, other methods have been attempted and compared to each other by other researchers, and most of the more prominent methods (such as Branch and Bound and Simulated Annealing) have been applied to the stability problem. Thus, the state-of-the-art for the stability problem seems fairly up to date, though a deeper dive into quadratic programming methods could be taken.

> *2. Can the stability- and accuracy problems of flash calculations be improved using state-of-the-art (NLP-)optimisation techniques?*

This question is answered in a less direct way. As discussed, finding the global minimum of the *tpd* can be done using optimisation techniques such as mentioned in Chapter 4, improving the accuracy of stability tests, which in turn would improve the accuracy of flash calculations, and consequently the accuracy and stability of dynamic simulations. It is no panacea for the problems associated with the dynamic simulations of VLEs, but certainly offers an improvement.

> *3. Can the stability- and accuracy problems of flash calculations be improved when approached from an almost purely mathematical point of view, as opposed to a more thermodynamics-oriented one?*

As the problem can be formulated using only language from mathematics, and as a consequence of Research Question 2., this is an option, and possibly even the best approach for the problem from the side of stability problems, though the "perfect" solution likely lies somewhere in between: a tailored method inspired by thermodynamics and implemented using mathematics (that is mostly conjecture, however).

6.1.3 Further Conclusions

As was seen in Chapter 3, current theories (equations of state in combination with e.g. tangent plane analysis) in combination with modelling software is fairly accurate in predicting (theoretical) behavior of mixtures. This however, only holds provided that the number of phases and the "logical" feed phase are known beforehand. It has also been observed that the properties of a mixture (such as the global minimum to the tangent plane distance) are harder to determine for a combination of P and T close to the critical point. On a more abstract level, this is because solutions to phase equilibrium equations look similar to the trivial solution. Also relevant for dynamic simulations where the P and/or T can vary, the surface of the *tpd* seems to shift fairly gradually when the temperature and pressure are shifted only slightly [1] , *except* when the mixture shifts from a more liquid-like to a more vapor-like substance (even more so than when going from the two-phase region to the single-phase region), at least for the [0.4 0.4 0.2] H_2O-CH_4O-CO_2 mixture.
As can be seen from Chapters 3 and 5, on a more concrete level this is a consequence of near-critical mixtures having global minima that are relatively shallow (close to zero) in surfaces that include local minima and discontinuities.
Going off of the results of Chapter 5, it seems that the hypothesis in Chapter 4 that a stochastic method would work better than a deterministic one was fairly correct. For low dimensional cases (2 or 3 components), finding the global minimum can be done fairly quickly regardless of the method, but a deterministic method such as DIRECT performs rather fast for these cases, 2 to 6 times faster than the stochastic method (contrary to what we hypothesised). Repeated local search can also be quite efficient in these cases (when performed from less than 10 initial points). However, for the higher dimensional (and perhaps more interesting) methods the stochastic method shows a better scalability, performing more than 10 times as well for $C = 6$. From this, we conclude that, as often with complicated problems like these, there is no "one size fits all" method to find the global minimum of the *tpd* in all cases However, a stochastic method seems suitable in its absence.

6.2 Discussion

There is still improvements to be made both on the thermodynamics- and the mathematics sides of the problem. Not only is the efficiency of the optimisation methods dependent on the number of components in the mixture and the interaction between the components of the mixture[2], but the chosen equation of state can have quite the effect on which method to apply and how to formulate it. For quadratic equations of state such as SRK and Peng-Robinson, the optimisation methods work comparably, but when more complicated (but more physically accurate) equations

[1] For the [0.4 0.4 0.2] H_2O-CH_4O-CO_2 mixture, changes around $5{\cdot}10^5$Pa in P and 5K in T are small enough.
[2] Even for cases with a small number of components, problems can be quite complex, see for example the 2 component H_2S-CH_4 mixture [2].

are used, such as SAFT, a direct formulation of ϕ would no longer be feasible, among other things. Another thing to note is that solutions to Gibbs Energy minimisation/tpd minimisation are only candidate solutions, because the number and kind of phases present in equilibrium must be assumed beforehand to formulate the stability problem.

6.3 Suggestions for Future Work

This is not the first research on stability tests and ways of improving the stability of dynamic simulations, and it is not expected to be the last. Below some recommendations for possible future research directions are given.
In Chapter 4 and 5, we have discussed that the choice of local optimisation method is important for many (all but DIRECT) of the tested methods. To further study the methods with e.g. a faster but less stable local optimiser, other local optimisation methods than UNIRANDI could be tested For example, Quasi-Newton with a David-Fletcher-Powell update formula can be tried which is faster than UNIRANDI but more stable. As UNIRANDI was sufficiently stable even for repeated local optimisation, testing a less stable method could prove interesting.
Additionally, the field of nonconvex- and quadratic programming in optimisation is still expanding, and improvements in this field should be studied for VLE modelling to stay up to date.
Also interesting is expanding stability tests beyond VLEs, or research methods that are able to predict what kind of equilibrium will be formed. In this work only mixtures attaining a Vapor-Liquid Equilibrium have been considered, but in practice some mixtures under the right P and T might achieve a Liquid-Liquid Equilibrium, an equilibrium containing at least one solid, or more than two phases. These equilibria have been studied less extensively than VLEs. The results of performing calculations on different types of equilibria could force a more generally applicable method to be required, such as methods that are able to predict the kind of equilibrium.
In this case, quadratic equations of state (such as SRK) might no longer be sufficient for the modelling, which leads us to our next point. While we have mentioned that we expect the performance of the methods to be roughly equal for quadratic equations of state (see Chapter 2), and in the methods discussed in Chapter 5 any type of equation of state can be used to calculate the fugacity, it could be interesting to see if methods can be devised that incorporate an equation of state more directly (aside from calculating the objective function). Using more accurate equations of state (like e.g. SAFT) could enable a more "direct" minimisation of the state functions by staying in a more physics-focused realm, instead of "translating" to mathematics.

Chapter 7

Bibliography

[1] L.E. Baker, A.C. Pierce, K.D. Luks.
"Gibbs Energy Analysis of Phase Equilibria".
Tulsa, Oklahoma.
1981.

[2] J. Balogh[1], T. Csendes[2], R.P. Stateva[3].
"Application of a Stochastic Method to the Solution of the PHase Stability Problem: Cubic Equations of State".
[1] Department of Computer Science, Juhasz Gyula Teachers' Training Colelge, University of Szeged, Szeged, Hungary. [2]Department of Applied Informatics, University of Szeged, Szeged, Hungary. [3]Insittue of Chemical Engineering, Bulgarian Academy of Sciences, Sofia, Bulgaria.
2003.

[3] B. Bánhelyi, T. Csendes, B. Lévai, L. Pál, D. Zombori.
"The GLOBAL Optimization Algorithm: Newly Updated With Java Implementation and Parallelization".
Department of Computational Optimization, University of Szeged, Szeged, Hungary.
2018.

[4] C.G.E. Boender[1], A.H.G. Rinnooy Kan[1], G.T. Timmer[1] and L. Stougie[2].
[1]Econometric Institute, Erasmus University, Rotterdam, The Netherlands; [2] Mathematical Centre Amsterdam, The Netherlands.
"A Stochastic Method for Global Optimization".
1981.

[5] A. Bonilla-Petriciolet[1], J.G. Segovia-Hernández[2].
[1]Instituto Tecnológico de Aguascalientes, Chemical Engineering Deprtment, Aguascalientes, Mexico.[2] Universidad de Guanajuato, Chemical Engineering Department, Guanajuato, Mexico.
"A Comparative Study of Particle Swarm Optimization and its Variants for Phase STability and Equilibrium Calculations in Multicomponent Reactive and Non-Reactive Systems".
2010.

[6] A. Bonilla-Petriciolet, R. Vázquez-Román, G.A. Iglesias-Silva, K.R. Hall.
"Performance of Stochastic Global Optimization Methods in the Calculation of Phase Stability Analyses for Nonreactive and Reactive Mixtures".

Departamento de Ingenieria Quimica, Instituto Tecnoloógica de Aguascalientes, Aguascalientes, Mexico.
2006.

[7] W.G. Chapman[1,2], K.E. Gubbins[1], G. Jackson[1], M. Radosz[2].
"New Reference Equation of State for Associating Liquids".
[1] Cornell University, School of Chemical Engineering Ithaca, NY, USA and [2] Exxon Research and Engineering Company,. Corporate Research, Annanale, NJ, USA.
1990.

[8] W.G. Chapman[*,**], K.E. Gubbins[*], G. Jackson[*], M. Radosz[**].
"SAFT: Equation-of-State Solution Model for Associating Fluids".
[*] Cornell University, School of Chemical Engineering Ithaca, NY, USA and [**] Exxon Research and Engineering Company, Corporate Research, Annanale, NJ, USA.
1989.

[9] T. Csendes.
"Nonlinear Parameter Estimation by Global Optimization - Efficiency and Reliability".
University of Szeged.
2005.

[10] Dortmünd Data bank.
`http://www.ddbst.com/ddb.html`

[11] V. Gaganis, D. Marinakis, N. Varotsis.
"A General Framework of Model Functions for Rapid and Robust Solution of Rachford-Rice Type of Equations".
Mineral Resrouces Engineering Department, Technical Univerity of Cretea, Greece.
2012.

[12] R.L.Graham.
"An Efficient Algorith for Determining the Convex Hull of a Finite Planar Set."
Bell Telephone Laboratories, Incorporated Murray Hill, New Jersey, USA.
1972.

[13] S.T. Harding, C.A. Floudas.
"Phase Stability with Cubic Equations of State: Gobal Optimization Approach".
Dept of Chemical Engineering, Princeton Universtiy, Princeton.
2000.

[14] N. Henderson, L. Feitas, G.M. Platt
"Prediction of Critical Points: A New Methodology Using Global OPtimization".
Thermodynamics and Optimization Group, Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Brazil.
2004.

[15] S. Horstmann, A. Jabloniec, J. Krafczyk, K. Fischer and J. Gmehling.
"PSRK group contribution equation of state: comprehensive revision and extension IV, including critical constants and α-function parameters for 1000 components."
Fluid Phase Equilibria 227.
2005.

[16] B.B. Ivanov, A.A. Galushko, R.P. Stateva
"Phase Stability Analysis with Equations of State - A Fresh Look from a Different Perspective".
Institute of Chemcial Engineer, Bulgarian Academy of Sciences, Sofia, Bulgaria.
2013.

[17] F. Jalali[1], J.D. Seader[2], S. Khaleghi[1].
"Global Solution Approaches in Equilibirium and Stability Analysis Using Homotopy Continuation in the Complex Domain."
[1]Chemical Engineering Department, University of Tehran, Tehran. [2]Chemical Engineering Department, University of Utah, Salt Lake City.
2007.

[18] D.R. Jones[1], C.D. Perttunen[2], B.E. Stuckman[2].
"Lipschitzian Optimization Without the Lipschitz Constant."
[1]: General Motors Research and Development Center, Warren, Michigan & [2] Brooks and Kushman, Department of Patent and Computer Law, Southfield, Michigan.
1993.

[19] N.S. Junior[1], N.S.M. Cardozo[1], A.R. Secchi[2], K. Wada[1].
"Adaptive Random Search: A Promising Method for Determining the Stability of Mixtures".
[1]Department of Chemical Engineer, Universidade Federal do Rio Grande, Porto Alegre. [2] Horácio Macedo, Centro de Tecnologia, Rio de Janeiro.
2009.

[20] Kaylaiacovino
`http://www.kaylaiacovino.com/Petrology_Tools/Critical_Constants_and_Acentric_Factors.htm`

[21] Y. Li[1], R.T. Johns[2], K. Ahmadi[1].
"A Rapid and Robust Alternative to Rachford-Rice in Flash Calculations".
[1]: University of Texas at Austin, United States and [2] : The Pennsylvania State University at University Park, United States.
2011.

[22] N. Lindeloff, M.L. Michelsen.
Technical University of Denmark.
"Phase Envelope Calculations for Hydrocarbon-Water Mixtures."
2002.

[23] Á. Martín, M.D. Bermejo, F.A. Mato, M.J. Cocero.
"Teaching Advanced Equations of State in Applied Thermodynamics Courses Using Open Source Programs".
Department of Chemical Engineering and Environmental Technology, University of Valladolid, Valladolid, Spain.
2011.

[24] C.M. McDonald and C.A. Floudas.
"Global Optimization for the Phase Stability Problem".
Dept. of Chemical Engineering, Princeton University, Princeton, NJ 08544.
1995.

[25] M.L. Michelsen.
"The Isothermal Flash Problem. Part I. Stability".
Instituttet for Kemiteknik, Danmarks Tekniske Højskole, Byning 229, 2800 Lyngby.
1981.

[26] M.L. Michelsen.
"The Isothermal Flash Problem. Part II. Phase-Split Calculation".
Instituttet for Kemiteknik, Danmarks Tekniske Højskole, Byning 229, 2800 Lyngby.
1981.

[27] M.L. Michelsen
"Phase Equilibrium Calculations. What is Easy and What is Difficult?"
Institut for Kemiteknik, Bygning 229, Danmarks Tekniske Højskole, Lyngby, Denmark
1992

[28] M.L. Michelsen, J.M. Mollerup.
"Thermodynamic Models: Fundamentals & Computational Aspects".
Holte, Denmark.
2004.

[29] L.X. Nghiem, K. Aiz, Y.K. Li.
Society of Petroleum Engineers Journal, Computer Modelling Group.
"A Robust Iterative Method for Flash Calculations Using the Soave-Redlich-Kwong or the Peng-Robinson Equation of State".
1983.

[30] D.V. Nichita[1], F. García-Sánchez[2], S.Gomez[3].
"Phase Stability Analysis Using the PC-SAFT Equation of State and the Tunneling Global Optimization Method".
[1]Laboratoire des Fluides Complexes, Université de Pau et es Pays de l'Adour, Pau Cedex. [2]Laboratorio de Termodinámica, México. [3]Instituto de Investigacion en Matematicas Aplicadas y Systemas, Apdo.
2007.

[31] D.V. Nichita[1,2], S. Gomez[3].
"Efficient Location of Multiple Global Minima for the Phase Stability Problem". [1]Laboratoire des Fluides Complexes, Université de Pau et es Pays de l'Adour, Pau Cedex. [2]Instituto Mexicano del Petroleo, Madero. [3]Instituto de Investigacion en Matematicas Aplicadas y Systemas, Apdo.
2009.

[32] U. Olsbye, S. Svelle, M. Bjorgen,P. Beato, T.V.W. Janssens, F. Joensen, S. Bordiga, K.P. Lillerud.
"Conversion of Methanol to Hydrocarbons: How Zeolite Cavity and Pore Size Controls Product Selectivity".
Angew. Chem. Int. Ed.
2012.

[33] F.M.Orr Jr.
"Theory of Gas Injection Processes".
Stanford University, Stanford, California.
2005.

[34] A. Péneloux[1], E. Rauzy[1], R. Frèze[2].
"A Consistent Correction for Redlich-Kwong-Soave Volumes".
[1]Faculté des Sciences de Luminy, Marseille and [2]Faculté de Sciences et Techniques de St Jérôme, Marseille.
1981.

[35] D.Y. Peng, D.B. Robinson.
"A New Two-Constant Equation of State."
Industrial and Engineering Chemistry: Fundamentals.
1976.

[36] B.E. Poling, J.M. Prausnitz, J.P. O'Connell.
"The Properties of Gases and Liquids", fifth edition.
McGraw-Hill, New York.
2000.

[37] I. Rahman[1], A.K. Das[2], R.B. Mankar[2], B.D. Kulkarni[1].
"Evaluation of Repulsive Particle Swarm Method for Phase Equilibrium and PHase Stability Problems".
[1]Chemical Engineering and Process Development Division, National Chemical Laboratory India. [2] Department of Petrochemical Technology, Laximinarayan Institute of Technology, Nagpur, India.
2009.

[38] G.P. Rangaiah.
"Evaluation of Genetic Algorithms and Simulated Annealing for Phase Equilibrium and Stability Problems.".
Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore.
2001.

[39] O. Redlich, J.N.S. Kwong.
"On the thermodynamics of Solutions. An equation of State. Fugacities of Gaseous Solutions."
Shell Development Company, emeryville, California.
1948.

[40] N. Saber, J.M. Shaw.
"Rapid and Robust Phase Behaviour Stability Analysis Using Global Optimization".
Dept of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta.
2007.

[41] E. Schechter.
`https://math.vanderbilt.edu/schectex/courses/cubic/`
'The Cubic Formula (Solve Any 3rd Degree Polynomial Function)'.

[42] J. Seader, E. J. Henley, and D. K. Roper.
"Separation Process Principles.".
Hoboken: John Wiley & Sons, fourth ed.
2016.

[43] B. Shubert.
"A Sequential Method Seeking the Global Maximum of a Function.".

Department of operations Analysis, Naval Postgraduate School, Monterey, California.
1972.

[44] G. Soave.
"Equilibrium constants from a modified Redlich-Kwong equation of state."
Chemical Engineering Science.
1972.

[45] M. Srinivas, G.P. Rangaiah.
"A Study of Differential Evolution and Tabu Search for Benchmark, Phase Equilibrium and Phase Stability Problems".
Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore.
2006.

[46] R.P. Stateva[a], G.S. Cholakov[b], A.A. Galushko[a], W.A. Wakeham[c].
"A Powerful Algorithm for Liquid-Liquid-Liquid Equilibria Predictions and Calculations".
[a] : Institute of Chemical Engineering, Bulgarian Academy of Sciences, Bulgaria, [b] : Department of Petroleum and Solid Fuels, University of Chemical Technology and Metallurgy, Bulgaria, [c] : Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, London.
2000

[47] R.P. Stateva, S.G.A. Tsvetkov.
"A Diverse Approach for the Solution of the Isothermal Multiphase Flash Problem. Application to Vapor-Liquid-Liquid Systems".
Department of Chemical Engineering and Chemical Technology, Imperial College of Science,Technology and Medicine, London.
1994.

[48] R.P. Stateva, W.A. Wakeham.
"Phase Equilibrium Calculations for Chemically Reacting Systems".
Department of Chemical Engineering and Chemical Technology, Imperial College of Science,Technology and Medicine, London.
1997.

[49] A.C. Sun, W.D. Seider.
"Homotopy-Continuation Method for Stability Analysis in the Global Minimization of the Gibbs Free Energy".
Department of Chemical Engineering, University of Pennsylvania, Philadelphia.
1995.

[50] C. Tsonopoulos, J.M. Prausnitz.
Cryogenics Volume 9 issue 5.
"A Review for Engineering Applications".
1969.

[51] C.H. Twu, J.E. Coon, D. Bluck.
"Comparison of the Peng-Robinson and Soave-Redlich-Kwong Equations of State Using a New Zero-Pressure-Based Mixing Rule for the Prediction of High-Pressure and High-Temperature Phase Equilibria.".
Simulation Scienes Inc. Brea, California.
1998.

[52] J.O. Valderrama[1], S. Obaid-Ur-Rehman[2].
"Application of a New Cubic Equation of State to Hydrogen Sulfide Mixtures".
[1]Chemical Engineering Department, King Fahd University of Petroleum & Minerals, Dahran, Saudi Arabia and [2]Departamento de Ingenieria Quimica, Universidad del Norte, Antofagasta, Chile.
1987.

[53] J.D. van der Waals.
"Over de Continuiteit van den Gas- en Vloeistoftoestand (On the Continuity of the Gas and Liquid State".
Leiden, the Netherlands.
1873.

[54] S.M. Walas.
"Phase Equilibria in Chemical Engineering".
Butterword-Heineman, Newton MA.
1985.

[55] G.M. Wilson.
"A Modified Redlich-Kwong Equation of State, Application to General Physical Data Calculations".
65th Nationnal AlChE Meeting, Cleveland OH.
1969.

[56] G. M. Wilson.
"Vapor-Liquid Equilibrium XI. A New Expression for the Excess Free energy of Mixing".
Department of Chemistry, Massachusetts Institute of Technology, Cambridge 39 Massachusetts.
1963.

[57] H. Zhang[1], A. Bonilla-Petriciolet[2], G.P. Rangiah[1].
"A Review on Global Optimization Methods for Phase Equilibrium Modeling and Calculations".
[1]Department of Chemical & Biomolecular Engineering, Singapor, [2]Instituto Tecnológico de Aguascalientes, Departamento de Ingenieria Quimica, Aguascalientes, Mexico.
2011.

[58] *Excerpt from a book by the NTNU in Trondheim. Actual reference details could not be found, but the pdf can be found by looking for "Flash English Edition 2009 NTNU" at the time of writing this work.*

Appendices

Appendix A

List of Symbols

a. Attraction parameter in the equation of state.

a_i. Attraction parameter in the equation of state for mixtures.

a_{ij}. Combined attraction parameter in the equation of state for mixtures.

α. A parameter in the equation of state, dependent on the acentric factor ω.

α (SSC). A parameter of the clustering procedure. See also section 5.3.1.

b. Repulsion parameter in the equation of state.

b_i. Repulsion parameter in the equation of state for mixtures.

b_{ij}. Combined attraction parameter in the equation of state for mixtures.

β. The vapor fraction, the amount of vapor in relation to the amount of liquid.

C. The total number of components in the mixture.

CH_4. Methane.

CH_4O. Methanol.

C_2H_6. Ethane.

C_3H_8. Propane.

C_4H_{10}. n-butane.

C_5H_{12}. $n - Pentane$.

C_i. Component i in the mixture.

CO_2. Carbon Dioxide.

δ (DIRECT). One third of a rectangle length. Also δ_j when referring to rectangle j.

δ_i. Parameter with which different equations of state can be defined from the general formula $\frac{P}{RT} = \frac{1}{v-b} - \alpha\frac{b}{(v+\delta_1 b)(v+\delta_2 b)}$.

f. The fugacity, sometimes f_i (fugacity of component i) or $f^{(w)}$ (fugacity of phase w).

ϕ. The fugacity coefficient, sometimes ϕ_i (fugacity coefficient of component i), $\phi^{(w)}$ (fugacity coefficient of phase w) or $\phi_i(\mathbf{w})$ (the fugacity coefficient of component i in the phase associated to the composition vector $\mathbf{w}$). $\phi = \frac{f}{P}$.

g^E. Gibbs Energy. $g^E = g^{id} + g^{excess}$ For an ideal liquid $g^{id,L} = \sum_i x_i \ln x_i$, and for an ideal

vapor $g^{id,V} = \sum y_i \ln y_i$

γ. Decision variable used to transform the tangent plane distance function, see Section 2.5.

γ (SSC). The percentage of points with lowest objective value from the n sampled points that will be clustered in SSC.

H_2O. Water.

K. $K_i = \frac{y_i}{x_i}$, the **K**−factor, or distribution ratio.

κ. A parameter in the Peng-Robinson equation of state, dependent on ω.

M. Molar mass, sometimes M_i (molar mass of component i).

MeOH. Methanol.

μ_i The chemical potential of component i.

n. The number of points sampled in one go.

n'. In Stochastic Sampling and Clustering, the total number of points sampled (a multiple of n).

N_2. Nitrogen.

ω. The acentric factor, a theoretical property useful when performing calculations involving substances. Sometimes ω_i, the acentric factor of component i.

P. Pressure, in Pascal (Pa).

P_c. Critical pressure, pressure of a substance at its critical point. Sometimes P_{ci}, the critical temperature of component i in the mixture.

P_r. Reduced pressure, $P_r = \frac{P_c}{P}$.

R. The ideal gas constant,
$R \approx 8.31446261815324 \mathrm{m}^3 \cdot \mathrm{Pa} \cdot \mathrm{K}^{-1} \cdot \mathrm{mol}^{-1}$.

ρ. Density.

tpd. Tangent plane distance.

tpd. The tangent plane distance function. The objective function that is to be minimised for solving the stability problem.

T. Temperature, in Kelvin (K).

T_c. Critical temperature, temperature of a substance at its critical point. Sometimes T_{ci}, the critical temperature of component i in the mixture.

T_r. Reduced temperature, $T_r = \frac{T_c}{T}$.

v. The molar volume, $v = \frac{\sum_{i=1}^{N} x_i M_i}{\rho_{mixture}}$

V. Volume, in m^3.

V_c. Critical volume, volume of a substance at its critical point. Sometimes V_{ci}, the critical temperature of component i in the mixture.

V_r. Reduced volume, $V_r = \frac{V_c}{V}$.

w. The composition vector associated with a test phase (e.g. of the new phase being formed when doing a stability test), as a vector of fractions of length C, the number of components, $0 \leq w_i \leq 1$. The vector sums to 1.

x. The composition vector associated with the liquid phase, as a vector of fractions of length C, the number of components, $0 \leq x_i \leq 1$. The vector sums to 1.

y. The composition vector associated with the vapor phase, as a vector of fractions of length C, the number of components, $0 \leq y_i \leq 1$. The vector sums to 1.

z. The composition vector associated with the (feed) mixture, as a vector of fractions of length C, the number of components, $0 \leq z_i \leq 1$.

Z. The compressibility factor, sometimes Z_i (compressibility of component i) or $Z^{(w)}$ (compressibility of phase w). Z is between 0 and 1, a measure of how much a substance differs from an ideal gas.

Appendix B

List of Terms

Acentric Factor (ω). A measure of the "non-sphericity" (centricity) of molecules. The higher ω is, the higher the boiling point.

Attraction parameter (a). A parameter in an equation of state used to model the attraction between molecules in a mixture or substance. F **Bubble point.** The point where a (multicomponent) liquid starts to boil/evaporate (point when the first vapor bubble is formed). For a single component liquid (e.g. pure water), the bubble point is equal to the dew point, and it is called the boiling point.

Chemical Potential (μ). Energy that can be absorbed or released due to a change in the number of particles of a given substance, such as in the event of a chemical reaction or phase transition. When both T and P are set contant, the chemical potential is equivalent to the molar Gibbs free energy. At phase equilibrium, we have $\sum_i \mu_i = 0$, as the free energy is at a minimum.

Compressibility Factor (Z). A factor which describes how strongly a gas deviates from an ideal gas. It is defined as the ratio of the molar volume of a gas to the molar volume of an ideal gas at the same temperature and pressure. In general, a gas will deviate more from an ideal one the closer it is to a phase change, the lower the temperature and the higher the pressure. Z can be obtained from e.g. an EOS. The closer to 1, the less it differs.

$$Z = \frac{PV}{RT}$$

Corresponding States Principle (CSP). Also called the Law of Corresponding States. It states that dimensionless properties of all substances will follow universal variations of suitably dimensionless variables of state and other dimensionless quantities. In other words, it states that for all fluids, if they have the same temperature and pressure, they will also have the same compressibility, i.e. that all fluids differ from their ideal version with the same factor.

Critical point (critical state). The end point of a phase equilibrium, a specific combination of pressure and temperature before which a liquid and its vapor can coexist.

Critical region. The region close to the critical point. What "close to" means is context (and mixture) dependent.

Dew point. The point where a (multicomponent) vapor starts to dew/condensate (when the first liquid droplet is formed). For a single component vapor (e.g. pure water steam), the dew point is equal to the bubble point, and it is called the boiling point.

Enthalpy. A thermodynamic system's internal energy plus the product of its pressure and volume. In closed systems with processes at constant pressure, the absorbed/released heat equals the change in enthalpy.

Entropy. A measure of disorder/randomness/irreversibility/energy dispersal in a system. Never

decreases for an irreversible process in an isolated system. It expresses the number of different configurations a system can assume. Another way to look at it: when a system is too hard to look at on a micro scale (position, momentum, kinetic energy, etc.), we look at it on a macro scale (in terms of pressure, temperature, density, etc.). Entropy is then a measure of how much information we lose in going from micro to macro. the more exact information you can use to describe a system, the smaller your entropy.

Equation of State. An algebraic relation between (for example) P, V and T, used to model the behavior of a mixture under certain PVT conditions.

Equilibrium, phase. A state in which nothing changes. For example, when a compound is in a phase-equilibrium (between vapor and liquid), there is no change in the number of vapor- and liquid molecules (every moment an equal number of molecules go from liquid to vapor phase as the other way around).

Fugacity. An effective partial pressure which replaces the mechanical partial pressure in an accurate computation of the chemical equilibrium constant. It is equal to the pressure of an ideal gas with the same temperature and enthalpy. Fugacities are usually determined experimentally or estimated from models such as the van der Waals equation. Fugacity is closely related to thermodynamic activity. Using fugacity instead of pressure in calculations lets you "pretend" that a real gas behaves like an ideal gas. The fugacity and the compressibility factor Z are linked via

$$f = P \exp \int \frac{Z-1}{P} dP$$

The **fugacity coefficient** $\phi = \frac{f}{P}$ is often used instead of the "regular" fugacity.

Gibbs (Free) Energy. See enthalpy.

Ideal Gas. A strictly theoretical gas in which the particles are points who do not interact outside of perfectly elastic collisions (collisions where the total kinetic energy stays the same). It considerably easier to perform calculations and theoretical predictions on an Ideal Gas thana non-ideal one.

Interaction Parameter (Binary-). Parameters that give a measure of interactivity between substances in a mixture. They help model the "non-ideality" of a mixture (as in a perfect mixture, there is zero interaction between any two components). These depend on the substances involved and the model being used.

Mixing Rules. Not as much a rule as a weighted mean used to predict properties of mixtures. It allows for combining e.g. equations of state for single components into ones for mixtures.

Molality. A measure of the concentration of a chemical species (solute in a solution). It is in terms of amount of substance per mass volume (e.g moles per kg, mol/kg). This is in contrast with molarity, which looks at the concentration in terms of *volume*.

Molarity. A measure of the concentration of a chemical species (solute in a solution). It is in terms of amount of substance per unit volume (e.g. moles per litre, mol/L). This is in contrast with molarity, which looks at the concentration in terms of *mass*.

Mole. The unit of measurement for the amount of substance present, equal to $6.02214076 \cdot 10^{23}$ particles (atoms, molecules, ions or electrons). From this, we get the definitions of **molar mass** (the mass of one mole of substance) and **molar volume** (the volume occcupied by one mole of substance).

Phase. The physically distinctive form a chemical species takes, e.g. solid, liquid or gas.

Pure substance. Not a mixture, consisting of only a single substance.

Rachford-Rice (Equation).

$$\sum_i \frac{z_i(K_i - 1)}{1 + \beta(K_i - 1)} = 0$$

with z_i the mole fraction of the component i in the feed liquid (assumed to be known), β the fraction of the feed that is vaporised, and $K_i = \frac{y_i}{x_i}$ the equilibrium constants. It is solved for β in flash calculations, from which we can calculte x and y as follows:

$$x_i = \frac{z_i}{1 + \beta(K_i - 1)}$$
$$y_i = K_i x_i$$

Multiple solutions for β are possible, but only one guarantees that x_i and y_i are positive. If there is multiple solutions for β for which x_i and y_i are positive, then $K_{max} < 1$ or $K_{min} > 1$, thus there is either no gas phase ($\beta = 0$) or no (liquid)f phase ($\beta = 1$).

Repulsion parameter (b). A parameter in an equation of state used to model the repulsion between molecules in a mixture or substance, use.

Saturation Pressure (Vapour Pressure). The pressure of a vapour in equilibrium with its solid or liquid. Also, the maximum pressure possible by vapour at a given temperature.

Species (chemical). A chemical substance, more accurately a collection of atoms or molecules of a substance.

SRK. Soave-Redlich-Kwong, a quadratic equation of state. See also Section 2.2.3.

Supercritical region. The region of combination of P and T (or P and V) where the points lie above the critical point ($P > P_c$, $T > T_c$).

State Function. A mathematical expression which defines the value of a state property (such as pressure, temperature, volume, etc).

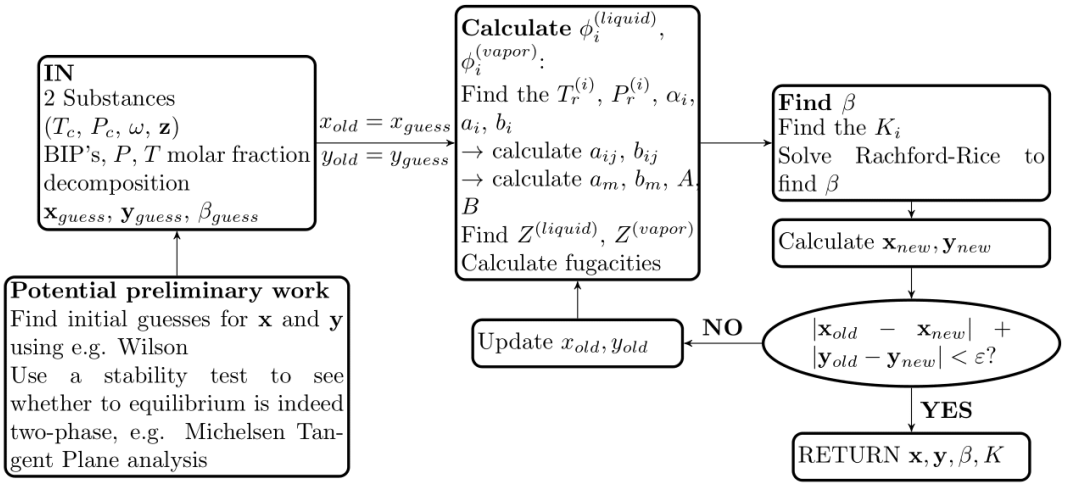
Trial phase composition (w). The composition of a theoretical phase. Used for example to calculate the Gibbs Energy if a new phase with composition **w** would be formed.

Vapor. Different word for gas.

Vapor-liquid-equilibrium (VLE). Describes the distribution of a species between the vapor- and liquid phase when in equilibirum. In non-pure systems, often fugacity is employed.

Appendix C

Block Diagram of Flash



Equations and Variables

$i \in \{1, 2\}$

$T_r^{(i)} = \frac{T}{T_c^{(i)}}$, $P_r^{(i)} = \frac{P}{P_c^{(i)}}$

$\alpha_i = (1 + (0.485 + 1.551\omega - 0.1566\omega^2)(1 - \sqrt{T_r}))^2$

$a_i = 0.427\frac{R^2(T_c^{(i)})^2}{P_c^{(i)}}$, $b_i = 0.087\frac{RT_c^{(i)}}{P_c^{(i)}} \cdot \alpha_i$

$a_{ij} = (1 - k_1^{(i,j)})\sqrt{a_i a_j}$, $(1 - k_2^{(i,j)})\sqrt{b_i b_j}/2$

$a_m = \sum_{p,q} x_p x_q a_{pq}$, $b_m = \sum_{p,q} x_p x_q b_{pq}$

$A = \frac{a_m P}{R^2 T^2}$, $B = \frac{b_m P}{RT}$

$\bar{Z} = ReRoots(Z^3 - Z^2 + (A - B - B^2)Z - AB$,

$Z^{(liquid)} = \min \bar{Z}$, $Z^{(vapor)} = \max \bar{Z}$

Let $\Sigma_i = \sum_k x_k a_{k,i}$

$\phi_i^{(phase)} = \exp \frac{b_i}{b_m}(Z^{(phase)} - 1) - \log(Z^{(phase)} - B) - \frac{A}{B}(\frac{2\Sigma_i}{a_m} - \frac{b_i}{b_m})\log(\frac{1+B}{Z^{(phase)}})$

$K_i = \frac{\phi_i^{(liquid)}}{\phi_i^{(vapor)}}$

Rachford-Rice: solve for β: $\sum_i \frac{z_i(K_i - 1)}{1 + \beta(K_i - 1)} = 0$, where $z_i = (1 - \beta)x_i + \beta x_i$ is the mole fraction of component i in the mixture (both liquid- and vapor phase)

$x_{new}^{(i)} = \frac{z_i}{1 + \beta(K_i - 1)}$, $y_{new}^{(i)} = K_i x_i$

What could go wrong?
When finding $Z's$: when $Z^{(phase)} \leq B$, $\log(Z - B)$ is complex.
When $P = 0$, $A = B = 0 \Rightarrow \frac{A}{B} = NaN$.
When $T = 0$, $A = B = \infty \Rightarrow Z$ cannot be solved for.
NOTE: Rachford-Rice always has exactly one solution when there is a two-phase equilibrium (Gaganis, 2012) .

Where

- T_r, P_r the reduced temperature and pressure, respectively, the fraction of the critical temperature T_c to the temperature T (and likewise for P).
- ω the acentric factor, a property of the substance
- $\mathbf{z}$ the feed composition.
- BIP the binary interaction parameters, which represent the interaction between the two substances.
- $\mathbf{x}_{guess}, \mathbf{y}_{guess}, \beta_{guess}$ the initial guesses for $\mathbf{x}, \mathbf{y}, \beta$.
- $\mathbf{x}_{old}, \mathbf{y}_{old}$ the values of $\mathbf{x}, \mathbf{y}$ in the previous iteration.
- $\mathbf{x}_{new}, \mathbf{y}_{new}$ the values of $\mathbf{x}, \mathbf{y}$ calculated at the end of the current iteration.
- $\phi_i^{(liquid)}, \phi_i^{(vapor)}$ the fugacity coefficients for component i in the liquid- or vapor phase, respectively.
- $T_r^{(i)}, P_r^{(i)}$ the reduced temperature and pressure for component i, $T_r^{(i)} = \frac{T_c^{(i)}}{T}$ and likewise for P.
- $\alpha_i, a_i, b_i, a_{ij}, b_{ij}, a_m, b_m, A, B$ parameters of the equation of state
- $Z^{(liquid)}, Z^{(vapor)}$ the mixture compressibility factors for the liquid- or gas phase, measures of how much a substance (in a specific phase) differs from an ideal gas.
- β the vapor fraction, the fraction of total amount of substance that is in vapor phase.
- K (K_i), the distribution ratio (of component i), $K = \frac{y}{x}$.

Appendix D

Fugacity Coefficient

For the SRK equation of state:

$$\ln \phi_i(\mathbf{w}) = \frac{b_i}{b^{(m)}}(Z-1) - \log(Z-B) - \frac{A}{B}\left(2\frac{\sum_j w_j a_{ij}}{a^{(m)}} - \frac{b_i}{b^{(m)}}\log\frac{1+B}{Z}\right) \tag{D.1}$$

$$Z \text{ obtained from } Z^3 - Z^2 + (A - B - B^2) - AB = 0 \tag{D.2}$$

$$A = a^{(m)}\frac{P}{R^2T^2} \tag{D.3}$$

$$B = b^{(m)}\frac{P}{RT} \tag{D.4}$$

$$R \approx 8.31446261815324 \tag{D.5}$$

$$a^{(m)} = \sum_{i,j} w_i w_j a_{ij} \tag{D.6}$$

$$b^{(m)} = \sum_{i,j} w_i w_j b_{ij} \tag{D.7}$$

$$a_{ij} = (1 - k_{ij})\sqrt{a_i a_j} \tag{D.8}$$

$$a_i = 0.42747\frac{R^2T_c^2}{P_c}\alpha_i \tag{D.9}$$

$$b_{ij} = (1 - k_{ij})\frac{b_i + b_j}{2} \tag{D.10}$$

$$b_i = 0.08554\frac{RT_c}{P_c} \tag{D.11}$$

$$k_{ij} \text{ a constant associated with the substances} \tag{D.12}$$

$$\alpha_i = (1 + (0.37464 + 1.65226\omega_i - 0.26992\omega_i^2)(1 - \sqrt{T_r}))^2 \tag{D.13}$$

$$\text{where } T_r = \frac{T}{T_c}, P_r = \frac{P}{P_c}, \omega \text{ a property of the substances} \tag{D.14}$$

$$\text{where } T_c, P_c \text{ properties of the substances.} \tag{D.15}$$

other expressions for the fugacity coefficient for the van der Waals, Virial, RK, Peng-Robinson and other cubic equations of state can be found in the book by Walas [54].

Below, we estimate the order of magnitude of evaluations required to calculated equation (D.1):

$$\text{all } a_i, b_i \rightarrow O(C) \tag{D.16}$$

$$\text{all } a_{ij}, b_{ij} \rightarrow O(C^2) \tag{D.17}$$

$$a^{(m)}, b^{(m)} \rightarrow O(a_{ij})O(C^2) = O(C^4) \tag{D.18}$$

$$A, B \rightarrow O(a^{(m)}) = O(C^4) \tag{D.19}$$

$$Z \rightarrow O(A) = O(C^4) \text{ to formulate the equation, } O(1)v\text{to solve it[41]} \tag{D.20}$$

which allows us to estimate the order of magnitude for each of the terms of equation (D.1):

Term	$\frac{b_i}{b^{(m)}}(Z-1)$	$\log(Z_B)$	$\frac{A}{B}\left(\frac{\sum w_j a_{ij}}{a^{(m)}}\right)$	$\frac{A}{B}\left(\frac{b_i}{b^{(m)}}\log\left(\frac{1+B}{Z}\right)\right)$
Magnitude	$O(C^4)$	$O(C^4)$	$O(C^4)$	$O(C^4)$

and thus

$$O(\sum_{i=1}^{C} \ln \phi_i) = O(C)O(C^4) = O(C^5) \tag{D.21}$$

Appendix E

Notes on the Implementation of the Methods

E.1 DIRECT (Dividing Rectangles)

E.1.1 Graham's Scan

The coordinates in S, (x, y), were converted to polar coordinates (Step 5 in **Algorithm** 4) via $r^2 = (x-P_x)^2+(y-P_y)^2$ (Pythagorean theorem) and $\tan(\theta) = \frac{P_y+y}{P_x+x}$, or $r = \sqrt{(P_x+x)^2+(P_y+y)^2}$,

$$\theta = \begin{cases} \cos^{-1}\left(\frac{y-P_y}{x-P_x}\right) \text{ if } (y-P_y) \geq 0 \\ -\cos^{-1}\left(\frac{y-P_y}{x-P_x}\right) \text{ if } (y-P_y) < 0 \\ \text{undefined if } r = 0 \end{cases},$$

where P_x, P_y are the horizontal and vertical coordinates of P respectively.
The angles α and β (see Figure 4.5) where calculated using the sine- and the cosine rule: via the cosine rule, the distance between points $k+1$ and k is

$$d_{k+1,k} = \sqrt{r_{k+1}^2 + r_k^2 - 2r_{k+1}r_k\cos(\theta_{k+1}-\theta_k)}$$

and via the cosine rule

$$\alpha = \arccos\left(\frac{d_{k+1,k}^2 + d_{k+1,P}^2 - d_{k,P}^2}{2d_{k+1,k}\cdot d_{k+1,P}}\right)$$

Similarily, we obtain

$$\beta = \arccos\left(\frac{d_{k+1,k+2}^2 + d_{k+1,P}^2 - d_{k+2,P}^2}{2d_{k+1,k+2}\cdot d_{k+1,P}}\right)$$

E.1.2 Procedure for Dividing Rectangles

In this procedure, and in DIRECT as a whole, the rectangles are saved by their center $\mathbf{c}_j$ and the lengths of each of their dimensions as such: $j = (\mathbf{c}, l_1, \ldots, l_C)$, where l_i is the length of the side

of the rectangle in dimension i. This is in opposition to what was recommended by Jones et al. [18], namely to save each rectangle by just storing information on the "search tree" of rectangles and subrectangles: the parent nodes, search tree depth, child type (left or right) etc. to save on storage space. As memory space was not a requirement in this case, and the dimensions do not go very high for chemical mixtures (less than 20 usually), it was decided for ease of implementation to store triangles as a $2 \times C$ matrix, where the first row represent the center, and the second row the length of each dimension.

E.1.3 DIRECT

Checking whether a rectangle is potentially optimal (Definition 4.2.2) was done both using a system of equations solver, and with the determination of the convex hull in MatLab (see also Section 5.2.1) for sake of comparison (see also Section 5.3). Finding the potentially optimal rectangles using a system of equations is done as follows. We will rewrite equations (4.3) and (4.4) as

$$\tilde{K}(d_j - d_i) \geq f(\mathbf{c}_j) - f(\mathbf{c_i}) \tag{E.1}$$

$$\tilde{K}d_j \geq f(\mathbf{c}_j) - (f_{min} - \varepsilon|f_{min}|) \tag{E.2}$$

In this case, we solve with $\bar{K}$ as the only variable, and

$$A = \begin{pmatrix} d_j - d_1 \\ d_j - d_2 \\ \vdots \\ d_j - d_m \\ d_j \end{pmatrix} \tag{E.3}$$

$$b = \begin{pmatrix} f(c_j) - f(c_1) \\ f(c_j) - f(c_2) \\ \vdots \\ f(c_j) - f(c_m) \\ f(c_j) - (f_{min} - \varepsilon|f_{min}|) \end{pmatrix} \tag{E.4}$$

where m is the total number of rectangles. This system is then solved using the fmincon function[1].

E.2 Stochastic Sampling and Clustering

We give the derivation of $m(S) = \text{Vol}(S)^2 = \left(\frac{(-1)^C}{(C!)^2 2^C}(CM(v_1, \ldots, v_C)\right)$ where the v_i are the vertices of the simplex and CM is the Cayley-Menger determinant

$$CM(v_1, \ldots, v_C) = \det \begin{pmatrix} 0 & d_{12}^2 & d_{13}^2 & \cdots & d_{1C}^2 & 1 \\ d_{12}^2 & 0 & d_{23}^2 & \cdots & d_{2C}^2 & 1 \\ d_{13}^2 & d_{23}^2 & 0 & \cdots & d_{3C}^2 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{1C}^2 & d_{2C}^2 & d_{3C}^2 & \cdots & 0 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \tag{E.5}$$

[1] The *solve* function proved unable to solve the system for numbers of rectangles over 500, which was frequently reached

where d_{ij} is the distance between v_i and v_j. The vertices of the simplex are affinely independent (if S has volume greater than 0), thus $(-1)^C CM(v_1, \ldots, v_n) > 0$. Because S is the standard simplex in this case, the vertices v_i are equal to e_i, the standard unit vectors. Thus,

$$d_{ij} = d(v_i, v_j) \text{ (see Section 4.3)} \tag{E.6}$$
$$= ((v_i - v_j)^T (v_i - v_j))^{1/2} \tag{E.7}$$
$$= ((e_i - e_j)^T (e_i - e_j))^{1/2} \tag{E.8}$$
$$= \sqrt{e_i^2 + e_j^2} = \sqrt{2} \tag{E.9}$$

Giving us $CM(v_1, \ldots, v_C) = (-1)^C 2^{C-1} C$.
α (from the threshold value (4.7))is defined as follows: let H_0 be the hypothesis that a cluster $K \subseteq S$ is a subset over which the same distribution as over S still holds. In this case, that means that he hypothesis states that the points in K are uniformly distributed. H_0 is rejected if no point $x \in K \subseteq S$ exists such that $m(K) = m(S)(1-\alpha^{1/n})$, where α corresponds to the probability of a so-called type I error, i.e. a false positive. Effectively this means that α represents the probability of rejecting a truth, i.e. the probability that we reject hypothesis H_0 while it actually holds. In practice, α is a chosen parameter, usually around $\alpha = 0.025$ (this is a convention in statistics).

Note that the clustering procedure might have some trouble expanding the cluster around a local minimum for some specific cases. Take a look, for example at the *tpd* surface for [0.4 0.4 0.2] H_2O-MeHOH-CO_2, Figure E.1 (see also Figure 3.6). Because the local minimum lies next to a discontinuity and a large jump, this causes the points in S_T of minimal distance to the local minimum $\mathbf{x}^*$ (and thus to its cluster K) to be highly likely to lie beyond the discontinuity. The approximated gradient will (correctly) not point in the direction of the local minimum, and the point will be rejected.

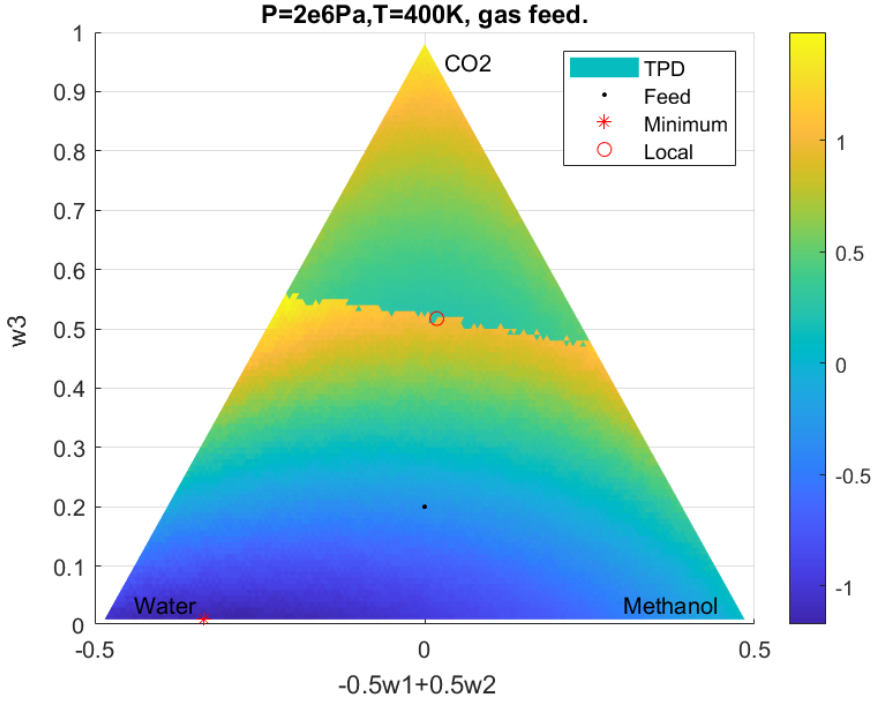


Figure E.1: The red o is the local minimum referred to in the text.

E.3 Transforming *tpd* Using Taylor Series

E.3.1 MacLaurin Series of Separate Parts

$$\ln \phi_i(\mathbf{w}) = \frac{b_i}{b^{(m)}}(Z-1) - \log(Z-B) - \frac{A}{B}\left(2\frac{\sum_j w_j a_{ij}}{a^{(m)}} - \frac{b_i}{b^{(m)}}\log\frac{1+B}{Z}\right) \tag{E.10}$$

where $A, B, a^{(m)}, b^{(m)}, Z$ and $\sum_{w_i}$ are the parts dependent on $\mathbf{w}$.

$$a^{(m)}(\mathbf{0}) = 0 \tag{E.11}$$

$$\frac{\partial a^{(m)}}{\partial w_i}(\mathbf{0}) = (\sum_{i\neq j;i,j=1}^{C} w_j a_{ij})(\mathbf{0}) + (2a_{ij}w_i)(\mathbf{0}) = 0 \tag{E.12}$$

$$\frac{\partial^2 a^{(m)}}{\partial w_i \partial w_j}(\mathbf{0}) =^{i\neq j} a_{ij} \tag{E.13}$$

$$\frac{\partial^2 a^{(m)}}{\partial w_i \partial w_i}(\mathbf{0}) = 2a_{ij} \tag{E.14}$$

$$\frac{\partial^k a^{(m)}}{\partial \cdots}(\mathbf{0}) = 0 \text{ for higher orders} \tag{E.15}$$

$$\Rightarrow T(a^{(m)}) = \sum_{i\neq j}\frac{w_i w_j}{1}a_{ij} + \sum \frac{w_i^2}{2}a_{ii} = \sum w_i w_j a_{ij} = a^{(m)} \tag{E.16}$$

$$T(b^{(m)}) = b^{(m)} \tag{E.17}$$

$$T(A) = A \tag{E.18}$$

$$T(B) = B \tag{E.19}$$

where $T(f)$ is the MacLaurin series of the function f. Using online mathematical solver WolframAlpha, we find that

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0 \tag{E.20}$$

$$\Rightarrow Z = \frac{0.41997(3A - 3B^2 - 3B - 1)}{Z_{AB}} + 0.26457 Z_{AB} + \frac{1}{3} \tag{E.21}$$

$$\frac{(0.20999 + 0.36371i)(3A - 3B^2 - 3B - 1)}{Z_{AB}} - (0.13228 - 0.22912i)Z_{AB} + \frac{1}{3} \tag{E.22}$$

$$\frac{(0.20999 - 0.36371i)(3A - 3B^2 - 3B - 1)}{Z_{AB}} - (0.13228 + 0.22912i)Z_{AB} + \frac{1}{3} \tag{E.23}$$

$$\tag{E.24}$$

where

$$Z_{AB} = \{\sqrt{4(3A - 3B^2 - 3B - 1)^3 + (27AB - 9A + 9B^2 + 9B + 2)^2} + (27B - 9A + 9B^2 + 9B + 2)\}^{\frac{1}{3}} \tag{E.25}$$

$$= (36B - 9A + \sqrt{(9B - 9A + 9B^2 + 27AB + 2)^2 - 4(3B^2 + 3B - 3A + 1)^3} + 9B^2 + 2)^{\frac{1}{3}} \tag{E.26}$$

Appendix F

Purely Mathematical Description

In stability tests, we are looking to solve the following minimisation problem:

$$\min\ tpd(\mathbf{w}) = \sum_i w_i(\ln w_i + \ln \phi_i(\mathbf{w}) - \ln z_i - \ln \phi_i(\mathbf{z}) \tag{F.1}$$

$$\text{subject to } w_i > 0,\ i = 1,2,\ldots,\text{C}, \tag{F.2}$$

$$\sum_i^C w_i - 1 = 0 \tag{F.3}$$

where $\mathbf{w} \in [0,1]^C$ is a variable vector of fractions, $\mathbf{z}$ a constant vector of fractions, and ϕ is a nonlinear, nonconvex function. This is a nonconvex quadratic programming problem in the standard $C-$simplex in C dimensions. ϕ is a function dependent on the type of model we are using, for our case:

$$\ln \phi_i(\mathbf{w}) = \frac{b_i}{b^{(m)}}(Z-1) - \log(Z-B) - \frac{A}{B}(2\frac{\sum_j w_j a_{ij}}{a_m} - \frac{b_i}{b^{(m)}} \log \frac{1+B}{Z})$$

where $A, B, a^{(m)}, b^{(m)}, Z$ are all dependent on $\mathbf{w}$. A more rigorous enumeration of all the variables and parameters follows below:

$$\ln \phi_i(\mathbf{w}) = \frac{b_i}{b^{(m)}}(Z-1) - \log(Z-B) - \frac{A}{B}(2\frac{\sum_j w_j a_{ij}}{a_m} - \frac{b_i}{b^{(m)}} \log \frac{1+B}{Z}) \tag{F.4}$$

$$Z \text{ obtained from } Z^3 - Z^2 + (A - B - B^2) - AB = 0 \tag{F.5}$$

$$A = a^{(m)} \frac{P}{R^2 T^2} \tag{F.6}$$

$$B = b^{(m)} \frac{P}{RT} \tag{F.7}$$

$$R \approx 8.31446261815324 \tag{F.8}$$

$$a^{(m)} = \sum_{i,j} w_i w_j a_{ij} \tag{F.9}$$

$$b^{(m)} = \sum_{i,j} w_i w_j b_{ij} \tag{F.10}$$

$$a_{ij} = (1 - k_{ij})\sqrt{a_i a_j} \tag{F.11}$$

$$a_i = 0.42747 \frac{R^2 T_c^2}{P_c} \alpha_i \tag{F.12}$$

$$b_{ij} = (1 - k_{ij}) \frac{b_i + b_j}{2} \tag{F.13}$$

$$b_i = 0.08554 \frac{RT_c}{P_c} \tag{F.14}$$

$$k_{ij} \text{ a pre-defined constant} \tag{F.15}$$

$$\alpha_i = (1 + (0.37464 + 1.65226\omega_i - 0.26992\omega_i^2)(1 - \sqrt{T_r}))^2 \tag{F.16}$$

$$\text{where } T_r = \frac{T}{T_c}, P_r = \frac{P}{P_c}, \omega \text{ constants} \tag{F.17}$$

$$\text{where } T_c, P_c \text{ constants} \tag{F.18}$$

From experiments, the function $tpd(\mathbf{w})$ has shown to contain local minima and sometimes discontinuities (for certain P, T and substances).
For Problem F.4, it is in many cases practically sufficient to know whether the global minimum is negative. Thus, when any feasible negative value for the objective function is found, the search can often be terminated.