



Norwegian University of
Science and Technology

Dynamic Simulation of Heat Exchanger with Multicomponent Phase Change

Cristina Florina Zotica

Chemical Engineering

Submission date: June 2017

Supervisor: Sigurd Skogestad, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This thesis is written as the final part of my Master of Science Degree in Chemical Engineering at the Norwegian University of Science and Technology.

I would like to thank my supervisor, Sigurd Skogestad for his patience, comments, suggestions and valuable support and time invested in the project. I would also like to thank Vladimiro Minasidis for the helpful discussions we had towards the end of the thesis.

Declaration of Compliance

I, Cristina Zotică, hereby declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology.

Trondheim, June 22nd 2017

Cristina Zotică

Abstract

A dynamic model for a heat exchanger with multicomponent phase change, part of reliquefaction cycle of natural gas, is developed. The finite control volume method is used to spatially discretize the heat exchange into a series of lumps with constant volume. Vapor-liquid equilibrium is assumed in the vapor-liquid region of the phase envelope. The model is written in terms of differential and algebraic equations applied to each lump (or cell).

Different algebraic equations are valid in each of the phase regions (e.g. vapor, liquid, vapor-liquid), namely the vapor-liquid equilibrium condition is not satisfied in either of the single phases. Therefore, each phase region has its own set of differential and algebraic equations. The number required to describe the two-phase region is higher compared to the single regions. Hence, dummy variables and equations (without a physical meaning) are used in the single regions to in order to have the same number of equations in all phases such that the same model can be used for simulating all phase regions. A logical conditions is implemented to select the corresponding set of equations.

The model is written and implemented in Matlab[®] for simulation purposes. The phase change detection is automatically handled by an event function inside the solver. A few additional examples are given to investigate how the *ode15s* solver treats non-smooth systems, or how the algebraic equations are solved.

The possibility of formulating the model as a mathematical problem with complementarity constraints is also investigated.

Contents

Preface	i
Abstract	ii
1 Introduction	1
1.1 Scope	2
1.2 Modelling approach of this work	3
1.3 Thesis structure	3
2 Background	5
2.1 Process description	5
2.2 Literature overview	7
2.2.1 Finite control volume versus moving boundary	7
2.2.2 Overview of related work	9
3 Theory	15
3.1 Thermodynamic model	15
3.1.1 Soave modification of Redlich-Kwong cubic equation of state	16

3.1.2	Vapor-liquid equilibrium calculation	18
3.1.3	Enthalpy calculation	21
3.1.4	Implementation of the thermodynamic model	22
3.2	Process modelling approaches	24
3.2.1	Differential and algebraic equations	24
3.2.2	Mathematical problem with complementarity constraints	25
4	Heat exchanger model	27
4.1	Phase envelope	27
4.2	Assumptions	29
4.3	Model illustration	30
4.4	Inputs and states	31
4.5	Model stiffness	32
4.6	Model formulations	33
4.6.1	Different set of equations for each phase region	33
4.6.2	Model with complementarity constraints	50
4.6.3	Alternative formulations	51
5	Simulation methods	55
5.1	The <i>ode15s</i> solver	55
5.1.1	Mass matrix	56
5.1.2	Non-smooth continuous ODE	57
5.1.3	Investigating how the algebraic equations are solved	58

5.1.4	Event function	65
5.2	MPCC solving methods	66
5.2.1	Discretizing the differential equations	67
6	Simulation results	69
6.1	Results of simulations with ode15s	69
6.1.1	Initial conditions	70
6.1.2	Dew and bubble temperatures calculations	71
6.1.3	Vapor phase simulation	72
6.1.4	Two phase simulation	79
6.1.5	Vapor phase to vapor-liquid region simulation	86
6.1.6	Vapor phase to vapor-liquid region with event detection simulation	88
7	Discussion	95
7.1	The heat exchanger model	95
7.1.1	The assumptions	96
7.2	The solver and simulations	97
8	Conclusions	99
8.1	Future work	100
	Bibliography	101
A	Notations	iii
A.1	Acronyms	iii

A.2	List of Symbols	iv
A.3	Subscripts	vi
A.4	Greek Letters	vii
B	Thermodynamic constants	ix
B.1	Coefficients for specific heat	ix
B.2	Interaction Parameters	ix
B.3	Critical Pressure, Temperature and Acentric Factor	x
C	Matlab scripts	xiii
C.1	Main script one phase simulation	xiii
C.2	Main script two phase simulation	xv
C.3	Main script one phase to two phase simulation	xvi
C.4	Heat exchanger model	xix
C.5	Event function	xxiv
C.6	Soave modification of Redlich-Kwong	xxv
D	Test of ode15s	xxix
D.1	Example from section 5.1.2	xxix
D.2	Example from section 5.1.3	xxx
D.2.1	Main script	xxx
D.2.2	DAE system solved by ode15s in one layer approach from section 5.1.3.1	xxxi
D.2.3	DAE system solved in two layer approach from section 5.1.3.2 . . .	xxxii

List of Tables

2.1	Process conditions for mixed refrigerant	6
2.2	Mixed refrigerant molar composition	7
4.1	Possible phase transitions	29
4.2	Model States	32
5.1	Initial conditions	62
6.1	Dew and bubble points for a pressure variation of 5 mbar	71
6.2	Simulation and design parameters for vapor phase	73
6.3	Simulation and design parameters for two phase	79
6.4	Simulation and design parameters for two phase with event function	90
B.1	Polynomial coefficients for calculation of $\frac{C_p}{R}$	ix
B.2	Interaction parameters	xi
B.3	Critical Pressure, Temperature and Acentric Factor	xii

D.1 Results for DAE example from section 5.1.3.2, initialized with complex values for z_2 xxxii

List of Figures

2.1	Countercurrent heat exchanger	6
2.2	Finite control volume method	8
2.3	Moving boundary	8
3.1	Schematic representation of vapor-liquid equilibrium	19
4.1	Phase envelope of a nitrogen-methane-ethane-propane mixture with composition from table 2.2	28
4.2	Heat exchanger lumped model	31
4.3	Representative cell of the heat exchanger	34
4.4	Time dependent heat function	46
5.1	Nonsmooth continuous function	58
5.2	Integration of function from equation 5.3	59
5.3	Change in the number or real roots for equation 5.5	61
5.4	Solving methods for equations 5.4	64
5.5	Polynomial approximation for state profile across a finite element	68

6.1	Temperature profile for vapor phase simulation	74
6.2	Steady-state temperature profile for vapor phase simulation	74
6.3	Steady-state liquid volume profile for vapor phase simulation	75
6.4	Dynamic total holdup profile for vapor phase simulation	75
6.5	Steady-state total holdup profile for vapor phase simulation	76
6.6	Dynamic pressure profile for vapor phase simulation	76
6.7	Steady-state pressure profile for vapor phase simulation	77
6.8	Temperature profile when heat is varying heat with time	78
6.9	Total holdup profile when heat is varying heat with time	78
6.10	Pressure profile when heat is varying heat with time	78
6.11	Temperature profile for two phase simulation	80
6.12	Steady-state temperature profile for two phase simulation	80
6.13	Dynamic liquid volume profile for two phase simulation	81
6.14	Steady-state liquid volume profile for two phase simulation	81
6.15	Dynamic total holdup profile for two phase simulation	82
6.16	Steady-state total holdup profile for two phase simulation	82
6.17	Dynamic pressure profile for two phase simulation	83
6.18	Steady-state pressure profile for two phase simulation	83
6.19	Temperature profile when heat is varying heat with time	84
6.20	Total holdup profile when heat is varying heat with time	85
6.21	Liquid volume profile when heat is varying heat with time	85
6.22	Pressure profile when heat is varying heat with time	86

6.23 Profiles for a transition from vapor phase to vapor-liquid region with algorithm 3	88
6.24 Dynamic temperature profile when using event function	91
6.25 Dynamic liquid volume profile when using event function	92
6.26 Dynamic total holdup profile when using event function	92
6.27 Dynamic Pressure profile when using event function	93

Chapter 1

Introduction

Heat exchangers are a common equipment present in any industrial plant or in daily life activities. A special research interest is directed towards cryogenic process, also known as refrigeration cycles, where heat exchangers are the core components. The drive behind this interest is to minimize the high energy consumption of refrigeration by process optimization which would lead to a decrease in operation costs. An important step towards optimal operation of refrigeration cycles is to develop a robust dynamic model of the heat exchanger. This type of model can either be used to simulate the cryogenic process, or it can be further used in an equation oriented approach (i.e. all model equations are solved simultaneously by the optimizer solver). The latter is a considerable more powerful tool compared to sequential modular approach (i.e. numerical results from simulating each unit individually are passed to an optimizer solver) for determining the optimal operation conditions for a given process [Kamath et al., 2010].

Developing a heat exchanger model for a cryogenic process is not trivial, considering a stream is susceptible to a phase transition from any region of the phase envelope to any other region (e.g. vapor, liquid-vapor, liquid). Thus, the main challenge is not knowing beforehand where and when a phase transition happens [Watson et al., 2015]. It is essential to handle the appearance and disappearance of phases, because the topology of the process changes accordingly. In other words, different sets of equations are valid in the single and two phase regions respectively, and identifying the correct location of the transition allows using a valid set of equations.

1.1 Scope

The scope of this thesis is to continue and improve the work begun in the specialization project [Zotică, 2016] where an incipient dynamic model with non-ideal thermodynamics for a heat exchanger part of a simple reliquefaction cycle was presented. In the previous work, it was described that it is desired to have a simple and robust dynamic heat exchanger model which can be used regardless of what the stream conditions are without having complicated logical propositions or similar mathematical formulations. The model in the specialization project encounter a series of numerical problems which had to be dealt with. Among other are a differential index greater than 1, arose from considering constant pressure in the system, and inconsistency of initial conditions arose from not initializing the system at a steady-state. Although, the heat exchanger was simulated for different regions of the phase envelope, the phase transition was not detected in the previous work.

In this thesis, it is now better understood that the phase transition of a stream is more challenging and a different formulation of the model should be constructed to properly handle the phase changes. The key assumption of the project was discretizing the heat exchanger into a given number of cells with constant volume, and using a phase equilibrium equation based on K-values to determine the composition of each phase in each of the cells, regardless of the phase region. However, the assumption of phase equilibrium does not provide a feasible numerical solution when one of the phases does not exist. In contrast, this work provides a better understanding and explanation of how and why the model should change with the phase transition, followed by a method for phase transitions detection. In order to overcome this challenge, different approaches for handling the appearance and disappearance of phases have been tried in parallel and are discussed in chapter 4.

Moreover, this thesis also revises a few of the assumptions and equations presented in the specialization project, which are further developed in chapter 4. In addition, the simulations carried in the project took longer time than desired and this aspect is also to be improved in the present work.

The work in this thesis is carried out having in mind that the final use of this dynamic heat exchanger model with phase change is found in an equation-oriented approach for process optimization of a simple LNG refrigeration cycle. However, only

model simulations are part of this work and thus optimization of the process is not further developed.

1.2 Modelling approach of this work

The modelling approach of this thesis shares a few common points with the methodologies presented in section 2.2. The model is based on dynamic flash calculations which are formulated in an equation oriented environment, meaning that all equations are solved simultaneously by the solver. The key difference compared to the works from section 2.2 where a stream was divided into substreams, is that the heat exchanger in this thesis is discretized into a given number of cells or lumps with constant and equal volume and the model equations are solved for in each of the cells. Hence, a finite control volume method is used instead of a moving boundary method (see section 2.2.1). Further, the model assumes vapor-liquid equilibrium in the two phase region, equation based on K-values. The non-ideal behaviour of the vapor-liquid system is accounted by incorporating in the model Soave modification of Redlich-Kwong cubic equation of state, while P eneloux correction is used for the liquid density. In addition, a dynamic model is chosen in favour of a steady-state model. The base of this choice is that a dynamic model offers information about how the process reacts when different parameters or variables (such as heat rate) are modified. Moreover, a dynamic model offers information about the steady-state of the system. Last but not least, this framework also incorporates a strategy to detect a phase transition based on triggering an event inside the solver when the stream conditions are changing from one region to another.

The model is formulated as a set of differential and algebraic equations and the *ode15s* solver in Matlab[®] is used for simulation.

1.3 Thesis structure

The structure of this thesis is the following:

Chapter 1 offers a introduction to the work along with its scope

Chapter 2 presents a literature overview of relevant work

Chapter 3 presents the thermodynamic model and different model approaches

Chapter 4 formulates the heat exchanger model

Chapter 5 describes simulation methods investigated

Chapter 6 presents the simulation results

Chapter 7 contains a final discussion of the work

Chapter 8 consists of conclusions and future work.

Chapter 2

Background

This chapter contains the process description and a brief overview of relevant work.

2.1 Process description

The heat exchanger considered in this work is part of the natural gas reliquefaction plant for small gas carriers described by Nekså et al. [2010]. During transport by tankers, LNG is contained very close to its vaporization point and due to unavoidable heat losses, part of the fluid is naturally evaporated as boil-off gas. To maintain the carrier tank pressure, the boil-off gas has to be removed which leads to product losses. A solution to avoid losses is to reliquefy the boil-off gas and send it back to the tank.

Hence, the goal of the process described by Nekså et al. [2010] is to develop a small reliquefaction cycle for boil-off gas on board of liquid natural gas carriers such that the transport has high energy efficiency and low costs. The former and the latter are achieved by using standard refrigeration equipment such as two-streams copper brazed plate heat exchanger as opposite to the traditional and more costly multistream heat exchanger which is normally used in refrigeration cycles. The process has a number of five copper brazed plate heat exchangers which operate in all phase regions (vapor, vapor-liquid and liquid). A robust heat exchanger model for this process has to be developed in such way that it is valid in all operating conditions and for all heat exchangers.

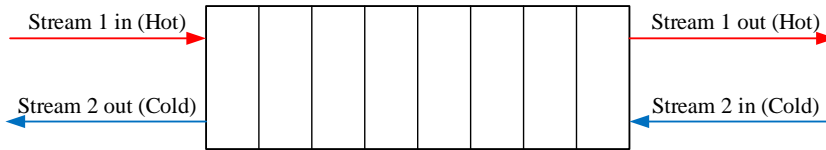


Figure 2.1: Countercurrent heat exchanger

However, due to a complicated flow pattern of a copper brazed heat exchanger and not having enough design parameters available, the heat exchanger is modelled as a countercurrent one in this work, as illustrated in figure 2.1. Other assumptions and considerations of the model are given in section 4.2.

In reference to figure 2.1, one of the streams is called mixed refrigerant (which can be both cold and hot), while the others can be boil-off gas (hot stream) or seawater (cold stream). However, this work looks closely only on what happens to the mixed refrigerant stream, as further discussed in section 4.2. For this reason, only the process conditions corresponding to the mixed refrigerant are given below. The inlet molar flow and pressure are presented in table 2.1, while the mixed refrigerant compositions is presented in table 2.2. For all the heat exchangers, a pressure drop of 5 mbar is considered. Other particular parameters, such as inlet temperature, is given in the case of each simulations in section 6.1.

Table 2.1: Process conditions for mixed refrigerant

Feed parameters	value	units
Flow	4073	$\frac{\text{kg}}{\text{h}}$
Flow	2.67	$\frac{\text{kmol}}{\text{min}}$
Pressure	18	atm
	1.8239	MPa

This chapter contains a literature overview relevant to modelling of phase change processes together with the theory needed to develop the heat exchanger model of this work.

Table 2.2: Mixed refrigerant molar composition

Component	Molar fraction
Nitrogen	0.06
Methane	0.4
Ethane	0.4
Propane	0.14

2.2 Literature overview

A brief literature overview is presented in this section. The methods for choosing a control volume for the process are presented in subsection 2.2.1, while in subsection 2.2.2 a few methods for handling a phase transition are presented.

2.2.1 Finite control volume versus moving boundary

Before any modelling approach of the know process is attempted, one should have an idea of the geometry of the system, or at least what are the boundaries of the system and what is the control volume for which the equations are further written and solved. There are two different methods available in literature used to describe the control volume of a multiphase heat exchanger for dynamic simulations, and these are [Pangborn et al., 2015]:

1. finite control volume: the heat exchanger is spatially discretized into a given number of cells with fixed volume as illustrated in figure 2.2. Each cell is considered perfectly mixed. Each cells corresponds to only one region of the phase envelope (e.g. vapor, vapor-liquid or liquid)
2. moving boundary : the heat exchanger is divided into regions corresponding to superhetead fluid, two phase fluid and subcooled fluid and a control volume is attributed to each of them, meaning that the boundary of the control volume of each region is capable of changing both in time and space as illustrated figure 2.3

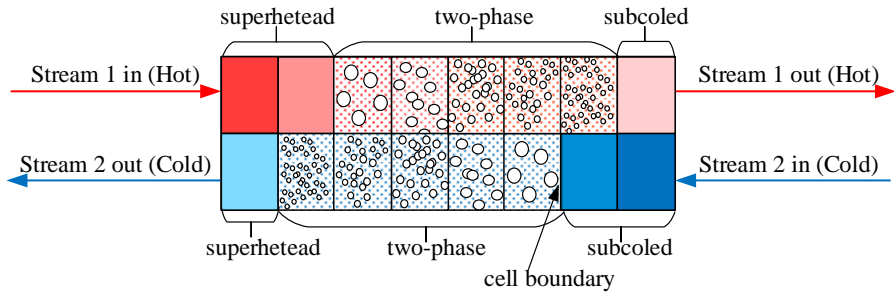


Figure 2.2: Finite control volume method

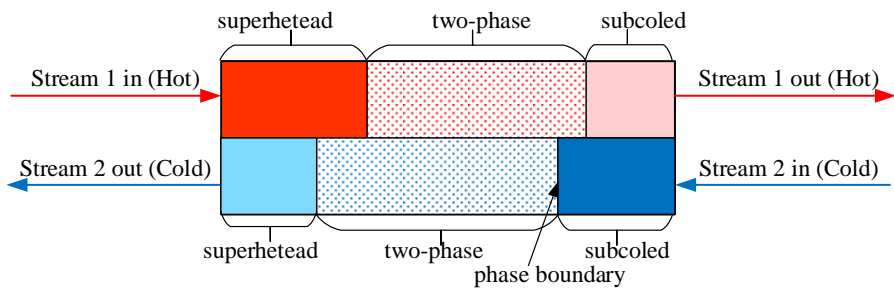


Figure 2.3: Moving boundary

2.2.1.1 Comparison between finite control volume and moving boundary

The most obvious difference between the two methods is the number of control volumes. By comparing figures 2.2 and 2.3 it can be observed that the maximum number of control volume in the moving boundary method is maximum 3, while in the finite control volume there is no such limit. In the latter method, the number of cells can be chosen based on a trade off between accuracy (higher number of cells) and computation speed (lower number of cells) [Pangborn et al., 2015]. As the model equations are written and solved for in each of the control volumes, the number of variables and equations is expected to be higher in the finite control volume method.

Other considerations between the two methods are [Pangborn et al., 2015]:

- it is more convenient to formulate a model with a finite control volume method as the boundaries of the cells are not moving and thus it is easier to implement
- the finite control volume method is more flexible to different geometries of heat exchangers
- the moving boundary has slightly faster simulation speed since it has fewer control volumes and thus fewer variables and equations
- both methods have the same accuracy when it comes to comparison with experimental data

The method chosen in this project is the finite control volume method since it is more convenient to implement. Further, it can represent different geometries of a heat exchanger and it can also account for back-mixing of the fluid. Moreover, it has good accuracy in representing experimental data [Pangborn et al., 2015].

2.2.2 Overview of related work

The appearance and disappearance of phases transform the set of equations in a non-smooth system and thus solving the system of equations becomes more challenging with conventional solvers, considering that the system is non-differentiable at the non-smooth point. There are a few noteworthy modelling approaches available in the open literature which deal with the non-smoothness caused by a phase transition.

Kamath et al. [2010] presents a method which implements complementarity constraints to exclude the phase that is not present from steady state flash calculations followed by smoothing the resulting non-smooth formulation by relaxing the vapor-liquid equilibrium equation such that a feasible solution can be obtained both in single and two phase region. However, this methods requires solving an optimization problem to determine the relaxation parameter β and the slack variables s_V and s_L found in the complementary constraints, and naturally the model becomes computationally demanding to solve with the increase of variables which is an disadvantage. Their smooth model based on a flash calculation is presented in equations 2.1 [Gopal and Biegler, 1999]. The resulted model formulation is also known as a mathematical program with complementarity constraints (MPCC).

$$F = L - V \quad (2.1a)$$

$$Fz_i = Lx_i - Vy_i \quad i = 1, n \quad (2.1b)$$

$$y_i = \beta K_i(P, T, x)x_i \quad i = 1, n \quad (2.1c)$$

$$\sum_i y_i - \sum_i x_i = 0 \quad (2.1d)$$

$$\beta - 1 = s_V - s_L \quad (2.1e)$$

$$0 \leq V \perp s_V \geq 0 \quad (2.1f)$$

$$0 \leq L \perp s_L \geq 0 \quad (2.1g)$$

$$0 \leq x_i, y_i \leq 1 \quad (2.1h)$$

$$L, V \leq F \quad (2.1i)$$

The relaxation method and complementarity constraints can be explained as following Kamath et al. [2010]:

- if $\beta = 1 \Rightarrow L, V > 0$, the phase equilibrium is not relaxed and the system is in two phase region;
- if $\beta > 1 \Rightarrow V = 0, s_V > 0, s_L = 0$, the phase equilibrium is relaxed to obtain a feasible solution and the system is in the liquid region;
- if $\beta < 1, \Rightarrow L = 0, s_V = 0, s_L > 0$, the phase equilibrium is relaxed to obtain a feasible solution and the system is in the vapor region.

Kamath et al. [2012] describes an equation oriented approach to model, simulate

and optimize at steady state a multistream heat exchanger (MHEX), where one of several stream might change phase. The core concept of their work is based on modification of the pinch analysis presented by Duran and Grossmann [1986] to ensure the minimum driving force criteria of such heat exchangers. The main disadvantage of using a pinch analysis is that it relies on the approximation of constant heat capacities of the streams which does not hold in a process with phase change, considering the large variation between the vapor and liquid heat capacities. To improve this approximation Kamath et al. [2012] proposes dividing each stream that might change phase into three different substreams corresponding to superheated, two phase and subcooled, substreams. These can in turn be divided into n segments with constant heat capacities and thus having a better approximation for the pinch analysis. Further, the complementarity constraints from the formulation of Gopal and Biegler [1999] are replaced with disjunctions and logical propositions, meaning that only the heat load of the substreams that exist is taken into consideration when solving the model. The vapor-liquid equilibrium equations is not relaxed based on the assumption that the process is operated inside (and at) the boundary of the two phase region, and hence a flash calculation will always have a feasible solution. At the same time, it also implies that the conditions (temperature at the given pressure) of the superheated stream correspond to the dew point, while the conditions of the subcooled stream are equal to the bubble point. This however, represents a limitation that is imposed on a real refrigeration cycle which can of course operate outside the two phase boundary. As a result of having disjunctions in the model, this formulation falls into the category of mixed integer non-linear program (MINLP), which as the MPCC formulation is not trivial to solve. Since a pinch analysis is beyond the scope of this thesis, the equations presented in the work of Kamath et al. [2009] are not included in this report.

A different way of handling the phase transition is reported in the work of Watson et al. [2016]. Similar to the previous approach, the authors chose an equation oriented environment to simulate a multistream heat exchanger at steady-state, again with the help of the pinch analysis. The method differs from the previous ones in that instead of smoothing the set of equations or using disjunctions and logical propositions, it applies a generalized gradient algorithm to generate derivative-like information and thus solving the non-smooth formulation of the model to capture the phase transition. The model consists of [Watson et al., 2015]:

- an energy balance between the hot (F) and cold (f) stream with constant capac-

ities c_p , equation 2.2a

- a non-smooth *minimum* function for the pinch location (equation 2.2b), modified from the work of Duran and Grossmann [1986]
- an equation that relates the heat exchanger area and the total heat transfer via the overall heat transfer coefficient and the logarithmic mean temperature difference, equation 2.2c
- a second non-smooth but continuous *mid* function to determine the vapor fraction, equation 2.3

$$\sum_{i \in H} F \cdot c_{p,i} \cdot (T_i^{in} - T_i^{out}) - \sum_{j \in C} f \cdot c_{p,j} \cdot (T_j^{in} - T_j^{out}) = 0 \quad (2.2a)$$

$$\min_{p \in P} \{EBP_H^p - EBP_C^p\} = 0 \quad (2.2b)$$

$$UA - \sum_{\substack{d \in D \\ d \neq D}} \frac{\Delta Q^d}{\Delta T_{LM}^d} = 0 \quad (2.2c)$$

Again, one of the disadvantages of the pinch analysis used is considering a linear variation of enthalpy with temperature, or in other words a constant heat capacity, which is not a good representation of the phase change process, especially since the streams are multicomponents. Thus, Watson et al. [2016] implement the same approach proposed by Kamath et al. [2012] of dividing each stream into superheated, two phase and subcooled substreams which can be subsequently divided into n segments with constant heat capacity. Then, the temperature of each substream (or segment) is written as a non-smooth continuous function of the dew point, bubble point and the inlet or outlet temperature of the stream.

Since the vapor fraction calculation is the most interesting in respect with this thesis, only this function is presented in equation 2.3 and its mechanism is further explained. The three terms of the mid function can be seen as representing the vapor fraction of the outlet stream when this is in vapor, liquid or in the two phase region respectively. When the system is in vapor-liquid region, the first term is between zero and one ($[0 : 1]$), the second term is negative while the third term is equal to zero, which leads to an evaluation to 0 of the last of the terms. The same logic is applied for the other

regions. It should be noticed that the third term is in fact the Rachford-Rice equation used to determine the vapor fraction of a flash calculation at steady-state.

$$\text{mid} \left\{ \frac{V}{F}, \frac{V}{F} - 1, - \sum_{i=1}^{n_c} \frac{z_i \cdot (K_i - 1)}{1 + \frac{V}{F} \cdot (K_i - 1)} \right\} \quad (2.3)$$

All the modelling approaches described above can be used either to simulate the heat exchanger from a refrigeration cycle, or for optimal operation of the heat exchanger by an equation oriented approach at steady-state. With these formulations, the process is simulated by setting the objective function to a constant value and impose all the equations of the model as constraints to the optimizer solver. Thus, one common drawback is that they do not capture the dynamics of the phase change process.

On the other hand, other modelling procedures available in the literature propose a dynamic model and only perform simulations of the heat exchanger. The key difference between simulation and optimization of a phase change process is that derivatives information are need for the latter but not for the former. Non-smooth continuous functions can be integrated but are non-differentiable in all the definition domain.

Wilhelmsen et al. [2013] describes a model also based on flash calculation consisting of a differential and algebraic system of equations where the conditions of the current iteration point are tracked on the phase envelope to switch between different types of valid algebraic equations corresponding to the different regions. However, the vapor-liquid equilibrium equation is imposed in the single-gas region, which means that the stream in this phase has to be saturated vapor, which again might not accurately represent a cryogenic process. Another setback is represented by the use of look-up table to determine the thermo-physical properties of the components, which may lead to model convergence problems compared to the procedure where the thermo-physical properties are given by a equation of state incorporated in the model [Reyes-Lúa et al., 2016]. Moreover, even though the model in the work of Wilhelmsen et al. [2013] is desired to be dynamic, the vapor fraction in the two phase region is determined with Rachford-Rice equation, which can only be applied at steady-state. What is important to extract from it is the possibility of triggering events in the integration algorithm to detect a phase transition.

The work of Sahlodin et al. [2016] lies in the same area of only simulating a dynamic model for a phase change process. This approach is based on a dynamic flash calcula-

tion with the relaxation of vapor-liquid equilibrium presented by Kamath et al. [2010], together with a non-smooth function applied to correctly solve for the vapor fraction. The latter is an extension to a dynamic formulation of the non-smooth function presented by Watson et al. [2016]. Thus equation 2.3 is reformulated considering the vapor and liquid holdup (which are time variant) as a substitute for of the steady-state flows for in the inlet and vapor and liquid outlets resulting in equation 2.4. Only the non-smooth *mid* function of the model presented by Sahlodin et al. [2016] is given here since it offers a good possibility of transitioning from one phase region to another when the purpose is to simulate in a dynamic regime.

$$\text{mid} \left\{ \frac{n_V}{n_V + n_L}, \frac{n_V}{n_V + n_L} - 1, - \sum_{i=1}^{nc} \frac{n_i \cdot (K_i - 1)}{\sum_{j=1}^{nc} 1 + \frac{n_V}{n_V + n_L} \cdot (K_i - 1)} \right\} \quad (2.4)$$

However, what this work does not account for, is that relaxing the vapor-liquid equilibrium requires to optimize for the parameter β .

Chapter 3

Theory

This chapter presents the thermodynamic model used in this work, along with general methods for formulating the model equations.

3.1 Thermodynamic model

There are several approaches to predict the non-ideal behaviour of either vapor, liquid, solid or multiphase systems. Commonly, a thermodynamic equation, also known as equation of state, is applied to relate state functions such as pressure, volume or temperature to determine properties of pure fluids or mixtures of fluids [Kamath et al., 2010]. In the case of a multiple phase system, the options are applying the same equation of state for all phases or using a different equation of state for each phase, the choice being made such that the thermodynamics accurately describe the characteristics of the process [Reid et al., 1987]. Among other equations of the state, the cubic equations of state are quite popular in describing chemical processes which is a result of their advantages: they are simple to apply, require few calculated parameters and have a low computational overhead.

Having in mind the framework and scope of this work, the same cubic equation of state is considered to be enough to describe the properties of all the fluids present in this cryogenic process. In an LNG refrigeration cycle, it is expected that the compo-

nents are nonpolar or slightly polar substances and the only phases are vapor or liquid. Thus the Soave modification of Redlich-Kwong cubic equation of state represents a good thermodynamic model to predict the required properties of this model [Soave, 1972].

3.1.1 Soave modification of Redlich-Kwong cubic equation of state

This section presents how the necessary thermodynamic properties are obtained from Soave modification of Redlich-Kwong cubic equation of state. The general form of a cubic equation of state is given in equation 3.1, where the roots of the function, ξ , can either be molar volume or compressibility [Kamath et al., 2010]. It should be noticed that this equation can either have three real roots, or two complex conjugate and one real roots. Particularly, in the two phase region the equation has three real roots, the largest one being for the vapor phase, the smallest one being for the liquid phase, while the middle one does not have a physical meaning. In the single phase region, the equation has two imaginary conjugate and one real, the latter belonging to the phase that the mixture or process stream is in (e.g. vapor or liquid).

$$\xi^3 + a_1 \xi^2 + a_2 \xi + a_3 = 0 \quad (3.1)$$

The thermodynamic model for this work is written in terms of compressibility Z , which is a measure of deviation from ideal gas behavior of a fluid. The equations are given for a multicomponent system since this type of fluids are expected to be found as process stream in a LNG refrigeration cycle. SRK cubic equation of state for a mixture of NC components is given in equation 3.2. For this formulation Van der Waals mixing rules are used. First a geometric average of each parameter is calculated, followed by a weighting based on molar composition to determine the average mixture parameters, equations 3.2c, 3.2e and 3.2b [Reid et al., 1987]. The interaction between components, one of the reasons for non-ideal behavior, is also accounted by using interaction parameters $k_{i,j}$.

$$Z^3 - Z^2 + Z(A - B - B^2) - A \cdot B = 0 \quad (3.2a)$$

$$A = \sum_i^{NC} \sum_j^{NC} x_i \cdot x_j \cdot A_{i,j} \cdot (1 - k_{i,j}) \quad (3.2b)$$

$$A_{i,j} = \sqrt{A_i \cdot A_j} \quad (3.2c)$$

$$A_i = 0.42747 \cdot \alpha_i(T) \cdot \frac{P_r}{T_r^2} \quad (3.2d)$$

$$B = \sum_i^{NC} x_i \cdot B_i \quad (3.2e)$$

$$B_i = 0.08664 \cdot \frac{P_{r,i}}{T_{r,i}} \quad (3.2f)$$

$$P_{r,i} = \frac{P}{P_{c,i}} \quad (3.2g)$$

$$T_{r,i} = \frac{T}{T_{c,i}} \quad (3.2h)$$

$$\alpha_i = [1 + m_i(1 - T_r^{0.5})]^2 \quad (3.2i)$$

$$m_i = 0.48 + 1.574 \cdot \omega_i - 0.176 \cdot \omega_i^2 \quad (3.2j)$$

$$i, j = 1 \dots NC \quad i \neq j$$

The fugacity of each component in a mixture is calculated as an explicit function of compressibility from equation 3.3a, while again the components' interaction with each other is corrected with a factor δ given by equation 3.3b. This is required for vapor-liquid equilibrium calculations described in section 3.1.2.

$$\ln \phi_i = (Z - 1) \frac{B_i}{B} - \ln(Z - B) - \frac{A}{B} \ln \left(\frac{Z + B}{Z} \right) \left(2\delta_i \frac{A_i^{0.5}}{A} - \frac{B_i}{B} \right) \quad (3.3a)$$

$$\delta_i = \sum_j^{NC} x_j \cdot A_j^{0.5} \cdot (1 - k_{i,j}) \quad (3.3b)$$

3.1.1.1 Molar volume calculation

The molar volume of each phase can also be computed with parameters resulted from SRK CEOS. However, while the explicit equation 3.4 where the independent variables

compressibility, temperature and pressure determine the dependent variable molar volume of the vapor phase with good accuracy, a correction is needed for the liquid phase molar volume.

$$V_m = \frac{ZRT}{P} \quad (3.4)$$

Thus, the correction presented by P eneloux et al. [1982] is used in this work, given by equation 3.5b. The correction factor c takes into account Rackett compressibility factor [Spencer and Danner, 1972], which is an explicit function of the acentric factor.

$$V_m = \frac{ZRT}{P} - c \quad (3.5a)$$

$$c = 0.40768 \cdot \left(0.29441 - Z_{RA} \right) \frac{RT_c}{Pc} \quad (3.5b)$$

$$Z_{RA} = 0.29056 - 0.0877 \cdot \omega \quad (3.5c)$$

3.1.2 Vapor-liquid equilibrium calculation

Vapor-liquid equilibrium for the two phase region is one of the assumptions on which this heat exchanger model is developed (see section 4.2). This calculation is important in this work because it provides the corresponding liquid and vapor compositions, which are necessary in mixing rules in the SRK CEOS (section 3.1.1), enthalpy calculation (section 3.1.3) or are given as states in the system of equations for the heat exchanger (see section 4). For illustration purposes a schematic and idealized representation of a system in vapor-liquid equilibrium is represented in figure 3.1, adapted from Skogestad [2008].

Thermodynamic equilibrium between two or several phases implies that the energy decreases until minimum such that the multiphase system is stable. Depending on the parameters that are kept constant, different energy are minimized at the equilibrium point. When the pressure and temperature of the system is kept constant, Gibbs energy is minimized while if the volume and temperature are fixed, Helmholtz energy is minimized. In this work a finite control volume method (i.e. constant volume of each cell) is applied to model the heat exchanger (see section 4.2), meaning that the Helmholtz energy (eq.3.6) is minimized. A minimum criterion implies that the

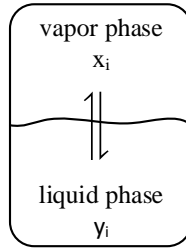


Figure 3.1: Schematic representation of vapor-liquid equilibrium

derivative of the Helmholtz energy is zero (at given volume, temperature and number of moles) $(dA)_{T,V,n} = 0$, condition out of which results that the chemical potentials of the components i are equal in both phases as in equation 3.9 [Haug-Warberg]. This is derived mathematically below.

The variation of the Helmholtz energy of a system is given by the sum of variation of energy in each phase for each component i , the latter given as the product of the chemical potential and the variation of number of moles for component i , as given by equation 3.6.

$$(dA)_{T,V,n} = \sum_i^{NC} \mu_i^V dN_i^V + \sum_i^{NC} \mu_i^L dN_i^L = 0 \quad (3.6)$$

The total number of moles of the system is given by the sum of moles in the vapor phase and the moles in the liquid phase. Since there is no chemical reaction, the total number of moles of the system is constant, which mathematically is formulated as $dN = 0$. Thus the sum of variation in the number of moles in the vapor and liquid phase is zero, as can be seen in equation 3.7.

$$\begin{aligned} dN_i^V + dN_i^L &= dN \\ &= 0 \end{aligned} \quad (3.7)$$

From equation 3.7, the variation of the liquid phase moles can be explicitly expressed as $dN_i^L = -dN_i^V$. By introducing this expression in equation 3.6, the variation in Helmholtz energy can be expressed as a function of the chemical potential of com-

ponent i in each phase, and the variation of number of moles in the vapor phase and equation 3.8 results.

$$(dA)_{T,V,n} = \sum_i^{NC} (\mu_i^V - \mu_i^L) dN_i^V = 0 \quad (3.8)$$

The variation of number of moles cannot be zero in the vicinity of an equilibrium point [Haug-Warberg], resulting the chemical potential of component i in vapor phase is equal to one in liquid phase, according to equation 3.9.

$$dN_i^V \neq 0 \implies \mu_i^L = \mu_i^V \quad (3.9)$$

However, expressing the phase equilibrium by using chemical potentials is not that common in engineering applications and a K-values formulation is preferred, mainly due to its simplicity. The K-value method explicitly relates the compositions of the liquid phase x_i to the composition of the vapor phase y_i via the equilibrium constant K , as shown in equation 3.10 [Skogestad, 2008].

$$y_i = K_i \cdot x_i \quad (3.10)$$

The chemical potential μ_i and K are related via the component fugacity ϕ_i (i.e. deviation of fluid pressure from ideal gas pressure), as it can be observed by looking at equations 3.11 and 3.12a.

$$RT \ln \phi_i = \mu_i(T, p, n) \quad (3.11)$$

Several approaches can be used to determine K , and they can be classified based on the number and type of state equations used to describe their ideal or non-ideal behaviour [Haug-Warberg]:

1. same equation of state is used for both vapor and liquid phases:
 - ideal vapor and liquid phases, when the fugacity in both phases is equal to 1, when K is independent of composition and can be calculated from Henry's

law for diluted mixtures equation

- non-ideal vapor and liquid (eq. 3.12b) when K is dependent of composition and it is calculated as the report of the vapor fugacity to the liquid fugacity, both of them being obtained from the same equations of state

$$K_i = \frac{\phi_i^L}{\phi_i^V} \quad (3.12a)$$

$$\phi_i^V y_i = \phi_i^L x_i \quad (3.12b)$$

2. one equation of state for vapor phase and another for liquid phase:

- ideal liquid phase and non-ideal vapor phase when only the non-ideal behaviour of the vapor phase is modelled and a fugacity model is used for this purpose
- non-ideal vapor and liquid when K is dependent of composition. This time a fugacity model is used for the vapor phase while an activity coefficient (γ_i) model is used for the liquid phase as in equations 3.13

$$\phi_i^V y_i = \phi_i^L (\gamma_i) x_i \quad (3.13)$$

The approach chosen in this work for calculating the K -value is based on using the same equation of state for both phases and expressing K as the report of the vapor fugacity to the liquid fugacity, both of them being determined from Soave-Redlich-Kwong cubic equation of state as mentioned in section 3.1.1.

3.1.3 Enthalpy calculation

It is important in a heat transfer process with phase change to have a good and reliable estimation of the enthalpy of a stream (or phase). The best way to achieve this is to have a non-linear state dependent function to express the non-ideal enthalpy. In this model, the enthalpy of a phase is determined by molar weighting the real enthalpies of each component of the phase, equation 3.14. The compositions in each phase (given here as x_i are used for the molar weighting).

$$H = \sum_i^{NC} x_i \cdot H_i \quad (3.14)$$

The non-ideal enthalpy of each component is calculated by subtracting a departure factor from the ideal enthalpy according to equation 3.15 [Reid et al., 1987].

$$H_i = H_{ideal,i} - H_{SRK,i} \quad (3.15)$$

The departure factor is calculated from an explicit function of temperature and other parameters obtained from the SRK EOS (A and B), equation 3.16 [Reid et al., 1987].

$$H = \frac{1}{BRT} \cdot \left(\frac{AR^2T^2}{P^2} - T \frac{\partial A}{\partial T} \right) \ln \left(\frac{Z}{Z+B} + RT(Z-1) \right) \quad (3.16)$$

The derivative factor $\frac{\partial A}{\partial T}$ is calculated by applying the mixing rules 3.3 which result in equation 3.17.

$$\frac{dA}{dT} = -\frac{R}{2} \left(\frac{0.42747}{T} \right)^{0.5} \sum_i^{NC} \sum_j^{NC} x_i \cdot x_j \cdot \left(m_i \sqrt{\left| A_i \cdot \frac{T_{c,j}}{P} \frac{T^2}{P_{c,j} \cdot R^2} \right|} + m_j \sqrt{\left| A_j \frac{T_{c,i}}{P} \frac{T^2}{P_{c,i} \cdot R^2} \right|} \right) \quad (3.17)$$

The ideal enthalpy is calculated by integrating in respect to temperature a polynomial function of the heat capacity with temperature, equation 3.18 [Reid et al., 1987].

$$H_{ideal,i} = \int_{T_{ref}}^T C_{P,i}(T) dT \quad (3.18)$$

Where $C_{P,i}$ is expressed by equation 3.19 [Reid et al., 1987].

$$C_{P,i} = A_i + B_i T + CT^2 + D_i T^3 + E_i T^4 \quad (3.19)$$

The non-linearity of the enthalpy with temperature for this multicomponent system with phase change is therefore accounted by having a polynomial expression for the specific heat capacity.

3.1.4 Implementation of the thermodynamic model

Two different approaches are available for implementing a thermodynamic model in a process simulation or optimization framework:

1. two layers approach, when the thermodynamic model is solved separately and the numerical results are passed to the process model
2. one layer approach, when the thermodynamic model and the process model are solved simultaneously

The two layer approach is by far the most common between the two, mainly because of its robustness in solving equilibrium equations such as flash calculations. With this method, the roots of the cubic equation of state are determined either analytically or numerically using a root function in a nested subroutine for example [Kamath et al., 2010]. The advantage is that one can include logical preposition to select the correct roots for each phase, considering that the maximum value of the roots corresponds to the vapor phase, while the minimum root corresponds to the liquid phase. The main disadvantage is that at each iteration point the thermodynamic layer must converge to a feasible solution before the process layer is converged. Another disadvantage is that only part of the information of the thermodynamic model (e.g. numerical values of thermodynamic properties) is passed to the process layer, which may lead to numerical problems when the entire model is simulated or optimized [Kamath et al., 2010].

The one layer approach can be used both for simulation and optimization. For optimization purposes, the cubic equations of state 3.2a is added to the process model and passed to the optimizer solver as equality constraints, while the first and second derivative are passed as inequality constraints to assign the roots of CEOS to the correct phase [Kamath et al., 2010]. Both the first and second derivatives of the CEOS are positive when evaluated at the vapor phase root. For the liquid phase, the first derivative is positive while the second derivative is negative.

For simulation purposes, a cubic equation of state can be added for each phase as an additional algebraic equation to the the differential and algebraic system of equations formed by the process model [Skogestad, 2008]. Since the number of states must match the number of equations, two additional states are added to the model: one for vapor compressibility and one for the liquid compressibility. The compressibility factor is added as a state of the model since it cannot be expressed as an explicitly function of the other states. With this method, it is avoided to solve in a subroutine for the roots of the cubic equations of state. The deficiency of this method is that it requires a careful initialization and the user must supply a feasible or very close to feasible point as initial conditions to the solver [Reyes-Lúa et al., 2016].

In this work, it is chosen to add the vapor Z_V and liquid Z_L compressibility as states along with two algebraic equations 3.2a resulted from the thermodynamic model from section 3.1.1. This implementation is presented in chapter 4.

3.2 Process modelling approaches

Two process modelling approaches have been studied for this thesis and they are:

1. Differential and algebraic equations (DAE)
2. Mathematical problem with complementarity constraints (MPCC)

The focal point of this work is the differential and algebraic equations modelling approach while the complementarity constraint is a secondary option briefly studied towards the end of the available time for this thesis because it could potentially offer better answers for the challenges posed by modeling a multicomponent phase change process.

The general formulations for both methods are presented in the next two sections.

3.2.1 Differential and algebraic equations

The general form for differential and algebraic equations is given in equation 3.20. The first equation 3.20a represents the differential part of the system while the equation 3.20b represents the algebraic part [MathWorks, b].

$$y' = f(t, y, z) \tag{3.20a}$$

$$0 = g(t, x, z) \tag{3.20b}$$

Where y' are the differential variables, x is a vector containing the algebraic variables and t is time.

A process model consisting of differential and algebraic equations is commonly seen in chemical processes. Usually the differential equations represent balance equations, such as total mass, component mass, energy or momentum, while the algebraic equations represent conservation laws or constitutive equations [Preisig, 2016].

The model formulated with differential and algebraic equations is presented in section 4.6.1, while the solver used to solve the system is presented in section 5.1.

3.2.2 Mathematical problem with complementarity constraints

Formulating a process model as a mathematical problem with complementary constraints (MPCC) arises from reformulating a mixed integer nonlinear problem (MINLP) which in turn derives from reformulating a generalized disjunctive optimization problem.

The generalized disjunctive optimization problem is given in equation 3.21 [Herty and Steffensen, 2012]. This formulation is of interest for this work since disjunctions could account for the change of the model topology when a phase change happens. For simulation purposes, only the equality and inequality constraints form the process model, while the objective function $\Psi(x)$ does not have a significance. Therefore it can be set as a constant values (e.g. $\Psi(x) = 0$) such that it does not alter the results obtained.

$$\begin{aligned}
 & \min_{x,Y} \Psi(x) + \sum_{k \in K} b_k \\
 & \text{s.t. } H_0(x) = 0 \\
 & \quad G_0(x) \geq 0 \\
 & \quad \forall i \in D_k \begin{bmatrix} Y_{i,k} \\ H_{i,k}(x) = 0 \\ G_{i,k} \geq 0 \\ b_k = \gamma_{i,k} \end{bmatrix}, \quad k \in K \\
 & \quad \Omega(Y) = \text{True} \\
 & \quad Y_{i,k} \in \{\text{True}, \text{False}\} \\
 & \quad K = \{1, \dots, m\} \\
 & \quad D_k = \{1, 2, \dots, n_k\}
 \end{aligned} \tag{3.21}$$

Where x is a continuous variable, $Y_{i,k}$ is a discrete decision variable and b_k represents a continuous scalar that is equal to a fixed charge $\gamma_{i,k}$. The problem has a total of m disjunctions each of them containing n_k terms. $\forall_{i \in D_k}$ is a logical operator that signifies an *exclusive or*, meaning that when the decision variable $Y_{i,k}$ is set to *True*, the i -th constraints become valid [Herty and Steffensen, 2012].

To avoid a model with disjunctions and decision variables, the generalized disjunctive optimization problem is converted into a mixed integer nonlinear program (MINLP) with binary constraints, given in equation 3.22 [Herty and Steffensen, 2012].

$$\begin{aligned}
 & \min f(x) \\
 & \text{s.t. } g(x) = 0 \\
 & \quad g(x) \geq 0 \\
 & \quad x_1 \in \{0, 1\}^p
 \end{aligned} \tag{3.22}$$

However, the binary variables are not easy to handle and therefore the binary constraints are relaxed and replaced with complementarity constraints. Thus, a mathematical problem with complementarity constraints (MPCC) with continuous variables as in equation 3.23 is constructed [Herty and Steffensen, 2012].

$$\begin{aligned}
 & \min f(x) \\
 & \text{s.t. } g(x) = 0 \\
 & \quad g(x) \geq 0 \\
 & \quad 0 \leq x_1 \perp 1 - x_1 \geq 0
 \end{aligned} \tag{3.23}$$

The possibilities available in the literature for solving a mathematical problem with complementarity constraints are discussed in section 5.2. How this formulation could be applied for dynamic modelling of a process, is presented in section 5.2.1.

Chapter 4

Heat exchanger model

In section 2.1 it was discussed that the scope of this work is to develop a robust heat exchanger model which is representative for all the heat exchangers of the reliquefaction process described by Nekså et al. [2010], regardless of their operating conditions or if a phase transition happens or not. The main challenge for a model for a phase change process is to handle the appearance and disappearance of phases which affect the topology of the model, in addition to the infeasibility of the vapor-liquid equilibrium condition in the single phase.

Before formulating a mathematical model of a phase change heat exchanger, the phenomena behind it should be better understood. This is achieved by constructing the phase envelope in section 4.1, discussing the model assumption in section 4.2, illustrating the cell heat exchanger in section 4.3, presenting the states in section 4.4 and finally the equations in section 4.6. All the discussion in this chapter is made in connection to the schematic representation of the countercurrent heat exchanger from figure 2.1.

4.1 Phase envelope

The phase envelope is constructed to have a better overview of the conditions in which each phase (e.g. vapor, vapor-liquid and liquid) exist. The phase envelope from figure

4.1 is represented for a mixed refrigerant fluid with the composition from table 2.2 in section 2.1 since this is the stream that is modelled as phase changing fluid (see section 4.2).

The pressure and temperature data is obtained from the commercial process simulator Aspen Plus[®] and their relation is shown in figure 4.1. The red curve represents dew points (when the first drop of liquid condenses) and it is the boundary between the vapor and vapor-liquid regions. The blue curve represents the bubble point (when the first bubble of vapor evaporates) and it is the boundary between the vapor-liquid and the liquid regions. It can be observed that the two curves do not meet. That point corresponds to the critical point, where there is no phase boundary.

Lets consider that initially the system is in single vapor phase. As heat is removed from the fluid, the dew point is reached and liquid droplets start condensing and the system transitions into two phase region. As more heat is removed from the cells, the bubble point is reached and the system transitions into liquid region.

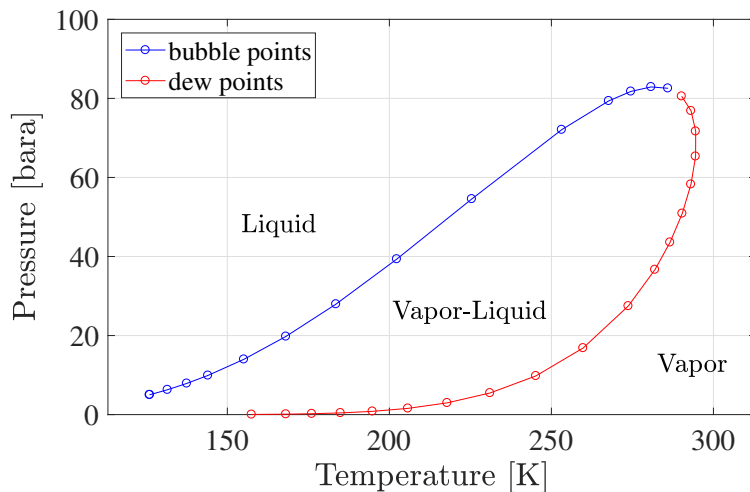


Figure 4.1: Phase envelope of a nitrogen-methane-ethane-propane mixture with composition from table 2.2

In relation to the phase envelope from figure 4.1 and the countercurrent heat exchanger from figure 2.1, there are nine possible transitions for one stream and they are represented in table 4.1.

Table 4.1: Possible phase transitions

Case number	Inlet	Outlet
1		V
2	V	L-V
3		L
4		V
5	L-V	L-V
6		L
7		V
8	L	L-V
9		L

4.2 Assumptions

This section presents the assumptions and considerations taken into account in the model. The differences compared to the project are vapor-liquid equilibrium only in the multiphase region and modelling the outlet flow with a valve equation.

Assumptions & other considerations:

- dynamic model to capture as many process characteristics as possible
- the heat exchanger is discretize into M cells (or lumps) via finite control volume method
- only one of the two streams of the heat exchanger is represented in the model, since it is firstly desired to find a way to overcome the challenge brought by appearance and disappearance of phases
- given total transferred heat to account for the second stream of the heat exchanger that is not represented in the model
- equal transferred heat to/from each cell, this is equivalent to having the second stream at constant temperature
- no heat loses
- vapor-liquid equilibrium only for the two phase region
- different topology (set of equations) in each phase region

- perfect mixing in each cell, resulting in constant conditions (temperature, pressure, composition etc.) in each cell individually
- linear averaging rules to calculate different mixing properties in the two phase region
- no slip between the vapor and liquid phases (i.e. they have the same velocities)
- SRK equations of state incorporated in the model with liquid and vapor compressibilities as states of the model
- neglected wall heat capacity
- maximum 5 mbar pressure drop
- a valve equation gives the outlet flow of each cell

4.3 Model illustration

In order to formulate a model, one should first have a clear picture of the process phenomena. In figure 4.2, it is illustrated how the finite control volume method is discretizing the heat exchanger into M cells of equal and constant volume. As a stream, illustrated as hot in the figure, with the specified inlet conditions: flow $F_{H,0}$, composition z_H , pressure $P_{H,in}$ and enthalpy $H_{H,in}$ as a function of temperature $T_{H,in}$ enters the first cell, heat Q is removed thus changing the conditions in each cell. In each cell, the temperature, pressure and compositions are considered constant; meaning that the temperature profile along the heat exchanger is a discrete one, as it is showed in the bottom part of figure 4.2. However, as the number of cells increases, the temperature profile will better match the real profile given by the red curve.

Remark: The figure is not quantitative and is for illustration purpose only. The red curve is not drawn based on process or simulation data, and its actual shape is subject to change.

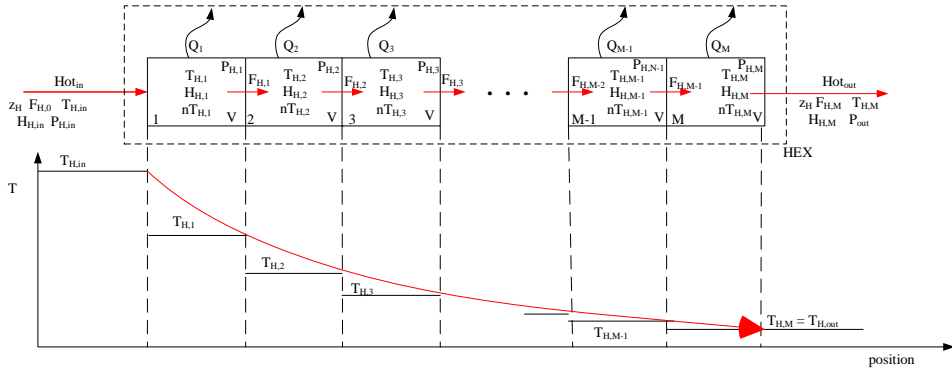


Figure 4.2: Heat exchanger lumped model

4.4 Inputs and states

Before formulating the model equations it is necessary to know what the model should be solved for, or in other words, what are the inputs (given variables) and the outputs (unknown variables). It should be restated that the model is not developed with a final design scope but for process optimization, and hence a few design parameters (volume and pressure drop) are considered given. The model inputs are the following:

- inlet stream process conditions: molar flow, temperature, pressure, composition
- design parameters such as volume, or pressure drop
- number of cells
- heat load (further discussed in section 4.6.1.4)

The outputs are related to the states of the model since the latter offer the minimum information about the current conditions of a process required to know how the process will evolve [Preisig, 2016]. Table 4.2 presents the states of the model together with their total number in each cell. The states in one cell are: one total holdup, NC components holdup, one internal energy, one temperature, NC liquid and vapor compositions, one liquid volume, one vapor compressibility, one liquid compressibility and one pressure.

The total number of states for the lumped HEX is obtained by multiplying the number of states per cell with the number of cells, M . It should be pointed out that if the stream remains in vapor phase then the states corresponding to the liquid (e.g. liquid volume, liquid composition, liquid compressibility) do not have a physical meaning and in this case they are considered as dummy variables used to keep the same number of states in the model regardless of the number of phases. This will also be discussed in the model equations in section 6.1.3. Of course, vice versa is valid for the case when the stream is only liquid, case when the vapor states are passed as dummy variables.

Table 4.2: Model States

State	Symbol	Units	Number
Total holdup	N	kmol	M
Component holdup	n	kmol	$M \cdot NC$
Internal energy	U	$\frac{\text{MJ}}{\text{kmol}}$	M
Temperature	T	K	M
Liquid composition	x	$\frac{\text{kmol}}{\text{kmol}}$	$M \cdot NC$
Vapor composition	y	$\frac{\text{kmol}}{\text{kmol}}$	$M \cdot NC$
Liquid volume	V_L	m^3	M
Vapor compressibility	Z_G	–	M
Liquid compressibility	Z_L	–	M
Pressure	P	MPa	M

4.5 Model stiffness

Stiffness represents a property of the mathematical model and it is associated with having slow and fast changes in the model functions or states [Shalashilin and Kuznetsov, 2003]. In this specific model of the heat exchanger, the mechanical momentum correlated to a change in pressure has a much more faster variation compared to the temperature variation which is slow. As a result the model has different time scales and it

is stiff. It is important to know if the process model is stiff or not in order to choose a proper numerical method to capture all the required characteristics of phase change process. The solver chosen is further presented in section 5.

4.6 Model formulations

As mentioned in section 2.1, different approaches to handle the appearance and disappearance of phases have been studied together with the infeasibility of the vapor-liquid equilibrium in single phase.

The approaches can be generalized into two main categories:

1. using different topologies in each phase from the phase envelope figure 4.1
 - 1.1. having one different sets of differential and algebraic equations for each phase region and a conditional algorithm to switch between them
 - 1.2. having a single set of equations with complementarity constraints to eliminate the terms for the non-existent phase
2. alternatives formulations which imply using an artificial mathematical mean to have a single set of differential and algebraic equations, while still being able to satisfy the VLE conditions even if one of the phases is not present
 - 2.1. cancelling the compressibility algebraic equations for the non-existing phase
 - 2.2. adding trace components to keep the system always in two phase region
 - 2.3. giving fictions compositions to the non-existent phase

Each of them is discussed in the next sections.

4.6.1 Different set of equations for each phase region

This approach is the most studied in this work and is therefore the most amply presented and discussed. Here the vapor-liquid equilibrium algebraic equation is used only for the two phase region, which means that the topology of the model (type of

equations and number of valid states) will change with the appearance and disappearance of phases. The first step taken in formulating the model is to propose one set of equations without VLE in the single phases and a different set, with VLE for the multi-phase region, followed by implementing logical propositions to switch between topologies.

This section describes the equations needed to determine the conditions for each cell of the heat exchanger. First, the model is written for the two phase region where the VLE condition has a feasible solution and subsequently the two-phase model is reduced to a single phase model. A representative cell j of the heat exchanger along with the input (from cell $j-1$) and outputs is illustrated in figure 4.3. The equations for all phases are formulated for cell $j = 1 \dots M$ and components $i = 1 \dots NC$. The model contains both differential and algebraic equations, and thus falls into the category of differential and algebraic system of equations (DAE). The next two sections 4.6.1.1 and 4.6.1.2, present the equations used for both single and two phase region.

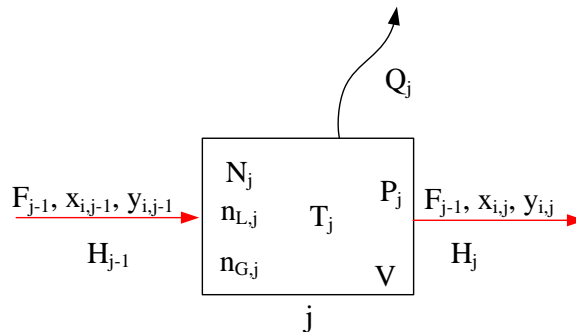


Figure 4.3: Representative cell of the heat exchanger

4.6.1.1 Equations for two phase region

The base for the two phase region model is a dynamic flash calculations adapted from Skogestad [2008]. An important note should be made in connection with the number of outlet streams of a classic flash tank and this heat exchanger cell model. The design of a flash tank implies two isolated vapor and liquid outlet streams. As the heat exchanger

in this work is modelled as a series of lumps and perfect mixing is assumed, each cell has only one outlet, which can have either multiphase or single phase condition.

Compared to the specialization project, a detailed derivation of of the differential model equations is presented in this work. The dynamic equations are written considering that the accumulation of a quantity Θ in time in a cell is determined by the sum of two terms. The first of them is the difference of the inlet and outlet of the respective quantity through the system boundaries. The second of them is the difference between the generated and loss of quantity internally in the system according to the general balance equation 4.1 [Skogestad, 2008]. The generation and loss term are specific for systems with chemical reactions and as a consequence are not needed in a heat exchanger model and the general balance equation can be simplified. The two phase model is given in equation 4.10.

$$\begin{aligned}\frac{d\Theta}{dt} &= \Theta_{in} - \Theta_{out} + \Theta_{generated} - \Theta_{Loss} \\ &= \Theta_{in} - \Theta_{out}\end{aligned}\tag{4.1}$$

The model has a dynamic mass (eq. 4.2), component (eq. 4.5) and energy balances (eq. 4.9). A dynamic momentum equation accounting for the pressure drop is missing since not enough design and process data is available.

In equation 4.2 the accumulation of mass (given as total holdup N) in time in a cell j is given by the difference between the inlet and the outlet flows.

$$\frac{dN_j}{dt} = F_{j-1} - F_j\tag{4.2}$$

Remark: The flow pattern for the multiphase region is modelled considering perfect mixing (i.e. vapor perfectly dispersed into liquid) and same phase velocities. With this assumption, one mass balance equations can be written to account for both the vapor and liquid phases. The opposite alternative corresponds to the assumptions of separated flow pattern (stratified or annular), when the vapor and liquid phases are not mixed and thus can be seen as two different systems separated by a vapor-liquid inter-

face. In the latter case, one mass balance equation should be written for each vapor and liquid phase respectively and also the mass transferred through the vapor-liquid interface should be included as a outlet stream [Bratland, 2013].

In equation 4.5, the accumulation of component i (given as component holdup n_i) in time in a cell j is determined by the inlet and outlet flows of the respective component, modelled considering perfect mixing rule. Thus the total flow of component i is given by the sum of its vapor and liquid flows which in turn are expressed as a product of the total flow, vapor fraction or liquid fraction respectively and concentration in the respective phase. It should be noted out that the liquid and vapor fraction sum to 1 and thus the liquid fraction can be expressed as a explicit function of the vapor fraction 4.4. The derivation of 4.5 is given in equations 4.3 and 4.4. General component mass balance is:

$$\frac{dn_{i,j}}{dt} = F_{i,j-1} - F_{i,j} \quad (4.3)$$

Where the inlet F_{j-1} and outlet F_j flows for each component i can be expressed as:

$$\begin{aligned} F_i &= F_{V_i} + F_{L_i} \\ &= F_i \cdot v + F_i \cdot l \\ &= F_i \cdot v + F_i \cdot (1 - v) \\ &= F \cdot y_i \cdot v + F \cdot x_i \cdot (1 - v) \end{aligned} \quad (4.4)$$

By combining equations 4.3 and 4.4 the final form of the component mass balance results in equation 4.5:

$$\frac{dn_{i,j}}{dt} = F_{j-1} \cdot v_{j-1} \cdot x_{i,j-1} + F_{j-1} \cdot (1 - v_{j-1}) \cdot y_{i,j-1} - F_j \cdot v_j \cdot x_{i,j} - F_j \cdot (1 - v_j) \cdot y_{i,j} \quad (4.5)$$

Another form of the component mass balance can be formulated on the basis that the overall mixture composition does not change along the heat exchanger (due to perfect mixing and no slip assumptions). Thus, there is no need to used an explicit formulation of the vapor and liquid compositions and fractions, as it was implemented in the specialization project. The two formulations are identical as long as the overall

composition is constant. The form 4.5 is preferred since it can better capture the dynamics of the system and is more adequate if this heat exchanger model would be used to simulate all refrigeration cycle or if a slip relation (i.e. different phase velocities) is introduced, cases where overall composition is subject to change.

The energy balance is given in equation 4.9 where the accumulation of energy in time in a cell j is determined by difference between the heat load of inlet flow, heat load of the outlet flows and the heat that is removed from each cell Q . The heat load of the inlet and outlet flows are modelled using the same perfect mixing rules as in the component balance i . No heat is lost to the environment. The general energy balance for this cell HEX model is given by equation 4.6.

$$\frac{du_j}{dt} = h_{j-1} - h_j - Q_{cell} \quad (4.6)$$

Where in inlet h_{j-1} and outlet h_j heat rate is given by equation 4.7 where the phase specific molar enthalpy H is calculated based on the thermodynamic model presented in section 3.1.3.

$$\begin{aligned} h &= h_V + h_L \\ &= F_V \cdot H_V + F_L \cdot H_L \\ &= F \cdot v \cdot H_V + F \cdot l \cdot H_{Lj} \\ &= F \cdot v \cdot H_V + F \cdot (1 - v) \cdot H_L \end{aligned} \quad (4.7)$$

The heat removed is considered to be equal for each individual cell and is equal to the total heat removed divided to the number of cells according to equation 4.8.

$$Q_{cell} = \frac{Q_{HEX}}{M} \quad (4.8)$$

By combining equations 4.6, 4.7 and 4.8 the final form of the energy balance result

in equation 4.9.

$$\frac{du_j}{dt} = F_{j-1} \cdot v_{j-1} \cdot H_{V_{j-1}} + F_{j-1} \cdot (1 - v_{j-1}) \cdot H_{L_{j-1}} - F_j \cdot v_j \cdot H_{V_j} - F_j \cdot (1 - v_j) \cdot H_{V_j} - Q_{cell} \quad (4.9)$$

In addition to dynamic balance equations the model also has the following algebraic equations:

- total internal energy (eq. 4.10d), derived from the the thermodynamic definition of enthalpy
- component holdup (eq. 4.10e), considering it as linear combination of total holdup and component composition
- vapor-liquid equilibrium (eq. 4.10f), as described in section 3.1.2
- total holdup (eq. 4.10g), considering perfect mixing of phases
- vapor compresibility (eq. 4.10h), as described in section 3.1.1
- liquid compresibility (eq. 4.10i), as described in section 3.1.1
- composition consistency (eq. 4.10j).

The equations for calculating the vapor and liquid compresibility are hereby incorporated in the set of equations and therefore are not calculated in a subroutine. This is equivalent to having a single layer approach where both the process and the thermodynamic equations are solved simultaneously which means that all equations will converge to a solution at the same time whereas in the case where they are solved separately, the thermodynamic model has to converge first in order to provide the thermodynamic properties required by the process equations [Reyes-Lúa et al., 2016].

For a better overview of the model, the algebraic equations are presented together with the differential equations in 4.10.

$$\frac{dN_j}{dt} = F_{j-1} - F_j \quad (4.10a)$$

$$\frac{dn_{i,j}}{dt} = F_{j-1} \cdot v_{j-1} \cdot x_{i,j-1} + F_{j-1} \cdot (1 - v_{j-1}) \cdot y_{i,j-1} - F_j \cdot v_j \cdot x_{i,j} - F_j \cdot (1 - v_j) \cdot y_{i,j} \quad (4.10b)$$

$$\frac{du_j}{dt} = F_{j-1} \cdot v_{j-1} \cdot h_{V_{j-1}} + F_{j-1} \cdot (1 - v_{j-1}) \cdot h_{L_{j-1}} - F_j \cdot v_j \cdot h_{V_j} - F_j \cdot (1 - v_j) \cdot h_{L_j} - Q_{cell} \quad (4.10c)$$

$$u_j + P_j \cdot V_{cell} - H_{V_j} \cdot n_{V_j} - H_{L_j} \cdot n_{L_j} = 0 \quad (4.10d)$$

$$n_{i,j} - x_{i,j} \cdot n_{L_j} - y_{i,j} \cdot n_{V_j} = 0 \quad (4.10e)$$

$$y_{i,j} - K_{i,j} \cdot x_{i,j} = 0 \quad (4.10f)$$

$$N_j - n_{L_j} - n_{V_j} = 0 \quad (4.10g)$$

$$Z_{V_j}^3 + Z_{V_j}^2 + Z_{V_j} \cdot (A_{V_j} - B_{V_j} - B_{V_j}^2) - A_{V_j} \cdot B_{V_j} = 0 \quad (4.10h)$$

$$Z_{L_j}^3 + Z_{L_j}^2 + Z_{L_j} \cdot (A_{L_j} - B_{L_j} - B_{L_j}^2) - A_{L_j} \cdot B_{L_j} = 0 \quad (4.10i)$$

$$\sum_{i=1}^{NC} x_{i,j} - \sum_{i=1}^{NC} y_{i,j} = 0 \quad (4.10j)$$

The last algebraic equation 4.10j is derived considering that the sum of all compositions in each phase is equal to unity (eq. 4.11) and one can subtract the latter from the former.

$$\sum_{i=1}^{NC} x_{i,j} = 1 \quad (4.11a)$$

$$\sum_{i=1}^{NC} y_{i,j} = 1 \quad (4.11b)$$

4.6.1.2 Equations for single phase region

On the grounds that heat exchanger in a refrigeration cycles can be operated in all phase regions, a robust model should be valid in all possible phases. However, as mentioned before, the VLE condition does not provide a feasible solution in either of the single phases and thus the topology of the model must be changed, implying that two phase model must be reduced to a single phase model. This is done by eliminating the VLE condition and neglecting the variables corresponding to the non-existing phase in the process and thermodynamic equations.

The set of equations for the vapor are presented in equation 4.12.

$$\frac{dN_j}{dt} = F_{j-1} - F_j \quad (4.12a)$$

$$\frac{dn_{i,j}}{dt} = F_{j-1} \cdot y_{i,j-1} - F_j \cdot y_{i,j} \quad (4.12b)$$

$$\frac{du_j}{dt} = F_{j-1} \cdot H_{V_{j-1}} - F_j \cdot H_{V_j} - Q_{cell} \quad (4.12c)$$

$$u_j + P_j \cdot V_{cell} - H_{V_j} \cdot n_{V_j} = 0 \quad (4.12d)$$

$$y_{i,j} - \frac{n_{i,j}}{N_j} = 0 \quad (4.12e)$$

$$Z_{V_j}^3 + Z_{V_j}^2 + Z_{V_j} \cdot (A_{V_j} - B_{V_j} - B_{V_j}^2) - A_{V_j} \cdot B_{V_j} = 0 \quad (4.12f)$$

$$N_j - n_{V_j} = 0 \quad (4.12g)$$

The corresponding set for the liquid phase are presented in equations 4.13.

$$\frac{dN_j}{dt} = F_{j-1} - F_j \quad (4.13a)$$

$$\frac{dn_{i,j}}{dt} = F_{j-1} \cdot x_{i,j-1} - F_j \cdot x_{i,j} \quad (4.13b)$$

$$\frac{du_j}{dt} = F_{j-1} \cdot H_{L_{j-1}} - F_j \cdot H_{L_j} - Q_{cell} \quad (4.13c)$$

$$u_j + P_j \cdot V_{cell} - H_{L_j} \cdot n_{L_j} = 0 \quad (4.13d)$$

$$x_{i,j} - \frac{n_{i,j}}{N_j} = 0 \quad (4.13e)$$

$$Z_{V_j}^3 + Z_{V_j}^2 + Z_{V_j} \cdot (A_{V_j} - B_{V_j} - B_{V_j}^2) - A_{V_j} \cdot B_{V_j} = 0 \quad (4.13f)$$

$$N_j - n_{L_j} = 0 \quad (4.13g)$$

This procedure results however in a change of the number of equations compared to the two phase region. If the phase location is known *a priori*, then different solvers can be used for the single and two phase formulations respectively. However, one of the premises of this model is that the phase transition is not known *a priori*, meaning that it is not known beforehand what model formulation to use. One must first detect a phase transition and then switch between the set of equations. In order to have a single robust implementation of the model, the switch between the two formulations should be handled automatically by the solver which requires that the number of states and equations is the same in both formulations. Therefore, dummy states and equations are introduced to account for the non-existent phase and keep the number of equations and states constant. It should be remarked that these dummy variables can have any numerical values without influencing either the process or the thermodynamic model. A possibility to create the dummies would be to attribute the value zero to all variables without a physical meaning. However this will create discontinuities in the model and it would make it unsolvable since convergence is not reached [Sahlodin et al., 2016]. To avoid discontinuities, the non-physical variables are initialized to values different than zero, given by the user. This could for example correspond to the dew or bubble points conditions such that a continuous transitions is made from single to multiphase region or vice versa.

The dummy equations introduced for vapor phase are given in equations 4.15.

$$x_{i,j} - x_{i,0} = 0 \quad (4.14a)$$

$$V_{L,j} - V_{L,0} = 0 \quad (4.14b)$$

$$Z_{L,j} - Z_{L,0} = 0 \quad (4.14c)$$

The reciprocal dummy equations and states should be introduced if the fluids are only in liquid state are given in equations 4.15.

$$y_{i,j} - y_{i,0} = 0 \quad (4.15a)$$

$$V_{V,j} - V_{V,0} = 0 \quad (4.15b)$$

$$Z_{V,j} - Z_{V,0} = 0 \quad (4.15c)$$

Thus the full model for the single vapor region consists of equations 4.12 and 4.15a, while the full model for the single liquid region consists of equations 4.13 and 4.14a.

4.6.1.3 Additional algebraic equations needed for all phases

It can be observed that all sets of equations 4.10, 4.12 and 4.13 equations have some unknowns variables which are not given as states and other algebraic equations are needed to compute them. These variables are not given as states since they can be explicitly expressed a function of the states from table 4.2. As contrary to the project work [Zotică, 2016], where the flow was "controlled" with two imaginary P-controller, the outlet flow of this work is modeled using the valve equation 4.16a. This can be imagined as there is a valve that controls the outlet flow of each cell, which of course does not have a physical meaning in this process and it is used for modelling purposes only. The pressure is assumed to have a linear drop along the heat exchanger, while the valve coefficient k is calculated by dividing the inlet flow F_0 to the total pressure drop ΔP and the number of times the pressure changes: M (number of cells) plus two (inlet and outlet streams).

The outlet flow of cell j is given by a linear combination between the pressure in cell j and pressure in cell $j+1$ respectively, accordingly to equation 4.16a.

$$F_j = k \cdot \sqrt{P_j - P_{j+1}} \quad j \in \{1 \dots M-1\} \quad (4.16a)$$

$$k = \frac{F_0}{\Delta P \cdot (M+2)} \quad (4.16b)$$

This equations can only be written for $j \in \{1 \dots M-1\}$ cells since at $j = M$ the term P_{j+1} becomes P_{M+1} , and according to table 4.2, the number of values for the pressure vector has to be equal to the number of cells. This means that pressure P_{M+1} of the outlet stream is not part of the states and therefore has to be given. Hence, the outlet pressure is given as $P_{M+1} = P_{out}$ equal to the difference between the inlet pressure P_{in} of the stream and the assumed pressure drop ΔP . The outlet flow of the last cell is given in equation 4.17.

$$F_M = k \cdot \sqrt{P_M - P_{out}} \quad (4.17a)$$

$$P_{out} = P_{in} - \Delta P \quad (4.17b)$$

Other unknowns variables are computed from the phase distribution. The volume of a cell is calculated with equation 4.18 by dividing the total heat exchanger volume V_{HEX} to the number of cell M since the volume of each cell is constant (finite control volume method).

$$V_{cell} = \frac{V_{HEX}}{M} \quad (4.18)$$

The vapor volume in each cell is given in equation 4.19, where it is calculated as the difference between the cell volume from equation 4.18 and the liquid volume V_L which is one of the states of the process from table 4.2.

$$V_{V_j} = V_{cell} - V_{L_j} \quad (4.19)$$

The vapor holdup in each cell n_V is determined with equation 4.20 by dividing the

the vapor volume V_V to the molar volume of the vapor phase V_{m_V} obtained from equation 3.4 of the SRK thermodynamic model.

$$n_{V_j} = \frac{V_{V_j}}{V_{m_{V,j}}} \quad (4.20)$$

The liquid holdup in each cell n_V is determined with equation 4.21 by dividing the the liquid volume V_V to the molar volume of the liquid phase V_{m_L} obtained from equation 3.5 of the SRK thermodynamic model.

$$n_{L_j} = \frac{V_{L_j}}{V_{m_{L,j}}} \quad (4.21)$$

The vapor fraction of each cell is calculated by dividing the vapor holdup to the total holdup, according to equation 4.22.

$$v_j = \frac{n_{V_j}}{N_j} \quad (4.22)$$

It can be observed that when the vapor fraction is equal to unity and the system is in vapor phase, the vapor holdup is equal to the total holdup while liquid volume is zero and the gas volume is equal to the cell volume. Similarly, when the vapor fraction is zero and the system is in liquid phase, the liquid holdup is equal to the total holdup while the vapor volume is zero and the liquid volume is equal to the cell volume.

The K-values needed for the VLE calculation are computed as the ratio of liquid fugacity ϕ_L and vapor fugacity ϕ_V , both obtained from equation 3.3a of the SRK the thermodynamic model.

$$K_{i,j} = \frac{\phi_L}{\phi_V} \quad (4.23)$$

4.6.1.4 Time dependent heat function

In the specialization project [Zotică, 2016] the removed heat Q_{HEX} was considered constant in time. Since the model is dynamic, it is desired to predict how the process would behave when the heat changes with time. For this reason, the time dependent heat function from algorithm 1 is implemented in this work. The algorithm considers that

no heat is removed from the heat exchanger before time t_1 . Then the removed heat is increased until time t_2 is reached, after which the heat is constant and is equal to the total value of the heat removed, equivalent to the maximum value of the transferred heat. A physical interpretation of this algorithm is that until time t_1 , the system is at a steady-state and all cells have conditions equal to the ones of inlet stream. After the point in time t_1 , the heat transferred from each cell increases and the temperature drops in each cell and as a consequence all other cell conditions are changed. In other words, the heat exchanger is simulated as it would have a switch on button for starting removing heat, which in practice would correspond to a start-up scenario

The parameters t_1 and t_2 can be receive any values depending on what characteristics of the process should be represented, as long as the following inequalities constraints are satisfied: $0 \leq t_1 \leq t_2$. The shape of the heat function is dependent on parameters t_1 and t_2 as indicated in figure 4.4. It can either be a step function when $t_1 = t_2$ as shown in figures 4.4a and 4.4b, or a ramp function when $t_1 \neq t_2$ as illustrated in figure 4.4c. The difference between figures 4.4a and 4.4b is that in the former heat is removed from the initial time point $t_0 = 0$ while in the latter a different time point is chosen. This scenario could be equivalent to a start-up of the refrigeration cycle which would correspond to figure 4.4a or to a disturbance in flow of the second stream of the process (modelled here as the total removed heat) that lasts between time t_1 and t_2 as in figures s 4.4a and 4.4b. In figure 4.4 the values for are given for arbitrarily for illustrations purposes and are not relevant to the process.

Algorithm 1: Heat ramp function

```

if  $t \leq t_1$  then
  |  $Q_{HEX} = 0$ ;
else if  $t_1 < t < t_2$  then
  |  $Q_{HEX} = \frac{Q_{max}}{t_2 - t_1} (t - t_1)$ ;
else
  |  $Q_{HEX} = Q_{max}$ ;
end

```

4.6.1.5 Comparison between the single and multiphase sets of equations

- Mass balance: equations 4.12a, 4.2 and 4.13a are the same

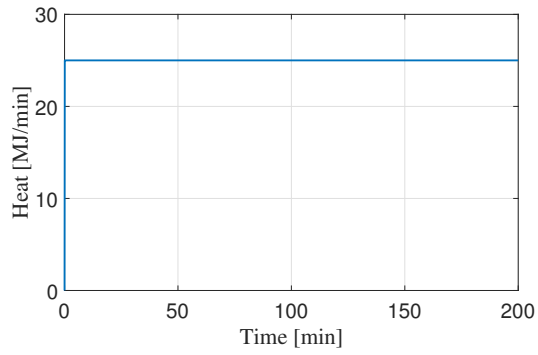
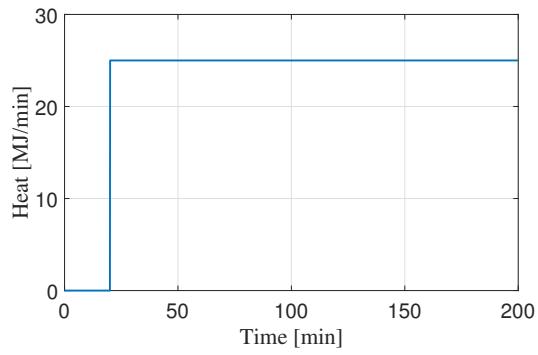
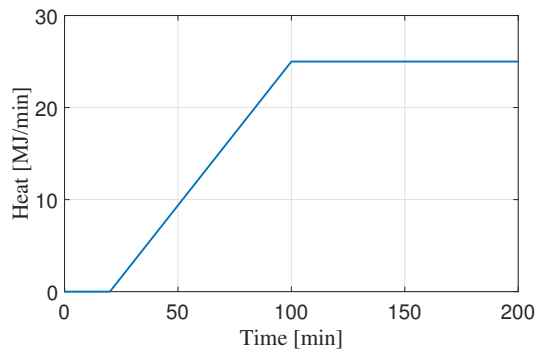
(a) $t_1 = t_2 = t_0$ (b) $t_1 = t_2 \neq t_0$ (c) $t_1 \neq t_2 \neq t_0$

Figure 4.4: Time dependent heat function

- Component mass balance: equation 4.12b is derived from 4.5 if the vapor fraction v_j is 1, while equations 4.13b is derived from 4.5 if the vapor fraction is 0
- Energy balance: equations 4.12c is derived from 4.9 if the vapor fraction is 1, while equations 4.13c is derived from 4.12c if the vapor fraction is 0
- Internal energy: equation 4.12d is derived from equation 4.10d if the liquid hold up is 0, while equation 4.13d is derived if the vapor hold up is 0
- Component holdup: equation 4.12e is derived from equation 4.10e if the liquid holdup is 0, while equation 4.13e is derived if the vapor hold up is 0
- VLE: is written only for two phase region in equation 4.10f, while for single phase region the dummy equations 4.14a or 4.15a are given
- Holdup: equation 4.12g is derived from equation 4.10g if the liquid holdup is 0
- Gas compressibility: equations 4.12f and 4.10h are the same and taken from SRK, while equation 4.15c is a dummy equation
- Liquid compressibility: equation 4.10i and 4.10i are the same and taken from SRK, while equation 4.14c is a dummy equation
- Compositions consistency: for two phase is given as sum of vapor composition minus sum of liquid composition, as in equation 4.10j, while for the single phase dummy equations 4.14b and 4.15b are used.

4.6.1.6 Switching algorithms

Two different options are implemented to change from one set of equations to the other. The first option implies comparing the temperature of each cell with the dew point or bubble point of the mixture. If the cell temperature is above the dew point, then the system is in vapor phase and the first set of equations is selected. If the cell temperature is smaller or equal to the the dew point and larger or equal to the bubble point, then the system is in vapor-liquid phase and the second set of equations is selected , otherwise, the third set is selected. The *elseif* condition is verified once the *if* conditions becomes false. In addition, it is known before hand that temperature decreases, hence the temperature of the cell is lower than the dew point in the second branch of the algorithm. Therefore, it is necessary only to compare the temperature of the cell with the bubble point, and there is no need to include the dew temperature in the *elseif* condition.

One aspect of comparing the cell temperature with either the dew or bubble point is that they are both implicitly dependent of the cell conditions (composition and pressure), thus the dew point and bubble points is different, in each cell, even if not by much. Thus the dew point and bubble points should be determined in each cell which represents a disadvantage since these calculations should be implemented in a nested subroutine which might negatively affect the convergence to a solution. In addition having a nested subroutine increases the calculation time. However, if one assumes constant composition and a small pressure drop, only a small variation of dew point and bubble point is expected along the heat exchanger and they can be approximated as constant. A detailed explanation of how this algorithm was implemented is presented in section 6.1. The pseudocode is presented in algorithm 2.

Algorithm 2: Selection of equations set based on cell temperature

```

if  $T_j - T_{dew} \geq tol$  then
  | system is in vapor region  $\implies$  solve equations 4.12 ;
else if  $T_j - T_{bub} \geq tol$  then
  | system is in vapor-liquid region  $\implies$  solve equations 4.10;
else
  | system is in liquid region  $\implies$  solve equations 4.13 ;
end

```

The second option implies comparing the liquid or the vapor volume of the cell with a given tolerance. If the liquid volume is below the given tolerance, then the system is in vapor phase and the first set of equations is selected. If the vapor volume is below the given tolerance, then the system is in liquid phase and the second set of equations is selected, otherwise set three is selected. The pseudocode is presented in algorithm 3.

Algorithm 3: Selection of equations set based on phase temperature

```

if  $V_{L_j} \leq tol$  then
  | system is in vapor region  $\implies$  solve equations 4.12 ;
else if  $V_{V_j} \leq tol$  then
  | system is in liquid region  $\implies$  solve equations 4.13 ;
else
  | system is in vapor-liquid region  $\implies$  solve equations 4.10;
end

```

4.6.1.7 Dew and bubble point calculation

Algorithm 2 uses the dew and bubble point as references to switch the model formulations and thus the method for calculating them are described in this section. Both dew and bubble point calculations are based on a flash calculation and the assumption that vapor-liquid equilibrium condition is valid. For a given composition in the vapor or liquid phases, a single pair pressure-temperature can exist at the dew or bubble points respectively [Skogestad, 2008].

The dew point of fluid in vapor phase represents the point when the first drop of liquid is formed, or in other words when the vapor begins to condense. The calculation uses a given vapor composition and can be made at given temperature when the pressure is adjusted until the vapor begins to condense, or the other way around, at given pressure when the temperature is adjusted [Skogestad, 2008]. In both cases, the unknown state (temperature or pressure) is adjusted until the conservation law for the liquid phase (eq. 4.24a is fulfilled which is equivalent to satisfying the VLE condition 4.10f, meaning equation 4.24a can be rewritten as 4.24b.

$$\sum_i^{NC} x_i = 1 \quad (4.24a)$$

$$\sum_i^{NC} \frac{K_i}{y_i} = 1 \quad (4.24b)$$

A dew point calculation is the reverse of the one of bubble point. The bubble point of fluid in liquid phase represents the point when the first bubble of vapor is formed, or in other words when the liquid begins to boil. The calculation uses a given liquid composition and can be made at given temperature when the pressure is adjusted until the liquid begins to boil, or the other way around, at given pressure when the temperature is adjusted [Skogestad, 2008]. In both cases, the unknown state (temperature or pressure) is adjusted until the conservation law for the vapor phase (eq. 4.25a) is fulfilled which is equivalent to satisfying the VLE condition 4.10f, meaning equation 4.25a can

be rewritten as 4.25b.

$$\sum_i^{NC} y_i = 1 \quad (4.25a)$$

$$\sum_i^{NC} K_i \cdot x_i = 1 \quad (4.25b)$$

The way these calculations are carried out is further discussed in section 6.1.2.

4.6.2 Model with complementarity constraints

A model formulation with complementarity constraints can account for the changes in topology while still using the same set of equations for all phase regions. The model equations for this formulation are almost the same as the model formulation for two-phase region presented in equation 4.10 presented in the previous section. There are a few key differences however. Firstly, the vapor-liquid equilibrium condition has to be relaxed such that this equation is also satisfied in the single phases (e.g. vapor and liquid), according to equation 4.26, where β is a relaxation parameter. This relaxation method originates in the work of Kamath et al. [2010] and was discussed in section 2.2.2.

$$y_i = \beta x_i \quad (4.26)$$

Secondly, slack variables are introduced with the help of complementarity constraints, according to equation 4.27. The equations is adapted from the formulation presented in equation 2.1. Each slack variable has designated a complementarity constraint, while the parameter β is determined based on equation 4.27c.

$$0 \leq n_V \perp s_V \geq 0 \quad (4.27a)$$

$$0 \leq n_L \perp s_L \geq 0 \quad (4.27b)$$

$$\beta - 1 - s_V + s_L = 0 \quad (4.27c)$$

The slack variables are orthogonal to the phase holdups, meaning that when the slack variable for the liquid phase $s_L > 0$, the liquid holdup $n_L = 0$, and when the slack variable for the vapor phase $s_V > 0$, the vapor holdup $n_V = 0$. These slack variables have the role of cancelling the flows for the non-existent phases.

Thirdly, one can make use of the possibility of implementing inequality constraints in this formulation. A good options would be the sign of the first and second derivatives of the cubic equation of state 3.2a. These give an indication on the type of roots that the equation has, and as a result, of the phase region that the system is in, as discussed in section 3.1.4. These inequality constraints are presented in equation 4.28 and are taken from the work of [Kamath et al., 2010].

$$f'(Z_V) \geq 0 \quad (4.28a)$$

$$f''(Z_V) \geq 0 \quad (4.28b)$$

$$f'(Z_L) \geq 0 \quad (4.28c)$$

$$f''(Z_L) \geq 0 \quad (4.28d)$$

It can be observed that this formulation is more complicated compared to a simple DAE formulation, namely because of the additional slack variables and relaxation parameter which normally are determined using an optimization solver. Methods for solving this model are given in section 5.2.

4.6.3 Alternative formulations

A few other modelling approaches have been tried in parallel to formulating the model with different set of equations in order to have a simple model without using logical proposition with a single set of equations for all phase regions while still remaining a reliable and accurate representation of the process. All the alternatives presented below forseek to circumvent the infeasibility of the vapor-liquid equilibrium condition in the single phase regions and the use only the model formulation from equations 4.10.

4.6.3.1 Canceling the compressibility algebraic equations for the non-existing phase

In this alternative, the liquid compressibility algebraic equation 4.10i was multiplied with the liquid fraction and the vapor compressibility equation 4.10h with the vapor fraction as is given in equation 4.29. The other equations are the same as in the two phase model formulations given in equations 4.10 in section 4.6.1.1. The idea behind it is to cancel the terms accounting for the non-existing phase and eliminated the necessity to solve for them. It can be observed that when the vapor fraction is equal to unity and the system is in vapor phase, equation 4.29a is equivalent to equation 4.10h while equation 4.29b is reduced to $0 = 0$. Since the algebraic equation is satisfied, it was expected that the solver would not search for a solution for the liquid compressibility Z_L . Similarly, when the vapor fraction is equal to zero and the system is in liquid phase, equation 4.29b is equivalent to equation 4.10i while equation 4.29a is reduced to $0 = 0$. Since the algebraic equation is satisfied, it was expected that the solver would not search for a solution for the vapor compressibility Z_V .

$$v_j \cdot [Z_{V_j}^3 + Z_{V_j}^2 + Z_{V_j} \cdot (A_{V_j} - B_{V_j} - B_{V_j}^2) - A_{V_j} \cdot B_{V_j}] = 0 \quad (4.29a)$$

$$(1 - v_j) \cdot [Z_{L_j}^3 + Z_{L_j}^2 + Z_{L_j} \cdot (A_{L_j} - B_{L_j} - B_{L_j}^2) - A_{L_j} \cdot B_{L_j}] = 0 \quad (4.29b)$$

However, this alternative failed to produce a feasible solution. A possible explanation is that, the algebraic equations 4.29 are non-smooth continuous functions and thus require a non-smooth solver to provide a feasible solution which is beyond the scope of this project.

4.6.3.2 Adding trace components to maintain the system in two phase

In this alternative, the approach is to introduce in the mixture a heavy component which is in liquid phase at the conditions that a process stream is in vapor region and a light component which is in vapor phase at the conditions that a process stream is in liquid region, both in small concentrations ([0.1 - 0.5 %]). The purpose is to "force" the system to be always in the multiphase region regardless of the operation conditions of the heat exchanger. All the equations are the same as in the two phase model formula-

tions given in equations 4.10 in section 4.6.1.1.

A possible light component is helium, chosen on the bases that it is in vapor phase far beyond the lowest operating temperatures of natural gas liquefaction process. In additions, helium is a natural byproduct of natural gas as result of natural radioactive decay of a source rock followed by its accumulation in the gas reservoir [Zartman et al., 1961] and it is reasonable to assume that helium is a trace component in a natural gas liquefaction process. A possible heavy component is an oil which in the real process is used as a lubricant for the screw compressor and it is reasonable to assume that it can be found as a trace component even if a lubricant separation unit is in place [Nekså et al., 2010]. This implementation was again unsuccessful and a good explanation is that while the the mixture may seem to be in the two phase region, in reality one of the phases contains only one component (trace component) while the other phase contains the other remaining components and thus the vapor-liquid equilibrium between all the components of the two phases does not hold and a feasible solution is therefor not found.

4.6.3.3 Giving fictions compositions to the non-existent phase

In this alternative, the approach was to give fictitious components concentrations in the non-existent phase and to set to zero the terms accounting for the holdup or flow of the non-existent phase. All the equations are the same as in the two phase model formulations given in equations 4.10 in section 4.6.1.1. The idea behind it was to "force" the system to always numerically be in the two phase region, even though is is not physical possible. Yet again, this alternative was not successful in providing a good numerical result for which several possible explanations can be found. Firstly, the phase location should be known before hand if one wants to cancel the terms describing the non-physical phase and to know when and where the fictitious compositions should be given. Secondly, while the VLE conditions is satisfied numerically, the liquid and vapor composition are dependent one of another and by having fictitious compositions for one phase, the composition of the other phase will not match a real state of the mixture which further results in inaccurate calculation of all thermodynamic properties of a stream since they are calculated based on compositions averages. This means that the fictitious compositions cannot be randomly given. At the same time it is difficult to find values for the compositions that do not have a physical meaning such that the model gives an accurate result. Another disadvantage of this method is that the phase

location should be known beforehand such that one is aware

All three alternatives with a single set of differential and algebraic equations which had the purpose to keep a simple model presented here seemed reasonable at the time they were initially proposed. Even though their implementation did not produce feasible results, these trials had a positive outcome in hindsight since the theory behind a phase change process was better understood by making an analysis and finding explanations for the failures.

Chapter 5

Simulation methods

The next step after formulating a dynamic model for a phase change heat exchanger is to test it through simulations. In section 4.6 two main modelling approaches for the phase change process were presented, one formulation consisting of differential and algebraic equations while the other formulations has in addition complementary constraints. Since the two formulations are different, one solving method has to be implemented for each model respectively.

The two different simulations methods have been studies in this thesis are:

1. *ode15s* - differential algebraic equations solver in Matlab[®]
2. solving a mathematical problem with complementarity constraints

5.1 The *ode15s* solver

It can be observed that set of equations formulated for the two phase region in equations 4.10, for the vapor region in equations 4.12 and for liquid region 4.13 match the general form of differential and algebraic equations 3.20. In order to simulated the model in an equation oriented environment (solve all equations simultaneously), a differential and algebraic equations solver must be used. The alternative is represented by

a two layer approach when firstly the algebraic equations are solved and the results are used in integrating the differential equations.

The general form of a differential and algebraic system of equations is written in its semi-explicit form in equations 3.20. In addition to the set of differential and algebraic equations, the initial condition of the problem, $y(t_0) = y_0$, must also be supplied. To solve this system of equations, Matlab[®] uses a numerical differential formula to obtain the solution iteratively starting at the initial condition y_0 over the desired time interval [Shampine and Reichelt]. The next solution is determined based on the results from the previous time step. The time step can be variable [MathWorks, b].

For this purpose of simulating the model formulated in section 4.6.1, the *ode15s* solver in Matlab[®] is chosen. The reasons behind choosing *ode15s* are are:

- it is already implemented in Matlab[®] which is an used friendly environment to which the author is familiar to
- it has a varying time step, meaning that it is applicable to stiff system as this model is expected to be according to section 4.5
- it can automatically detect events while simulating (see section 5.1.4)

The drawbacks of using *ode15s* are:

- to the best of the author notion, no literature is available for solving non-smooth models with this solver
- it is not that easy to know the algorithm behind it

5.1.1 Mass matrix

The *ode15s* solves differential and algebraic equations with a mass matrix M written in the form of equation 5.1. The mass matrix M can be either constant or time variant. The former case is valid in this work.

$$M_M(t, y) \cdot y' = f(t, y) \quad (5.1)$$

The mass matrix is in fact the term that indicates to the solver what are the differential and algebraic variables and equations. Therefore, it is important to understand its structure. M is a sparse square diagonal matrix which elements are 1 for the differential variables and 0 for the algebraic variables, as showed in equation 5.2. It is important to noticed that all algebraic variables are stacked after all the differential variables.

$$M_M \cdot y' = \begin{pmatrix} y'_1 & 0 & \cdots & 0 \\ 0 & y'_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix} \quad (5.2)$$

Considering the number of states from table 4.2, the size of the mass matrix for the model in section 4.6.1 becomes $(7 + 3 \cdot NC) \cdot M$ by $(7 + 3 \cdot NC) \cdot M$. The size of the mass matrix must be constant during integration and hereby the importance of having the same number of equations in both phases as mentioned in section 4.6.1 is demonstrated. Lets consider that the system is in single vapor phase and the simulation is started with the corresponding set of differential and algebraic equations 4.12. As heat is removed, the temperature in the cell drops, the dew point is reached and the system transitions in the two phase region where a different set of differential and algebraic equations is valid (eq. 4.10). If one wants to call the solver *ode15s* only once and continue simulating for the two phase region without stopping at the dew point where the switch in the set of equations happens, it is required to have the same number of equations in all phases. To have the same mass matrix regardless of the phase region, dummy variables and equations have been introduced for both single phases, as explained in section 4.6.1.2 (see equations 4.15a and 4.14a).

5.1.2 Non-smooth continuous ODE

By having a different set of equations for each phase and a switch condition to select the correct set of equation, the model from section 4.6.1 becomes non-smooth. Thus, one must first verify if the solver *ode15s* is applicable to such systems. To test the behaviour of the chosen solver, a simple example of a non-smooth continuous function is created and solved using *ode15s*. The function selected is a differential equation, given in equation 5.3. Until $y \leq 1$ the function take the value y while when $y > 1$ the function

takes the value $-y + 2$. An initial condition is also provided since it is required by the solver. The function is plotted in figure 5.1 in order to have a clear picture of its shape and where the non-smooth point is. It can be observed from figure 5.1 that the function is continuous and the non-smooth point is at $y = 1$.

$$y' = \begin{cases} y & \text{if } y \leq 1 \\ -y + 2 & \text{if } y > 1 \end{cases} \quad (5.3)$$

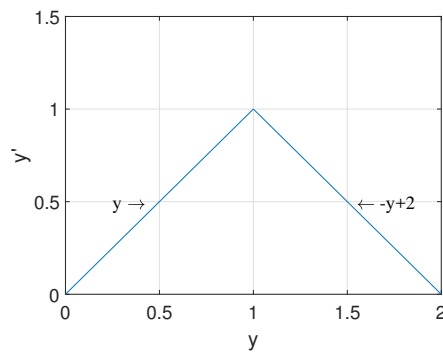


Figure 5.1: Nonsmooth continuous function from equation 5.3

Considering that *ode15s* uses an explicit method to numerically integrate starting at the initial conditions, and continues iteratively to the next solution, it is expected that solving for equation 5.3 will not produce any internal error in the solver or wrong results. Figure 5.2 illustrates the result of solving for equation 5.3 with a time span $t = (0, 10)$. Thus it is proven that *ode15s* can be used for non-smooth continuous differential system of equations, and it is valid solver to use for the non-smooth model formulation from section 4.6.1.

5.1.3 Investigating how the algebraic equations are solved

Beside the ability to solve non-smooth differential equations it is also important to understand how the algebraic equations of DAE system are solved with the *ode15s* solver. For this purpose, an example with two differential equations and one algebraic equation is created.

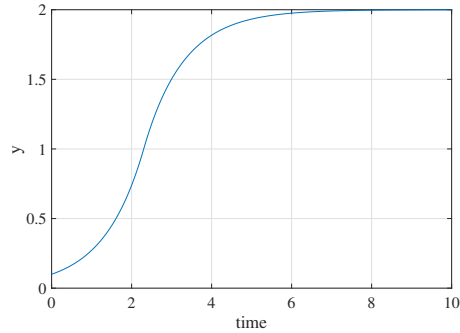


Figure 5.2: Integration of function from equation 5.3

The differential equations of the example considered are given in equation 5.4.

$$\frac{dx}{dt} = 1.5 - x - s \cdot z_1 - (1 - s) \cdot z_2 \quad (5.4a)$$

$$\frac{dy}{dt} = \frac{1}{y} \quad (5.4b)$$

The algebraic equation of the model is given in equation 5.5.

$$z^3 + y \cdot z^2 + z - 1.2 = 0 \quad (5.5)$$

Where x and y are the differential variables. z_1 and z_2 are algebraic variables and are two of the three roots of the algebraic equations 5.5. z_1 is always taken as a real root, while z_2 can either be one of the two conjugated roots, when they exist, or one of the other real roots otherwise. How the solver attributes the roots of the algebraic equation to z_1 and z_2 is discussed in sections 5.1.3.1 and 5.1.3.2, where the solving methods are described. s is a binary variable which takes values according to equation 5.6.

$$s = \begin{cases} 1, & \text{if all roots of equation 5.5 are real} \\ 0, & \text{if only one of the roots of equation 5.5 is real} \end{cases} \quad (5.6)$$

It is chosen to include the decision variable s to mimic what happens if one of the phases disappears.

Equation 5.5 is chosen such that its roots change from being one real and two complex to three real roots when the value of the differential variable y becomes equal to 2.694. The change in the number of real roots is shown in figure 5.3 for different values of y . In figure 5.3a, the algebraic equation 5.5 has a single real roots and two complex conjugate. In figure 5.3b, when $y = 2.694$, the equation starts to have three real roots with two of the being very close and thus almost indistinguishable. In figure 5.3c the three real roots become evident as y increases.

The first reason behind formulating the algebraic equation in this way is to mimic the behavior of the algebraic equation for compressibility 3.2a, equation which has one real root in either of the single region and three real roots in the two-phase region.

The second reason for choosing this equations is to understand how *ode15s* treats an algebraic equation that changes the type of roots from all real to one real and two complex.

Two methods are used to solve the differential and algebraic system formed by equations 5.4 and 5.5:

1. one layer approach when *ode15s* is used to solve both the differential and algebraic equations
2. two layer approach when the *root* function is used to solve for the roots of the algebraic equation while *ode15s* is used only for the differential equations

5.1.3.1 One layer approach

In order to obtain two roots for the algebraic equation 5.5 with the *ode15s*, the algebraic equations 5.5 has to be supplied two times to the solver, one for each roots that it desired to determine. Since the number of equations must be equal to the number of states, the new system has two differential states and two algebraic states: $w = [x \ y \ z_1 \ z_2]'$. By adding one algebraic equation for each algebraic state, the system becomes 5.7:

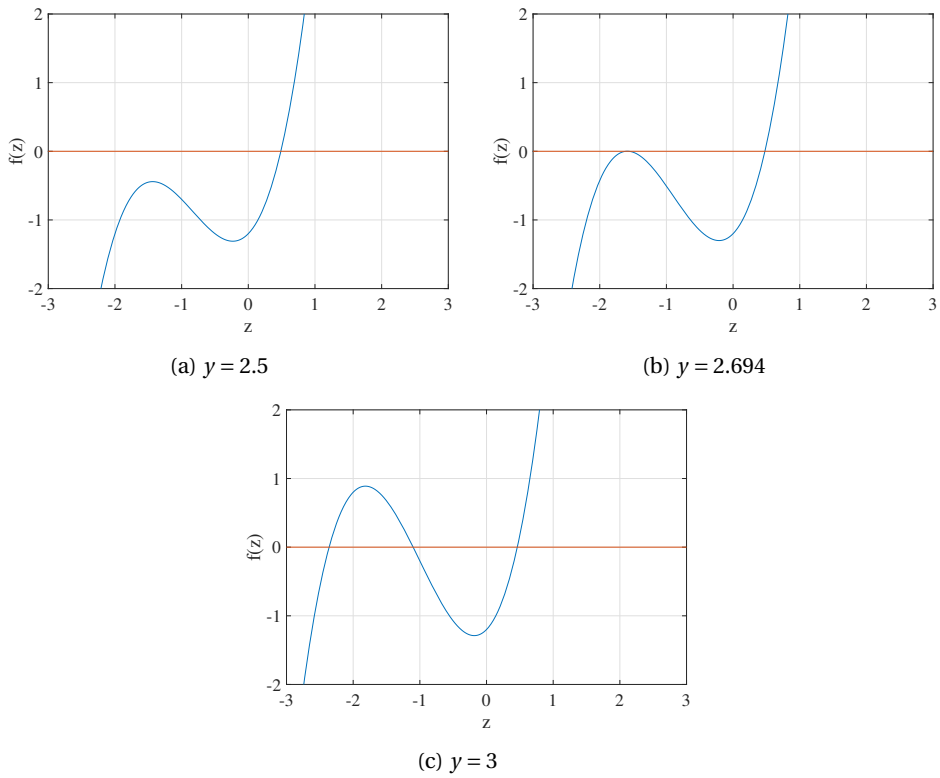


Figure 5.3: Change in the number of real roots for equation 5.5 with y

$$\frac{dx}{dt} = 1.5 - x - s \cdot z_1 - (1 - s) \cdot z_2 \quad (5.7a)$$

$$\frac{dy}{dt} = \frac{1}{y} \quad (5.7b)$$

$$z_1^3 + y \cdot z_1^2 + z_1 - 1.2 = 0 \quad (5.7c)$$

$$z_2^3 + y \cdot z_2^2 + z_2 - 1.2 = 0 \quad (5.7d)$$

To attribute the values to the binary variable s , algorithm 4 is implemented.

Algorithm 4: Algorithm for attributing the variable s as a function of the number of real roots

```

if  $z_1$  &  $z_2$  are real then
  |  $s = 1$ ;
else
  |  $s = 0$ ;
end

```

Two different set of initial conditions were tested. The first set has all real numbers for all states, while the second set has a complex values for the state z_1 . The reason for giving different states is to investigate how *ode15s* treats them in solving the system of differential and algebraic equations. The values for z_{10} and z_{20} are given by the roots of equation $z^3 + y \cdot z^2 + z - 1.2 = 0$, which are: $[0.5583; -1.0292 + 1.044i; -1.0292 - 1.044i]$. The initial conditions are given below:

Table 5.1: Initial conditions

Simulation	Initial conditions	x_0	y_0	z_{10}	z_{20}
Case 1		1	1.5	0.5583	0.5583
Case 2		1	1.5	0.5583	-1.0292+1.044 i

The results of these simulations for the one layer approach are presented together with the results of the two layer approach in section 5.1.3.2 in order to be able to make a better comparison and analysis.

5.1.3.2 Two layer approach

In the two layer approach the *roots* function in Matlab[®] is applied to determine all roots of the algebraic equation 5.5. If all roots are real, the smaller roots is attributed to the algebraic variable z_2 , while the largest root is attributed to the algebraic variable z_1 , in a similar was that roots of the compresibility equation 3.2a could be attributed to each phase in a two phase system (see section 3.1.4). The variable s takes values in the same manner as in the previous sections (see equation 5.6). After determining the roots, the differential equations are numerically integrated by the *ode15s* solver. The *roots* function represents the inner layer of the algorithm while the *ode15s* represents the outer layer. The algorithm used for this approach is given in algorithm 5. It should be noted that the last value of the output of *roots* function is always real, for the case of a cubic polynomial.

Algorithm 5: Two later approach algorithm

Determine the $r = \text{roots}$ of equation 5.5;

if all roots are real **then**

$z_1 = \max(r)$;

$z_2 = \min(r)$;

$s = 0$;

else

$z_1 = r(3)$ % real value ;

$z_2 = r(1)$ % complex value;

$s = 1$;

end

Solve differential equations 5.4 with *ode15s*;

The same initial conditions were given for the differential variables x_0 and y_0 as for the one layer approach (see table 5.1. No initial conditions are necessary for z_1 and z_2 . The results of these simulations for the one layer approach are presented together with the results of the two layer approach in section 5.1.3.2, in order to be able to make a better comparison and analysis.

5.1.3.3 Results of one layer and two layer solving methods

The results from the one layer approach and two layer approach for solving the differential and algebraic system formed with equations 5.4 and 5.5 are illustrated in figure 5.4. Both systems were integrated over a time interval of 10 (time units). For the one layer approach, only the case with the initial conditions all real (see table 5.1) was solved without any errors, and it is this case showed in figure 5.4a. On the other hand, the case where the algebraic state z_2 was initialized as a complex values gave an error while integrating: *Unable to meet integration tolerances without reducing the step size below the smallest value allowed* at time $t = 2.5$. However, over the time period $t = (0, 2)$ when *ode15s* was able to solve, the values for all states (including differential) are complex. Since all variable are complex, plotting them is not very representative and the numerical results are available in appendix D.

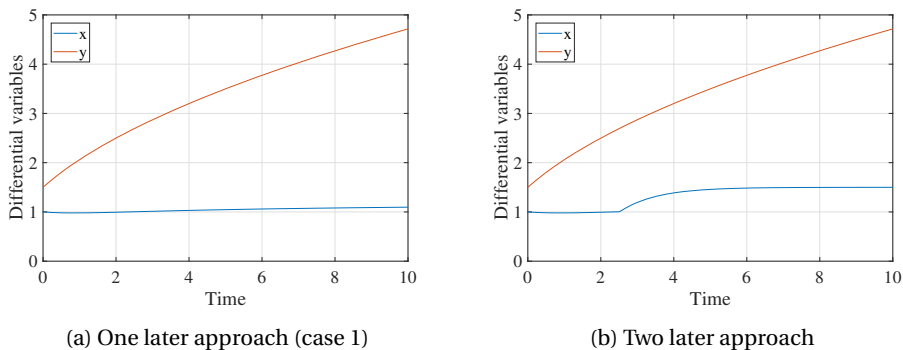


Figure 5.4: Solving methods for equations 5.4 and 5.5

The difference between the two methods are clear by comparing figure 5.4a and figure 5.4b. As the value for the differential variable y become greater than 2.694, the roots of the algebraic equations 5.5 transition from the complex plane to the real plane and the binary variable s changes from 1 to 0, it is expected to see a non-smoothness in the differential variable x , according to equation 5.4a. This behaviour is captured by the two later method and not by the single layer.

Thus one can conclude that solving for the roots of the algebraic equations in an inner layer with the *roots* function followed by solving the differential equations with *ode15s* is a more robust method as compared to using *ode15s* to solve both the differential and algebraic equations, in the case when the roots of the algebraic equations

change from complex to real. The drawback of this method, however, is that the *roots* functions is applicable only to polynomial functions.

It is now understood that *ode15s* determines only one of the roots for an algebraic equation even if the former has several roots. Further, the root that is determined is the one closest to the initial values given by the user, meaning that the solver is blind to a change in the roots from all real to some real and some complex, or vice versa, from complex to real. This means that when solving for the vapor and liquid compressibilities from 3.2a, the solver is not able to adjust the type of roots as they are sensitive to the initial conditions.

This proves once more that a model with a single set of equations for multicomponent phase change process cannot be used for simulation with *ode15s*, and it is advisable to split the model for each phase region, as described in section 4.6.1.

Moreover, this can be yet another numerical explanation on why the alternatives for circumventing the vapor-liquid equilibrium conditions presented in section 4.6.3.1, 4.6.3.2 and 4.6.3.3 were unsuccessful.

All these considerations demonstrate that the model has to be carefully constructed and carefully initialized for a successful simulation with *ode15s*.

This analysis should be perceived as only for the *ode15s* solver and should not be generalized for other DAE solvers which are not a product of Matlab[®].

5.1.4 Event function

An advantage of the *ODE* solvers in Matlab[®] is that it incorporates an function which detects when and event specified by the user happens during simulation. Indeed this is beneficial if one wants to trace the phase change in this heat exchanger model. The deficiency of *ode15s* event function is that it includes a detection along a single dimension, in this case, time. The implications for this heat exchanger model are given in section 6.1.6. In order to use the event function, one must first learn how to construct it. The event function detects an event when a mathematical expression supplied by the user is equal to zero. At that point, the integration can either be continued or stopped. The general form of the event function consists of three arguments [MathWorks, a]:

- value, a mathematical expression that is equal to 0 when an event happens
- isterminal = 0, if the the integration should continue when the event is triggered, and 0 otherwise
- direction = 0, if all zeros have to be detected; -1 if only the zeros where the function is decreasing have to be detected; +1 if only the zeros where the function is increasing have to be detected

The exact structure of the event function implemented for this heat exchanger model is presented along with the results in section 6.1.6 for a better overview.

5.2 MPCC solving methods

This section describes two methods available in the literature for solving a mathematical problem with complementarity constraints as well as an approach for how a dynamic model could fit to this formulation. This methods have only been briefly studied due to time constrictions of the thesis. A detailed overview of these methods is presented in Herty and Steffensen [2012].

The drawback of formulating a MINLP into a MPCC by changing the binary variables into continuous variables with the use of complementarity constraints (see section 3.2.2) is that the linear independent constraint qualification (LICQ - holds if the set of active constraint gradients is linearly independent [Nocedal and Wright, 2006]) and the Mangasarian–Fromovitz constraint qualification (MFCQ) are not satisfied at all the feasible solutions of an MPCC. This implies that this problem requires a special numerical solution [Herty and Steffensen, 2012].

The two approaches available for solving an MINLP reformulated as an MPCC are [Herty and Steffensen, 2012]:

1. a relaxation approach to MINLP
2. a penalty approach to MINLP

Both methods involve solving a non-linear-problem, and therefore, a non-linear solver is required. Moreover, as they are formulated as an optimization problem, the

objective function $f(x)$ has to be given as a constant (e.g. $f(x) = 0$) such that the process is simulated and not optimized.

The relaxation approach is given in equation 5.8. This method implies relaxing the complementarity constraints with the use of a relaxation parameter μ .

$$\begin{aligned}
 \min f(x) \text{ s.t. } & g(x) = 0 \\
 & g(x) \geq 0 \\
 & 1 \geq x_1 \geq 0 \\
 & \Phi(x_1, 1 - x_1) \leq \mu
 \end{aligned} \tag{5.8}$$

The penalty approach is given in equation 5.9. This method implies the use of one of the penalty function given in equation 5.10. These are a measure of the constraints violations done by the solver and are used redirect the solver towards to feasible solution.

$$\begin{aligned}
 \min f(x) + \rho \Psi(x_1, 1 - x_1) \\
 \text{s.t. } & g(x) = 0 \\
 & g(x) \geq 0 \\
 & 1 \geq x_1 \geq 0
 \end{aligned} \tag{5.9}$$

Where the penalty function can either be either of the equations below.

$$\begin{aligned}
 \Psi(a, b) &= \sum_{j=1}^p \Phi_j(a, b) \\
 \Psi(a, b) &= \sum_{j=1}^p a_j \cdot b_j
 \end{aligned} \tag{5.10}$$

5.2.1 Discretizing the differential equations

An important observation is that the MPCC formulation is written in terms equality, inequality and complementarity constraints, meaning that this formulation only can fit a process model written in terms of algebraic equations. Therefor if the dynamics of process is desired to be modelled, the differential equation resulted from the balance

equations have to be fully discretized and converted into algebraic equations. A possibility of achieving this is by applying a collocation formulation to the differential equations [Biegler, 2010]. A collocation formulation implies approximating the differential states y with polynomials of order $K + 1$ over a single element as in figure 5.5.

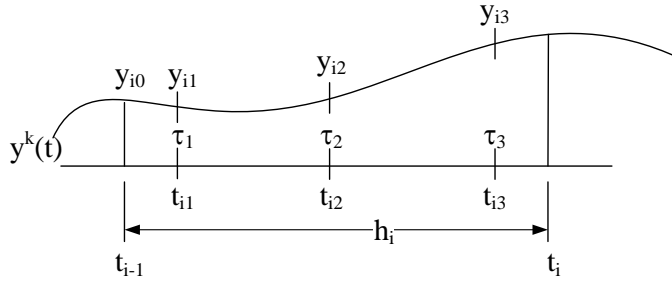


Figure 5.5: Polynomial approximation for state profile across a finite element

There are several equivalent ways of expressing the differential state as a polynomial. One of them is the power series representation given in equations 5.11.

$$y^K(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \dots + \alpha_K t^K \quad (5.11)$$

In order to determine the coefficient of the polynomial, the power series approximation of the differential state is introduced in the differential equation resulting in equations 5.12. This is further enforced as an algebraic equation.

$$\frac{dy^K}{dt}(t_{ik}) = f(y^K(t_{ik}), t_{ik}), \quad k = 1 \dots K \quad (5.12)$$

The advantage of using this method is that it allows solving simultaneously all equations with an NLP solver. An efficient NLP solver is needed for this solving large problems. A possibility is to use Newton-based barrier methods.

The implementation of this method requires a time step with variable size. The step size should be adjusted internally by the algorithm such that the topology of the model is changed exactly at the end of a time step [Biegler, 2010].

Chapter 6

Simulation results

This chapter presents only the results of the simulations carried out with *ode15s*. Due to insufficient time, the implementation and simulation of the model with complementarity constraints have not been carried out.

6.1 Results of simulations with *ode15s*

This section presents the results of simulating the model formulated as a different set of differential and algebraic equations for each region with a switch algorithm to select the valid set of equations. First, the model from section 4.6.1 is tested for single vapor then for two phase region followed by a transition from vapor to two phase region. The main focus of this work is to try to simulate a transition from vapor to multiphase region. Therefore simulations for the single liquid region have not been performed. However, they are not conceptually different as the simulations presented in this section.

In additions, different values for parameters such as the number of cells M or the variation of the heat removed Q_{HEX} with time are tested to observe the model behaviour to such changes.

For all simulations, the stream composition is the one from table 2.2, while the inlet feed flow and pressure are from table 2.1. For all simulations the reference temperature

T_{ref} required for the enthalpy calculation is set to 298 K. Physical constants such as critical temperature and pressure, acentric factor and coefficients for specific heat for each component required by the thermodynamic model (see section 3.1) are given in appendix B.

6.1.1 Initial conditions

The *ode15s* solve problem with an initial value, meaning that the initial conditions for simulating this model have to be supplied. The initial conditions have to be consistent of very close to, as discussed in section 5.1.

For these simulations a two step procedure was followed in order to obtain consistent initial conditions. At first, a simulation was carried out using as initial condition results from the specialization project. Then these results, taken at the last time step, when the system stabilizes and the conditions are not changing, were used again as initial conditions for a simulation with no heat removed (Q_{HEX}). These final results, again taken at the last time step, were used as initial conditions for the results presented in the next sections.

The idea behind this procedure is to obtain a set of initial conditions with the system is at steady-state, where the temperature, phase compositions and vapor fraction in each cell match the conditions of the inlet stream since no heat is removed. The initial conditions for the pressure in each cell on the other hand are given considering a linear and constant pressure drop from the inlet to the outlet, the total pressure drop being 5 mbar. A positive pressure drop is necessary to maintain a flow between the cells, since the outlet flow is given as function of the pressure drop (equation 4.16a). Another reason for using this method is to obtain consistent initial conditions for the compressibility equations, which was shown to be important in section 5.1.3.

This method of supplying initial conditions is equivalent to a transient analysis of the system. The system at one steady-state, and by removing heat, another steady-state is reached.

6.1.2 Dew and bubble temperatures calculations

Algorithm 2 compares the temperature in each cell with the dew or the bubble point respectively to switch between the set of equations for each phase. Therefore it is required to determine these temperatures. The theory behind calculating the dew and bubble point is presented in section 4.6.1.7. The results of these calculations are presented below. Considering that for these simulations the composition in each cell should remain constant due to the no slip assumptions, and a linear pressure drop of 5 mbar is considered, the dew and bubble points can be calculated before running the simulations.

The *fmincon* optimization function in Matlab is used to determine the dew and bubble points for a mixture with the composition from table 2.2, with given pressures between 1.8239 MPa (inlet pressure of the mixed refrigerant stream) and 1.8234 MPa (thus a pressure drop of 5 mbar), with a varying step of 1 mbar = 10^{-4} MPa. The purpose of using an optimization solver is to pass the vapor-liquid equilibrium condition 4.10f as an equality constraint to the solver and thus determining both the temperature and the concentrations that satisfy equation 4.24a. The initial conditions supplied to the solver have to be carefully chosen such that they are close to a feasible solution and thus they have been taken from the commercial process simulator Aspen Hysys[®]. Due to the fact that only the solution of the equality constraint is desired, the objective function is passed as a constant number. The calculations are adapted from the bubble point calculation script available online [Skogestad et al., 2013]. The results are presented in table 6.1.

Table 6.1: Dew and bubble points for a pressure variation of 5 mbar

Pressure [MPa]	Dew temperature [K]	Bubble Temperature [K]
1.8234	262.0048	165.2340
1.8235	262.0206	165.2362
1.8236	262.0363	165.2383
1.8237	262.0520	165.2405
1.8238	262.0678	165.2427
1.8239	262.0835	165.2448

It can be observed from table 6.1 that the variation in the dew temperature is $\Delta T_{dew} = 0.0787$ K while the variation in the bubble point is equal to $\Delta T_{bub} = 0.0108$ K. As expected, these values are small considering the small pressure drop. Thus, the dew and bubble points are assumed to be constant in time and space for all simulations. There-

fore the arithmetic mean is used in simulations. The values are given in equation 6.1.

$$T_{dew} = 262.05 \text{ K} = -11.1 \text{ }^\circ\text{C} \quad (6.1a)$$

$$T_{bub} = 165.24 \text{ K} = -107.9 \text{ }^\circ\text{C} \quad (6.1b)$$

The advantage of considering the dew and bubble temperatures constant is that it is not required to solve in an inner layer an optimization problem with *fmincon* to determine these points for each cell at each time step, meaning that the one layer approach (i.e. equation oriented environment) with all equations solved at the same time still applies.

6.1.3 Vapor phase simulation

This section presents the results for the heat exchanger operated in single vapor region. The model valid for this region is represented by equation 4.12, together with the dummy equations 4.14a, which account for the variables corresponding to the liquid phase, the non-existent phase in this case. The values for the dummy liquid composition and dummy liquid compressibility are initialized as equal to the dew point conditions, while the dummy liquid volume is given as zero $V_{L_0} = 0$. The reason why dummy equations and variables are needed is discussed in section 5.1.1.

The simulation and design parameters needed are given in table 6.2. The inlet temperature T_{inlet} is taken from the reliquefaction cycle presented in section 2.1 [Nekså et al., 2010]. The values for the heat exchanger volume, heat rate and pressure drop are not available for literature. Therefore, reasonable values are considered for these parameters.

In order to observe the behaviour of the model, the simulations are performed with different number of cells (i.e. control volumes) and constant or varying heat. These simulations along with their interpretation are presented in the next sections.

Table 6.2: Simulation and design parameters for vapor phase

Parameter	Value	Units
T_{inlet}	340.15	K
Q_{HEX}	10	MJ/min
V_{HEX}	100	m ³
ΔP_{max}	5	mbar
time	200	min

6.1.3.1 Varying the number of cells

Two values for the number of control volumes are used to test the trade off between accuracy and computational speed: $M = 10$ cells and $M = 60$ cells. The heat removed is constant in time (i.e., variables t_1 and t_2 from algorithm 1 are both set to 0). Both switching algorithm presented in section 4.6.1.6 give the same results for this case.

The dynamic temperature profile is illustrated in figure 6.1. The inlet and outlet temperature of these simulations are well within the single phase vapor region of the phase envelope illustrated in figure 4.1.

As expected, the temperature drops in each cells with time from the inlet temperature as heat is removed. It should be remembered that the initial conditions for all same are set to be equal to the inlet conditions, and that is why all cell have the same temperature at $t = 0$. Although the temperature profile for the case with 60 cells is more accurate in representing the conditions inside the heat exchanger since the cells are smaller in this case, the final temperature is the same for both numbers of control volumes used and it equal to $T_{outlet} = 269.12$ K. However, the simulation time is approximately 15 time larger for the case with 60 lumps. Therefore if one is interested only in the outlet temperature, 10 cells suffice for this case.

Another information that can be extracted from these simulations is the temperature profile once the system has stabilized and the cells temperature is not changing, which is equivalent to a steady-state temperature profile. This profile is shown in figure 6.2 for both 10 cells (figure 6.2a) and 60 cells (figure 6.2b). Due to the increasing the number of cells and thus decreasing the cell volume as well as the heat removed per cell, it is expected to have more refined profile for the case with 60 cells. It can also be observed that as the number of control volumes increases, the steady-state temperature

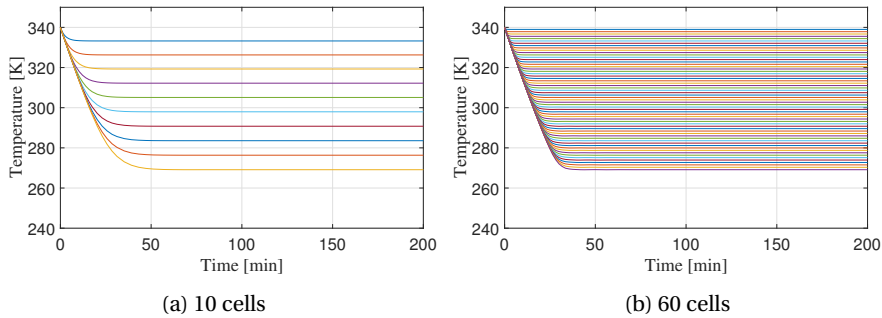


Figure 6.1: Temperature profile for vapor phase simulation

profile approaches to a linear profile, which is expected when there is no condensation (phase change).

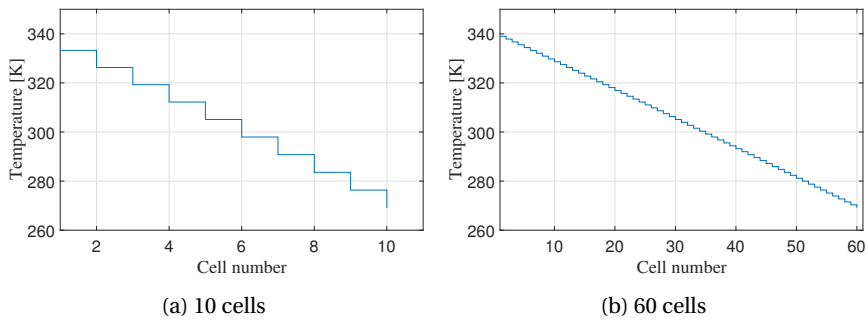


Figure 6.2: Steady-state temperature profile for vapor phase simulation

For the single vapor phase simulation the liquid volume is passed a dummy variable (see section 4.6.1.2 and equations 4.14a), and thus is it expected that the liquid volume remains at zero. This can in fact be seen in the steady-state profile for the liquid volume in figure 6.3 for both 10 and 60 cells simulations.

An increase in the total holdup is observed in figure 6.4 for each cell with time. The variation of the total holdup in time in each cell is dependent on the difference between the inlet and the outlet of a cells, which is turn are an explicit function of pressure (see equations 4.12a and 4.16a) and therefore, the allure of the plot is not straightforward to explain. However, as as in this case only vapor exists in the cells, a good explanation is that when the temperature decreases, the molar volume also decreases and the molar density increases as a result. As the volume of each cell is the same, it is expected to

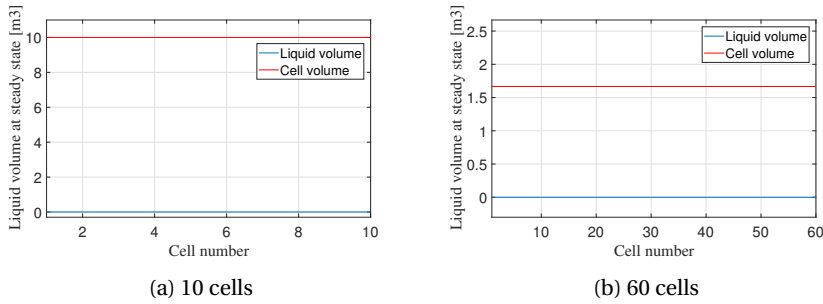


Figure 6.3: Steady-state liquid volume profile for vapor phase simulation

have an increase of holdup in time and along the heat exchanger.

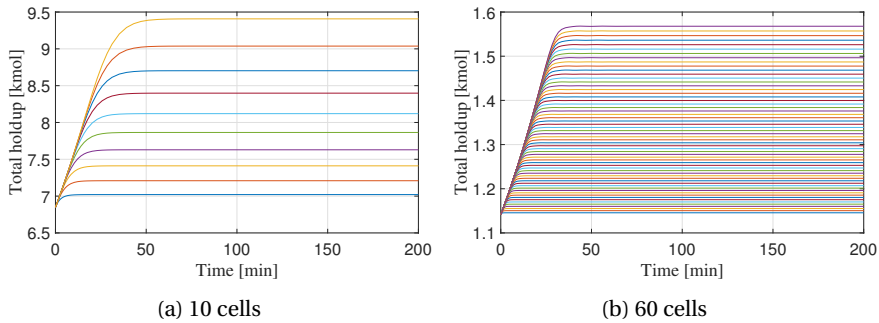


Figure 6.4: Dynamic total holdup profile for vapor phase simulation

The steady-state profile for the total holdup (again taken at the last time step of the simulation when the conditions are not changing) is illustrated in figure 6.5. As the volume of a cell is smaller in the case of 10 cells, a smaller holdup is observed in the case of 60 cells.

The variation of pressure in time in each cell is shown in figure 6.6, both for a number of 10 and 60 lumps. This pressure in each cell decreases at the first time step from the initial condition supplied to a smaller value when heat start being removed from the system. A plausible explanation is that removing heat acts as a disturbance from a pressure point of view and the system tries afterwards to reach a new steady-state. The reason why the pressure drops so fast in the first time steps is that momentum (or pressure) propagation happens instantaneously. Secondly, it is possible that the initial conditions supplied are not consistent enough and the solver *ode15s* searches for consistent values, thus slightly altering the values for the pressure.

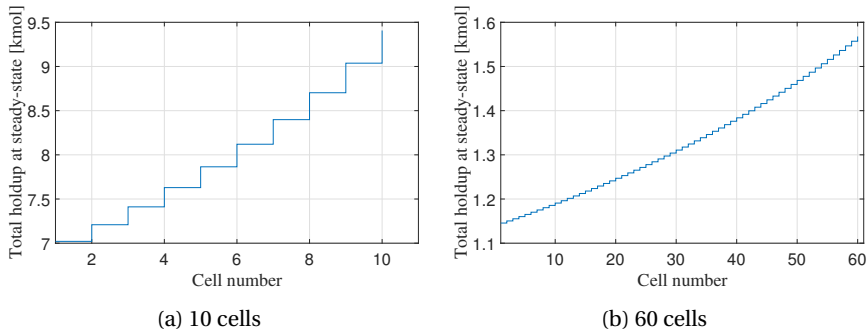


Figure 6.5: Steady-state total holdup profile for vapor phase simulation

A physical explanation for the increase in pressure in each cell in time at the first time steps is that as the total holdup in each cell increases in time, thus the higher the pressure that it exerts.

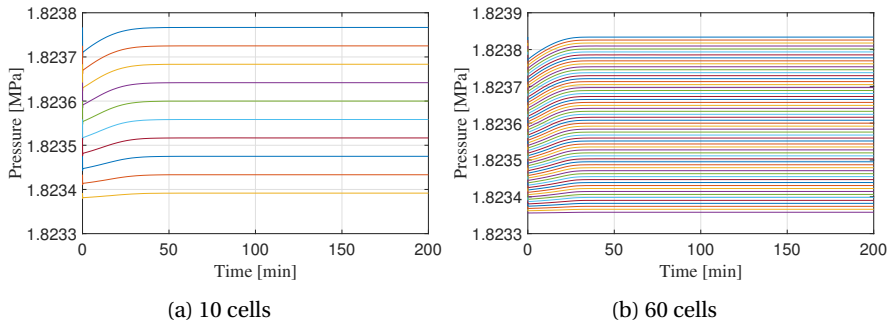


Figure 6.6: Dynamic pressure profile for vapor phase simulation

The steady-state pressure profile can be extracted by taking the values for pressure at the last time step, when the system is stable (i.e. not varying). The results are presented in figure 6.7. It can be observed that initially the total pressure drop assumed stands.

The results are quite similar for both cases of control volumes simulations, as it can be seen from figures 6.1a to 6.7. However, the simulations for 60 cells take around 15 times longer compared with the simulation for 10 cells. This means that there is not an actual simulation benefit from using more than 10 cells for the single vapor phase.

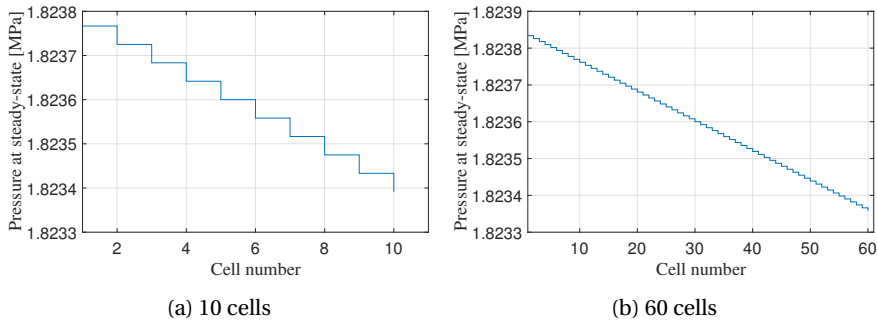


Figure 6.7: Steady-state pressure profile for vapor phase simulation

6.1.3.2 Varying heat removed with time

In section 4.6.1.4 algorithm 1 for the variation of heat removed with time from the heat exchanger is presented. Therefore, simulations with varying heat in time are presented in this section. As the number of cell does not influence significantly the results for the single vapor phase case, only a number of 10 control volumes are used. The simulation design and parameters are the same as in table 6.2. In connection to algorithm 1 the time constant until no heat is removed is $t_1 = 20$ min, while the time constant when the heat increases linearly until its maximum values is $t_2 = 100$ min.

The dynamic temperature profile is shown in figure 6.8. The temperature remain constant until time is equal to $t_1 = 20$ min when the temperature start decreasing. However, at the final time of the simulation the temperature in each cell has the same values as in the previous section, since the amount of total heat removed is the same.

The dynamic total holdup profile is shown in figure 6.9. As the temperature does not drop until $t_1 = 20$ the holdup also does not change until this point in time. The results at the final time step are the same as in figure 6.5, as the total heat removed is the same.

Compared to the temperature and holdup profiles, a difference appears in the dynamic pressure profile at the initial time step, as seen in figure 6.10. The pressure are not equal in all cells due to the pressure drop that has to be considered in order to maintain a flow between cells. A variation in pressure is observed in period between time $t_1 = 20$ min when the heat start being removed at time $t_2 = 100$ min when the heat reached its maximum value, after which the pressure reached a steady-state.

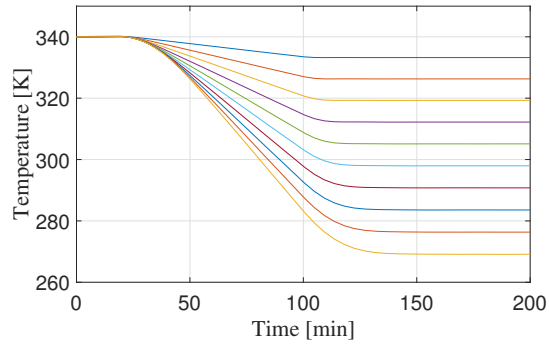


Figure 6.8: Temperature profile when heat is varying heat with time

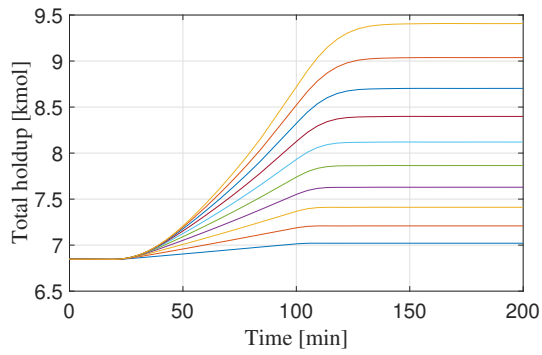


Figure 6.9: Total holdup profile when heat is varying heat with time

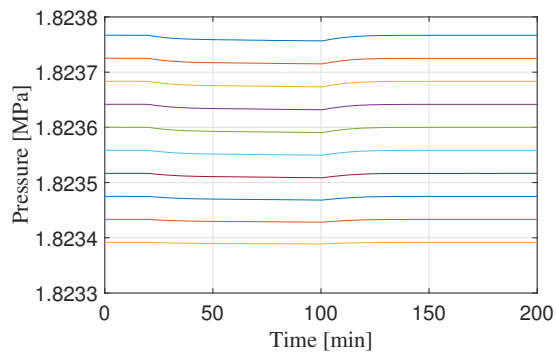


Figure 6.10: Pressure profile when heat is varying heat with time

6.1.4 Two phase simulation

The next step after the single vapor phase simulation is to simulate the case when the heat exchanger operates in two phase region, meaning that the inlet of the heat exchanger is in vapor-liquid phase and the outlet is expected to be also in vapor-liquid phase. The model valid for this phase is the one formulated in equation 4.10. The simulation and design parameters are presented in table 6.3. Both switching algorithm 2 and 3 give the same results. The dew and bubble point are considered constant and equal to the values given in equation 6.1.

Table 6.3: Simulation and design parameters for two phase

Parameter	Value	Units
T_{inlet}	262.05	K
Q_{HEX}	20	MJ/min
V_{HEX}	100	m ³
ΔP_{max}	5	mbar
time	200	min

6.1.4.1 Varying the number of cells

The same values for the number of cells are used in the two phase simulations as in the vapor phase simulation, i.e. 10 and 60 control volumes to test the trade off between accuracy and computational speed. The heat removed is constant in time (i.e., variables t_1 and t_2 from algorithm 1 are both set to 0). Both switching algorithm presented in section 4.6.1.6 give the same results for this case.

The dynamic temperature profile is illustrated in figure 6.11. The inlet and outlet temperature of these simulations are well within the two phase region of the phase envelope illustrated in figure 4.1. As expected, the temperature drops in each cell with time from the inlet temperature as heat is removed. The final temperature is the same in both case and it equal to $T_{outlet} = 218.6$ K. Even though the temperature profile for the case of 60 cells has more accuracy in representing the temperature inside the heat exchanger, the computational time is around 20 time larger compared to the case of 10 cells. Therefore, as for the case of single vapor phase simulation, the trade off between accuracy and computational speed is far superior for the 10 cells case for a robust simulation.

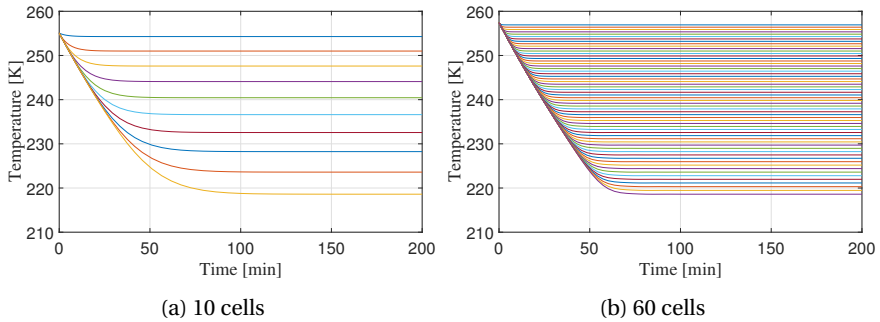


Figure 6.11: Temperature profile for two phase simulation

The steady-state temperature profile, considered when the temperatures are not longer changing, is shown in figure 6.12 for both 10 cells (figure 6.12a) and 60 cells (figure 6.12b).

Another information that can be extracted from these simulations is the temperature profile once the system has stabilized and the cells temperature are not changing, which is equivalent to a steady-state temperature profile. Due to increasing the number of cells, and thus decreasing the cell volume as well as the heat removed per cell, it is expected to have more refined profile for the case with 60 cells.

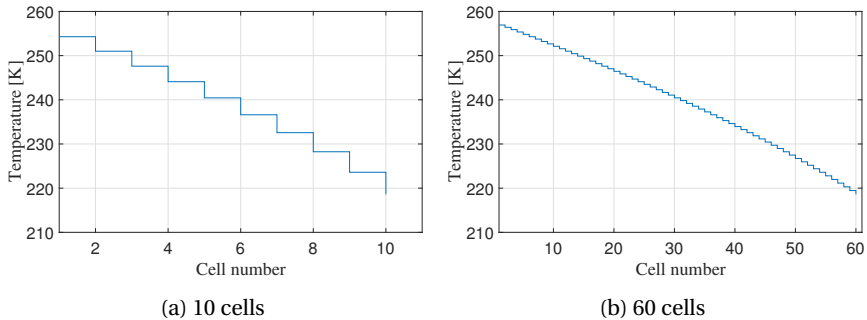


Figure 6.12: Steady-state temperature profile for two phase simulation

The dynamic profile of the liquid volume is illustrated in figure 6.13. The increases of liquid volume along the heat exchanger is in good accordance with the temperature profile. As the temperature in the cells decreases, more liquid condenses and the liquid volume increases in time in each cell. It can also be observed that the starting point of the liquid volume is very close to 0 in both case since the simulation is ran from the dew

point conditions of stream, which imply very small liquid volume.

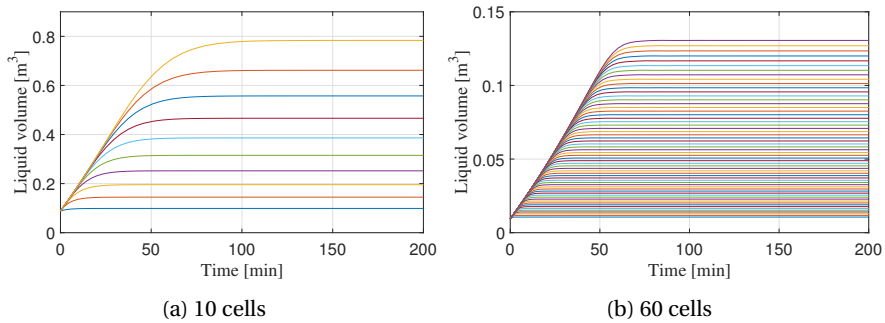


Figure 6.13: Dynamic liquid volume profile for two phase simulation

The liquid volume in each cell for the simulation of 10 control volumes (figure 6.13a) is approximately 6 times higher compared to the simulation of 60 control volumes (figure 6.13b), as one would expect considering that the cell volume in the former case is 6 times higher (see figure 6.14 for illustration of the cell volume for both cases). Figure 6.14 demonstrates yet again that 10 cells are enough for a robust simulation of the variation of liquid volume along the heat exchanger.

The steady-state profile for the liquid volume is obtained from the values at the last time step, when the conditions of the process are not varying anymore. This profile is illustrated in figure 6.14 for 10 and 60 control volumes respectively. The cell volume for both simulations is also shown on figure 6.14 with a red line. The cell volume for 10 lumps is $V_{cell} = 10m^3$, while the cell volume for the 60 lumps is $V_{cell} = 1.667m^3$. The cell volume is plotted to illustrated that none of the cell are overfilled during simulation.

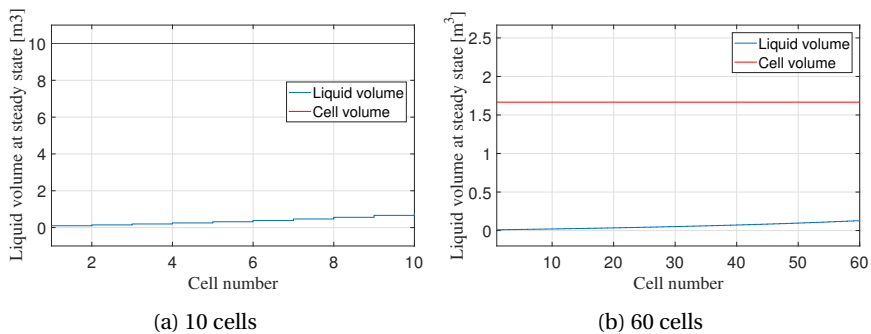


Figure 6.14: Steady-state liquid volume profile for two phase simulation

The dynamic total holdup profile is shown in figure 6.15 for 10 and 60 control volumes respectively. An increase in the total holdup for each cell with time which is in good correlation with a decrease of temperature which results in an decrease of the molar volume and increase of density.

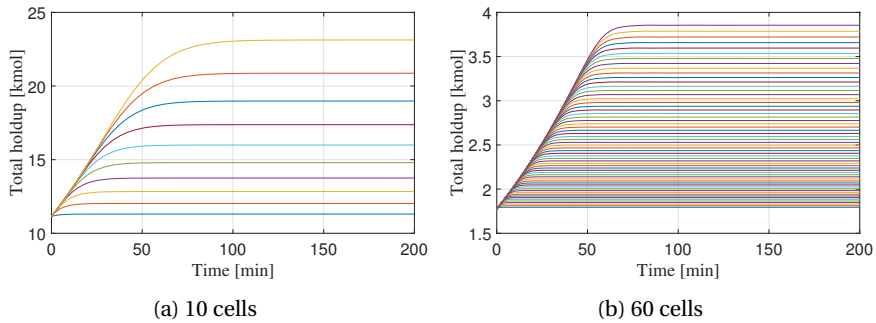


Figure 6.15: Dynamic total holdup profile for two phase simulation

The steady-state profile for the holdup is illustrated in figure 6.16 for 10 and 60 control volumes respectively. It can be observed that at steady-state the holdup for the case of 60 cells simulation is approximately 6 times lower compared to the 10 cells simulation which again matches the rapport of cell volumes for both cases.

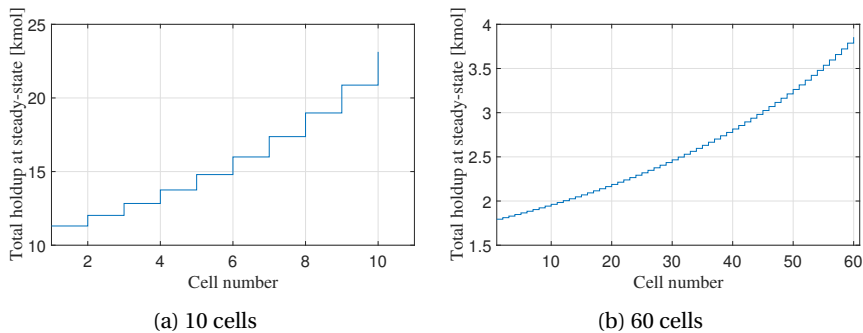


Figure 6.16: Steady-state total holdup profile for two phase simulation

The variation of pressure in time in each cell is shown in figure 6.17, both for a number of 10 and 60 lumps. The profile is similar with the vapor phase simulation pressure profile from figure 6.6 and thus, the same explanation for this profile applies.

The steady-state pressure profile is shown in figure 6.18, both for a number of 10 and 60 lumps and again is similar to the steady-state profile for vapor phase simulation

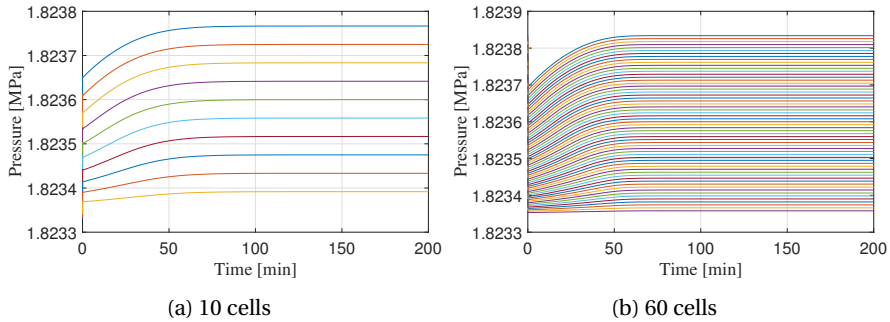


Figure 6.17: Dynamic pressure profile for two phase simulation

from figure 6.18.

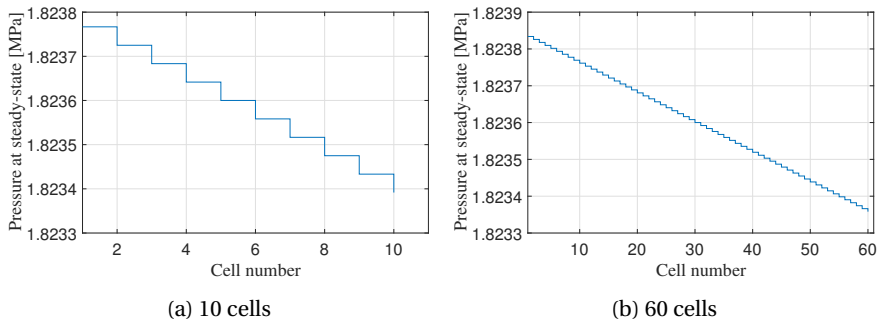


Figure 6.18: Steady-state pressure profile for two phase simulation

As in the case of vapor phase simulation, increasing the number of control volumes simulated from 60 cells to 10 cells produces a more accurate representation of the physical phenomena inside the heat exchanger, while the outlet conditions are equal in both cases. However, simulating the system for 60 cells takes around 20 time longer. Thus 10 cells are considered enough for a robust simulation for the case when the heat exchanger operates in the two phase region.

6.1.4.2 Varying heat with time

As in the case of the heat exchanger operated in vapor phase, a simulation of the heat exchanger with varying heat in time is performed for the two phase region. The simulation design and parameters are the same as in table 6.3. In connection to algorithm

1 the time constant until no heat is removed $t_1 = 20$ min while the time constant when the heat increases linearly until its maximum values is $t_2 = 100$ min.

The dynamic temperature profile is shown in figure 6.19 for both 10 and 60 cells. It can be observed in figure 6.19a that for the 10 control volume simulated the temperature is not constant for the first 20 minutes (i.e. the time when heat starts being removed) as it should be and as it for the case of 60 control volume simulated shown in figure 6.19b. However, in both cases the temperature starts decreasing after $t_1 = 20$ min and reaches a steady-state when the temperature in each cell has the same values as in the case with constant heat removed (figure 6.11, since the total amount of removed heat is the same).

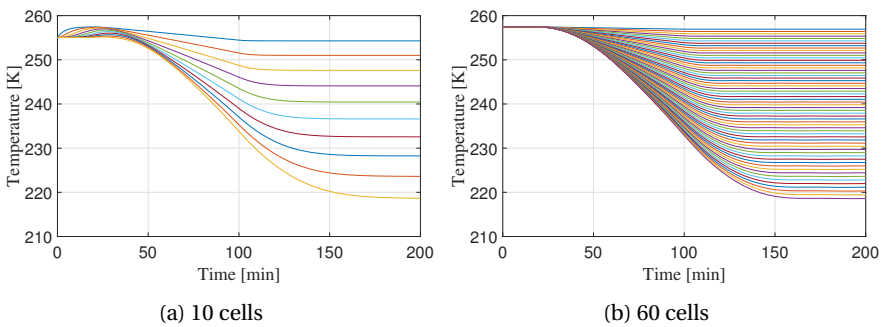


Figure 6.19: Temperature profile when heat is varying heat with time

The dynamic total holdup profile is shown in figure 6.20. The same variation in the total holdup profile for the first 20 minutes is observed in the case of 10 control volume simulated as in the temperature profile, phenomena which is expected since the two are correlated. Both holdup profiles start increasing as the temperature increases after $t_1 = 20$ min and reach a steady-state at the same conditions as in the case with constant heat removed shown in figure 6.16 as the total heat removed is the same.

The dynamic profile for liquid volume in figure 6.21 simulated for 10 and 60 control volumes follows the same tendencies as the temperature and holdup profiles.

The pressure profile also shows a difference between simulating for 10 cells versus simulating for 60 cells, as the profile for the first 20 minutes is not constant, though afterwards the same tendencies are observed. The profiles are similar with the profiles observed for the same type of simulation for vapor case shown in figure 6.10 as a variation in pressure exists between the time when the heat starts being removed at time

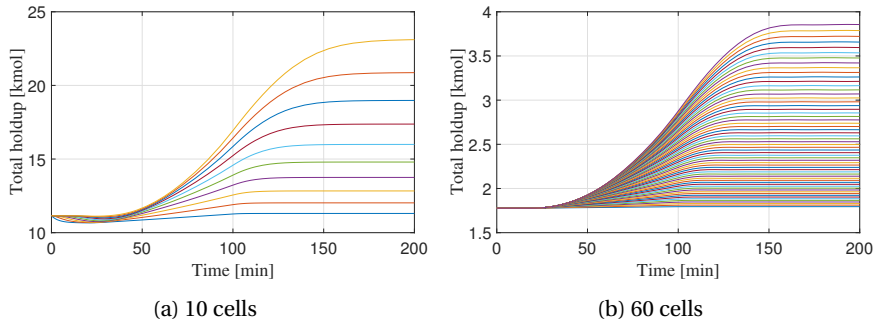


Figure 6.20: Total holdup profile when heat is varying heat with time

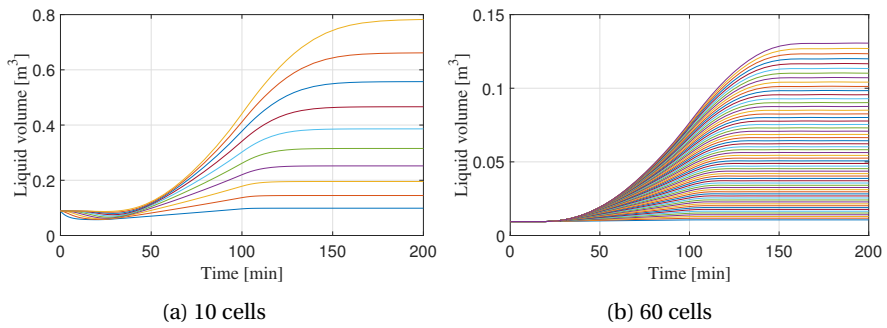


Figure 6.21: Liquid volume profile when heat is varying heat with time

$t_1 = 20$ and the time until it increases linearly to the maximum value of removed heat $t_2 = 100$. At the steady-state the pressure has the same profile compared to the simulation with constant heat shown in figure 6.17.

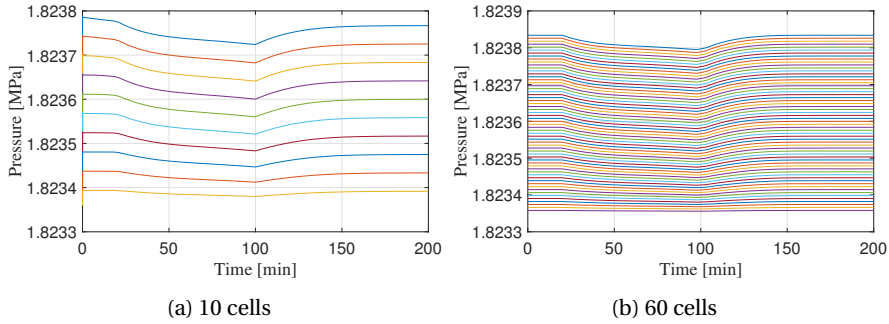


Figure 6.22: Pressure profile when heat is varying heat with time

Therefore, in this case, simulating for 60 control volumes gives better results at the beginning of the simulation. A reason for the differences could be an inconsistency of the initial conditions for the 10 cells case, meaning that the system is not actually at steady-state at the initial time step, and the solver tries to compensate for this.

6.1.5 Vapor phase to vapor-liquid region simulation

The next step after single vapor phase simulation and two phase simulation is to attempt to simulate the transition from vapor phase to vapor-liquid phase. The idea behind this simulation is to give as initial conditions values corresponding to the single phase region along with enough heat rate removed such that a transition between phases happens. The simulation and design parameters are similar with the one for single vapor simulation (see table 6.2), the only difference being in the values of the heat removed which is given as $Q_{HEX} = 12MJ/min$. A number of 10 control volumes are simulated and the heat removed is constant in time.

As the simulation is started in vapor phase the valid equations are equation 4.12 along with the dummy equations 4.14a for the variables accounting to the non-existent liquid phase. These variables are set to be equal to the dew point conditions, while $V_{L,0} = 0$. Then, as heat is removed from the fluid resulting in a temperature drop a transition into the two phase region occurs. At this point, the topology of the model

changes (i.e. the number of valid equations changes), and the switching algorithms presented in section 4.6.1.6 are applied to switch from the single phase equations to the two phase equations 4.10. This method of switching between equations has already been tested in the example presented in section 5.1 and it proved that *ode15s* can handle a change in the equations even if they are non-smooth as long as they are continuous.

Again both algorithm 2 which compares the cell temperature with the dew point and algorithm 3 which compares the liquid volume with a small tolerance for selecting the valid set of equations for each phase region are tested. However, both algorithm failed to correctly assign the valid set of equations when the phase change happens, and thus correctly determine the phase transition. A possible reason could be that in fact that at the switching point, the equations are in fact discontinuous, even though they were intended to be continuous in the model formulation.

The error that the solver *ode15s* gave with algorithm 2 when the temperature in the last cell became smaller than the dew temperature, was that it is unable to meet integration tolerance tolerance without reducing the step size below the smallest value allowed.

An interesting result is obtained with algorithm 3 which check if the liquid volume in a cell is below a certain tolerance. These simulation did give a result although not the expected one where the phase change is detected. In this case the temperature drops below the dew point of $T_{dew} = 262.05$ K (illustrated with a red line in figure 6.23b) as seen in the dynamic temperature profile and steady-state temperature profile from figure 6.23a and figure 6.23b respectively. The liquid volume does not increase above zero however as seen in figure 6.23c, meaning that in fact the system does not transitions to the two phase region. This state would correspond to a metastable vapor, i.e. a stable fluid that it is still in vapor phase even though it has a temperature and pressure corresponding to the two phase region and a small energy kick would make the phase transition happen. The explanation for this behavior of the model is that while dummy equations are used in order to have the same number of equations in all phases, they in fact are not correlated with the other variables of the system which have a physical meaning. In other words, the variable for the non-existent phase are constant in time and are independent of the other changing variables, meaning that there is no equations to account for a increase in liquid volume. Therefore this variable are always kept at its initialized value $V_{L_0} = 0$, and thus only the equations for the single phase are solved. Since these equations do not have a vapor-liquid equilibrium condition their

solution is not a problem for the solver.

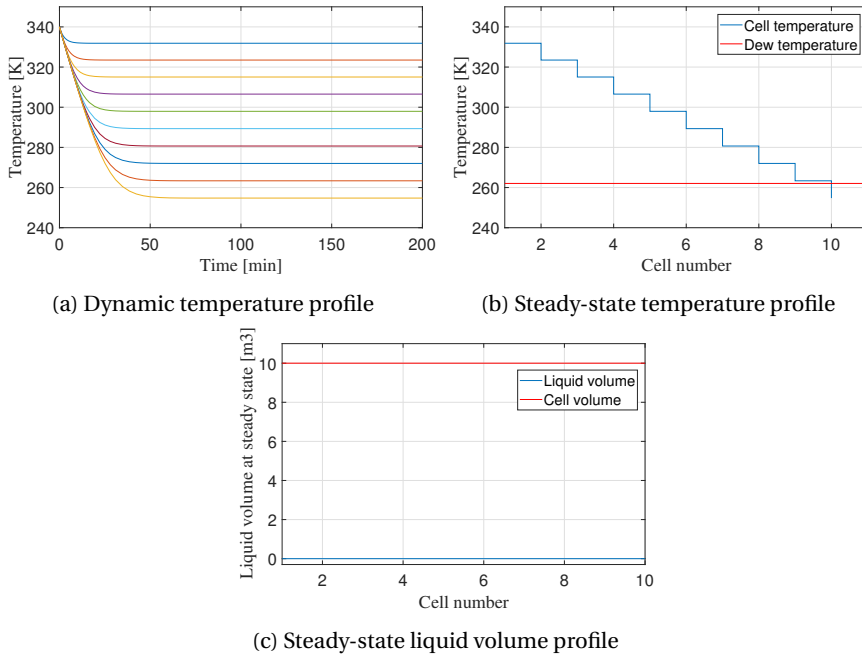


Figure 6.23: Profiles for a transition from vapor phase to vapor-liquid region with algorithm 3

Based on these results it is not understood that a robust switching algorithm must be based on a criteria containing a variable that has a physical meaning in both phases (as temperature) and not just in one of the phases since the variable in cause needs to modify its values in order that the algorithm achieves an accurate switch of topology.

6.1.6 Vapor phase to vapor-liquid region with event detection simulation

As integrating over the phase transition was not successfully a new approach has to be found. For this purpose the event function described in section 5.1.4 is passed as an option to the solver *ode15s*.

Referring to the structure of the event function, a mathematical expression has to be chosen such that its evaluation is equal to zero when an event occurs. A good candi-

date for the mathematical expression for detecting a transition between the vapor and vapor-liquid phase could be the difference between the cells temperature and the dew temperature. However, this implies that both values are equal to machine precision, which is no guarantee that it will happen. Therefore it is chosen to use the logical operator $>$ to compare the difference between the cells temperature with the dew point with a given tolerance in order to always obtain a binary result (e.g. true = 1; false = 0) and make sure that the event is triggered. The expression for the *value* argument of the event function is given in equation 6.2.

$$value = T - T_{dew} > tolerance \quad (6.2)$$

The event function has the options of either terminating or continuing integration when the event occurs (see section 5.1.4). Considering that integration over the phase change point described in the previous section 6.1.5 was unsuccessful, it is chosen that the integration is automatically terminated by the solver *ode15s* when event 6.2 occurs. Then, a new simulation is performed which uses as initial conditions the results of the first simulation. To be more precise, the conditions of the last cell resulted in the first simulation are used as initial conditions for all the cells in the second simulation, meaning that in the second simulation all the cells have dew point conditions at the initial time step.

The idea behind running this simulation follows the same pattern as in section 6.1.5. Therefore, the simulation starts in the single phase region and algorithm 2 is applied to select the set of equations valid for the single vapor region. As the event is detected automatically by the *ode15s* solver, integration terminates. Then, the simulation is started again, this time in the two phase region from the dew point, and the same algorithm 2 is applied to select the set of equations valid for the two phase region this time. Thus, the same Matlab[®] is used to run both simulations while the solver handles the phase change, which is important if one does not know *a priori* when the phase transition happens.

An important observation is that both the *value* argument and algorithm 2 compare the difference between the cells temperature and the dew temperature with a given tolerance. In order for the event function to successfully detect the phase change, the tolerance in the event function has to be larger (one order of magnitude) compared to the tolerance in algorithm 2 such that the event is triggered before the change in topol-

ogy and errors in integrating are avoided.

$$\text{tolerance in equation 6.2} > \text{tolerance in equation 2} \quad (6.3)$$

The simulation and design parameters are given in table 6.4. A number of 10 control volumes are used for each simulation (i.e. before and after the phase change). The total volume of the heat exchanger and the pressure drop are taken as double the values of the vapor phase or two phase simulations. The heat removed is considered to be constant in time.

Table 6.4: Simulation and design parameters for two phase with event function

Parameter	Value	Units
T_{inlet}	340	K
Q_{HEX}	20	MJ/min
V_{HEX}	$2 \cdot 100$	m^3
ΔP_{max}	$2 \cdot 5$	mbar
time	150	min
number of cells	$2 \cdot 10$	-

The results of the simulation from single vapor phase to two phase region with an automatically phase change detection handled by the solver are illustrated in figure 6.24 to 6.27. It can be observed that the transition between the two phase regions is detected at time $t_{transition} = 23 \text{ min}$. Until the mentioned time, the profiles correspond to the single vapor phase simulation, while after the profile corresponds to vapor-liquid region simulation. This implies that applied to the heat exchanger model, the event function will detect a phase transition in time and not in space (i.e. the cell number), as it would be ideally desired. With this approach, the location of the phase change is actually fixed, considering that the temperature decreases continuously and thus the last cell is the first cell where liquid droplets are formed in a dynamic regime. This further implies that the last cell has reached the conditions for the two phase region while all the other cells are still in single phase, meaning that not all cells have the same set of equations.

Another implication of this approach is that as the first liquid droplets are firstly formed in the last cell, the phase change will actually travel backwards along the heat exchanger, from the last cell M to cell $M - 1$ and so on. However, this only the phase

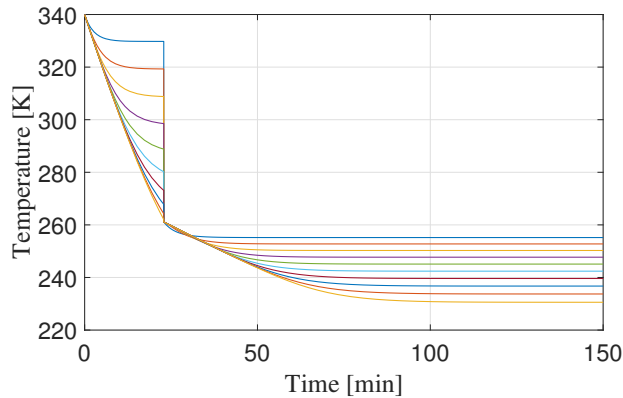


Figure 6.24: Dynamic temperature profile when using event function

change in the last cell is captured in this simulations and in the figures below.

The shape of the plots is explained considering that the first integration terminates when the temperature of the last cell reaches the dew point, even though the temperature in previous cells are well above the dew point. Then, in the second simulation, all cell start at the condition of the last cell of the first simulation. This implies that one has no knowledge what happens in the first 9 cells which are in the single region, and in the $10^t h$ cells which is at dew point condition while running the second simulation for cells 11 to 20, and therefor a discontinuity is observed in the figures.

The dynamic temperature profile is shown in figure 6.24. The temperature has values above the dew point in until the transition time and temperature equal of below the dew temperature afterwards.

The dynamic liquid volume profile is shown in figure 6.25. As the liquid volume is given as a dummy variable in the vapor phase, its values before the transition is zero, and after $t_{transition} = 23 min$, the liquid volume starts increasing in time in each cell.

The dynamic total holdup profile is shown in figure 6.26. This profile has a good correlation with the temperature profiles, as in the case of vapor phase and two phase simulations. The holdup increases as the temperature decreases since the fluid density increases.

The dynamic pressure profile is shown in figure 6.27. The variation of pressure fol-

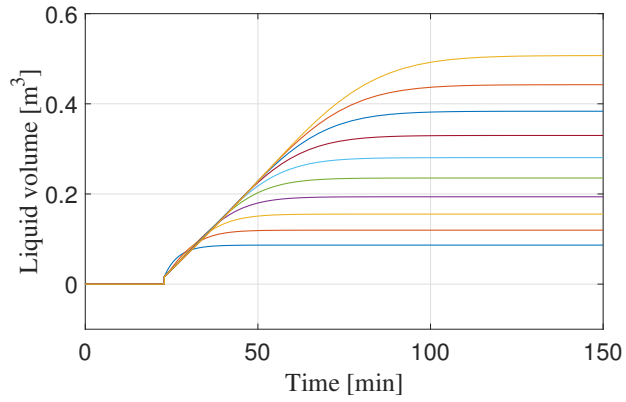


Figure 6.25: Dynamic liquid volume profile when using event function

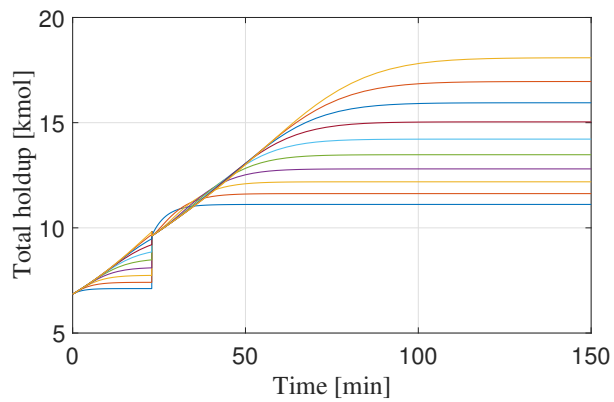


Figure 6.26: Dynamic total holdup profile when using event function

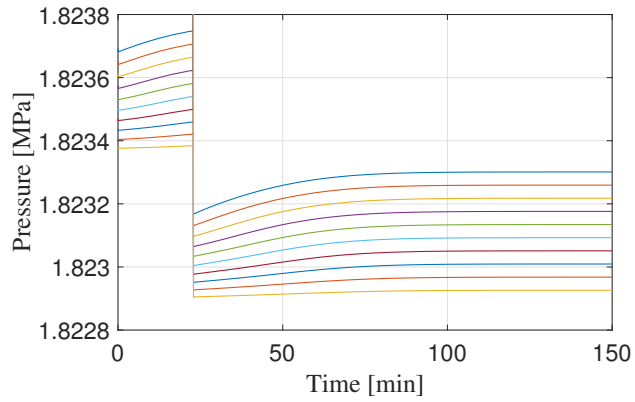


Figure 6.27: Dynamic Pressure profile when using event function

lows the same pattern as in the case of vapor and vapor-liquid simulations, i.e. although a positive pressure drop exists from one cell to another, there is a drop in all cells at the initial time steps. However, the pressure reaches a steady-state.

Chapter 7

Discussion

This chapter consists on a few supplementary considerations on the heat exchanger model, the solver used and the simulations performed.

7.1 The heat exchanger model

The initial idea of this work was to develop a simple and yet robust dynamic model for a multicomponent heat exchanger with phase change which could then be used to optimization and control purposes of the entire reliquefaction process described by [Nekså et al., 2010]. However, the trajectory of the work was adjusted along the way due to the fact that a multicomponent phase change process is more challenging to model and simulate than originally considered. The main challenge arise firstly from a change in the model topology (i.e. the number of valid equations and number of variables with a physical meaning) for each region of the phase diagram (from figure 4.1). The source in the change of the number of equations is the infeasibility of applying the vapor-liquid equilibrium condition in either of the single phase since the variables accounting for the non-existent phase do not have physical meaning and this will result in a numerical problem when solving. If one phase does not exist, the conditions for consistency of compositions, $\sum_i x_i = 1$, is clearly not satisfied since all components do not exist in the respective phase.

The ramifications of these challenges is that a robust model without logical conditions, disjunctions or making use of a non-smooth solver is potentially impossible and hopeless to formulate such that it represents with accuracy and precision the real phase change process with multicomponent streams. Indeed a few wishful thinking attempts of using artificial mathematical tricks in order to circumvent the change in the model topology in order to have a single set of equations for all phases have been tried, yet their implementation failed to produce a feasible solution to the problem. Therefore, the main focus of this work has been implementing in Matlab[®] a different set of equations for each phase and have a logical condition that compares the cells temperature with the dew and bubble points to select the valid set of equations, thus having a disjunctive model. The *ode15s* was not able to properly handle this model and the phase change was detected in time while its location is fixed at the last cell. For this reason the possibility of solving this problem as a mathematical problem with complementarity constraints (MPCC) was investigated towards the end of the thesis. This approach seems more optimistic and promising in dealing with the phase change both in space and time while maintain an equation oriented environment. However, due to insufficient time, a model for the heat exchanger based on MPCC was not fully developed and simulated and is consequently left as future work.

7.1.1 The assumptions

The model includes only one of the two streams of the actual heat exchanger as described in section 2.1. Instead of modelling the heat transferred between the two streams, this model considers that the removed heat rate from the modelled stream is given. Therefore, no heat transfer limitations are captured by the model. In addition, heat losses to the environment and wall capacity are neglected. The reason behind this simplification was to properly investigate how the phase change could be modelled and then extend the formulation to the second stream and include heat losses and wall capacity.

The "no slip" assumption is a coarse simplification for a multiphase flowing fluid. However it is a decent approximation considering this model is intended to be a robust model. If one wants to include the slip ratio and model for the different phase velocities and phase holdups, one must know beforehand the flow pattern which is not a trivial task.

In the real process the pressure drop is a function of the geometry of the equipment. However, as design data for the heat exchanger is not available, the pressure drop was assumed as given in this work. In the case that design data does become available, a momentum equations to account for the changes in pressure along the heat exchanger could be included in the model in order to have a more realistic variation of pressure in time and along the cells. As in this model the outlet flow and as a consequence the total holdup are given as an explicit function of the pressure drop, including a momentum balance for the pressure drop could potentially lead to a more accurate representation of the actual process.

7.2 The solver and simulations

The modelling tool used to solve the set of differential and algebraic equations was Matlabs[®] *ode15s* solver which comes with a few restrictions. Due to not knowing before hand the space-time coordinates of the phase change, yet wanting to call the same Matlab[®] script regardless of the phase region, the number of unknown variables and equations must be the same for all phases in order to keep the size of the mass matrix constant.

A drawback of using the *ode15s* solver with different set of equations is that the integration cannot continue after the phase transition has been detected with the event function. A possible explanation for this behavior is that after the last cells reaches the conditions for the two phase region, the cells before are still in the single region, meaning that at the same time step in the integration two different sets of equations have to be used for the respective cells. As Matlab[®] is designed to operate with matrices, all the model equations have to be stacked in a single matrix, which means that after the phase change, the structure of this matrix changes. Whether or not this may affect the solving algorithm, is not fully understood yet. A possibility to overcome this would be to simulate one cell at a time and pass the outlet conditions from one cell to the next one, although this method is not feasible for also simulating the countercurrent stream since the first cell of the first stream transfers heat with the last cell of the second stream. In order to determine the conditions of the last cells of the second stream, one needs to know the conditions of the previous cell and so on. This also extends to the non-applicability of Matlabs[®] event function when both counter-current stream change phase since this in fact would mean having minimum two and maximum four

simulations (two if both both change phase at the same time, four otherwise) and needing the result of the second simulation in order to conduct the first one, which of course is not possible.

Another shortcoming in using *ode15s* for solving a system of differential and algebraic equations is that the solver seemingly does not solve internally for all the roots of an algebraic equation, but it determines the root closest to the initial values supplied. In addition, the change from all real roots to some real and some complex conjugate is not captured by the solver as illustrated in section 5.1.3. Therefore, it may be beneficial for simulating a multicomponent phase change process in Matlab[®] with *ode15s* to use the more robust two layer approach and determine the roots of the cubic equation of states in an inner layer. Then, depending on the type of roots, decision variables could be used to cancel the terms or equations accounting for the non-existent phase, in a similar manner as it was investigated in the example from section 5.1.3.

As most of the time of this work was dedicated on investigating the implications of a transition from vapor to vapor-liquid, simulations for the liquid phase have not been performed. However, there is no reason to consider the transition from two phase region to liquid region as conceptually different to a vapor to vapor-liquid transition.

Chapter 8

Conclusions

A dynamic heat exchanger model for a multicomponent system that changes phase is developed. The finite control volume is used to spatially discretize the heat exchanger into a given number of cells with equal volume. Soave modification of the Redlich-Kwong cubic equation of state is used to model the thermodynamic properties of the model.

The main challenge was having different types and numbers of valid equations in each of the phase region. This is caused by the infeasibility of the vapor-liquid equilibrium condition in the single regions. This was addressed by formulating a different topology (i.e. set of equations) for each phase, where the vapor-liquid equilibrium (based on K-values) is included as an algebraic equation only for the two phase region. In order to call the same script for all phases, the number of equations and variables have to be the same in all phases. Therefore, dummy variables and equations (without a real significance for the process) were introduced for the single phase formulations. In order to change the model topology, a logical condition based comparison of the cell temperature with the dew or bubble point was successfully used.

Matlabs[®] *ode15s* was used for simulations. The heat exchanger is simulated for the single vapor region, the vapor-liquid region and for a transition between the vapor to the two phase region. The *event* function is passed as option to the *ode15s* solver to automatically detect a phase change from the vapor region to the two phase region. However, this was proven not to be the best approach for the purpose of this model.

In addition to the heat exchanger model, the *ode15s* solver is also tested for two additional examples, in order to investigate how the solver treats non-smooth systems, or how the algebraic equations are solved. It has been determined that this solver is able to handle non-smooth but continuous differential and algebraic equations systems. However, for the algebraic equations, the solver determines only one of the roots (if more than one exist), namely the root that is closest to the initial condition. Moreover, it does not keep track if the roots are changing from the real plane to the complex one.

A way forward for this project is seen in formulating the model as a mathematical problem with complementarity constraints. This approach was shortly investigated. However, due to time constrictions, this method was not implemented nor simulated.

8.1 Future work

The first step in a future work should probably be selecting a different solver than *ode15s* in Matlab[®]. This should be selected such that there is a better control on the phase change in space, not only in time, as it was the case in this thesis. In addition, the simulation should continue integrating after the phase change happens. For this purpose, it is recommended to continue the development of the dynamic heat exchanger model as a mathematical problem with complementarity constraints and simulated using the collocation method to discretize the differential equations and an non-linear programming (NLP) solver. This would also be an advantage in optimizing the phase change process. Further, the second stream should be integrated in the model and finally, this heat exchanger model should be coupled with models of all other components of the reliquefaction cycle. The last recommendation is to validate the model with real plant data.

Bibliography

- L. T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Society for Industrial and Applied Mathematics, 2010. ISBN 978-0-89871-702-0. doi: doi.org/10.1137/1.9780898719383. URL <http://epubs.siam.org/doi/book/10.1137/1.9780898719383>.
- O. Bratland. *Pipe Flow 2 : Multi-phase Flow Assurance*. 2013. ISBN 9786163359261. URL <http://www.drbratland.com/wp-content/uploads/2013/10/PipeFlow2-Chapter-2.pdf>.
- M. A. Duran and I. E. Grossmann. Simultaneous Optimization and Heat Integration of Chemical Processes. *AIChE Journal*, 32(1):123–138, 1986. doi: DOI:10.1002/aic.690320114.
- V. Gopal and L. T. Biegler. Smoothing Methods for Complementarity Problems in Process Engineering. *AIChE Journal*, 45(7), 1999.
- T. Haug-Warberg. 46 Workouts in Thermodynamics. Technical report, DEPARTMENT OF CHEMICAL ENGINEERING, NTNU. URL <http://folk.ntnu.no/haugwarb/KP8108/LectureNotes/notes.pdf>.
- M. Herty and S. Steffensen. MPCC Solution Approaches for a Class of MINLPs with Applications in Chemical Engineering. Technical report, Aachen Institute for Advanced Study in Computational Engineering Science, Aachen, 2012.
- R. S. Kamath, I. E. Grossmann, and L. T. Biegler. Modeling of multi-stream heat exchangers with phase changes for cryogenic applications. *Computer Aided Chemical Engineering*, 27(C):921–926, 2009. ISSN 15707946. doi: 10.1016/S1570-7946(09)70374-8.
- R. S. Kamath, L. T. Biegler, and I. E. Grossmann. An equation-oriented approach for handling thermodynamics based on cubic equation of state in process optimiza-

- tion. *Computers and Chemical Engineering*, 34(12):2085–2096, 2010. ISSN 00981354. doi: 10.1016/j.compchemeng.2010.07.028. URL <http://dx.doi.org/10.1016/j.compchemeng.2010.07.028>.
- R. S. Kamath, L. T. Biegler, and I. E. Grossmann. Modeling multistream heat exchangers with and without phase changes for simultaneous optimization and heat integration. *AIChE Journal*, 58(1):190–204, 2012. ISSN 00011541. doi: 10.1002/aic.12565.
- MathWorks. ODE Event Location, a. URL <https://se.mathworks.com/help/matlab/math/ode-event-location.html>.
- MathWorks. Solve Differential Algebraic Equations (DAEs), b. URL <https://se.mathworks.com/help/matlab/math/solve-differential-algebraic-equations-daes.html>.
- P. Nekså, E. Brendeng, M. Drescher, and B. Norberg. Development and analysis of a natural gas reliquefaction plant for small gas carriers. *Journal of Natural Gas Science and Engineering*, 2(2-3):143–149, 2010. ISSN 18755100. doi: 10.1016/j.jngse.2010.05.001. URL <http://dx.doi.org/10.1016/j.jngse.2010.05.001>.
- J. Nocedal and S. Wright. *Numerical optimization*. 2006. ISBN 9780387303031.
- H. Pangborn, A. G. Alleyne, and N. Wu. A comparison between finite volume and switched moving boundary approaches for dynamic vapor compression system modeling. *International Journal of Refrigeration*, 53:101–114, 2015. ISSN 01407007. doi: 10.1016/j.ijrefrig.2015.01.009. URL <http://dx.doi.org/10.1016/j.ijrefrig.2015.01.009>.
- A. Péneloux, E. Rauzy, and R. Fréze. A consistent correction for Redlich-Kwong-Soave volumes. *Fluid Phase Equilibria*, 8(1):7–23, 1982. ISSN 03783812. doi: 10.1016/0378-3812(82)80002-2.
- B. E. Poling, J. M. Prausnitz, and J. P. O’Connell. *The properties of gases and liquids*. McGraw-Hill, New York, 5th edition, 2001. ISBN 0070116822.
- H. A. Preisig. *Lecture notes. The ABC of modelling*. PhD thesis, NTNU, 2016.
- R. C. Reid, J. M. Prausnitz, and B. E. Poling. *The Properties of Gas & Liquids*. McGraw-Hill, 4th edition, 1987.

- A. Reyes-Lúa, M. Solvik, and S. Skogestad. Inclusion of thermodynamic equations for efficient steady-state process optimization. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering*, 2016. ISBN 9780444634283. doi: 10.1016/B978-0-444-63428-3.50107-7. URL <http://dx.doi.org/10.1016/B978-0-444-63428-3.50107-7>.
- A. M. Sahlodin, H. A. J. Watson, and P. I. Barton. Nonsmooth Model for Dynamic Simulation of Phase Changes. *AIChE*, 62(9), 2016. doi: DOI10.1002/aic.15378.
- V. Shalashilin and E. Kuznetsov. *Parametric Continuation and Optimal Parametrization in Applied Mathematics and Mechanics*. SPRINGER-SCIENCE+BUSINESS MEDIA, 1st edition, 2003. ISBN 9789048163915. doi: 10.1007/978-94-017-2537-8.
- L. F. Shampine and M. W. Reichelt. The Matlab ODE suite. URL https://www.mathworks.com/help/pdfdoc/ode/ode_suite.pdf.
- S. Skogestad. *Chemical and Energy Process Engineering*. CRC Press, 2008.
- S. Skogestad, A. Linhart, and M. Solvik. MATLAB routines that use the SRK equation of state, 2013. URL <http://www.nt.ntnu.no/users/skoge/book-cep/matlab/srk-flash/>.
- G. Soave. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chemical Engineering Science*, 27(6):1197–1203, 1972. ISSN 00092509. doi: 10.1016/0009-2509(72)80096-4.
- C. F. Spencer and R. P. Danner. Improved equation for prediction of saturated liquid density. *Journal of Chemical and Engineering Data*, 17(2):236–241, 1972. ISSN 0021-9568. doi: 10.1021/je60053a012. URL <http://pubs.acs.org/doi/abs/10.1021/je60053a012>.
- H. A. J. Watson, K. A. Khan, and P. I. Barton. Multistream Heat Exchanger Modeling and Design. *AIChE*, 61(10):3390–3403, 2015. doi: 10.1002/aic.14965.
- H. A. J. Watson, D. Kim, T. Gundersen, and P. I. Barton. *Modeling and simulation of phase change and non-ideality in multistream heat exchangers*, volume 38. Elsevier Masson SAS, 2016. ISBN 9780444634283. doi: 10.1016/B978-0-444-63428-3.50089-8. URL <http://dx.doi.org/10.1016/B978-0-444-63428-3.50089-8>.
- Ø. Wilhelmsen, G. Skaugen, M. Hammer, P. E. Wahl, and J. C. Morud. Time Efficient Solution of Phase Equilibria in Dynamic and Distributed Systems with Differential

Algebraic Equation Solvers. *Industrial & Engineering Chemistry Research*, 52:2130–2140, 2013. ISSN 0888-5885. doi: 10.1021/ie302579w.

R. E. Zartman, G. J. Wasserburg, and J. H. Reynolds. Helium , Argon , and Carbon in Some Natural Gases. 66(1), 1961. doi: 10.1029/JZ066i001p00277.

C. Zotică. *Dynamic modelling of a HEX with phase change*. Specialization project, NTNU, 2016.

Appendix A

Notations

A.1 Acronyms

CEoS Cubid equation of state

DAE Differential Algebraic Equations

EoS Equation of State

FCV Finite Control Volume

HEX Heat exchanger

LNG Liquid Natural Gas

MINLP mixed integer nonlinear program

MHEX Multistream Heat Exchnager

MPCC Mathematical problem with complementarity constraints

MR Mixed Refrigerant

NC Number of Components

NLP Nonlinear programming

ODE Ordinary Differential Equations

SRK Soave Redlich Kwong

VLE Vapor-Liquid Equilibrium

A.2 List of Symbols

A Helmholtz energy in section 3.1.2

A SRK parameter

B SRK parameter

c correction factor for liquid molar volume

c_p specific heat

ΔT_{LM} log-mean temperature difference

EBP extended pinch location function

F molar flow

G(x) inequality constraints in generalized disjunctive optimization problem

h enthalpy

H specific enthalpy

H(x) equality constraints in generalized disjunctive optimization problem

K VLE constant

k valve coefficient

k_{i,j} interaction parameter

M cells number

M_M Mass matrix

m SRK parameter (slope)

n holdup

T total holdup

v vapor fraction

P pressure

Q exchanged heat

R universal gas constant

s slack variable

t time

t₁ time until the removed heat is 0

t₂ time until the removed heat increase

T temperature

u internal energy

U specific internal energy

UA overall heat transfer coefficient multiplied by heat transfer area

V volume

V_m molar volume

V_L liquid volume

x vapour molar composition

x variable in general mathematical formulations

y liquid molar composition

y variable in general mathematical formulations

Y discrete decision variable

z molar composition

Z compressibility factor

Z_{RA} rackett compressibility factor

A, B, C, D, E polynomial coefficients for specific heat

∨ exclusive or

A.3 Subscripts

0 inlet/initial

bub bubble point

c critical in respect to SRK Eos

c cold in respect to HEX model

cell applies for one cell

dew dew point conditions

h hot

HEX applies along the heat exchanger

i component number

in inlet

j cell number

L liquid

max maximum

min minimum

out outlet

V vapor

r reduced

A.4 Greek Letters

α SRK correction parameter

β relaxation parameters for vapor-liquid equilibrium

δ fugacity correction parameter

Δ difference

γ activity coefficient

μ chemical potential in section 3.1.2

μ relaxation parameter in section 2.2.2

ξ parameter of the cubic equation of state. can be compressibility or volume

ϕ fugacity

ρ density

θ general property

$\psi(x)$ objective function

ω acentric factor

Appendix B

Thermodynamic constants

B.1 Coefficients for specific heat

Poling et al. [2001]

Table B.1: Polinomial coefficients for calculation of $\frac{C_p}{R}$

Coefficient	A	B · 10 ³	C · 10 ⁵	D · 10 ⁸	E · 10 ¹¹
Nitrogen	3.539	-0.261	0.007	0.157	-0.099
Methane	4.568	-8.975	3.361	-3.407	1.091
Ethane	4.178	-4.427	5.660	-6.651	2.487
Propane	3.837	5.131	6.011	-7.893	3.079

B.2 Interaction Parameters

Taken from Aspen Hysys[®]V9

B.3 Critical Pressure, Temperature and Acentric Factor

Taken from Poling et al. [2001]

Table B.2: Interaction parameters

Component	Nitrogen	Methane	Ethane	Propane
Nitrogen	0	3.11990007758e-2	3.18990014493e-2	8.86000022292137e-2
Methane	3.119900077581e-2	0	2.24137306213379e-3	6.82878494262e-3
Ethane	3.189900144934e-2	2.24137306213e-3	0	1.25795602798e-3
Propane	8.860000222921e-2	6.82878494265e-3	1.25795602798e-3	0

Table B.3: Critical Pressure, Temperature and Acentric Factor

Component	P_c MPa	T_c K	ω
Nitrogen	3.398	126.2	0.037
Methane	4.599	190.56	0.011
Ethane	4.872	305.32	0.099
Propane	4.4248	369.83	0.0152

Appendix C

Matlab scripts

C.1 Main script one phase simulation

```
% June 2017. Main code for running a simulation for vapor phase.

% M-nr cells; NC-number of components (1)Nitrogen - (2)Methane - (3)Ethane
% - (4)Propane; V-HEX Volume, [m3]; F0-inlet flow, [kmol/min];
% Pin-inlet pressure, [MPa]; % DP-pressure drop along HEX, [MPa];
% Tin-inlet temp, [K]; Qmax-heat removed, [MJ/min]; t1-time
% when Q!=0, [min]; t2-time when Q reaches max, [min]; Z1P-Z for 1P; ZG0-Z
% for gas for 2P; z0-overall molar fraction; xin-liq inlet comp for 2 phase;
% yin-vap inlet comp for 2 phase; Tdew-dew point; Tbub-bubble point;
% param = [N,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin, Tdew,Tbub]

clear;close all;clc

M = 10; NC = 4; V = 100; F0 = 0.0445*60;
Pin = 18*0.101325; DP = 5*1e-3*0.1; Tin = 67+273.15;
Qmax = 10; t1 = 0; t2 = 0;
Z1P = 0.9421; % at 67 degC
ZG0 = 0.8527; ZL0 = 0.0674; v0 = 1; % at -11.1 degC
z0 = [0.06 0.4 0.4 0.14];
xin = [0.0035 0.0656 0.3985 0.5324];
yin = [0.06 0.4 0.4 0.14]; % at -11.5 degC, from dew point calculation
Tdew = -11.1+273.15; Tbub = -107.9+273.15; % mean from Pin to Pin-DP
```

```

param = [M,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin, Tdew, Tsub];

% load initial conditions for 1 phase (vapor) simulation

load w01P-10Q0 % 10 cells
%load w01P-60Q0 % 60 cells

Pout = Pin-DP;

nT0 = w1(end,1:M); % total holdup in (both phases), [kmol]
n0 = zeros(1,NC*M);
for j=1:M % component 1 holdup in tank (both phases), [kmol]
    n0(NC*j-3:NC*j) = nT0(j).*z0;
end

U0 = w1(end,M+NC*M+1:2*M+NC*M); % total internal energy (both phases), [MJ]
T0 = ones(1,M)*Tin; % temperature (same in both phases), [K]

y0=zeros(1,NC*M); %vapor composition = overall compositions (no phase change)
for j=1:M
    y0(j*NC-(NC-1):j*NC) = yin(1:4);
end

Zg0 = ones(1,M)*Z1P; % gas compressibility
P00 = linspace(Pin,Pin-DP,M+2); % pressure inlet to outlet, [MPa]
P0 = P00(2:end-1); % pressure inside the HEX

x0 = zeros(1,NC*M); % dummy liquid compositions
for j=1:M
    x0(j*NC-(NC-1):j*NC) = xin(1:4);
end

VL0 = w1(end,3*M+3*M*NC+1:4*M+3*M*NC); % dummy liquid volume
Zl0 = ones(1,M)*0.06; % dummy liquid comprsibility

w0 = [nT0,n0,U0,T0,x0,y0,VL0,Zg0,Zl0,P0]; % vector of initial condition

%ODE solver options*****

nrDyn = NC+2; %number of dynamic equations per cell
nrAlg = 2*NC+5; %number of algebraic eq per cell
d1 = ones(1,nrDyn*M);
d2 = zeros(1,nrAlg*M);
Mass = diag([d1,d2]);

```

```
options = odeset('mass',Mass,'Stats','on');

%ODE solver;integrate with algortihm based on comparison with the dew point
[t,w]=ode15s(@(t,w) HEX_T(t,w,param),[0 200],w0,options);
```

C.2 Main script two phase simulation

```
% June 2017. Main code for runing a simulation for vapor-liquid phase.

% M-nr cells; NC-number of components (1)Nitrogen - (2)Methane - (3)Ethane
% - (4)Propane; V-HEX Volume, [m3]; F0-inlet flow, [kmol/min];
% Pin-inlet pressure, [MPa];% DP-pressure drop along HEX, [MPa];
% Tin-inlet temp, [K]; Qmax-heat removed, [MJ/min]; t1-time
% when Q!=0, [min]; t2-time when Q reaches max, [min]; Z1P-Z for 1P; ZG0-Z
% for gas for 2P; z0-overall molar fraction; xin-liq inlet comp for 2 phase;
% yin-vap inlet comp for 2 phase; Tdew-dew point; Ttub-buble point;
% param = [N,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin, Tdew,Ttub]

clear;close all;clc

M = 10; NC = 4; V = 100; F0 = 0.0445*60;
Pin = 18*0.101325; DP = 5*1e-3*0.1; Tin = -11.1+273.15;
Qmax = 20; t1 = 20; t2 = 100;
Z1P = 0.9421; % at 67 degC
ZG0 = 0.8547; ZL0 = 0.066; v0 = 1; % at -11.5 degC
z0 = [0.06 0.4 0.4 0.14];
xin = [0.003 0.0639 0.3967 0.5364];
yin =[0.06 0.4 0.4 0.14]; % at -11.5 degC, from dew point calculation

Tdew = -11.1+273.15; Ttub = -107.9+273.15; % mean from Pin to Pin-DP

param = [M,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin, Tdew, Ttub];

% load initial conditions for 2 phase (vapor-liquid) simulation
load w02P-10Q0 % 10 cells
%load w02P-60Q0 % 60 cells

nT0 = w1(end,1:M); % total holdup in (both phases), [kmol]
n0 = zeros(1,NC*M);
for j=1:M % component holdup in tank (both phases), [kmol]
    n0(NC*j-3:NC*j) = nT0(j).*z0;
end
```

```

U0 = wl(end,M+NC*M+1:2*M+NC*M);           % total internal energy (both phases)
[kJ]
T0 = ones(1,M)*Tin;                         % temperature (same in both phases)
[K]
x0 = wl(end,3*M+M*NC+1:3*M+2*M*NC);        % liquid compositions
y0 = wl(end,3*M+2*M*NC+1:3*M+3*M*NC);      % vapor compositions

Zg0 = ones(1,M)*ZG0;                        % vapor compressibility
P00 = linspace(Pin,Pin-DP,M+2);            % pressure
P0 = P00(2:end-1);

VL0 = wl(end,3*M+3*M*NC+1:4*M+3*M*NC);
Zl0 = ones(1,M)*ZL0;

w0 = [nT0,n0,U0,T0,x0,y0,VL0,Zg0,Zl0,P0]; % vector of initial condition

%ODE solver options*****

nrDyn = NC+2;                               %number of dynamic equations per cell
nrAlg = 2*NC+5;                             %number of algebraic eq per cell
d1 = ones(1,nrDyn*M);
d2 = zeros(1,nrAlg*M);
Mass1 = diag([d1,d2]);
options = odeset('mass',Mass1,'Stats','on');

%ODE solver; integrate with algorithm based on comparison with the dew point
[t,w]=ode15s(@ (t,w) HEX_T(t,w,param),[0 200],w0,options);

```

C.3 Main script one phase to two phase simulation

```

% June 2017. INFO: uses event function to stop integrating when dew point
%is reached.
% M-nr cells; NC-number of components (1)Nitrogen - (2)Methane - (3)Ethane
% - (4)Propane; V-HEX Volume,[m3]; F0-inlet flow,[kmol/min];
% Pin-inlet pressure,[MPa]; % DP-pressure drop along HEX,[MPa];
% Tin-inlet temp,[K]; Qmax-heat removed,[MJ/min]; t1-time
% when Q!=0,[min]; t2-time when Q reaches max,[min]; Z1P-Z for 1P; ZG0-Z
% for gas for 2P; z0-overall molar fraction; xin-liq inlet comp for 2 phase;
% yin-vap inlet comp for 2 phase; Vliq-dummy liquid volume used to start 2P
% Tdew-dew point; Tsub-bubble point;
% param=[N,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin,Tdew,Tsub]

```

APPENDIX C. MATLAB SCRIPT ONE PHASE TO TWO PHASE SIMULATION

```

% OBS. tol in event function needs to be smaller then tol used in HEX.m
clear; close all;clc

M = 10; NC = 4; V = 100; F0 = 0.0445*60;
Pin = 18*0.101325; DP = 5*1e-3*0.1; Tin = 67+273.15;
Qmax = 15; t1 = 0; t2 = 0;
Z1P = 0.9421; % at 67 degC
ZG0 = 0.8519; ZL0 = 0.06; % at -11.1
v0 = 1; % at -11.5 degC
z0 = [0.06 0.4 0.4 0.14];
xin = [0.0035 0.0656 0.3985 0.5324];
yin = [0.06 0.4 0.4 0.14]; % at -11.1 degC, from dew point calculation
Tdew = -11.1+273.15; Tbub = -107.9+273.15; % at 1.8234 MPa

% load initial conditions for 1 phase (vapor) simulation
load w01P-10Q0 % 10 cells

nT0 = w1(end,1:M); % total holdup in (both phases) [kmol]
n0 = zeros(1,NC*M);
for j=1:M % component 1 holdup in tank (both phases) [kmol]
    n0(NC*j-3:NC*j) = nT0(j).*z0;
end

U0 = w1(end,M+NC*M+1:2*M+NC*M); % total internal energy (both phases), [MJ]
T0 = w1(end,2*M+NC*M+1:3*M+NC*M); % temperature (same in both phases), [K]

y0=zeros(1,NC*M);
for j=1:M
    y0(j*NC-(NC-1):j*NC) = z0(1:4);
end

Zg0 = ones(1,M)*Z1P; % gas compressibility
P00 = linspace(Pin,Pin-DP,M+2); % Pressure
P0 = P00(2:end-1);x0 = zeros(1,NC*M);

for j=1:M
    x0(j*NC-(NC-1):j*NC) = z0(1:4);
end

VL0 = w1(end,3*M+3*M*NC+1:4*M+3*M*NC);
Zl0 = ones(1,M)*0.06;
w0 = [nT0,n0,U0,T0,x0,y0,VL0,Zg0,Zl0,P0]; % vector of initial condition
%ODE solver options*****

```

C.3. MAIN SCRIPT ONE PHASE TO TWO PHASE SIMULATION APPENDIX C. MATLAB SCRIPTS

```

nrDyn = NC+2;           % number of dynamic equations per cell
nrAlg = 2*NC+5;        % number of algebraic eq per cell
d1 = ones(1,nrDyn*M);
d2 = zeros(1,nrAlg*M);
Mass = diag([d1,d2]);
options = odeset('mass',Mass,'Events',@efun);

%ODE solver
tstart = 0;
tfin = 200;
to=tstart;
wo=[];
teo=[];
weo=[];
ieo=[];

for i=1:2 % i = number of simulations
    % parameters passed to HEX.m
    param = [M,NC,V,F0,Pin,DP,Tin,Qmax,t1,t2,Z1P,ZG0,ZL0,v0,z0,xin, yin, Tdew, Ttub];
    % integrate
    [t,w,te,we,ie] = odel5s(@(t,w) HEX_T(t,w,param),[tstart tfin], w0, options);
    lt = length(t);

    %save output
    to = [to; t(2:lt)]; % integration time
    wo = [wo; w(2:lt,:)]; % output
    teo = [teo; te]; % time when event is triggered
    weo = [weo; we]; % values of the solutions when the event is triggered
    ieo = [ieo, ie]; % number of cell where the event is triggered

    % conditions for 2nd simulation
    tstart = te; % start a new simulation after the event is triggered
    LC = w(end,M:M:end); % results for cell # M
    A=ones(M,1);B=eye(4);C= repmat(B,1,M)';
    D=blkdiag(A,C,A,A,C,C,A,A,A); % selection matrix to pass conditions...
    %...from last cell of 1st simulation as initial conditions for all cells...
    % in 2nd simulation
    w0 = D*LC'; % new set of initial conditions
    Pin = LC(end); % new pressure...
    % ...to keep the linear pressure drop across HEX
    Tin=LC(7);
end

```

C.4 Heat exchanger model

```

function DAE = HEX_T(t,w,param)
% April 2017. All given data are passed as parameters from the main script.
% Compares the temperature in each cell with the dew point to select the
% valid set of equations

% I. Given data
M = param(1);           % Cells numbers
NC = param(2);         % 1-Nitrogen; 2-Methane; 3-Ethane; 4-Propane
V = param(3);          % Total volume of HEX

% Data common for both phases
F0 = param(4);         % Feed rate [kmol/min]
P0 = param(5);         % Inlet pressure, MPa
DP = param(6);         % Pressure drop along HEX, MPa
T0 = param(7);         % MR inlet temperature, K
z0 = param(15:15+NC-1); % Specified composition [kmol1/kmol]
Qmax = param(8);       % Given heat, [MJ/min]
t1 = param(9);         % time when the heat starts increasing, [min]
t2 = param(10);        % time when the heat stops increases, [min]
Tdew = param(end-1);   % Dew point, [K]
Tbub = param(end);     % Bubble point, [K]
Vcell = V/M;          % volume of each cell, [m3]

% Data for single phase (vapor or liquid)
Z1P = param(11);       % from FindZ.m
[~, h01P] = srkHEXV3(Z1P,z0,T0,P0); % Specific enthalpy for vapor[MJ/kmol]

% Data for vap-liquid phase
x0 = param(15+NC:15+2*NC-1); % liquid composition
y0 = param(15+2*NC:15+3*NC-1); % vapor composition
ZG0 = param(12);
ZL0 = param(13);
v0 = param(14);        % inlet vapor phase

[~, hfG] = srkHEXV3(ZG0,x0,T0,P0); % Specific enthalpy for vapor[MJ/kmol]
[~, hfL] = srkHEXV3(ZL0,y0,T0,P0); % Specific enthalpy for liquid[MJ/kmol]
h02P = v0*hfG + (1-v0)*hfL;      % Specific feed enthalpy [MJ/mol]

% Given heat as a function of time, [[MJ/min]
function Q = Q(t)
    if t <= t1

```

```

        Q = 0;
    elseif t > t1 && t < t2
        Q = Qmax/(t2-t1) * (t-t1);
    else
        Q = Qmax;
    end
end

Qcell = Q(t)/M; % equal heat removed from each cell

% II. States. Same for both regions.
N = w(1:M)'; % total holdup, [kmol]
n(1:NC*M) = w(M+1:M+M*NC)'; % component holdup, [kmol]
U = w(M+NC*M+1:2*M+NC*M)'; % total internal energy, [MJ]
T = w(2*M+NC*M+1:3*M+M*NC)'; % temperature, [K]
x(1:NC*M) = w(3*M+M*NC+1:3*M+2*M*NC)'; % liquid composition
y(1:NC*M) = w(3*M+2*M*NC+1:3*M+3*M*NC)'; % vapor composition
VL = w(3*M+3*M*NC+1:4*M+3*M*NC)'; % liquid volume, [m3]
Zg = w(4*M+3*M*NC+1:5*M+3*M*NC)'; % compressibility vapor [-]
Zl = w(5*M+3*M*NC+1:6*M+3*M*NC)'; % compressibility liquid [-]
P = w(6*M+3*M*NC+1:7*M+3*M*NC)'; % pressure, [MPa]

% III. System of equations

% IIIa. The outlet flow is calculated with a linear valve equation
F = zeros(1,M); % [kmol/min], preallocating memory
k = F0/DP*(M+2); % [kmol/min*MPa], valve constant...
% considering equal pressure drop in each cell
Pout = P0-DP; % [MPa], outlet pressure

for j=1:M-1
    F(j) = k*(P(j)-P(j+1)); % Outlet flow
end

for j=M
    F(j) = k*(P(j)-Pout);
end

% IIIb. SRK. Compute fugacity, enthalpy, molar volumes
% preallocating memory
phil = zeros(M,NC); phig = zeros(M,NC); hl = zeros(1,M); hg = zeros(1,M);
Vlm = zeros(1,M); Vgm = zeros(1,M); Al = zeros(1,M); Ag = zeros(1,M);
Bl = zeros(1,M); Bg = zeros(1,M);

for j=1:M

```



```

[phil_temp,h1_temp,Vlm_temp,Al_temp,B1_temp]= srkHEXV3(Z1(j),...
    [x(j*NC-3) x(j*NC-2) x(j*NC-1) x(j*NC)], T(j),P(j)); % liquid
[phig_temp,hg_temp,Vgm_temp,Ag_temp,Bg_temp]= srkHEXV3(Zg(j),...
    [y(j*NC-3) y(j*NC-2) y(j*NC-1) y(j*NC)], T(j), P(j)); % vapor

phil(j,1)= phil_temp(1); phil(j,2)= phil_temp(2);
phil(j,3)= phil_temp(3); phil(j,4)= phil_temp(4);
phig(j,1)= phig_temp(1); phig(j,2)= phig_temp(2);
phig(j,3)= phig_temp(3); phig(j,4)= phig_temp(4);
hl(j)= h1_temp; Vlm(j)= Vlm_temp; Al(j)= Al_temp; B1(j)= B1_temp;
hg(j)= hg_temp; Vgm(j)= Vgm_temp; Ag(j)= Ag_temp; Bg(j)= Bg_temp;

end

K = phil./phig; % VLE constant

% Phase distribution
VG = zeros(1,M); nG = zeros(1,M);v = zeros(1,M); nL = zeros(1,M);
f1 = zeros(1,M); f2 = zeros(1,M*(NC)); f3 = zeros(1,M); f4 = zeros(1,M);
f5 = zeros(1,M*NC); f6 = zeros(1,M*NC); f7 = zeros(1,M); f8 = zeros(1,M);
f9 = zeros(1,M); f10 = zeros(1,M); % preallocation memory

% Checking phase location, e.g. 1 phase (vapor or liquid)
% or 2 phase (vapor-liquid)

for j=1 % cell # 1.

% Compare the temperature in each cell with the dew point
    if T(j)-Tdew>1e-5 % [K]
        VG(j) = Vcell; % gas volume only for vapor phase, [m3]
        nG(j) = VG(j)/Vgm(j); % gas holdup, [kmol]

% IIIc. System of equations for single phase
        f1(j) = F0-F(j);
        f2((NC)*j-3:(NC)*j) = F0*z0 - F(j)*y((NC)*j-3:(NC)*j);
        f3(j) = F0*h01P - F(j)*hg(j)- Qcell;
        f4(j) = U(j) + P(j)*Vcell - hg(j)*nG(j);
        f5((NC)*j-3:(NC)*j) = y((NC)*j-3:(NC)*j) - n((NC)*j-3:(NC)*j) ./N(j);
        f6((NC)*j-3:(NC)*j) = x((NC)*j-3:(NC)*j) - x0(1:4);
        f7(j) = VL(j);
        f8(j) = Zg(j)^3-Zg(j)^2+Zg(j)*(Ag(j)-Bg(j)-Bg(j)^2)-Ag(j)*Bg(j);
        f9(j) = Z1(j) - ZL0;
        f10(j) = N(j)-nG(j);
    end
end

```

```

elseif T(j)-Tbub>1e-5 % system is in 2 phase region
% phase distribution for 2-phase region
    VG(j) = Vcell - VL(j);           % gas volume for 2 phase region
    nL(j) = VL(j)/Vlm(j);           % liquid holdup, [kmol]
    nG(j) = VG(j)/Vgm(j);           % gas holdup, [kmol]
    v(j) = nG(j)/N(j);              % vapor fraction

% IIIId. System of equations for 2 phase region

K1=K(j,:); % K-values

f1(j) = F0-F(j);
f2(NC*j-3:NC*j) = F0*v0*y0+ F0*(1-v0)*x0 - F(j)*v(j)*y(NC*j-3:NC*j)-...
    F(j)*(1-v(j))*x(NC*j-3:NC*j);
f3(j) = F0*h02P - F(j)*v(j)*hg(j) - F(j)*(1-v(j))*hl(j) - Qcell;
f4(j) = U(j) + P(j)*Vcell - hl(j).*nL(j) - hg(j).*nG(j);
f5(NC*j-3:NC*j) = n(NC*j-3:NC*j) - x(NC*j-3:NC*j).*nL(j) -...
    y(NC*j-3:NC*j).*nG(j);
f6(NC*j-3:NC*j) = y(NC*j-3:NC*j) - K1(1:NC).*x(NC*j-3:NC*j);
f7(j) = N(j)-nG(j)-nL(j);
f8(j) = (Zg(j)^3-Zg(j)^2+Zg(j)*(Ag(j)-Bg(j)-Bg(j)^2)-Ag(j)*Bg(j));
f9(j) = (Zl(j)^3-Zl(j)^2+Zl(j)*(Al(j)-Bl(j)-Bl(j)^2)-Al(j)*Bl(j));
f10(j) = sum(x(j*NC-3:j*NC)) - sum(y(j*NC-3:j*NC));

else %system is in liquid phase
    VL(j) = Vcell;                   % liquid volume [m3]
    nL(j) = VL(j)/Vgm(j);           % liquid holdup [kmol]

f1(j) = F0-F(j);
f2((NC)*j-3:(NC)*j) = F0*z0 - F(j)*x(NC*j-3:NC*j);
f3(j) = F0*h01P - F(j)*hl(j)- Qcell;
f4(j) = U(j) + P(j)*Vcell - hl(j).*nL(j);
f5((NC)*j-3:(NC)*j) = x(NC*j-3:NC*j) - n(NC*j-3:NC*j)./N(j);
f6((NC)*j-3:(NC)*j) = y(NC*j-3:NC*j) - y0(1:4);
f7(j) = VG(j);
f9(j) = (Zl(j)^3-Zl(j)^2+Zl(j)*(Al(j)-Bl(j)-Bl(j)^2)-Al(j)*Bl(j));
f9(j) = Zg(j) - ZG0;
f10(j) = N(j)-nL(j);
end % if statement

end % for loop

for j=2:M % Cells 2 to N

% Compare the temperature in each cell with the dew point

```

```

if T(j)-Tdew>1e-5 % [K]
    VG(j) = Vcell; % gas occupies entire cell volume [m3]
    nG(j) = VG(j)/Vgm(j); % gas holdup [kmol]

f1(j) = F(j-1)-F(j);
% overall mass balance
f2((NC)*j-3:(NC)*j) = F(j-1)*y(NC*j-7:NC*j-NC) - F(j)*y(NC*j-3:NC*j);
% component mass balance
f3(j) = F(j-1)*hg(j-1) - F(j)*hg(j) - Qcell;
% energy balance
f4(j) = U(j) + P(j)*Vcell - hg(j)*nG(j);
% internal energy
f5((NC)*j-3:(NC)*j) = y(NC*j-3:NC*j) - n(NC*j-3:NC*j)/N(j);
% vapor composition
f6((NC)*j-3:(NC)*j) = x(NC*j-3:NC*j) - x0(1:4);
% "liquid composition", dummy eq
f7(j) = VL(j);
% "liquid volume", dummy eq
f8(j) = Zg(j)^3-Zg(j)^2+Zg(j)*(Ag(j)-Bg(j)-Bg(j)^2)-Ag(j)*Bg(j);
% gas compressibility
f9(j) = Zl(j) - ZL0;
% "liquid compressibility", dummy eq
f10(j) = N(j)-nG(j);
% total holdup

elseif Tdew > T(j)>Tsub % system is in 2 phase region
% phase distribution for 2-phase region
    VG(j) = Vcell - VL(j); % gas volume for 2 phase region
    nL(j) = VL(j)/Vlm(j); % liquid holdup, [kmol]
    nG(j) = VG(j)/Vgm(j); % gas holdup, [kmol]
    v(j) = nG(j)/N(j); % vapor fraction

% IIIId. System of equations for 2 phase region

K1=K(j,:); % K-values for all components

f1(j) = F(j-1)-F(j);
f2(NC*j-3:NC*j) = F(j-1)*v(j-1)*y(NC*j-7:NC*j-NC)+F(j-1)*...
    (1-v(j-1))*x(NC*j-7:NC*j-NC) - F(j) *v(j)*...
    y(NC*j-3:NC*j) -F(j) *(1-v(j)) *x(NC*j-3:NC*j);
f3(j) = F(j-1)*(1-v(j-1))*hl(j-1) + F(j-1)*v(j-1)*hg(j-1) -...
    F(j)*(1-v(j))*hl(j) - F(j)*v(j)*hg(j) - Qcell;
f4(j) = (U(j) + P(j)*Vcell- hl(j)*nL(j) - hg(j)*nG(j));
f5(NC*j-3:NC*j) = n(NC*j-3:NC*j) - x(NC*j-3:NC*j).*nL(j) -...

```

```

        y(NC*j-3:NC*j) .* nG(j);
f6(NC*j-3:NC*j) = y(NC*j-3:NC*j) - K1(1:NC) .* x(NC*j-3:NC*j);
f7(j) = N(j) - nG(j) - nL(j);
f8(j) = (Zg(j)^3 - Zg(j)^2 + Zg(j) * (Ag(j) - Bg(j) - Bg(j)^2) - Ag(j) * Bg(j));
f9(j) = (Zl(j)^3 - Zl(j)^2 + Zl(j) * (Al(j) - Bl(j) - Bl(j)^2) - Al(j) * Bl(j));
f10(j) = sum(x(j*NC-3:j*NC)) - sum(y(j*NC-3:j*NC));

else % system is in liquid phase
VL(j) = Vcell; % liquid volume [m3]
nL(j) = VL(j)/Vgm(j); % liquid holdup [kmol]

f1(j) = F(j-1) - F(j);
f2((NC)*j-3:(NC)*j) = F(j-1) * x(NC*j-7:NC*j-NC) - F(j) * x(NC*j-3:NC*j);
f3(j) = F(j-1) * hl(j-1) - F(j) * hl(j) - Qcell;
f4(j) = U(j) + P(j) * Vcell - hl(j) * nL(j);
f5((NC)*j-3:(NC)*j) = x(NC*j-3:NC*j) - n(NC*j-3:NC*j) ./ N(j);
f6((NC)*j-3:(NC)*j) = y(NC*j-3:NC*j) - y0(1:4);
f7(j) = VG(j);
f8(j) = Zg(j)^3 - Zg(j)^2 + Zg(j) * (Ag(j) - Bg(j) - Bg(j)^2) - Ag(j) * Bg(j);
f9(j) = Zg(j) - ZG0;
f10(j) = N(j) - nL(j);

end % if statement

end % for statementant

DAE = [f1'; f2'; f3'; f4'; f5'; f6'; f7'; f8'; f9'; f10']; % System of DAE

end % end function structure

```

C.5 Event function

```

function [value, isterminal, direction] = efun(t,w)
% event function. stop integrating when T-Tdew>tol.

NC = 4;
N = 10;
T = w(2*N+NC*N+1:3*N+N*NC)';
Tdew = -11.1+273.15;
tol = 1e-3;
value = double((T-Tdew < tol));
% checks that at least one temperature is bellow the dew point

```

```

isterminal = ones(1,N);
% stop integration
direction = [];
% all zeros are to be located (the default)
end

```

C.6 Soave modification of Redlich-Kwong

```

function [phi,H,Vm,A,B]= srkHEXV3(Z,x,T,P)
% In: compositon x (vector), Temperature T [K], pressure p [MPa]
% OUT: Compressibility Z, Fugacity coeff.(vector) phi [-],
%enthalpy H [MJ/kmol], Molar Volume V [m3/kmol]

% Authors: Andreas Linhart and Sigurd Skogestad. NTNU. 2008
% CORRECTED phi to include kij correctly
%(authors: Marie Solvik and Sigurd Skogestad, NTNU, Nov. 2013)
% Adapted to a mixed refirgerent stream (Cristina Zotica, Feb 2017)
% removed 'liquid', 'vapor' as global variables
% added 5th term of Cp in % calculating H;
% changed P in MPa and H in MJ/kmol

%Components N2 C1 C2 C3 C4
NC=4; % Nr of components (1)Nitrogen - (2)Methane - (3)Ethane - (4)Propane

%Component data from: Reid RC, Prausnitz JM, Poling BE.
%The properties of gases & liquids (5th edition)

compData.Pc = [33.98 45.99 48.72 42.48 ]*0.1; % [MPa]
compData.Tc = [126.20 190.56 305.32 369.83]; % [K]
compData.w = [0.037 0.011 0.099 0.152 ]; % acentric factor [-]

R = 8.314/1000; %MJ/kmol/K

% Ideal has heat capacity (See Hid below):
compData.Cp(:, 1) = [3.539 -0.261e-3 0.007e-5 0.157e-8 -2.099e-11]' * R;
compData.Cp(:, 2) = [4.568 -8.975e-3 3.631e-5 -3.407e-8 1.091e-11]' * R;
compData.Cp(:, 3) = [4.178 -4.427e-3 5.660e-5 -6.651e-8 2.487e-11]' * R;
compData.Cp(:, 4) = [3.847 5.131e-3 6.011e-5 -7.893e-8 3.079e-11]' * R;

compData.Tref = 298.15; % [K], for Ideal gas heat capacity
compData.Pref = 0.1; % [MPa]

```

C.6. SOAVE MODIFICATION OF REDLICH-KWONG APPENDIX C. MATLAB SCRIPTS

```
% SRK binary interaction parameters from Aspen Hysys V9
kinteraction(:,1) = [0 3.11990007758141e-002 3.18990014493465e-002...
                    8.86000022292137e-002]; %Nitrogen
kinteraction(:,2) = [ 3.11990007758141e-002 0 2.24137306213379e-003...
                    6.82878494262695e-003]; %Methane
kinteraction(:,3) = [3.18990014493465e-002 2.24137306213379e-003 0 ...
                    1.25795602798462e-003]; %Ethane
kinteraction(:,4) = [8.86000022292137e-002 6.82878494262695e-003...
                    1.25795602798462e-003 0]; %Propane

%kinteraction = zeros(4,4);

%% Initialize

Pc = compData.Pc;
Tc = compData.Tc;
w = compData.w;
ZRA = 0.29056-0.08775*w;
Cp = compData.Cp;
Tref = compData.Tref;
Pref = compData.Pref;

% Calculations for given T, P and composition (x)
Tre = T./Tc;
Pre = P./Pc;
m = 0.480+1.574.*w-0.176.*w.^2;
a = (1+m.*(1-Tre.^0.5)).^2;
Ap = 0.42747.*a.*Pre./Tre.^2;
Bp = 0.08664.*Pre./Tre;

% Start calculations

% Binary a's:
%Mixing rules
Ab = zeros(NC,NC);
for i = 1:NC
    for j = 1:NC
        Ab(i,j) = (Ap(i)*Ap(j))^0.5;
    end
end

% Mixture a and b
A = 0;

for i = 1:NC
    for j = 1:NC
```

APPENDIX C. MATLAB SCRIPTS C.6. SOAVE MODIFICATION OF REDLICH-KWONG

```

        A = A+x(i)*x(j)*Ab(i,j)*(1-kinteraction(i,j));
    end
end

B = 0;
for i = 1:NC
    B=B+x(i)*Bp(i);
end

%phi
corrphi=zeros(1,NC);
for i = 1:NC
    for j =1:NC
        corrphi(i) = corrphi(i) + x(j) * Ap(j)^0.5 * (1-kinteraction(i,j));
    end
end

phi = exp((Z-1).*Bp/B-log(Z-B)-A/B*log((Z+B)/Z).*...
    ((corrphi.*2.*Ap.^0.5./A)-Bp./B));

%end

% Enthalpy
dadT = 0;
for i = 1:NC
    for j = 1:NC
        dadT = dadT -R / 2 * sqrt(abs(0.42748 / T)) * x(i) * x(j) *...
            (m(j) * sqrt(abs(Ap(i)*(Tc(j)/P*(T^2)/(Pc(j))*(R^2)))) +...
            m(i) * sqrt(abs(Ap(j) * (Tc(i)/P*(T^2)/(Pc(i))*(R^2)))));
    end
end

H = (1/ (B*R*T/P)) * ( A*R^2*T^2/P - T*dadT) * log(Z / (Z + B)) +...
    R * T * (Z - 1) ;

for i = 1:NC
    H = H+(x(i)* (Cp(1,i) * (T - Tref) + 1/2 * Cp(2,i) *(T^2 - Tref^2)+...
        1/3 * Cp(3,i) * (T^3 - Tref^3) +...
        1/4 * Cp(4,i) * (T^4 - Tref^4) + 1/5 * Cp(5,i) * (T^5 - Tref^5)));
end

% Density (more precisely: molar volume)

```

C.6. SOAVE MODIFICATION OF REDLICH-KWONG APPENDIX C. MATLAB SCRIPTS

```
if Z<0.1 % Correct liquid SRK-volume using Peneleoux correction
    c=0;
    for i=1:NC
        c=c+x(i) * (0.40768 * (0.29441 - ZRA(i)) * (R * Tc(i)) / (Pc(i))) ;
    end
    Vm = ((Z * R * T / P) - c);
else %vapor
    Vm = Z * R * T / P;
end
```


Appendix D

Test of ode15s

D.1 Example from section 5.1.2

```
clear;clc;close all

y0 = 0.1; % initial condition
tspan = [0 10]; % time span
options = odeset('MaxStep',0.01);
[t,y] = ode15s(@fun,tspan,y0,options);

figure
plot(t,y,'Linewidth',1)
xlabel('time')
ylabel('y')
set(gca,'fontsize',20)
grid on

function dydt = fun(t,y) % non-smooth point
if y<1
    dydt = y;
else
    dydt = -y+2;
end
end
```

D.2 Example from section 5.1.3

D.2.1 Main script

```

clc;clear;close all
% Model.
% dVdt = qin-c*V-s*Tg-(1-s)*Tl;
% dQdT = 1/Q;
% Tg^3 + Q*Tg^2+Tg-1.2 = 0; s*(Tl^3 + Q*Tl^2+Tl-1.2);
%(this can be written without s*, but the result is the same) for example 11.
% T^3 + Q*T^2+T-1.2 = 0; for example 12.

%% Example 11. Solves the ODE system using ode15s
% the algebraic are solved internally with root function
clear
clc
y0 = [1 1.5];
tspan = [0 10];
options = odeset('Stats','on');
[t,y,stats] = ode15s(@odefun11,tspan,y0,options);
figure
plot(t,y,'Linewidth',1)
legend('x','y','Location','northwest')
%axis([0 5 0 3.5])
xlabel('Time','FontSize',20,'Fontname','Timesnewroman')
ylabel('Differential variables','FontSize',20,'Fontname','Timesnewroman')
set(gca,'fontsize',20)
grid on

%% Example 12. Solves the DAE system using ode15s
% at t=0, uses an imaginary root for Tl, since ode15s computes only one
% root which is closest to the initial condition
% This leads to imaginary values of V however.

%y0 = [1 1.5 0.5583 -1.0292+1.044i];
y0 = [1 1.5 0.5583 0.5583];

tspan = [0 1000];
M = [1 0 0 0; 0 1 0 0; 0 0 0 0; 0 0 0 0];
options=odeset('Mass',M);
[t,y] = ode15s(@odefun12,tspan,y0,options);

```

```

[t,y] = ode15s(@odefun13,tspan,y0,options);
figure
plot(t,y(:,1:2), 'Linewidth',1)
legend('x','y','Location','northwest')
%axis([0 5 0 3.5])
xlabel('Time','FontSize',20,'Fontname','Timesnewroman')
ylabel('Differential variables','FontSize',20,'Fontname','Timesnewroman')
set(gca,'fontsize',20)
grid on

```

D.2.2 DAE system solved by ode15s in one layer approach from section 5.1.3.1

```

function dydt = f(t,w)
x = w(1);
y = w(2);
z1 = w(3);
z2 = w(4);

qin=1.5;c=1;

if isreal(z2)==1 && isreal(z1)==1
    s=0; % 3 real roots of f3
else
    s=1; % 1 real root
end
% if Q<2.6919
%     s=0;
% else
%     s=1;
% end
f1 = qin -c*x-s*z1-(1-s)*z2; % differential eq
f2 = 1/y; % differential eq
f3 = (z1^3 + y*z1^2+z1-1.2); % alg eq
f4 = (z2^3 + y*z2^2+z2-1.2); % alg eq, can also be written with s*

dydt = [f1;f2;f3;f4];
end

```

Table D.1: Results for DAE example from section 5.1.3.2, initialized with complex values for z_2

x	y	z_1	z_2
1 + 0i	1.5 + 0i	0.55834 + 0i	-1.0292 + 1.0441i
0.99935 + 0i	1.5075 + 0i	0.55769 + 0i	-1.0326 + 1.0419i
0.99871 + 0i	1.515 + 0i	0.55704 + 0i	-1.036 + 1.0396i
0.99809 + 0i	1.5225 + 0i	0.5564 + 0i	-1.0395 + 1.0374i
0.99257 + 0i	1.5954 + 0i	0.55028 + 0i	-1.0728 + 1.0148i
0.98826 + 0i	1.6651 + 0i	0.5446 + 0i	-1.1049 + 0.99132i
0.98499 + 0i	1.7321 + 0i	0.53932 + 0i	-1.1357 + 0.96708i
0.98263 + 0i	1.7965 + 0i	0.53438 + 0i	-1.1655 + 0.94201i
0.97996 + 0i	2.014 + 0i	0.51867 + 0i	-1.2663 + 0.84257i
0.9831 + 0i	2.21 + 0i	0.50564 + 0i	-1.3577 + 0.72745i
0.98913 + 0i	2.3903 + 0i	0.4945 + 0i	-1.4415 + 0.5883i
0.99517 - 1.5081e-21i	2.5308 - 3.9971e-19i	0.48634 + 1.0164e-20i	-1.5082 + 0.43816i
0.99709 - 3.176e-21i	2.5715 - 4.9484e-19i	0.48405 + 2.7303e-20i	-1.5278 + 0.38076i
0.99906 - 5.563e-21i	2.6115 - 5.3019e-19i	0.48183 + 2.9817e-20i	-1.5468 + 0.31326i
1.001 - 7.8641e-21i	2.651 - 5.5311e-19i	0.47968 + 2.624e-20i	-1.565 + 0.22757i
1.0016 - 8.5096e-21i	2.6627 - 5.6405e-19i	0.47904 + 3.0654e-20i	-1.5708 + 0.19353i
1.0023 - 9.1486e-21i	2.6743 - 5.2548e-19i	0.47841 + 2.8954e-20i	-1.5766 + 0.15255i
1.0029 - 9.7249e-21i	2.6859 - 4.8563e-19i	0.47779 + 2.6453e-20i	-1.5817 + 0.097174i
1.003 - 9.8797e-21i	2.6894 - 4.7957e-19i	0.4776 + 2.6122e-20i	-1.5835 + 0.071603i
1.0032 - 1.0028e-20i	2.6929 - 4.7739e-19i	0.47741 + 2.5586e-20i	-1.5852 + 0.030131i
1.0032 - 1.0041e-20i	2.6932 - 4.7758e-19i	0.4774 + 2.558e-20i	-1.5853 + 0.022051i
1.0032 - 1.0054e-20i	2.6935 - 4.7762e-19i	0.47738 + 2.5569e-20i	-1.5854 + 0.0093336i
1.0032 - 1.0055e-20i	2.6935 - 4.776e-19i	0.47738 + 2.5578e-20i	-1.5855 + 0.0079603i
1.0032 - 1.0056e-20i	2.6936 - 4.7759e-19i	0.47738 + 2.5575e-20i	-1.5855 + 0.0055753i
1.0032 - 1.0057e-20i	2.6936 - 4.7759e-19i	0.47738 + 2.5577e-20i	-1.5855 + 0.0043568i
1.0032 - 1.0057e-20i	2.6936 - 4.7758e-19i	0.47738 + 2.5577e-20i	-1.5855 + 0.0027801i
1.0033 - 1.0057e-20i	2.6936 - 4.7756e-19i	0.47738 + 2.5575e-20i	-1.5855 + 0.00061941i

D.2.3 DAE system solved in two layer approach from section 5.1.3.2

```
function dydt = f(t,w)
x = w(1);
y = w(2);

qin=1.5;c=1;

p = [1 y 1 -1.2];          % z^3 + y*z^2+z-1.2 % alg eq
```

```
R = roots(p)

if isreal(R)==0 % checks the number of real roots of polynomial p
    z1 = R(3); % last root is always real
    z2 = R(1); % first root is complex
    s = 1;
    f1 = qin-c*x-s*z1+(1-s)*z2; % differential eq
    f2 = 1/y; % differential eq

else % all roots are real
    z1 = max(R); % chose maximum root
    z2 = min(R); % chose minimum root
    s=0;
    f1 = qin-c*x-s*z1+s*z2; % differential eq
    f2 = 1/y; % differential eq

end

dydt = [f1;f2];
end
```