NTNU
Norwegian University of
Science and Technology

Dynamic model for two phase heat exchanger

Oliver Sale Haugberg

December 19, 2017

TKP4580 - Chemical Engineering, Specialization Project
Department of chemical engineering
Norwegian University of Science and Technology

Supervisor 1: Johannes Jäschke
Supervisor 2: Adriaen Verheyleweghen

Abstract

As part of a greater goal of developing a dynamic model of for a LNG liquefaction process, a dynamic heat exchanger model was formulated. The model was formulated with the intention of handling three different possible phase regions that can occur in the refrigeration cycle evaporators and condenser. These three regions being vapour phase, liquid phase and both.

The model was formulated as a series of nodes with systems of differential-algebraic equations solved for each node. Model simulations were done for cases where the streams were only in a single phase region. Although consistent initial conditions proved difficult to find for the two phase region, once found the model is robust for changes to input changes over time. Capabilities for handling phase region transitions were not included, as the model was formulated with a focus on the two phase region, which lead to difficulties for the implementation of these capabilities.

Table of Contents

Abstract **i**

Table of Contents **iii**

List of Tables **iv**

List of Figures **v**

Nomenclature **v**

1 Introduction **2**

2 Thermodynamics **4**

2.1 Equation of state 4

2.2 Antoine equation 5

2.3 Heat capacities and Enthalpy 5

3 Modelling **7**

3.1 Model requirements 8

3.2 First approach 8

3.3 Second approach 9

3.3.1 Equations for two phase region 9

3.3.2 Equations for one phase region 11

4 Solving the DAE systems **13**

4.1 Index of DAE systems 13

4.2 Model stiffness 13

4.3 Solver 14

4.4 Initial conditions 14

5 Model simulations **15**

5.1 Two phase region 15

5.1.1 Steady state 15

5.1.2 Dynamic response 17

5.2 One phase region 20

5.3 Stiffness and time scale differences 21

6 Discussion **23**

7 Conclusion 25

Bibliography 25

Appendix 27

Parameters and constants 27

Matlab code 28

List of Tables

5.1 Stream conditions used for two phase region simulations 15
5.2 Stream conditions used for single phase region simulations 20
5.3 Stream conditions used for time scale simulations 21

7.1 Critical temperature T_c, pressure P_c and acentric factor w used for Peng-Robinson equation of state 27
7.2 Constants for the Antonie equation 27
7.3 Coefficients used for calculating heat capacity of gas phases 27
7.4 Values used for liquid phase heat capacity and enthalpy of vaporization 27

List of Figures

1.1 Refrigeration cycle and Pressure enthalpy diagram 3

3.1 Lumped model for one side of the heat exchanger 7

5.1 Pressure and molar flow through the heat exchanger at steady state. 16
5.2 Liquid fraction at steady state 16
5.3 Liquid and vapour phase holdup at steady state 17
5.4 Vapour phase volume fraction at steady state 17
5.5 Molar flow rate and node pressure after step change of inlet flow rate 18
5.6 Liquid fraction and vapour holdup after step change of inlet flow rate 19
5.7 Liquid fraction after step change of inlet liquid fraction 19
5.8 Molar flow rate and pressure for step change of the inlet liquid fraction 20
5.9 Temperature change for step changes of inlet temperature and inlet molar flow rate . . . 21
5.10 Molar flow rate and liquid fraction after a step change in the inlet liquid fraction 22

Nomenclature

Symbol	**Explanation**
c_x	Coefficient x for gas phase heat capacity
A’	Constant in the Antoine equation
B’	Constant in the Antoine equation
C$_p$	Heat capacity at constant pressure
C’	Constant in the Antoine equation
e	Mathematical constant
H	Enthalpy
h	Specific enthalpy
H$_f$	Heavyside function
k	Flow equation constant
M	Mass matrix
m	Molar flow
N	Number of nodes
n	Holdup
P	Pressure
Q	Heat transfer
R	Gas constant
T	Temperature
t	Time
U	Internal energy
UA	Overall heat transfer coefficient for area
V	Volume

V_m	Molar volume
w	Acentric factor
x	Liquid fraction
Z	Compressibility

Subscripts

c	Critical value
g	Gas phase
i	For node i
in	For inlet stream
l	Liquid phase
ref	Reference value
Vap	Vaporization value

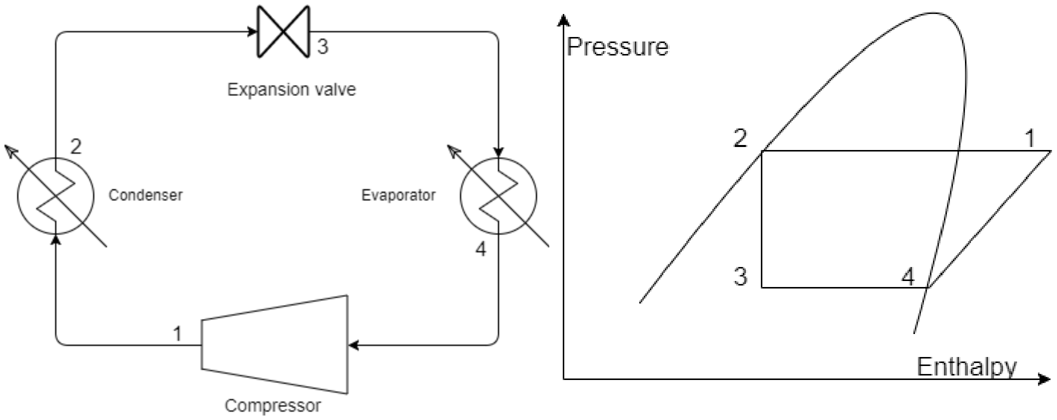
Chapter 1

Introduction

LNG liquefaction processes are a series of processes that use refrigeration cycles for the cooling and liquefaction of natural gas. One of the main reasons this is done is because transporting a cooled, non-pressurized liquid is easier and safer than a highly compressed gas, especially over large distances. There are a few different LNG liquefaction processes in commercial use today, with different designs choices and different refrigerants used. Each method done for optimizing operations for different conditions or plant scales. This project was mainly based on Verheyleweghen and Jäschke [2016], where a self optimizing control structure was applied to a cascade LNG liquefaction plant. In self optimizing control, one of the first steps is to find and define what the optimal operation conditions are for the process. In order to do this in Verheyleweghen and Jäschke [2016], a steady state model for the the cascade process was formulated and solved as a nonlinear optimization problem. With the goal being to minimize the energy usage in the refrigeration cycle whilst subject to the constraints in the system.

The original intent of this project was to further develop this steady state model into a dynamic model. This dynamic version of the process could be used for either dynamic optimization or implementation / development of control structures. After studying the process the objective was shifted to solely focusing on the development of a dynamic heat exchangers model, as this turns out to be quite challenging in on itself. In the steady state LNG cascade process model there are three refrigeration cycles. Each using the different refrigerants methane, ethane and propane respectively. In this project the requirements needed for modeling the condensers and evaporator in these refrigerant cycles will be explored. Heat exchangers used for heat transfer with the LNG stream will not be explored as the multicomponent LNG stream behaves differently.

Each of the cooling refrigerants go through a refrigeration cycle. In these cycles a refrigerant is continuously cycled from a condenser section to a evaporater section in order to move energy from a heat source to a heat sink. In order to accurately describe the the dynamics present in these cycles, robust models with a wide range of requirements are needed for both the evaporators and condensers. In the cycle the refrigerant in vapour phase first enters the compressor. The then high pressure vapour, point 1 in figure 1.1, is condensed in a condenser section. Then the cooled and high pressure fluid is sent through a Joule-Thompson valve, where an isenthalpic process causes the pressure, and temperature with it, to fall significantly, to point 3. The refrigerant is then sent to a evaporation section where it can either cool other refrigerant streams or the main LNG stream. After being fully evaporated it is sent back to the compressor.



(a) Refrigeration cycle

(b) Pressure enthalpy diagram

Figure 1.1: Refrigeration cycle and Pressure enthalpy diagram

The objective of this project is then to develop a dynamic heat exchanger model can handle streams consisting of pure vapour, pure liquid or both phases at the same time. A robust and proper working model of this kind could then be used to describe the dynamics present in the evaporator and condenser sections of these refrigeration cycles. The main source of complications in this problem is the requirement that the model be able to handle all three possible phase regions in the same heat exchanger, without explicitly stating where these regions occur.

Chapter 2

Thermodynamics

2.1 Equation of state

To relate the state variables in the heat exchanger an equation of state is needed. As the refrigerants continuously change phase in the evaporators and condensers, the main phase region occupied by the fluid in the process, is when both vapour and liquid is present. This phase region being the area inside the curve shown in Figure 1.1. It is therefore important that the equation of state accurately describes the fluid in this two phase are. The Peng-Robinson equation of state is therefore chosen as it accurately describes the properties of both the liquid and vapour phases. It was also developed with natural gas processes in mind. Peng and B. Robinson [1976]

The equation parameters a, b and α are determined by the substance critical condition values for temperature and pressure, T_c, P_c and the acentric factor ω with values given in Table 7.1.

$$P = \frac{RT}{V_m - b} - \frac{a\alpha}{V_m^2 + 2bV_m - b^2} \tag{2.1a}$$

$$a = \frac{0.45724R^2T_c^2}{P_c} \tag{2.1b}$$

$$b = \frac{0.007780RT_c}{P_c} \tag{2.1c}$$

$$\alpha = (1 + (0.37464 + 154226w - 0.26992w^2)(1 - (\frac{T}{T_c})^{0.5}))^2 \tag{2.1d}$$

This equation is then rewritten into polynomial where it can be solved for the compressibility factor Z instead.

$$0 = Z^3 - Z^2(1 - B) + Z(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{2.2a}$$

$$A = \frac{a\alpha P}{R^2T^2} \tag{2.2b}$$

$$B = \frac{bP}{RT} \tag{2.2c}$$

When solving the polynomial for Z the three roots have different physical meanings depending on the state. For the pure liquid or gas phases there will be a complex conjugate solution and one real valued solution. The real solution being the compressibility factor for that given phase. For systems in the two phase region there are thee real roots. The highest valued one being the solution for the gas phase, and the lowest one for the liquid phase, and the middle solution with no physical meaning. One issue that can then occur, is if a solver chooses the wrong root as a solution. Generally the solutions for the liquid phase and the gas phase are quite far apart. But as the middle solution has no physical meaning, a solver could theoretically ”jump” from one root to the other. One way of dealing with the multiple solutions is to use logical statements, but the problem can also be formulate in an equation oriented approach. In Kamath et al. [2010] an explanation of how incorporating constraints on the derivative can isolate the correct solutions. For the first derivative, both the liquid and gas phase roots are positive. $f'(r_V) \geq 0$ and $f'(r_L) \geq 0$ which isolates them from the middle solution. For the second derivative $f''(r_V) \geq 0$ and $f''(r_L) \leq 0$ which isolates them from each other. These properties could be implemented as constraints if necessary.

The compressibilities found can then be used to calculate the molar volumes V_m of both phases, as a correctional term in the ideal gas equation.

$$Z = \frac{PV}{nRT} \tag{2.3a}$$

$$\downarrow \tag{2.3b}$$

$$V_m = \frac{RTZ}{P} \tag{2.3c}$$

2.2 Antoine equation

At the phase equilibrium the pressure and temperature are related by the Antoine equation. The equation is empirical and derived from the Clausius-Clapeyron relation.

$$\log_{10}(p) = A' - \frac{B'}{C' + T} \tag{2.4}$$

The constants A’, B’ and C’ are component-specific , and are valid for a certain temperature range as the equation is not flexible enough for the entire equilibrium curve. The constants used are found in Table 7.2.

2.3 Heat capacities and Enthalpy

How the refrigerants react to the heat transferred in the heat exchanger, depends on the specific heat capacities and enthalpies of the streams phases. For a real gas enthalpy is generally estimated as a sum of ideal enthalpy plus a departure term depending on the equation of state.

$$h_i = h_i^{ideal} + h_i^{departure} \tag{2.5}$$

The Peng-Robinson equation of state does have a departure function for enthalpy, but it has not been included in this project leaving only the ideal term.

$$h_i^{ideal} = \int_{T_{ref}}^{T} C_p(T)dT \tag{2.6}$$

For the gas phases, the heat capacities $c_{p,g}$ are calculated as a function of temperature

$$c_{p,g} = c_1 + c_2T + c_3T^2 + c_4T^3 \tag{2.7}$$

The coefficients used for the different refrigerants are found in Table 7.3. For the liquid phase the heat capacities are assumed to be constant . Using a reference temperature bellow the boiling point will then result in the equations.

$$h_{l,i}(T) = c_{p,l}(T - T_{ref}) \tag{2.8a}$$

$$h_{g,i}(T) = h_{l,i}(T) + h_{vap} + \int_{T_{Vap}}^{T} c_{p,g} dT \tag{2.8b}$$

The enthalpy of evaporation h_{vap}, the heat required for phase change, is also assumed to be constant. The liquid phase heat capacities and the enthalpy of evaporation values used are found in Table 7.4.

Chapter 3

Modelling

To maximize the efficiency of the heat transferred during liquefaction, large and complicated purpose built heat exchanger are used. In cascade processes they are normally multi-stream using either spiral wound, or plate fin designs. These type of designs are used to minimize the mean temperature differences throughout the process which will maximize efficiency.

Because the heat exchanger geometry adds a new layer of complexity to the problem, the heat exchangers are instead modelled as a series of counter current heat exchangers with one cold and one hot side. This simplifies the aspect of how heat is transferred in the system. Then the behaviour inside each stream, how it is affected by the heat transfer, can instead be the focus. The streams in the heat exchangers are inherently distributed parameter systems [Seborg et al., 2010]. This meaning that the variables describing the state are functions of space, position through the heat exchanger, and time. By ignoring any radial movements this leaves a system where the temperature is described by a partial differential equation. The equation having partial derivatives in z, distance from the heat exchanger inlet, and t time. Because a numerical solver is used to solve the problem, it is required that one of the variable be discretized, in order to transform the partial differential equation to a system of ordinary differential equations. To then solve the problem a finite volume method approach is used. This means discretization of the system in the z direction, creating a series of lumped nodes instead. These nodes are then treated individually where balance equations can be formulated for each node. The streams are then described as a system of ordinary differential equations with derivatives with respect to time. Each set of differential equations describe the dynamics of the streams at a node point.

One side of the heat exchanger as a series of volume nodes is shown in figure 3.1. With Q_i being the heat transferred to one node.

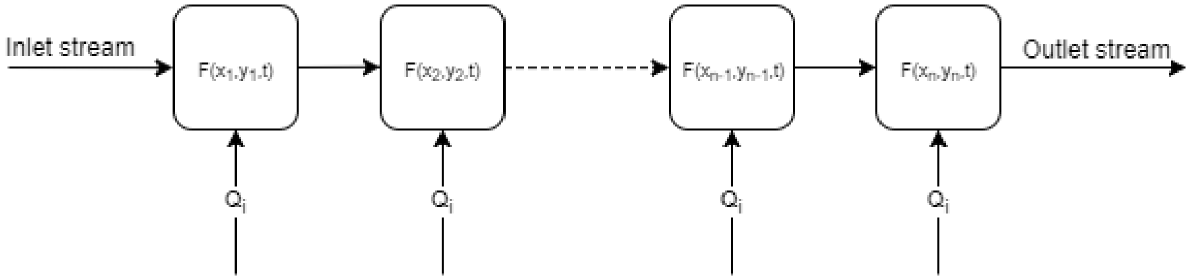


Figure 3.1: Lumped model for one side of the heat exchanger

3.1 Model requirements

In the optimal solution found in Verheyleweghen and Jäschke [2016] the refrigeration streams are always in the two phase region, in other words experiencing no sub-cooling or super heating. Sub-cooling being if point 2 in Figure 1.1 moves to the left into the liquid region, and super-heating if point 4 moves to the left into the vapour region. This can be done for a steady state model as the input and output of the evaporators and condensers is explicitly stated, but for the dynamic model this is no longer done. It is assumed that in this model disturbance or change in operational conditions can lead to either pure vapor or liquid phase being present inside a heat exchanger, causing sub-cooling or super-heating. This is not necessarily the case for refrigeration cycles as zero sub-cooling and super-heating can be achieved by design Jensen [2008].

This requirement allowing for sub-cooling or super-heating is what cause most challenges with trying to formulate the model. A real model should be robust enough to be able to handle all phase regions at any point in the exchanger. Equations describing the system and its dynamics are required for all three possible phase regions. It should also be able to handle possible phase region transitions that can occur. The main difference for the regions being that while in the single phase region, heat transfer causes temperature change, whilst in the two phase region it causes evaporation or condensation. Because it is not possible to determine when and where the transitions might occur, the model must be able to handle this during simulation, and act accordingly.

3.2 First approach

Having a state variable go from one equation to another will cause the model to be non-smooth at the switching points. This combined with the need for need for multiple sets of equations is not ideal if the end goal from the model is optimization. Because of this in the first approach it was investigated if all the phase regions could be described by one set of equations. The idea being one set of equations where at the switching point, a smooth and continuous change between the equations is done.

To do this the idea was to use an analytic approximation of the Heavyside step function as a part of the variable equations.

$$H_f(x) \approx \frac{1}{1+e^{k(s-x)}} \tag{3.1}$$

Where for a large value of k, H_f will quickly switch from 0 to 1 at the point $x = s$. If the goal is the function switching from zero to one above a certain temperature, the equation becomes

$$H_f(T) \approx \frac{1}{1+e^{k(T_{sat}-T)}} \tag{3.2}$$

If then T, the temperature, is above the saturation temperature, the function value will be 1. A more complex switching expression can be expresses as a function of temperature and the liquid fraction, requiring both to be of certain values. For example if the temperature is bellow the saturation temperature, and the liquid fraction is sufficiently far bellow 1, that would indicate that the system is in the two phase region.

The switching function H_f would then be included in the variable equations to effectively mute the parts not valid in the current area. This did not work for a few reasons. First off all, if sufficiently small steps were not taken during simulation, the system would oscillate back and forth between phase regions as certain switching functions would switch every step. Another problem occurs with variable that have no physical meaning in certain phase regions, such as for the compressibility factors. Because of this no further work was done in this project using this approach. Approaches using smoothing methods can however be expressed using complementary conditions to handle the existence or non existence of phases. Gopal and Biegler [1999]

3.3 Second approach

For the second approach three different sets of equations are instead developed for each possible phase region. The idea being a model where the solver, depending on the node's state, using a logical statement applies the correct set of equations to that node. As the different sets of equations contain both differential equations and algebraic equations they are a form of differential equations known as differential-algebraic equations (DAE). This approach is based on Zotica [2017] where a similar approach is taken, although as the stream conditions differ, the resulting equation sets are different.

As the nodes are lumped systems, the heat transferred between two nodes them can be expressed by

$$Q_i = UA_i(T_h - T_c) \tag{3.3}$$

With U being the overall heat transfer coefficient, A_i the node area, and T_h and T_c the hot and cold temperature. As only one side of the heat exchanger is modeled, the total heat transfer Q is set to be constant. Resulting in the heat transferred to a node being

$$Q_i = \frac{Q}{N} \tag{3.4}$$

This simplification implies that the non modeled side of the heat exchanger has constant a temperature. Also as the total heat transfer coefficient U is a function of both streams heat transfer coefficient h, they are also thereby assumed to be constant. For the following sections number of nodes N and the total volume of the modeled side V are given as model parameters. First the equations for the two phase region are formulated, and then for the single phase regions.

3.3.1 Equations for two phase region

When the system is in the two phase region, heat transferred to and from the system causes evaporation or condensation. Because of this the set of equations where formulated around mass balances for the two phases. In the following equations, all variables are for node i if not otherwise stated. These differential equations for the liquid and vapor phase holdups are

$$\frac{dn_{l,i}}{dt} = m_{i-1}x_{i-1} - m_i x_i + \frac{Q_i}{h_{vap}} \tag{3.5}$$

$$\frac{dn_{g,i}}{dt} = m_{i-1}(1 - x_{i-1}) - m_i(1 - x_i) - \frac{Q_i}{h_{vap}} \tag{3.6}$$

Here n_l and n_g are the molar holdups for the liquid and gas phases in node i. The molar flow and the liquid fraction m and x represent the inlet and the outlet flow from the node. The change in holdup from evaporation/condensation for both phases is the $\frac{Q_i}{h_{vap}}$ term. The functions change slightly for the first node where the m_{i-1} and x_{i-1} terms are replaced by the inlet molar flow and inlet liquid fraction m_{in} and x_{in}. It is therefore assumed that the molar flow of a phase is directly proportional to the liquid fraction of the node it is leaving. The liquid fraction is expressed as a fraction of the molar holdup of of the phases.

$$x_i = \frac{n_{l,i}}{n_{l,i} + n_{g,i}} \tag{3.7}$$

In order to have the flow rate be a function of the node states, an always open valve equation is introduced. The flow rate is then determined by the square of the pressure drop ΔP, and a constant k.

$$m = k\sqrt{\Delta P} \tag{3.8}$$

The ideal gas law with the compressibility factor from 2.3c is used to calculate the pressure in the node.

$$P = \frac{n_g RT}{V_g} Z_g \tag{3.9}$$

Where V_g is the volume of the gas phase, R is the gas constant, T the temperature and Z_g the compressibility of the vapor phase. The gas phase holdup is already known from equation 3.5.

The volume of the vapour phase is found by first calculating the liquid phase volume V_l. As the volume of each node is constant, the vapour volume is then the liquid volume subtracted from the node volume V_{node} .

$$V_l = n_l V_{m,l} \tag{3.10}$$

$$V_g = V_{node} - V_l \tag{3.11}$$

$V_{m,l}$ is the molar volume of the liquid phase and is calculated by the equation of state 3.14. The total volume of the node V_{node} is constant and is given by the total heat exchanger volume divided by the number of discretization points N.

$$V_{node} = \frac{V}{N} \tag{3.12}$$

The temperature is calculated using the Antoine equation 2.4.

$$T = \Big(\frac{B'}{A' - log_{10}(P)}\Big) - C' \tag{3.13}$$

The compressibility factors for the gas and liquid phase are then calculated from the Peng-Robinson polynomial equation with the different solutions for each phase as explained in 2.1.

$$0 = Z_g^3 - Z_g^2(1-B) + Z_g(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{3.14}$$

$$0 = Z_l^3 - Z_l^2(1-B) + Z_l(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{3.15}$$

One of the problems with the valve equation is if the pressure difference is negative. Without any logical operator to change the sign, this would result in the square root of a negative number being calculate. Because of this, any possible backwards flow is not included in the model, with the equation reformulated as

$$m_i = k\sqrt{max((P_i - P_{i+1}), 0)} \tag{3.16}$$

Here the max equation will return 0 if the pressure difference is negative. For the last node, the P_{i+1} term is replaced with the outlet pressure P_{out}. The equation constant k relates the molar flow to the pressure difference between the nodes.

$$\Delta P_{node} = \frac{\Delta P}{N} \tag{3.17a}$$

$$k = \frac{m_{in}}{\sqrt{\Delta P_{node}}} \tag{3.17b}$$

Note that k is a constant and is only calculated once before simulations. The term ΔP in equation 3.17 is desired pressure drop given for the flow m_{in}.

As the heat transfer term is included in the mass balances, the internal energy of each node can be calculated explicitly.

$$U = h_l n_l + h_g n_g - PV \tag{3.18}$$

To then summarize, the equations that are solved for each node through the heat exchanger are

$$\frac{dn_{l,i}}{dt} = m_{i-1}x_{i-1} - m_i x_i + \frac{Q_i}{h_{vap}} \tag{3.19a}$$

$$\frac{dn_{g,i}}{dt} = m_{i-1}(1 - x_{i-1}) - m_i(1 - x_i) - \frac{Q_i}{h_{vap}} \tag{3.19b}$$

$$x_i = \frac{n_{l,i}}{n_{l,i} + n_{g,i}} \tag{3.19c}$$

$$V_l = n_l V_{m,l} \tag{3.19d}$$

$$V_g = V_{node} - V_l \tag{3.19e}$$

$$P = \frac{n_g RT}{V_g} Z_g \tag{3.19f}$$

$$T = \Big(\frac{B'}{A' - log_{10}(P)}\Big) - C' \tag{3.19g}$$

$$0 = Z_g^3 - Z_g^2(1 - B) + Z_g(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{3.19h}$$

$$0 = Z_l^3 - Z_l^2(1 - B) + Z_l(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{3.19i}$$

$$m_i = k\sqrt{max((P_i - P_{i+1}), 0)} \tag{3.19j}$$

$$U = h_l n_l + h_g n_g - PV \tag{3.19k}$$

3.3.2 Equations for one phase region

Because the model was developed with a focus on the two phase region, certain problems arise when trying to implement the DAE equations for the pure liquid/vapour phase. From the solvers point of view, there should be no difference between the sets of equations. For each node the same set of variables have to be computed in each iteration regardless of phase region. Some of the variables such as for volume V_g and V_l can just be set to zero for the non present phase. The same can be done for the molar holdup of the non present phase, or the liquid fraction. The same quick solution can not be done for the pressure. In the two phase regions the pressure is found from the equation of state combined with compressibility and the Antonie equation as described in 3.3.1. This can no longer be done as the Antoine equation is no longer valid. Heat transfer no longer causes evaporation or condensation but temperature change. The same equation can then not be used for pressure drop, as it would result in major pressure differences occurring. The pressure drop is generally quite low, with density and compressibility changing with temperature.

The pressure is then instead assumed to have a constant linear drop trough the single phase regions. Constant pressure differences cause the molar flow also being constant in these regions. This simplification causes there to be no mass dynamics in the single phase regions.

The equations for the pure gas phase region are then as follows. As the pressure and volume is constant in each node, the change in internal energy is equal to the change in enthalpy for a node.

$$\Delta H_i = \dot{H}_{i-1} - \dot{H}_i \pm Q_i \tag{3.20}$$

Rewriting the enthalpy streams for temperature, holdup, molar flow and specific enthalpy gives

$$n_{g,i} c_{p_{g,i}} \frac{dT}{dt} = m_{i-1}h_{g,i-1} - m_i h_{g,i} \pm Q_i \tag{3.21a}$$

$$\downarrow \tag{3.21b}$$

$$\frac{dT}{dt} = \frac{m_{i-1}h_{g,i-1} - m_i h_{g,i}}{n_{g,i} c_{p_{g,i}}} \pm \frac{Q_i}{n_{g,i} c_{p_{g,i}}} \tag{3.21c}$$

The enthalpy and heat capacity for the phase is are calculated as described in 2.3. The new equation for the pressure is then set as a linear drop from the input pressure.

$$P_i = P_{i-1} - \Delta P_{node} \tag{3.22}$$

For the first node the P_{i-1} term is replaced with the pressure of the input stream P_{in}. In this way the pressure can be changed during simulation, it will simply propagate through for each step. The same is done for the molar flow.

$$m_i = m_{i-1} \tag{3.23}$$

Because the volume of the gas phase is constant, it is set to be equal to the node volume and used to calculate the holdup in each node.

$$n_g = \frac{V_g}{V_{m,g}} \tag{3.24}$$

Where $V_{m,g}$ is the molar volume of the gas phase calculated from the equation of state.

$$V_{m,g} = \frac{RTZ_g}{P} \tag{3.25}$$

The holdup for the phase will then change with temperature and pressure, but because it is not connected to the molar flow it will not be balanced. It is instead used to calculate how the node heat capacity changes with density. The compressibility is found with the same equation used in the two phase region equation 3.14.

The equations for the single phase region are then

$$\frac{dT}{dt} = \frac{m_{i-1}h_{g,i-1} - m_i h_{g,i}}{n_{g,i}c_{p_{g,i}}} \pm \frac{Q_i}{n_{g,i}c_{p_{g,i}}} \tag{3.26a}$$

$$P_i = P_{i-1} - \Delta P_{node} \tag{3.26b}$$

$$m_i = m_{i-1} \tag{3.26c}$$

$$n_g = \frac{V_g}{V_{m,g}} \tag{3.26d}$$

$$V_{m,g} = \frac{RTZ_g}{P} \tag{3.26e}$$

$$0 = Z_g^3 - Z_g^2(1-B) + Z_g(A - 2B - 3B^2) - (AB - B^2 - B^3) \tag{3.26f}$$

For the liquid only region, the equations are the same except the gas phase variables are changed with their liquid phase counterparts. It is also then the volume of the liquid phase which is equal to the node volume, the gas volume which is zero and the liquid fraction set to 1.

Chapter 4

Solving the DAE systems

4.1 Index of DAE systems

DAE systems are are a type of differential equations where some of the variable derivatives are not explicitly stated but instead expressed by algebraic equations. The way DAE's are then generally solved is by transformation into a system of ordinary differential equations. A general expression for the DAE system is the

$$\dot{x} = f(x, y, t)$$
$$0 = g(x, y, t)$$

If g can then be solved for y, by differentiating once, the system is of index 1. The number of times this must be done is known as the *index* of the system. The resulting equations are then a system of ordinary differential equations. The solver used in this project is ode15s from Matlab which can solve DAE systems of index 1.

4.2 Model stiffness

In the DAE set for the two phase region, because pressure is introduced as a dynamic variable, the equations become what is known as a stiff. Stiffness arises when there are variables in the equations that can experience very rapid changes in some areas. For the numerical solution to then be stable, the required step size for the solver must be extremely small, if using explicit or fixed step methods. Step sizes to large can miss certain events causing the error between the numerical solution and the true solution to be very large. It can be shown that for step sizes above certain limit can cause the solution to oscillate and diverge. Süli and Mayers [2003]

As said the stiffness of the two phase region is because of the pressure. Changes in pressure for a node can occur extremely fast while changes in say the liquid fraction occur significantly slower. Although the pressure can change fast, the magnitude of the changes are very small. But this is not true for the molar flow, which is very sensitive to changes in pressure. To illustrate this a brief example is given where the pressure of the system is 30 bars with 10 discretization nodes in the model. If the then desired model pressure drop is 10 mbar, and the given flow rate is 1 mole per second. This would result in the calculated k constant from equation 3.17 being 31.6. If there is then an instantaneous 1% increase in the pressure the resulting flow rate would see a 1700% increase.

4.3 Solver

To avoid instability and unreasonable calculation time, the ode15s solver uses a variable step, variable order method using numerical differentiation formulas, MathWorks. The solver works by taking the system of equations, a given time span and initial conditions as input parameters. It then returns solutions for all evaluations points. As explained in section 3.3 the equations repeat for each node, resulting in the number of equations being 12 times the number of nodes. Ode15s uses a mass matrix for determination of which equations are differential and which are algebraic given by the form

$$M(t, y)y' = f(t, y) \tag{4.2}$$

Where $M(t, y)$ is a matrix of zeroes with either 1 for differential or 0 for algebraic in the diagonal positions. If only one set of DAE equations are used this matrix will be constant as the equations stay the same through out the simulations. Using multiple sets would require updating the mass matrix to indicate which equation go from algebraic to differential or vice versa.

4.4 Initial conditions

When solving the DAE equations consistent initial conditions are required for the differential and the algebraic equations. If the given initial conditions are far from any real solution, the solver will have problems converging to a solution even when high tolerances are specified. Luckily once a solution has been found, it can be stored and used as initial conditions for other simulation, where it will also be consistent. One way to make this process easier is to first find a steady state solution by setting the heat transfer rate to zero. By setting Q equal to zero, the solver should, for a given feed, find the steady state solution with correct pressure, temperature and liquid compositions etc. Higher error tolerance can also make this process easier. If a solution is then found, the saved state values are used to initiate other simulations where heat transfer, or other simulation events can be included. If the inlet stream specifications change drastically, or other modeling parameters change such as heat exchanger volume, number of node, or specified pressure drop are changed a new set of initial conditions have to be calculated.

Chapter 5

Model simulations

In this section the equations for the different phase regions are studied individually to make sure they are behaving as expected. The two phase region will be focused on, as it is this region which is of most importance for modeling the streams in the refrigeration cycles.

5.1 Two phase region

The stream conditions used to simulate behaviour in the two phase region are taken from the original optimal conditions from Verheyleweghen and Jäschke [2016]. In particular looking at the heat exchanger in the methane cooling refrigeration cycle after the Joule-Thompson valve. In this evaporator low pressure, low temperature methane evaporates as the final cooling step for the natural gas stream. Conditions are shown in Table 5.1.

Table 5.1: Stream conditions used for two phase region simulations

Variable	Value	
T_{in}	118.7	[K]
P_{in}	1.76	[bar]
x_{in}	0.97	[%]
m_{in}	3	$[\frac{mol}{s}]$
ΔP	0.01	[bar]
Nodes	50	[-]
V	1	$[m^3]$
Q	10000	$[\frac{j}{s}]$

As the exact liquid fraction after the Joule-Thompson vale is not known, the liquid fraction is set to be 0.97 $\%$. The other modeling parameters such as molar flow, and heat exchanger volume are set to make the simulating conditions easier. The pressure drop over the heat exchanger is set at 0.01 bar, which is quite high, and the number of nodes is set to 50. This is done as it slightly reduces the issues regarding the stiffness of the model. The total heat transfer is set to 10 000 joules per second, as described in section 3.3, because the other side of the heat exchanger is not modeled.

5.1.1 Steady state

A set of consistent initial conditions are first found by setting the total heat transfer to zero. Then while using these consistent initial conditions the heat transfer is set to 10 000 and the solver is set to solve over a time span of 10 000 seconds, where the system will have reached a new steady state.

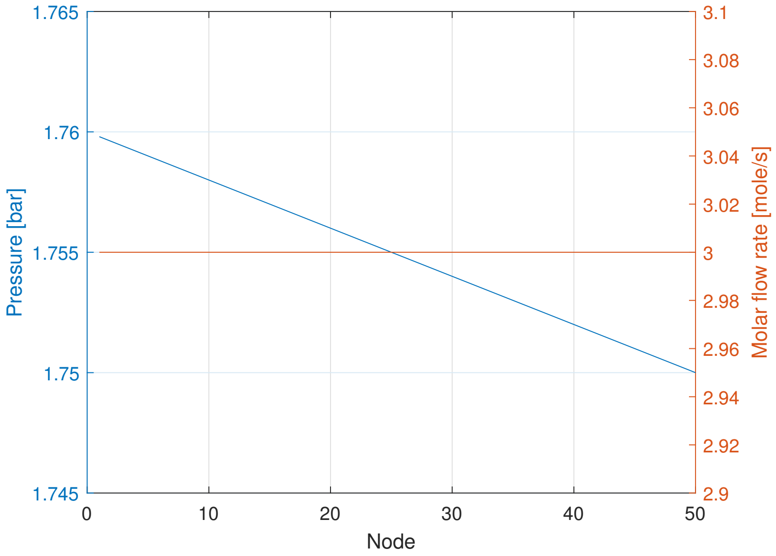


Figure 5.1: Pressure and molar flow through the heat exchanger at steady state.

As figure 5.1 shows the molar flow rate has a steady state value of 3 as intended for all nodes when the pressure drop is equal to what was used to calculate k in equation 3.16.

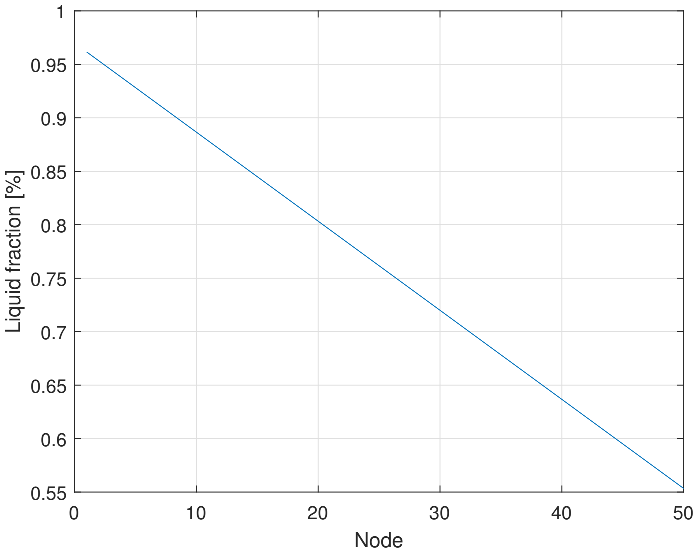
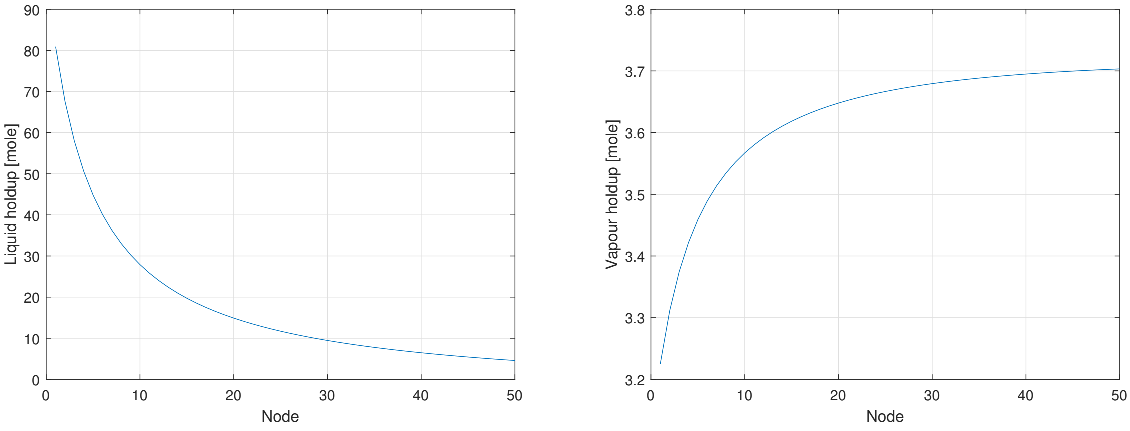


Figure 5.2: Liquid fraction at steady state

The liquid fraction decreases linearly through the heat exchanger as a result of the heat transfer to each node being kept constant. If both sides where to be modeled, one could expect the heat transfer rate to decrease as the heat transfer coefficient decreases with liquid fraction.

More interesting the liquid phase holdup and the vapour phase holdup change rapidly in the first few nodes, then see slower absolute change. This absolute change per cell slows down through the heat exchange, even though the heat transferred is the same. This can be explained by the major different in density for the two phases. As the liquid phase is significantly denser, the total molar holdup drops significantly through the heat exchanger as vapour volume comes to dominate the volume in each node.



(a) Liquid holdup

(b) Vapour holdup

Figure 5.3: Liquid and vapour phase holdup at steady state

As the liquid fraction gets smaller and smaller, the volume of each cell is mainly made up of the gas phase as shown in figure 5.4.

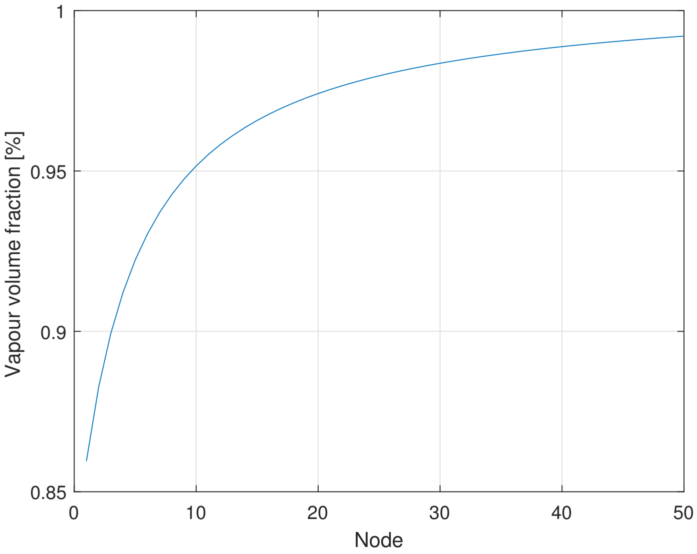


Figure 5.4: Vapour phase volume fraction at steady state

5.1.2 Dynamic response

The main rational for a dynamic model is so the can handle changing condition over time. Therefore how the model handles certain step changes in the input variables will be simulated. For the two phase region the temperature and pressure cant be changed independently, as the fluid would no longer be in the two phase region. Therefore the model response to a rapid change in the molar flow rate, and the liquid fraction will be explored. Note that whereas in the previous plots where at a given time with the x-axis being nodes, in the next few plots changes occurring in nodes over time are shown.

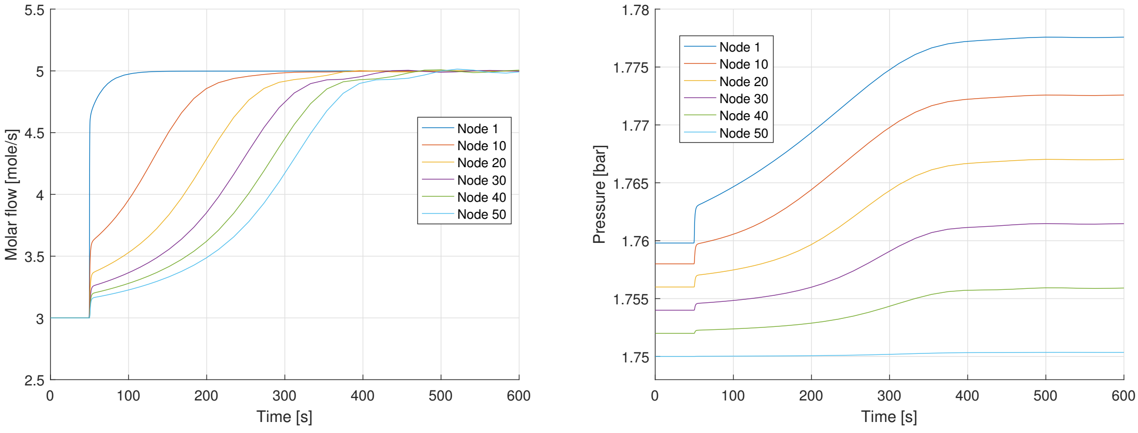
As instantaneous step changes would result in the derivative being infinite, the step changes are simulated as occurring over the span of roughly one second. The step changes simulated are changing the inlet flow rate from 3 to 5, and the inlet liquid fraction from 0.97 to 0.95. The equations for the step changes are as follows.

$$m_{in} = m_{in} + \frac{2}{1 + e^{-10(t-50)}} \tag{5.1a}$$

$$x_{in} = x_{in} + \frac{-0.02}{1 + e^{-10(t-50)}} \tag{5.1b}$$

Both step changes occur at the 50 second mark, and are simulated separately.

For the flow rate step change, the immediate effect are a jump in the flow rate for all nodes, with the largest jumps being in the first few nodes. This can be explained by figure 5.5, as the pressure increase for each node after the 50 second mark. Because the pressure changes are larger in the first nodes, the corresponding molar flow changes are larger. After roughly 500 seconds things have stabilized and all molar flow rates are at the new steady state value. Figure 5.5 shows how the pressure difference between nodes has increased as a result of the higher flow rate.

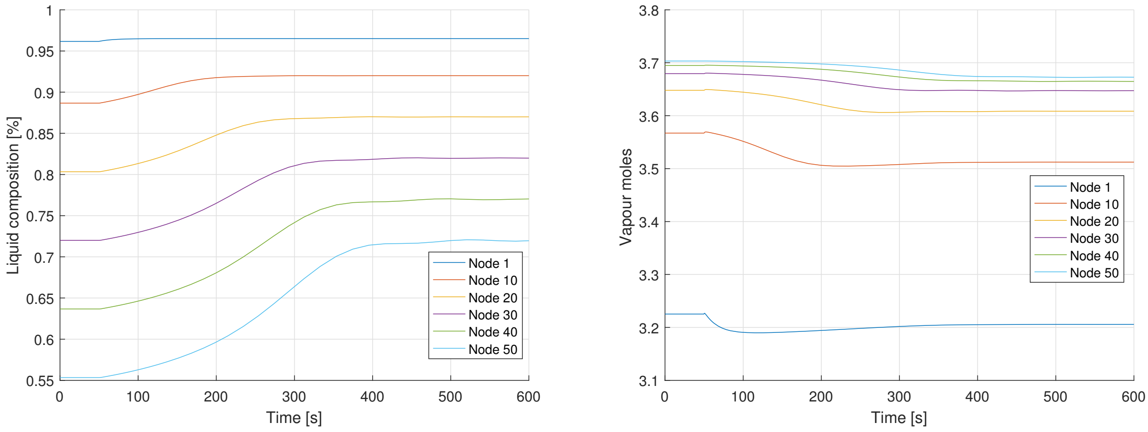


(a) Molar flow

(b) Node pressure

Figure 5.5: Molar flow rate and node pressure after step change of inlet flow rate

After the initial rapid changes caused by the pressure, there is a slower dynamic moving the system to a new steady state. This is best shown in figure 5.6, where it can be seen that the liquid fraction has a slower response. When the molar flow rate is increased, the new steady state liquid fraction for each node is also higher, as less refrigerant can evaporate. This change occurs slower due to the slow change in the makeup of the holdup for each node.



(a) Liquid fraction

(b) Vapour fraction

Figure 5.6: Liquid fraction and vapour holdup after step change of inlet flow rate

For the step change in the inlet liquid fraction the changes in the liquid fractions are shown in figure 5.7. For the first node at the entrance , the liquid fraction quickly readjusts itself to a few percentage point under the inlet stream as expected. What might not be as expected is the change in the rest of the nodes, especially towards the end. The liquid fraction actually increases quite dramatically before falling to the new steady state value after more time has passed.

What is actually happening is easier understood by looking at the change in pressure and molar flow shown in figure 5.8. As the new inlet makeup enters there is a spike in the molar flow which moves through the heat exchanger as a wave. This increase in flow rate, causes refrigerant with higher a liquid fraction to be more quickly pushed through. As it is moving quicker, the heat transferred to the bulk has less of an effect and a lesser amount of it is evaporated. After this wave has pushed through, and the molar flow falls back to its steady state value where the new liquid fraction has fallen to a lower value than before the step change.

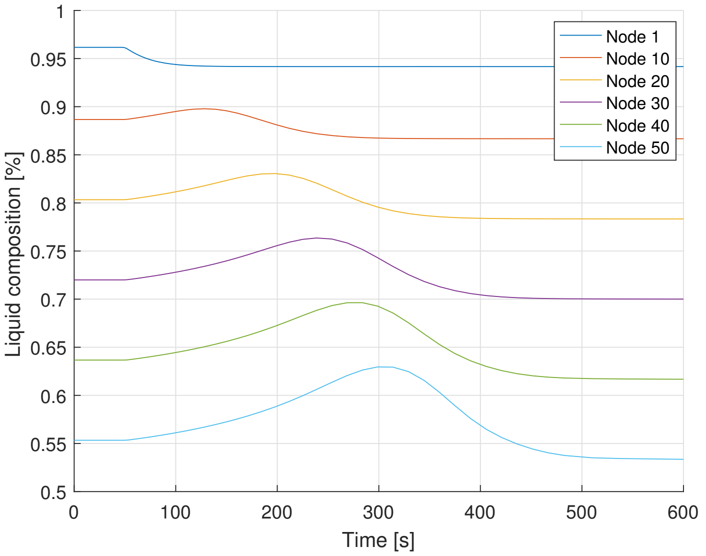
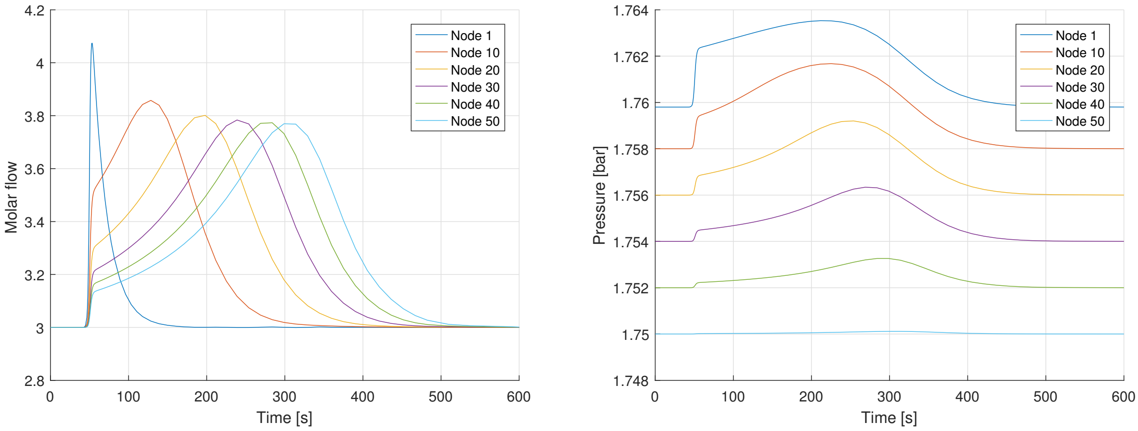


Figure 5.7: Liquid fraction after step change of inlet liquid fraction



(a) Molar flow

(b) Pressure

Figure 5.8: Molar flow rate and pressure for step change of the inlet liquid fraction

The reason for the pressure jumps is because when the inlet liquid fraction is decreased, the vapour flow rate is increased. This results in the volumetric flow rate significantly increasing due to the large density differences. As the molar flow rate is constant, changing the liquid fraction from 97% to 95% causes a significant relative increase in the vapour flow rate. Then because the pressure is calculated solely from the vapour phase there are spikes in pressure and molar flow rate.

5.2 One phase region

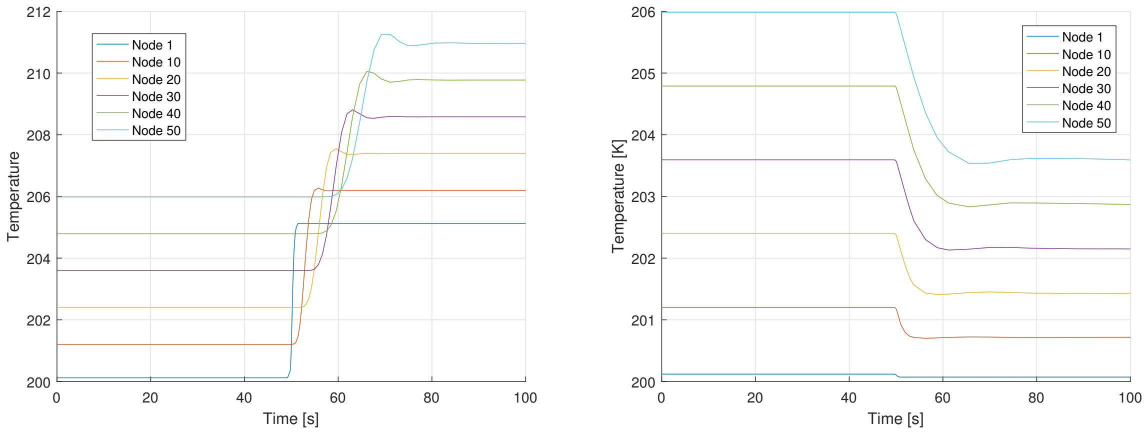
Because of the introduced simplifications for the equations of the one phase region, the dynamics are significantly simpler than those for the two phase region. Although the equations can handle changes in molar flow or pressure, there are no dynamics or delays present. The new values are simple propagated through the nodes at each time step. This is not the case for the temperature which does have its derivative in the equation set. Two step changes are simulated for the single phase, one increase in the inlet temperature of 5 K and one step change for the molar flow rate. Both are done in in a similar manner as for the two phase region.

The stream simulation is of a pure gas phase stream being heated. The simulated conditions are given in Table 5.2

Table 5.2: Stream conditions used for single phase region simulations

Variable	Value	
T_{in}	200	[K]
P_{in}	2	[bar]
x_{in}	0	[%]
m_{in}	3	$[\frac{mol}{s}]$
ΔP	0.01	[bar]
Nodes	50	[-]
V	1	$[m^3]$
Q	-1500	$[\frac{j}{s}]$

For the steady state conditions in the single phase region , there is an almost linear increase in temperature. It is not linear as the change in temperature cause changes in density , which in turn changes the total heat capacity for the node.



(a) Temperature step change

(b) Flow rate step change

Figure 5.9: Temperature change for step changes of inlet temperature and inlet molar flow rate

Both plots in figure 5.9 are as expected. For the temperature change, the same jump makes it's way through the heat exchanger with the flow. For the flow rate increase, in a similar manner to the two phase are, as the retention time decreases, the total heat transfers to each unit is decreased, causing the outlet temperature to also decrease.

5.3 Stiffness and time scale differences

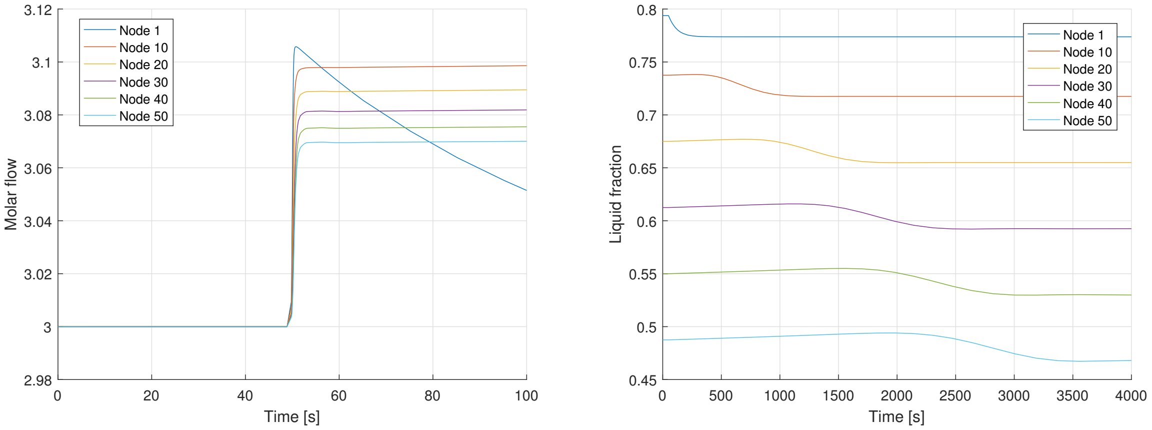
To highlight the different time scales of the dynamics in the two phase region, simulations were done for a stream where the pressure is significantly higher. One major difference when the pressure is higher is the difficulty in finding consistent initial conditions. As the modeled pressure drop is still the same, if the initial conditions are wrong and cause large pressure differences estimated, flow rates at the beginning might be to large for the model to handle. This is especially the case for when the liquid fraction is high as the volume of the gas phase is small.

Table 5.3: Stream conditions used for time scale simulations

Variable	Value	
T_{in}	181	[K]
P_{in}	34.4	[bar]
x_{in}	0.8	[%]
m_{in}	3	$[\frac{mol}{s}]$
ΔP	0.01	[bar]
Nodes	50	[-]
V	1	$[m^3]$
Q	7500	$[\frac{j}{s}]$

On the other hand, as a result of the high pressure the densities for the liquid and vapour phase are significantly closer relative to the previous case. It can therefore handle rapid changes in liquid fraction better. This is because the relative volume changes are smaller. Figure 5.10 shows a response to the same two percent change in liquid fraction for a stream with the new conditions shown in Table 5.3.

The different time scales for molar flow and liquid fraction are clear. The changes to molar flow occur immediately while changes for the liquid fraction occur over thousands of seconds.



(a) Molar flow rate

(b) Liquid fraction

Figure 5.10: Molar flow rate and liquid fraction after a step change in the inlet liquid fraction

Chapter 6

Discussion

The three equations sets work quite well for their respective regions. The model can handle streams in each phase region if given consistent initial conditions. Using the Peng-Robinson equation of state, the molar volumes are inline with data from NIST for both the liquid and and vapour phase. The only drawback of using compressibility is the need to manually set the initial conditions for the different compressibility roots to ensure the correct roots are used for their corresponding phase. For the two phase region consistent initial conditions can generally always be found, with the exception when there is a very high liquid fractions, especially for high pressure conditions. However once a consistent set of initial conditions have been found, the system is quite robust to changes. The two step changes from section 5.1.2 are very large and occur very quickly, and under real conditions changes of these magnitudes would not occur over the small time frames simulated.

The downside of the simulations done is that there are very few perturbations to the system over time. A problem arising from the stiffness in the system is the dramatic increase in function evaluations required when changes are done to the system. If there are no changes to the inlet stream this is not a problem, as a new steady state is quickly found and the solver uses increasingly large intergration steps. But if the inlet stream were to consistently change, or some noise were to be present in the inlet variable, the model would could slow down significantly as variables such as the flow rate would consistently have major jumps.

The simulations done for the model have only been while operating in one of the three phase regions at a time. One of the original requirements from section 3.1 was that the same model be able to transition freely between the different phase regions. The model can not handle these transitions for a few reasons. As a result of the model having been built with a focus on the two phase region, the set of variables solved for, lend themselves poorly for being used to handle the other phase regions. Most noticeably because some of the variables equations go from being algebraic to differential. Although the ODE15s solver does allow for the mass matrix to be state dependent, meaning that if a node where to be detected in a new phase region, the mass matrix could be updated to signal the change in the equation type. The better approach however would be to rather rewrite the system of equations to have differential equations that are valid for all phase regions. For instance, instead of balancing over each phase in the two phase region, having one mass balance and one energy balance. Then the same equations would at least be differential in all phase regions such as in Zotica [2017].

Another challenge with handling the transition is the need that the variables be consistent from one region to the next. In the current implementation this is not included as transitions were not simulated. The equations used for the two phase region would also lend themselves poorly for approaching the liquid - two phase transition. As the liquid fraction approaches one, both n_g and V_g approach zero. The equation for pressure would then eventually fail and give significant errors due to numerical approximations. Another problem during transition would be the compressibility factors. When one of the phases disappears the corresponding compressibility solution becomes complex, so some logical action must be taken to ensure smooth transitions. A solution to this could be to remove the compressibility from the set of equations and instead have the solution to the polynomial be a part of the logical statement deciding which phase region a node is in.

Having only one flow rate parameter would also cause issues. Because then the liquid fraction of the flow out of a node will then always be the same as for the node. Problems then arise for the first node after a phase region transition. The first node in a stream leaving the two phase region would then have an inlet flow containing some part of the phase not present in the receiving node. The receiving node would then not included equations for describing that phase. A new set of equations should then instead have two flow rates, one for the liquid and one for the vapour phase. These flow rates could then be calculated independently.

As explained there are several problems with the current model formulations that make including phase region transitions very problematic. Although some of the solution and equations introduced might be elegant for their own region, they often result in major problems when trying to fulfill the bigger picture requirements for the model. It is clear that they way forward would be to first reformulate the equations with a greater focus on having them work for the transition events. Although the first approach was dropped due to the issues laid out in section 3.2, it could be interesting to check weather different solvers could handle the idea better. If the problems related to the step sizes could be solved it might be an elegant solution if formulated properly.

Chapter 7

Conclusion

In this project the requirements for a heat exchanger model to be used for evaporators and condensers in a dynamic LNG liquefaction process were investigated. A model was developed with the intention of being able to handle all three phase regions that can occur, these being vapour, liquid and both at the same time. Although the model can handle all three regions independently, it cant handle any transitions between the regions inside the heat exchanger.

The model was formulated as a series of lumped nodes, with a set of DAE's describing the changes for each node over time. Three different sets of DAE's were formulated, each set for it's own phase region. Simulations where done to see how the systems of equations behave for streams only occupying one of the phase regions at a time. Although finding consistence initial conditions proved to be challenging for the two phase region, once these had been found the DAE's where quite robust given the stiffness involved in the system. The equations can handle large changes in both the inlet flow rate, and the inlet liquid fraction even when occurring over small time frames.

As a result of the model being developed with a focus on the two phase region, certain model decisions lead to major difficulties in implementing capabilities required for phase region transition events. It is unclear how transitions could be implemented with the current sets of equation. The first steps in further work should be to first redevelop the sets equations for each phase, or to shift to the modeling to focus on smoothing methods instead.

Bibliography

Adriaen Verheyleweghen and Johannes Jäschke. Self-optimizing control of an lng liquefaction plant. 2016.

Ding-Yu Peng and Donald B. Robinson. New two-constant equation of state. *Industrial Engineering Chemistry Fundamentals*, 15, 02 1976.

Ravindra S. Kamath, Lorenz T. Biegler, and Ignacio E. Grossmann. An equation-oriented approach for handling thermodynamics based on cubic equation of state in process optimization. *Computers Chemical Engineering*, 34(12):2085 – 2096, 2010.

D.E. Seborg, D.A. Mellichamp, T.F. Edgar, and F.J. Doyle. *Process Dynamics and Control*. John Wiley & Sons, 2010.

Jørgen Bauck Jensen. *Optimal Operation of Refrigeration Cycles*. PhD thesis, NTNU, 05 2008.

V. Gopal and L. T. Biegler. Smoothing methods for complementarity problems in process engineering. *AIChE Journal*, 45(7), 1999.

Cristina Florina Zotica. Dynamic Simulation of Heat Exchanger with Multicomponent Phase Change. Master's thesis, NTNU, 2017.

E. Süli and D.F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

MathWorks. Solve differential algebraic equations (daes). `https://www.mathworks.com/help/matlab/math/solve-differential-algebraic-equations-daes.html`.

P.J. Linstrom and Eds W.G. Mallard. *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*. National Institute of Standards and Technology.

B.E. Poling, J.M. Prausnitz, and J.P. O'Connell. *The properties of gases and liquids*. McGraw-Hill, 2001.

Appendix

Parameters and constants used in the simulation

Parameters and constants used in the simulations are given in the tables bellow. For Tables 7.1, 7.2 and 7.4 all values are taken from the NIST chemistry WebBook [Linstrom and W.G. Mallard]. Coefficients for gas phase heat capacity calculations in Table 7.3 are taken from Poling et al. [2001].

Table 7.1: Critical temperature T_c, pressure P_c and acentric factor w used for Peng-Robinson equation of state

	T_c [K]	P_c [Bar]	w [-]
Methane	190.564	45.99	0.01142
Ethane	305.33	48.718	0.0993
Propane	369.825	42.4766	0.1524

Table 7.2: Constants for the Antonie equation

	A	B	C
Methane	4.22061	516.689	11.223
Ethane	3.93835	659.739	-16.719
Propane	3.98292	819.296	-24.417

Table 7.3: Coefficients used for calculating heat capacity of gas phases

	c_1	c_2	c_3	c_4
Methane	1.925*10^{-1}	5.213*10^{-2}	1.197*10^{-5}	-1.132*10^{-8}
Ethane	5.409*10^{-0}	1.781*10^{-1}	-6.938*10^{-5}	8.713*10^{-9}
Propane	-4.224*10^{-0}	3.063*10^{-1}	-1.586*10^{-4}	3.215*10^{-8}

Table 7.4: Values used for liquid phase heat capacity and enthalpy of vaporization

	$c_{p,l}$ [J/mol K]	h_{vap} [J/mol]
Methane	53	8170
Ethane	68.5	14700
Propane	98.3	19040

Matlab Code

Main simulation file

```
%Main simulation file

%Constants and model parameters
const = struct();
const.T_in = 181 ;   %Inlet Temperature [k]
const.P_in = 34.4;     %Inlet Pressure    [bar]
const.m_in = 3;        %Inlet molar flow  [mol/s]
const.x_in = 0.8;     %Inlet liquid fraction

%Indicator of refrigerant type(Methane:1, Ethane:2, Propane:3)
%Scenario of stream (Two phase:1, Gas phase:2, Liquid phase:3)
const.ind =  1;
const.scenario = 1;

const.V = 1;           %Heat exchanger volume [m^3]
const.N = 50;          %Heat exchanger volume
const.Q = 5000;           %Number of nodes
const.delta_P = 0.01;  %Design pressure drop [bar]

%Load initial conditions
if (exist('y0.mat') == 2)
    load('y0.mat')
else
    y0 = init(const);
end


%Specification of the mass matrix
%Two phase mass matrix
m1 = ones(1,const.N*2);
m2 = zeros(1,const.N*11);

%Single phase mass matrix
m3 = zeros(1,const.N*5);
m4 = ones(1,const.N*1);
m5 = zeros(1,const.N*7);
if const.scenario == 1
    m = diag([m1,m2]);
end
if const.scenario == 2 || const.scenario == 3
    m = diag([m3,m4,m5]);
end

%Solver options
options = odeset('Mass',m,'RelTol',1*10^-5,'stats','on');

[t,y]= ode15s(@(t,y) HX(y,const,t),[0 10000],y0,options);
```

Initial conditions

```
%If no consisten initial conditions are saved
% a quick estimation can be done
function y0 = init(const)
%Constants
N = const.N;
```

```
V_cell = const.V/N;
R = 8.314;

%Guess for two phase scenario
if const.scenario == 1

    [A,B,C] = antonie(const.ind);
    T0(1:N) = 0;
    P0(1:N) = 0;
    for i = 1:N
        P0(i) = const.P_in -(const.delta_P/N)*i; %Linear pressure drop
        T0 (i) = (B/(A-log10(P0(i)))- C);        % Temperature from antonie
    end

    m0(1:N) = const.m_in;
    x0(1:N) = const.x_in;

    %Compressibility guesses have to be explicitly stated
    Z_l0(1:N) = 0.140331;
    Z_g0(1:N) = 0.53975;

    %Molar volume
    Vm_l0(1:N) = Z_l0.*T0*R./(P0*10^5);
    Vm_g0(1:N) = Z_g0.*T0*R./(P0*10^5);
    %Molar volume guess of intlet stream
    Vm_g_in = Z_g0(1)*const.T_in*R/(const.P_in*10^5);
    Vm_l_in = Z_l0(1)*const.T_in*R/(const.P_in*10^5);

    % Estimation of liquid volume fraction
    Vl_in = const.m_in.*x0*Vm_l_in;
    Vg_in = const.m_in.*(1-x0)*Vm_g_in;
    frac = Vl_in./(Vg_in+Vl_in);
    %Phase volumes
    Vl0(1:N) = const.V/N*frac;
    Vg0(1:N) = const.V/N - Vl0;
    %Holdups
    n_l0(1:N) = Vl0./Vm_l0;
    n_g0(1:N) = Vg0./Vm_g0;

    [cp_g,cp_l,h_g,h_l,h_vap] = heatandenthalpy(T0,P0,const);
    U_0(1:N) = - P0*V_cell + n_l0.*h_l + n_g0.*h_g;
end

%Initial conditions for gas only scenario
if const.scenario == 2

    n_l0(1:N) = 0;
    Vg0(1:N) = V_cell;
    Vl0(1:N) = 0;
    x0(1:N) = 0;
    m0(1:N) = const.m_in;
    Vm_l0(1:N) =0;
    Z_g0(1:N) = 0.9854;
    Z_l0(1:N) = 0;
    T0(1:N) = const.T_in;
    for i = 1:N
        P0(i) = const.P_in -(const.delta_P/N)*i;
        Vm_g0(i) = Z_g0(i)*T0(i)*R/(P0(i)*10^5);
        n_g0(i) = V_cell/Vm_g0(i);
    end

    [cp_g,cp_l,h_g,h_l,h_vap] = heatandenthalpy(T0,P0,const);
```

```
     U_0(1:N) = - P0*V_cell + n_g0.*h_g;
end
%Initial conditions for liquid only scenario
if const.scenario == 3

    n_g0(1:N) = 0;
    Vl0(1:N) = V_cell;
    Vg0(1:N) = 0;
    x0(1:N) = 1;
    m0(1:N) = const.m_in;
    Vm_g0(1:N) = 0;
    Z_g0(1:N) = 0;
    Z_l0(1:N) = 0.12436611795530638
    T0(1:N) = const.T_in;
    for i = 1:N
        P0(i) = const.P_in -(const.delta_P/N)*i;
        Vm_l0(i) = Z_l0(i)*T0(i)*R/(P0(i)*10^5);
        n_l0(i) = V_cell/Vm_l0(i);
    end

    [cp_g,cp_l,h_g,h_l,h_vap] = heatandenthalpy(T0,P0,const);
     U_0(1:N) = - P0*V_cell + n_l0.*h_l;
end

%Return initial guess array
y0 = [n_l0,n_g0,Vl0,Vg0,P0,T0,x0,m0,U_0,Vm_g0,Vm_l0,Z_g0,Z_l0];

end
```

DAE system

```
function system = HX(y,const,t)

R = 8.314*10^-5;  %Gas constant [m^3 bar /K mol]
N = const.N;
Vtot = const.V;
Vcell = Vtot/N;   %Node volme [m^3]
Qtot = const.Q ;
Qcell = Qtot/N;   % Node heat transfer [j/s]

%Calculation of valve constant
delta_P = const.delta_P;
dp_cell = delta_P/N;
P_out = const.P_in-(delta_P/N)*(N+1);
k = const.m_in/sqrt(dp_cell);

%Inlet stream parameters
T_in = const.T_in;
x_in = const.x_in;
P_in = const.P_in;

%Enthalpy inlet stream
[cp_g_in,cp_l_in,h_g_in,h_l_in,h_vap_in] = heatandenthalpy(T_in,P_in,const);
h_in = x_in*h_l_in+(1-x_in)*h_g_in;

%Changes to inlet stream over time

m_in = const.m_in ;  %+ 2/(1+exp(-10*(t-50)));
x_in = const.x_in ;  %-0.02/(1+exp(-10*(t-50)));
T_in = const.T_in ;  %+ 5/(1+exp(-10*(t-50)));
```

```
%Extract current values of all state variables
n_l = y(1:N);               %Liquid holdup
n_g = y(N+1:N+N);           %Vapour holdup
Vl  = y(2*N+1:2*N+N);       %Liquid volume
Vg  = y(3*N+1:3*N+N);       %Vapour volume
P = y(4*N+1:4*N+N);         %Pressure
T = y(5*N+1:5*N+N);         %Temperature
x = y(6*N+1:6*N+N);         %Liquid fraction
m = y(7*N+1:7*N+N);         %Molar flow rate
U = y(8*N+1:8*N+N);         %Internal energy
Vm_g = y(9*N+1:9*N+N);      %Vapour phase molar volume
Vm_l = y(10*N+1:10*N+N);    %Liquid phase mlar volume
Z_g = y(11*N+1:11*N+N);     %Gas phase compressibility
Z_l = y(12*N+1:12*N+N);     %Liquid phase compressibility

%Calculation of enthalpies, heat capacities compressibility
% polynomial coefficients and extraciton of antonie constants
[h_g,h_l,cp_g,cp_l,h_vap,A,B,C,A_PR,B_PR] = data(y,const);

%Prealocation memory for arrays used for functions
f1(1:N) = 0;
f2(1:N) = 0;
f3(1:N) = 0;
f4(1:N) = 0;
f5(1:N) = 0;
f6(1:N) = 0;
f7(1:N) = 0;
f8(1:N) = 0;
f9(1:N) = 0;
f10(1:N) = 0;
f11(1:N) = 0;
f12(1:N) = 0;
f13(1:N) = 0;

%Equations for two phase stream scenario
if (const.scenario ==1 )

    %Eqauation for the first node
    f1(1) = m_in*x_in-m(1)*x(1)- Qcell/h_vap(1);
    f2(1) = m_in*(1-x_in)-m(1)*(1-x(1))+ Qcell/h_vap(1);
    f3(1) = Vl(1) -(n_l(1)*R*T(1)*Z_l(1))/P(1);
    f4(1) = Vg(1) - (Vcell-Vl(1));
    f5(1) = P(1) - n_g(1)*R*T(1)*Z_g(1)/(Vg(1));
    f6(1) = T(1) - (B/(A-log10(P(1)))- C);
    f7(1) = x(1) - n_l(1)/(n_l(1)+n_g(1));
    f8(1) = m(1) -k*sqrt(max(P(1)-P(2),0));
    f9(1) = U(1) + P(1)*Vcell - n_l(1)*h_l(1)-n_g(1)*h_g(1);
    f10(1) = Vm_g(1) - Z_g(1)*T(1)*R/(P(1));
    f11(1) = Vm_l(1) - Z_l(1)*T(1)*R/(P(1));
    f12(1) = Z_g(1)^3 - Z_g(1)^2*(1-B_PR(1))+...
                 Z_g(1)*(A_PR(1)-2*B_PR(1)-3*B_PR(1)^2)-...
                 (A_PR(1)*B_PR(1)- B_PR(1)^2 - B_PR(1)^3);
    f13(1) = Z_l(1)^3 - Z_l(1)^2*(1-B_PR(1))+...
                Z_l(1)*(A_PR(1)-2*B_PR(1)-3*B_PR(1)^2)-...
                (A_PR(1)*B_PR(1)- B_PR(1)^2 - B_PR(1)^3);

    %Equations for all other nodes
    for i = 2:N
        f1(i) = m(i-1)*x(i-1)-m(i)*x(i) - Qcell/h_vap(i);
        f2(i) = m(i-1)*(1-x(i-1))- m(i)*(1-x(i))+ Qcell/h_vap(i);
        f3(i) = Vl(i) -(n_l(i)*R*T(i)*Z_l(i))/P(i);
        f4(i) = Vg(i) - (Vcell-Vl(i));
        f5(i) = P(i) - n_g(i)*Z_g(i)*R*T(i)/(Vg(i));
        f6(i) = T(i) - (B/(A-log10(P(i)))- C);
        f7(i) = x(i)- n_l(i)/(n_l(i)+n_g(i));
        f9(i) = U(i) + P(i)*Vcell - n_l(i)*h_l(i)-n_g(i)*h_g(i);
        f10(i) = Vm_g(i) - T(i)*Z_g(i)*R/(P(i));
        f11(i) = Vm_l(i) - T(i)*Z_l(i)*R/(P(i));
        f12(i) = Z_g(i)^3 - Z_g(i)^2*(1-B_PR(i))+...
                 Z_g(i)*(A_PR(i)-2*B_PR(i)-3*B_PR(i)^2)-...
                 (A_PR(i)*B_PR(i)- B_PR(i)^2 - B_PR(i)^3);
        f13(i) = Z_l(i)^3 - Z_l(i)^2*(1-B_PR(i))+...
                 Z_l(i)*(A_PR(i)-2*B_PR(i)-3*B_PR(i)^2)-...
                 (A_PR(i)*B_PR(i)- B_PR(i)^2 - B_PR(i)^3);

        if i == N   %Different molar flow equations for last node
            f8(i) = m(i) -k*sqrt(max(P(i)-P_out,0));
            break
        end
        f8(i) = m(i) -k*sqrt(max(P(i)-P(i+1),0));
    end

end


%Equations for gas phase stream scenario
if (const.scenario == 2)

    %Eqauation for the first node
    f1(1) = n_l(1) - 0;
    f2(1) = n_g(1) - (Vcell/Vm_g(1));
    f3(1) = Vl(1) - 0;
    f4(1) = Vg(1) - Vcell;
    f5(1) = P(1) - (P_in - dp_cell);
    f6(1) = (m_in*h_in - m(1)*h_g(1))/(n_g(1)*cp_g(1)) - Qcell/(n_g(1)*cp_g(1));
    f7(1) = x(1) -0;
    f8(1) = m(1) - m_in;
    f9(1) = U(1) + P(1)*Vcell -n_g(1)*h_g(1);
    f10(1) = Vm_g(1) - T(1)*R*Z_g(1)/(P(1));
    f11(1) = Vm_l(1) -0 ;
    f12(1) = Z_g(1)^3 - Z_g(1)^2*(1-B_PR(1))+...
                 Z_g(1)*(A_PR(1)-2*B_PR(1)-3*B_PR(1)^2)-...
                 (A_PR(1)*B_PR(1)- B_PR(1)^2 - B_PR(1)^3);
    f13(1) = Z_l(1) - 0;

    %Eqauation for all other nodes
    for i = 2:N
        f1(i) = n_l(i) -0;
        f2(i) = n_g(i) - (Vcell/Vm_g(i));
        f3(i) = Vl(i)-0;
        f4(i) = Vg(i) - Vcell;
        f5(i) = P(i) - (P(i-1) - dp_cell);
        f6(i) = (m(i-1)*h_g(i-1)-m(i)*h_g(i))/(n_g(i)*cp_g(i)) - Qcell/(n_g(i)*cp_g(i));
        f7(i) = x(i)-0;
        f8(i) = m(i) -m(i-1);
        f9(i) = U(i) + P(i)*Vcell -n_g(i)*h_g(i);
        f10(i) = Vm_g(i) - T(i)*R*Z_g(i)/(P(i));
        f11(i) = Vm_l(i) -0 ;
        f12(i) = Z_g(i)^3 - Z_g(i)^2*(1-B_PR(i))+...
                 Z_g(i)*(A_PR(i)-2*B_PR(i)-3*B_PR(i)^2)-...
                 (A_PR(i)*B_PR(i)- B_PR(i)^2 - B_PR(i)^3);
        f13(i) = Z_l(i) - 0;
    end
```

```
end


if (const.scenario == 3)

    %Eqauation for the first node
    f1(1) = n_l(1) - (Vcell/Vm_l(1));
    f2(1) = n_g(1) - 0;
    f3(1) = Vl(1) - Vcell;
    f4(1) = Vg(1) - 0;
    f5(1) = P(1) - (P_in - dp_cell);
    f6(1) = (m_in*h_in - m(1)*h_l(1))/(n_l(1)*cp_l(1)) - Qcell/(n_l(1)*cp_l(1));
    f7(1) = x(1) -1;
    f8(1) = m(1) - m_in;
    f9(1) = U(1) + P(1)*Vcell -n_l(1)*h_l(1);
    f10(1) = Vm_g(1) - 0;
    f11(1) = Vm_l(1) -(T(1)*R*Z_l(1))/(P(1)) ;
    f12(1) = Z_g(1) -0;
    f13(1) = Z_l(1)^3 - Z_l(1)^2*(1-B_PR(1))+...
                 Z_l(1)*(A_PR(1)-2*B_PR(1)-3*B_PR(1)^2)-...
                 (A_PR(1)*B_PR(1)- B_PR(1)^2 - B_PR(1)^3);
    %Eqauation for all other nodes
    for i = 2:N
        f1(i) = n_l(i) -(Vcell/Vm_l(i));
        f2(i) = n_g(i) - 0;
        f3(i) = Vl(i)- Vcell;
        f4(i) = Vg(i) - 0;
        f5(i) = P(i) - (P(i-1) - dp_cell);
        f6(i) = (m(i-1)*h_l(i-1)-m(i)*h_l(i))/(n_l(i)*cp_l(i)) - Qcell/(n_l(i)*cp_l(i));
        f7(i) = x(i)-1;
        f8(i) = m(i) -m(i-1);
        f9(i) = U(i) + P(i)*Vcell -n_l(i)*h_l(i);
        f10(i) = Vm_g(i) - 0;
        f11(i) = Vm_l(i) -(T(i)*R*Z_l(i))/(P(i)) ;
        f12(i) = Z_g(i) - 0;
        f13(i) = Z_l(i)^3 - Z_l(i)^2*(1-B_PR(i))+...
                 Z_l(i)*(A_PR(i)-2*B_PR(i)-3*B_PR(i)^2)-...
                 (A_PR(i)*B_PR(i)- B_PR(i)^2 - B_PR(i)^3);
    end

end

%Return system of equations
system = [f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13]';

end
```

Data computation

```
function [h_g,h_l,cp_g,cp_l,h_vap,A,B,C,A_PR,B_PR] = data(y,const)

N = const.N;
T = y(5*N+1:5*N+N);
P = y(4*N+1:4*N+N);

%Calculation of heat and enthalpy for both phases
[cp_g,cp_l,h_g,h_l,h_vap] = heatandenthalpy(T,P,const);
```

```
%Extraction of antonie constans
[A,B,C] = antonie(const.ind);

%Calculation of compressibility polynomial coefficients
[A_PR,B_PR] = P_R(T,P,const.ind);

end
```

Equation of state

```
function[A,B] = P_R(T,P_in,ind)
R = 8.314*10^-5;
P = P_in;
%Critical conditions and acentric factor from Nist chemistry WebBook
%"http://webbook.nist.gov/chemistry/"
Tc = [190.564 305.33 369.825];
Pc = [45.99 48.718 42.4766];
w = [ 0.01142 0.0993 0.1524];

a = (0.45724*R^2*Tc(ind)^2)/Pc(ind);
b = (0.07780*R*Tc(ind))/Pc(ind);
Tr = T/Tc(ind);
alpha = (1+(0.37464+1.54226*w(ind)-0.2699*w(ind)^2)*(1-Tr.^0.5)).^2;

A = (a.*alpha.*P)./(R^2.*T.^2);
B = (b.*P)./(R.*T);
end
```

Heat capacity and enthalpy

```
function [cp_g,cp_l,h_g,h_l,h_vap] = heatandenthalpy(T,P,const)
T_ref = 100;
ind = const.ind;
N = const.N;

%All values are calcualted regardless of
%simulation conditions
[A,B,C] = antonie(ind);
T_evap = (B./(A-log10(P))- C);
h_vap = ones(size(T));
cp_l = ones(size(T));

%Depending on the refrigerant stream different values are used
%Coefficients from [Poling et al., 2001]
%Enthalpies and liquid heat capacities from NIST chemistry WebBook
%"http://webbook.nist.gov/chemistry/"
switch ind
    case 1
        h_vap = h_vap*8170;
        cp_l =cp_l*53 ;
        A = 1.925*10^(1);
        B = 5.213*10^(-2);
        C = 1.197*10^(-5);
        D = -1.132*10^(-8);
    case 2
        h_vap = h_vap*14700;
        cp_l =cp_l*68.5 ;
        A = 5.409*10^(0);
        B = 1.781*10^(-1);
        C = -6.938*10^(-5);
        D = 8.713*10^(-9);
    case 3
        h_vap = h_vap*19040;
        cp_l =cp_l*98.3 ;
        A = -4.224*10^(0);
        B = 3.063*10^(-1);
        C = -1.586*10^(-4);
        D = 3.215*10^(-8);
end

heatCapacity_g = @(T) A + B*T + C*T.^2 + D*T.^3;
cp_g = heatCapacity_g(T);                        %Gas phase heat capacity
enthalpy_g = @(T) A*(T-T_evap)+(1/2)*B*(T.^2-T_evap.^2)+(1/3)*C*(T.^3-T_evap.^3)...
    +(1/4)*D*(T.^4-T_evap.^4);
h_l = cp_l.*(T-T_ref);                            %Liquid phase enthalpy
h_g = h_l + h_vap + enthalpy_g(T);                %Vapour phase enthalpy

end
```

Antonie constants

```
function [A,B,C] = antonie(ind)

%Constants taken from Nist chemistry WebBook
%"http://webbook.nist.gov/chemistry/"
switch ind
    case 1 % Methane
        A = 4.22061;
        B = 516.689;
        C = 11.223;
    case 2 % Ethane
        A = 3.93835;
        B = 659.739 ;
        C = -16.719 ;
    case 3 % Propane
        A = 3.98292;
        B = 819.296;
        C = -24.417;
end
```