NTNU
Norwegian University of
Science and Technology

Modelling and Fouling Monitoring of Condenser at Snøhvit

Rebecca Sian Williams

Chemical Engineering

Submission date:	June 2011
Supervisor:	Magne Hillestad, IKP
Co-supervisor:	Arne Olav Fredheim, Statoil

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This report is the result of the work done in the Master Thesis spring 2011 (TKP 4900), at The Department of Chemical Engineering, NTNU. The project title, and title for this report, is ”Modelling and fouling monitoring of a condenser at Snøhvit”. The work has been done in collaboration with Statoil ASA.

I would like to thank my supervisor, Magne Hillestad, for his help with this assignment. I would also like to thank everybody at Statoil ASA for their valuable input, especially Arne Olav Fredheim and Geir Iversen. Thanks to family and friends for backing me up this semester. Finally I would like to thank Stine Karlsen and Sigrid Lædre for valuable input and proof reading.

I declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology

Trondheim, 09.06.2011

Rebecca Williams

Rebecca Sian Williams

Contents

List of Symbols

Large lettered symbols

Symbol	Description	Unit
A	Area	[m^2]
A_1	Area for cooling	[m^2]
A_2	Area for integral condensation	[m^2]
A_3	Area for differential condensation	[m^2]
$C_{p,sw}$	heat capacity of sea water	[J/kmolK]
D_{AB}	Diffusional constant	[m/s]
J^*_{Az}	Diffusional flux	[kmol/m^2,s]
M_i	Molar mass of component i	[kg/kmol]
N_A	Total flux of A	[kmol/m^2,s]
P	Pressure	[Pa]
P_c	Critical pressure	[bar]
R	Gas constant	[kJ/kmol,K]
R	Fouling Resistance	[m^2,K/W]
T	Temperature	[K]
T_c	Critical temperature	[K]
U	Total heat transfer coefficient from condensate film to bulk cooling medium	[W/m^{2}K]
U_s	Total heat transfer coefficient for cooling	[W/m^{2}K]
V	Volume	[m^3]
$V_{M,0}$	Initial mass flow of vapour	[kg/s]
Z	compressibility factor	[-]

Dimensionless numbers

Symbol	Description
Nu	Nusselt number
Pr	Prandtl number
Re	Reynolds number

Small lettered symbols

Symbol	Description	Unit
c_A	consentration of A	[kmol/m^3]
d	diameter	[m]
h_g^0	heat transfer coefficient in vapour phase, adjusted for mass transfer	[W/m^{2}K]
$h_{v,i}$	heat of vaporisation for component i	[J/kmol]
h_{vf}	heat of vaporisation	[J/kmol]
h	film heat transfer coefficient	[W/m^{2}K]
h_s	heat transfer coefficient, shell side	[W/m^{2}K]
k_p	Constant for pressure calculation	[]
k_w	conductivity of metal	[W/mK]
l	characteristic length	[m]
m_sw	mass flow of sea water	[kg/s]
n_{rc}	number of tube rows on the centre stream line within 1 cycle	[-]
n_{rp}	number of tube rows	[-]
n_i	flux of component i	[kmol/m^2,s]
n_p	number of baffles	[-]
n_t	total flux of matter	[kmol/m^2,s]
q	heat flux	[kJ/m^2,s]
v_M	average velocity of bulk phase	[m/s]
$x_{i,I}$	molfraction of comp. I in condensate at interface	[-]
x_i	molfraction of comp. I in condensate	[-]
$y_{i,I}$	molfraction of comp. I in vapour at interface	[-]
y_i	molfraction of comp. I in vapour	[-]
z	length in z-direction	[m]

Greek letter symbols

Symbol	Description	Unit
β	helical angle	[deg]
β_{12}	binary mass transfer coefficient	[kmol/m^2,s]
ϵ	ackermann factor	[-]
λ	Conductivity in fluid	[W/m,K]
μ	viscosity	[Ns/m^2]
ω	acentric factor	[-]
ϕ	Rate factor for binary mixture	[-]
Φ	mixing parameter	[-]
ξ	normalized independent variable for collocation method	[-]
ΔT	Temperature approach	[K]
ΔT_l	log mean of terminal temperature difference	[K]

Indices

Symbol	Description
1/3	differential section (for heat transfer coefficients)
A	component A
a	sea water
e	ethane
f	fouling layer
g	based on vapour
i	for inner area
i	component i
I	interface
o	for outer area
o	based on cooling medium
p	propane
s	interface

Abstract

This report is a result of the work performed during the Master Thesis TKP 4900 at the Department of Chemical Engineering, NTNU in the spring of 2011. The problem assigned was ”Modelling and fouling monitoring of a condenser at Snøhvit”. The work was done in collaboration with Statoil ASA on a Helixchanger condenser at the LNG plant at Melkøya outside of Hammerfest, Norway. The condenser, 25-HA-112, condenses a mixture of propane and ethane by heat exchanging against sea water.

A model describing the condenser 25-HA-112, based on the film method, was built in MATLAB. The model consisted of three sections; a cooling section, a section for integral condensation and a section for differential condensation. The differential model equations describing these were approximated by a collocation routine. The equation set consisted of both algebraic and differential equations and was solved using Newton’s method. The thermodynamic model, based on Peng-Robinson equation of state, proved to be a challenge for the analysis. When changing the initial conditions for the condenser model, the thermodynamic model was unstable. This resulted in only small adjustments for each iteration could be made.

A second model, constructed in the simulation software Unisim, was used to validate the MATLAB model. The results from the two models seemed to coincide, strengthening the condenser model constructed in MATLAB.

A sensitivity analysis was carried out on the model using five parameters; vapour heat transfer coefficient, fouling resistance, vapour mass flow, sea water temperature and initial vapour composition. The results obtained in the sensitivity analysis performed as expected from theory, supporting the MATLAB model.

Experimental data obtained for 25-HA-112 at Melkøya were compared to a case model built in MATLAB. The results show that the MATLAB model does not predict a satisfactory result for the cooling and integral section of the condenser. Reasons for this deviation are discussed. The main causes are thought to be the use of a too coarse model, the assumption of a binary component model and a higher heat transfer coefficient in the real condenser due to turbulence.

Chapter 1

Introduction

The energy consumption of the world has been growing steadily for a long time, and will probably keep growing in the future. Energy is mainly supplied by fossil resources, and natural gas is vital for many countries. As natural gas is a fossil fuel, the reservoirs will one day empty. To postpone that date, the efficiency of the oil and gas industry must be high and methods for reduction of energy consumption must be developed.

At the Norwegian plant at Melkøya, Statoil ASA produces Liquid Natural Gas (LNG) by cooling the natural gas coming in from the reservoirs offshore. The liquefaction is done in a series of heat exchangers and condensers against different refrigerants. In some heat exchangers sea water is used as a cooling agent, introducing problems with fouling. Fouling is a layer of deposits on the heat exchanger surface which decreases the heat flux from one fluid to the other. This layer will increase over time, eventually making the heat exchanger inefficient.

This thesis is partly a continuation of the work done by Jordhøy and Stræte [2009]. Jordhøy and Stræte carried out two experimental studies on the condenser 25-HA-112 in their master thesis together with building a model of the condenser. This thesis will concentrate on the modelling of the condenser. The assignment given constituted of the following points:

1. A literature survey
2. Modelling of phase change (dew-point calculations)
3. Modelling the exchanger with the front of phase change and temperature profile along the condenser
4. Tuning of model against plant measurements

5. Calculate/estimate the fouling based on historical data

Due to points 3 and 4 being time consuming, the estimation of the fouling was not done. It was chosen, in collaboration with supervisor, to focus on the model and conduct a sensitivity study instead.

This thesis is built up in the following manner; Firstly, a description of the process at Snøhvit is given, before theoretical background for the work is presented. This consists of basic theory of heat and mass transfer, followed by a closer description of the condensation process, the Helixchanger, equations of state and some mathematical principals.

The next chapter constitutes of a description of the method developed for this thesis. Here the different models and equations used are described, together with how the equations were solved. The results and discussion follows before a suggestion for further work and the conclusion. Most of the MATLAB-scripts are placed in the Appendices, together with other results and some theory.

Chapter 2

Theory and Background

2.1 Process Description

The condenser 25-HA-112 investigated in this project is one of many heat exchangers installed at Statoil's Snøhvit Natural Gas Liquefaction Plant at Melkeøya outside Hammerfest in Norway. This chapter will give a brief introduction to this process and the role of 25-HA-112 in the process. An overview of the process is shown in Figure 2.1.

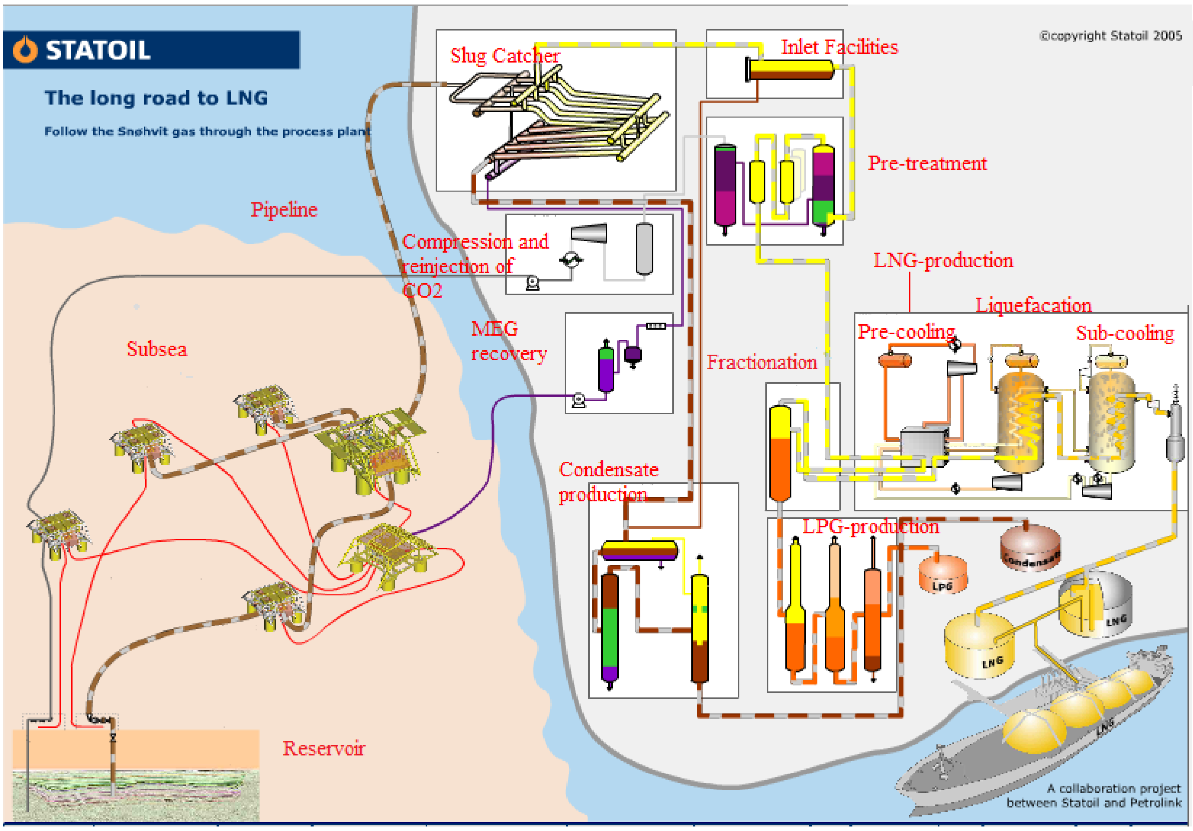


Figure 2.1: Process Description of LNG production [Statoil, 2005]

In the Barents Sea outside the northern part of Norway, lie the reservoirs Snøhvit, Albatross and Askeladd. Askeladd will be started up in 2014/2015[Statoil, 2007], but the two other reservoirs are already delivering gas into the liquefaction plant at Melkeøya through a 143 km long pipeline.

At the plant, the gas is first sent through a slug catcher. The incoming stream contains condensate, a mixture of water and Mono Ethylen Glycol (MEG), as well as the natural gas and nitrogen (N_2). The first separation into gas, condensate and water is carried out in the slug catcher. The gas is then heated up and separated from the condensate before it is sent to preconditioning. The preconditioning is carried out in three stages where CO_2, water and mercury are separated from the gas. To meet the specifications for sale of the LNG, the heavier gas components must be taken out. The separation is done in a fractionation tower. The heavy components are taken out as a bottom stream and sent to further production of Liquefied Petroleum Gases (LPG). The top stream is sent to the LNG production plant. Here it is cooled from 13 °C to minus 163 °C, by heat exchanging with a refrigerant. This is done in three stages: pre-cooling, liquefaction and sub-cooling. Before the LNG is stored in tanks and then shipped off to Europe and the

USA, some of the N_2 gas must be taken out in order to meet the sale specifications for LNG[Statoil, 2005].

The condenser 25-HA-112 is situated in the pre-cooling stage. The cooling medium in this circuit consists mainly of ethane and propane, which is used to cool the natural gas down to the dew-point. The hydrocarbon cooling medium evaporates during the cooling and is sent through a two-stage compression before entering condenser 25-HA-112. The mix is condensed by heat exchanging against sea water.

2.2 Heat Transfer

2.2.1 Basic Principles of Heat Transfer

The transport of thermal energy from one region to another is defined as heat transfer. This transport requires a temperature difference, and the heat will be transported from the region with the higher temperature to the region with the lower temperature. Heat can be transported by conduction, convection and radiation. Conduction occurs when the transport is through solids or fluids at rest. Convection is when the heat is transferred due to a moving fluid, making it an extension of conduction. The movement of the fluid adds to the conduction. The last transport mode, radiation, occurs when the thermal energy is transported by means of electromagnetic waves.[Hagen, 1999]

The mechanisms for conduction vary with the structure of the material transporting the heat[Hagen, 1999]. In liquids and vapours, energy is transported when a molecule diffuses from the high temperature area to the low temperature area and collides with other molecules. For solid materials in a crystalline structure, the conduction mechanism is the vibration of the lattice structure. For metals, the movement of free valence electrons is the primary mechanism, whereas amorphous solids transport heat by random motion.

When a fluid moves across a solid surface, convection takes place. The surface will have one temperature and the bulk of the fluid will have another temperature. Between these two points there will be a temperature gradient. In the same manner there will be a velocity difference between the bulk fluid and the surface. At the surface the velocity is zero due to viscosity effects whereas the bulk will have the free stream velocity[Hagen, 1999]. This velocity profile will influence the convective heat transfer.

Heat Transfer Coefficient

The heat transfer coefficient describes the rate of heat transfer through a fluid or a solid object. As it is not a thermal property, it cannot be looked up in tables. The coefficient depends on several factors as listed by Hagen [1999].

- *Type of fluid (liquid or gas)*
- *Flow Condition (laminar or turbulent)*
- *Forced or natural convection or phase change*
- *Free Stream velocity*
- *Surface geometry and roughness*
- *Position along the surface*
- *Temperature dependence of fluid properties*

These factors have to be taken into account when calculating the heat transfer coefficient and several correlations have been developed for different conditions.

The Total Heat Transfer Coefficient

The total heat transfer coefficient, U, is built up of several elements and describes the joint heat transfer rate through several layers of fluids and solids. In addition to the different heat transfer coefficients, it depends on the surface area, and different equations are used for different geometries. For a tube, the Equation given in 2.1, is used to calculate the coefficient[Sinnot, 2005].

$$U_o = \frac{1}{\frac{1}{h_o} + \frac{d_o \ln\left(\frac{d_o}{d_i}\right)}{2k_w} + \frac{d_o}{d_i} \cdot \frac{1}{h_i}} \tag{2.1}$$

In Equation 2.1, the total coefficient is based on the outside area of the tube, hence the subscript ”o”. The subscript ”i” refers to the inside parameters. h_o and h_i are, respectively, the outside and inside fluid film heat transfer coefficient. d_i and d_o are the inside and outside diameter of the tube and k_w is the thermal conductivity of the tube wall material.

Fouling

Fouling is a common phenomenon when a material comes in contact with sea water. It is a build up of a film between the surface and the water[Nebot et al., 2007], and is usually one of three types: biological, corrosion or precipitation fouling[Characklis and Marshall, 1990] (as cited by Nebot et al. [2007]). The extra layer adds a new resistance to the heat transfer between hot and cold side. This extra resistance is found in the overall heat transfer coefficient, U, as shown in Equation 2.2. When comparing this equation with Equation 2.1, the extra parts accounting for the fouling, $h_{f,o}$ and $h_{f,i}$, are identified.

$$U_o = \frac{1}{\frac{1}{h_o} + \frac{d_o \ln\left(\frac{d_o}{d_i}\right)}{2k_w} + \frac{d_o}{d_i} \cdot \frac{1}{h_i} + \frac{d_o}{d_i} \cdot \frac{1}{h_{f,i}} + \frac{1}{h_{f,o}}} \tag{2.2}$$

The fouling resistances for the outer and inner, respectively $\frac{1}{h_{f,o}}$ and $\frac{1}{h_{f,i}}$, are time dependent parameters, and will increase with time as the exchanger gets more and more fouled. This results in a decrease of U_o with time and as a consequence of this, the amount of heat that is transported between the streams will decrease.

For heat exchangers using water at a temperature range not far from the natural temperature ranges, biological fouling can be a problem[Bott, 1995]. There are several parameters that affect the growth and deposition of living matter on the surfaces of heat exchangers. The biofilm thickness has a peak for water velocity around 1 m/s[Bott, 1995, p. 243]. A lower velocity results in a high mass transfer resistance of nutrients to the bacteria on the surface, whereas a higher velocity results in higher shear forces, hindering the biofilm to develop further. The temperature can also affect the growth, as all micro-organisms have an optimum temperature range for growth[Bott, 1995], hence the growth is usually higher in the summer months. Other effects that will induce biofilm growth is pH-neutrality, trace elements and an absent of suspended solids. The surface material will also affect the growth[Bott, 1995]. The growth of a fouling layer can be slowed down by adding antifouling agents like biocides [Nebot et al., 2007]. A popular choice is chlorine as it is both cheap and effective[Characklis, 1991] (as cited by Nebot et al. [2007]).

The condenser 25-HA-112 has suffered from fouling and Pictures 2.2 and 2.3 taken of the warm and cold end of the condenser confirms this.

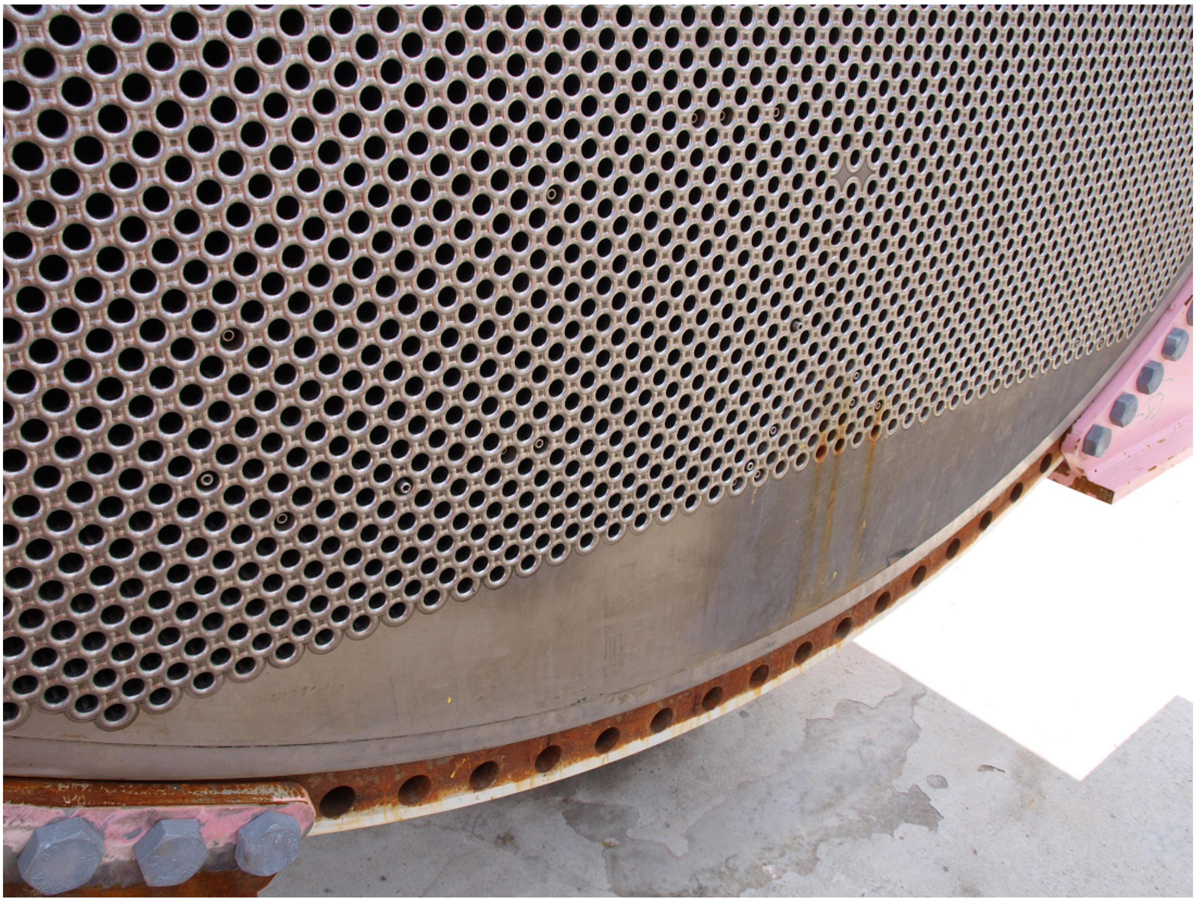


Figure 2.2: Picture of sea water inlet (cold end) for 25-HA-112[Iversen, 2011]

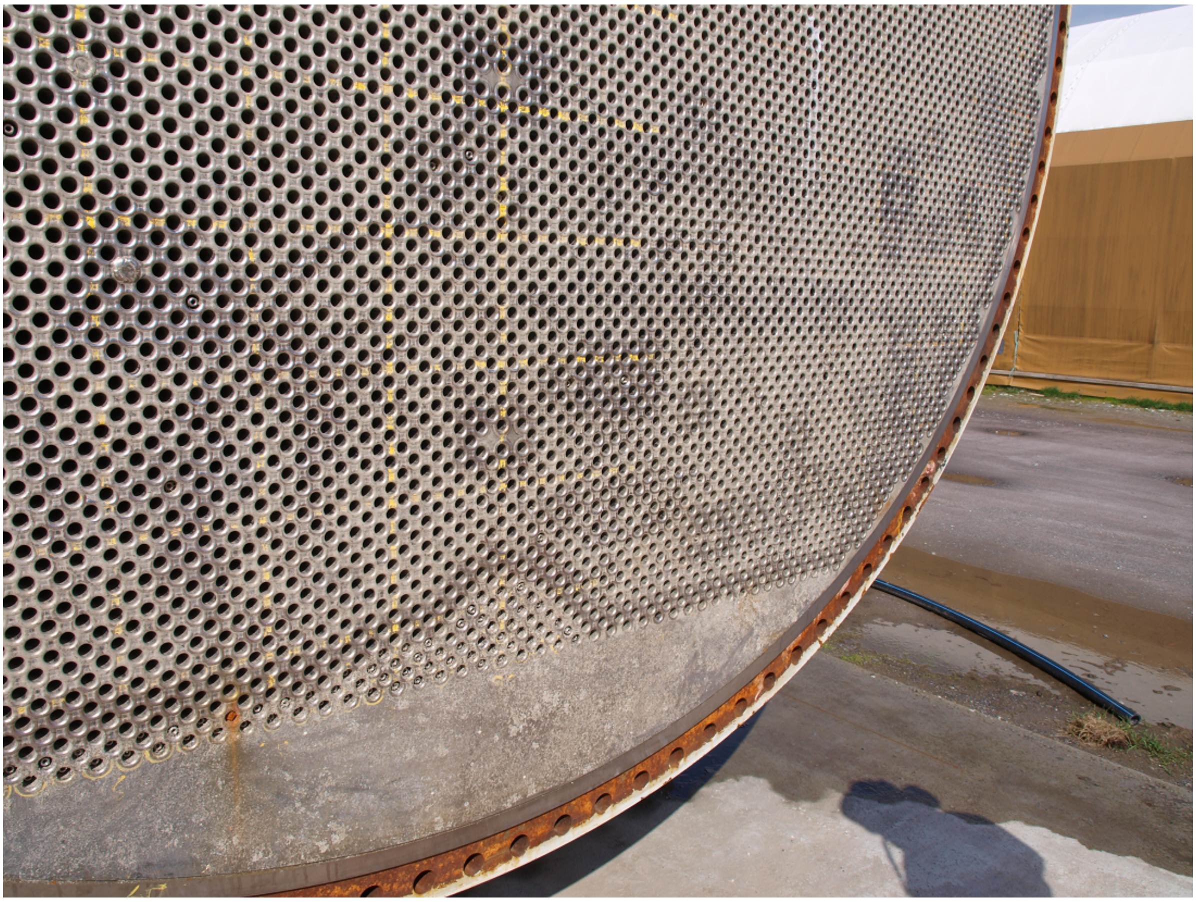


Figure 2.3: Picture of sea water outlet (warm end) for 25-HA-112[Iversen, 2011]

2.3 Mass Transfer

Mass transfer is a fundamental transfer process in the same manner as heat transfer. The transfer of mass can be by diffusion or by convection. Diffusional transfer of matter can be described by Fick's law as in Equation 2.3.

$$J^*_{Az} = -D_{AB}\frac{dc_A}{dz} \tag{2.3}$$

If a fluid is agitated, the mass transfer can be increased due to convective mass transfer. The total flux is thus represented by Equation 2.4 where v_M is the average velocity of the bulk phase, and two mechanisms can control the mass transfer.

$$N_A = J^*_{Az} + c_A v_M \tag{2.4}$$

2.3.1 Mass Transfer in Condensation

In condensation there will occur mass transfer between the two phases, in addition to mass transfer within the each phase. The mass resistance in the different phases will give rise to the different rates, where the slowest rate will be rate determining. The resistance in the vapour phase is usually considered as a stagnant gas film, enabling the use of the Chilton-Colburn analogy[Chilton and Colburn, 1934] (as cited by Karlsson and Vamling [2005b]). The liquid is usually assumed to have no mass resistance, but as Karlsson and Vamling [2005b] points out, this can be of importance in some cases. The liquid flow is assumed to be laminar, allowing only diffusional transport in the condensate. Thus Fick's law can be used.

2.4 Condensation

Condensation is the process where a substance in vapour phase goes over to liquid phase, as shown in Equation 2.5, and occurs when the vapour temperature is below the saturation temperature[Hagen, 1999].

$$A(g) \rightarrow A(l) \tag{2.5}$$

In a condenser the warm vapour is in direct contact with the cool surface and condensate is formed on the surface through surface condensation[Hagen, 1999]. This type of condensation is either dropwise or filmwise, where the difference between the two lies in how the condensate droplets behave. For filmwise condensation the droplets spread out and form a film covering the surface. This is the normal condensation mode in commercial condensers[Sinnot, 2005]. The droplets in dropwise condensation remain intact, giving a higher heat transfer coefficient, but they are too unstable for commercial use[Hagen, 1999].

When the condensing vapour is a binary mixture, several additional features must be considered[Sinnot, 2005];

1. The heavier components condensate out first, changing the composition and dewpoint. The process is thus not isothermal and the mixture is called a zeotropic mixture.

2. Both the vapour and condensate will liberate sensible heat when being cooled to the dew-point and the outlet temperature, respectively.

3. The physical properties of the vapour and condensate vary throughout the condenser due to varying composition.

4. The rate of condensation is dependent on the rate of heat transfer and the rate of diffusion, as the heavy component must diffuse through the lighter component to reach the surface.

2.4.1 Integral and Differential condensation

A further classification of condensation can be done by assessing the liquid-flow pattern[Sinnot, 2005]. The two limiting cases are integral and differential condensation. For integral condensation it is assumed that the condensate and the vapour are in equilibrium with each other[Marriot, 1989]. When the condensation is operating in differential mode, the vapour and condensate bulk are not in equilibrium with each other. This is a consequence of the liquid no longer being in contact with the vapour[Marriot, 1989], for instance when the condensate is separated from the vapour. This separation can happen when the flow velocity in the condenser drops to a level where the gravity separates the phases[Pettersen et al., 2010].

2.4.2 Modelling of Condensation

Much work has been done to model the condensation process. There are many factors that affect the process, making the problem complex. The vapour molecules have to diffuse through the gas film to the interface where they condense. The heat flux from the gas bulk to the cooling medium must also be considered, and will not be constant throughout the path. This can be seen in Figure 2.4. The heat flux from the gas bulk, $\dot{q}_g$, to the interface is given by Equation 2.6, whereas the heat flux to the cooling medium, $\dot{q}_o$, is given by Equation 2.7. To equal these two, the heat of condensation must be added to $\dot{q}_g$, as shown in Equation 2.8.

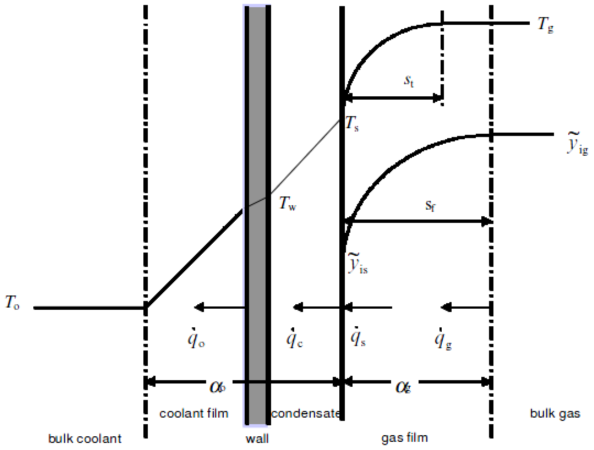


Figure 2.4: Illustration of concentration and temperature profiles in a cross-sectional area of a condenser[Webb et al., 1999]

$$\dot{q}_g = h_g^0 \left(T_g - T_s\right) \tag{2.6}$$

$$\dot{q}_o = U \left(T_s - T_a\right) \tag{2.7}$$

$$\dot{q}_o = h_g^0 \left(T_g - T_s\right) + \dot{n}_t \Delta h_{vf} \tag{2.8}$$

Colburn and Hougen [1934] outlined a method for obtaining the condenser surface. This method uses trial and error to obtain the right T_s in Equations 2.7 and 2.8 so that they equal. The q_o found is equal to UΔT. To obtain the correct area, Equation 2.9 is used. The q_o is plotted against $1/U\Delta T$, and the area is found by computing the area under the curve. By using this method, the variations in U are taken care of and a greater accuracy is obtained than by using the mean temperatures at the outlets. This offered a great improvement at the time. However, as pointed out in the same article, the cooling of the condensate is not considered in this method. The heat transferred to the cooling medium is greater than that calculated from this method. The error was calculated to be 4.5 per cent for the point values of $1/U\Delta T$, in the example presented in Colburn and Hougen [1934]. When calculating the total area, the condensate is taken into account, and thus only a small error is introduced. This method is usually referred to as the film-method

$$A = \frac{\dot{q}_0}{U\Delta T_m} \tag{2.9}$$

The competing method to Colburn's is the method introduced by Silver, also known as the equilibrium method. This is an approximate method and is described by Silver [1947] and further modified by Bell and Ghaly [1972]. The method does not calculate the detailed mass transfer equations. Instead, the sensible heat change of the saturated gas mixture is assumed to be transferred through the gas film[Webb et al., 1996]. The equilibrium method was also investigated by Krishna et al. [1976], and it was found to give a small overdesign. As the equilibrium method is not based on a genuine theory for simultaneous mass and heat transfer, this result is deemed a chance of luck.

Krishna et al. [1976], described a design procedure for multi-component mixture condensation where the differential equations and the rate equations are combined. The method takes into consideration the effects of diffusional interaction. This is done by introducing the Maxwell-Stephan equations in the mass transfer. These can be solved by using the multi-component film model of Krishna and Standart[Krishna et al., 1976]. The work showed that by neglecting diffusional interactions, the condenser area might be highly underdesigned.

The two methods have been compared by several people and adjustments have been made. McNaught pointed out that the equilibrium method of Silver is inaccurate when the inlet vapour is superheated. The vapour closest to the wall will have a lower temperature than the gas bulk. By adding a modification, condensation can occur when the wall temperature falls below the dew-point[McNaught, 1981] (as cited by Webb and Kim [1999]). Even with this correction factor, the equilibrium method is less accurate than the film model. However, the equilibrium model is often preferred as it requires less physical data and computational effort. Webb et al. [1996] investigated the relationship between the two models and suggest a new way of correcting the equilibrium approach. This correction brings the equilibrium approach closer to the film method. The Lewis number, given in Equation 2.10, influences the relationship between temperature and composition in the vapour. At Lewis numbers below 1, the two methods are in agreement. However, with Lewis numbers of 1 and higher, the equilibrium method becomes unsafe when predicting the heat transfer coefficient. Errors up to 50% have been reported[Webb et al., 1996], leading to condensers with too large surfaces[Webb and Kim, 1999, Webb et al., 1999].

$$Le = \frac{Sc}{Pr} \tag{2.10}$$

Karlsson and Vamling [2005a,b] and Sajjan et al. [2004] present a calculation procedure for describing the profiles through a condenser. This is based on the film model and can handle both integral and differential condensation. This is done by introducing the relationship shown in Equation 2.11, where Φ is the degree of mixing and "int" and "diff" indicate the molfraction in integral or differential conditions. They have found that the mixing parameter Φ is highly influential when calculating the performance of a condenser. A correction factor ϵ is also introduced to all heat transfer coefficients to make sure that the outlet conditions are met.

$$x_{i,I} = \Phi x_i^{\text{int}} + (1 - \Phi) x_i^{\text{diff}} \tag{2.11}$$

2.5 Shell and Tube Heat Exchangers with Helical Baffles

A common heat exchanger in many industrial processes is the shell and tube heat exchanger. This consists of a shell with several tubes passing through it. One stream is led through the tubes whereas the other stream passes on the outside on the shell side. To increase the surface area, and thus increase the efficiency of heat exchanging, the tubes are made small and the number of tubes is increased to handle the large streams.

The most common design for shell and tube exchangers has segmental baffles[Zhang et al., 2010]. These baffles force the fluid on the shell side into a zigzag pattern. The higher pressure drop of this configuration is compensated by a higher heat transfer. In addition to the high pressure drop, there are two other significant drawbacks with segmental baffles. 1) Between two adjacent baffles there is a dead zone which increases the fouling. 2) The zigzag pattern can cause vibration failure on the long, unsupported tubes in the tube bundles[Zhang et al., 2010]. A picture of a tube with helical baffles is shown in Figure 2.5.

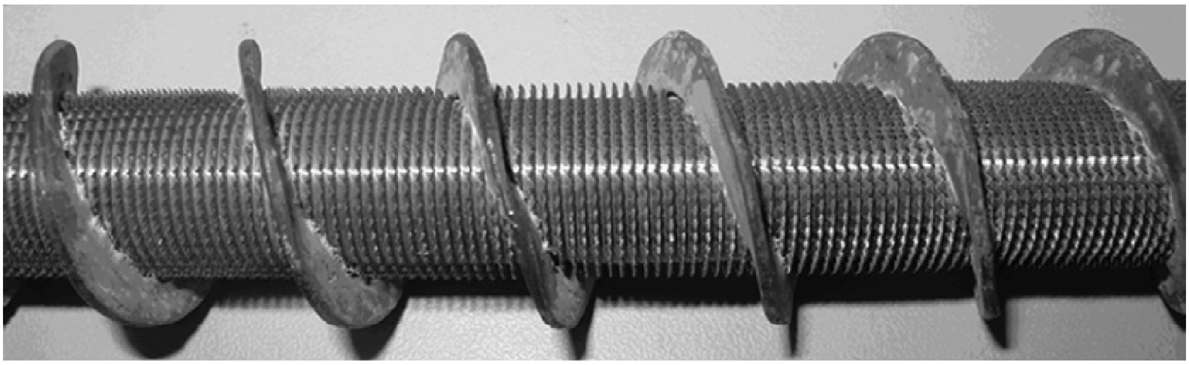


Figure 2.5: Picture of Helixchanger[Zhang et al., 2008]

A new baffle design, consisting of helical baffles, was introduced by Lutcha and Nemcansky [1990] (as cited by Zhang et al. [2010]). This new configuration increases the average temperature driving force by forcing the shell side fluid to approach plug flow with a low degree of back mixing[Zhang et al., 2008]. The design gives a lower pressure drop compared with the conventional shell-and-tube exchanger with segmental baffles.

2.5.1 Correlations for Helixchangers

Much work is done on the heat transfer coefficients for conventional shell-and-tube heat exchangers with segmental baffles, but correlations for helixchangers are not in abundance. Zhang et al. [2010] present correlations for heat transfer coefficients and pressure drop in helixchangers. These are based on the work of Stehlik et al. [1994]. The correlations for heat transfer are presented here as they are presented by Zhang et al. [2010].

Heat Transfer Coefficient for Shell side

The heat transfer coefficient for the shell side is given by Equation 2.13. Equations 2.13 to 2.17 are applicable for 10<Re<10^6, 10<Pr<10^3, n_{re} >10 and 5°≤ β ≥45°. β is the helix angle and n_{rc} is given by Equation 2.12, where n_{rp} is the number of tube rows in the cross section of heat exchanger and n_p is the number of baffles.

$$n_{rc} = n_{rp}(n_p - 1) \tag{2.12}$$

$$h_S = \frac{Nu_S \lambda_S}{l} \tag{2.13}$$

Nu$_S$ is the average Nusselt number given by Equation 2.15, λ_S is the thermal conductivity of the shell side fluid and l is the characteristic length of the shell side. l is given by Equation 2.14.

$$l = \frac{\pi d_o}{2} \tag{2.14}$$

$$\mathrm{Nu_S} = 0.62 \cdot \left(0.3 + \sqrt{\mathrm{Nu}_{lam}^2 + \mathrm{Nu}_{turb}^2}\right) \cdot Y_2 \cdot Y_3 \cdot Y_4 \cdot Y_7 \cdot Y_8 \cdot Y_9 \cdot Y_{10} \tag{2.15}$$

where Nu$_{lam}$ and Nu$_{turb}$ are defined as follows

$$\mathrm{Nu}_{lam} = 0.664 \cdot \mathrm{Re}^{0.5} \cdot \mathrm{Pr}^{0.33} \tag{2.16}$$

$$\mathrm{Nu}_{turb} = \frac{0.037 \cdot \mathrm{Re}^{0.7} \cdot \mathrm{Pr}}{1 + 2.433 \cdot \mathrm{Re}^{-0.1}\left(\mathrm{Pr}^{0.67} - 1\right)} \tag{2.17}$$

In Equation 2.15, the Y_i are correction factors. Y_2 corrects for thermal-physical property effects. Y_3 accounts for scale-up to a tube bundle from a single tube row. Y_4 corrects for the unfavourable temperature gradient. Y_7 accounts for bypass streams. Y_8 accounts for baffle spacing in the inlet and outlet section. Y_9 accounts for the cross-flow in the heat exchanger and Y_{10} accounts for turbulent enhancement. Equations for calculating Y_2 to Y_{10} are given in Appendix B.

2.6 Equation of State

To describe the thermodynamics of a system, an Equation Of State (EOS) can be used. This predicts how the pressure of a fluid depends on the volume, temperature and composition of the fluid[Skogestad, 2003]. The ideal gas law, though simple to work with, does not describe phase transitions and is thus not adequate for real fluids. In these instances a cubic EOS is a better option. The name cubic derives from the form the compressibility factor obtains, as shown in Equation 2.18.

$$Z^3 + (1 - B)Z^2 + (A - 3B^2 - 2B)Z + (-AB + B^2 + B^3) = 0 \tag{2.18}$$

2.6.1 Peng-Robinson Equation of State

The Peng-Robinson EOS (PR EOS) is one of the cubic EOS derived from Van der Waals EOS. The equation was to satisfy four points suggested by Peng and Robinson [1976]. The four points are summarized as follows by Nasri and Binous [2009]:

1. *Parameters of this EOS should be defined in terms of the critical properties and the acentric factor.*
2. *Reasonable accuracy near the critical point, particularly for calculations of the compressibility factor and liquid density.*
3. *A single binary interaction parameter, which should be independent of temperature, pressure and composition, is needed for the mixing rules.*
4. *PR EOS should be applicable in natural gas processes.*

The Peng-Robinson Equation of State is shown in Equation 2.19 [Peng and Robinson, 1976, Nasri and Binous, 2009]

$$P = \frac{RT}{(V-b)} - \frac{a}{[V\,(V+b) + b\,(V-b)]} \tag{2.19}$$

The constants a and b are dependent on the critical values, as well as the acentric factor and the temperature, and are defined in Equations 2.20 to 2.23.

$$b = 0.07780\frac{RT_c}{P_c} \tag{2.20}$$

$$a = 0.45724\frac{(RT_c)^2}{P_c}\left[1 + m\left(1 - \sqrt{T_r}\right)\right]^2 \tag{2.21}$$

$$T_r = \frac{T}{T_c} \tag{2.22}$$

$$m = 0.37464 + 1.54226\omega - 0.26992\omega^2 \tag{2.23}$$

2.7 Mathematical Background

This section gives a short presentation of some mathematical theorems and methods that are used in this report.

Collocation

The collocation method is one of the most popular weighted residual functions[Jakobsen, 2008] and can be used to solve differential equations numerically. The equations are discretisation in the independent value, for instance the length or time. By this the differential equations are approximated by algebraic equations. For each discrete point a residual is calculated and is forced to pass through zero. There are as many collocation points as residuals, and a system of equations for a matrix is obtained. This system can be solved by using Newton's method, explained later in this section.

By choosing a normalised independent variable ξ, $y(\xi)$ can be calculated according to Equation 2.24. The first and second derivative are calculated as in Equations 2.26 and 2.27. The coefficients A$_i$ and B$_i$ are the matrix of first and second derivative weights, respectively.

$$y(\xi) = \sum_{i=1}^{n+2} y_i \phi_i(\xi) \tag{2.24}$$

$$\phi_i(\xi) = \prod_{k=1,k\neq i}^{n+2} \frac{(\xi - \xi_k)}{(\xi_i - \xi_k)} \tag{2.25}$$

$$\left(\frac{dy}{d\xi}\right)_{\xi_i} = \sum_{i=1}^{n+2} A_{i,j} y_j \tag{2.26}$$

$$\left(\frac{d^2y}{d\xi^2}\right)_{\xi_i} = \sum_{i=1}^{n+2} B_{i,j} y_j \tag{2.27}$$

If the independent variable spannes a domain larger than one, the domain must be normalised as shown in the following equations. Equation 2.28 presents a system of differential equations.

$$\frac{dX}{dA} = f(X) \tag{2.28}$$

The normalized variable ξ is given by Equation 2.29.

$$\xi = \frac{A}{A^*} \tag{2.29}$$

Equation 2.28 can be written as Equation 2.30

$$\frac{dX}{d\xi} = A^* \cdot f(X) \tag{2.30}$$

The individual equations are thus approximated as in Equation 2.31. The state is named x_i whereas x_j is the collocation point.

$$\sum_{j=1}^{n+2} A_{ij} x_j = A^* f(x_i) \tag{2.31}$$

Newton's Method

Newton's method is a method for finding a root, x^*, of a function f(x). f(x^*) is replaced by a linear function, given by Equation 2.32, at an approximation x^0. The new approximation, x^1, is taken as the root of l. This is repeated until the difference between two succeeding approximations is smaller than a certain pre-defined value. The iteration given by Equation 2.33 is thus obtained[Ortega and Rheinboldt, 2000].

$$l(x) = \alpha(x - x^0) + f(x^0) \tag{2.32}$$

$$x^{k+1} = x^k - \alpha^{-1} f(x^k) \tag{2.33}$$

To expand to a multi-variable system the one-dimensional variable α with the matrix A. l(x) and f(x) are replaced with the vectors L and F.

Chapter 3

Methods

3.1 Unisim-model

To get an understanding of the system and how the condenser is built up, a model (HEX1-system.usc) was built in the simulation software Unisim. The condenser was built up as a set of heat exchangers, separators and mixers, as shown in Figure 3.1. The vapour and sea water stream was defined according to Tables 3.1 and 3.2.

Table 3.1: Stream definitions for Unisim model

Stream Name	1	sw1
Vapour / Phase Fraction	1	0
Temperature [C]	31.2	6.1
Pressure [kPa]	2074	0.914129711
Mass Flow [kg/h]	1721880	18997164

Table 3.2: Composition defined for stream 1 in Unisim model

	Molfractions
Ethane	0.64
Propane	0.36
H2O	0

The sea water was first sent through a pump, P-100, to get the right pressure, before entering the heat exchanger network. The vapour inlet is located on the other side, and passes firstly through a cooling section. The outlet was specified by setting the vapour fraction to 1, indicating that the vapour should be at the dew point at the outlet.

The next heat exchanger, Integral condenser, represents the integral condensation zone. The outlet from this heat exchanger is set to have a temperature equal to 12.03 °C, as found by Jordhøy and Stræte [2009]. The outlet of the integral condenser represents the separation point within the condenser. The stream, now having a vapour fraction of 0.1316, is sent to a separator. The vapour is sent on to the next heat exchanger, Differential condenser - 1, were it is further cooled down. It was chosen to cool the vapour 0.5 °C in the heat exchanger. The stream, now containing 97 percent vapour, was sent to a new separator. The vapour was sent on to Differential condenser - 2, whereas the condensate was mixed with the condensate out of the Integral condenser. This procedure is repeated until the vapour out of Differential condenser - 4 (DC-4)reaches 10 °C. The vapour and condensate are then mixed together in the mixer MIX-103, where the temperature is raised to 11.76 °C. This temperature rise is due to the direct condensation that occurs because of the difference in chemical potential. The vapour remaining from DC-4 is condensed during the mixing and the stream out of MIX-103 contains only liquid. To get the condensate down to the outlet temperature, the heat exchanger ”Condensate cooler” is added.

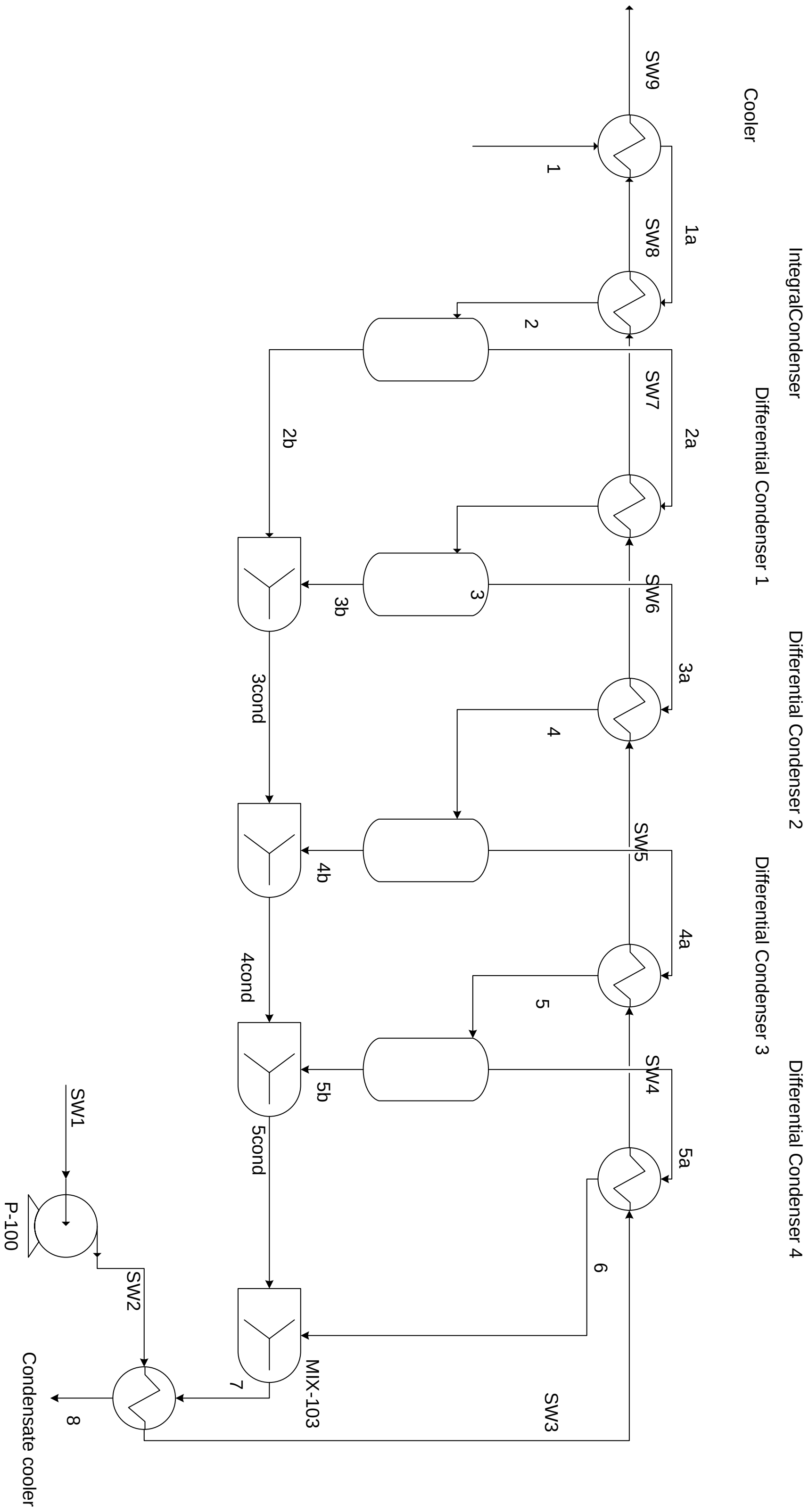


Figure 3.1: Model built in Unisim

3.2 Equations for Condensation Model

This section presents the equations used in this thesis for describing the condenser. The condenser is divided into several sections, based on the different cooling and condensation processes.

1. Cooling section
2. Integral condensation
3. Differential condensation

According to Zhang et al. [2008] the shell side flow will be close to plug flow, thus the plug flow transport equations are used for modelling temperatures and composition. The sea water in the tubes is also modelled as a plug flow. This will be discussed further in Section 5.1.

3.2.1 Cooling section

The hydrocarbon vapour enters the condenser slightly above the dew-point, resulting in the first section cooling the vapour down to the dew-point for the mixture at the given pressure. The dew-point is found by the program ”dewpoint.m” based on Peng-Robinson EOS. For temperatures above the dew-point, the following equations are employed.

$$\frac{dT}{dA} = -\frac{U_s\left(T - T_a\right)}{n_{tot}C_P} \tag{3.1}$$

$$\frac{dP}{dA} = -k_p \tag{3.2}$$

$$\frac{dn_i^j}{dA} = 0 \tag{3.3}$$

$$\frac{dT}{dA} = -\frac{U_s\left(T - T_a\right)}{m_{sw}C_{p,sw}} \tag{3.4}$$

As shown in Equation 3.3, there is no mol flux from vapour to condensate in the cooling section. When the vapour reaches the dew point, the condensation starts, and the process is described by integral condensation.

3.2.2 Integral Condensation section

The equations describing the process in the integral section are based on the work of Webb et al. [1996, 1999], Botsch and Stephan [1997] and Karlsson and Vamling [2005a].

The condensation rate for component i is described by Equation 3.5 and is a combination of the diffusive flux and the convective flux. For Equation 3.5, the assumption of binary vapour is applied to solve the diffusive flux. As there is only one independent equation for the component diffusive flux across the interface, a second equation is needed to calculate the total flux. This can be the energy balance over the interface, given by Equation 3.7. This equation equals the heat transferred to the cooling medium with the heat needed to cool the vapour. Pursuant to Webb et al. [1996], the heat of subcooling is small and to neglect this part in the energy balance provides only a small error. However, according to Karlsson and Vamling [2005a], the cooling of vapour usually accounts for less than 4 % of the heat flux whereas the sub-cooling of condensate might be more important. Thus a new energy balance was derived, Equation 3.9. In this thesis Equation 3.9 is used to calculate the total mol flux, as this was found to work best.

$$\dot{n}_i = \beta_{12} \frac{\phi}{\exp(\phi) - 1} \left(y_i - y_{i,I}\right) + y_i \dot{n}_T \tag{3.5}$$

where ϕ is given by Equation 3.6

$$\phi = \frac{\dot{n}_T}{\beta_{12}} \tag{3.6}$$

$$\dot{n}_T = \frac{U\left(T_s - T_a\right)}{\Delta h_{vf} + \frac{c_{Pg}(T_g - T_I)}{1 - e^{-\varepsilon}}} \tag{3.7}$$

where ε is given by Equation 3.8

$$\varepsilon = \frac{\dot{n}_T c_{Pg}}{h_g} \tag{3.8}$$

$$\dot{n}_{T,n} = \frac{U\left(T_s - T_a\right) - c_{p,l} \cdot N_l \cdot \frac{(T_{s,n-1} - T_{s,n})}{dA}}{\Delta h_{vf}} \tag{3.9}$$

The last flux is calculated through Equation 3.10

$$\dot{n}_j = \dot{n}_T - \dot{n}_i \tag{3.10}$$

The heat of condensation, Δh_{vf} is calculated from the individual mass fluxes and heats of condensation together with the total condensation flux as in Equation 3.11.

$$\Delta h_{vf} = \frac{\dot{n}_e h_{v,e} + \dot{n}_p h_{v,p}}{\dot{n}_T} \tag{3.11}$$

The integral condensation is defined as having a fully mixed condensate film. This implies that the molfraction at the interface equals the molfraction of the condensate bulk. When assuming that the condensate and the vapour are in equilibrium with each other, the gas fraction, $y_{i,I}$, and the temperature at the interface, T_I, can be calculated from the EOS. The coefficient β is calculated according to Equation 3.24 in section 3.4.

To describe the condenser the following differential equations are used:

$$\frac{dN_E^v}{dA} = -\dot{n}_E \tag{3.12}$$

$$\frac{dT}{dA} = -\frac{h_{g0}\left(T_g - T_s\right)}{N_g c_{Pg}} \tag{3.13}$$

$$\frac{dT_a}{dA} = -\frac{U\left(T_s - T_a\right)}{m_{sw} c_{Psw}} \tag{3.14}$$

where h_{g0} is the heat transfer coefficient on the gas film side, given in Equation 3.15. The coefficient is corrected for the condensation by the Ackermann coeficienent as given in Equation 3.8

$$h_{g0} = h_g \frac{\varepsilon}{e^{\varepsilon} - 1} \tag{3.15}$$

3.2.3 Differential Condensation

When assuming differential condensation, the condensate is considered unmixed. This implies that the condensate molfraction at the interface is given by the fluxes arriving at the given position. The total condensate flux can be given by Equation 3.16

$$\dot{n}_t = -\beta_{12} \ln \left(\frac{x_{i,I} - y_i}{x_{i,I} - y_{i,I}} \right) \tag{3.16}$$

To obtain the different molfractions at the interface, the interfacial temperature must be calculated. This can be done by employing the procedure of Sajjan et al. [2004], modified by Jordhøy and Stræte [2009]. Jordhøy and Stræte used two equations for the total heat flux, 3.17 and 3.18, and iterated on the interface temperature, T$_I$, until these were equal.

$$q_1 = h_{g0} \left(T_g - T_I \right) + \sum \dot{n}_i h_{fg,i} \tag{3.17}$$

$$q_2 = U \left(T_a - T_I \right) \tag{3.18}$$

With the interfacial temperature calculated, the condensation fluxes can also be calculated, as given in equation 3.19

$$\dot{n}_i = x_{i,I} \dot{n}_t \tag{3.19}$$

However, the same equations as in the integral condensation can be used. If these are to be used, the amount of mols in the condensate, N_E^l and N_P^l, must be defined as n_e and n_p. The latter procedure was chosen in this thesis.

3.2.4 Choice of heat transfer coefficient

The heat transfer coefficients for the model were chosen based on previous work by Jordhøy and Stræte. The vapour side heat transfer coefficient was set to 300 W/m^{2}K, as this was close to the heat transfer coefficient found in Jordhøy and Stræte [2009]. The condensate film heat transfer coefficient was set to 1151.4 W/m^{2}K based on the same work by Jordhøy and Stræte. The total heat transfer coefficient U, describing the heat flux from the interface to the bulk of the sea water, was calculated to 709,5

W/m^{2}K based on the specifications in the data sheet[Linde AG, 2005]. The heat transfer coefficient for the cooling section was set to 500 W/m^{2}K.

3.3 Thermodynamic Model

The Thermodynamic model used in this thesis is based on the code by Nasri and Binous [2009] for the Peng-Robinson EOS. The code is given in Appendix A.1. The PR EOS requires the critical pressure and temperature in addition to the acentric factor for propane and ethane. These parameters are given in Table 3.3[Matschke and Thodos, 1962]. The interaction parameter was assumed to be equal to zero.

Parameter	Unit	Ethane	Propane
T$_c$	[K]	305.32	369.83
P$_c$	[bar]	48.72	42.48
ω	[-]	0.099	0.152

Table 3.3: Parameters for the Peng-Robinson EOS[Matschke and Thodos, 1962]

The program ”vlecurve.m” calculates the bubble and dew point for different temperatures at a given pressure to obtain the phase diagram. To verify that the program was adequate, the data computed through ”vlecurve.m” was compared with data collected from Unisim. The result for a pressure of 20 bar is given in Figure 3.2. The two models seem to give the same result and the Thermodynamic model was assumed accurate enough for this purpose.

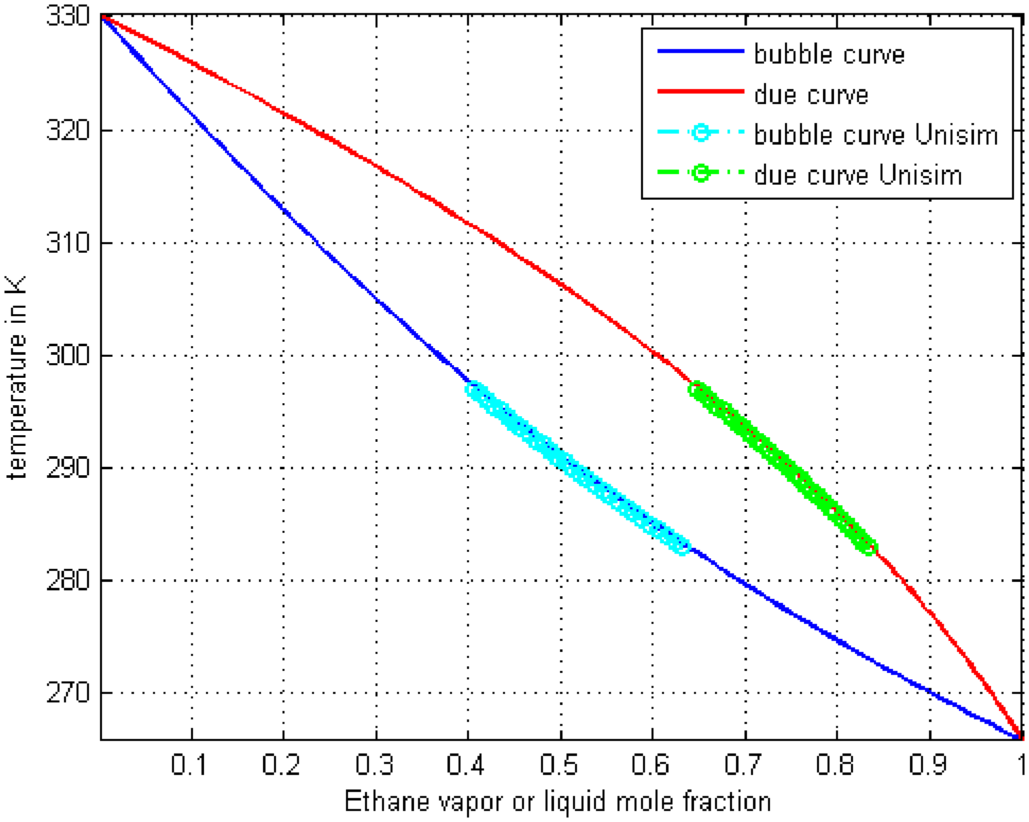


Figure 3.2: Unisim versus MATLAB

As there is a pressure drop throughout the condenser, the flash-calculations were performed for different pressures. The pressure was varied from 20 bar to 21 bar with a step size of 0.1 bar.

3.4 Physical Properties

This section describes how the physical properties were calculated.

Heat capacity

The heat capacity of the vapour was calculated using the routine "heatcapacity.m". This routine fits the heat capacity to Equation 3.20. The coefficients are given in Table 3.4

and are taken from Table 2-155 in Perry and Green [2008].

$$C_p = C_1 + C_2T + C_3T^2 \tag{3.20}$$

Substance	C_1	C_2	C_3
Propane	31.742	26.567	0.12927
Ethane	26.675	147.04	0

Table 3.4: Heatcapacity coefficients for Equation 3.20[Perry and Green, 2008]

The individual heat capacities were calculated for the bulk temperature, and then weighted according to molfraction as shown in Equation 3.21.

$$C_p = C_{p,E}x_E + C_{p,P}x_P \tag{3.21}$$

Heat of Vaporisation

The Heat of Vaporisation is calculated by the function ”hcond.m”. This function calculates the heat of condensation based on the formula given in Equation 3.22. The coefficients are taken from Table 2-150 in Perry and Green [2008] and are given in Table 3.5.

$$\Delta h = C_1(1 - T_r)^{\left(C_2 + C_3T_r + C_4T_r^2 + C_5T_r^3\right)} \tag{3.22}$$

Substance	C_1	C_2	C_3	C_4
Propane	$2.9209 \cdot 10^7$	0.78237	-0.77319	0.39246
Ethane	$2.1091 \cdot 10^7$	0.60646	-0.55492	0.32799

Table 3.5: Coefficients for Equation 3.22[Perry and Green, 2008]

Mass transfer

To calculate the mass transfer coefficients for the diffusive flux, the following equations were used. Equation 3.23 is termed Fullers method for calculating the diffusivity.

$$D_{EP} = \frac{10^{-7} T^{1.75} \Big(\frac{1}{M_E} + \frac{1}{M_P}\Big)^{0.5}}{P\Big(\left(\sum \upsilon_E\right)^{1/3} + \left(\sum \upsilon_P\right)^{1/3}\Big)^2} \tag{3.23}$$

The coefficients $\sum \upsilon_i$ are the ”diffusional volumes” and are calculated according to the atoms in the molecules. The diffusional volumes are given in Table 3.6 and are calculated from data given in Fuller et al. [1966].

Substance	$\sum \upsilon_i$
Ethane	44.88
Propane	65.34

Table 3.6: Diffusional volumes[Fuller et al., 1966]

The diffusion coefficient β_{EP} is calculated by means of Equation 3.24 [Sajjan et al., 2004].

$$\beta_{EP} = \frac{h_g}{C_{p,g}} \Big(\frac{cDC_{p,g}}{\lambda}\Big)^{2/3} \tag{3.24}$$

Where c is the concentration and λ is the conductivity. The concentration was calculated by assuming ideal gas.

Conductivity

The conductivity of the gas is calculated by the function ”conductivity.m”. The conductivity data was taken from the simulation program Unisim (”HEX1-system.usc”). The data was then fitted to an equation using linear regression in Excel. The data is shown in Appendix F. The conductivity of the vapour is given by Equation 3.25.

$$\lambda = 6 \cdot 10^{-5} \cdot (T - 273) + 0.0213 \tag{3.25}$$

Viscosity

The viscosity of the vapour was calculated in the same manner as the conductivity. The data was taken from the simulation ”HEX1-system.usc” and Equation 3.26 was fitted using Excel. The data is shown in Appendix F.

$$\mu = (2 \cdot 10^{-5} \cdot (T - 273) + 0.0096) \cdot 10^{-3} \tag{3.26}$$

Temperature and vapour composition at interface

The temperature at the interface, T_I, is calculated under the assumption that the condensate and the vapour are in equilibrium at the interface. The molfraction of the condensate is calculated according to which assumption is used for condensation, integral or differential. The interface temperature and composition are found through finding the temperature and vapour composition that corresponds with the given condensate composition. This is done through the routines ”Tinterface.m” and ”yequilibrium.m”. When the functions are called up, the pressure is identified and the correct equilibrium data is loaded. The correct temperature and vapour composition is then found by interpolation.

3.5 Solving the Equations

The model given in Section 3.2 was built in MATLAB R2010b. The different functions and scripts are given in Appendix A. The states describing the system, temperature (T), sea water temperature (Ta), pressure (P), mols of ethane in vapour (N_E^v), mols of propane in vapour (N_P^v), total mol flux (n_t) and mol flux of ethane (n_e), are given in Equation 3.27.

$$\underline{x} = \begin{bmatrix} T \\ T_a \\ P \\ N_E^v \\ N_P^v \\ n_t \\ n_E \end{bmatrix} \tag{3.27}$$

The model equations are ordinary differential equations and the flux equations are algebraic equations, as described in Section 3.2. As the condenser has counter current flow, the inlet temperature for the cooling water is given at the outlet of the vapour/condensate. This results in a two-point boundary value problem. The introduction of algebraic equations results in a Differential Algebraic System. The first attempts to solve the equation system involved using a routine in MATLAB, bvp4c. The discription of this attempt is shown in Appendix G.

3.5.1 Using a custom built collocation routine

To get a better understanding of the procedure and thus better control over the situation, a collocation routine was built specifically for the problem. The algebraic flux equations and property equations were solved simultaneously as the differential equations. All equations were set up as residuals in a matrix, and the fsolve-function was used to solve the equation system in MATLAB. This is a standard procedure in MATLAB which minimizes the residuals based on Newton's method.

Description of The Model

This section will describe the model set-up as it is used in the MATLAB-scripts given in Appendix A. A flowsheet of the simulation routine is shown in Figure 3.3.

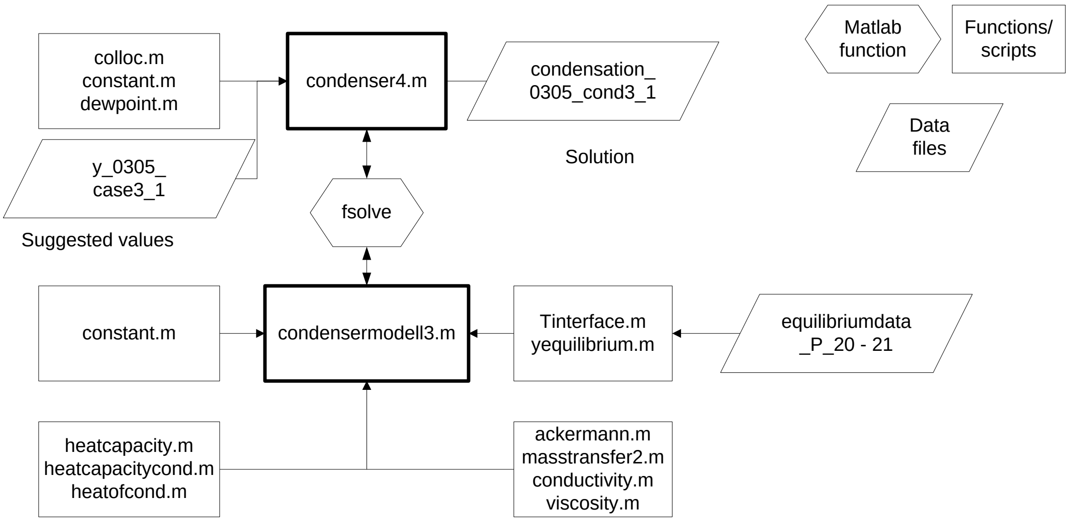


Figure 3.3: Flowsheet of simulation routine in MATLAB for condenser4.m

The main script is ”condenser4.m” which starts by calculating the collocation points utilizing the program ”colloc.m”, written by John W. Eaton. Each section in the condenser has its own set of collocation points and variables that will be transferred to the script containing the set of equations, ”condensermodell3.m”. ”condenser4.m” will also load the script ”constant.m”. This file contains some geometric and physical constants needed in the calculations. The function ”dewpoint.m” calculates the dewpoint for the mixture based on the composition. The datafile ”y_0305_cond3_1” is the suggested values for all collocation points used by fsolve. The solution produced is ”condensation_0305_cond3_1”.

The function containing the equations, ”condensermodell3.m”, is called up by ”condenser4.m” through ”fsolve”. ”condensermodell3.m” also needs the constants in ”constant.m”. The thermodynamic model is called through the functions ”Tinterface.m” and ”yequilibrium.m”, which, in addition, load the datafiles containing the correct equilibriumdata (”equilibriumdata_P_20 - 21”). The parameters described in Section 3.4 are calculated through the functions ”ackermann.m”, ”masstransfer2.m”, ”conductivity.m”, ”viscosity.m”, ”heatcapacity.m”, ”heatcapacitycond.m” and ”heatofcond.m”. All of these scripts and functions are found in Appendix A.3.

Solving the Set of Equations

This section will concentrate on how the model was built up and how the solutions were obtained.

As the collocation method uses a normalized independent variable, all differential equations must be multiplied by A to obtain the right scale as shown in Section 2.7. A is the surface area specified for each section and was found as described in this subsection.

Firstly, the surface area needed to cool the vapour down to the saturation temperature had to be found. The script ”coolingtest.m” was used to find this constant. This script is a simplified version of ”condenser4.m”, containing only the cooling section. To find this area, A_1, a new equation was added to the procedure. This equation specified the temperature, T, at the end of the cooling section equal the saturation temperature. To save computational time a simplified version of the script ”coolingtest.m”, with only the differential equations for the vapour and the cooling medium, was used. The heat capacities and heat transfer coefficients were held constant. The effective area for ”coolingtest.m” was found to be 678.64 m^2. Then ”coolingtest.m” was extended to include the heat capacities. A_1 was then found to be 715.23 m^2. This number was used as a starting value for the more complicated script ”cooling.m” where all the equations were implemented.

When working on the integral condensation, the program ”condenser2.m” and ”condensermodell.m” where used. In this model the equations for condensation and mass transfer where implemented. In this section the differential equations where also multiplied with an effective area A_2. This area was found by the same method as in ”cooling.m”. According to Jordhøy and Stræte [2009], the vapour amount out of the integral condensation zone should be 0.12 weight percent. To obtain this area, a new equation was introduced, as described in Equation 3.28. This defines the weight percent of vapour out of the condenser as 0.12.

$$\frac{N_E^v M_E + N_P^v M_P}{\left(N_E^v + N_E^l\right) M_E + \left(N_P^v + N_P^l\right) M_P} = 0.12 \tag{3.28}$$

The differential condensation was also tested in the program ”condenser4.m” and ”condensermodell2.m”. The equations for differential condensation were implemented into this script. The differential condensation was to carry on until the vapour reached 10°C. To achieve the effective area A_3, a third constraint was added to the model, as shown in Equation 3.29.

$$T_{g,end} = 283 \tag{3.29}$$

It was assumed that the pressure drop in the condenser followed a linear model as shown in Equation 3.30. To obtain the variable k_p, Equation 3.31 was added to the set of equations. According to Jordhøy and Stræte [2009], the pressure out of the integral condensation was assumed to be half of the total pressure drop. This was calculated to be 2040 kPa.

$$\frac{dP}{dA} = -k_p \tag{3.30}$$

$$P_{int,end} = 2040 \cdot 10^3 \tag{3.31}$$

To couple each section together, equations for the section boundaries were included. These stated that the boundary of one section should equal the boundary for the preceding section. For the sea water this was done in the reverse order as the sea water inlet is on the opposite side than the hydrocarbon vapour.

When determining the interface temperature in the function ”Tinterface.m”, the molfraction in the condensate is calculated. To avoid zero in the denominator, a small amount of each component was assumed to be in the liquid. The distribution was decided according to the liquid equilibrium composition at the dew point. The initial values were set to 0.41 for ethane and 0.59 for propane, and lowered in steps to a satisfactory value. The calculated values for the vector ”y” were used as guessed values for the next iteration.

3.6 Comparing the Model with Unisim

To test the model developed in MATLAB, the results were compared with results obtained from the Unisim model described in Section 3.1. To make the two models comparable, the temperature of the sea water in stream sw3 was set to the same temperature as Ta0 in the MATLAB model, 6.5 °C. The vapour molfractions and temperatures in streams 1, 1a, 2, 3, 4, 5 and 6 were extracted and compared to the molfractions and temperatures obtained in the MATLAB model. The temperature of the sea water was extracted from streams sw3 to sw9 and compared to the temperature of the sea water obtained in the model.

3.7 Sensitivity Analysis

To explore the model and analyse its sensitivity to different parameters, a sensitivity analysis was performed. A selection of parameters were altered and the effects were analysed. In this section the execution of the tests are described whereas the results of the tests are presented in Section 4.3.

The parameters chosen for the sensitivity analysis are parameters that can change during operation of a condenser, and thus are relevant for the performance of the condenser. The following parameters were chosen:

- Vapour Heat Transfer Coefficient
- Fouling Resistance
- Sea Water Inlet Temperature
- Mass Flow
- Intitial Composition

For each parameter, the value from the design case was chosen as an initial value. Values above and below these values were chosen to analyse the effect of the parameters. The values chosen are presented together with the results in Section 4.3.

3.7.1 Execution of Tests

All the tests were carried out in MATLAB, using ”condenser4.m” and ”condensermodell3.m”. The different parameters were changed in the script, and the solution of the design case, ”y_0305_cond3_2.mat”, was used as guessed values for the solution of the equation set. If a solution for test conditions closer to the current test conditions was accessible, this solution was used as initial guess values instead. It was experienced that the non-linear model was very sensitive to the guessed values, and bad guesses would result in the thermodynamic model failing. To minimize this problem, the amount of ethane and propane in the initial liquid was raised to the order of 10^{-1}. When the solution for this initial concentration was found, this was used as guessed values whilst the amount of initial liquid was reduced by the order of 10^{-1} for each iteration, until it was in the order of 10^{-4}. This amount was chosen as this represents the same order as the flux, and thus the first condensation at the dew-point was accounted for and not implemented in the set of equations.

It was also found that the time taken to reach a solution varied a lot. Having initial guesses close to the solution did not always result in a solution being found or that the computational time was relatively small. If this was the case, new guess values were chosen. It was experienced that subtracting a small value, for instance 0.001, from the original guess values, could yield a solution.

When performing the tests concerning the initial concentration, several problems arose. First, ”condenser4.m” and ”condensermodell3.m” were used with the guessed values, ”y_0305_cond3_2.mat”. This resulted in the thermodynamic model failing in the differential condensation section. To identify the problem, ”condenser.m” and ”condensermodell6.m” were used. ”condensermodell6.m” does not include the equations for the differential condensation. The guessed values were taken as the initial values, and no equations for the different areas were included. The areas A_1 and A_2 were set as 600 m^2 and
3000 m^2, respectively. This was done to get good guessed values for calculating the areas. The solution was used for an extension of ”condensermodell6.m” where the equations for the areas were included. When investigating the results for this case, it was found that all the propane had condensed and no constraints had been added to ensure that the vapour flow was kept above 0 kmol/s. A new equation, stating that the amount of mols of propane in the vapour at the end of the condensation was 0.3 kmol/s, was used to define the outlet of the integral condensation zone. This was done to ensure that the condensation ended before negative mol values were obtained.

3.7.2 Mixed Effects

To see if the effects of the different parameters could be separated in the model, a series of tests were carried out were two parameters were chosen. In addition to changing the fouling resistance, the mass flow and sea water temperature were changed. For each mass flow or sea water temperature there were three fouling resistances.

3.8 Validating Model against Experimental Values

The model was tested against experimental data from condenser 25-HA-112 at the LNG-plant at Hammerfest. The initial conditions are presented in Table 3.7.

Table 3.7: Initial conditions for Case study

Parameter	Value	Unit
Mass flow	381.9	kg/s
Temperature	33	C
Sea Water Temp	4.8	C
Pressure	20.5	bar
molfraction ethane	0.64	-

The pressure drop was adjusted so that the pressure at the outlet was 20,1 bar.

Table 3.8: Temperature measurements along condenser in °C

Length [mm]	1680	4500	7280	10040	12820	15610	17310	17710	18450	18325	19715
from top [Deg]											
25.4	20.6			*	5.2	*		-	-	*	*
49.7	21.1	16.2	10.3	6	4.4	3.5		-	-	3.4	3
87.1	15.8	12.4	7.3	6	3.7	2.7		-	-	2.7	2.8
137.8	15.6	13.1	4.8	3.7	3.4	3.9	3.1	-	-		3.4
176.3	13.5	13.4	11.7	9.2	7.8	6.6	6.4	-	-		
202.9	13.6	11.6	10	8.3	8.3	6.8	6.1	-	-		
221.3	12	6.1	5.5	4.2	5.8	5.4	5.3	-	-		

The process data used is presented in Table 3.8. As there are several temperature measurements at the same length, the highest temperature was chosen for the comparison with the Matlab model. As the Matlab model operates in terms of area, the length was transformed to area by multiplying with 675.56 m^2. This number represents the area per meter. The composition measurements are given in Appendix C.

To adjust the Matlab model to fit the experimental data, the equation set was changed. From the measurements, the area for separation was identified at approximately 5000 m^2. At this point the temperature was approximately 285 K. Due to the sensitive model, large adjustments were not made. The areas A2 and A3 were set to 5000 m^2 each, and the heat transfer coefficients, U, Us and hg, were manually changed to obtain comparable results.

Traces of other components in vapour stream

From a measurement sample taken from the history logs at Statoil, the composition of vapour was found to contain traces of methane and butane. In Unisim it was investigated if this could be of significance when evaluating the performance of the condenser. A simple test was carried out, were the composition in the vapour was defined as in Table 3.9. The performance of each heat exchanger was compared for the two cases.

Component	molfraction
Ethane	0.635
Propane	0.361
Butane	0.004

Table 3.9: Initial composition for Trace test

Chapter 4

Results

4.1 The Model

The initial results of the model are presented in this section.

From Figure 4.1 the temperature profiles along the condenser can be seen. The profiles are given according to the surface area. It can be seen that the initial cooling section is not significantly large, but that the gradient for the vapour temperature is quite large. When the vapour reaches the dew-point, the vapour temperature rises, before it falls again. The temperature declines towards the end, and the integral condensation section terminates with the vapour percent reaching 12 weight%. From this point the differential condensation takes over. As can be seen in Figure 4.1, the gradient changes once more when entering this section, with the temperature falling much faster. The sea water temperature increases from the inlet at the cold end towards the outlet at the warm end. The interface temperature, which can approximate the condensate temperature, lies below the vapour temperature during throughout the most of the condenser. At the initial condensation zone the interface temperature is situated above the vapour temperature.

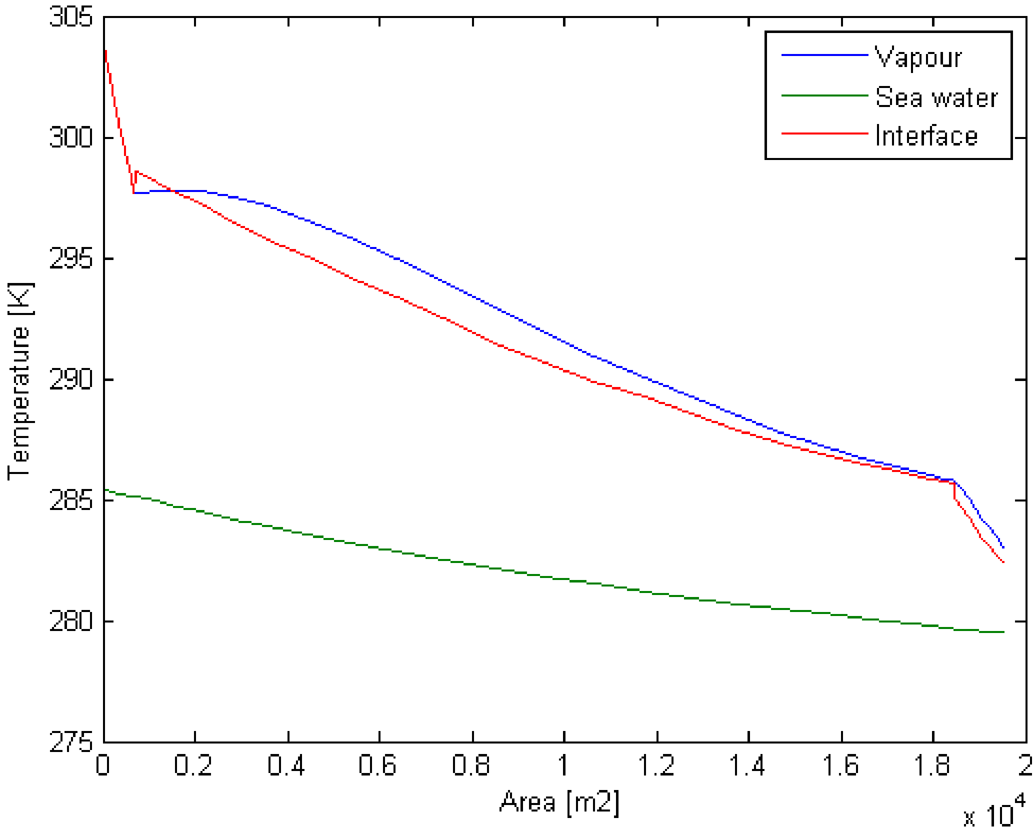


Figure 4.1: Temperature profile along Condenser

Figure 4.2 presents the flux profiles throughout the condenser. The total flux reaches its highest value at the initial phase of condensation. The rate of condensation falls gradually through the condenser until the zone for differential condensation is reached. Here the flux makes a drastic jump down before continuing to fall. The individual condensation rates behave a bit differently. The condensation of propane has the same profile as the total flux in the integral condensation zone. However in the differential zone the flux of propane makes a jump up before it continues to fall. The flux of ethane increases slightly from the start of condensation, reaching a maximum at approximately $0.6 \cdot 10^4$ m^2 before declining slightly towards the end of the integral section. The flux of ethane is quite constant throughout the integral section. In the differential section the flux of ethane makes a jump down before declining smoothly towards the end.

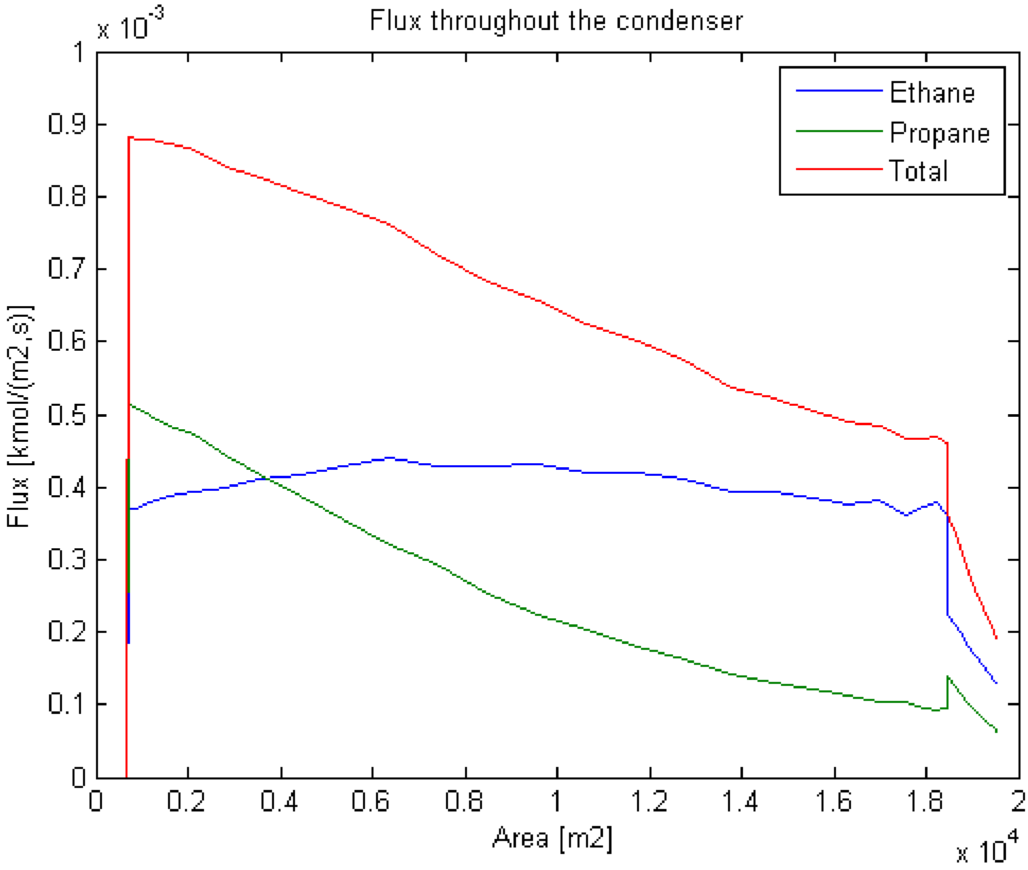


Figure 4.2: Flux profile along condenser

In Figure 4.3 the composition profiles are presented. The amount of ethane in the vapour decreases almost linearly through the integral section, whereas the amount of propane does not have a linear profile. At approximately $0.6{\cdot}10^4$ m^2 the amount of propane in the vapour is equal to the amount of propane in liquid. Slightly further down the condenser the amount of propane in the liquid is exceeded by the amount of ethane in the liquid. When entering the differential section, the profile for propane in both liquid and vapour seems to be continuous, as opposed to the ethane profiles, which have a change in gradient.

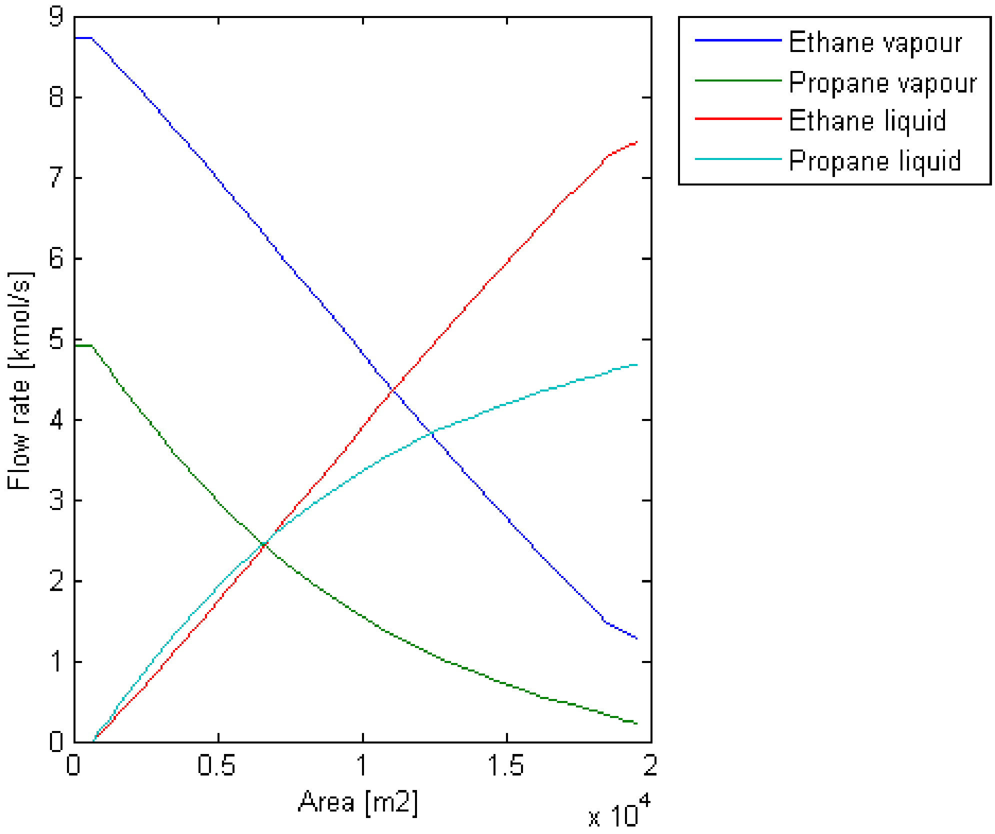


Figure 4.3: Flowrate along Condenser

The molfractions of both vapour and condensate are presented in Figure 4.4. The molfraction of ethane in the vapour rises continuously through the integral condensation section. The change in molfraction for the vapour increases drastically in the differential condensation section. The molfraction goes from 0.81 to 0.85 in a matter of 1000 m^2. The liquid composition does, however, not change so dramatically.

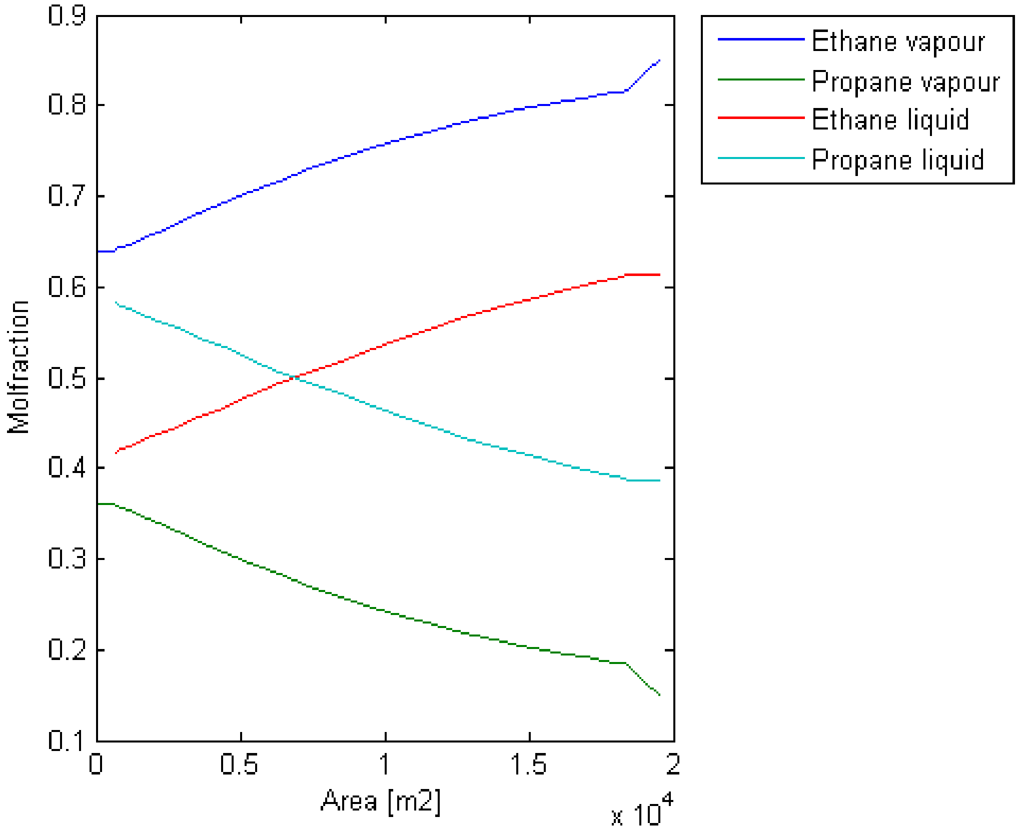


Figure 4.4: Molfraction along Condenser

The pressure profile is presented in Figure 4.5. The pressure declines linearly through the condenser.

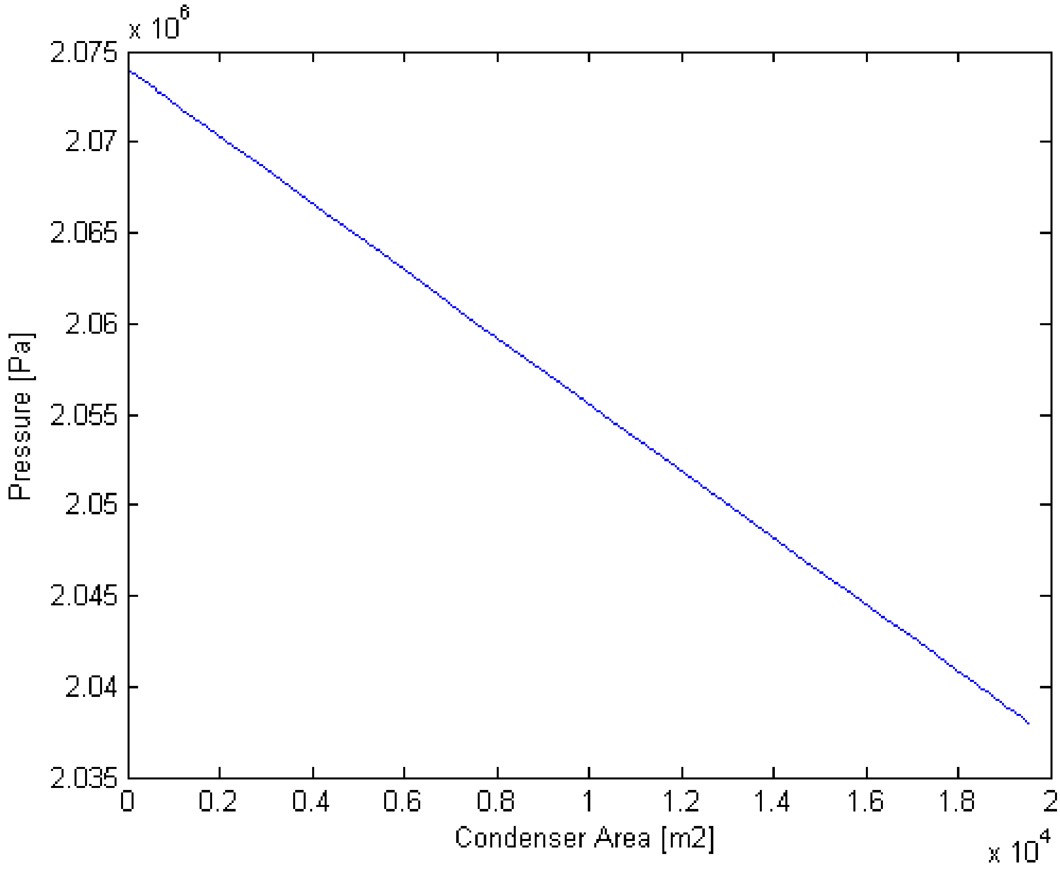


Figure 4.5: Pressure along condenser

4.2 Comparison with Unisim

In this section the results of the comparison between the Matlab model and the Unisim model are presented.

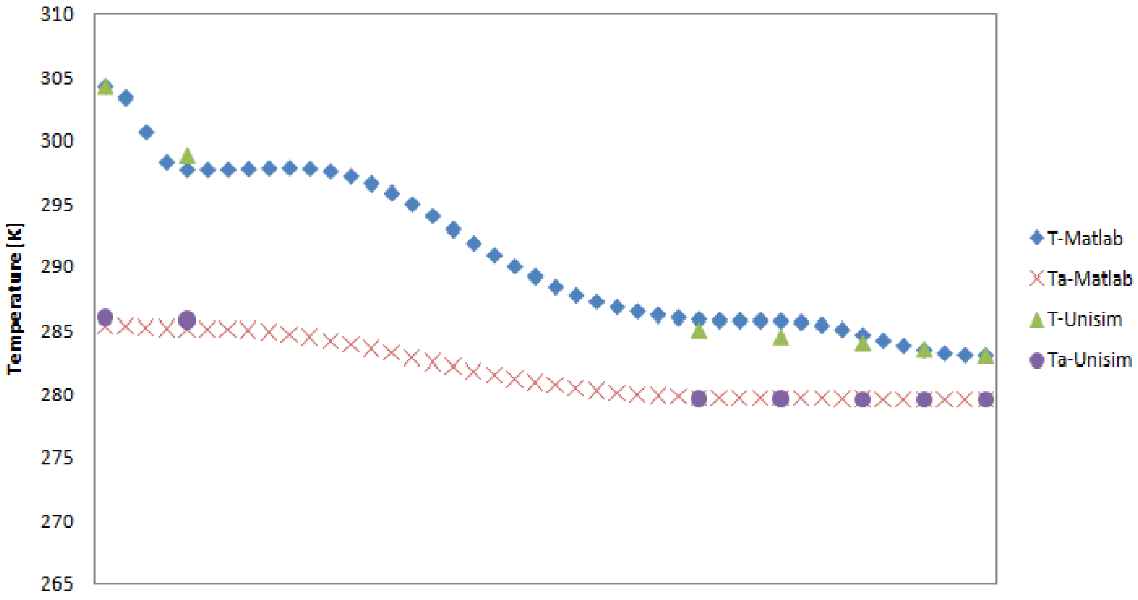


Figure 4.6: Comparison of Temperature profiles, Matlab vrs Unisim, T=vapour temperature, Ta=sea water temperature

From Figure 4.6 it can be seen that the temperature profiles follow the same trends. The Unisim model tends to calculate higher temperatures than the Matlab model for the cooling section, whereas the condensation section temperatures obtained from Unisim lie a bit below the temperatures calculated in Matlab.

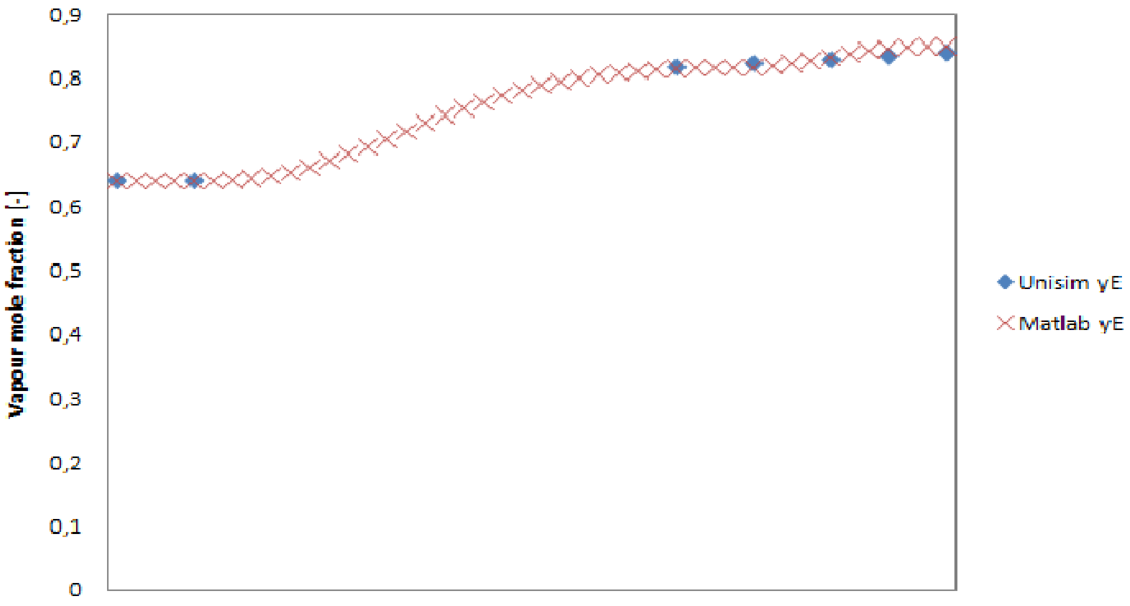


Figure 4.7: Comparison of molfractions, Matlab vrs Unisim

In Figure 4.7 the vapour molfractions calculated by Unisim and Matlab are shown. The composition the trends are similar and there seems to be small differences between the two. The percentual difference is shown in Table 4.1, and at most the difference is 1.3 %.

Table 4.1: Comparison of molfractions - Unisim and Matlab

yE_M	yE_U	Deviation	Relative deviation in percent
0.64	0.64	0	0
0.64	0.64	0	0
0.815	0.818	-0.003	-0.330
0.833	0.829	0.003	0.408
0.846	0.831	0.015	1.844
0.850	0.841	0.010	1.137

4.3 Results of Sensitivity Analysis

This section presents the results from the sensitivity analysis as described in Section 3.7. A_1 is the area for cooling from initial temperature down to the dew-point, A_2 is the integral condensation part, whereas A_3 is the differential condensation part. For each parameter the area required, temperature approach and conditions at outlet were investigated. For the change in vapour heat transfer coefficient the temperature approach was not investigated. The temperature approach was taken at a fixed point for each series to allow comparable data. This point was taken as the area for the series giving the smallest total area.

4.3.1 Variation in Vapour Heat Transfer Coefficient

The values used for the analysis are presented in Table 4.2.

Table 4.2: Vapour side heat transfer coefficients for sensitivity analysis

Vapour side heat transfer coefficient [W/m^{2}K]	Percent change from design case [%]
300	0
350	16.67
400	33.33
600	100
900	200

In Figure 4.8, the percentual change in area required with percentual change in the vapour side heat transfer coefficient, h, is shown. It can be seen that A_1 increases slightly with increasing h. A_2 and A_3 decrease, with A_3 decreasing considerably more than A_2.

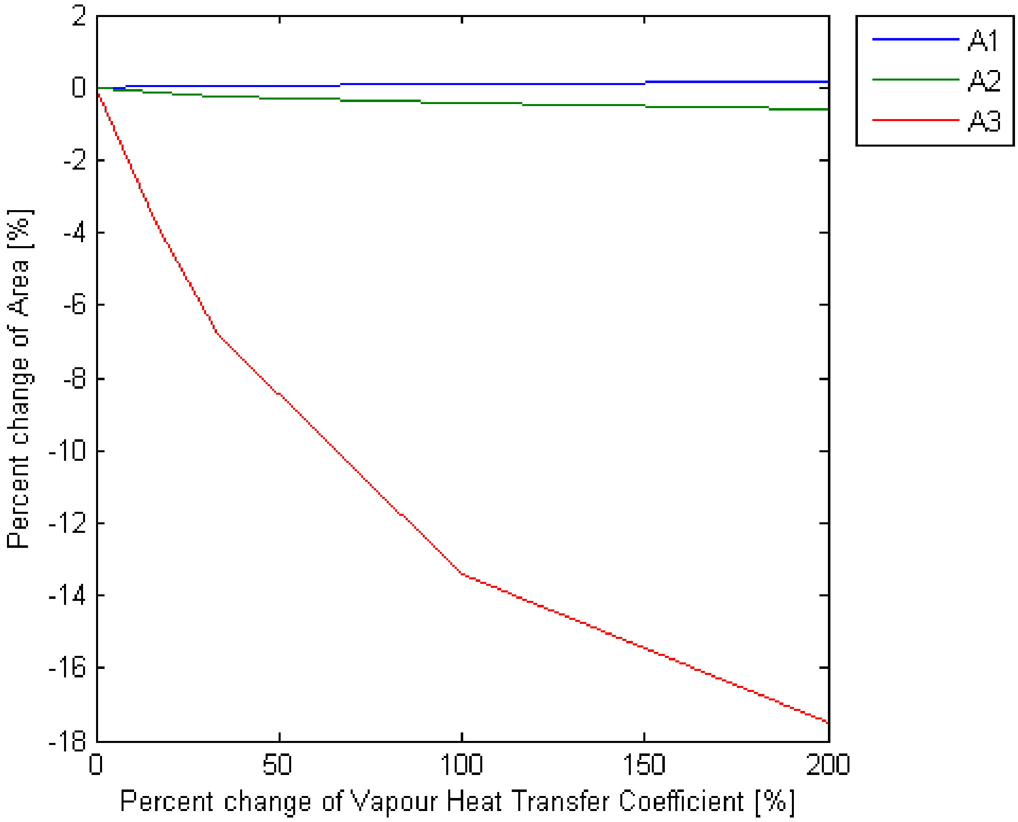


Figure 4.8: Percentual change in Area with percentual change in Gas side Heat transfer coefficient

From Figure 4.9 it can be seen how the outlet conditions vary with the vapour side heat transfer coefficient. The higher heat transfer coefficients give more vapour left at the outlet and a lower molfraction of ethane in the vapour.

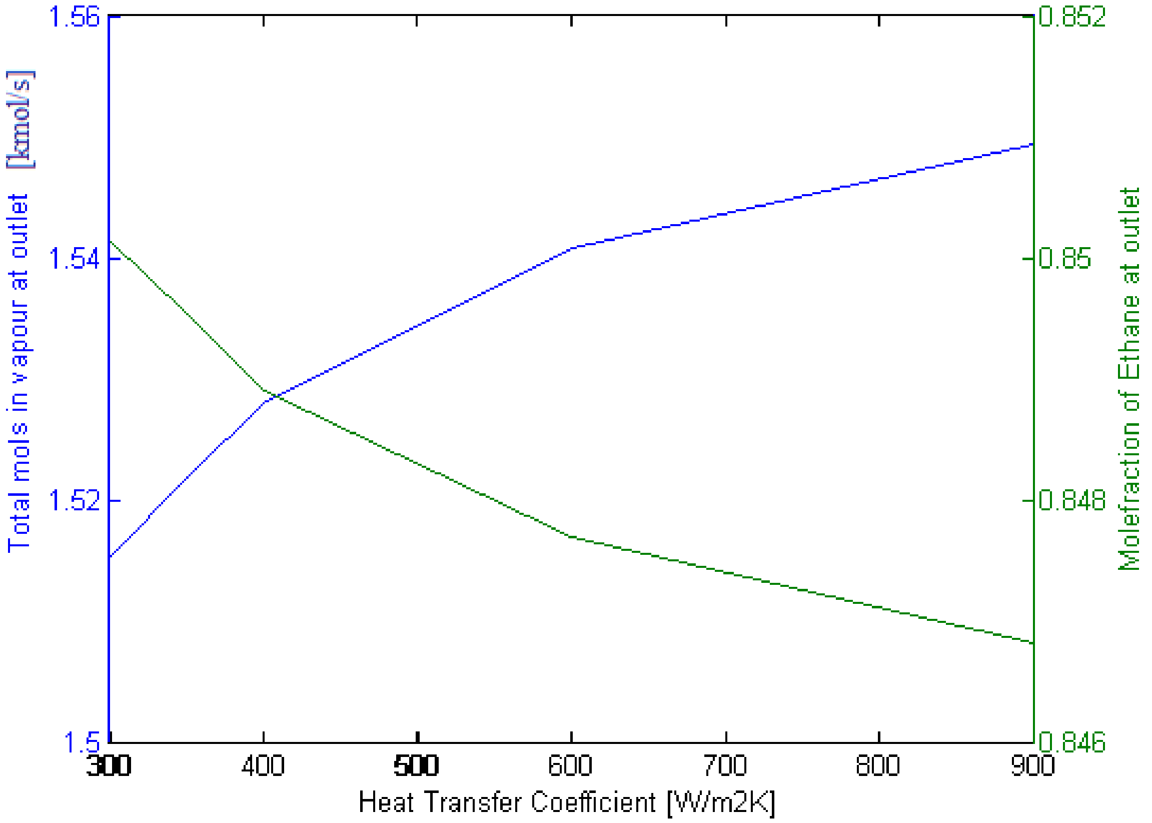


Figure 4.9: Outlet conditions for different vapour side heat transfer coefficients

4.3.2 Variation in Mass flow

The values used for the analysis are presented in Table 4.3.

Table 4.3: Mass flows for sensitivity analysis

Mass Flow [kg/s]	Percent change from design case [%]
478.31	0
500	4.53
600	25.44

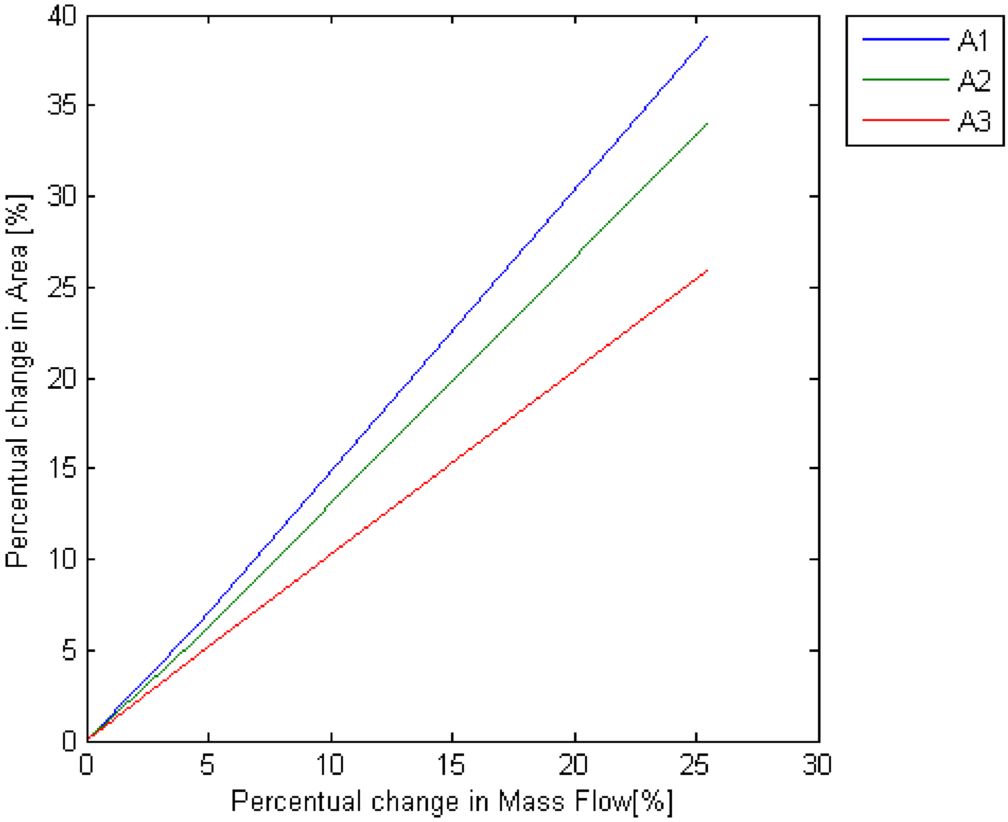


Figure 4.10: Percentual change in Area with change of mass flow

The plot in Figure 4.10 presents the percentual change in area with percentual change of mass flow. All the areas increase with increasing mass flow, with A_1 increasing the most and A_3 increasing the least.

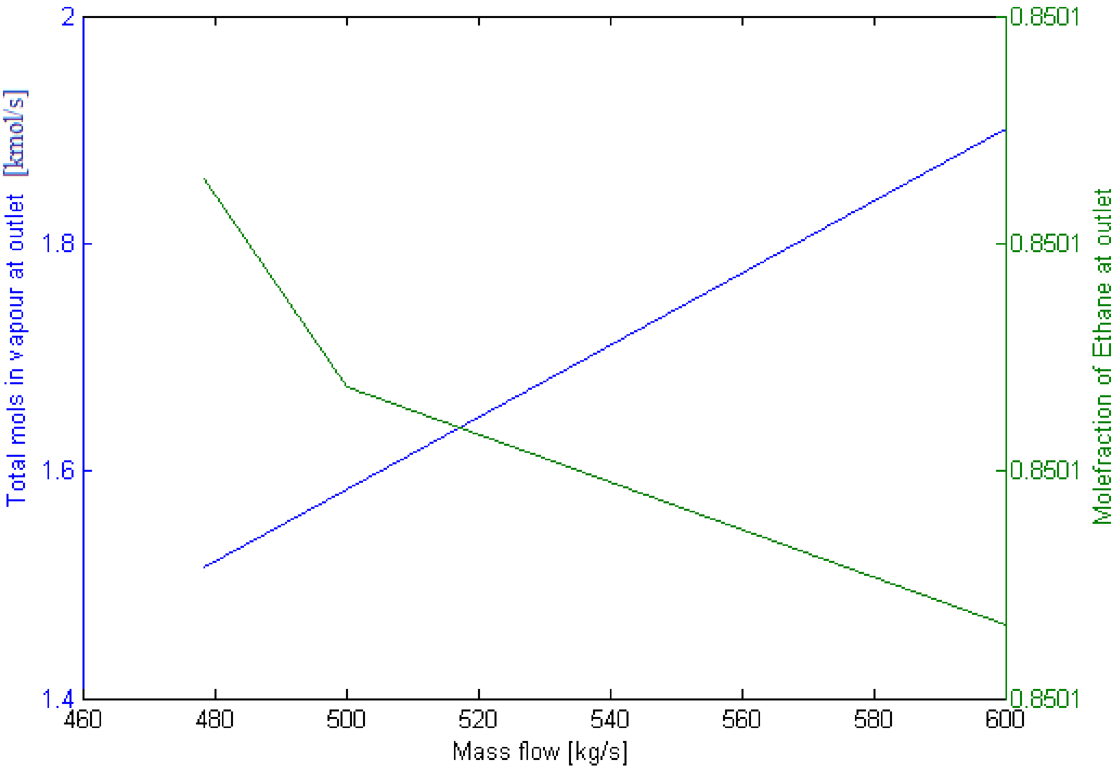


Figure 4.11: Outlet conditions for different mass flows

In Figure 4.11 the variation in outlet conditions with variation in mass flow is presented. There seems to be a linear increase in amount of vapour left at outlet with increase of mass flow. However, there is not a linear decrease of molfraction of ethane with increase of mass flow.

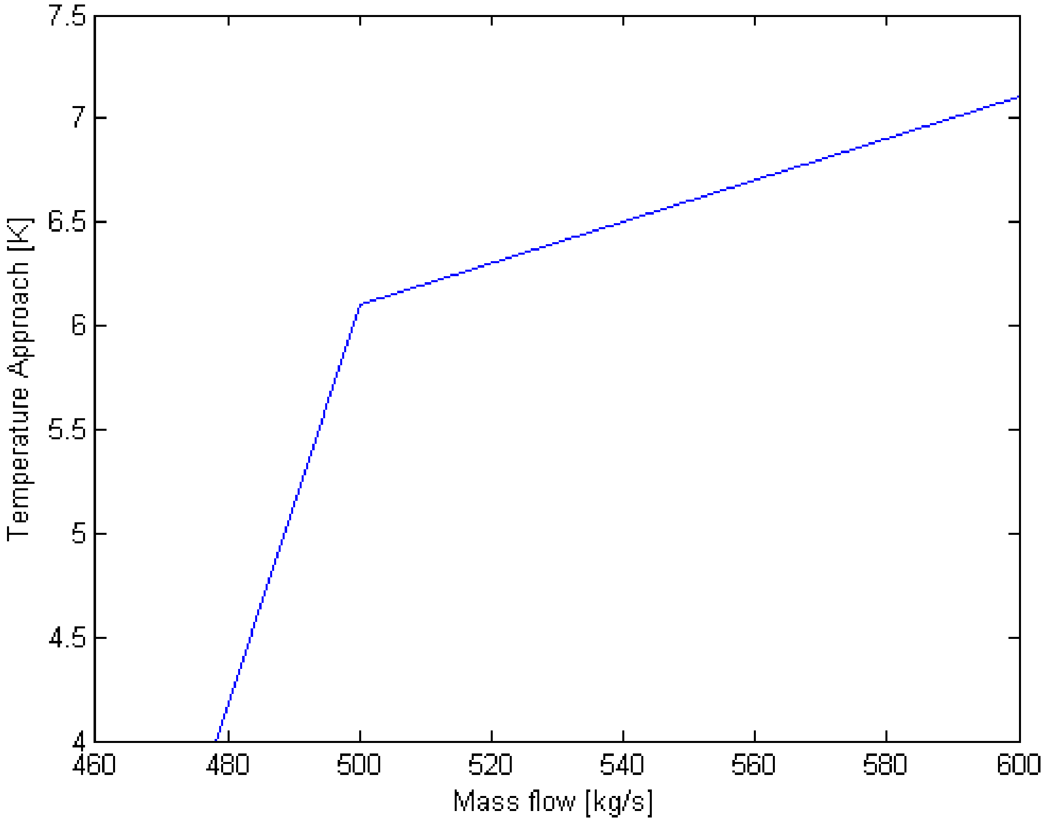


Figure 4.12: Change in Temperature approach with change in mass flow

In Figure 4.12 the temperature approach for different mass flows are presented. With an increase of mass flow the temperature approach increases.

4.3.3 Variation in Sea Water Inlet Temperature

The values used for the analysis are presented in Table 4.4

Table 4.4: Sea Water Temperatures for sensitivity analysis

Sea Water Temperature [K]	Percent change from design case
279	-7.7
279.5	0
279.6	1.53
279.7	3.1

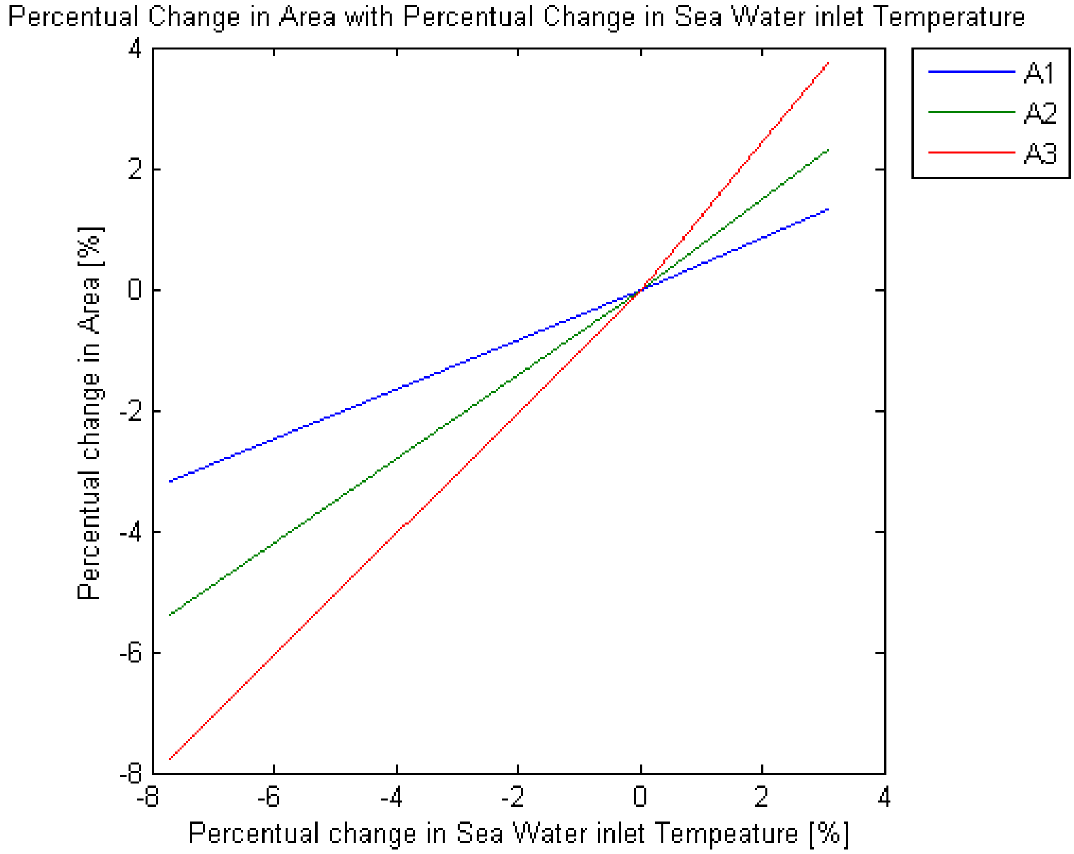


Figure 4.13: Percentual change in Area with change in Sea Water Inlet temperature

In Figure 4.13 the percentual change in area with change of sea water inlet temperature, T_{a0}, is presented. All three areas increase with increasing T_{a0}, but A_3 increases the most and A_1 increases the least.

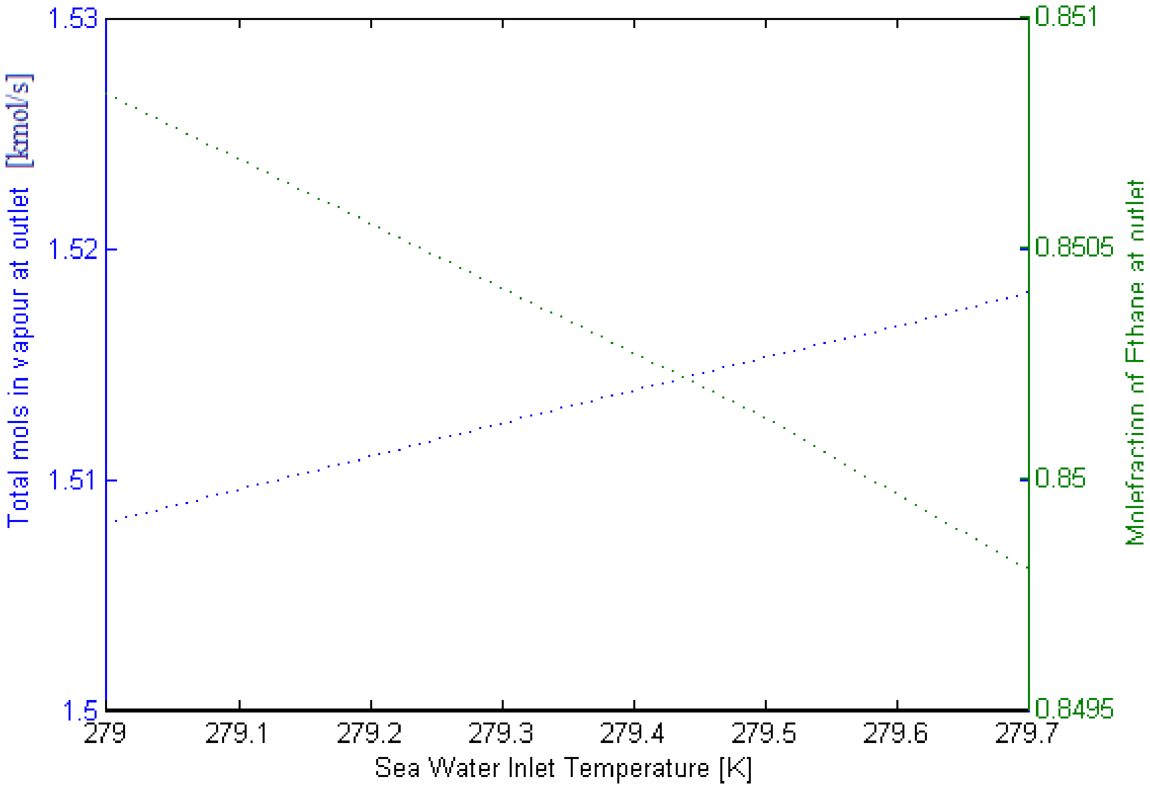


Figure 4.14: Outlet conditions for different Sea Water Inlet temperatures

From Figure 4.14 it can be seen that there is a linear increase of vapour at outlet with increase in sea water inlet temperature. The molfraction of ethane at the outlet seems to have a linear decrease with increase of the sea water temperature. Note that the temperature range is quite narrow.

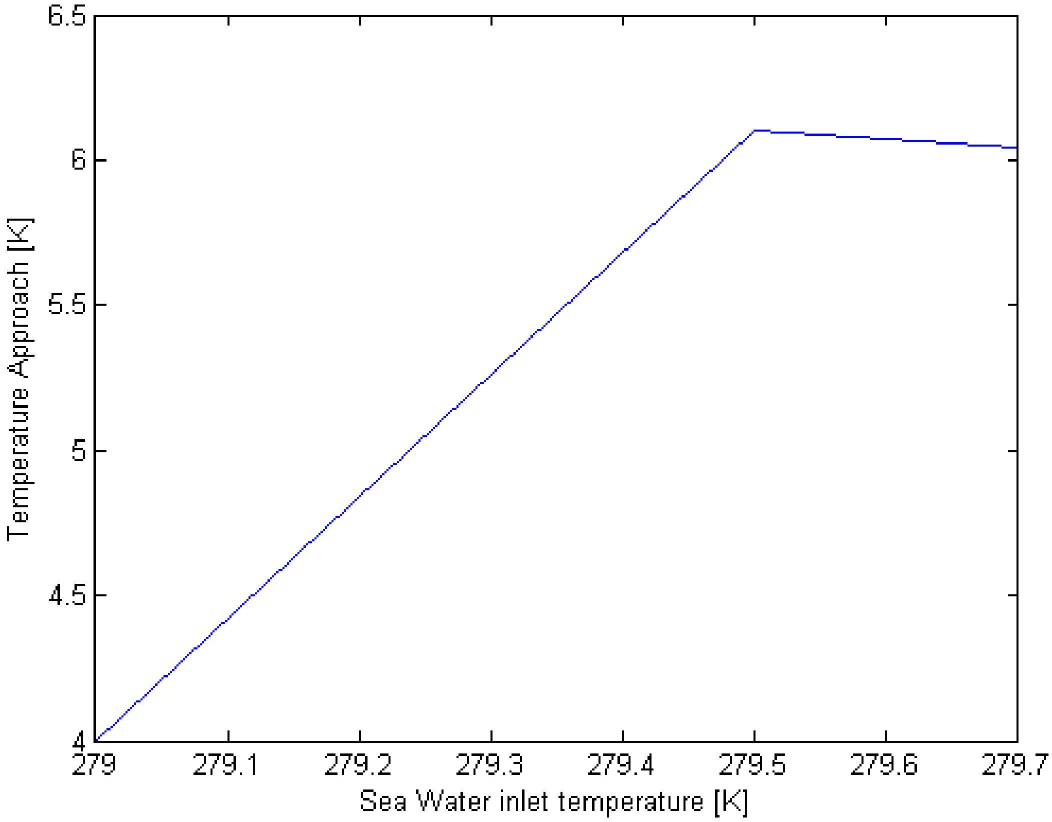


Figure 4.15: Change in Temperature approach with change in Sea Water Inlet Temperature

In Figure 4.15 the temperature approach for different sea water temperatures are presented. With an increase of sea water temperature the temperature approach increases. The temperature approach profiles throughout the condenser for the different cases are presented in Figure 4.16.

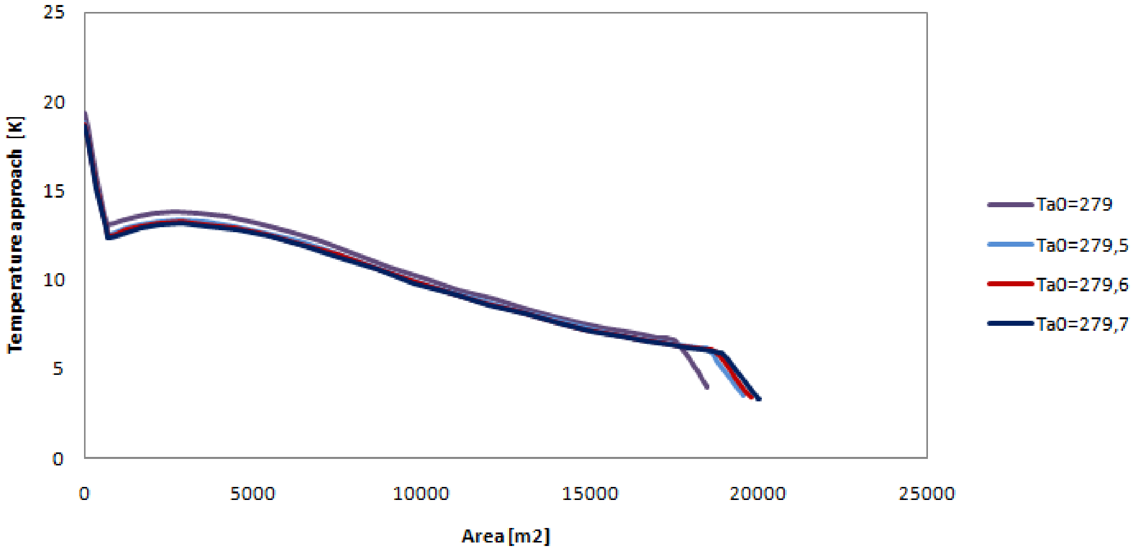


Figure 4.16: Temperature approach profile

4.3.4 Variation in Fouling Resistance

The values used for the analysis are presented in Table 4.5.

Table 4.5: Fouling Resistances for sensitivity analysis

Fouling Resistance [m^2K/W]
0.00
$1.78{\cdot}10^{-04}$
$3.56{\cdot}10^{-04}$
$5.34{\cdot}10^{-04}$

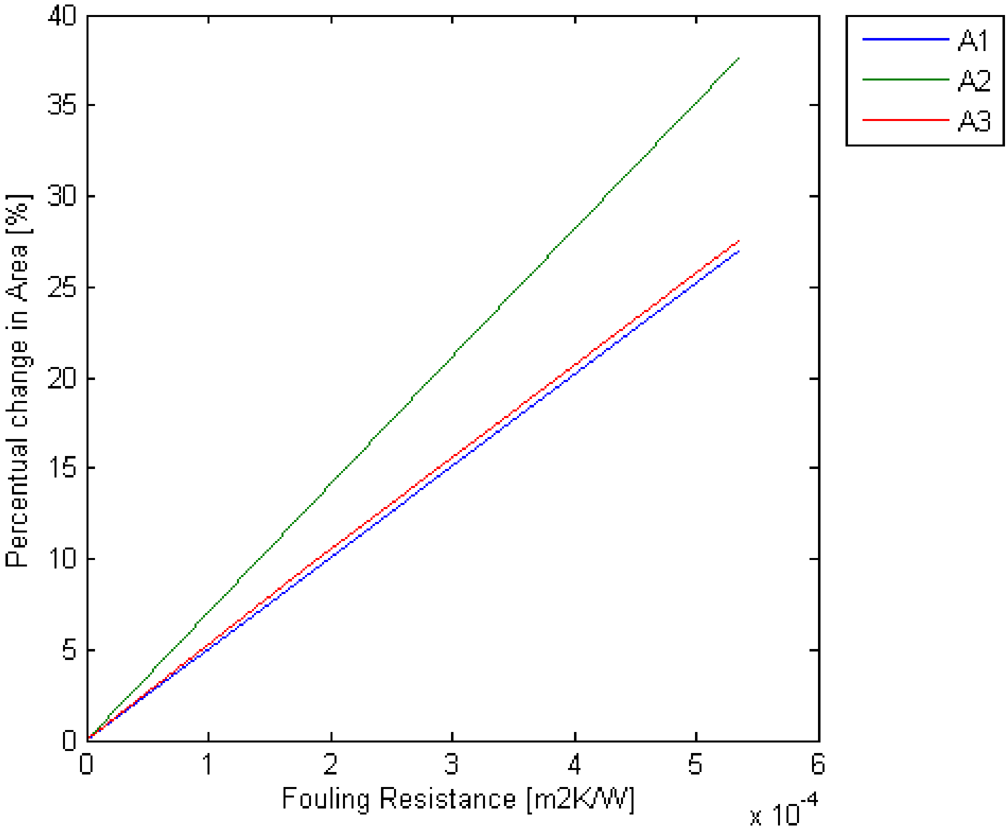


Figure 4.17: Percentual change in Area with change in Fouling Resistance

Figure 4.17 presents the percentual change in area with change of fouling resistance. It can be seen that all areas increase with increasing fouling. A_1 and A_3 increase almost equally, whereas A_2 increases the most.

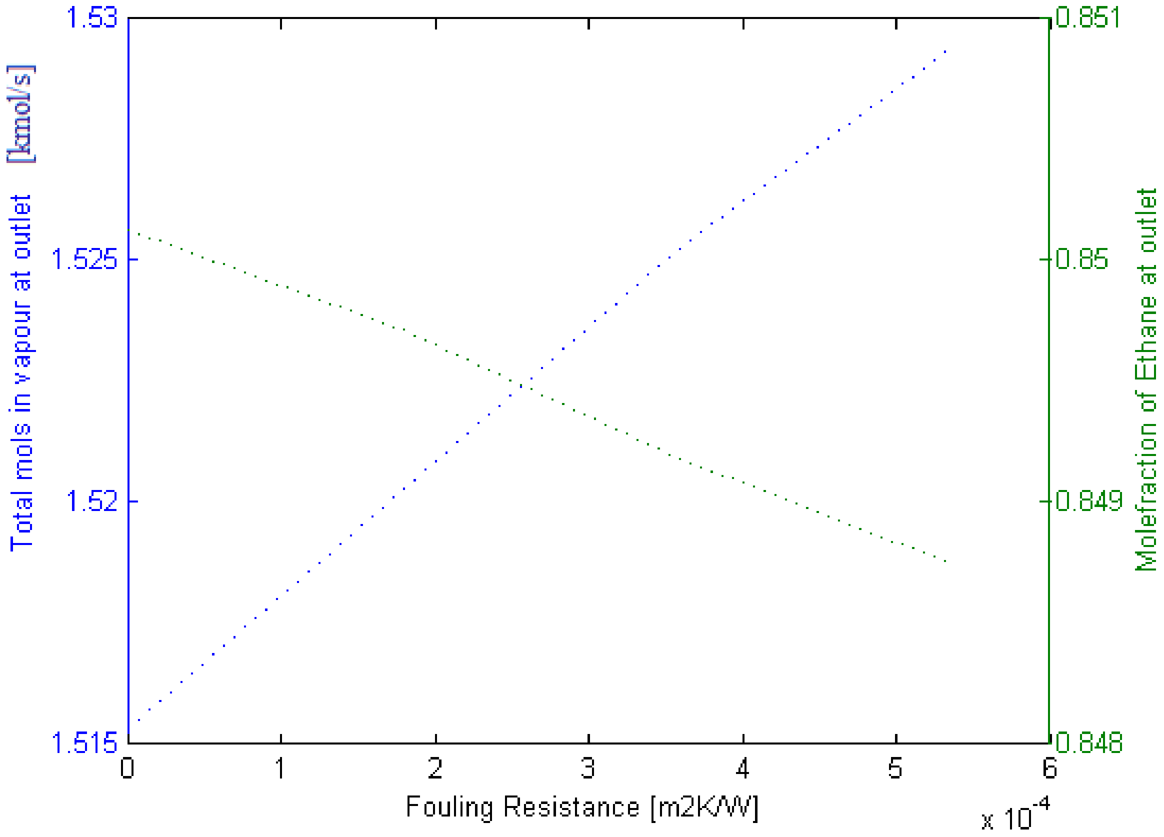


Figure 4.18: Outlet conditions for different fouling resistances

In Figure 4.18 the variation of outlet conditions with variation in fouling resistance is presented. It can be seen that the amount of vapour at outlet increases with increasing fouling resistance. The fraction of ethane decreases with increased fouling resistance.

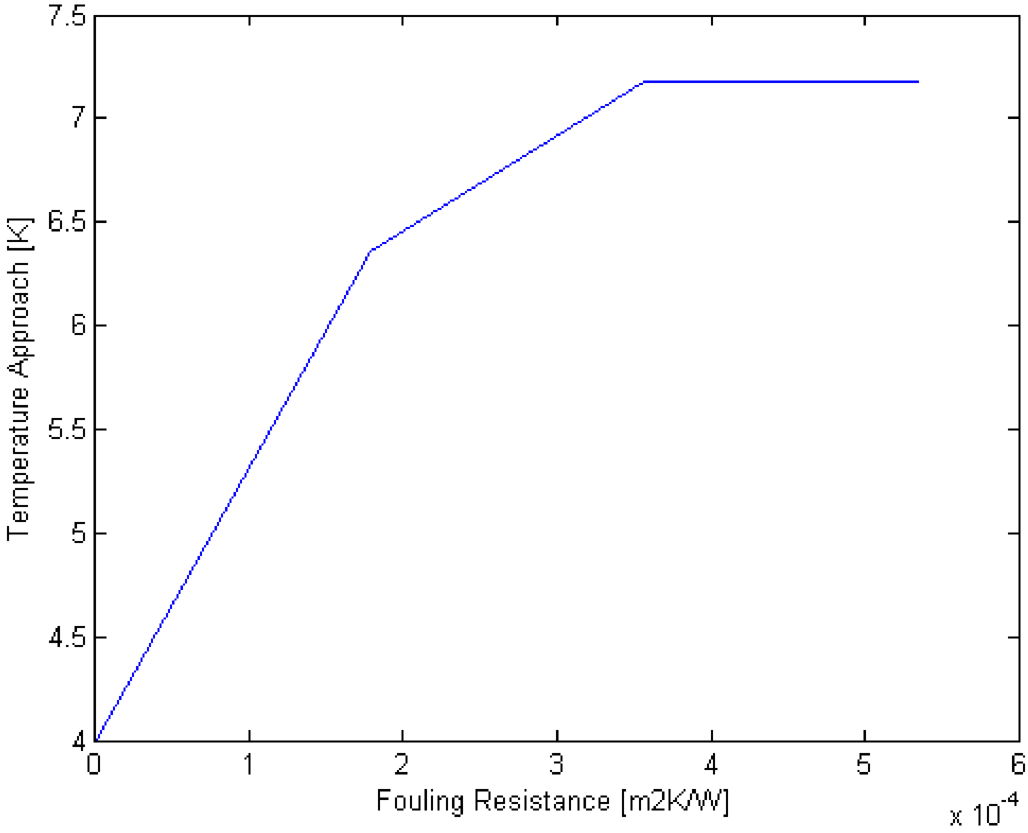


Figure 4.19: Change in Temperature approach with change in Fouling Resistance

In Figure 4.19 the temperature approach for different fouling resistances are presented. With an increase of fouling the temperature approach increases.

4.3.5 Change in composition

The values used for the analysis are presented in Table 4.6

Table 4.6: Initial compositions for sensitivity analysis

Molfraction Ethane [-]
0.64
0.63
0.65

The area required for the different zones with different initial compositions are presented in Table 4.7. The percentual change in area is presented in Figure 4.20. From both the table and the figure it can be seen that the area increases with increase of amount of initial ethane. The percentual change in area for cooling is larger than the percentual change of condensation.

Table 4.7: Area required versus initial composition

A_1 [m^2]	A_2 [m^2]	y_E [-]
585	16 751	0.63
671	17 967	0.64
760	19 083	0.65

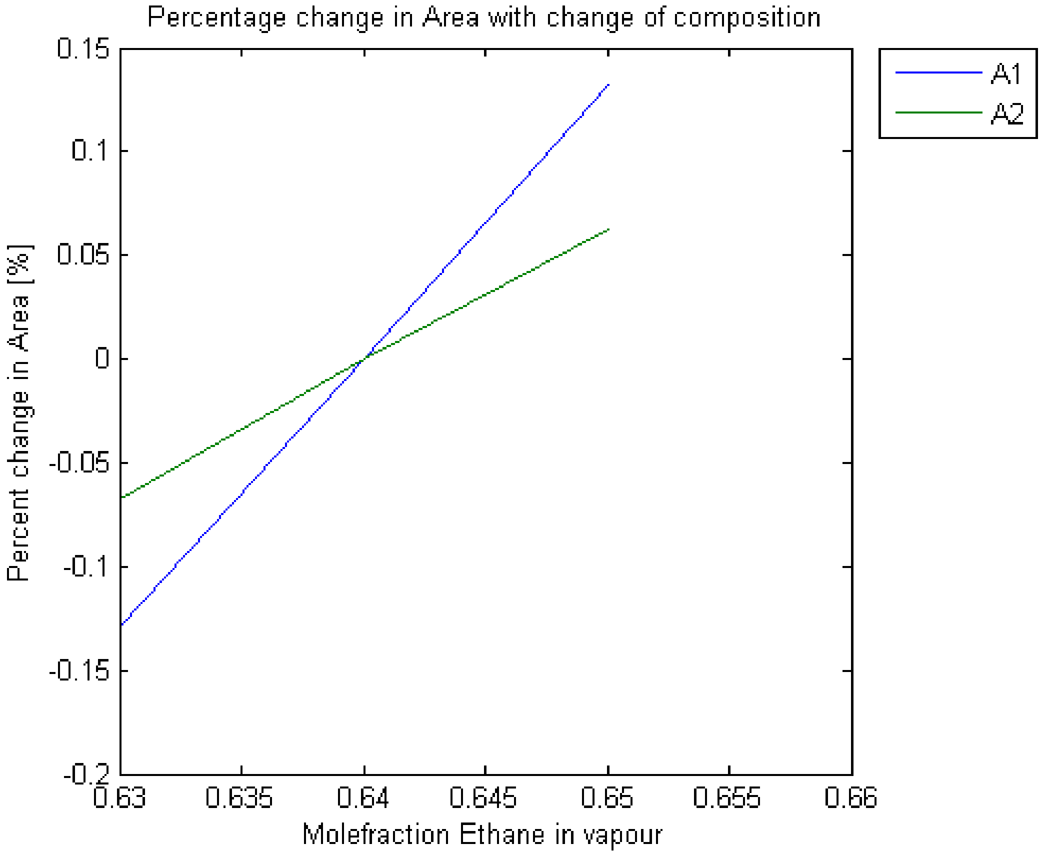


Figure 4.20: Percentual change of Area with change in initial composition

4.3.6 Results for Mixed Effects

This section presents the results for the mixed effects tests. The input values for mass flow, temperature and fouling resistance are presented in Tables 4.8 and 4.9.

Table 4.8: Parameters for Mixed Effects, Mass flow vs. fouling resistance

Mass Flow [kg/s]	Fouling Resistance [m^2K/W]		
478.31	0	$1.78\cdot10^{-4}$	$3.56\cdot10^{-4}$
500	0	$1.78\cdot10^{-4}$	$3.56\cdot10^{-4}$

Table 4.9: Parameters for Mixed Effects, Sea water temperature vs. fouling resistance

Sea Water Temperature [K]	Fouling Resistance [m^2K/W]		
279	0	$1.78\cdot10^{-4}$	$3.56\cdot10^{-4}$
279.5	0	$1.78\cdot10^{-4}$	$3.56\cdot10^{-4}$

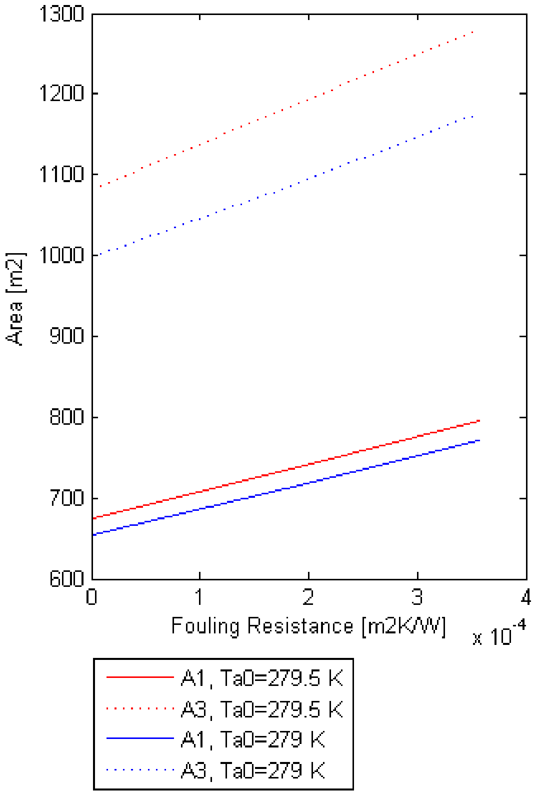


Figure 4.21: Fouling Resistance versus area for different sea water temperatures

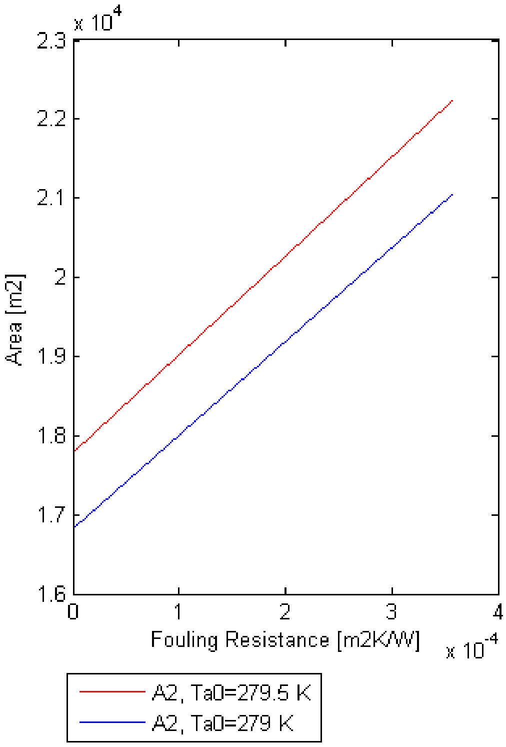


Figure 4.22: Fouling Resistance versus area for different sea water temperatures

In Figure 4.21 and Figure 4.22 the areas are plotted against the fouling resistance for two different sea water temperatures. The areas increase with increased sea water temperature and increased fouling resistance.

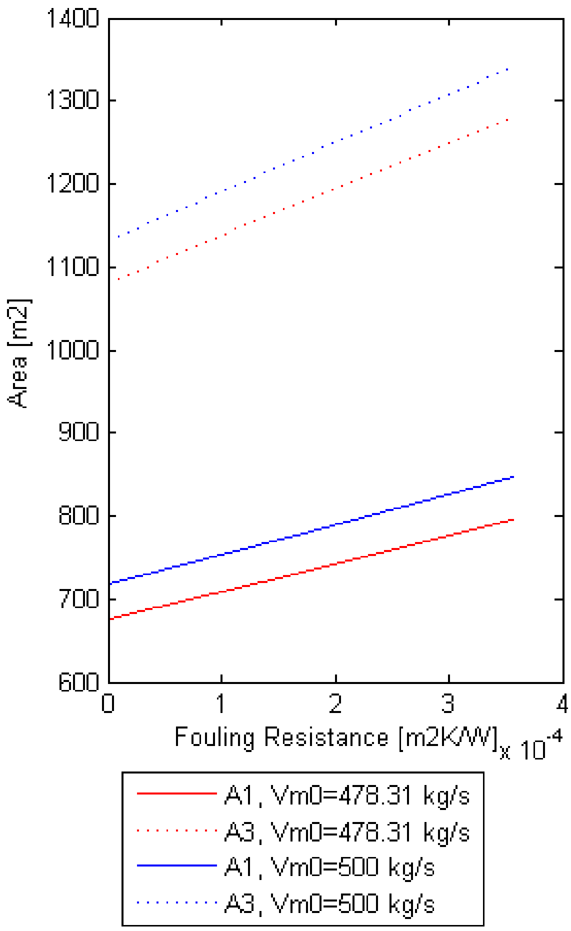


Figure 4.23: Fouling Resistance versus area for different mass flows

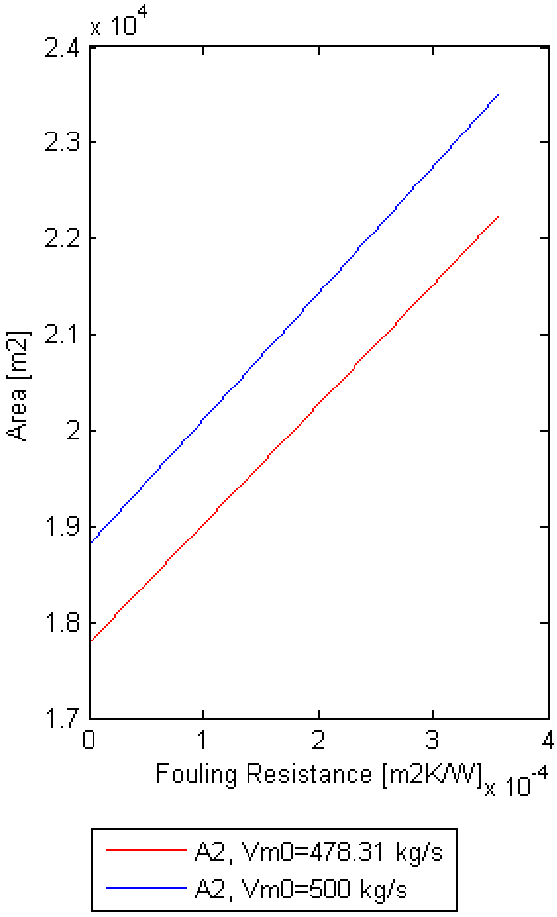


Figure 4.24: Fouling Resistance versus area for different mass flows

In Figure 4.23 and Figure 4.24 the areas are plotted against the fouling resistance for two different mass flows. The areas increase with increased mass flow and increased fouling resistance. This coincides with the trends obtained for the single effects test presented in Figure 4.10 and Figure 4.17

4.4 Results for experimental validation

The first results of the Matlab model were compared with the experimental results, as presented in Figure 4.25. As can be seen, the Matlab profile was situated at much higher temperatures than the experimental values.

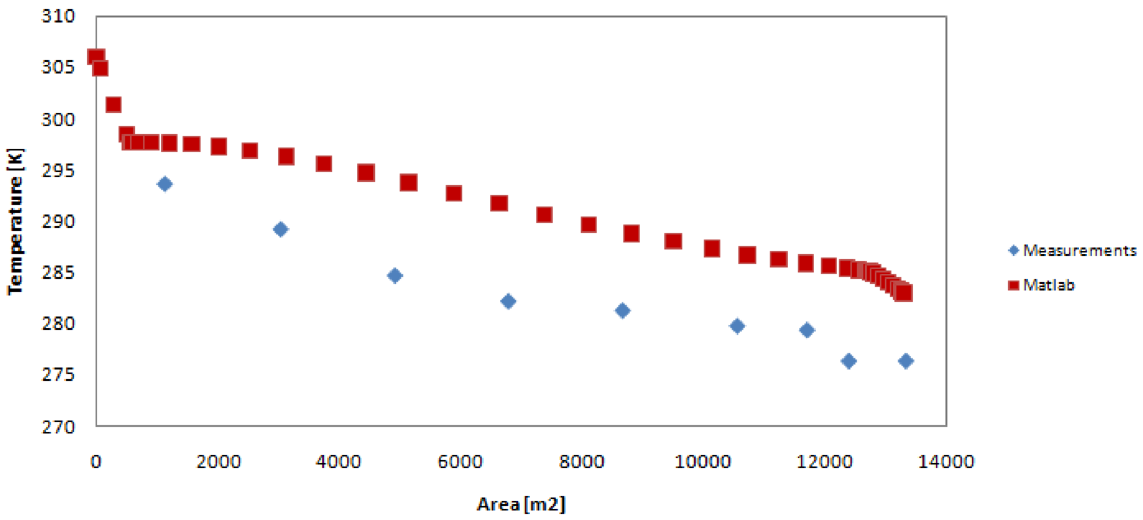


Figure 4.25: Temperature profile comparison, Matlab vrs Measurements

After some adjustments, the temperature and composition profiles were compared again as shown in Figure 4.26 and Figure 4.27. The adjustments made are presented in Table 4.10. The profiles are close but not yet optimal. At approximately 8000 m^2 the two profiles match, both in temperature and composition.

The temperature profiles throughout the condenser are shown in Figure 4.28. Here the vapour, interface and sea water temperatures are plotted. It can be seen that the interface temperature and vapour temperature are close in the integral condensation section, whereas the distance is larger in the differential section. The interface temperature is also situated above the vapour temperature in the initial stage of the integral condensation.

Table 4.10: Adjusted parameters for Case

Parameter	Value	
A1	389.54	m^2
A2	5000	m^2
A3	5000	m^2
hg	8000	W/m^2K
hg3	700	W/m^2K
U	1000	W/m^2K
U1	1000	W/m^2K
Us	700	W/m^2K

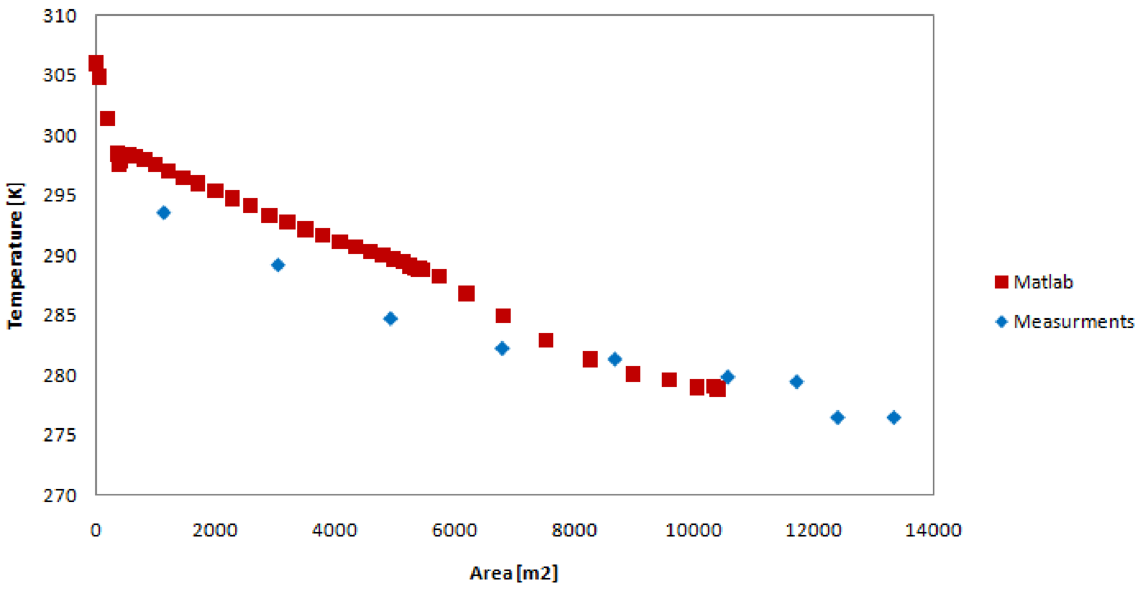


Figure 4.26: Temperature profile comparison, Matlab vs. Measurements, after adjustments

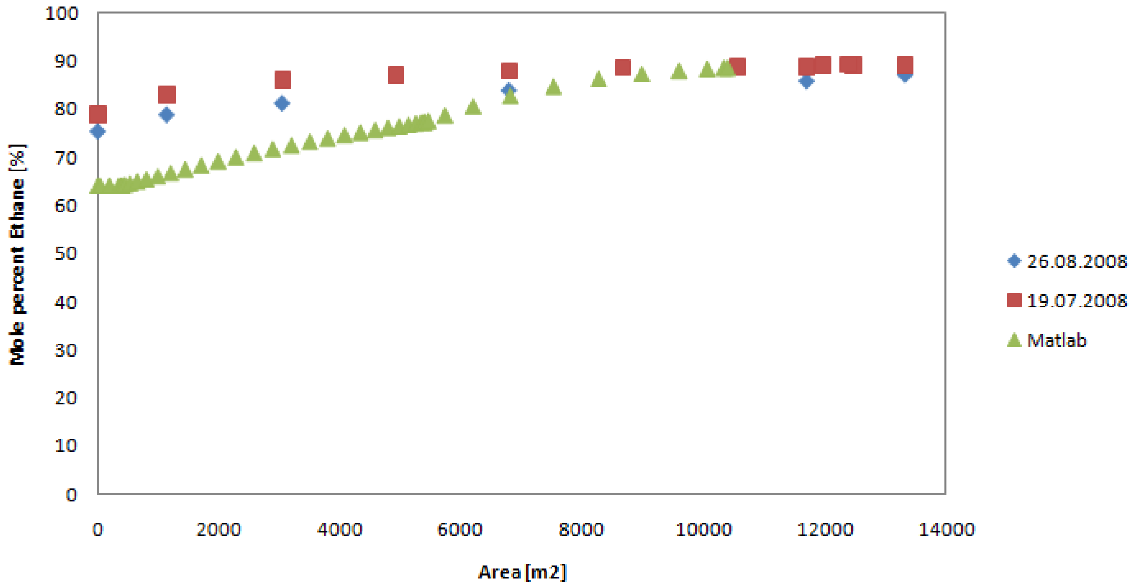


Figure 4.27: Composition profile comparison, Matlab vs. Measurements

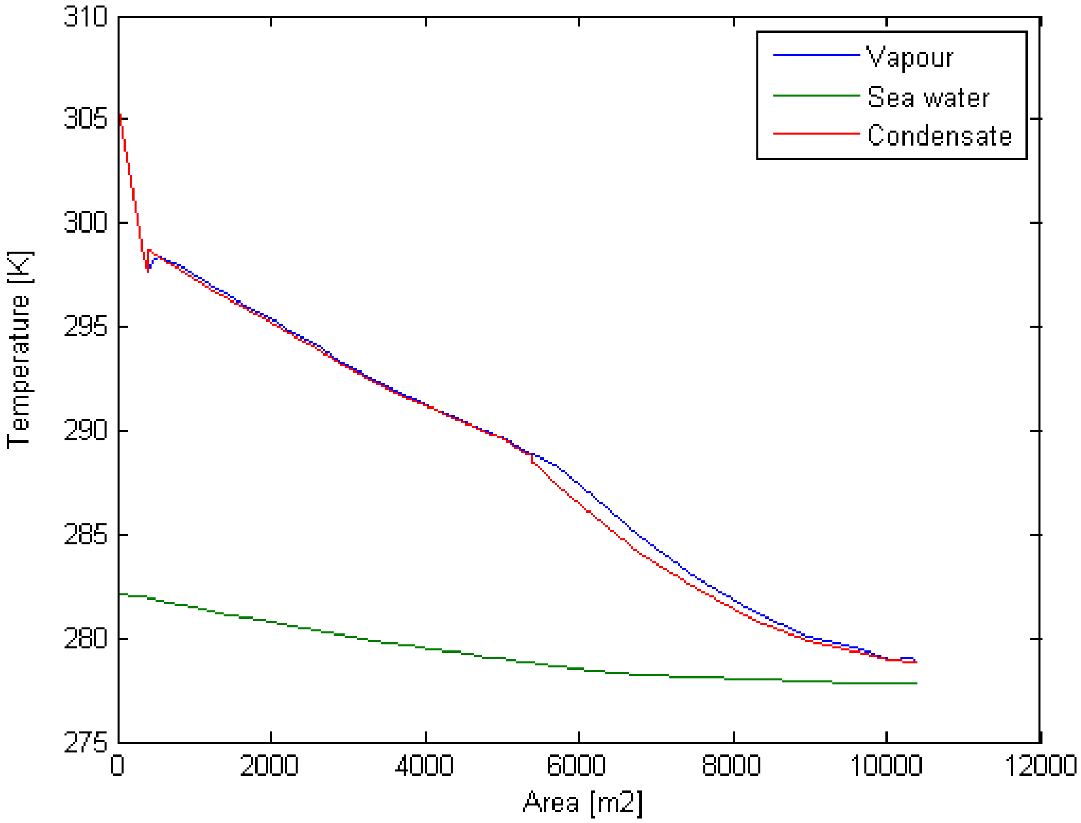


Figure 4.28: Temperature profiles throughout condenser

Traces of other components in vapour stream

Stream	Ethane, with traces	Ethane, without traces	Difference	Propane, with traces	Propane, without traces	Difference
1a	1.60	1.58	$1.53{\cdot}10^{-2}$	$6.12{\cdot}10^{-1}$	$6.04{\cdot}10^{-1}$	$7.89{\cdot}10^{-3}$
2	1.34	1.34	$1.50{\cdot}10^{-4}$	$4.69{\cdot}10^{-1}$	$4.69{\cdot}10^{-1}$	$-1.15{\cdot}10^{-4}$
3	1.33	1.33	$1.34{\cdot}10^{-4}$	$4.64{\cdot}10^{-1}$	$4.65{\cdot}10^{-1}$	$-1.06{\cdot}10^{-4}$
4	1.32	1.32	$1.20{\cdot}10^{-4}$	$4.60{\cdot}10^{-1}$	$4.60{\cdot}10^{-1}$	$-9.79{\cdot}10^{-5}$
5	1.31	1.31	$1.07{\cdot}10^{-4}$	$4.56{\cdot}10^{-1}$	$4.56{\cdot}10^{-1}$	$-9.03{\cdot}10^{-5}$

Table 4.11: K-values from Unisim, for streams with and without traces of Butane, together with the difference between the two

As can be seen in Table 4.11, the k-values change when adding a trace of butane in the stream. The k-values, or distribution coefficients, give the ratio between the molfraction in vapour and the molfraction in the liquid. The difference between the k-values is largest for the first streams.

Chapter 5

Discussion

5.1 Choice of Equations

The choice of equations will affect the way the model behaves and how well the calculations agree with the measured values. In this thesis it was chosen to use the film method for modelling the process. This was done to get a better picture of what happens inside the condenser. As MATLAB was used for the calculations, a system of many non-linear equations could be solved. According to Webb et al. [1999], the film method is better than the equilibrium method, as this allows prediction of the flux of the individual components. As mentioned in Section 2.4, the equilibrium method is not based on a sound theory for mass and heat transfer, whereas the film method is more in accordance with theory.

When building the model, it was chosen to simplify the flow by a plug flow model based on Zhang et al. [2010]. The model does not take into account the spiral flow of the vapour stream and multiple tubes for the sea water. As a consequence of this, the length the two streams traverse will be different. To compensate for this, it was chosen to use the area as independent variable instead of the length. This choice can also be questioned, as the sea water temperature in one tube at a specified length will be different than in another tube situated directly above or below the first tube. This could have been incorporated in the model in the same manner as Karlsson and Vamling [2005a]. This would result in a much more detailed grid, requiring more computational time.

Equation 3.9 was chosen for determining the mass flux. It was found that this equation worked best at the time. It was also chosen to model the differential section in the same way as the integral section, by changing the definition of mols in the condensate to the fluxes. This was done for the same reason as the choice of total flux, as this proved to be the most stable method. However the instabilities could have arisen from faults in the model that were found later. Equation 3.7 was tested and found to work to a certain extent. The fact that Equation 3.9 worked with faults in the model, indicates a more forgiving model. However the results of the faulty model were not sound, leading to the discovery of the faults.

The pressure drop throughout the condenser was assumed to be linear, and calculated from either the design pressure drop or the measured pressure drop. There are correlations for determining the pressure drop, but this would have increased the computational time for the model. Webb et al. [1999] calculate the pressure drop from measured values, thus the same procedure was used in this work.

When the model was examined towards the end of the project, it was found that the program ”dewpoint.m” did not correct for the pressure variations. This will lead to a wrong dew-point, especially if the pressure is far from the pressure used in ”dewpoint.m” (20 bars). The deviation between the dew-point at 20 bars and 21 bars was found to be only 3 degrees. The equilibrium calculations do however correct for the pressure changes and this error is thought to give only instabilities in the transition between the cooling section and the integral condensation section.

Karlsson and Vamling [2005b] discuss the effects of diffusional resistance in the liquid phase. As mentioned in Section 2.3, an inclusion of mass transfer resistance in the condensate might be important. However, the simplified method used in, amongst others Webb et al. [1999], does not include this and thus it was not included in this thesis.

As mentioned in Section 3.2, the heat transfer coefficients were not calculated, but set to a specific value. The heat transfer coefficients will vary throughout the condenser, as the flow, composition and temperature change. To improve the model, these differences should be included. Correlations for the vapour side are presented in Section 2.5, but were not implemented in the model. This would give a more accurate heat transfer coefficient. The correlations presented in Section 2.5 are unfortunately not complete, and it was thus decided to use a fixed value. When constructing the model, calculating the heat transfer coefficient was not as important. When performing the sensitivity analysis, the effects of the different parameters on the heat transfer coefficient are not included. However, when performing the sensitivity analysis for the heat transfer coefficient, the profiles do not change much with increase of the coefficient. It was thus deemed good enough for these purposes.

Resulting from these simplifications, the model might be too coarse to describe the process adequately. It is, however, an improvement from the equilibrium method, as it gives a clearer picture of the course in the condenser. When dealing with computational fluid dynamics (CFD), there will always be a trade-off between the detail of the model and the computational resources needed to solve the system.

5.2 MATLAB profiles

The profiles obtained from the MATLAB-model presented in Section 4.1 will be discussed in this section.

The zeotropic cooling curve one would expect is obtained with the model. When the condensation initiates, the temperature decrease will not be as steep as for the cooling section, but there will be a temperature decrease. In Figure 4.1 it can be seen that there is a temperature increase at the start of the condensation. This occurs as the vapour condenses and releases the heat, where the heat transfer rate is not large enough to transport this heat to the cooling medium. The interface temperature situated between the vapour and sea water temperature is in accordance with what one would expect. The only deviation is in the beginning of the integral condensation section. Here the interface temperature is situated above the vapour temperature. This can be due to the model initiating condensation at the moment the vapour temperature reaches the dew point. The interface temperature should reach the dew point before the bulk vapour reaches this temperature and condensation should initiate at an earlier stage. This is an extra challenge when the vapour is superheated, and gave rise to McNaught's modification as explained in Section 2.4. Another reason might be the lack of pressure correction in the dewpoint calculations. However, the interface temperature decreases rapidly and is situated below the vapour before 2000 m^2.

The characteristic temperature glide for zeotropic mixtures occurs as the condensation rate is different for the two components, confirmed by the flux profiles in Figure 4.2. Propane, being the less volatile component, will initially condense in a greater amount than ethane, depleting the vapour of propane. As the vapour is enriched with ethane, the dew-point is shifted towards the right in Figure 3.2, giving a lower temperature for condensation. This should, according to Webb et al. [1999], give a lower temperature driving force in the cooling curve. This behaviour can also be seen in Figure 4.1, as the temperature of the vapour approaches the interface temperature at the end of the integral condensation zone. The lower driving force will in turn give a lower condensation rate. When evaluating Equation 3.9, a smaller temperature difference in the first term would result in a lower flux. The described behaviour is observed in Figure 4.2. For the differential condensation zone, the amount of condensate, N_l, used in Equation 3.9 will be smaller, affecting the condensation rates.

As noted in Section 4.1, the vapour molfractions in the differential section will change more dramatically than the liquid molfractions. When changing from integral to differential condensation, the flux of propane increases. This increase will result in a faster depletion of propane in the vapour for the differential section than is observed in the integral section, giving the observed changes in vapour molfractions. The stability of the liquid molfractions can be a result of the larger amount of condensate present. As the amount of vapour decreases, small changes in condensation will give big changes in the molfraction.

5.3 Validation of Model

When the model was built, it had to be validated to make sure it gave good results. This section will concentrate on the discussion of the validation methods in Sections 3.1 and 3.7 with the results presented in Sections 4.2 and 4.3.

5.3.1 Validation against Unisim Model

Firstly, the model was validated against the model built in the commercial simulator, Unisim. When comparing the results from these two models, it can be seen that they give approximately the same results and profile for both temperature and composition. The Unisim results were placed at the positions that corresponded to the same position as the points taken from the MATLAB model, and not at the indicated area specifications. However, the temperature and composition correspond for the two models, strengthening the MATLAB-model.

5.3.2 Validation against Sensitivity Analysis

Another way to validate the model is to perform a sensitivity analysis and assess the variations in the results. This was done, both to validate the model, but also to examine the stability of the model and its usefulness in state monitoring.

Five parameters were used in the sensitivity analysis; vapour heat transfer coefficient, fouling resistance, vapour mass flow, sea water temperature and initial vapour composition. The pressure and sea water mass flow were not included in this analysis. By increasing or decreasing the sea water mass flow, Equation 3.14 would be affected. A change in the sea water temperature gradient would influence the other equations, but would give a similar result as a change in the sea water temperature. A change in the pressure results in a change in the dew point and temperature profile through the condenser. It was chosen not to perform this sensitivity analysis, as an analysis on the thermodynamic model was performed by varying the initial composition.

Variation in vapour heat transfer coefficient

The results obtained by varying the vapour heat transfer coefficient,h show that the different sections respond differently with the increase of the heat transfer coefficient. The same value for h for both the integral and the differential condensation section were used. The increase of h seems to affect the differential condensation area the most. h will affect how fast the heat is transferred from the vapour to the interface. A higher heat transfer coefficient will result in a smaller area required to reach a certain temperature. As A_3 is the only area that is directly linked to a specific outlet temperature and the vapour side heat transfer coefficient, this area will be affected the most. This corresponds with the results obtained in Figure 4.9, as the total amount of vapour left at 283 K will be higher for a higher heat transfer coefficient. With a greater amount of vapour left, the depletion of propane has not carried on as far and the molfraction of ethane will be higher.

The total heat transfer coefficient in the cooling section was not adjusted in this test. The reason the area A_1 changes slightly with a change in h, might be due to slight variations in the sea water temperature profiles for the different cases. This will give a change in the temperature driving force. The area A_2 decreases also with increase in h, but not as drastically as the increase in A_3. As A_2 is defined by the vapour percent, this area will not be as affected.

Variation in mass flow rate

When comparing the results for the different mass flow rates, it can be seen that all the areas increase linearly. This is a sound result, as an increase in mass flow results in more vapour for condensing and thus more area for heat transfer is required. A_1 is directly linked to the amount of vapour present, and will thus experience the greatest change. The differential condensation is only dependent on the local fluxes, thus the total amount of condensate will not influence the condensation in A_3. However, 12 weight% vapour will be a greater amount of vapour for the cases with the greater load. This will increase the demand for surface area for condensing and cooling the vapour down to 283 K. The total amount of mols left at the outlet increases linearly with the increase of mass flow. The molfraction of ethane at outlet does not vary significantly with the increase of mass flow, indicating that it is mostly the amount of area required that is affected by this parameter.

The temperature approach will increase with an increased mass flow, as the total amount of heat necessary to transfer from the vapour to the sea water is higher with a higher mass flow. The vapour temperature for a greater mass flow will thus be higher at a specified position than a smaller mass flow.

Variation in Sea Water temperature

The increase in sea water temperature results in an increase in all three areas. The increase in sea water temperature gives a decrease in temperature driving force. The capacity for taking up the heat is lower, and more area is needed to transfer the heat, giving a lower condensation flux. Because of this all three areas will increase in size. With the lowered temperature drive, less vapour will condensate out, giving a higher amount of mols left in the vapour at the outlet. This will also give an enrichment of ethane in the vapour with an increase of sea water temperature. These trends are shown in Figures 4.13 and 4.14.

The temperature approach depicted in Figure 4.15 shows a slight difference in trend than the other temperature approaches. The temperature approach increases from 279 K to 279.5 K, before it decreases. The other temperature approaches presented in Section 4.3 also show an increase before they flatten out. As mentioned in Section 4.3, the temperature approach was taken at a specific area in the condenser and not at the outlet (283 K). By comparing the temperature at a specific area, the degree of differential condensation will be different for the different cases. When the differential condensation starts, the vapour temperature decreases faster than in the integral condenser section. This can be seen in Figure 4.16. The case were the sea water temperature is 279 K gives a temperature approach of 4 K at 18 500 m^2, whereas the other cases have a temperature approach of approximately 6 K at the same area. However, if the temperature approach was investigated at
15 000 m^2, the case with 279 K gives a higher temperature approach than the others. If the four cases were compared at the 20 000 m^2, the sequence might be opposite than the one at 15 000 m^2. The temperature profiles might also flatten out at the end, giving less difference. This has not been investigated.

Variation in Fouling Resistance

An increase in the fouling resistance gives a decrease in the heat transfer coefficient, and a lower condensation rate according to Equation 3.9. A decreased condensation rate gives an increase in the required area for both differential and integral condensation, as can be seen in Figure 4.17. The area required for cooling down to the dew-point will also increase as the heat transfer rate is lower. The increase in area is highest for the integral condensation, indicating that this part will be more sensitive to fouling than the other parts.

The outlet conditions show the same trends as for the other analysis, indicating that an increase in fouling resistance will give a decreased efficiency for condensing the vapour and thus a lower fraction of ethane in the vapour. The temperature approach will also increase with increasing fouling resistance. This is in line with known practice, as the temperature approach is used to monitor the efficiency of the condenser.

Variation in Composition

The final parameter that was used in the sensitivity analysis was the composition. The change in composition resulted in new challengers for the model, thus only the areas A_1 and A_2 were calculated. However, it can be seen that a change in the composition will affect both areas, as the thermodynamic model will be shifted. A greater amount of ethane will result in a lower dew-point, thus increasing the size of A_1. The increase in A_2 might be due to the slightly lower interface temperature from the thermodynamic model, giving a lower condensation rate.

The fact that the model gave negative mol values for certain compositions is disturbing. This indicates that the model is not applicable for all situations.

Mixed Effects

The results from the mixed effects tests show no mixed effects from the sea water temperature or the mass flow and the fouling resistance. This indicates that the model should be able to distinguish between the effects of fouling and the effects from the process parameters, making it suitable to use in monitoring of the fouling resistance.

5.4 Experimental results versus Model

The aim of building the MATLAB-model was to make a set-up that would able the monitoring of fouling in a better way. To verify that the model was indeed capable of this, a comparison with experimental results was carried out. As can be seen in Figure 4.25, the experimental values were much lower than the values from MATLAB. This indicated that the heat transfer between the vapour and the sea water was much better than the one used in the original set-up. Thus the heat transfer coefficient U was increased from 709 W/m^{2}K to 1000 W/m^{2}K. The original value was taken from the design of the condenser, and should indicate a clean condenser. Thus an increase of U indicates an even higher heat transfer than the theoretical clean condenser. This can come from better heat transfer in both the condensate and the sea water than predicted, for instance due to high turbulence. The higher velocity will increase the convective heat transfer.

In Figure 4.26 the results after the adjustments were presented. The temperature profile from the MATLAB-model is still situated above the experimental results until approximately 8000 m^2. At this area, the two profiles match, giving the same temperature and composition profiles for the outlet. The total heat transfer coefficient was attempted increased to follow the experimental profile even better, but this resulted in a temperature cross in the differential section. From Figure 4.28 it can be seen that the temperature drive between the interface and vapour is quite small, indicating that the potential lies in the total heat transfer coefficient. An attempt to increase the heat transfer coefficient in only the integral section was done, but no good results were obtained. However, it seems that the heat transfer is greater in the initial phase of the condenser, thus making it probable that a heat transfer coefficient with a varying value should be implemented.

The fouling on 25-HA-112 is mostly biological fouling on the sea water side (Personal Correspondence with Geir Iversen, Statoil ASA, 2011) and the parameters affecting the growth of fouling should be correlated to the sea water. As presented in Section 2.2, the fouling increases with higher temperature and lower velocities. These conditions are found at the warm end of the condenser, indicating a thicker fouling layer and a lower U at the vapour inlet. This is supported by the pictures presented in Figure 2.2 and Figure 2.3, where it is clear that the fouling at the warm end, sea water outlet, is greater than at the cold end, sea water inlet.

As the composition and temperature values for the two models for the differential section are quite similar, the MATLAB-model seems to require a greater heat transfer area for the integral condensation. When the two profiles were compared, the experimental values were converted from length to area specific. This conversion might be too coarse, making it difficult to compare. In addition, the sea water and vapour will travel different lengths as explained in Section 5.1. Thus the area with the biggest gradients, from 0 to 8000 m^2, might be too inaccurately described to give the desired results.

Another explanation for these deviations can be the build-up of the model into three sections with the first section being cooling. As mentioned in Section 4.4, the interface temperature is situated above the vapour at the initial stage of the condensation. This could lead to the condensation initiating at a later stage, giving a shift in the profiles. If the model was modified to exclude the cooling section, it might give a better fit to the experimental data. This was not done in this thesis. Another reason can be the lack of pressure correction in the dew-point calculations. If the model was corrected to include the pressure adjustments, the experimental data and MATLAB results might coincide to a greater extent.

5.4.1 Significanse of trace components

As mentioned in Sections 3.8 and 4.4, the vapour does contain small amounts of other components. These components might influence the equilibrium giving different results than those obtained by a binary model. This might also be a reason for the experimental results. The k-values presented in Table 4.11, show that the distribution of propane and ethane in the vapour versus condensate varies with the addition of butane. The butane increases the k-values for ethane, giving a higher amount of ethane in the vapour. This could explain the trends shown in Figure 4.27. Vapour with slightly less ethane would give a slightly lower area as was discussed in Section 4.3. This could also explain the lower area required for condensing in the experimental results. Butane is less volatile than ethane, and the small changes in composition might give significant changes that can explain the trends shown in Figure 4.26.

5.5 Further Work

The model developed in MATLAB does not describe the processes in the condenser based on the experimental profiles in a sufficient way. There are several adjustments that could be made to improve the model:

Firstly, a correlation for the heat transfer coefficient could be implemented. Thus the coefficient could be corrected for the different process parameters, giving a more detailed picture of the heat transfer.

Another improvement would be to model the helical flow of the vapour and the independent tubes. This would require a more detailed grid, and the computational time would increase, but the outcome might be a model that is more in accord with the experimental measurements.

The effect of ommitting the cooling section could be investigated. The significance of the fault in the dew-point calculations could also be of intereset. These are minor improvements of the model that might give significant effects.

As mentioned in Section 5.4, the trace components might be of significance for the model. Thus a multi-component model might be necessary to describe the process correctly. This would require a completely new set-up and thermodynamic model.

A fully functional model would be applicable to estimate the fouling of the condenser based on historical data. The model does show the variations expected for such a condenser, making it possible to use with historical data.

Chapter 6

Concluding Remarks

The MATLAB model, based on the film method, seems to give satisfactory results. The profiles obtained for temperature, pressure, fluxes and compositions make it easier to understand the progress through the condenser.

The temperature and composition profilese obtained in the MATLAB model are quite similar to the profiles obtained in Unisim.

The sensitivity tests give results in accordance with theory, strengthening the models reliability.

The experimental results for condenser 25-HA-112 did not coincide completely with the results obtained from the model. This can be due to a too-coarse model, traces of other components, a significantly higher heat transfer coefficient than used in this project or lack of pressure correction in the dew-point calculation.

References

K.J. Bell and M.A. Ghaly. An approximate generalised design method for multicomponent partial condensers. In *AlChE Symp 69*, volume 131, pages 72–79, 1972.

Tilmann W. Botsch and Karl Stephan. A model to describe the steady-state and dynamic behaviour of vertical condensers fed by mixtures. *Revue Générale de Thermique*, 36 (6):481–491, 1997.

T. R. Bott. *Biological growth on heat exchanger surfaces*, pages 223–267. Elsevier Science B.V., Amsterdam, 1995.

W.G. Characklis. Biofouling:effects and control. proceedings of the international workshop on biofouling and biocorrosion. Springer Verlag, 1991.

W.G. Characklis and K.C. Marshall. *Biofilms.* John Wiley, New York, 1990.

T.H. Chilton and A. P. Colburn. Mass transfer (absorption) coefficients, predictions from data on heat transfer and fluid friction. *Industrial & Engineering Chemistry*, 16 (11):1183–1187, 1934.

A. P. Colburn and O. A. Hougen. Design of cooler condensers for mixtures of vapors with noncondensing gases. *Industrial & Engineering Chemistry*, 26(11):1178–1182, 1934.

Edward N. Fuller, Paul D. Schettler, and J. Calvin Giddings. New method for prediction of binary gas-phase diffusion coefficients. *Industrial & Engineering Chemistry*, 58(5): 18–27, 1966.

Kirk D. Hagen. *Heat Transfer with Applications.* Prentice Hall, New Jersey, USA, 1999.

Geir Iversen, 2011. Personal Correspondence with Geir Iversen at Statoil.

Hugo A. Jakobsen. *Chemical Reactor Modeling - Multiphase Reactive Flows.* Springer, Trondheim, 2008.

Kristin Løbach Jordhøy and Karianne Thorsen Stræte. Kondensasjon av blandinger i en helix-varmeveksler, 2009. Master thesis, NTNU.

Tord Karlsson and Lennart Vamling. Flow field in shell-and-tube condensers: comparison of a pure refrigerant and a binary mixture. *International Journal of Refrigeration*, 28:706–713, 2005a.

Tord Karlsson and Lennart Vamling. Suprising effects of combined vapour and liquid mass transfer resistances when condensing a mixture outside tube banks. *International Journal of Heat and Mass Transfer*, 48:403–412, 2005b.

R. Krishna, C.B. Panchal, D. R. Webb, and I. Coward. An ackerman-colburn and drew type analysis for condensation of multicomponent mixture. *Letters in Heat and Mass Transfer*, 3:163–172, 1976.

Linde AG. *Process Engineering Data Sheet 25-HA-112*, 2005. Document Number: E066-AB-P-DP-2545.

J. Lutcha and J. Nemcansky. Performance improvement of tubular heat exchangers by helical baffles. *Trans IChemE*, 15(1):55–65, 1990.

J. Marriot. Condensation of mixed vapors - integral vs. differential. *Heat Transfer Engineering*, 10(4):50–53, 1989.

D. E. Matschke and George Thodos. Vapor-liquid equilibria for the ethane-propane system. *Journal of Chemical & Engineering Data*, 7(2):232–234, 1962.

J. McNaught. An assessment of design methods for condensation of vapours from a non-condensing gas. In *ICHMT Seminar on Advances in Heat Exchangers*, 1981.

Zakia Nasri and Housam Binous. Applications of the peng-robinson equation of state using matlab. *Chemical Engineering Education*, 43(2), 2009.

E. Nebot, J.F. Casanueva, T. Casanueva, and D. Sales. Model for fouling deposition on power plant steam condensers cooled with seawater: Effect of water velocity and tube materia. *International Journal of Heat and Mass Transfer*, 50:3351–3358, 2007.

J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Society for Industrial and Applied Mathematics, 2000.

Ding-Yu Peng and Donald B. Robinson. A new two-constant equation of state. *Industrial & Engineering Chemistry Fundamentals*, 15(1):59–64, 1976.

Robert H. Perry and Don W. Green. *Perry's chemical engineers' handbook*. McGraw-Hill, New York, 8 edition, 2008.

J. Pettersen, A.O. Fredheim, and H. Ormbostad. Experience with water cooling mixed refrigerant condensation. Technical report, Statoil ASA, 2010.

Dave Sajjan, Tord Karlsson, and Lennart Vamling. Reasons for drop in shell-and-tube condenser performance when replacing r22 with zeotropic mixtures. part 1. analysis of experimental findings. *International Journal of Refrigeration*, 27(5):552–560, 2004.

L. Silver. Gas cooling with aqueous condensation. *Trans IChemE*, 25:30–42, 1947.

R.K. Sinnot. *Chemical Engineering Design*, volume 6 of *Coulson & Richardson's Chemical Engineering Series*. Elsevier, Oxford, UK, 4 edition, 2005.

Sigurd Skogestad. *Prosessteknikk*. Tapir Akademiske Forlag, Trondheim, 2 edition, 2003.

Statoil. Veien til LNG. `www.statoil.com/no/OurOperations/pipelines/Documents/veien_til_lng_norsk_enkel.swf` (Accessed on December 4th, 2010), 2005.

Statoil. Snøhvit. `www.statoil.com/en/ouroperations/explorationprod/ncs/snoehvit/Pages/default.aspx` (Accessed on November 23th, 2010), 2007.

P. Stehlik, J. Nemcansky, D. Kral, and L. W. Swanson. Comparison of correction factors for shell-and-tube heat exchangers with segmental or helical baffles. *Heat Transfer Engineering*, 15(1):55–65, 1994.

D. R. Webb and J. S. Kim. A modification to the equilibrium model for condensation of vapours of immiscible liquids. *Chemical Engineering Research and Design*, 77(2): 110–116, 1999.

D. R. Webb, M. Fahrner, and R. Schwaab. The relationship between the colburn and silver methods of condenser design. *International Journal of Heat and Mass Transfer*, 39(15):3147–3156, 1996.

D.R. Webb, A.J. Dell, R.W. Stevenson, and K. Ax. Assessment of condensation curves in the design of binary vapour condensers. *Trans IChemE*, 77, 1999.

Jian-Fei Zhang, Ya-Ling He, and Wen-Quan Tao. A design and rating method for shell-and-tube heat exchangers with helical baffles. *Journal of Heat Transfer*, 132(5): 051802, 2010.

Zhengguo Zhang, Dabin Ma, Xiaoming Fang, and Xuenong Gao. Experimental and numerical heat transfer in a helically baffled heat exchanger combined with one three-dimensional finned tube. *Chemical Engineering and Processing: Process Intensification*, 47(9-10):1738–1743, 2008.

Appendices

Appendix A

MATLAB code

A.1 Thermodynamics

flash.m

%This function is based on the code written by Nasri and Binous [2009],
%and has only been modified to fit data for ethane and propane
%together with setting the pressure.
% critical temperature and pressure and acentric factor
% for Ethane and Propane, obtained form Matschke and Thodos [1962]

```
function f=flash(x)
global z
phi=0;
% critical temperature and pressure and acentric factor
% for Ethane and Propane
Pc=[48.72 42.48];
Tc=[305.32 369.83];
w=[0.099 0.152];
% pressure is set equal to 20 bars
P=20;
% reduced temperature and pressure
Tre=x(5)./Tc;
Pre=P./Pc;
% m, a, Ai, Bi, Aij, A, B for the PR EOS
m=0.37464 + 1.54226.*w-0.26992.*w.^2;
a=(1+m.*(1-Tre.^0.5)).^2;
```

```
Ap=0.45724.*a.*Pre./Tre.^2;
Bp=0.07780.*Pre./Tre;

for i=1:2
    for j=1:2
        Ab(i,j)=(Ap(i)*Ap(j))^0.5;
    end
end
Av=0;
for i=1:2
    for j=1:2
        Av=Av+x(i+2)*x(j+2)*Ab(i,j);
    end
end
Bv=0;
for i=1:2
    Bv=Bv+x(i+2)*Bp(i);
end
Bl=0;
for i=1:2
    Bl=Bl+x(i)*Bp(i);
end
Al=0;
for i=1:2
    for j=1:2
        Al=Al+x(i)*x(j)*Ab(i,j);
    end
end
Alsum=[0 0];
for i=1:2
    for j=1:2
        Alsum(i)=Alsum(i)+x(j)*Ab(i,j);
    end
end
Avsum=[0 0];
for i=1:2
    for j=1:2
        Avsum(i)=Avsum(i)+x(j+2)*Ab(i,j);
    end
end

% liquid and gas phase compressibility factors
Zv=max(roots([1 -1+Bv Av-3*Bv^2-2*Bv -Av*Bv+Bv^2+Bv^3]));
Zl=min(roots([1 -1+Bl Al-3*Bl^2-2*Bl -Al*Bl+Bl^2+Bl^3]));
```

```
% vapor and liquid phase fugacity coefficients
phiv=exp((Zv-1).*Bp/Bv-log(Zv-Bv)...
-Av/(2*sqrt(2)*Bv)*log((Zv+(1+sqrt(2))*Bv)/...
(Zv+(1-sqrt(2))*Bv)).*(2.*Avsum./Av-Bp./Bv));
phil=exp((Zl-1).*Bp/Bl-log(Zl-Bl)...
-Al/(2*sqrt(2)*Bl)*log((Zl+(1+sqrt(2))*Bl)/...
(Zl+(1-sqrt(2))*Bl)).*(2.*Alsum./Al-Bp./Bl));
% equilibrium constant
K=phil./phiv;
% the system of five algebraic equations
for i=1:2
    f(i)=x(i+2)-K(i)*x(i);
end
for i=1:2
    f(i+2)=x(i)-z(i)/(1+phi*(K(i)-1));
end
f(5)=0;
for i=1:2
    f(5)=f(5)+z(i)*(K(i)-1)/(1+phi*(K(i)-1));
end
```

vlecurve.m

%This function is based on the code written by Nasri and Binous [2009], and has not
%been modified

```
clc
global z
clear sol
% flash calculation using fsolve and a zeroorder
%collocation method
z=[0.0001 0.9999];
options = optimset('Display','off');
[X]=fsolve(@flash,[0.01 0.9 0.01 0.9 360],options);
x0=X;
sol(1,1)=X(1);
sol(2,1)=X(3);
sol(3,1)=X(5);

Z(:,1)=z';
for i=1:100
```

```
    z=[0.01*i 1-0.01*i];
    [X]=fsolve(@flash,x0,options);
    x0=X;
    sol(1,i+1)=X(1);
    sol(2,i+1)=X(3);
    sol(3,i+1)=X(5);
    Z(:,1+i)=z';
end
% plotting bubble curve
h=plot(sol(1,:),sol(3,:),'b');
set(h,'linewidth',2)
hold on
% plotting due curve
h=plot(sol(2,:),sol(3,:),'r');
set(h,'linewidth',2)
axis tight
legend ('bubble␣curve', 'due␣curve');
xlabel('Ethane␣vapor␣or␣liquid␣mole␣fraction')
ylabel('temperature␣in␣K')
grid on
hold off
```

A.2 Modell for Condenser

condenser4.m

```
%This script will model the condenser using collocation

clear all
%Decide internal collocation points
n=3; %For cooling
m=25; %For integral condensation
o=10; %For differential condensation
[xn, An, Bn, qn] = colloc( n ,1 ,1);
[xm, Am, Bm, qm] = colloc( m ,1 ,1);
[xo, Ao, Bo, qo] = colloc( o ,1 ,1);

%Import constants
constant

%The starting values are from table 5.1 in \cite{master09}
```

```
T0= 31.2 + 273; % K
P0= 20.740*10^5; % Pa
Ta0=6.5 + 273;% K
L0=0; % Liquid molar flow [mol vapour/s]
Vm0=478.31; %vapour mass flow [kg/s]
yeth0=0.64;
yprop0=0.36;
u0=0.61;      %velocity [m/s]

M0=yeth0*Meth+Mprop*yprop0; %molar mass of vapour feed
V0=Vm0/M0; %Calculate vapour molar flow [kmol/s]

ntot=V0;
Nev0=yeth0*V0;
Npv0=yprop0*V0;
Nel0=0.00041;
Npl0=0.000595;

%Calculate dewpoint
Tdew=dewpoint(yeth0);

%load guess values for collocation points
load y_0305_cond3_1;



t=(n+2)*3+(m+2)*7+(o+2)*7;

t1=n+2;
t2=m+2;
t3=o+2;

%guess values for collocation points if good guess values were not
%available

% y=ones(t+2,1);
%
% y(1:t1)=T0.*y(1:t1);
% y(t1+1:2*t1)=P0.*y(t1+1:2*t1);
% y(2*t1+1:3*t1)=Ta0.*y(2*t1+1:3*t1);
% % y(3*t1+1:4*t1)=u0.*y(3*t1+1:4*t1);
%
%
```

```
% y(3*t1+1:3*t1+t2)=Tdew.*y(3*t1+1:3*t1+t2);
% y(3*t1+t2+1:3*t1+2*t2)=P0.*y(3*t1+t2+1:3*t1+2*t2);
% y(3*t1+2*t2+1:3*t1+3*t2)=Ta0.*y(3*t1+2*t2+1:3*t1+3*t2);
% y(3*t1+3*t2+1:3*t1+4*t2)=Nev0.*y(3*t1+3*t2+1:3*t1+4*t2);
% y(3*t1+4*t2+1:3*t1+5*t2)=Npv0.*y(3*t1+4*t2+1:3*t1+5*t2);
%
% % y(4*t1+1:4*t1+t2)=Tdew.*y(4*t1+1:4*t1+t2);
% % y(4*t1+t2+1:4*t1+2*t2)=P0.*y(4*t1+t2+1:4*t1+2*t2);
% % y(4*t1+2*t2+1:4*t1+3*t2)=Ta0.*y(4*t1+2*t2+1:4*t1+3*t2);
% % y(4*t1+3*t2+1:4*t1+4*t2)=Nev0.*y(4*t1+3*t2+1:4*t1+4*t2);
% % y(4*t1+4*t2+1:4*t1+5*t2)=Npv0.*y(4*t1+4*t2+1:4*t1+5*t2);
% % y(4*t1+7*t2+1:4*t1+8*t2)=u0.*y(4*t1+7*t2+1:4*t1+8*t2);
% % y(end)=715;
%
% y(3*t1+7*t2+1:3*t1+7*t2+t3)=Tdew.*y(3*t1+7*t2+1:3*t1+7*t2+t3);
% y(3*t1+7*t2+t3+1:3*t1+7*t2+2*t3)=...
%         P0.*y(3*t1+7*t2+t3+1:3*t1+7*t2+2*t3);
% y(3*t1+7*t2+2*t3+1:3*t1+7*t2+3*t3)=...
%         Ta0.*y(3*t1+7*t2+2*t3+1:3*t1+7*t2+3*t3);
% y(3*t1+7*t2+3*t3+1:3*t1+7*t2+4*t3)=...
%         Nev0.*y(3*t1+7*t2+3*t3+1:3*t1+7*t2+4*t3);
% y(3*t1+7*t2+4*t3+1:3*t1+7*t2+5*t3)=...
%         Npv0.*y(3*t1+7*t2+4*t3+1:3*t1+7*t2+5*t3);
% y(t+1)=600.*y(t+1);
% y(t+2)=6000.*y(t+2);
%
%Initial values
y0=[T0 P0 Ta0 u0];

%Define options
opt=optimset('Display','iter','MaxFunEvals',50000,'MaxIter',100000);
%Call up fsolve
y=fsolve(@(y) condensermodell3(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,...
    y0,Nel0,Npl0,Nev0,Npv0),y,opt);

%Unpack y-vector
T(1:t1)=y(1:t1);
T(t1+1:t1+t2)=y(3*t1+1:3*t1+t2);
T(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+1:3*t1+7*t2+t3);
T=T';
P(1:t1)=y(t1+1:2*t1);
P(t1+1:t1+t2)=y(3*t1+t2+1:3*t1+2*t2);
P(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+t3+1:3*t1+7*t2+2*t3);
P=P';
```

```
Ta(1:t1)=y(2*t1+1:3*t1);
Ta(t1+1:t1+t2)=y(3*t1+2*t2+1:3*t1+3*t2);
Ta(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+2*t3+1:3*t1+7*t2+3*t3);
Ta=Ta';
Nev(1:t1)=Nev0.*ones(1,n+2);
Nev(t1+1:t1+t2)=y(3*t1+3*t2+1:3*t1+4*t2);
Nev(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+3*t3+1:3*t1+7*t2+4*t3);
Nev=Nev';
Npv(1:t1)=Npv0.*ones(1,n+2);
Npv(t1+1:t1+t2)=y(3*t1+4*t2+1:3*t1+5*t2);
Npv(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+4*t3+1:3*t1+7*t2+5*t3);
Npv=Npv';

Nev0v=Nev0*ones(1,m+2+n+2+o+2)';
Nel0v=Nel0*ones(1,m+2+n+2+o+2)';
Npv0v=Npv0*ones(1,m+2+n+2+o+2)';
Npl0v=Npl0*ones(1,m+2+n+2+0+2)';

%Calculate moles in condensate
Nel=Nev0v-Nev;
Npl=Npv0v-Npv;

nt(1:t1)=zeros(1,t1);
nt(t1+1:t1+t2)=y(3*t1+5*t2+1:3*t1+6*t2);
nt(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+5*t3+1:3*t1+7*t2+6*t3);
nt=nt';
ne(1:t1)=zeros(1,t1);
ne(t1+1:t1+t2)=y(3*t1+6*t2+1:3*t1+7*t2);
ne(t1+t2+1:t1+t2+t3)=y(3*t1+7*t2+6*t3+1:3*t1+7*t2+7*t3);
ne=ne';
np=nt-ne;
kp=y(t+3);

%Calculate molefractions
xE=Nel./(Nel+Npl);
xP=1-xE;
yE=Nev./(Nev+Npv);
yP=1-yE;

%Calculated areas
A1=y(t+1);
A2=y(t+2);
A3=y(t+4);
```

```
%Calculate plot points
xt(1:t1)=A1.*xn;
xt(t1+1:t1+t2)=A1+A2.*xm;
xt(t1+t2+1:t1+t2+t3)=A1+A2+A3.*xo;

%Save solution
save condensation_0206

%plot figures
figure(1)
plot(xt,T,xt,Ta)
title('Temperature')
legend('Vapour','Sea water')

figure(2)
plot(xt,Nev,xt,Npv,xt,Nel,xt,Npl)
title('Composition profile along condenser')
legend('Ethane vapour','Propane vapour','Ethane liquid',...
    'Propane liquid',-1);

figure(3)
plot(xt,P)
title('Pressure')
```

condensermodell3.m

```
function f=condensermodell3(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,...
    y0,Nel0,Npl0,Nev0,Npv0)

%Define variable z
z=n+2;

%Define variables for cooling
T1=sqrt(y(1:z).^2);
P1=sqrt(y(z+1:2*z).^2);
Ta1=sqrt(y(2*z+1:3*z).^2);


xs=3*z;
x=m+2;

%Define variables for integral condensation
T2=sqrt(y(xs+1:xs+x).^2);
P2=sqrt(y(xs+x+1:xs+2*x).^2);
```

```
Ta2=sqrt(y(xs+2*x+1:xs+3*x).^2);
Nev2=sqrt(y(xs+3*x+1:xs+4*x).^2);
Npv2=sqrt(y(xs+4*x+1:xs+5*x).^2);
nt2=y(xs+5*x+1:xs+6*x);
ne2=y(xs+6*x+1:xs+7*x);



xd=3*z+7*x;

xx=o+2;
%Define variables for differential condensation
T3=sqrt(y(xd+1:xd+xx).^2);
P3=sqrt(y(xd+xx+1:xd+2*xx).^2);
Ta3=sqrt(y(xd+2*xx+1:xd+3*xx).^2);
Nev3=sqrt(y(xd+3*xx+1:xd+4*xx).^2);
Npv3=sqrt(y(xd+4*xx+1:xd+5*xx).^2);
nt3=y(xd+5*xx+1:xd+6*xx);
ne3=y(xd+6*xx+1:xd+7*xx);

%Define variables for area
A1=y(xd+7*xx+1);
A2=y(xd+7*xx+2);
kp=y(xd+7*xx+3);
A3=y(xd+7*xx+4);

%load constant
constant;

%Define heat transfer coefficients
hg=300;
Us=500;
U=709.5;
hg3=300;



%Cooling section
%boundry conditions at xn=0;
f(1)=T1(1)-y0(1,1); %
f(z+1)=P1(1)-y0(1,2); %
f(2*z+1)=An(1,:)*Ta1(:)+Us*A1*(T1(1)-Ta1(1))/(msw*cpsw); %Ta


%for n internal collocation points
```

```
for j=2:n+1
      f(j)=An(j,:)*T1(:)+Us*A1*(T1(j)-Ta1(j))/...
            ((Nev0+Npv0)*heatcapacity(T1(j),Nev0,Npv0));
      f(z+j)=An(j,:)*P1(:)+kp*A1;%*xn(j); %
      f(2*z+j)=An(j,:)*Ta1(:)+Us*A1*(T1(j)-Ta1(j))/(msw*cpsw);

end

%for xn=1
f(z)=An(n+2,:)*T1(:)+Us*A1*(T1(n+2)-Ta1(n+2))/...
      ((Nev0+Npv0)*heatcapacity(T1(n+2),Nev0,Npv0));
f(2*z)=An(n+2,:)*P1(:)+kp*A1;%
f(3*z)=Ta1(n+2)-Ta2(1);



%Integral condensation section

%Pre-define parameters
ack=ones(1,m+2)';
cpg=ones(1,m+2)';
beta=ones(1,m+2)';
teta=ones(1,m+2)';
hg0=ones(1,m+2)';
hvf=ones(1,m+2)';
yE=ones(1,m+2)';
Tint2=297.*ones(1,m+2)';

%Define moles in condensate and flux of propane for
%integral section
Nel2=Nev0*ones(1,m+2)'-Nev2+Nel0;
Npl2=Npv0*ones(1,m+2)'-Npv2+Npl0;
np2=nt2-ne2;

%Define moles in condensate and flux of propane for
%differential section
np3=nt3-ne3;
Nel3=ne3;
Npl3=np3;


%Boundry conditions for xm=1
f(xs+1)=T2(1)-T1(n+2);
f(xs+x+1)=P2(1)-P1(n+2);
```

```
f(xs+2*x+1)=Am(1,:)*Ta2(:)+U*A2*(T2(1)-Ta2(1))/(msw*cpsw);
f(xs+3*x+1)=Nev2(1)-Nev0;
f(xs+4*x+1)=Npv2(1)-Npv0;
f(xs+5*x+1)=nt2(1);
f(xs+6*x+1)=ne2(1);


%For m internal collocation points
    for i=2:m+1
        %calculate parameters
        ack(i)=ackermann(ne2(i),np2(i),hg,T2(i));
        cpg(i)=heatcapacity(T2(i),Nev2(i),Npv2(i));
        beta(i)=masstransfer2(T2(i),P2(i),hg,Nev2(i),Npv2(i));
        teta(i)=nt2(i)/beta(i);
        hg0(i)=hg*ack(i)/(exp(ack(i))-1);
        hvf(i)=(-ne2(i)*heatofcond(T2(i),1)-...
            np2(i)*heatofcond(T2(i),0))/nt2(i);
        yE(i)=Nev2(i)/(Nev2(i)+Npv2(i));
        Tint2(i)=Tinterface(Nel2(i),Npl2(i),P2(i));


        %Calculate residuals
        f(xs+i)=Am(i,:)*T2(:)+hg0(i)*A2*(T2(i)-Tint2(i))/...
            ((Nev2(i)+Npv2(i))*cpg(i));
        f(xs+x+i)=Am(i,:)*P2(:)+kp*A2;
        f(xs+2*x+i)=Am(i,:)*Ta2(:)+U*A2*(Tint2(i)-Ta2(i))/...
            (msw*cpsw);
        f(xs+3*x+i)=Am(i,:)*Nev2(:)+ne2(i)*A2;
        f(xs+4*x+i)=Am(i,:)*Npv2(:)+np2(i)*A2;
        f(xs+5*x+i)=nt2(i)-(U*(Tint2(i)-Ta2(i))-...
            heatcapacitycond(Tint2(i),Nel2(i),Npl2(i))*...
            (Nel2(i)+Npl2(i))*(Tint2(i-1)-Tint2(i))/A2)/hvf(i);
        f(xs+6*x+i)=ne2(i)-(beta(i)*teta(i)*(yE(i)-...
            yequilibrium(Nel2(i),Npl2(i),P2(i)))/...
            (exp(teta(i))-1)+yE(i)*nt2(i));

    end

%Boundry conditions at xm=1

    %Calculate parameters
    ack(m+2)=ackermann(ne2(m+2),np2(m+2),hg,T2(m+2));
    cpg(m+2)=heatcapacity(T2(m+2),Nev2(m+2),Npv2(m+2));
    beta(m+2)=masstransfer2(T2(m+2),P2(m+2),hg,Nev2(m+2),Npv2(m+2));
    teta(m+2)=nt2(m+2)/beta(m+2);
    hg0(m+2)=hg*ack(m+2)/(exp(ack(m+2))-1);
    hvf(m+2)=(-ne2(m+2)*heatofcond(T2(m+2),1)-np2(m+2)*...
        heatofcond(T2(m+2),0))/nt2(m+2);
    yE(m+2)=Nev2(m+2)/(Nev2(m+2)+Npv2(m+2));
    Tint2(m+2)=Tinterface(Nel2(m+2),Npl2(m+2),P2(m+2));


    %Calculate residuals
    f(xs+x)=Am(m+2,:)*T2(:)+hg0(m+2)*A2*(T2(m+2)-Tint2(m+2))/...
        ((Nev2(m+2)+Npv2(m+2))*cpg(m+2));
    f(xs+2*x)=Am(m+2,:)*P2(:)+kp*A2;
    f(xs+3*x)=Ta2(m+2)-Ta3(1);
    f(xs+4*x)=Am(m+2,:)*Nev2(:)+ne2(m+2)*A2;
    f(xs+5*x)=Am(m+2,:)*Npv2(:)+np2(m+2)*A2;
    f(xs+6*x)=nt2(m+2)-(U*(Tint2(m+2)-Ta2(m+2))-...
        heatcapacitycond(Tint2(m+2),Nel2(m+2),Npl2(m+2))*...
        (Nel2(m+2)+Npl2(m+2)-Nel0-Npl0)*...
        (Tint2(m+1)-Tint2(m+2))/A2)/hvf(m+2);
    f(xs+7*x)=ne2(m+2)-(beta(m+2)*teta(m+2)*...
        (yE(m+2)-yequilibrium(Nel2(m+2),Npl2(m+2),P2(m+2)))/...
        (exp(teta(m+2))-1)+yE(m+2)*nt2(m+2));



%Differential condensation
%Boundry conditions at xo=1

%Pre-define parameters
ack3=ones(1,o+2)';
cpg3=ones(1,o+2)';
beta3=ones(1,o+2)';
teta3=ones(1,o+2)';
hg03=ones(1,o+2)';
hvf3=ones(1,o+2)';
yE3=ones(1,o+2)';
Tint3=296.*ones(1,o+2)';

%Calculate residuals
f(xd+1)=T3(1)-T2(m+2);
f(xd+xx+1)=P3(1)-P2(m+2);
f(xd+2*xx+1)=Ao(1,:)*Ta3(:)+U*A3*(T3(1)-Ta3(1))/...
    (msw*cpsw); %Ta
f(xd+3*xx+1)=Nev3(1)-Nev2(m+2);
```

```
f(xd+4*xx+1)=Npv3(1)-Npv2(m+2);
f(xd+5*xx+1)=nt3(1)-nt2(m+2);
f(xd+6*xx+1)=ne3(1)-ne2(m+2);


%for o internal collocation points
    for i=2:o+1
        %Calculate parameters
        ack3(i)=ackermann(ne3(i),np3(i),hg3,T3(i));
        cpg3(i)=heatcapacity(T3(i),Nev3(i),Npv3(i));
        beta3(i)=masstransfer2(T3(i),P3(i),hg3,Nev3(i),Npv3(i));
        teta3(i)=nt3(i)/beta3(i);
        hg03(i)=hg3*ack3(i)/(exp(ack3(i))-1);
        hvf3(i)=(-ne3(i)*heatofcond(T3(i),1)-...
            np3(i)*heatofcond(T3(i),0))/nt3(i);
        yE3(i)=Nev3(i)/(Nev3(i)+Npv3(i));
        xE3(i)=Xequilibrium(Tint3(i),P3(i));
        Tint3(i)=Tinterface(Nel3(i),Npl3(i),P3(i));

        %Calculate residuals
        f(xd+i)=Ao(i,:)*T3(:)+hg03(i)*A3*(T3(i)-Tint3(i))/...
            ((Nev3(i)+Npv3(i))*cpg3(i));
        f(xd+xx+i)=Ao(i,:)*P3(:)+kp*A3;
        f(xd+2*xx+i)=Ao(i,:)*Ta3(:)+U*A3(1)*(Tint3(i)-Ta3(i))/...
            (msw*cpsw);
        f(xd+3*xx+i)=Ao(i,:)*Nev3(:)+ne3(i)*A3(1);
        f(xd+4*xx+i)=Ao(i,:)*Npv3(:)+np3(i)*A3(1);
        f(xd+5*xx+i)=nt3(i)-(U*(Tint3(i)-Ta3(i))...
            -heatcapacitycond(Tint3(i),Nel3(i),Npl3(i))*...
            (Nel3(i)+Npl3(i))*(Tint3(i-1)-Tint3(i))/A3)/hvf3(i);
        f(xd+6*xx+i)=ne3(i)-(beta3(i)*teta3(i)*(yE3(i)-...
            yequilibrium(Nel3(i),Npl3(i),P3(i)))/...
            (exp(teta3(i))-1)+yE3(i)*nt3(i));

    end

%Boundry conditions at xo=1
    %Calculate parameters
    ack3(o+2)=ackermann(ne3(o+2),np3(o+2),hg,T3(o+2));
    cpg3(o+2)=heatcapacity(T3(o+2),Nev3(o+2),Npv3(o+2));
    beta3(o+2)=masstransfer2(T3(o+2),P3(o+2),hg,Nev3(o+2),Npv3(o+2));
    teta3(o+2)=nt3(o+2)/beta3(o+2);
    hg03(o+2)=hg*ack3(o+2)/(exp(ack3(o+2))-1);
    hvf3(o+2)=(-ne3(o+2)*heatofcond(T3(o+2),1)-...
        np3(o+2)*heatofcond(T3(o+2),0))/nt3(o+2);
    yE3(o+2)=Nev3(o+2)/(Nev3(o+2)+Npv3(o+2));
    xE3(o+2)=Xequilibrium(Tint3(o+2),P3(o+2));
    Tint3(o+2)=Tinterface(Nel3(o+2),Npl3(o+2),P3(o+2));

    %Calculate residuals
    f(xd+xx)=Ao(o+2,:)*T3(:)+hg03(o+2)*A3*(T3(o+2)-Tint3(o+2))/...
        ((Nev3(o+2)+Npv3(o+2))*cpg3(o+2));
    f(xd+2*xx)=Ao(o+2,:)*P3(:)+kp*A3;
    f(3*xx+xd)=Ta3(o+2)-y0(1,3);
    f(4*xx+xd)=Ao(o+2,:)*Nev3(:)+ne3(o+2)*A3;
    f(5*xx+xd)=Ao(o+2,:)*Npv3(:)+np3(o+2)*A3;
    f(xd+6*xx)=nt3(o+2)-(U*(Tint3(o+2)-Ta3(o+2))-...
        heatcapacitycond(Tint3(o+2),Nel3(o+2),Npl3(o+2))*...
        (Nel3(o+2)+Npl3(o+2))*...
        (Tint3(o+1)-Tint3(o+2))/A3)/hvf3(o+2);
    f(xd+7*xx)=ne3(o+2)-(beta3(o+2)*teta3(o+2)*(yE3(o+2)-...
        yequilibrium(Nel3(o+2),Npl3(o+2),P3(o+2)))/...
        (exp(teta3(o+2))-1)+yE3(o+2)*nt3(o+2));

%Calculate constraints for each section
f(7*xx+xd+1)=T1(n+2)-Tdew;
Mv=Nev2(m+2)*Meth+Npv2(m+2)*Mprop;
Ml=Nel2(m+2)*Meth+Npl2(m+2)*Mprop;
vappros=Mv./(Mv+Ml);
f(7*xx+xd+2)=vappros-0.1237;
f(7*xx+xd+3)=P2(m+2)-2040*10^3;
f(7*xx+xd+4)=T3(o+2)-(283);
```

colloc.m

```
function [r, A, B, q]=colloc(n,left,right)
% colloc: Calculate collocation weights
%         [r, A, B, q] = colloc( n [,'left'] [,'right'])
% inputs:
%                 n - number of interior node points
%            'left' - include left boundary
%           'right' - include right bounary also
% outputs:
%            r - vector of roots
%            A - Matrix of first derivative weights
%            B - Matrix of second derivative weights
%            q - Quadrature weights.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Copyright (C) 1996, 1997 John W. Eaton
%
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 2, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
% General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with Octave; see the file COPYING.  If not, write to the Free
% Software Foundation, 59 Temple Place - Suite 330, Boston, MA
% 02111-1307, USA.
%
% Adapted from Octave's colloc.cc by Steve Swinnea.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n0 = 0 ; n1 = 0;
if (nargin > 1)
  if (strcmp(left,'left') | strcmp(left,'l') )
    n0 = 1;
  elseif (left == 0 | left == 1 )
    n0 = left;
  else
    error('Second␣argument␣should␣be␣the␣string␣left␣or␣l')
  end
end
if (nargin > 2)
  if (strcmp(right,'right') | strcmp(right,'r') )
    n1 = 1;
  elseif ( right == 1 | right == 0 )
    n1 = right;
  else
    error('Third␣argument␣should␣be␣the␣string␣right␣or␣r')
  end
end

[dif1,dif2,dif3,r]=jcobi(n,n0,n1,0,0);
q = dfopr(n,n0,n1,0,3,dif1,dif2,dif3,r);
for i=1:(n+n0+n1)
    vect = dfopr(n,n0,n1,i,1,dif1,dif2,dif3,r);
    A(i,:) = vect';
end
for i=1:(n+n0+n1)
    vect = dfopr(n,n0,n1,i,2,dif1,dif2,dif3,r);
    B(i,:) = vect';
end

%%%%%  jcobi  %%%%%%
function [dif1,dif2,dif3,root]=jcobi(n,n0,n1,alpha,beta)
if (n0 ~= 0) & (n0 ~= 1)
    error('**␣VILERR␣:␣Illegal␣value␣%␣␣N0␣');
end
if (n1 ~= 0) & (n1 ~= 1)
    error('**␣VILERR␣:␣Illegal␣value␣for␣N1␣');
end
if (n+n0+n1 < 1)
    error('**␣VILERR␣:␣Number␣of␣interpolation␣points␣less␣than␣1');
end
%
% -- FIRST EVALUATION OF COEFFICIENTS IN RECURSION FORMULAS.
% -- RECURSION COEFFICIENTS ARE STORED IN DIF1 AND DIF2.
%
nt = n+n0+n1;
dif1=zeros(nt,1);
dif2=zeros(nt,1);
dif3=zeros(nt,1);
root=zeros(nt,1);
ab = alpha+beta;
ad = beta-alpha;
ap = beta*alpha;
dif1(1) = (ad/(ab+2)+1)/2;
dif2(1) = 0;

if (n >= 2)
    for i=2:n
        z1 = i-1;
        z = ab + 2*z1;
        dif1(i) = (ab*ad/z/(z+2)+1)/2;
        if (i == 2 )
            dif2(i) = (ab+ap+z1)/z/z/(z+1);
        else
            z = z*z;
            y = z1*(ab+z1);
            y = y*(ap+y);
            dif2(i) = y/z/(z-1);
        end
    end
end
%
% -- ROOT DETERMINATION BY NEWTON METHOD WITH SUPPRESSION OF
% -- PREVIOUSLY DETERMINED ROOTS
%
x = 0;
for i=1:n
  z = 1;
  while ( abs(z) > 1e-9 )
    xd = 0;
    xn = 1;
    xd1 = 0;
    xn1 = 0;
    for j=1:n
        xp = (dif1(j)-x)*xn - dif2(j)*xd;
        xp1 = (dif1(j)-x)*xn1 - dif2(j)*xd1 - xn;
        xd = xn;
        xd1 = xn1;
        xn = xp;
        xn1 = xp1;
    end
    zc = 1;
    z = xn/xn1;
    if ( i ~= 1 )
        for j = 2:i
            zc = zc - z/(x-root(j-1));
        end
    end
    z = z/zc;
    x = x-z;
  end
  root(i) = x;
  x = x +.0001;
end
%
% -- ADD INTERPOLATION POINTS AT X = 0 AND/OR X = 1
%
if (n0 ~= 0)
    root = [ 0 ; root(1:nt-1) ];
end
if (n1 == 1)
```

```
    root(nt) = 1;
end
[dif1 dif2 dif3] = dif( root );

%%%% dfopr %%%%%
function vect = dfopr(n,n0,n1,i,id,dif1,dif2,dif3,root)
nt = n+n0+n1;
vect = zeros(nt,1);
if (n0 ~= 0) & (n0 ~= 1)
    error('** VILERR : Illegal value %  N0 ');
end
if (n1 ~= 0) & (n1 ~= 1)
    error('** VILERR : Illegal value for N1 ');
end
if (nt < 1)
    error('** VILERR : Number of interpolation points less than 1');
end
if (id ~= 1 & id ~= 2 & id ~= 3 )
    error('** VILERR : Illegal ID in DFOPR ')
end
if ( id ~= 3 )
    if ( i < 1 )
        error('** VILERR : Index less than zero in DFOPR ')
    end
    if ( i > nt )
        error('** VILERR : Index greater than NTOTAL in DFOPR ')
    end
end

%
% -- EVALUATE DISCRETIZATION MATRICES AND GAUSSIAN QUADRATURE
% -- WEIGHTS.  QUADRATURE WEIGHTS ARE NORMALIZED TO SUM TO ONE.
%
if ( id ~= 3 )
    for j = 1:nt
        if (j == i)
            if (id == 1)
                vect(i) = dif2(i)/dif1(i)/2;
            else
                vect(i) = dif3(i)/dif1(i)/3;
            end
        else
            y = root(i)-root(j);
            vect(j) = dif1(i)/dif1(j)/y;
            if (id == 2 )
                vect(j)=vect(j)*(dif2(i)/dif1(i)-2/y);
            end
        end
    end
else
    y=0;
    for j = 1:nt
        x = root(j);
        ax = x*(1-x);
        if (n0 == 0)
            ax = ax/x/x;
        end
        if (n1 == 0)
            ax = ax/(1-x)/(1-x);
        end
        vect(j) = ax/dif1(j)^2;
        y = y + vect(j);
    end
    vect = vect/y;
end

%%%% dif %%%%
function [dif1,dif2,dif3] = dif( root )
nt = length( root );
dif1 = zeros(nt,1);
dif2 = zeros(nt,1);
dif3 = zeros(nt,1);
if ( nt < 1 )
    error('**␣VILERR␣:␣Number␣of␣interpolation␣points␣less␣than␣1');
end
for i = 1:nt
    x = root(i);
    dif1(i) = 1;
    dif2(i) = 0;
    dif3(i) = 0;
    for j = 1:nt
        if ( j ~= i)
            y = x - root(j);
            dif3(i) = y*dif3(i) + 3*dif2(i);
            dif2(i) = y*dif2(i) + 2*dif1(i);
            dif1(i) = y*dif1(i);
        end
    end
```

```
end
```

constant.m

```
%This script gives the constants and geometry for the condenser


Asec=0.56; %cross sectional area for flow
D=sqrt(Asec*4/pi);
d0=19.05E-3;          %outer diameter of tube
dh=pi*d0/2;           %diameter for Re and Pr-nmb
nmbt=11288;           %number of tubes
Aeff=675.56;          %heat area pr length [m2/m]

%Some constants
Meth=30.1; % molar weigth [kg/kmol]
Mprop=44.1; % molar weigth [kg/kmol]
R=8.314;     %Universal gas constant [kJ/(kmol K)]


%For sea water
msw=5025.70;     %mass flow [kg/s]
cpsw=4.195E3;      %spesific heat capasity [J/kgK]
```

A.3 Physical properties and parameters

ackermann.m

%This function calculates the Ackermann factor according to Webb et al. [1999]. Heat-capacity is calculated according to Perry and Green [2008].

```
%Input:
%   ne  flux of ethane
%   np  flux of propane
%   hg  vapour heat transfer coefficient
%   T   vapour temperature
%Output:
%   eps ackermann factor

function eps=ackermann(ne,np,hg,T)
```

```
%calculate heatcapacity
cpetv=31742+26.567*T+0.12927*T.^2;
cpprv=26675+147.04*T;

%calculate ackermann factor
eps=(ne.*cpetv+np.*cpprv)./hg;
```

conductivity.m

```
%This function calculates the conductivity of the fluid
%Input:
%   T temperature
%   state indicates if fluid is vapour or condensate
%Output:
%   lamda conductivity

function lamdas=conductivity(T,state)

Tc=T-273;

%state=0: vapour
%state=1: condensate

if state==0
    lamdas=6E-5.*Tc+0.0213;
else
    lamdas=2E-5.*Tc.^2-0.0009.*Tc+0.0954;
end
```

dewpoint.m

```
%This function calculates the dew point of the mixture
%Input:
%   yet initial molefraction of ethane in vapour
%Output:
%   Tdew dewpoint temperature

function Tdew=dewpoint(yet)

%load equilibriumdata for finding dewpoint
load equilibriumdata;
```

```
l1=yet-0.005;
l2=yet+0.005;

j=find(yeti>l1 & yeti<l2);

Tdew=Ti(j);
```

heatcapacity.m

```
%This function calculates the heatcapacity of the stream
%The temperature is in Kelvin and the molfrac is the
%total molefraction


function CP=heatcapacity(T,Nev,Npv)

%Total molfractions
xetv=Nev./(Nev+Npv);
xprv=Npv./(Nev+Npv);

%calculate heatcapacities
cpetv=31742+26.567*T+0.12927*T.^2;
cpprv=26675+147.04*T;

%Calculate total heatcapacity
CP=(xetv.*cpetv+xprv.*cpprv); %J/kmol,K
```

heatcapacitycond.m

%This function calculates the heatcapacity of the condensate based on %Perry and Green [2008]. The critical temperatures are from Matschke and Thodos [1962]

```
%Input:
%   T temperature
%   Nel moles ethane in condensate
%   Npl moles propane in condensate
%Output:
%   Cp heatcapacity %J/kmol,K
```

```
function Cp=heatcapacitycond(T,Nel,Npl)

%Calculate molefractions
xE=Nel./(Nel+Npl);
xP=1-xE;

%Critical temperatures
Tce=305.32 ;
Tcp=369.83;

%Calculate te and tp
te=1-T./Tce;
tp=1-T./Tcp;

%Calculate individual heatcapacities
cpetl=44.009^2./te+89718-2*44.009*918.77.*te+44.009*1886.*te.^2-...
    918.77^2.*te.^3./3+918.77*1886.*te.^4./2-1886^2.*te.^5./5;
cpprl=62.983^2./tp+113630-2*62.983*633.21.*tp+...
    62.983*873.46.*tp.^2-633.21^2.*tp.^3./3+...
    633.21*873.46.*tp.^4./2-873.46^2.*tp.^5./5;

%Calculate total heatcapacity
Cp=xE.*cpetl+xP.*cpprl;
```

heatofcond.m

%This function calculates the heat of condensation for the different
%temperatures based on Perry and Green [2008]. The critical temperatures are from Matschke and Thodos [1962]

```
%Input:
%   T temperature
%   comp indicates which component is used
%Output:
%   hcond heat of condensation


function hcond=heatofcond(T,comp)

%Critical temperatures from
Tce=305.32 ;
```

```
Tcp=369.83;

%Calculate reduced temperatures
TrE=T./Tce;
TrP=T./Tcp;

%Parameters for Ethane
C1E=2.1091E7;
C2E=0.60646;
C3E=-0.55492;
C4E=0.32799;

%Parameters for Propane
C1P=2.92091E7;
C2P=0.78237;
C3P=-0.77319;
C4P=0.39246;

%Calculate heat of condensation
%As it is the heat of vapourisation that is used
%in the equations, the minus sign is also used
%in the condensermodels
if comp==1
hcond=-(C1E.*(1-TrE).^(C2E+C3E.*TrE+C4E.*TrE.^2)); %J/kmol ethane
else
hcond=-(C1P.*(1-TrP).^(C2P+C3P.*TrP+C4P.*TrP.^2)); %J/kmol propane
end
```

masstransfer2.m

```
%This function calculates the mass transfer coefficient

function betaEP=masstransfer2(T,P,hg,Nev,Npv)

%Input:
%   Nev moles of ethane in vapour [kmol/s]
%   Npv moles of propane in vapour [kmol/s]
%   hg  vapour heat transfer coefficient [W/(m2,K)]
%   P   pressure [Pa]
%   T   temperature [K]
%Output:
%   betaEP mass transfer coefficient [kmol/(s,m2)]
```

```
%Calculate heatcapacity of vapour
Cpg=heatcapacity(T,Nev,Npv);

Meth=30.1; % molar weigth [kg/kmol]
Mprop=44.1; % molar weigth [kg/kmol]
R=8.314;     %Universal gas constant [kJ/(kmol K)]

%Calculate conductivity of vapour
 kg1=conductivity(T,0);     %W/mK

%Calculate pressure in bar
P1=P*9.869E-6;

%Calculat diffusivity coefficient according to Fullers method
D=1E-7*T.^(1.75).*(1/Meth+1/Mprop)^(0.5)./...
    (P1.*((44.88)^(1/3)+(65.34)^(1/3)));

%Calculate consentration based on ideal gas law
c1=P./(R.*T*1000);    %[kmol/m3]

 %Calculate mass transfer coefficent
 betaEP=hg./(Cpg).*(c1.*D.*Cpg./kg1).^(2/3); %[kmol/(s,m2)]
```

Tinterface.m

```
%This function calculates the equilibrium composition
%Input:
%   Nel moles ethane in condensate
%   Npl moles propane in condensate
%   P pressure [Pa]
%Output:
%   Tint interface temperature

function [Tint]= Tinterface(Nel,Npl,P)

%Calculate molefraction ethane in condensate
xE=Nel./(Nel+Npl);

%Load correct equilibriumdata based on pressure
if P<2.01e+6
load equilibriumdata_P_20;
elseif P<2.02e+6
    load equilibriumdata_P_201;
elseif P<2.03e+6
```

```
    load equilibriumdata_P_202;
elseif P<2.04e+6
    load equilibriumdata_P_203;
elseif P<2.05e+6
    load equilibriumdata_P_204;
elseif P<2.06e+6
    load equilibriumdata_P_205;
elseif P<2.07e+6
    load equilibriumdata_P_206;
elseif P<2.08e+6
    load equilibriumdata_P_207;
elseif P<2.09e+6
    load equilibriumdata_P_208;
elseif P<2.1e+6
    load equilibriumdata_P_209;
else
    load equilibriumdata_P_21;
end

%Interpolate to find correct interface temperature
l1=xE+0.1;
l2=xE-0.1;

j=find(xeti<l1 & xeti>l2);

[m n]=size(j);


if n==0
    l1=xE+0.2;
    l2=xE-0.2;


    j=find(xeti<l1 & xeti>l2);

    [m n]=size(j);

    if n==0
        l1=xE+0.3;
        l2=xE-0.3;

        j=find(xeti<l1 & xeti>l2);

        [m n]=size(j);
```

```
if n==0
    l1=xE+0.4;
    l2=xE-0.4;

    j=find(xeti<l1 & xeti>l2);

    [m n]=size(j);

    if n==0
        l1=xE+0.5;
        l2=xE-0.5;

        j=find(xeti<l1 & xeti>l2);

        [m n]=size(j);
        if n==1

            Tint=Ti(j);
        else

            xeinter=xeti(j);
            Tinter=Ti(j);

            Tint=(xE-xeinter(2))*(Tinter(2)-Tinter(1))/...
                (xeinter(2)-xeinter(1))+Tinter(1);
        end
    elseif n==1

            Tint=Ti(j);
    else

            xeinter=xeti(j);
            Tinter=Ti(j);

            Tint=(xE-xeinter(2))*(Tinter(2)-Tinter(1))/...
                (xeinter(2)-xeinter(1))+Tinter(1);
    end

elseif n==1

            Tint=Ti(j);
else
```

```
                xeinter=xeti(j);
                Tinter=Ti(j);

                Tint=(xE-xeinter(2))*(Tinter(2)-Tinter(1))/...
                    (xeinter(2)-xeinter(1))+Tinter(1);
        end

    elseif n==1
                Tint=Ti(j);
    else
                xeinter=xeti(j);
                Tinter=Ti(j);

                Tint=(xE-xeinter(2))*(Tinter(2)-Tinter(1))/...
                    (xeinter(2)-xeinter(1))+Tinter(1);
    end

elseif n==1

                Tint=Ti(j);
else
                xeinter=xeti(j);
                Tinter=Ti(j);

                Tint=(xE-xeinter(2))*(Tinter(2)-Tinter(1))/...
                    (xeinter(2)-xeinter(1))+Tinter(1);
end
```

viscosity.m

```
%this function calculates the viscosity of the fluid
%Input:
%   T temperature [K]
%   state indicates if fluid is vapour or condensate

function visc=viscosity(T,state)

%Adjust temperature to \textdegree C
Tc=T-273;

%state=0: vapour
%state=1: condensate

%Calculate viscosity
```

```
if state==0
    visc=(2E-5.*Tc+0.0096)*1E-3;
else
    visc=(3E-5.*Tc-0.0011.*Tc+0.0839)*1E-3;
end
```

Xequilibrium.m

```
%This function calculates the equilibrium composition
%Input:
%   Tint interface temperature
%   P pressure [Pa]
%Output:
%   Xet molefraction condensate


function [Xet]= Xequilibrium(Tint,P)


%load correct equilibriumdata according to pressure

if P<2.01e+6
load equilibriumdata_P_20;
elseif P<2.02e+6
    load equilibriumdata_P_201;
elseif P<2.03e+6
    load equilibriumdata_P_202;
elseif P<2.04e+6
    load equilibriumdata_P_203;
elseif P<2.05e+6
    load equilibriumdata_P_204;
elseif P<2.06e+6
    load equilibriumdata_P_205;
elseif P<2.07e+6
    load equilibriumdata_P_206;
elseif P<2.08e+6
    load equilibriumdata_P_207;
elseif P<2.09e+6
    load equilibriumdata_P_208;
elseif P<2.1e+6
    load equilibriumdata_P_209;
else
    load equilibriumdata_P_21;
end
```

```
%Interpolate to find correct interface composition
l1=Tint+0.1;
l2=Tint-0.1;

j=find(Ti<l1 & Ti>l2);

[m n]=size(j);


if n==0
    l1=Tint+0.2;
    l2=Tint-0.2;


    j=find(Ti<l1 & Ti>l2);

    [m n]=size(j);

    if n==0
        l1=Tint+0.3;
        l2=Tint-0.3;

        j=find(Ti<l1 & Ti>l2);

        [m n]=size(j);

         if n==0
            l1=Tint+0.4;
            l2=Tint-0.4;

            j=find(Ti<l1 & Ti>l2);

            [m n]=size(j);

            if n==0
                l1=Tint+0.5;
                l2=Tint-0.5;

                j=find(Ti<l1 & Ti>l2);

                [m n]=size(j);
                if n==1
                    Xet=xeti(j);
                else
                    xeinter=xeti(j);
                    Tinter=Ti(j);

                    Xet=(Tint-Tinter(2))*(xeinter(2)-xeinter(1))/...
                        (Tinter(2)-Tinter(1))+xeinter(1);
                end

            elseif n==1
                    Xet=xeti(j);
            else
                    xeinter=xeti(j);
                    Tinter=Ti(j);

                    Xet=(Tint-Tinter(2))*(xeinter(2)-xeinter(1))/...
                        (Tinter(2)-Tinter(1))+xeinter(1);
            end

         elseif n==1
                     Xet=xeti(j);
         else
                    xeinter=xeti(j);
                    Tinter=Ti(j);

                    Xet=(Tint-Tinter(2))*(xeinter(2)-xeinter(1))/...
                        (Tinter(2)-Tinter(1))+xeinter(1);
         end

    elseif n==1
                 Xet=xeti(j);
    else
                    xeinter=xeti(j);
                    Tinter=Ti(j);

                    Xet=(Tint-Tinter(2))*(xeinter(2)-xeinter(1))/...
                        (Tinter(2)-Tinter(1))+xeinter(1);
                end
elseif n==1
                    Xet=xeti(j);
else
                    xeinter=xeti(j);
                    Tinter=Ti(j);
                    Xet=(Tint-Tinter(2))*(xeinter(2)-xeinter(1))/...
                        (Tinter(2)-Tinter(1))+xeinter(1);
                end
```

yequilibrium.m

```
%This function calculates the equilibrium composition
%Input:
%   Nel moles ethane in condensate
%   Npl moles propane in condensate
%   P pressure [Pa]
%Output:
%   yet interface molefraction ethane in vapour

function [yet]= yequilibrium(Nel,Npl,P)

%Calculate molefraction ethane of condensate
xE=Nel./(Nel+Npl);

%Load correct equilibriumdata based on pressure
if P<2.01e+6
load equilibriumdata_P_20;
elseif P<2.02e+6
    load equilibriumdata_P_201;
elseif P<2.03e+6
    load equilibriumdata_P_202;
elseif P<2.04e+6
    load equilibriumdata_P_203;
elseif P<2.05e+6
    load equilibriumdata_P_204;
elseif P<2.06e+6
    load equilibriumdata_P_205;
elseif P<2.07e+6
    load equilibriumdata_P_206;
elseif P<2.08e+6
    load equilibriumdata_P_207;
elseif P<2.09e+6
    load equilibriumdata_P_208;
elseif P<2.1e+6
    load equilibriumdata_P_209;
else
    load equilibriumdata_P_21;
end
```

```
%Interpolate to find correct interface molefraction
%of ethane in vapour
l1=xE+0.1;
l2=xE-0.1;

j=find(xeti<l1 & xeti>l2);

[m n]=size(j);


if n==0
    l1=xE+0.2;
    l2=xE-0.2;

    j=find(xeti<l1 & xeti>l2);

    [m n]=size(j);

    if n==0
        l1=xE+0.3;
        l2=xE-0.3;

        j=find(xeti<l1 & xeti>l2);

        [m n]=size(j);

         if n==0
            l1=xE+0.4;
            l2=xE-0.4;

            j=find(xeti<l1 & xeti>l2);

            [m n]=size(j);

            if n==0
                l1=xE+0.5;
                l2=xE-0.5;

                j=find(xeti<l1 & xeti>l2);

                [m n]=size(j);
                if n==1
```

```
              yet=yeti(j);
            else

              yetinter=yeti(j);
              xeinter=xeti(j);

              yet=(xE-xeinter(2))*(yetinter(2)-yetinter(1))/...
                  (xeinter(2)-xeinter(1))+yetinter(1);
            end
        elseif n==1

              yet=yeti(j);
        else

              yetinter=yeti(j);
              xeinter=xeti(j);

              yet=(xE-xeinter(2))*(yetinter(2)-yetinter(1))/...
                  (xeinter(2)-xeinter(1))+yetinter(1);
        end

     elseif n==1

              yet=yeti(j);
     else

              yetinter=yeti(j);
              xeinter=xeti(j);

              yet=(xE-xeinter(2))*(yetinter(2)-yetinter(1))/...
                  (xeinter(2)-xeinter(1))+yetinter(1);
     end

elseif n==1
              yet=yeti(j);
else

              yetinter=yeti(j);
              xeinter=xeti(j);

              yet=(xE-xeinter(2))*(yetinter(2)-yetinter(1))/...
                    (xeinter(2)-xeinter(1))+yetinter(1);
end
```

```
elseif n==1

                    yet=yeti(j);
else

                    yetinter=yeti(j);
                    xeinter=xeti(j);

                    yet=(xE-xeinter(2))*(yetinter(2)-yetinter(1))/...
                        (xeinter(2)-xeinter(1))+yetinter(1);
end
```

A.4 Variations of Model

condenser2.m

```
%This script will model the condenser using collocation

clear all
n=3; %For cooling
m=25; %For integral condensation
[xn, An, Bn, qn] = colloc( n ,1 ,1);
[xm, Am, Bm, qm] = colloc( m ,1 ,1);
Ao=Am;
o=m;
xo=xm;

constant

%The starting values are from table 5.1 in previous master thesis

T0= 31.2 + 273; % K
P0= 20.740*10^5; % Pa
Ta0=6.5 + 273;% K
L0=0; % Liquid molar flow [mol vapour/s]
Vm0=478.31; %vapour mass flow [kg/s]
yeth0=0.64;%;
yprop0=0.36;%;
u0=0.61;      %velocity [m/s]

M0=yeth0*Meth+Mprop*yprop0; %molar mass of vapour feed
V0=Vm0/M0; %Calculate vapour molar flow [kmol/s]
```

```
ntot=V0;
Nev0=yeth0*V0;
Npv0=yprop0*V0;
Nel0=0.41;
Npl0=0.59;

Tdew=dewpoint(yeth0);

 load y_0405_cond6_2

t=(n+2)*3+(m+2)*7;
t1=n+2;
t2=m+2;


%%
% y=ones(t+1,1);
% y(1:t1)=T0.*y(1:t1);
% y(t1+1:2*t1)=P0.*y(t1+1:2*t1);
% y(2*t1+1:3*t1)=Ta0.*y(2*t1+1:3*t1);
%% y(3*t1+1:4*t1)=u0.*y(3*t1+1:4*t1);
%
%
% y(3*t1+1:3*t1+t2)=Tdew.*y(3*t1+1:3*t1+t2);
% y(3*t1+t2+1:3*t1+2*t2)=P0.*y(3*t1+t2+1:3*t1+2*t2);
% y(3*t1+2*t2+1:3*t1+3*t2)=Ta0.*y(3*t1+2*t2+1:3*t1+3*t2);
% y(3*t1+3*t2+1:3*t1+4*t2)=Nev0.*y(3*t1+3*t2+1:3*t1+4*t2);
% y(3*t1+4*t2+1:3*t1+5*t2)=Npv0.*y(3*t1+4*t2+1:3*t1+5*t2);
%%
% y(4*t1+1:4*t1+t2)=Tdew.*y(4*t1+1:4*t1+t2);
% y(4*t1+t2+1:4*t1+2*t2)=P0.*y(4*t1+t2+1:4*t1+2*t2);
% y(4*t1+2*t2+1:4*t1+3*t2)=Ta0.*y(4*t1+2*t2+1:4*t1+3*t2);
% y(4*t1+3*t2+1:4*t1+4*t2)=Nev0.*y(4*t1+3*t2+1:4*t1+4*t2);
% y(4*t1+4*t2+1:4*t1+5*t2)=Npv0.*y(4*t1+4*t2+1:4*t1+5*t2);
% y(4*t1+7*t2+1:4*t1+8*t2)=u0.*y(4*t1+7*t2+1:4*t1+8*t2);
% y(t+1)=500;
% y(t+2)=10000;
% y(t+3)=1.8;




y0=[T0 P0 Ta0 u0];
```

```
opt=optimset('Display','iter','MaxFunEvals',50000,...
    'MaxIter',100000,'TolX',1E-10);
y=fsolve(@(y) condensermodell6(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,y0,...
    Nel0,Npl0,Nev0,Npv0),y,opt);

T(1:t1)=y(1:t1);
T(t1+1:t1+t2)=y(3*t1+1:3*t1+t2);
T=T';
P(1:t1)=y(t1+1:2*t1);
P(t1+1:t1+t2)=y(3*t1+t2+1:3*t1+2*t2);
P=P';
Ta(1:t1)=y(2*t1+1:3*t1);
Ta(t1+1:t1+t2)=y(3*t1+2*t2+1:3*t1+3*t2);
Ta=Ta';
Nev(1:t1)=Nev0.*ones(1,n+2);
Nev(t1+1:t1+t2)=y(3*t1+3*t2+1:3*t1+4*t2);
Nev=Nev';
Npv(1:t1)=Npv0.*ones(1,n+2);
Npv(t1+1:t1+t2)=y(3*t1+4*t2+1:3*t1+5*t2);
Npv=Npv';

Nev0v=Nev0*ones(1,m+2+n+2)';
Nel0v=Nel0*ones(1,m+2+n+2)';
Npv0v=Npv0*ones(1,m+2+n+2)';
Npl0v=Npl0*ones(1,m+2+n+2)';

Nel=Nev0v-Nev;
Npl=Npv0v-Npv;

nt(1:t1)=zeros(1,t1);
nt(t1+1:t1+t2)=y(3*t1+5*t2+1:3*t1+6*t2);
nt=nt';
ne(1:t1)=zeros(1,t1);
ne(t1+1:t1+t2)=y(3*t1+6*t2+1:3*t1+7*t2);
ne=ne';

np=nt-ne;

% A1=300;
% A2=3000;
A1=y(t+1);
A2=y(t+2);
```

```
xt(1:t1)=A1.*xn;
xt(t1+1:t1+t2)=A1+A2.*xm;

save condensation_0206_condmod6_1
```

condensermodell6.m

```
%This function is the same as condensermodell3.m except that
%the equations for differential equations are taken out.
%The constraints for ending the Integral condensation
%is given by defining Nel(m+2)=0.3. For comments, see
%condensermodell3.m

function f=condensermodell6(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,...
    y0,Nel0,Npl0,Nev0,Npv0)


z=n+2;

T1=sqrt(y(1:z).^2);
P1=sqrt(y(z+1:2*z).^2);
Ta1=sqrt(y(2*z+1:3*z).^2);


xs=3*z;
x=m+2;
T2=sqrt(y(xs+1:xs+x).^2);
P2=sqrt(y(xs+x+1:xs+2*x).^2);
Ta2=sqrt(y(xs+2*x+1:xs+3*x).^2);
Nev2=sqrt(y(xs+3*x+1:xs+4*x).^2);
Npv2=sqrt(y(xs+4*x+1:xs+5*x).^2);
nt2=y(xs+5*x+1:xs+6*x);
ne2=y(xs+6*x+1:xs+7*x);


xd=3*z+7*x;


A1=y(xd+1);


A2=y(xd+2);
kp=y(xd+3);
```

```
constant;

hg=300;
Us=500;

U=709.5;


%boundry conditions at x=0;

f(1)=T1(1)-y0(1,1); %
f(z+1)=P1(1)-y0(1,2); %
f(2*z+1)=An(1,:)*Ta1(:)+Us*A1*(T1(1)-Ta1(1))/(msw*cpsw); %Ta


%for n internal collocation points

for j=2:n+1
    f(j)=An(j,:)*T1(:)+Us*A1*(T1(j)-Ta1(j))/...
        ((Nev0+Npv0)*heatcapacity(T1(j),Nev0,Npv0));
    f(z+j)=An(j,:)*P1(:)+kp*A1;
    f(2*z+j)=An(j,:)*Ta1(:)+Us*A1*(T1(j)-Ta1(j))/(msw*cpsw);

end

%for x=1
f(z)=An(n+2,:)*T1(:)+Us*A1*(T1(n+2)-Ta1(n+2))/((Nev0+Npv0)*...
    heatcapacity(T1(n+2),Nev0,Npv0));
f(2*z)=An(n+2,:)*P1(:)+kp*A1;
f(3*z)=Ta1(n+2)-Ta2(1);


%

ack=ones(1,m+2)';
cpg=ones(1,m+2)';
beta=ones(1,m+2)';
teta=ones(1,m+2)';
hg0=ones(1,m+2)';
hvf=ones(1,m+2)';
yE=ones(1,m+2)';
Tint2=299.*ones(1,m+2)';
```

```
Nel2=Nev0*ones(1,m+2)'-Nev2+Nel0;
Npl2=Npv0*ones(1,m+2)'-Npv2+Npl0;

np2=nt2-ne2;




f(xs+1)=T2(1)-T1(n+2);
f(xs+x+1)=P2(1)-P1(n+2);
f(xs+2*x+1)=Am(1,:)*Ta2(:)+U*A2*(T2(1)-Ta2(1))/(msw*cpsw);  %Ta
f(xs+3*x+1)=Nev2(1)-Nev0;
f(xs+4*x+1)=Npv2(1)-Npv0;
f(xs+5*x+1)=nt2(1);
f(xs+6*x+1)=ne2(1);



    for i=2:m+1
        ack(i)=ackermann(ne2(i),np2(i),hg,T2(i));
        cpg(i)=heatcapacity(T2(i),Nev2(i),Npv2(i));
        beta(i)=masstransfer2(T2(i),P2(i),hg,Nev2(i),Npv2(i));
        teta(i)=nt2(i)/beta(i);
        hg0(i)=hg*ack(i)/(exp(ack(i))-1);
        hvf(i)=(-ne2(i)*heatofcond(T2(i),1)-np2(i)*...
            heatofcond(T2(i),0))/nt2(i);
        yE(i)=Nev2(i)/(Nev2(i)+Npv2(i));
        Tint2(i)=Tinterface(Nel2(i),Npl2(i),P2(i));


        f(xs+i)=Am(i,:)*T2(:)+hg0(i)*A2*(T2(i)-Tint2(i))/...
            ((Nev2(i)+Npv2(i))*cpg(i));
        f(xs+x+i)=Am(i,:)*P2(:)+kp*A2;
        f(xs+2*x+i)=Am(i,:)*Ta2(:)+U*A2*(Tint2(i)-Ta2(i))/...
            (msw*cpsw);
        f(xs+3*x+i)=Am(i,:)*Nev2(:)+ne2(i)*A2;
        f(xs+4*x+i)=Am(i,:)*Npv2(:)+np2(i)*A2;
       f(xs+5*x+i)=nt2(i)-(U*(Tint2(i)-Ta2(i))-...
           heatcapacitycond(Tint2(i),Nel2(i),Npl2(i))*...
           (Nel2(i)+Npl2(i))*(Tint2(i-1)-Tint2(i))/A2)/hvf(i);
        f(xs+6*x+i)=ne2(i)-(beta(i)*teta(i)*(yE(i)-...
            yequilibrium(Nel2(i),Npl2(i),P2(i)))/...
            (exp(teta(i))-1)+yE(i)*nt2(i));
    end


    ack(m+2)=ackermann(ne2(m+2),np2(m+2),hg,T2(m+2));
    cpg(m+2)=heatcapacity(T2(m+2),Nev2(m+2),Npv2(m+2));
    beta(m+2)=masstransfer2(T2(m+2),P2(m+2),hg,Nev2(m+2),Npv2(m+2));
    teta(m+2)=nt2(m+2)/beta(m+2);
    hg0(m+2)=hg*ack(m+2)/(exp(ack(m+2))-1);
    hvf(m+2)=(-ne2(m+2)*heatofcond(T2(m+2),1)-np2(m+2)*...
        heatofcond(T2(m+2),0))/nt2(m+2);
    yE(m+2)=Nev2(m+2)/(Nev2(m+2)+Npv2(m+2));
    Tint2(m+2)=Tinterface(Nel2(m+2),Npl2(m+2),P2(m+2));



    f(xs+x)=Am(m+2,:)*T2(:)+hg0(m+2)*A2*(T2(m+2)-Tint2(m+2))/...
        ((Nev2(m+2)+Npv2(m+2))*cpg(m+2));
    f(xs+2*x)=Am(m+2,:)*P2(:)+kp*A2;
    f(xs+3*x)=Ta2(m+2)-y0(1,3);
    f(xs+4*x)=Am(m+2,:)*Nev2(:)+ne2(m+2)*A2;
    f(xs+5*x)=Am(m+2,:)*Npv2(:)+np2(m+2)*A2;
    f(xs+6*x)=nt2(m+2)-(U*(Tint2(m+2)-Ta2(m+2))-...
        heatcapacitycond(Tint2(m+2),Nel2(m+2),Npl2(m+2))*...
        (Nel2(m+2)+Npl2(m+2)-Nel0-Npl0)*...
        (Tint2(m+1)-Tint2(m+2))/A2)/hvf(m+2);
    f(xs+7*x)=ne2(m+2)-(beta(m+2)*teta(m+2)*(yE(m+2)-...
        yequilibrium(Nel2(m+2),Npl2(m+2),P2(m+2)))/...
        (exp(teta(m+2))-1)+yE(m+2)*nt2(m+2));


f(xd+1)=T1(n+2)-Tdew;

f(xd+2)=Npv2(m+2)-0.3;
f(xd+3)=P2(m+2)-2040*10^3;
```

condensermodell7.m

```
%This function is similar to condensermodell3.m, except a change in
%U and Us. For comments, see condensermodell3.m


function f=condensermodell7(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,...
    y0,Nel0,Npl0,Nev0,Npv0)
```

```
z=n+2;

T1=sqrt(y(1:z).^2);
P1=sqrt(y(z+1:2*z).^2);
Ta1=sqrt(y(2*z+1:3*z).^2);


xs=3*z;
x=m+2;
T2=sqrt(y(xs+1:xs+x).^2);
P2=sqrt(y(xs+x+1:xs+2*x).^2);
Ta2=sqrt(y(xs+2*x+1:xs+3*x).^2);
Nev2=sqrt(y(xs+3*x+1:xs+4*x).^2);
Npv2=sqrt(y(xs+4*x+1:xs+5*x).^2);
nt2=y(xs+5*x+1:xs+6*x);
ne2=y(xs+6*x+1:xs+7*x);


xd=3*z+7*x;

xx=o+2;
T3=sqrt(y(xd+1:xd+xx).^2);
P3=sqrt(y(xd+xx+1:xd+2*xx).^2);
Ta3=sqrt(y(xd+2*xx+1:xd+3*xx).^2);
Nev3=sqrt(y(xd+3*xx+1:xd+4*xx).^2);
Npv3=sqrt(y(xd+4*xx+1:xd+5*xx).^2);
nt3=y(xd+5*xx+1:xd+6*xx);
ne3=y(xd+6*xx+1:xd+7*xx);

A1=y(xd+7*xx+1);
A2=y(xd+7*xx+2);
kp=y(xd+7*xx+3);
A3=y(xd+7*xx+4);
constant;

hg=300;

Us=424;

U=566.32;

hg3=300;
```

```
%boundry conditions at x=0;

f(1)=T1(1)-y0(1,1); %
f(z+1)=P1(1)-y0(1,2); %
f(2*z+1)=An(1,:)*Ta1(:)+Us*A1*(T1(1)-Ta1(1))/(msw*cpsw);


%for n internal collocation points

for j=2:n+1
    f(j)=An(j,:)*T1(:)+Us*A1*(T1(j)-Ta1(j))/((Nev0+Npv0)*...
        heatcapacity(T1(j),Nev0,Npv0));
    f(z+j)=An(j,:)*P1(:)+kp*A1;
    f(2*z+j)=An(j,:)*Ta1(:)+Us*A1*(T1(j)-Ta1(j))/(msw*cpsw);

end

%for x=1
f(z)=An(n+2,:)*T1(:)+Us*A1*(T1(n+2)-Ta1(n+2))/((Nev0+Npv0)*...
    heatcapacity(T1(n+2),Nev0,Npv0));
f(2*z)=An(n+2,:)*P1(:)+kp*A1;
f(3*z)=Ta1(n+2)-Ta2(1);





ack=ones(1,m+2)';
cpg=ones(1,m+2)';
beta=ones(1,m+2)';
teta=ones(1,m+2)';
hg0=ones(1,m+2)';
hvf=ones(1,m+2)';
yE=ones(1,m+2)';
Tint2=297.*ones(1,m+2)';


Nel2=Nev0*ones(1,m+2)'-Nev2+Nel0;
Npl2=Npv0*ones(1,m+2)'-Npv2+Npl0;
np2=nt2-ne2;

np3=nt3-ne3;
Nel3=ne3;
Npl3=np3;
```

```
f(xs+1)=T2(1)-T1(n+2);
f(xs+x+1)=P2(1)-P1(n+2);
f(xs+2*x+1)=Am(1,:)*Ta2(:)+U*A2*(T2(1)-Ta2(1))/(msw*cpsw);
f(xs+3*x+1)=Nev2(1)-Nev0;
f(xs+4*x+1)=Npv2(1)-Npv0;
f(xs+5*x+1)=nt2(1);
f(xs+6*x+1)=ne2(1);


    for i=2:m+1
        ack(i)=ackermann(ne2(i),np2(i),hg,T2(i));
        cpg(i)=heatcapacity(T2(i),Nev2(i),Npv2(i));
        beta(i)=masstransfer2(T2(i),P2(i),hg,Nev2(i),Npv2(i));
        teta(i)=nt2(i)/beta(i);
        hg0(i)=hg*ack(i)/(exp(ack(i))-1);
        hvf(i)=(-ne2(i)*heatofcond(T2(i),1)-np2(i)*...
            heatofcond(T2(i),0))/nt2(i);
        yE(i)=Nev2(i)/(Nev2(i)+Npv2(i));
        Tint2(i)=Tinterface(Nel2(i),Npl2(i),P2(i));


        f(xs+i)=Am(i,:)*T2(:)+hg0(i)*A2*(T2(i)-Tint2(i))/...
            ((Nev2(i)+Npv2(i))*cpg(i));
        f(xs+x+i)=Am(i,:)*P2(:)+kp*A2;
        f(xs+2*x+i)=Am(i,:)*Ta2(:)+U*A2*(Tint2(i)-Ta2(i))/...
            (msw*cpsw);
        f(xs+3*x+i)=Am(i,:)*Nev2(:)+ne2(i)*A2;
        f(xs+4*x+i)=Am(i,:)*Npv2(:)+np2(i)*A2;
        f(xs+5*x+i)=nt2(i)-(U*(Tint2(i)-Ta2(i))-...
            heatcapacitycond(Tint2(i),Nel2(i),Npl2(i))*...
            (Nel2(i)+Npl2(i))*(Tint2(i-1)-Tint2(i))/A2)/hvf(i);
        f(xs+6*x+i)=ne2(i)-(beta(i)*teta(i)*(yE(i)-...
            yequilibrium(Nel2(i),Npl2(i),P2(i)))/...
            (exp(teta(i))-1)+yE(i)*nt2(i));

    end


    ack(m+2)=ackermann(ne2(m+2),np2(m+2),hg,T2(m+2));
    cpg(m+2)=heatcapacity(T2(m+2),Nev2(m+2),Npv2(m+2));
    beta(m+2)=masstransfer2(T2(m+2),P2(m+2),hg,Nev2(m+2),Npv2(m+2));
    teta(m+2)=nt2(m+2)/beta(m+2);
    hg0(m+2)=hg*ack(m+2)/(exp(ack(m+2))-1);
    hvf(m+2)=(-ne2(m+2)*heatofcond(T2(m+2),1)-np2(m+2)*...
        heatofcond(T2(m+2),0))/nt2(m+2);
    yE(m+2)=Nev2(m+2)/(Nev2(m+2)+Npv2(m+2));
    Tint2(m+2)=Tinterface(Nel2(m+2),Npl2(m+2),P2(m+2));



    f(xs+x)=Am(m+2,:)*T2(:)+hg0(m+2)*A2*(T2(m+2)-Tint2(m+2))/...
        ((Nev2(m+2)+Npv2(m+2))*cpg(m+2));
    f(xs+2*x)=Am(m+2,:)*P2(:)+kp*A2;
    f(xs+3*x)=Ta2(m+2)-Ta3(1);
    f(xs+4*x)=Am(m+2,:)*Nev2(:)+ne2(m+2)*A2;
    f(xs+5*x)=Am(m+2,:)*Npv2(:)+np2(m+2)*A2;
    f(xs+6*x)=nt2(m+2)-(U*(Tint2(m+2)-Ta2(m+2))-...
        heatcapacitycond(Tint2(m+2),Nel2(m+2),Npl2(m+2))*...
        (Nel2(m+2)+Npl2(m+2)-Nel0-Npl0)*(Tint2(m+1)-...
        Tint2(m+2))/A2)/hvf(m+2);
    f(xs+7*x)=ne2(m+2)-(beta(m+2)*teta(m+2)*(yE(m+2)-...
        yequilibrium(Nel2(m+2),Npl2(m+2),P2(m+2)))/...
        (exp(teta(m+2))-1)+yE(m+2)*nt2(m+2));

ack3=ones(1,o+2)';
cpg3=ones(1,o+2)';
beta3=ones(1,o+2)';
teta3=ones(1,o+2)';
hg03=ones(1,o+2)';
hvf3=ones(1,o+2)';
yE3=ones(1,o+2)';
Tint3=296.*ones(1,o+2)';


f(xd+1)=T3(1)-T2(m+2);
f(xd+xx+1)=P3(1)-P2(m+2);
f(xd+2*xx+1)=Ao(1,:)*Ta3(:)+U*A3*(T3(1)-Ta3(1))/(msw*cpsw);  %Ta
f(xd+3*xx+1)=Nev3(1)-Nev2(m+2);
f(xd+4*xx+1)=Npv3(1)-Npv2(m+2);
f(xd+5*xx+1)=nt3(1)-nt2(m+2);
f(xd+6*xx+1)=ne3(1)-ne2(m+2);



    for i=2:o+1
        ack3(i)=ackermann(ne3(i),np3(i),hg3,T3(i));
        cpg3(i)=heatcapacity(T3(i),Nev3(i),Npv3(i));
        beta3(i)=masstransfer2(T3(i),P3(i),hg3,Nev3(i),Npv3(i));
        teta3(i)=nt3(i)/beta3(i);
        hg03(i)=hg3*ack3(i)/(exp(ack3(i))-1);
        hvf3(i)=(-ne3(i)*heatofcond(T3(i),1)-...
            np3(i)*heatofcond(T3(i),0))/nt3(i);
        yE3(i)=Nev3(i)/(Nev3(i)+Npv3(i));
        xE3(i)=Xequilibrium(Tint3(i),P3(i));
        Tint3(i)=Tinterface(Nel3(i),Npl3(i),P3(i));

        f(xd+i)=Ao(i,:)*T3(:)+hg03(i)*A3*(T3(i)-Tint3(i))/...
            ((Nev3(i)+Npv3(i))*cpg3(i));
        f(xd+xx+i)=Ao(i,:)*P3(:)+kp*A3;
        f(xd+2*xx+i)=Ao(i,:)*Ta3(:)+U*A3(1)*(Tint3(i)-Ta3(i))/...
            (msw*cpsw);
        f(xd+3*xx+i)=Ao(i,:)*Nev3(:)+ne3(i)*A3(1);
        f(xd+4*xx+i)=Ao(i,:)*Npv3(:)+np3(i)*A3(1);
        f(xd+5*xx+i)=nt3(i)-(U*(Tint3(i)-Ta3(i))-...
            heatcapacitycond(Tint3(i),Nel3(i),Npl3(i))*...
            (Nel3(i)+Npl3(i))*(Tint3(i-1)-Tint3(i))/A3)/hvf3(i);
        f(xd+6*xx+i)=ne3(i)-(beta3(i)*teta3(i)*(yE3(i)-...
            yequilibrium(Nel3(i),Npl3(i),P3(i)))/(exp(teta3(i))-1)+...
            yE3(i)*nt3(i));



    end


    ack3(o+2)=ackermann(ne3(o+2),np3(o+2),hg,T3(o+2));
    cpg3(o+2)=heatcapacity(T3(o+2),Nev3(o+2),Npv3(o+2));
    beta3(o+2)=masstransfer2(T3(o+2),P3(o+2),hg,Nev3(o+2),Npv3(o+2));
    teta3(o+2)=nt3(o+2)/beta3(o+2);
    hg03(o+2)=hg*ack3(o+2)/(exp(ack3(o+2))-1);
    hvf3(o+2)=(-ne3(o+2)*heatofcond(T3(o+2),1)-...
        np3(o+2)*heatofcond(T3(o+2),0))/nt3(o+2);
    yE3(o+2)=Nev3(o+2)/(Nev3(o+2)+Npv3(o+2));
    xE3(o+2)=Xequilibrium(Tint3(o+2),P3(o+2));
    Tint3(o+2)=Tinterface(Nel3(o+2),Npl3(o+2),P3(o+2));

    f(xd+xx)=Ao(o+2,:)*T3(:)+hg03(o+2)*A3*(T3(o+2)-...
        Tint3(o+2))/((Nev3(o+2)+Npv3(o+2))*cpg3(o+2));
    f(xd+2*xx)=Ao(o+2,:)*P3(:)+kp*A3;
```

```
    f(3*xx+xd)=Ta3(o+2)-y0(1,3);
    f(4*xx+xd)=Ao(o+2,:)*Nev3(:)+ne3(o+2)*A3;
    f(5*xx+xd)=Ao(o+2,:)*Npv3(:)+np3(o+2)*A3;
    f(xd+6*xx)=nt3(o+2)-(U*(Tint3(o+2)-Ta3(o+2))-...
        heatcapacitycond(Tint3(o+2),Nel3(o+2),Npl3(o+2))*...
        (Nel3(o+2)+Npl3(o+2))*(Tint3(o+1)-Tint3(o+2))/A3)/hvf3(o+2);
    f(xd+7*xx)=ne3(o+2)-(beta3(o+2)*teta3(o+2)*(yE3(o+2)-...
        yequilibrium(Nel3(o+2),Npl3(o+2),P3(o+2)))/...
        (exp(teta3(o+2))-1)+yE3(o+2)*nt3(o+2));

f(7*xx+xd+1)=T1(n+2)-Tdew;
Mv=Nev2(m+2)*Meth+Npv2(m+2)*Mprop;
Ml=Nel2(m+2)*Meth+Npl2(m+2)*Mprop;
vappros=Mv./(Mv+Ml);
f(7*xx+xd+2)=vappros-0.1237;
f(7*xx+xd+3)=P2(m+2)-2040*10^3;


f(7*xx+xd+4)=T3(o+2)-(283);
```

condenser4case.m

```
function f=condensermodellcase3(y,An,n,xn,Am,m,xm,Ao,o,xo,Tdew,...
    y0,Nel0,Npl0,Nev0,Npv0)

%Define variable z
z=n+2;
%Define variables for cooling
T1=sqrt(y(1:z).^2);
P1=sqrt(y(z+1:2*z).^2);
Ta1=sqrt(y(2*z+1:3*z).^2);


xs=3*z;
x=m+2;
%Define variables for integral condensation
T2=sqrt(y(xs+1:xs+x).^2);
P2=sqrt(y(xs+x+1:xs+2*x).^2);
Ta2=sqrt(y(xs+2*x+1:xs+3*x).^2);
Nev2=sqrt(y(xs+3*x+1:xs+4*x).^2);
Npv2=sqrt(y(xs+4*x+1:xs+5*x).^2);
nt2=y(xs+5*x+1:xs+6*x);
ne2=y(xs+6*x+1:xs+7*x);
```

```
xd=3*z+7*x;
%Define variables for differential condensation
xx=o+2;
T3=sqrt(y(xd+1:xd+xx).^2);
P3=sqrt(y(xd+xx+1:xd+2*xx).^2);
Ta3=sqrt(y(xd+2*xx+1:xd+3*xx).^2);
Nev3=sqrt(y(xd+3*xx+1:xd+4*xx).^2);
Npv3=sqrt(y(xd+4*xx+1:xd+5*xx).^2);
nt3=y(xd+5*xx+1:xd+6*xx);
ne3=y(xd+6*xx+1:xd+7*xx);

%Define area and pressure
A1=y(xd+7*xx+1);
A2=5000;
kp=y(xd+7*xx+2);
A3=5000;

%Load constants
constant;

%Set heat transfer coefficients
hg=8000;
Us=700;      %Us for cooling section
U=1000;      %U for integral section
U3=1000;     %U for differential section
hg3=700;

%Cooling section
%boundry conditions at x=0;

f(1)=T1(1)-y0(1,1); %
f(z+1)=P1(1)-y0(1,2); %
f(2*z+1)=An(1,:)*Ta1(:)+Us*A1*(T1(1)-Ta1(1))/(msw*cpsw); %Ta

%for n internal collocation points
for j=2:n+1
    f(j)=An(j,:)*T1(:)+Us*A1*(T1(j)-Ta1(j))/...
        ((Nev0+Npv0)*heatcapacity(T1(j),Nev0,Npv0));
    f(z+j)=An(j,:)*P1(:)+kp*A1;% %
    f(2*z+j)=An(j,:)*Ta1(:)+Us*A1*(T1(j)-Ta1(j))/(msw*cpsw);

end
```

```
%for x=1
f(z)=An(n+2,:)*T1(:)+Us*A1*(T1(n+2)-Ta1(n+2))/...
    ((Nev0+Npv0)*heatcapacity(T1(n+2),Nev0,Npv0));
f(2*z)=An(n+2,:)*P1(:)+kp*A1;%
f(3*z)=Ta1(n+2)-Ta2(1);

%Integral condensation section

%Pre-define parameters

ack=ones(1,m+2)';
cpg=ones(1,m+2)';
beta=ones(1,m+2)';
teta=ones(1,m+2)';
hg0=ones(1,m+2)';
hvf=ones(1,m+2)';
yE=ones(1,m+2)';
Tint2=297.*ones(1,m+2)';

%Define moles in condensate and flux of propane
%for integral section
Nel2=Nev0*ones(1,m+2)'-Nev2+Nel0;
Npl2=Npv0*ones(1,m+2)'-Npv2+Npl0;
np2=nt2-ne2;
%Define moles in condensate and flux of propane
%for differential section
np3=nt3-ne3;
Nel3=ne3;
Npl3=np3;


%Boundry conditions for xm=1
f(xs+1)=T2(1)-T1(n+2);
f(xs+x+1)=P2(1)-P1(n+2);
f(xs+2*x+1)=Am(1,:)*Ta2(:)+U*A2*(T2(1)-Ta2(1))/(msw*cpsw); %Ta
f(xs+3*x+1)=Nev2(1)-Nev0;
f(xs+4*x+1)=Npv2(1)-Npv0;
f(xs+5*x+1)=nt2(1);
f(xs+6*x+1)=ne2(1);


%For m internal collocation points
    for i=2:m+1
        %Calculate parameters
        ack(i)=ackermann(ne2(i),np2(i),hg,T2(i));
        cpg(i)=heatcapacity(T2(i),Nev2(i),Npv2(i));
        beta(i)=masstransfer2(T2(i),P2(i),hg,Nev2(i),Npv2(i));
        teta(i)=nt2(i)/beta(i);
        hg0(i)=hg*ack(i)/(exp(ack(i))-1);
        hvf(i)=(-ne2(i)*heatofcond(T2(i),1)-...
            np2(i)*heatofcond(T2(i),0))/nt2(i);
        yE(i)=Nev2(i)/(Nev2(i)+Npv2(i));
        Tint2(i)=Tinterface(Nel2(i),Npl2(i),P2(i));

        %Calculate residuals
        f(xs+i)=Am(i,:)*T2(:)+hg0(i)*A2*(T2(i)-Tint2(i))/...
            ((Nev2(i)+Npv2(i))*cpg(i));
        f(xs+x+i)=Am(i,:)*P2(:)+kp*A2;
        f(xs+2*x+i)=Am(i,:)*Ta2(:)+U*A2*(Tint2(i)-Ta2(i))/...
            (msw*cpsw);
        f(xs+3*x+i)=Am(i,:)*Nev2(:)+ne2(i)*A2;
        f(xs+4*x+i)=Am(i,:)*Npv2(:)+np2(i)*A2;
        f(xs+5*x+i)=nt2(i)-(U*(Tint2(i)-Ta2(i))-...
            heatcapacitycond(Tint2(i),Nel2(i),Npl2(i))*...
            (Nel2(i)+Npl2(i))*(Tint2(i-1)-Tint2(i))/A2)/hvf(i);
        f(xs+6*x+i)=ne2(i)-(beta(i)*teta(i)*(yE(i)-...
            yequilibrium(Nel2(i),Npl2(i),P2(i)))/(exp(teta(i))-1)+...
            yE(i)*nt2(i));

    end

%Boundry conditions at xm=1

    %Calculate parameters
    ack(m+2)=ackermann(ne2(m+2),np2(m+2),hg,T2(m+2));
    cpg(m+2)=heatcapacity(T2(m+2),Nev2(m+2),Npv2(m+2));
    beta(m+2)=masstransfer2(T2(m+2),P2(m+2),hg,Nev2(m+2),Npv2(m+2));
    teta(m+2)=nt2(m+2)/beta(m+2);
    hg0(m+2)=hg*ack(m+2)/(exp(ack(m+2))-1);
    hvf(m+2)=(-ne2(m+2)*heatofcond(T2(m+2),1)-...
        np2(m+2)*heatofcond(T2(m+2),0))/nt2(m+2);
    yE(m+2)=Nev2(m+2)/(Nev2(m+2)+Npv2(m+2));
    Tint2(m+2)=Tinterface(Nel2(m+2),Npl2(m+2),P2(m+2));


    %Calculate residuals
    f(xs+x)=Am(m+2,:)*T2(:)+hg0(m+2)*A2*(T2(m+2)-Tint2(m+2))/...
        ((Nev2(m+2)+Npv2(m+2))*cpg(m+2));
    f(xs+2*x)=Am(m+2,:)*P2(:)+kp*A2;
    f(xs+3*x)=Ta2(m+2)-Ta3(1);
    f(xs+4*x)=Am(m+2,:)*Nev2(:)+ne2(m+2)*A2;
    f(xs+5*x)=Am(m+2,:)*Npv2(:)+np2(m+2)*A2;
    f(xs+6*x)=nt2(m+2)-(U*(Tint2(m+2)-Ta2(m+2))-...
        heatcapacitycond(Tint2(m+2),Nel2(m+2),Npl2(m+2))*...
        (Nel2(m+2)+Npl2(m+2))*(Tint2(m+1)-Tint2(m+2))/A2)/hvf(m+2);
    f(xs+7*x)=ne2(m+2)-(beta(m+2)*teta(m+2)*(yE(m+2)-...
        yequilibrium(Nel2(m+2),Npl2(m+2),P2(m+2)))/...
        (exp(teta(m+2))-1)+yE(m+2)*nt2(m+2));

%Differential condensation
%Boundry conditions at xo=1

%Pre-define parameters
ack3=ones(1,o+2)';
cpg3=ones(1,o+2)';
beta3=ones(1,o+2)';
teta3=ones(1,o+2)';
hg03=ones(1,o+2)';
hvf3=ones(1,o+2)';
yE3=ones(1,o+2)';
Tint3=296.*ones(1,o+2)';

%Calculate residuals
f(xd+1)=T3(1)-T2(m+2);
f(xd+xx+1)=P3(1)-P2(m+2);
f(xd+2*xx+1)=Ao(1,:)*Ta3(:)+U3*A3*(T3(1)-Ta3(1))/(msw*cpsw); %Ta
f(xd+3*xx+1)=Nev3(1)-Nev2(m+2);
f(xd+4*xx+1)=Npv3(1)-Npv2(m+2);
f(xd+5*xx+1)=nt3(1)-nt2(m+2);
f(xd+6*xx+1)=ne3(1)-ne2(m+2);


%for o internal collocation points
    for i=2:o+1
        %Calculate parameters
        ack3(i)=ackermann(ne3(i),np3(i),hg3,T3(i));
        cpg3(i)=heatcapacity(T3(i),Nev3(i),Npv3(i));
        beta3(i)=masstransfer2(T3(i),P3(i),hg3,Nev3(i),Npv3(i));
        teta3(i)=nt3(i)/beta3(i);
        hg03(i)=hg3*ack3(i)/(exp(ack3(i))-1);
        hvf3(i)=(-ne3(i)*heatofcond(T3(i),1)-...
            np3(i)*heatofcond(T3(i),0))/nt3(i);
        yE3(i)=Nev3(i)/(Nev3(i)+Npv3(i));
        xE3(i)=Xequilibrium(Tint3(i),P3(i));
        Tint3(i)=Tinterface(Nel3(i),Npl3(i),P3(i));

        %Calculate residuals
        f(xd+i)=Ao(i,:)*T3(:)+hg03(i)*A3*(T3(i)-Tint3(i))/...
            ((Nev3(i)+Npv3(i))*cpg3(i));
        f(xd+xx+i)=Ao(i,:)*P3(:)+kp*A3;
        f(xd+2*xx+i)=Ao(i,:)*Ta3(:)+U3*A3(1)*(Tint3(i)-Ta3(i))/...
            (msw*cpsw);
        f(xd+3*xx+i)=Ao(i,:)*Nev3(:)+ne3(i)*A3(1);
        f(xd+4*xx+i)=Ao(i,:)*Npv3(:)+np3(i)*A3(1);
        f(xd+5*xx+i)=nt3(i)-(U3*(Tint3(i)-Ta3(i)) -...
            heatcapacitycond(Tint3(i),Nel3(i),Npl3(i))*...
            (Nel3(i)+Npl3(i))*(Tint3(i-1)-Tint3(i))/A3)/hvf3(i);
        f(xd+6*xx+i)=ne3(i)-(beta3(i)*teta3(i)*(yE3(i) -...
            yequilibrium(Nel3(i),Npl3(i),P3(i)))/...
            (exp(teta3(i))-1)+yE3(i)*nt3(i));

    end

%Boundry conditions at xo=1
    %Calculate parameters
    ack3(o+2)=ackermann(ne3(o+2),np3(o+2),hg,T3(o+2));
    cpg3(o+2)=heatcapacity(T3(o+2),Nev3(o+2),Npv3(o+2));
    beta3(o+2)=masstransfer2(T3(o+2),P3(o+2),hg,Nev3(o+2),Npv3(o+2));
    teta3(o+2)=nt3(o+2)/beta3(o+2);
    hg03(o+2)=hg*ack3(o+2)/(exp(ack3(o+2))-1);
    hvf3(o+2)=(-ne3(o+2)*heatofcond(T3(o+2),1) -...
        np3(o+2)*heatofcond(T3(o+2),0))/nt3(o+2);
    yE3(o+2)=Nev3(o+2)/(Nev3(o+2)+Npv3(o+2));
    xE3(o+2)=Xequilibrium(Tint3(o+2),P3(o+2));
    Tint3(o+2)=Tinterface(Nel3(o+2),Npl3(o+2),P3(o+2));

    %Calculate residuals
    f(xd+xx)=Ao(o+2,:)*T3(:)+hg03(o+2)*A3*(T3(o+2)-Tint3(o+2))/...
        ((Nev3(o+2)+Npv3(o+2))*cpg3(o+2));
    f(xd+2*xx)=Ao(o+2,:)*P3(:)+kp*A3;
    f(3*xx+xd)=Ta3(o+2)-y0(1,3);
    f(4*xx+xd)=Ao(o+2,:)*Nev3(:)+ne3(o+2)*A3;
    f(5*xx+xd)=Ao(o+2,:)*Npv3(:)+np3(o+2)*A3;
    f(xd+6*xx)=nt3(o+2)-(U3*(Tint3(o+2)-Ta3(o+2)) -...
        heatcapacitycond(Tint3(o+2),Nel3(o+2),Npl3(o+2))*...
        (Nel3(o+2)+Npl3(o+2))*(Tint3(o+1)-Tint3(o+2))/A3)/hvf3(o+2);
    f(xd+7*xx)=ne3(o+2)-(beta3(o+2)*teta3(o+2)*(yE3(o+2)-...
        yequilibrium(Nel3(o+2),Npl3(o+2),P3(o+2)))/...
        (exp(teta3(o+2))-1)+yE3(o+2)*nt3(o+2));

  %Calculate constraints
f(7*xx+xd+1)=T1(n+2)-Tdew;
f(7*xx+xd+2)=P3(o+2)-(2010*10^3);
```

Appendix B

Correction factors for heat transfer coefficient

This appendix presents the correction factors used for the heat transfer coefficient for a shell and tube heat exchanger with helical baffles. The correlations are presented as by Zhang et al. [2010].

Y$_2$ is calculated by comparing the dynamic viscosity at the bulk with the viscosity at the wall as shown in Equation B.1.

$$Y_2 = \left(\frac{\eta_S}{\eta_{S,W}} \right)^{0.14} \tag{B.1}$$

The correlations for Y$_3$ depend on the arrangement of the tubes. For in-line arrangement Y$_3$ is calculated by Equation B.2, whereas Equation B.3 is used for staggered arrangement.

$$Y_3 = 1 + \frac{0.7 \cdot \left(\frac{b}{a} - 0.3 \right)}{\varepsilon^{1.5} \cdot \left(\frac{b}{a} + 0.7 \right)^2} \tag{B.2}$$

$$Y_3 = 1 + \frac{2}{3b} \tag{B.3}$$

a and b are defined as in Figure B.1. ε is depends on b. If b$\geq$1, Equation B.4 is used, otherwise Equation B.5 is used.

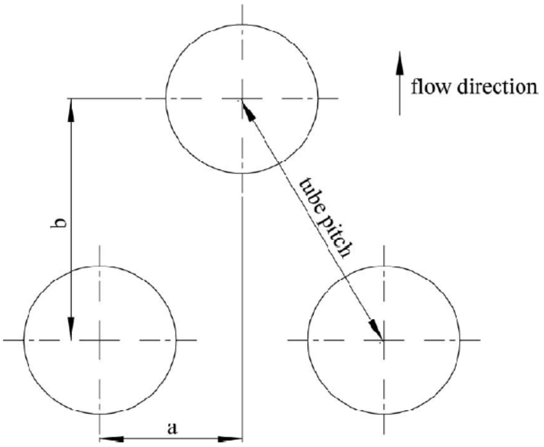


Figure B.1: Parameters used for Y3[Zhang et al., 2010]

$$\varepsilon = 1 - \frac{\pi}{4a} \tag{B.4}$$

$$\varepsilon = 1 - \frac{\pi}{4ab} \tag{B.5}$$

Y_7 is more complicated to calculate and the correlation is given in Equation B.6. In the following formulas, x is substituted for $t_t n_{pt}/D_t$ and y for S_{ss}/S_{2z}. t_t is the tube pitch, n_{pt} is the number of stealing strip pairs and D_t is the inside diameter of the shell. S_{ss} is the bundle-to-tube cross-flow bypass area per baffle and is calculated according to Equation B.7. S_{2z} is the cross-flow area at the centreline and is calculated according to Equation B.8.

$$Y_7 = \exp\left(-1.343x(1-(2y)^{0.338}\right) \tag{B.6}$$

$$S_{ss} = 0.5\left(B - \frac{S_p}{\cos\beta}\right)(D_I - D_s - S_{tt}) \tag{B.7}$$

$$S_{2z} = 0.5\left(B - \frac{S_p}{\cos\beta}\right)\left(D_I - D_s + \frac{D_I - d_o}{t_t}(t_t - d_o)\right) \tag{B.8}$$

S_p is the baffle thickness, S_{tt} is the distance between two tubes from outer diameter to outer diameter. D_s is the diameter of the tube bundle and B is the baffle pitch for helical baffles.

Y_8 is dependent on the ratio (l_{tc}-l_{to})/l_{tc} and the ratio B/D_I and calculated according to Equation B.9 with x and y instead of the ratios. l_{tc} is the effective length of tube bundle whereas l_{to} is the baffled length of tube bundle.

$$Y_8 = 1.079y^{0.0487} - 0.445y^{-0.301}x^{1.2} \tag{B.9}$$

Y_9 and Y_10 are only influenced by the helical angle. For β¡18°, Equation B.10 should be used for Y_9. For helical angles between 18 and 45 °, Equation B.11 should be used. If the helical angle is smaller than 25°Y_{10} is calculated according to Equation B.12. For angles between 25 and 45 °, Equation B.13 is used.

$$Y_9 = 1 \tag{B.10}$$

$$Y_9 = 0.977 + 0.00455\beta - 0.0001821\beta^2 \tag{B.11}$$

$$Y_{10} = 1 \tag{B.12}$$

$$Y_{10} = -56.39 + 8.28\beta - 0.46\beta^2 + 0.012\beta^3 - 1.64 \cdot 10^{-4}\beta^4 + 8.19 \cdot 10^{-7}\beta^5 \tag{B.13}$$

Appendix C

Experimental Composition data for 25-HA-112

The experimental composition data for 25-HA-112 is given in this appendix. Two series are used and the results are presented in Tables C.1 and C.2.

	A06	A10	A13	A15	A16	A17
Species	cold end					warm end
O2	0,04	0,02	0,01	0,03	0,03	0,02
N2	0,02	0	0	0	0	0
C1	0,07	0,06	0,05	0,04	0,04	0,02
C2	87,12	85,81	83,89	81,26	78,92	75,49
C3	12,74	14,08	16,02	18,65	20,98	24,42
I-C4	0,01	0,01	0,01	0,01	0,02	0,02
N-C4	0,01	0,01	0,01	0,01	0,02	0,02

Table C.1: Experimental Molefractions for condenser, taken 26.08.2008

Species	A06	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17
	cold end											warm end
O2	0,03	0,03	0,02	0,03	0,03	0,02	0,03	0,03	0,03	0,01	0,05	0,03
N2	0	0	0	0	0	0	0	0	0	0	0	0
C1	0,16	0,15	0,15	0,15	0,15	0,15	0,15	0,14	0,13	0,12	0,1	0,07
C2	89,19	89,24	89,18	89,16	88,8	88,77	88,59	87,91	87,13	86,07	83,04	78,92
C3	10,63	10,58	10,65	10,67	11,03	11,06	11,24	11,93	12,71	13,78	16,79	20,96

Table C.2: Experimental Molefractions for condenser, taken 19.07.2008

The effective area was calculated according to Equation C.1

$$A_{eff} = A \cdot nmbt = \pi \cdot d_0 \cdot l \cdot nmbt = 675.55 \tag{C.1}$$

where the number of tubes ($nmbt$) is 11 288, the tube outer diameter (d_o) is $19.05 \cdot 10^{-3}$ and the length, l is 1 m.

Appendix D

Areas calculated in Sensitivity Analysis

This Appendix gives the different areas obtained in the different sensitivity tests performed in Section 3.7.

A1 [m^2]	A2 [m^2]	A3 [m^2]	h [$W/m^2.K$]
674.63	17781.14	1081.02	300
674.82	17755.68	1040.36	350
674.96	17738.88	1007.41	400
675.31	17701.77	936.06	600
675.56	17675.16	891.73	900

Table D.1: Area required versus vapour side heat transfer coefficient

A1 [m^2]	A2 [m^2]	A3 [m^2]	Vm0 [kg/s]
674.63	17781.14	1081.02	478.31
717.52	18794.40	1131.79	500
936.59	23832.32	1361.66	600

Table D.2: Area required versus mass flow

A1 [m^2]	A2 [m^2]	A3 [m^2]	Ta0 [K]
653.24	16825.51	997.08	279
674.63	17781.14	1081.02	279.5
679.07	17981.48	1101.35	279.6
683.58	18191.01	1121.49	279.7

Table D.3: Area required versus sea water inlet temperature

A1 [m^2]	A2 [m^2]	A3 [m^2]	R [m^2K/W]
674.63	17781.14	1081.02	0
735.06	20007.70	1182.54	0.000178
795.87	22233.19	1280.70	0.000356
856.15	24458.60	1378.75	0.000534

Table D.4: Area required versus fouling resistance

Appendix E

Temperatures from Matlab and Unisim

This Appendix presents the temperatures used to compare the results obtained in Unisim and MATLAB presented in Section 4.2. The temperatures are presented in Table E.1.

T-Matlab	Ta-Matlab	T-Unisim	Ta-Unisim
304.2	285.387912	304.2	286.071555
297.648678	285.140662	298.82903	285.820311
285.866403	279.689663	285.03	279.60838
285.749981	279.645945	284.5	279.578923
284.638777	279.573166	284	279.551952
283.426628	279.517878	283.5	279.525654
283	279.5	283	279.5

Table E.1: Comparison of Temperature profiles - Unisim and Matlab

Appendix F

Conductivity and Viscosity for Model

The correlations used for calculating the viscosity and conductivity for the model where obtained from the model defined in Unisim. Linear regression in Excel was used to obtain the equations given in Equation F.1 and Equation F.2. Plots are given in Figures F.1 and F.2.

$$\lambda = 6 \cdot 10^{-5} \cdot (T - 273) + 0.0213 \tag{F.1}$$

$$\mu = (2 \cdot 10^{-5} \cdot (T - 273) + 0.0096) \cdot 10^{-3} \tag{F.2}$$

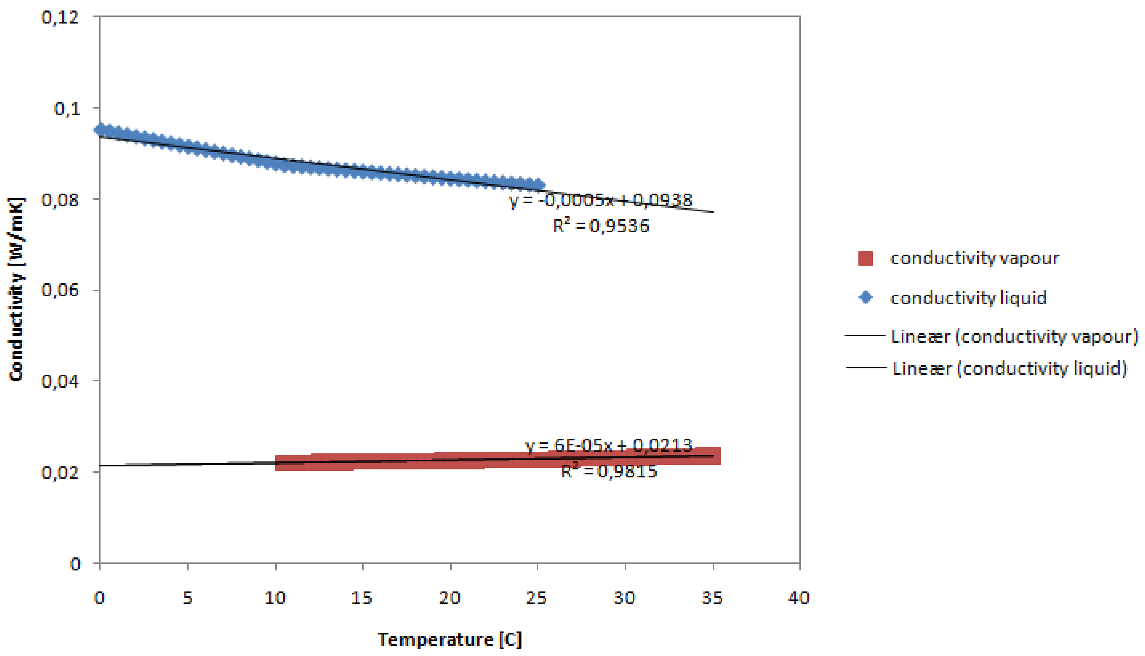


Figure F.1: Conductivity and linear regression

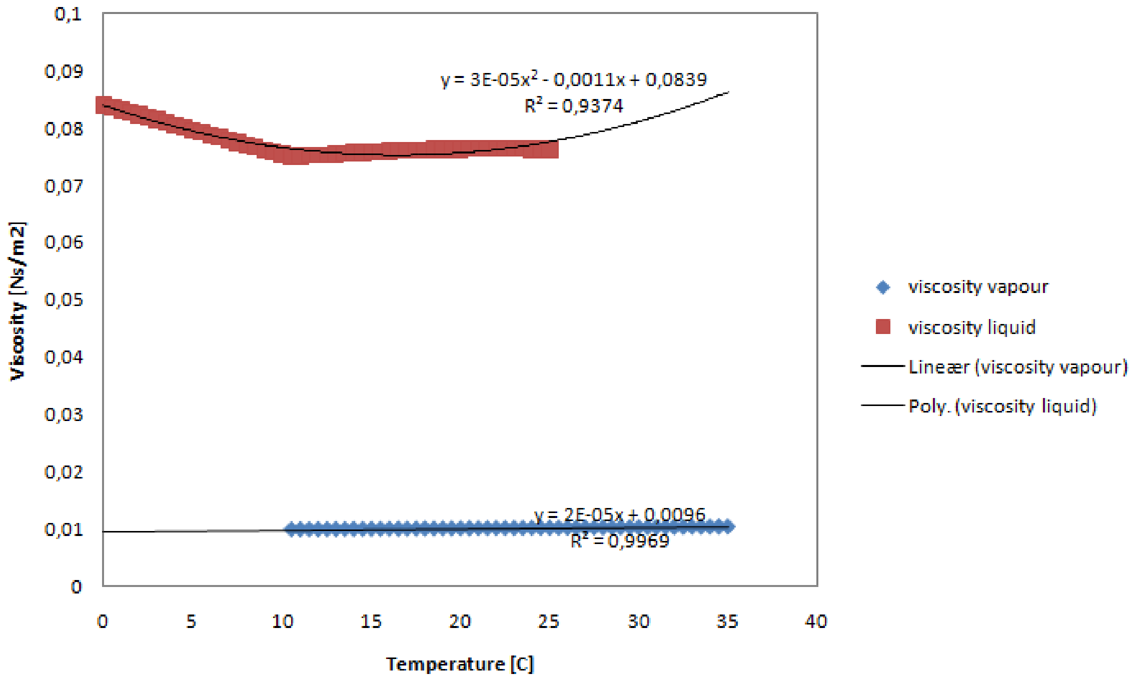


Figure F.2: Viscosity and linear regression

Appendix G

Using the bvp4c routine

To solve the equations a bvp4c routine in MATLAB was used. bvp4c uses a collocation method, described in Section 2.7.

Employing an underlying iteration routine

The first method investigated takes the differential equations in together with a suggested solution at each point. The guessed solution was taken to be the starting values for each state. The script ”condenser.m” was used as the main file, whereas the file ”cooling.m” contained the differential equations that were to be solved. The boundary conditions were given by ”condenserbc.m”.

To get the correct values for the differential equations, the flux equations had to be satisfied. This was done in an underlying iteration routine where the values for $\dot{n}_i$ and $\dot{n}_t$ where calculated using the new values until the difference between $\dot{q}_1$ and $\dot{q}_2$ was smaller than a pre-defined value. This value was set to 10^{-20}, to avoid noise in the overlying iteration of the bvp4c-routine.

The iteration scheme turned out to be unstable for temperatures close to the dew point where the amount of moles in the liquid was small. With an initial value for these in the order of 10^{-6}, the condensation rates near the dew point where unstable, giving negative moles in the liquid and unsound mole fractions. These unphysical properties exceeded the boundaries of the thermodynamic model, resulting in no solution. When the initial amount of moles was raised in the order of 10^{-1}, the instabilities were not as significant, and solutions were obtained. A better solution that could handle the instabilities around the dew point where the first condensation occurs was thus crucial.

Later several errors were found in this program that may have led to the unsatisfactory answers and great instabilities. It was chosen not to pursue this method further, due to low insight and influence in the procedure.

Appendix H

Risk Assessment

NTNU	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	05.03.2010
HMS		Godkjent av	Side	Erstatter
		Rektor	1 av 2	01.12.2006

Enhet: IKP

Dato: 05.06.2011

Deltakere ved kartleggingen (m/ funksjon):
Rebecca Williams (Utførende)

Kort beskrivelse av hovedaktivitet/hovedprosess:
Datasimulering kun på lesesal

ID nr.	Aktivitet/prosess	Ansvarlig	Lov, forskrift o.l.	Eksisterende dokumentasjon	Eksisterende sikringstiltak	Kommentar
1	Datasimulering	-	-	NTNU HMS håndbok	-	Ingen risiko

NTNU	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	05.03.2010
		Godkjent av	Side	Erstatter
HMS		Rektor	2 av 2	01.12.2006

Hva

Aktivitet/prosess
Kartlegging av aktiviteter/prosesser som inngår i risikovurderingen og som kan inneha risiko for skade på menneske eller miljø. Dette kan f.eks. være maskiner, labaktivitet, verkstedarbeid, bruk av visse kjemikalier osv. Bryt ned hele aktiviteten/prosessen i enkeltdeler, beskriv kort hver del.

Aktivitetene velges enten som gjennomgang av en prosess (aktivitet fra f.eks. innkjøp av materiale, via bearbeiding til ferdig produkt), eller alle aktiviteter som oppleves mulig risikofylte i et gitt område. Deltagerne må bli enige om omfanget av aktivitetene som skal risikovurderes,

Man må ta stilling til
- "Risiko for hvem" - for ansatte, for nabomiljø, for enkeltindivider eller en gruppe, for bedriften, for samfunnet
- "Risiko fra hva" - hvilket anlegg, hvilken aktivitet, fra hvilke ulykkeskategorier
- "For hvilket tidsrom"- risikonivået kan variere over tid

Risiko er et potensielt tap, ikke et oppstått tap. Aktiviteter/prosesser som i utgangspunktet blir vurdert som lite risikofylte, kan medføre økt risiko under gitte forhold.

Ansvarlig
Hvem er ansvarlig for de enkelte aktiviteter/prosesser? Er det noen andre enheter som har ansvar eller oppgaver for å redusere risiko? Organisasjonskart kan brukes for å avklare ansvarsforhold

Lov, forskrift o.l.
Hvilke lover, forskrifter og andre myndighetskrav gjelder for aktiviteten/prosessen? Se HMS-håndboka, HMSRV-20/01, www.lovdata.no, www.arbeidstilsynet.no, www.hmsetatene.no, kommunale bestemmelser (www.trondheim.kommune.no)

Eksisterende dokumentasjon
Enheten skal finne fram eksisterende sentrale og lokale retningslinjer, tegningsunderlag, sertifikater, kranførerbevis, truckførerbevis, gjennomført opplæring, serviceavtaler, bruksanvisninger, sjekklister osv. Se også tidligere risikovurderinger og HMS-runder, lokalt HMS-hefte, labhåndbok, NTNUs stoffkartotek osv.

Sikringstiltak
Hva finnes av eksisterende sikringstiltak for området/utstyret? Feks. ventilasjon, personlig verneutstyr, nødstopp, merking/ilting, skillevegger osv.

NTNU	Risikovurdering	utarbeidet av	Nummer	Dato	
		HMS-avd.	HMSRV2603	4.3.2010	
		godkjent av	side	Erstatter	
HMS/KS		Rektor	1 av 2	9.2.2010	

Enhet:IKP **Dato:** 05.06.2011

Linjeleder: Professor Magne Hillestad

Deltakere ved risikovurderingen (m/ funksjon): Rebecca Williams

	Aktivitet fra kartleggings-skjemaet	Mulig uønsket hendelse/ belastning	Vurdering av sannsyn-lighet	Vurdering av konsekvens:				Risiko-verdi	Kommentarer/status Forslag til tiltak
ID nr			(1-5)	Menneske (A-E)	Ytre miljø (A-E)	Øk/ materiell (A-E)	Om-dømme (A-E)		
1	Datasimulering	Ingen	-	-	-	-	-	-	Ingen risiko knyttet til oppgaven

Sannsynlighet
1. Svært liten
2. Liten
3. Middels
4. Stor
5. Svært stor

Konsekvens
A. Svært liten
B. Liten
C. Moderat
D. Alvorlig
E. Svært alvorlig

Risikoverdi (beregnes hver for seg):
Menneske = Sannsynlighet x Konsekvens Menneske
Ytre miljø = Sannsynlighet x Konsekvens Ytre miljø
Økonomi/materiell = Sannsynlighet x Konsekvens Øk/matriell
Omdømme = Sannsynlighet x Konsekvens Omdømme

NTNU	Risikovurdering	utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2603	4.3.2010
		godkjent av	side	Erstatter
HMS/KS		Rektor	2 av 2	9.2.2010

Sannsynlighet vurderes etter følgende kriterier:

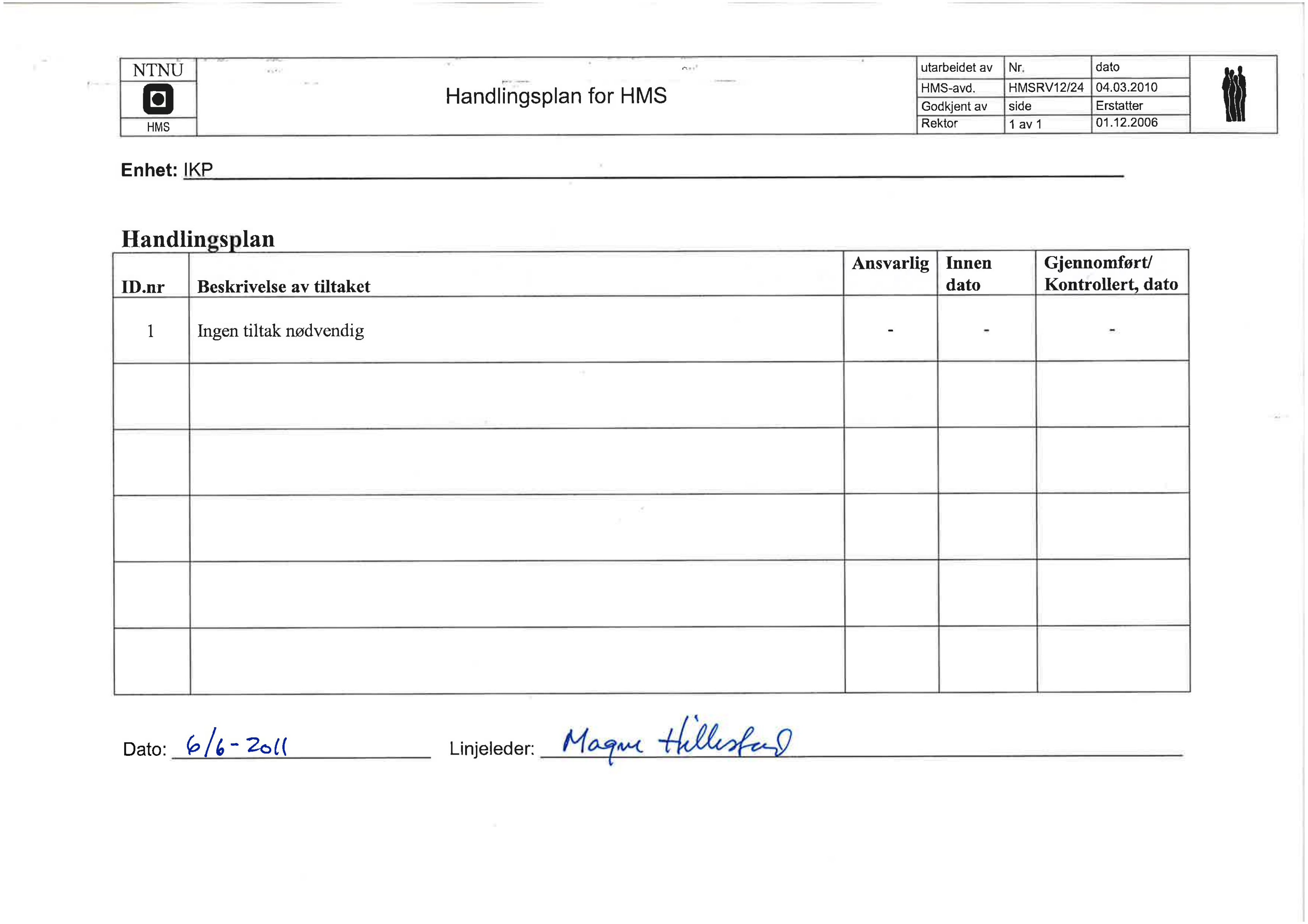
Svært liten 1	Liten 2	Middels 3	Stor 4	Svært stor 5
1 gang pr 50 år eller sjeldnere	1 gang pr 10 år eller sjeldnere	1 gang pr år eller sjeldnere	1 gang pr måned eller oftere	Skjer ukentlig

Konsekvens vurderes etter følgende kriterier:

Gradering	Menneske	Ytre miljø Vann, jord og luft	Øk/materiell	Omdømme
E Svært Alvorlig	Død	Svært langvarig og ikke reversibel skade	Drifts- eller aktivitetsstans >1 år.	Troverdighet og respekt betydelig og varig svekket
D Alvorlig	Alvorlig personskade. Mulig uførhet.	Langvarig skade. Lang restitusjonstid	Driftsstans > ½ år Aktivitetsstans i opp til 1 år	Troverdighet og respekt betydelig svekket
C Moderat	Alvorlig personskade.	Mindre skade og lang restitusjonstid	Drifts- eller aktivitetsstans < 1 mnd	Troverdighet og respekt svekket
B Liten	Skade som krever medisinsk behandling	Mindre skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1uke	Negativ påvirkning på troverdighet og respekt
A Svært liten	Skade som krever førstehjelp	Ubetydelig skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1dag	Liten påvirkning på troverdighet og respekt

Risikoverdi = Sannsynlighet x Konsekvens
Beregn risikoverdi for Menneske. Enheten vurderer selv om de i tillegg vil beregne risikoverdi for Ytre miljø, Økonomi/materiell og Omdømme. I så fall beregnes disse hver for seg.

Til kolonnen "Kommentarer/status, forslag til forebyggende og korrigerende tiltak":
Tiltak kan påvirke både sannsynlighet og konsekvens. Prioriter tiltak som kan forhindre at hendelsen inntreffer, dvs. sannsynlighetsreduserende tiltak foran skjerpet beredskap, dvs. konsekvensreduserende tiltak.



NTNU HMS	Handlingsplan for HMS	utarbeidet av	Nr.	dato	
		HMS-avd.	HMSRV12/24	04.03.2010	
		Godkjent av	side	Erstatter	
		Rektor	1 av 1	01.12.2006	

Enhet: IKP

Handlingsplan

ID.nr	Beskrivelse av tiltaket	Ansvarlig	Innen dato	Gjennomført/ Kontrollert, dato
1	Ingen tiltak nødvendig	-	-	-

Dato: 6/6-2011 Linjeleder: Magne Hillestad