Gas Dehydration



Thermodynamic simulation of the water/glycol mixture

February 2009

K10 – Aalborg University Esbjerg

Dan Laudal Christensen

Title page

An M.Sc.Eng project from:	Aalborg University Esbjerg (AAUE) Niels Bohrs Vej 8 6700 Esbjerg Denmark
In cooperation with:	Atkins Esbjerg Dokvej 3, sektion 4 6700 Esbjerg Denmark
M.Sc.Eng profile:	CCE – Computational Chemical Engineering
Semester:	10 th semester (4 th semester of M.Sc.Eng)
Semester theme:	Master thesis.
Project title:	Gas dehydration.
Subtitle:	Thermodynamic simulation of the water/glycol mixture.
Project period:	September 2008 – February 13th 2009
Authors:	Dan Laudal Christensen
Supervisor at AAUE:	Inge-Lise Hansen
Supervisor at Atkins Esbjerg:	Per Stoltze
Front page pictures:	Dehydration Plant

Esbjerg February 13th 2009.

Dan Laudal Christensen

Abstract

English:

The dehydration is an important process in offshore gas processing. The gas is dehydrated offshore to avoid dangers associated with pipeline transport and processing of wet gas. The problems include corrosion, water condensation and plugs created by ice or gas hydrates.

Thermodynamic simulation of gas dehydration is difficult due to the interaction between water and glycol. The interaction is due to non-ideal liquid behaviour of water and glycol mixture. The interaction is impossible to simulate with the normally used thermodynamic equations of state like Peng-Robinson.

To investigate the problems with the equations of state, the water/glycol mixture is simulated in MATLAB to investigate the phase behaviour of the mixture. The mixture is simulated with Peng-Robinson and Peng-Robinson-Stryjek-Vera equation of state. Peng-Robinson is calculated with both the van der Waals and the Wong-Sandler mixing rule. The Wong-Sandler mixing rule is used because it incorporates the excess Gibbs energy and activity coefficient that describes non-ideal liquid behaviour. The MATLAB simulations were unsuccessful in simulate the water/glycol mixture.

The entire dehydration process has also been simulated in HYSYS, with two thermodynamic packages. The HYSYS simulation is conducted with the glycol package, which is created specifically to simulate gas dehydration, and Peng-Robinson. Both thermodynamic packages are able to simulate the dehydration process, although it can not be determined witch package that gives the most accurate result.

Dansk:

Gas tørring er en vigtig proces i offshore gas behandling. Gas tørres offshore for at undgå de farer der er forbundet med rørledningstransport og proces behandling af våd gas. Disse problemer inkluderer korrosion, vand kondensering og blokering af rør og eller procesudstyr pga. is eller gas hydrater.

Termodynamisk simulering af gas tørring vanskeliggøres af den vekselvirkning der er mellem vand og glykol. Vekselvirkningen skyldes at vand og glykol danner en ikke idel væskeblanding. Denne vekselvirkning er umulig at simulere med de normalt benyttede termodynamiske tilstandsligninger som Peng-Robinson.

For at undersøge problemet med tilstandsligningerne er vand/glykol blandingen simuleret i MATLAB for at undersøge blandingens fase tilstand. Blandingen er simuleret med Peng-Robinson og Peng-Robinson-Stryjek-Vera tilstandsligningerne. Peng-Robinson er beregnet med både van der Waals og Wong-Sandler blandingsreglerne. Wong-Sandlers blandingsregel benyttes fordi den tager højde for Gibbs overskudsenergi og aktivitets koefficienterne, som beskriver ikke ideel væske blandinger. MATLAB simuleringerne var ude af stand til at simulere vand/glykol blandingen tilfredsstillende.

Den samlede gas tørrings proces simuleres også i HYSYS, med to forskellige termodynamiske pakker. HYSYS simuleringerne udføres med glykol pakken, der er speciel udviklet til at simulere gas tørring, og med Peng-Robinson. Begge termodynamiske pakker kan simulere gas tørrings processen, selvom det ikke kan afgøres hvilken pakke der giver det mest præcise resultat.

Preface

This report is a master thesis in M.Sc.Eng in Chemical Engineering at Aalborg University Esbjerg, under the profile Computational Chemical Engineering.

The project is provided by Atkins Oil and Gas Esbjerg, who has also been helpful with advice throughout the project

The report is intended for students in chemistry and chemical engineering, and others with interest in oilfield process engineering and thermodynamic simulation in MAT-LAB and thermodynamic process simulation in HYSYS. It is thus presumed that the reader is familiar with chemical and physical terminology.

References are made as [Bx], [Ax], [Wx] and [Ox] in the report, where x represent the source number and the letters the type of source. B stands for books, A for articles, W for web pages and O for other. The sources of the references can be seen in section 10. The articles and other used can be found on the attached CD in the path \SOURCES\.

Figures and tables are marked sequentially in each section of the report. Cross references are marked as:

Reference:	Refers to:
App. x	Appendix x
Figure s.x	Figure s.x
Table s.x	Table s.x
(s.x)	Equation s.x

Where s represents the section number and x again is the number of the reference.

There is a CD attached to the project. This CD contains the project, MATLAB programs, HYSYS simulations and results and the articles used in this project. Any references to the contents on the CD are made to the path where the file is placed. The CD is inserted between the report and the appendix.

In this report the SI-measuring units are used (with the exception of pressure that are given in bar and temperature which is in centigrade). Many operation parameters in the literature are given in oilfield units, if a value from the literature has been converted into SI-units the original value in oilfield units is given in brackets afterwards e.g. $\Delta T=5^{\circ} C$ (9° F).

The hydrocarbons in gas and oil are sometime named by there number of carbon atoms, e.g. C2 that stand for ethane. Some time the hydrocarbons are grouped by there size, making C2+ ethane and any hydrocarbons larger than ethane.

Index

1	INTI	RODUCTION	6
	1.1	OFFSHORE OIL AND GAS PRODUCTION	6
	1.2	PIPELINE TRANSPORT	6
	1.2.1	Water in gas	8
	1.2.2	Gas hydrates	9
	1.3	PROCESSES IN OFFSHORE PRODUCTION	10
	1.3.1	Separation	11
	1.3.2	Gas treatment	11
	1.3.3	Water treatment	13
2	INIT	IATING PROBLEM	14
3	GAS	DEHYDRATION	15
	31	DEHYDRATION METHODS	15
	3.1	Comparison of the methods	16
	3.2	WATER ABSORPTION	16
	3.2	Glycols used for dehydration	17
	3.2.2	Dry Gas	18
	3.3	THE GLYCOL DEHYDRATION PROCESS	19
	3.3.1	Process description	19
	3.3.2	Process plant	23
	3.4	PART DISCUSSION/CONCLUSION	23
1	тиг	ΦΜΟΡΥΝΑΜΙΟ	25
-	THE		23
	4.1	GENERAL THEORY	25
	4.1.1	Phase equilibrium	25
	4.1.2	Excess energy	26
	4.2	EQUATIONS OF STATE	28
	4.2.1	Cubic Equations of State	29
	4.2.2	Critical Data	29
	4.5	PENG-KOBINSON EQUATION OF STATE	30
	4.3.1	Multi component systems	3U 21
	4.3.2	Phase equilibrium	22
	4.5.5	Denic Podinicon Struer Veda FOS	34 34
	4.4	VONG-SANDI ED MIXING DII E	34
	4.5	PART DISCUSSION/CONCLUSION	37
5	SIM		20
3	51111		39
	5.1	MESH ELEMENTS	39
	5.1.1	Material balance	40
	5.1.2	Equilibrium	40
	5.1.3	Summation	41
	5.1.4	Enthalpy	41
	5.1.5	Freedom analysis	41
	5.2	FLASH SEPARATION	42
	5.2.1	Kachjora-Kice	42 11
	5.2.2	Henley-Kosen	44 16
	J.J 5 2 1	SIMULATION MODEL	40 16
	520	Input auta The MATLAR program	40 10
	5.3.2	THE MATLAD PLOGIUM.	40 57
	5.4 5 / 1	Case 1	57 57
	547	Case 2	57
	5.4.3	Case 3	58
		······································	

	5.4.4	Case 4	61
	5.4.5	Case 5	61
	5.4.6	б Case б	62
	5.5	PART DISCUSSION/CONCLUSION	62
6	FINA	AL AIM	63
7	PRO	CESS SIMULATION	64
	7.1	PROCESS SIMULATION PROGRAMS	64
	7.1.1	HYSYS	65
	7.2	SIMULATION MODEL	65
	7.2.1	Dehydration simulation	65
	7.2.2	Dehydration plant specifications	66
	7.2.3	Dehydration plant design	67
	7.2.4	Creating the simulation model	68
	7.3	SIMULATION RESULTS	71
	7.3.1	Case 11	71
	7.3.2	Case 12	73
	7.4	PART DISCUSSION/CONCLUSION	74
8	DISC	CUSSION	75
9	CON	ICLUSION	76
10	REF	ERENCES	77

Structure of the report.



1 Introduction

The offset for this report is the offshore oil and gas production in the Danish sector in the North Sea. The specific focus of the report is gas dehydration and the processes involved. This report is therefore introduced with a brief description of the Danish offshore sector and offshore processing of reservoir fluid into oil, gas and water. Because the main focus of the report is gas dehydration, the problems associated with water in the gas will also be described.

1.1 Offshore oil and gas production

There are two defining characteristics for the Danish offshore production, namely the shallow water with depths form 35 to 70 m [B1] and that the reservoirs are relatively thin layers with a limited permeability.

All the platforms in the Danish sector of the North Sea are either production or process platforms. Because of the low water depth all drilling are preformed with Jack-Up rigs leased with this specific purpose. This limits the cost of platform construction, because no space is needed for drilling operations, thus limiting the size of the platform.

Production platforms are either unmanned wellhead platforms or part of a process platform complex. Because of the water depth it is economically viable to install multiple platforms connected by walkways, or use them as support for bridge modules. The advantages of platform complexes, consisting of several smaller platforms, are the construction cost and a better safety in case of an emergency situation.

The problem with relative thin reservoirs has been solved with drilling of horizontal wells. The low reservoir permeability reduces the yield, to increase the yield enhanced recovery methods are used, primarily by water injection.

[**B**1]

1.2 Pipeline transport

In the Danish part of the North Sea all the platforms are connected by pipelines. From the wellhead platforms there are multiphase pipelines to the process platforms. On the process platforms the reservoir fluid is separated and treated as described in section 1.3. The oil and gas produced on the platforms is collected before it is exported to shore.

The oil is transported to the Gorm platform; here the oil export pipeline has its origin. There are two gas pipelines to the Danish shore; they start from Tyra East and Harald. There is an additional gas export pipeline on Tyra West; this pipeline is connected to the Dutch NOGAT pipeline. This enables export of the Danish excess gas production to the Netherlands. The platforms and pipelines in the Danish sector in the North Sea are illustrated in Figure 1-1.



Figure 1-1: The Danish sector of the North Sea [B2].

All the pipelines are regularly cleaned and inspected by pigs. Pigs come in two versions, one version is used to clean the pipelines by pushing all sediments before it; this type of pig is illustrated in Figure 1-2.



Figure 1-2: Pig used for pipeline cleaning. [W1]

The second type of pig is equipped with measuring instruments; this is used for inspections of the inside of the pipe. Common for all pigs is that they come in a wide range of sizes, fitting to the pipe that they are used in. The pig is driven forward by the flow in the pipeline.

There are several problems concerning pipelines, although similar, the problems are unique for gas, oil and multiphase flow pipelines. For gas pipelines the main problem is water in the gas.

1.2.1 Water in gas

Water is a problem in the gas phase, both in gas processing and in pipeline transport. The main problems with water in gas are:

- Corrosion
- Liquid water formation
- Ice formation
- Hydrate formation

In pipelines where it is known that the gas is wet, the problem can be countered. If it is known in the design phase the pipeline can be designed with more corrosion resistant materials or increased material thickness. If the problem occurs during production, the problem can be minimized by injecting inhibitors into the gas.

In dry gas pipelines the problems ought not to occur, but can occur in case of insufficient dehydration. If not discovered the problems are more serious here, because the pipelines are not designed for these conditions. When discovered inhibitors can be added until adequate dehydration is available again.

Liquid water in the pipeline is a problem, not only concerning liquids in compressors, but also a problem because the liquid water can create liquid plugs and increase corrosion.

Ice formation is only a problem when the temperatures are adequately low for ice to form. Ice is especially a problem in process equipment and valves, where the ice can create blockages. Ice are manly a problem in low temperature gas treatment like NGL recovery and gas liquefaction (see section 1.3.2). When low temperature gas treatment is utilized ultralow water contents are required, making the requirements for the dehydration process more stringent. Although ice is a problem, gas hydrates are often more troublesome.

[B3], [B4]

1.2.2 Gas hydrates

Gas hydrates are crystals of natural gas and water which can appear fare above the temperature where ice is formed. Gas hydrates are a caged structure containing a gas molecule like methane, the cage is formed by water through hydrogen bonding, as illustrated in Figure 1-3. Because the gas hydrate crystals are similar to ice crystals, the problems with gas hydrates are similar to those with ice, although gas hydrates are more troublesome because of the higher formation temperature.



Figure 1-3: Gas hydrate [W2].

Because hydrates can form in pipelines, large amounts of hydrates can be in the gas simultaneously; this can create plugs in the pipeline. Because of the potentially high hydrate contents in the gas the blockage can arise within minutes without any prior warning.

Prevention

Because of the potential dangers from gas hydrates they must be prevented. There are several methods to prevent gas hydrate formation, they are:

- Gas dehydration
- Raising the temperature
- Reducing the pressure
- Adding inhibitors

Gas dehydration is the most efficient way to prevent hydrate formation, but there may be practical limitation to the use of dehydration, e.g. one central dehydration unit. Gas dehydration will be treated further in section 3. If the gas stream can not be dehydrated, one of the other prevention methods must be used. Raising the temperature of a pipeline is very impractical, likewise is reducing the pressure, because such a reduction will reduce the pipeline flow. The only practical solution is therefore to ad inhibitors to the gas.

Inhibitors

Inhibitors acts as antifreeze in the gas, the usual inhibitors are:

- Alcohols
- Glycols

Methanol and monoethylene glycol (MEG) are the most commonly used inhibitors, low doses are often injected continuously in pipeline where hydrate formation is a problem. Higher doses of especially methanol are used temporally to dissolve hydrate plugs. MEG is more viscous than methanol, but has the advantage of being easier to regenerate from the gas than methanol, because methanol regeneration is usually not feasible.

MEG is the most commonly used glycol, because it is more efficient at a given mass concentration than diethylen glycol (DEG). DEG may nevertheless be used as inhibitor in the pipeline, but only if DEG also is the glycol used in the dehydration process afterwards. The different glycols are treated more thoroughly in section 3.2.1.

There are other possible inhibiters that prevent hydrate formation they are:

- Salts
- Ammonia
- Monoethanolamine

Salts are very rarely used because of the risk of corrosion and deposits. Ammonia is corrosive, toxic and can form solid deposits of carbonates obtained with carbon dioxide and water. Monoethanolamine is only attractive if it after pipe transport is used (and thereby recovered) for gas sweetening.

[B3], [B4]

1.3 Processes in offshore production

On the process platforms the main purpose is to process the reservoir fluid into oil, gas and water. This has to be done in such a manor that oil, gas and water meets the requirements before oil and gas can be exported and the water released into the sea.

Demands on oil may be the vapour pressure, to insure that no vapour is produced in the pipeline during transport to shore. Likewise a demand for gas may be no water dew in the pipeline; other gas demands may be the methane contents or heating value. For the oil and gas it is also a demand that the pipeline pressure is reached, before it can be exported from the platform. Water is a by-product, which needs to be cleaned before it can be disposed off.

To divide the reservoir fluid and insure that the requirements for the three phases are meet the reservoir fluid is processed. The process equipment can be divided into three parts.

- 1. Separation, including oil treatment and export
- 2. Gas treatment, including gas export
- 3. Water purification

The processes associated with these three systems may differ for different composition of reservoir fluid, especially for the gas treatment.

1.3.1 Separation

The first task when the reservoir fluid enters the process equipment is to separate it into its three phases. This is done in a series of three-phase separators, the number of which depends upon the inlet pressure of the reservoir fluid.

The first separator divides the reservoir fluid into its three phases. Subsequent separators are used to improve the purity of the oil and increase the gas recovery. When the first separation is completed there will still be gas dissolved in the oil, and probably also some water if the retention time is too small to ensure total separation between the two liquid phases.

Before the next separation, the pressure of the oil is lowered; this releases more of the dissolved gas. In case of additional water this will be separated off in the subsequent separators. This continues until the oil has the required purity, often two or three separators are enough. When the quality is as desired it is pumped to the pipeline pressure, before it is exported of the platform and to shore.

The gas released from the oil in the subsequent separators needs to be recompressed before it can be send to the gas treatment system. Figure 1-4 illustrates a separation system with two separators and gas recompression.



When gas is compressed, it is necessary to cool the gas and separate off any condensed liquid. In case of more separators than in Figure 1-4, each new separator will also be equipped with a compressor.

There will also be some liquid recycled from the gas treatment and the water purification system, but these streams have been excluded here for simplicity.

[B5]

1.3.2 Gas treatment

The purpose of gas treatment is to clean the gas for unwanted impurities and get it to the desired condition before it is exported. The composition of the gas is the decisive factor for which gas treatment procedures that are used. The most common cleaning procedures are gas sweetening, dehydration and hydrocarbon recovery; more seldom treat-

ments can be removal of inorganic elements. The purpose of cleaning the gas of its impurities is to improve the gas quality, avoid dangers to the process plant or pipeline from e.g. corrosion or enable the gas to be brought to the desired export condition.

After purification usually only compression is required, for the gas to reach its desired export condition. In rare cases the desired export condition could require liquefaction of the gas.

Gas sweetening

To minimize corrosion it is often necessary to remove acid components in the gas. It is manly CO_2 and H_2S that are removed, although in some cases other sulphur components are present in the gas and must therefore also be removed.

The most common sweetening procedure is absorption of the acid, with amines in an aqueous solution. Afterwards the rich amine solution is regenerated before it can be reused. Because the amines are in an aqueous solution, the sweet gas will be water saturated. Amine sweetening must therefore be conducted before gas dehydration.

Absorption is the most common procedure, but other procedures can also be used. E.g. membrane processes if only carbon dioxide are to be removed.

Dehydration

The problems with wet gas have already been described in section 1.2.1, where dehydration was deemed to be the most efficient way to solve the problems associated with wet gas. Dehydration is usually done by absorption, although other processes like adsorption, membrane processes and refrigeration may be used. The dehydration process will be described in section 3.

Hydrocarbon recovery

In gas with a high content of C2+ components, there is a risk of NGL (Natural Gas Liquids) formation. NGL may be removed from the gas to avoid liquid in the pipeline or to sell the more expensive NGL separately, instead of as a part of the gas.

Hydrocarbon recovery is preformed by cooling the gas below its dew point temperature, condensing the more heavy hydrocarbons in the gas, the condensed liquid is then removed in a separator. The easiest way to cool the gas is in heat exchangers; this is most efficient at high pressure.

Hydrocarbon recovery by cooling with heat exchangers may not yield the desired gas purity depending on the initial composition. In these cases the temperature can be lowered further by flashing the gas in a Jules-Thompson valve or in a turbo-expander. Because of the low temperatures achieved by flashing the gas, low water content is essential to prevent ice formation. Further improvements in hydrocarbon recovery can be achieved by distilling the liquid from the NGL recovery, thus recovering the methane condensed in this treatment.

Inorganic contents

If the gas quality is below pipeline quality because of contamination by inorganic elements, it is necessary to remove these impurities. Some of the inorganic components are only present in trace amounts, but can none the less create problems.

The most common inorganic component is nitrogen, the nitrogen contents might be high, either naturally or if nitrogen is used for injection into the reservoir to improve hydrocarbon recovery. Nitrogen can be recovered by cryogenic distillation, adsorption or membrane separation.

Radon may be present in the gas, it is radioactive, but with a half-life of 3.8 days the health problems from radon is minimal. The problem is that it decays into radioactive lead, which eventually will turn into non-radioactive lead. The result is that low-level radioactive materials will sediment in the process equipment and pipes; this constitutes a problem because cleaning produces radioactive waste.

Other contaminants

Benzene, Toluene, Ethylbenzene and Xylene (BTEX) are a problem because of environmental concerns. BTEX is removed from the gas during glycol dehydration, a smaller amount BTEX may also be removed during gas sweetening. When the glycol is regenerated the BTEX will be removed with the water, and thereby be vented to the atmosphere. BTEX are also a problem in cryogenic gas treatment because they can freeze like water.

BTEX can not be removed from the gas before the dehydration. The BTEX problem can be reduced by using a light glycol, because BTEX is more solvable in larger glycols. Alternatively the vented gas from the glycol regenerator can be flared or treated to remove the BTEX before it is vented to the atmosphere

Compression

The gas is compressed from the process pressure to the pipeline pressure in one or more steps, depending on the pressure difference. After each compression the gas is cooled and condensed liquids are separated off.

Liquefaction of the gas

Liquefied natural gas is an advantage when gas is stored or transported by non pipeline transport. Liquefaction of methane requires extensive refrigeration to temperatures as low as -161 °C (-258 °F). A very low water contents are therefore required.

[B3], [B4]

1.3.3 Water treatment

Unlike oil and gas treatment, water treatment is an environmental issue. Water is a waste product in oil and gas production; therefore it is released into the sea or used for well injection.

When water is separated off in the three-phase separation it still has a small hydrocarbon contents. This hydrocarbon contents constitutes no problem when the water is used for well injection, only when it is released into the sea. Because of environmental concerns the hydrocarbons needs to be removed from the water so the contents is below the threshold limit value for water released into the sea.

The hydrocarbons in the water are oil that did not separate off in the separators and dissolved gas. First the oil is removed using hydrocyclones; the oil is lead back to the separator system. The gas is removed from the water by decreasing the pressure thus decreasing the solvability in the water. The gas is separated off before the water is released into the sea.

2 Initiating problem

Removing the water from the gas offshore is essentially because it decreases the problems associated with water in the gas. This makes the dehydration process an essential part of the offshore gas treatment.

The first step in simulating a dehydration unit is investigating the process design. The next step is the simulation; the simulation is calculated with thermodynamic equations. The thermodynamic equations are originally created for non-polar components like hydrocarbons. The main part of simulation of the dehydration process is calculating the water/glycol interaction. Because of this mixtures complex nature, more specific thermodynamic equations that can describe the interaction must be used. This has resulted in the initiating problem:

What problems exist in thermodynamic simulation of gas dehydration with glycols?

To answer this problem nine other questions have been formulated, the answer of these will help to clarify some of the aspects associated with the initiating problem.

- What methods exist for gas dehydration?
- Why is glycol dehydration the preferred dehydration process?
- What requirements are given for the dehydration process?
- What processes are involved in the glycol dehydration process?
- What is the thermodynamic theory used in process simulation?
- What thermodynamic equations are used in process simulation?
- What is required to simulate the water/glycol mixture thermodynamically?
- What is required in process simulation calculations in addition to the thermodynamic equations?
- What is the result of simple phase equilibrium calculations of the water/glycol mixture?

The main focus of the initiating question and the subquestions is the simulation aspects of the dehydration process. The project is therefore limited to cover only this aspect of the dehydration process. Associated aspects like process safety, energy consumption and similar is out side the scope of this report.

3 Gas dehydration

There are four methods that are used for gas dehydration; they vary in efficiency and cost.

3.1 Dehydration methods

The methods used for gas dehydration are absorption, adsorption, membrane processes and refrigeration. The methods may be used by themselves or be combined to reach the desired water contents.

In dehydration by absorption water is removed by a liquid with strong affinity for water, glycols being the most common. The lean (dry) glycol removes the water from the gas in an absorption column known as a contactor. After the contactor the rich (wet) glycol must be regenerated before it can be reused in the contactor. The regeneration is done by distilling the glycol thus removing the water. With glycol absorption it is possible to lower the water contents down to approximately 10 ppm_{vol}, depending on the purity of the lean glycol [B4]. Gas dehydration by glycol absorption will be treated more thoroughly in section 3.3.

Dehydration by adsorption is done with a two bed system, where the beds are filled with adsorbents e.g. silica gel. The gas is lead through one of the adsorbers, where water is removed. Meanwhile the other adsorber is regenerated by blowing hot dry gas through it, this gas is then cooled and the water condenses. The Water is separated off and the gas is lead back to the wet gas, this is illustrated in Figure 3-1.



Figure 3-1: Gas dehydration by adsorption. [B4]

The efficiency of the adsorption process depends on the adsorbent used; there are several types of adsorbents available. The most efficient adsorbents are molecular sieves, this is aluminosilicates that have been altered to improve the adsorption characteristics, achieving a water contents as low as below 0.1 ppm_{vol} [B4].

In membrane processes the gas passes through a membrane that separates of the water. Membrane processes yields water content between 20-100 ppm_{vol} [B4]. The problem with membrane processes are that they only become economically viable compared to glycol absorption at flows below $1.5 \cdot 10^6$ Nm³/d (56 MMscfd) [B4].

Gas dehydration by refrigeration is a low cost dehydration method. Water condenses when the gas is cooled; the water is then removed in a separator. The separation method can be conducted numerous times. The method is most efficient at high pressure. The amount of water removed in the refrigeration process is often insufficient. Because of the low cost the refrigeration process are often used before the other dehydration processes.

3.1.1 Comparison of the methods

The two most efficient dehydration methods are absorption and adsorption. Absorption with glycol is the preferred dehydration method because it is more economical than adsorption. This is due to the following differences between absorption and adsorption:

- Adsorbent is more expensive than glycol.
- It requires more energy to regenerate adsorbent than glycol.
- Replacing glycol is much cheaper than replacing an adsorption bed.
- Glycol can be changed continuously, while changing an adsorption bed requires a shutdown.

Some low temperature treatment like liquefaction requires water content below what glycol plants can achieve. In these cases an adsorption plant is required, to minimize the cost this can be combined with a glycol plant that removes the majority of the water.

[B3], [B4]

3.2 Water absorption

The basis for gas dehydration by absorption is the absorbent; there are certain requirements for absorbents for gas dehydration:

- Strong affinity for water to minimize the required amount of absorbent.
- Low affinity for hydrocarbons to minimize hydrocarbon loss during dehydration.
- Low volatility at the absorption temperature to minimize vaporization losses.
- Low solubility in hydrocarbons, to minimize losses during absorption.
- Low tendency to foam and emulsify, to avoid reduction in gas handling capacity and minimize losses during absorption and regeneration.
- Low viscosity for ease of pumping and good contact between the gas and liquid phases.
- Large difference in volatility and boiling point compared to water to minimize vaporization losses during regeneration.
- Good thermal stability to prevent decomposition during regeneration.
- Low potential for corrosion.

The most critical property for a good dehydrator is off course the high affinity for water. The other criteria are used to evaluate potential absorbents practical applicability in the industry. In practice glycols are the most commonly used absorbents for dehydration.

3.2.1 Glycols used for dehydration

Glycol is a common name for diols; with the two alcohols these substances have a high affinity for water. In dehydration 1,2-ethandiol also known as Monoethylen glycol (MEG) and the small polymers of MEG (diethylen glycol (DEG), triethylen glycol (TEG) and tetraethylen glycol (TREG)) are the most commonly used for absorbents. Higher polymers than TREG is usually not used for dehydration because they become too viscous compared to the smaller polymers.

Table 3-1: Properties for MEG, DEG, TEG, TREG [B3], [B4] and water [B6].					
	MEG	DEG	TEG	TREG	Water
Formula	$C_2H_6O_2$	$C_4 H_{10} O_3$	$C_6H_{14}O_4$	$C_8H_{18}O_5$	H_2O
Molar mass [kg/kmol]	62.07	106.12	150.17	194.23	18.015
Normal boiling point [°C]	197.1	245.3	288.0	329.7	100.0
Vapor pressure @ 25 °C [Pa]	12.24	0.27	0.05	0.007	3170
Density @ 25 °C [kg/m ³]	1110	1115	1122	1122	55.56
Viscosity @ 25 °C [cP]	17.71	30.21	36.73	42.71	0.894
Viscosity @ 60 °C [cP]	5.22	7.87	9.89	10.63	0.469
Maximum recommended regenera-	163	177	204	224	-
Onset of decomposition [°C]	-	240	240	240	-

Properties for MEG, DEG, TEG, TREG and water are compared in Table 3-1.

In Table 3-1 the important values are the normal boiling point, vapor pressure, viscosity, maximum recommended regeneration temperature and the onset of decomposition.

The normal boiling point and vapor pressure has an influence in the distillation. The greater the difference for these properties between the top and bottom product, the easier it is to separate the components. The separation between glycol and water is important because the water contents in the lean glycol determine the amount of water the glycol can remove from the gas.

The larger polymers TEG and TREG have the best properties for dehydration. TREG has slightly better properties than TEG, but because of the additional cost of TREG, TEG offers the best cost/benefit compromise and is therefore the most commonly used glycol. [B3]

The decomposition temperature is the point where DEG, TEG and TREG begin to react with the water and decompose into MEG. The temperatures in [B4] (240 °C) originates from manufacturer data, but there are some doubts about these temperatures, because [B4] also give this temperature for TEG as 196 °C, and as 207 °C (404 °F) in [B5]. These temperatures are just below and above the maximum recommended regeneration temperature of 204 °C (400 °F), which is given in [B3], [B4], [B5] and [B7]. This indicates that some TEG will decompose at 204 °C. At this temperature there will be some hot-spots in the boiler where the temperature will exceed 207 °C.

17

When TEG decomposes it becomes MEG and DEG, therefore it will not influence the dehydration process, only give a slightly larger glycol loss because MEG and DEG are more volatile than TEG.

[B3], [B4], [B5], [B7]

3.2.2 Dry Gas

The efficiency of the dehydration is measured on the water contents in the dry gas. The dew-point temperature for the water in the gas is often a more useful parameter than the total water contents. The dew-point temperature must be below the minimum pipeline temperature, to avoid liquid in the gas pipeline. Figure 3-2 shows the relation between dew-point temperature and the water contents in the lean TEG at different temperatures.



A dew-point temperature of 6 to 11 $^{\circ}$ C (10 to 20 $^{\circ}$ F) below the desired dew-point may be used to insure against non-ideal situations.

The water dew-point may differ from the gas dew-point; the total gas dew-point may be influenced by other hydrocarbons in the gas. This can result in condensation of hydro-

carbons in the gas pipeline; this is also undesirable but much less so than water condensation.

[B3], [B4], [B5], [B7]

3.3 The glycol dehydration process

The dehydration process can be divided into two parts, gas dehydration and glycol regeneration. In dehydration water is removed from the gas using glycol and in regeneration water is removed from the glycol, before it can be reused for dehydration.

Dehydration

Dehydration always consists of an inlet scrubber and a contactor. Sometime it might be preferable to lower the gas inlet temperature before the dehydration, so an inlet cooler might also be used.

Regeneration

The main function in the glycol regeneration system can be divided into three:

- 1. Achieve the optimal pressure and temperature conditions for regeneration of the rich glycol.
- 2. Glycol regeneration.
- 3. Readjust glycol temperature and pressure for optimal dehydration conditions in the contactor.

Besides these three main points there are some additional features to be considered when designing a dehydration plant.

- Installing a flash separator before the regeneration column. This separator removes the majority of the hydrocarbons in the dissolved in the glycol.
- Filtering the rich glycol if there is solid particles or liquid hydrocarbons in the glycol
- Integrating the heat exchangers, so the lean glycol is cooled by heading the rich glycol, thus minimizing the energy consumption.
- Glycol make up to replace the glycol loss, e.g. in a storage tank.

Because of these considerations the design of the regeneration process varies with the design of the plant. The integration of heat exchangers is especially important, because this reduces the overall energy consumption of the plant.

3.3.1 Process description

The process is described by the equipment used in the glycol plant.

Inlet cooler

An inlet cooler may be used because dehydration is more efficient at low temperatures. Another benefit of inlet cooling is that some water (and hydrocarbons) in the gas will condense, and be removed in the inlet scrubber, instead of in the contactor.

An inlet cooler is used when the inlet gas temperature is higher than the desired temperature in the contactor. It is also a helpful tool in simulation if the temperature in the contactor needs to be optimized.

Inlet scrubber

The inlet scrubber removes free liquid and liquid droplets in the gas, both water and hydrocarbons. Removing liquid water in the scrubber decreases the amount of water that has to be removed in the contractor. This decreases the size of the contactor and the glycol needed in it, to reach the required conditions for the outlet gas. Liquid hydrocarbons are also a problem in the contactor because they increase the glycols tendency to foam, thereby decreasing the contactors efficiency and increasing the glycol loss in the contractor and from the regeneration system. Another problem is that hydrocarbons can be accumulated in the glycol polluting it and thereby decreasing the dehydration efficiency.

Contactor

The contactor is the absorption column where the gas is dried by the glycol. The lean glycol enters at the top of the contactor while the rich glycol is collected at the bottom of the contactor and sent to regeneration. The wet gas enters the contactor at the bottom, while the dry gas leaves at the top.

The required water dew-point of the dry gas dictates the lean glycol temperature and purity. This is illustrated in Figure 3-2. The glycol temperature into the contactor must be 3 to 11 $^{\circ}$ C (5 to 20 $^{\circ}$ F) higher than the gas entering the contactor to minimize hydrocarbon condensation into the glycol [B4], [B5].

At contactor temperatures below 10 °C (50 °F) TEG becomes too viscous, thus reducing the column efficiency. The contactor temperature may be as high as 66 °C (150 °F), but glycol vaporization loss is often deemed unacceptably high above 38 °C (100 °F) [B5].

The glycol flow into the contactor is dictated by the water content in the gas and numbers of trays in the column. A usual glycol flow is 0.017 to 0.042 m³ Lean TEG per kg water in the gas (2 to 5 gal TEG/lb Water). Contactor columns with four to six trays usually operate with 0.025 m³ TEG/kg Water (3 gal/lb), in larger columns with eight or more trays the flow is usually reduced to 0.017 m³/kg (2 gal/lb) [B4], [B5].

Flash valve

After the contactor column the pressure is reduced to the regeneration pressure by a flash valve. The pressure drop over this valve depends on the pressure in the contactor and the pressure loss in the pipes and equipment until the regeneration column.

Two places in the system unwanted gas is vented off the system, in the flash separator and the regenerator. To prevent blowback the pressure in these units must be higher than where they vent to. The slightly higher pressure also acts as a propellant in transporting the gas from the dehydration system.

Flash separator

It is a good idea to install a separator after the flash valve. Because of the decreased pressure hydrocarbons absorbed in the glycol will be released.

Without a separator the gas in the glycol will be released together with water in the regenerator. In the regenerator the water vapour is usually just vented to the atmosphere, thus increasing the plants emission of hydrocarbons. With a flash separator the hydrocarbon rich gas, can be used as process gas in the plant.

The pressure in the flash separator must be above the pressure in the system that the gas is vented too; the separator pressure will therefore differ between plants.

Filters

Filters are only necessary if there is a problem with solid particles or liquid hydrocarbons in the glycol.

Solid particles in the glycol accumulate, increasing the wear on the equipment and can create plugs in heat exchangers. Solid particles can easily be removed with sock filters, which can be made of cloth fabrics, paper or fibreglass.

Liquid hydrocarbons like condensate and BTEX can be removed from the glycol by activated carbon filters.

Heat exchangers

The numbers of heat exchangers varies with the design of the process plant. Because of the large temperature difference between the contactor and regenerator column, rich glycol needs to be heated while lean glycol must be cooled. With proper design of heat exchangers between the rich and lean glycol most of the energy can be conserved.

Rich glycol may be heated before and/or after the flash separation. Heating before the flash separator increases hydrocarbon recovery along with glycol loss. Heating before the flash separator is preferable if hydrocarbon contents in the rich glycol after the separation are too high.

Besides the heat exchangers the glycol is heated in the regenerator boiler. The lean glycol temperature may also need to be adjusted before it enters the contactor, this can be done with the dry gas or a cooler.

Regenerator

The regenerator is a distillation column, where glycol and water is separated. The rich glycol is preheated in heat exchangers before it is feed to the regenerator column.

At the top of the column is a partly condenser, this provide reflux thus improving the separation between water and glycol. The condenser also minimizes glycol loss from the regenerator. The remaining water vapour leaves the condenser and is vented to the atmosphere. The temperature in the condenser is given as 98.9 °C (210 °F) [B5].

The energy required to separate glycol and water is supplied by the reboiler at the regenerator column. The reboiler temperature is dictated by the glycol used for the dehydration as described in section 3.2.1. For TEG the recommended maximum temperature in the reboiler is 204 $^{\circ}$ C (400 $^{\circ}$ F). The Lean glycol is taken from the reboiler and is transferred to a storage tank before it is recycled or is recycled directly from the reboiler.

The pressure in the regeneration system is just above atmospheric pressure, this is to insure that no air can enter the system from the atmospheric vent.

The operating conditions for the regenerator influence the purity of glycol. At 204 °C TEG yields a lean glycol concentration of 98.6 wt% [B4]. If this purity of glycol is inadequate it can be improved by using more advanced regeneration techniques.

Some simple ways to increase the lean glycol purity is to ad a stripping gas to the regenerator or regenerate by vacuum distillation. Stripping gas can be added to the regenerator boiler or in a stripping column after the regenerator column. By adding stripping gas to the regenerator boiler the TEG purity can be increased up to 99.6 wt% [B5]. Vacuum distillation yields TEG purities up to 99.98 wt% [B4].

Stripping column

Glycol purities up to 99.9 wt% can be achieved by using a stripping column after the regenerator [B5]. The stripping gas from the top of the stripping column is routed to the regenerator boiler, like when stripping without the stripping column.

The stripping gas is usually nitrogen, dry gas or flash gas from the flash separator. The water can be removed from the stripping gas by cooling it well below waters dew-point. If hydrocarbon rich gas is used the gas from the regenerator must be dried or used as process gas.

To achieve 99.9 wt% pure glycol (or 99.6 wt% without the stripping column), the stripping gas flow must be $28.3 \text{ Nm}^3 \text{ gas/m}^3 \text{ TEG}$ (4 scf gas/gal TEG) [B5].

Cool stripping gas can be used in the stripping column, because the glycol needs to be cooled after the regenerator. If on the other hand stripping gas is added directly to the regenerator boiler it might be preferable to preheat the gas, to keep a uniform temperature in the boiler.

Glycol storage tank

This is an optional instalment that ensures a constant glycol flow to the contactor column. Because there will be a loss of glycol in the dehydration system, a storage tank can act a buffer to prevent insufficient glycol flow, and also be used to measure the glycol contents in the system.

Glycol circulation pump

Because of the pressure difference between the regenerator and the contactor, the glycol pressure needs to be increased. This is done with the glycol regeneration pump. The glycol is cooled below 80 °C before pumping to protect the pump.

[B3], [B4], [B5], [B7]

3.3.2 Process plant

A possible design of a dehydration plant is given in Figure 3-3.



The design in Figure 3-3 incorporates most of the units described in section 3.3.1, with the exception of the stripping column and a glycol storage tank. Dehydration plant design often differs from the one given in Figure 3-3; it can be the units or integration of heat exchangers.

[B3], [B4], [B5], [B7]

3.4 Part discussion/conclusion

There are four models for gas dehydration. They are refrigeration, membrane processes, adsorption and absorption. Refrigeration does in many case not remove enough water from the gas, it is however often used in combination with the other dehydration methods. Membrane processes is only economical for small gas flows, which excludes it in most dehydration cases. The adsorption yields the lowest water contents in the gas, dependent on the adsorbent. Even though the absorption process can not remove as much water as adsorption it is often the preferred method. This is because it removes sufficient water to reach the required criteria for the dry gas, as well as gives a better cost/benefit result than the adsorption process. In some cases where low temperature gas treatment is involved adsorption dehydration is required. In those cases the cost is often reduced by combining adsorption plant with an absorption plant.

The efficiency of a dehydration process is evaluated by the water contents in the gas after the dehydration. The water contents after the dehydration is often given as the water dew-point, this is to insure that no water will condense in the pipeline. The water dew-point is therefore more practicable because it is directly comparable with the pipeline operating conditions.

The glycol dehydration process can be divided into two parts. First lean glycol dries the wet gas, thereby making the glycol rich. In the second part of the process water is removed from the rich glycol making it lean once again. In the second part of the plant pressure and temperature is change to achieve the optimal operating conditions both in the contactor and the regenerator. The changing of pressure and temperature creates a wide range of possibilities of the final design of the plant.

The purpose in this report is the simulation of the dehydration process. This requires calculations of the interaction between the components in the dehydration process; these are done with thermodynamic calculations.

4 Thermodynamic

Thermodynamics are used to describe the relationship between energy, temperature, pressure and volume for pure components and mixtures. The classic example of the relationship between these different factors is the steam engine, where energy is transformed from heat to work through waters thermodynamic properties.

In process simulation thermodynamics are used to calculate the relationship between energy consumption/release, pressure, temperature, volume and phase equilibrium.

4.1 General theory

There are some general relationships in thermodynamic calculations; these will be described in this section. This is the phase equilibrium calculations, which are used in determining phase changes and component distribution between phases. It is the excess energy that influences the behaviour of non-ideal liquids. And finally it is the equations of state, which are used to calculate the behaviour of chemical components.

4.1.1 Phase equilibrium

In a vessel with two (or more) phases, there will be equilibrium between the phases, given that there is sufficient time for the system to reach equilibrium. The phase equilibrium depends on the temperature, pressure and phase composition.

The basis for phase equilibrium calculation is the fugacity f. The fugacity describes components tendency to prefer one phase over another. Components moves from phases with high fugacity to phases with low fugacity. At equilibrium the fugacity of the components are identical in each phase as illustrated by (4.1).

$$f_i^I = f_i^{II} \tag{4.1}$$

The fugacity is form of adjusted pressure, the relationship between the fugacity and the actual pressure is described by the fugacity coefficient φ , as shown in (4.2)

$$\varphi = \frac{f}{P} \tag{4.2}$$

For an ideal gas the fugacity equals the pressure, making the fugacity coefficient one. The size of the fugacity coefficient can therefore be used to describe the non-ideal behaviour of the vapour phase. Even though φ describes the non-ideal vapour behaviour it also influences the fugacity in liquid phases. The fugacity for the gas phase is defined as (4.3) and as (4.4) for the liquid phase.

$$f_i^V = P \cdot \varphi_i^V \cdot y_i \tag{4.3}$$

$$f_i^L = P \cdot x_i \cdot \varphi_i^L \cdot \gamma_i \tag{4.4}$$

The activity coefficient γ describes the non-ideal behaviour of liquids, which is due to the excess energy. The relation between activity constant and Gibbs excess energy is given in (4.5).

$$\ln \gamma_i (T, P, \underline{x}) = \frac{\overline{G}_i^{ex} (T, P, \underline{x})}{RT}$$
(4.5)

The excess energy will be treated further in section 4.1.2. Ideal liquid behaviour is often assumed, making the excess energy zero, and the activity coefficient one. The thermodynamic equations given in this report will be for ideal liquids, unless something else is noted.

In calculations the excess energy is often zero, because ideal liquid behaviour is assumed. The equations given in these sections are all for ideal liquids, unless something ells is noted.

Equilibrium calculations

Phase equilibrium is used for five types of calculations namely dew- and bubble-point temperature and pressure and for flash calculations. Dew-point calculations are either at constant pressure or temperature, while the other is calculated for a known gas mixture. Bubble-point calculations are similar to the dew-point calculations with the difference that it is for a known liquid mixture. The Flash calculations is used to calculate the phase composition of a known mixture at a given temperature and pressure.

At equilibrium the pressure is identical in all phases, therefore (4.1), (4.3) and (4.4) can be rewritten as (4.6).

$$\boldsymbol{\varphi}_i^V \cdot \boldsymbol{y}_i = \boldsymbol{\varphi}_i^L \cdot \boldsymbol{x}_i \tag{4.6}$$

In all these calculations it is often necessary to have a value for the distribution of the individual components between the two phases; this is described by the equilibrium ratio K_i .

$$K_i = \frac{y_i}{x_i} = \frac{\varphi_i^L}{\varphi_i^V} \tag{4.7}$$

The K-value can be used to calculate the amount of a component in the gas phase when the content in the liquid phase is known. In flash calculations it might be necessary to have an initial estimate for K to solve the problem by updating K in an iterative process.

[B8]

4.1.2 Excess energy

The excess energy is used to describe the non-ideal behaviour of liquid mixtures. Non-ideal behaviour is most noticeable when mixing liquids to a non-ideal mixture. There are two ways non-ideal behaviour can manifest; this is described in Table 4-1.

Table 4-1: Volume and enthalpy for ideal and non-ideal mixtures		
Property	Ideal mixtures	Non-Ideal mixtures
Volume	$V = \sum_i x_i \cdot \underline{V}_i$	$V = \Delta V + \sum_{i} x_i \cdot \underline{V}_i$
Enthalpy	$H = \sum_{i} x_{i} \cdot \underline{H}_{i}$	$H = \Delta H + \sum_{i} x_{i} \cdot \underline{H}_{i}$

When mixing one or both of these properties manifests for non-ideal mixtures. The non-ideal mixtures are actually real mixtures, but because most mixtures are assumed ideal, this distinction is used to emphasize the non-ideal mixtures.

Non-ideal mixtures consist mainly of polar or partly polar components, where the polar attraction between the molecules attract or repulses each other.

The thermodynamic equations used in the calculations are mainly designed for hydrocarbon mixtures, where the liquid interaction between the molecules is minimal. Therefore liquid mixtures are often assumed to be ideal.

Activity coefficient calculations

To calculate the data for non-ideal mixtures, the value of the excess energy is required; these data are based on experimental data. The activity coefficient is calculated by fitting the data to an equation. There are different equations the data can be fitted to; the one that gives the best fit is used to calculate the activity coefficient.

The most common equations are Margules (4.8), van Laar (4.9), Wilson (4.10) and NRTL (4.11), these equations gives the activity coefficients in two component systems.

$$\ln \gamma_{1} = \left[A_{12} + 2(A_{21} - A_{12})x_{1} \right] x_{2}^{2}$$

$$\ln \gamma_{2} = \left[A_{21} + 2(A_{12} - A_{21})x_{2} \right] x_{1}^{2}$$
(4.8)

$$\ln \gamma_{1} = A_{12} \left(\frac{A_{21} x_{2}}{A_{12} x_{1} + A_{21} x_{2}} \right)^{2}$$

$$\ln \gamma_{2} = A_{21} \left(\frac{A_{12} x_{1}}{A_{12} x_{1} + A_{21} x_{2}} \right)^{2}$$
(4.9)

$$\ln \gamma_{1} = \ln \left(x_{1} + \Lambda_{12} x_{2} \right) + x_{2} \left(\frac{\Lambda_{12}}{x_{1} + \Lambda_{12} x_{2}} - \frac{\Lambda_{21}}{\Lambda_{21} x_{1} + x_{2}} \right)$$

$$\ln \gamma_{2} = -\ln \left(\Lambda_{21} x_{1} + x_{2} \right) - x_{1} \left(\frac{\Lambda_{12}}{x_{1} + \Lambda_{12} x_{2}} - \frac{\Lambda_{21}}{\Lambda_{21} x_{1} + x_{2}} \right)$$

$$\ln \gamma_{1} = x_{2}^{2} \left[\tau_{21} \left(\frac{G_{21}}{x_{1} + x_{2} G_{21}} \right)^{2} + \left(\frac{\tau_{12} G_{12}}{(x_{2} + x_{1} G_{12})^{2}} \right) \right]$$

$$(4.10)$$

$$(4.11)$$

$$\ln \gamma_2 = x_1^2 \left[\tau_{12} \left(\frac{G_{12}}{x_1 G_{12} + x_2} \right)^2 + \left(\frac{\tau_{21} G_{21}}{\left(x_1 G_{12} + x_2 \right)^2} \right) \right]$$
(4.11)

Fitting data to the equations are a comprehensive task. The experimental data is therefore collected to databases and books, an example is DECHEMA. Unfortunately these databases does not include all mixtures, the data is limited to the most common components, with some sporadic data for more rare components.

[B8], [B9]

Multicomponent Excess Gibbs free energy NRTL model

The notation given in (4.11) is the NRTL model for two components; there are also other notation methods for the model. The NRTL calculations for multicomponent systems are given in (4.12) and (4.13)

$$\frac{\underline{G}^{ex}}{RT} = \sum_{i=1}^{c} x_i \frac{\sum_{j=1}^{c} \tau_{ji} G_{ji} x_j}{\sum_{k=1}^{c} G_{ki} x_k}$$
(4.12)

$$\ln \gamma_{i} = \frac{\sum_{j=1}^{c} \tau_{ji} G_{ji} x_{j}}{\sum_{k=1}^{c} G_{ki} x_{k}} + \sum_{l=1}^{c} \frac{x_{l} G_{il}}{\sum_{m=1}^{c} x_{m} G_{ml}} \left(\tau_{il} - \frac{\sum_{j=1}^{c} x_{n} \tau_{nl} G_{nl}}{\sum_{o=1}^{c} x_{o} G_{ol}} \right)$$
(4.13)

The definition of the different terms in (4.12) and (4.13) are given in (4.14) and (4.15).

$$G_{ij} = \exp\left(-\tau_{ij}\alpha_{ij}\right) \tag{4.14}$$

$$\tau_{ij} = \frac{a_{ij} + b_{ij}T}{RT} \tag{4.15}$$

The input data is given by a_{ij} , b_{ij} and α_{ij} for the different components, the pure component values are zero, the value of $\alpha_{ij} = \alpha_{ji}$.

[B8], [O1]

4.2 Equations of State

The behaviour of a gas can be described by the compressibility Z.

$$Z = \frac{PV}{RT}$$
(4.16)

Z describes the relationship between the temperature, pressure and molar volume of a gas. A specific case of (4.16), namely for Z=1 is better known as the ideal gas law. The ideal gas law (4.17), is the first example of an equation of state (hence EOS).

$$P = \frac{RT}{\underline{V}} \tag{4.17}$$

The ideal gas law is a very simple form of an EOS, which does not take into account that most components are not ideal gasses. All gasses do however approach ideal gas state, when the pressure decreases, or the molar volume approaches infinity.

When dealing with real fluids the ideal gas law must be replaced by a more accurate EOS. The first EOS valid for non-ideal situations that have been created is van der Waals EOS (hence VDW) (4.18).

$$P = \frac{RT}{\underline{V} - b} - \frac{a}{V^2} \tag{4.18}$$

In (4.18) a and b are correction factors to account for non-ideal conditions. The a-value describes the size of the molecule in the fluid, while the b-value gives the volume of the molecule at infinite pressure, or at absolute zero temperature. a and b is defined from the components critical data. The VDW was a definitive improvement of the ideal gas law, although the accuracy could be improved further. One such improvement was to

make the EOS temperature dependent. This has been the basis for several more accurate EOS, most common are Soave-Redlich-Kwong EOS (SRK) (4.19) and Peng-Robinson EOS (PR) (4.20).

$$P = \frac{RT}{\underline{V} - b} - \frac{a(T)}{\underline{V}(\underline{V} + b)}$$
(4.19)

$$P = \frac{RT}{\underline{V} - b} - \frac{a(T)}{\underline{V}(\underline{V} + b) + b(\underline{V} - b)}$$
(4.20)

The problem with EOS like SRK and PR in this form is that it is difficult to use in calculation, the equations have therefore been rearranged to a cubic form.

[B8]

4.2.1 Cubic Equations of State

In the cubic EOS the classic forms of the EOS have been rearranged as functions of the compressibility factor Z. The cubic equations can be generalized to (4.21).

$$Z^3 + \alpha Z^2 + \beta Z + \gamma = 0 \tag{4.21}$$

(4.21) is valid for different EOS, they only differ is the specification of α , β and γ , Z is defined as (4.16). When α , β and γ is known Z can be calculated e.g. using Newton-Raphson method, the definition of α , β and γ for different EOS is given in Table 4-2.

	Table 4-2: α , β and γ in selected EOS		
	VDW	SRK	PR
α	-1-B	-1	-1+B
β	А	$A-B-B^2$	$A-3B^2+2B$
γ	-AB	-AB	$-AB+B^2+B^3$

The variables A and B in Table 4-2 are described in (4.22) and (4.23).

$$A = \frac{aP}{\left(RT\right)^2} \tag{4.22}$$

$$B = \frac{bP}{RT} \tag{4.23}$$

A and B depends on a and b, like the classic form of the EOS. The calculation of a and b are based on the critical data for the components.

[B8]

4.2.2 Critical Data

The factors a and b in the EOS is used to describe how real components react. They are based on the critical temperature (T_C) and pressure (P_C) for the components.

VDW was the first EOS to use the critical date in the calculations, but accuracy still needed some improvement, especially at the critical point. The compressibility at the critical point (Z_C) in VDW is always 0.375, while Z_C for most real fluids ranges from 0.23 to 0.31.

<u>·</u>	$-\frac{1}{2}\left(\frac{1}{2}+\frac{1}{2}\right)$
P = RT	a(T)
$I = \frac{1}{V-h}$	$\overline{V(V+b)} + b(V-b)$

The problem can be rectified by specifying one more parameter, Z_C is an obvious suggestion, unfortunately the parameter is difficult to determine with great accuracy for many substances. Therefore another parameter, the acentric factor ω , which is easier to measure, has been specified (4.24).

$$\omega = -1.0 - \log_{10} \left[\frac{P^{vap} \left(T_r = 0.7 \right)}{P_C} \right]$$
(4.24)

 $P^{vap}(T_r=0.7)$ is the vapour pressure at the reduced temperature equal to 0.7, where the reduced temperature T_r is described by (4.25).

$$T_r = \frac{T}{T_c} \tag{4.25}$$

The critical data is used to calculate a and b, this is done with equations that are specific for the different EOS. PR is considered the most accurate of the EOS described in this section VDW and SRK will not be treated any further in this report.

[B8]

4.3 Peng-Robinson Equation of State

The unknown factors in solving the cubic EOS now is a and b, they are calculated from the critical data for the components. In PR (4.20) a is a function of temperature, it is given in (4.26).

$$a(T) = 0.457235 \frac{R^2 T_c^2}{P_c} \cdot \alpha(T)$$

$$(4.26)$$

The last term in (4.26) is given in (4.27).

$$\alpha(T) = \left(1 + \kappa \left(1 - \sqrt{\frac{T}{T_c}}\right)\right)^2 \tag{4.27}$$

In (4.27) κ is a constant that depends on the acentric factor as given in (4.28).

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \tag{4.28}$$

Compared to the calculation of a, the calculation of b is simple (4.29).

$$b = 0.077796 \frac{RT_c}{P_c}$$
(4.29)

The calculations with PR thus far have only been for one component systems. For mixtures the a and b values must be combined to a value for the mixture.

[B8], [A1]

4.3.1 Multi component systems

There are only a slightly difference between one component systems and multi component systems. The difference is in the interaction between the different components in the mixture. The molecular interactions in the mixture are calculated by mixing the individual a and b values to a_m and b_m which are valid for the entire mixture. In a multiphase system, the components must be mixed in each phase.

Van der Waals one-fluid mixing rules

The most commonly used mixing rule are van der Waals mixing rule, it is defined by (4.30) and (4.31).

$$a_m = \sum_i \sum_j x_i x_j a_{ij} \tag{4.30}$$

$$b_m = \sum_i \sum_j x_i x_j b_{ij} \tag{4.31}$$

The mixing rule are based on the interaction parameter between components i and j (a_{ij} and b_{ij}). The interaction parameter are calculated from the pure component a_{ii} and b_{ii} values, this is done with the combining rules (4.32) and (4.33).

$$a_{ij} = \sqrt{a_{ii}a_{jj}} \left(1 - k_{ij}\right) = a_{ji}$$
(4.32)

$$b_{ij} = \frac{1}{2} (b_{ii} + b_{jj}) = b_{ji}$$
(4.33)

If (4.33) is inserted in (4.31), the mixing rule for b_m becomes (4.34).

$$b_m = \sum_i x_i b_i \tag{4.34}$$

In (4.32) a new parameter k_{ij} is introduced, this is the binary interaction parameter. This parameter is used to obtain better results when calculating mixtures; k_{ij} is fund by fitting the results to real mixture data. The problem with k_{ij} are the same that arises every time a new parameter is introduced, the improved accuracy requires additional input data. Unlike ω , the introduction of k_{ij} only comes with limited input data and they are all limited to the most common components. Because of the difference in accuracy in the different EOS, the k_{ij} value is specific for each EOS. The problem with the missing k_{ij} values are often solved by setting this value equal to zero, or estimating a value from data for similar mixtures.

When a_m and b_m are calculated the EOS can be solved, and the compressibility used to calculate the PTV data. There are other important uses of the EOS than just calculating the PTV data.

[B8]

4.3.2 Phase equilibrium

When calculating the phase equilibrium, the fugacity for the different components in each phase is calculated. There are formulas to calculate the fugacity coefficient from the compressibility factor, the formulas differs with the different EOS.

The fugacity coefficient for PR can be calculated with (4.35), that are for the liquid phase. The equation is identical to the equation for the vapour phase, although the constants used must be for the calculated phase.

$$\ln \overline{\varphi}_{i}^{L}(T, P, \underline{x}) = \frac{b_{i}}{b_{m}} \left(Z^{L} - 1 \right) - \ln \left(Z^{L} - \frac{b_{m}P}{RT} \right)$$
$$- \frac{a_{m}}{2\sqrt{2}b_{m}RT} \left[\frac{2\sum_{j} x_{j}a_{ij}}{a_{m}} - \frac{b_{i}}{b_{m}} \right] \ln \left[\frac{Z^{L} + \left(1 + \sqrt{2} \right) \frac{b_{m}P}{RT}}{Z^{L} + \left(1 - \sqrt{2} \right) \frac{b_{m}P}{RT}} \right]$$
(4.35)

For one component systems (4.35) is reduced to (4.36).

$$\ln \varphi^{L} = (Z^{L} - 1) - \ln \left(Z^{L} - \frac{bP}{RT} \right) - \frac{a}{2\sqrt{2}bRT} \ln \left[\frac{Z^{L} + (1 + \sqrt{2})\frac{bP}{RT}}{Z^{L} + (1 - \sqrt{2})\frac{bP}{RT}} \right]$$
(4.36)

The phase equilibrium calculation is vital in the calculation of the component distribution in multi-phase situation.

4.3.3 Departures

When the energy in the system are changed the calculation of the conditions at the new energy level are calculated by departure functions. Enthalpy and entropy are state functions, making the energy difference between the two states independent of path. The calculation of energy change is therefore conducted by departures to ideal gas state. The idea with the departure to ideal gas conditions are that enthalpy and entropy of ideal gasses are well defined, while real-fluid energy changes are undefined. The enthalpy and entropy change of ideal gas depends on the heat capacity (4.37).

$$C_{P} = a + bT + cT^{2} + dT^{3}$$
(4.37)

The definition of C_P may vary from (4.37), depending on which source that are used. In some cases the factor dT^3 may be removed or additional factors added (eT^4 , fT^5 ...).

The relationship between the heat capacity and the enthalpy are given in (4.38) and with the entropy in (4.39).

$$\Delta H^{IG} = \int_{T_1}^{T_2} C_P dT$$
 (4.38)

$$\Delta S^{IG} = \int_{T_1}^{T_2} \frac{C_P}{T} dT$$
 (4.39)

(4.38) and (4.39) are only valid at constant pressure, to achieve this, the departure function for the enthalpy reduces the pressure to zero, while the entropy departure increases the molar volume to infinity. The ideal gas departure for mixtures equals the sum of the departure energy of the individual components.

Like with the fugacity the departure calculations depend on whether the departures are for pure components or mixtures. The one component departures are given in (4.40) and (4.41).

$$H(T,P) - H^{IG}(T,P) = RT(Z-1) + \frac{T\left(\frac{da}{dT}\right) - a}{2\sqrt{2}b} \ln\left[\frac{Z + \left(1 + \sqrt{2}\right)\frac{bP}{RT}}{Z + \left(1 - \sqrt{2}\right)\frac{bP}{RT}}\right]$$
(4.40)

$$S(T,P) - S^{IG}(T,P) = R \ln\left(Z - \frac{bP}{RT}\right) + \frac{\frac{da}{dT}}{2\sqrt{2}b} \ln\left[\frac{Z + \left(1 + \sqrt{2}\right)\frac{bP}{RT}}{Z + \left(1 - \sqrt{2}\right)\frac{bP}{RT}}\right]$$
(4.41)

Because of the derivative da/dT, the departure function depends on which EOS that are used, the derivative function for PR are given in (4.42).

$$\frac{da}{dT} = -0.45724 \frac{R^2 T_c^2}{P_c} \kappa \sqrt{\frac{\alpha(T)}{TT_c}}$$
(4.42)

The departure function for mixtures are almost identical to the one component departures with the exception that the mixture properties must be used, as defined in (4.43) and (4.44).

$$\underline{H}(T, P, \underline{x}) - \underline{H}^{IGM}(T, P, \underline{x}) =$$

$$RT(Z_m - 1) + \frac{T\left(\frac{da_m}{dT}\right) - a_m}{2\sqrt{2}b_m} \ln\left[\frac{Z_m + (1 + \sqrt{2})\frac{b_m P}{RT}}{Z_m + (1 - \sqrt{2})\frac{b_m P}{RT}}\right]$$

$$\underline{S}(T, P, \underline{x}) - \underline{S}^{IGM}(T, P, \underline{x}) =$$

$$R \cdot \ln\left(Z_m - \frac{b_m P}{RT}\right) + \frac{\frac{da_m}{2\sqrt{2}b_m}}{2\sqrt{2}b_m} \ln\left[\frac{Z_m + (1 + \sqrt{2})\frac{b_m P}{RT}}{Z_m + (1 - \sqrt{2})\frac{b_m P}{RT}}\right]$$
(4.43)
$$(4.44)$$

The main problem in departure calculations for mixtures is the derivative for mixtures. The mixture derivative are calculated with (4.45) and (4.46).

$$\frac{da_m}{dT} = \sum_i \sum_j x_i x_j \frac{da_{ij}}{dT}$$
(4.45)

$$\frac{da_{ij}}{dT} = \frac{a_{ij}\frac{da_{ii}}{dT} + a_{ii}\frac{da_{jj}}{dT}}{2a_{ii}}$$
(4.46)

If the mixture is multiphase, the mixture departure must be calculated for each individual phase.

[B8]

4.4 Peng-Robinson-Stryjek-Vera EOS

The Stryjek-Vera modification of PR are introduced to increase the accuracy of calculations for polar components in PR, thus creating the Peng-Robinson-Stryjek-Vera EOS (hence PRSV). The basis for PRSV is a modification of the κ calculations in (4.28). The PRSV calculation of κ is more accurate, and includes a term which describes the polar behaviour of polar components. The PRSV modification are described by (4.47) and (4.48).

$$\kappa = \kappa_0 + \kappa_1 \left(1 + \sqrt{T_r} \right) \left(0.7 - T_r \right) \tag{4.47}$$

$$\kappa_0 = 0.378893 + 1.4897153\omega + 0.17131848\omega^2 + 0.0196554\omega^3$$
(4.48)

The κ_1 parameter is individual for different components, this is especially important for pure polar components.

The problem with introducing a new parameter like κ_1 are the same as with the binary interaction parameter k_{ij} , namely that only limited additional data are available. The value must be fitted to experimental data, be estimated from similar components or assumed to be zero.

The introduction of PRSV requires some modification in the departure calculations, because the derivative da/dT depends on the κ function. The new derivative is (4.49).

$$\frac{da}{dT} = 0.45724 \frac{R^2 T_C}{P_C} \left(\frac{1 - \sqrt{T_r}}{T_r} \right) \left[\left\{ \left(\frac{0.7 - T_r}{2} \right) - \left(1 + \sqrt{T_r} \right) \sqrt{T_r} \right\} \kappa_1 \left(1 - \sqrt{T_r} \right) - \kappa \right] (4.49) \right]$$

If $\kappa_1 = 0$ the derivative can be simplified to (4.50)

$$\frac{da}{dT} = -0.45724 \frac{R^2 T_C}{P_C} \kappa \left(\frac{1 - \sqrt{T_r}}{T_r}\right)$$

$$\tag{4.50}$$

[B8]

4.5 Wong-Sandler mixing rule

Satisfactorily results can be reached for ideal liquid mixtures, using van der Waals one fluid mixing rule. For non-ideal mixtures the accuracy is limited by the excess energy. In these cases a more accurate mixing rule incorporating the excess energy must be used. The Wong-Sandler mixing rule can be used together with all EOS, only one parameter needs to be adjusted.

The Wong-Sandler mixing rule incorporates the excess energy and gives a better fit to the boundary condition for a and b than van der Waals mixing rule. The Wong-Sandler mixing rule is given in (4.51) and (4.52).

$$\frac{a_m}{RT} = Q \frac{D}{1-D} \tag{4.51}$$

$$b_m = \frac{Q}{1 - D} \tag{4.52}$$

Where Q and D are given by (4.53) and (4.54).
$$Q = \sum_{i} \sum_{j} x_{i} x_{j} \left(b_{ij} - \frac{a_{ij}}{RT} \right)$$
(4.53)

$$D = \sum x_i \frac{a_i}{b_i RT} + \frac{\underline{G}_{\gamma}^{ex}(T, P, \underline{x})}{C^* RT}$$
(4.54)

The constant C^* in (4.54) is specific for which EOS the mixing rule is used with. For VDW it is equal to -1 while the value for PR it is:

$$C^* = \frac{\ln\left(\sqrt{2} - 1\right)}{\sqrt{2}} = -0.62323$$

The cross term in (4.53) is defined by the combining rules (4.55) and (4.56), which one used is optional.

$$b_{ij} - \frac{a_{ij}}{RT} = \sqrt{\left(b_{ii} - \frac{a_{ii}}{RT}\right) \left(b_{jj} - \frac{a_{jj}}{RT}\right)} \left(1 - k_{ij}\right)$$
(4.55)

$$b_{ij} - \frac{a_{ij}}{RT} = \frac{1}{2} \left(b_{ii} + b_{jj} \right) - \frac{1}{RT} \sqrt{a_{ii} a_{jj}} \left(1 - k_{ij} \right)$$
(4.56)

There is a clear resemblance between (4.56) and van der Waals mixing rule, this is due to the fact that they are defined from the same boundary condition. When a_m and b_m are calculated the EOS can be solved, and the compressibility used to calculate the PTV data.

Fugacity for Wong-Sandler mixing rule

The fugacity calculation is more complicated when using the Wong-Sandler mixing rule, due to the fact that the definition of a_m and b_m is changed. The new calculation is given in (4.57).

$$\ln \overline{\varphi}_{i}(T, P, \underline{x}) = \frac{1}{b} \left(\frac{\partial Nb}{\partial N_{i}} \right)_{T, N_{j \neq i}} (Z-1) - \ln \left(Z - \frac{bP}{RT} \right) + \frac{a}{2\sqrt{2}bRT} \left[\frac{1}{Na} \left(\frac{\partial N^{2}a}{\partial N_{i}} \right)_{T, N_{j \neq i}} - \frac{1}{b} \left(\frac{\partial Nb}{\partial N_{i}} \right)_{T, N_{j \neq i}} \right] \ln \left[\frac{Z + \left(1 + \sqrt{2} \right) \frac{bP}{RT}}{Z + \left(1 - \sqrt{2} \right) \frac{bP}{RT}} \right]$$
(4.57)

This new fugacity calculation contains two partial derivatives, these are described in (4.58) and (4.59).

$$\left(\frac{\partial Nb}{\partial N_{i}}\right)_{T,N_{j\neq i}} = \frac{1}{1-D} \frac{1}{N} \left(\frac{\partial N^{2}Q}{\partial N_{i}}\right)_{T,N_{j\neq i}} - \frac{Q}{\left(1-D\right)^{2}} \left[1 - \left(\frac{\partial ND}{\partial N_{i}}\right)_{T,N_{j\neq i}}\right]$$
(4.58)

$$\frac{1}{N} \left(\frac{\partial N^2 a}{\partial N_i} \right)_{T, N_{j \neq i}} = RTD \left(\frac{\partial Nb}{\partial N_i} \right)_{T, N_{j \neq i}} + RTb \left(\frac{\partial ND}{\partial N_i} \right)_{T, N_{j \neq i}}$$
(4.59)

[B8], [A1]

Two additional partial derivatives are used in (4.58) and (4.59), these are given in (4.60) and (4.61).

$$\frac{1}{N} \left(\frac{\partial N^2 Q}{\partial N_i} \right)_{T, N_{j \neq i}} = 2 \sum_j x_j \left(b_{ij} - \frac{a_{ij}}{RT} \right)$$
(4.60)

$$\left(\frac{\partial ND}{\partial N_{i}}\right)_{T,N_{j\neq i}} = \frac{a_{i}}{b_{i}RT} + \frac{1}{C^{*}RT} \left(\frac{\partial N\underline{G}_{\gamma}^{ex}(T,\underline{x})}{\partial N_{i}}\right)_{T,N_{j\neq i}} = \frac{a_{ii}}{b_{ii}RT} + \frac{\ln\gamma_{i}}{C^{*}}$$
(4.61)

The Wong-Sandler mixing rule incorporates the Gibbs excess energy in (4.54), and the activity coefficient in the calculation of the fugacity coefficient in (4.61). Therefore the non-ideal liquid behaviour has been incorporated in the fugacity coefficient value. This means the equilibrium calculations based on (4.7), are valid for non-ideal liquids, when using the Wong-Sandler mixing rule.

Problems with the Wong-Sandler mixing rule

There are some concerns with the cross term in Wong-Sandler mixing rule

From the definition of a and b in PR we know that

$$a = 0.457235 \frac{R^2 T_c^2}{P_c} \cdot \alpha(T)$$
(4.62)

$$b = 0.077796 \frac{RT_c}{P_c}$$
(4.63)

The definition of a can therefore be rewritten to

$$a = 5.877359 \cdot bRT_c \alpha(T) \tag{4.64}$$

At temperatures below the critical temperatures the value of $\alpha(T)$ is higher than one see (4.65).

$$\alpha(T) = \left(1 + \kappa \left(1 - \sqrt{\frac{T}{T_c}}\right)\right)^2 \tag{4.65}$$

The cross term in Wong-Sandler (4.66), must therefore at temperatures below the critical temperatures be (4.67).

$$b - \frac{a}{RT} \tag{4.66}$$

$$b - \frac{5.877359 \cdot bT_c \alpha(T)}{T} = b \left(1 - \frac{5.877359 \cdot T_c \alpha(T)}{T} \right)$$
(4.67)

Because the temperature is lower than the critical temperature, the overall value of the cross term will be negative. Problems may however arise if the temperature is higher than the critical temperature for one or more components.

There are given two possible equations for the cross terms for the mixing rule (4.68) and (4.69).

$$b_{ij} - \frac{a_{ij}}{RT} = \sqrt{\left(b_{ii} - \frac{a_{ii}}{RT}\right)\left(b_{jj} - \frac{a_{jj}}{RT}\right)} \left(1 - k_{ij}\right)$$
(4.68)

$$b_{ij} - \frac{a_{ij}}{RT} = \frac{1}{2} \left(b_{ii} + b_{jj} \right) - \frac{1}{RT} \sqrt{a_{ii} a_{jj}} \left(1 - k_{ij} \right)$$
(4.69)

Unfortunately the source [B8] does not clarify that it is the negative root that is required in (4.68). (4.68) is also problematic when the temperature is noticeable higher than the critical temperature for one of the components (e.g. a gas), then the cross term becomes the square root of a negative number. Therefore (4.69) will be used in the practical application of the mixing rule in this report.

Because the cross term must be negative the value of Q (4.70) must also be negative.

$$Q = \sum_{i} \sum_{j} x_{i} x_{j} \left(b_{ij} - \frac{a_{ij}}{RT} \right)$$
(4.70)

Because Q is negative and a_m (4.71) and b_m (4.72) is positive the value of D (4.73) must be higher than one.

$$a_m = Q \frac{D}{1 - D} RT \tag{4.71}$$

$$b_m = \frac{Q}{1 - D} \tag{4.72}$$

$$D = \sum x_i \frac{a_i}{b_i RT} + \frac{\underline{G}_{\gamma}^{ex} \left(T, P, \underline{x}\right)}{C^* RT}$$
(4.73)

The value of Q and D can therefore bee used to investigate the calculations in the Wong-Sandler mixing rule.

[B8], [A1]

4.6 Part discussion/conclusion

The conclusion on the thermodynamic section is that there are several thermodynamic EOS available for process simulation. PR is good examples of EOS that gives a reliable result in process simulation of hydrocarbon systems. PRSV is designed to be more accurate than PR especially for polar components like water.

Both EOS can be used with both van der Waals one fluid mixing rule and the Wong-Sandler mixing rule. The Wong-Sandler mixing is preferred in cases where non-ideal liquid mixtures are involved as with the water/glycol mixture.

The Wong-Sandler requires input data for the Gibbs excess energy and the activity coefficient. These data can be calculated with several activity coefficient models; here the NRTL multicomponent mixture model is selected. This model can be used to calculate both the excess Gibbs energy for the mixture and the activity coefficients for the individual components, all of this from the same input data.

In this report the main concern in the simulation is the interaction between water and glycol. These two components interact by polar attraction between the two components,

allowing the glycol to absorb the water. This process is complicated to simulate with the classic thermodynamic equations like PR. To rectify the problem the Wong-Sandler mixing rule can be used.

The thermodynamic equations can not stand alone in process simulation; they must be accompanied by additional equations that describe the process equipment.

5 Simulation

The thermodynamic calculations on themselves, can only describe the thermodynamic transformations that take place inside the process elements. It is therefore necessary to have some equations that describe the different processes in the simulation. It is also necessary to know what input data the simulation requires and which results they yield.

All this will in this section be combined into a simple simulation of the water/glycol mixture. This is used to illustrate some of the problems associated to the simulation of the glycol dehydration process.

5.1 MESH elements

In process simulation, the individual components in the process plant must be described by a simulation element. The description must contain the overall information of what comes in and out of the element; this is illustrated by Figure 5-1.



The schematic in Figure 5-1 shows one possible combination of streams to and from an element, streams may be added or removed to fit to the process equipment it describes. Different elements may then be connected into the process plant that is to be described. Complex process equipment like columns is described by one MESH element per theoretically (or actual) tray, plus boiler and condenser if these elements are attached.

The overall schematic of the element itself is only a visual remainder of what comes in and out of the element. The main part of the element is the equations inside; the equations give the explanation why the elements are labelled MESH elements. MESH is an abbreviation of the four categories of calculations, they are:

- M: Material balance
- E: Equilibrium
- S: Summation
- H: Enthalpy (energy)

The equations in the MESH elements may differ depending on which variable and which that are defined. The variables in the MESH equations are:

- Pressure
- Temperature
- Flow
- Mole fraction
- Heat flow

In some elements the defined variables exclude some of the MESH equations, in these cases the elements are not actual MESH elements, but can be treated as such. An example of such elements are process units with only one phase, and thereby no equilibrium calculations. Although these elements are not actual MESH elements the overall treatment are the same with some equations excluded.

MESH calculations may be conducted for continuous or batch systems. Continuous systems are the easiest, because they can be assumed to be time independent making them steady state. Batch calculations are time dependent, thus increasing the complexity of the equations that must incorporate the time variation on the flow. Similar situations may arise for continuous systems during start up and shut down situations, or if there are fluctuation in the flow.

The different types of calculations will be more thoroughly described in the following sections

5.1.1 Material balance

The material balances ensures conservation of mass. This is important because mass can neither be destroyed nor created, only redistributed, this is described by (5.1).

Input + Production = Output + Accumulation
$$(5.1)$$

The material balance in (5.1) is general, the accumulation term only applies for time dependent processes, while the production (or consumption) term only applies in case of a chemical reaction. For a steady state, non-reaction unit the material balance becomes (5.2).

Input = Output
$$(5.2)$$

In a MESH unit there is a material balance for each chemical species. This is especially important in elements where multiple flows go in and/or out of the element.

5.1.2 Equilibrium

In multiphase systems there will be equilibrium between the individual components in the different phases. This signifies that the phase composition is dependent on the phase equilibrium of the chemical species. At equilibrium the fugacity of a component are equal for all phases as demonstrated in (5.3).

$$f_i^{I} = f_i^{II} = f_i^{III} = \dots$$
(5.3)

From the definition of the fugacity coefficient described in 4, the equilibrium constant K can be calculated from the fugacity coefficient for the different phases as demonstrated in (5.4).

$$K_i = \frac{y_i}{x_i} = \frac{\varphi_i^L}{\varphi_i^V}$$
(5.4)

The equilibrium calculation is important in dew-, bubble-point and flash calculations. In equilibrium calculations it is a requirement that there is sufficient time to reach equilibrium, if this is not the case this must be incorporated in the calculations.

Phase equilibrium calculations are conducted by thermodynamic calculations, as described in section 4. The fugacity coefficient of the different species is calculated for each phase. In columns equilibrium must be reached on all trays before an overall equilibrium is reached. The equilibrium calculations become increasingly complicated if the system contains more than two phases.

5.1.3 Summation

The summation rule is only valid if the composition is calculated as mole fraction, if the flow is defined on molar or mass basis the summation equations are excluded from the calculations. The summation equation is given in (5.5).

$$\sum_{i} x_i = 1 \tag{5.5}$$

There is a summation equation for each phase or an overall summation if the phase composition is unknown.

The summation equations are important because the composition of the different phases often change in the mass or equilibrium calculations. When the phase composition have changed, the new composition must be calculated, this is easily done when the new sum must equal one. The summation rule is also important because the subsequent calculations are based on the mole fractions, these will yield a wrong answer if the summation rule are not applied.

5.1.4 Enthalpy

The connection between energy and work are described by enthalpy (H), the definition of enthalpy are given in (5.6).

$$H = U + pV \tag{5.6}$$

Where U is the internal energy, and described by (5.7).

$$\Delta U = Q + W \tag{5.7}$$

Here Q is heat energy added and W is work done on the system. This means that the enthalpy changes if there is added or removed heat or work is done on or by the system along with changes to the pressure or volume.

Enthalpy is a state function, enthalpy differences only depend on the start and end state not the route. Because of this enthalpy calculations are conducted via the departure functions, which are defined in section 4.3.3.

The enthalpy is also used to express the energy consumed or released in a chemical reaction. Because enthalpy is a state function, the reaction energy can be calculated from the difference between reactants and products.

There are some elements where there are no enthalpy calculations, that are elements where no enthalpy changes are assumed e.g. in flash calculations at constant temperature.

5.1.5 Freedom analysis

When MESH units are calculated it is important that the all the necessary information are given or defined, this is investigated with a freedom analysis. To reach a conclusive result the number of unknowns (NU) must equal the number of equations (NE). A free-

dom analysis is therefore used to find the number of equations and unknown, with three possible outcomes:

- 1. NE < NU => Infinite solutions
- 2. $NE = NU \Longrightarrow$ One solution
- 3. NE > NU => No solution

In the first case the system is under defined, to solve this some of the unknown variables must be defined, until case two is reached. In case three it is necessary to make some of the known variables unknown. The problem can also be solved by adjusting the number of equations; this is especially in case three where the answer of an equation may already have been defined.

5.2 Flash separation

Flash separation is used to calculate the phase-composition for a multiphase system where only the overall component composition is known. Besides the thermodynamic equations used to calculate the phase behaviour of the components, there are two algorithms that are used to calculate the component distribution in the phases.

The distribution of the components in the different phases is calculated using the Rachford-Rice algorithm for two-phase systems, while three-phase systems are calculated with the Henley-Rosen algorithm.

5.2.1 Rachford-Rice

The Rachford-Rice algorithm are used to solve two-phase flash separations, this is usually vapour/liquid separation, although it is equally useful for liquid/liquid separations. Here the classic vapour/liquid form of the algorithm will be demonstrated, for liquid/liquid calculations the vapour phase is replaced by the additional liquid phase.

In a two-phase system the overall mass balance is given by (5.8).

$$F = V + L \tag{5.8}$$

Where F is the total flow into the system, V is the vapour flow and L is the liquid flow out of the system. For the individual components the mass balance is given by (5.9).

$$F \cdot z_i = V \cdot y_i + L \cdot x_i \tag{5.9}$$

The vapour fraction ψ must be between zero and one and is given by (5.10)

$$\psi = \frac{V}{F} \tag{5.10}$$

The total size of the two phases is given by (5.11) and (5.12).

$$V = \psi \cdot F \tag{5.11}$$

$$L = (1 - \psi) \cdot F \tag{5.12}$$

At equilibrium the relationship between the components in the two phases are given by the equilibrium constant K as described in (5.4), that here have been rearranged into (5.13).

$$y_i = K_i \cdot x_i \tag{5.13}$$

By inserting (5.11), (5.12) and (5.13) into (5.9) and rearranging, (5.14) is reached.

$$x_{i} = \frac{z_{i}}{1 + \psi(K_{i} - 1)}$$
(5.14)

The Rachford-Rice equation is based on the summation rule (5.5). Rachford-Rice combined the summation for the two phases into

$$\sum_{i} x_{i} - \sum_{i} y_{i} = 0 \tag{5.15}$$

By inserting (5.13) and (5.14) into (5.15) the Rachford-Rice equation can be reached, this is (5.16)

$$\sum_{i} \frac{z_i \left(1 - K_i\right)}{1 + \psi \left(K_i - 1\right)} = 0$$
(5.16)

The Rachford-Rice equation is used to calculate ψ ; this can be done in an iterative process e.g. using the Newton-Raphson method.

Rachford-Rice algorithm

The Rachford-Rice algorithm is a two-phase flash algorithm where the Rachford-Rice equation is used to solve the vapour/liquid split. In the Rachford-Rice algorithm the composition into the system must be given along with some initial estimates for K_i and ψ . These initial estimates are then update in the algorithm in an iterative process, the algorithm is illustrated in Figure 5-2.



Input data for the system and guessed values for K

The solution is printed

Figure 5-2: The Rachford-Rice algorithm

The initial estimates of the equilibrium constant can be guessed values updated with the result of the program when it have been solved.

[B10]

5.2.2 Henley-Rosen

The Henley-Rosen algorithm is similar to the Rachford-Rice algorithm, with the difference that Henley-Rosen solves three-phase flash separations. In three-phase flash the overall mass balance is given by (5.17)

$$F = V + L_1 + L_2 \tag{5.17}$$

Because of the additional phase the liquid/liquid split ξ is introduced alongside the vapour fraction ψ . They are defined by (5.18) and (5.19).

$$\psi = \frac{V}{F} \tag{5.18}$$

$$\xi = \frac{L_1}{L_1 + L_2} \tag{5.19}$$

There is also an additional equilibrium constant (5.20) and (5.21).

$$K_i^1 = \frac{y_i}{x_i^1}$$
(5.20)

$$K_i^2 = \frac{y_i}{x_i^2}$$
(5.21)

The composition of the three phases can be calculated by (5.22), (5.23) and (5.24).

$$y_{i} = \frac{z_{i}}{\frac{\xi(1-\psi)}{K_{i}^{1}} + \frac{(1-\psi)(1-\xi)}{K_{i}^{2}} + \psi}$$
(5.22)

$$x_{i}^{1} = \frac{z_{i}}{\xi(1-\psi) + (1-\psi)(1-\xi)\left(\frac{K_{i}^{1}}{K_{i}^{2}}\right) + \psi K_{i}^{1}}$$
(5.23)

$$x_{i}^{2} = \frac{z_{i}}{\xi (1 - \psi) \left(\frac{K_{i}^{2}}{K_{i}^{1}}\right) + (1 - \psi) (1 - \xi) + \psi K_{i}^{2}}$$
(5.24)

Instead of solving (5.22), (5.23) and (5.24), two of the equations can be replaced by (5.20) and (5.21).

Because there are two unknown ψ and ξ there are two Henley-Rosen equations (5.25) and (5.26).

$$\sum_{i} \frac{z_{i} \left(1 - K_{i}^{1}\right)}{\xi \left(1 - \psi\right) + \left(1 - \psi\right) \left(1 - \xi\right) \frac{K_{i}^{1}}{K_{i}^{2}} + \psi K_{i}^{1}} = 0$$
(5.25)

$$\sum_{i} \frac{z_{i} \left(1 - \frac{K_{i}^{1}}{K_{i}^{2}}\right)}{\xi (1 - \psi) + (1 - \psi) (1 - \xi) \frac{K_{i}^{1}}{K_{i}^{2}} + \psi K_{i}^{1}} = 0$$
(5.26)

The two Henley-Rosen equations must be solved simultaneously by an appropriately method like the Newton-Raphson for two unknowns.

Henley-Rosen algorithm

The Henley-Rosen algorithm are similar to the Rachford-Rice algorithm, the only difference is that initial values must be given for K_i^1 , K_i^2 , ψ and ξ . In the algorithm ψ and ξ are updated simultaneously as is K_{i}^{1} and K_{i}^{2} . Otherwise the algorithm is identical to Figure 5-2.

[B10]

5.3 Simulation model

In nature water and glycol will mix and create a single water/glycol phase, due to the polar attraction between the components. This mixture is complicated to simulate because of the polar interaction between water and glycol. The purpose of the simulations is to investigate the thermodynamic equations ability to simulate the water/glycol mixture correctly.

In case the thermodynamic equations can not reproduce the mixture, the result will be a two liquid phase solution. Therefore the calculations are conducted as flash calculations, using the Rachford-Rice algorithm. The algorithm requires two phases at all times, or else it collapses. To insure that two liquid phases are available at all times, the components includes decane in addition to water and glycol. Decane creates a stable liquid phase because of its low tendency to evaporate.

The water/glycol mixture is also simulated in a three-phase system using the Henley Rosen algorithm. The three-phase flash calculations are conducted to evaluate the components tendency to evaporate, and its influence on the two-phase calculations. In the three-phase calculations methane is added to the system to create a permanent gas-phase.

Besides investigating the water/glycol equilibrium in two- and three-phase systems the equilibrium is investigated with three different EOS. This is PR, PRSV and PR with the Wong-Sandler mixing rule (PR-WS). This creates a total of six different cases; these are described in Table 5-1.

Table 5-1: Cases					
EOS	Two-phase flash	Three-phase flash			
PR	Case 1	Case 4			
PRSV	Case 2	Case 5			
PR-WS	Case 3	Case 6			

The investigations in the six cases are conducted at the same operation conditions, and for the same components, according to the required phases.

5.3.1 Input data

The required input data for the system is limited to the operating conditions for the separation, composition of the flow into the system and the thermodynamic data for the components along with initial estimates for the equilibrium constants.

Operating conditions

The required operating conditions for the system are limited to the temperature and pressure:

- Temperature: 25 °C
- Pressure: 0 barg

Composition

The components used differ between the two- and three-phase calculations. The twophase system consists of water, glycol and decane. Because of the additional phase in the three-phase calculations methane is added to create the gas phase.

With the components determined, the next step is to determine the composition of the components in the inflow. Because water and glycol are the main components, these are determined first and the rest are added to reach a total flow of one mole into the system.

In the glycol dehydration plant described in section 3.3 the flow of TEG is between 0.017 and 0.042 m³ Lean TEG/kg water in the gas (2 to 5 gal/lb), depending on the size and efficiency of the column. In this report a flow of 0.025 m³ lean TEG/kg water will be used. This value must be converted to a molar ratio before it can be used in the MATLAB program; the data for the calculations is given in Table 3-1.

$$0.025 \frac{m^{3}TEG}{kg H_{2}O} \cdot 1122 \frac{kg TEG}{m^{3}TEG} = 28.05 \frac{kg TEG}{kg H_{2}O}$$
$$28.05 \frac{g TEG}{g H_{2}O} \cdot \frac{18.015 \frac{g H_{2}O}{mol H_{2}O}}{150.17 \frac{g TEG}{mol TEG}} = 3.365 \frac{mol TEG}{mol H_{2}O}$$

The ratio is only valid at 25 °C because this is the temperature for which the density is given. The ratio used in the MATLAB program is 1:3.4.

With the water/glycol ratio determined the composition of the flow into the system is given in Table 5-2.

Table 5-2: Flow into the simulations						
Component	Two-phase flash	Three-phase flash				
	[mol]	[mol]				
Water	0.10	0.10				
TEGlycol	0.34	0.34				
Methane	0.00	0.20				
Decane	0.56	0.36				

Table 5-2: Flow into the simulations

From Table 5-2 it can be seen that the flow into the system is one mole

Thermodynamic data

There are some problems with the thermodynamic data for TEG because the critical point can not be measured [A2]. This means that these values may not be available in all sources and that they may differ more than usual between the sources.

In this report the source of all the thermodynamic data have been the thermodynamic database in HYSYS. HYSYS is a thermodynamic process simulation program; as such its database is very extensive and gives the thermodynamic data for a wide range of components. The database is so extensive that it also contains data for the more specific correction parameters like κ_1 in PRSV; it also contains values for NRTL calculations of the excess Gibbs energy and activity constant. The constants in HYSYS do unfortunately not include the binary interaction parameter k_{ij} , which can have significant influ-

ence on the final outcome. Because the source of the thermodynamic data is HYSYS the values used will be given in App. 1.

The thermodynamic data given in the appendix also include the data and results used in the NRTL calculations of the excess energy and activity coefficients. The NRTL calculations are done with a MATLAB program available on the attached CD in the folder MATLABACTIVITY.

The thermodynamic data does not include the equilibrium constant K, these values have been estimated for case 1 and 4. The estimates have been corrected to the final value calculated in the cases. The initial estimate for remaining cases has been the final values from case 1 and 4. The values for the phase splits ψ and ξ are treated in the same manner.

Because the equilibrium constants and phase splits are estimated values, the final values are given in the result section.

5.3.2 The MATLAB program

The program code for case 1 will be described in this section. The program has been written using the algorithms from section 5.2 and the thermodynamic equations in section 4. The special code lines that differ between case 1 and cases 2, 3 and 4 will also be described. The modifications between case 4 and case 5 and 6, equals the differences between case 1 and the cases 2 and 3. The program code for case 1 and 4 is given in App. 2; all the program codes are available on the CD in the folder \MATLAB\.

The settings in the programs have been set in such a way that the results given in section 5.4 can be reproduced. This is especially important in these cases where the program does not reach a satisfying result.

For the description of the program it has been divided into smaller peaces. First the twophase PR program itself is explained and then the modifications made for PRSV, Wong-Sandler mixing rule and the three-phase algorithm.

Input data

The first part of the program is the global variables. It is actually not necessary to make these variables global in this case because the MATLAB program is limited to one script. The global variable is only necessary when several scripts share the same variables. This is shown in Figure 5-3.

Besides the global variables Figure 5-3 also contains the input data for one component. The code for the remaining components are similar to the component code in Figure 5-3, the only different are the component specific values.

```
Global Variables
global C;
                   %Number of components
C = 3;
global R;
                  %The Gas Constant
                  %[J/(mol*K)]
R=8.314;
global pc;
global Tc;
                  %Critical pressure [pa]
                 %Critical temperature [K]
global omega;
                  %Acentric factor
       Critical data for components
%____
%Data is from HYSYS critical component data database - K-values are
quessed
for c=1
               %Water
    Tc(c)=647.25;
                        %Critical Temperature [K]
   pc(c)=22.12e6;
omega(c)=0.344;
                       %Critical Pressure [Pa]
                      %Acentric factor
    K(c)=1.439e-2;
                        %Water in water
End
```

Figure 5-3: The MATLAB code for global variables and critical data.

The data required to solve the program also consists of the pressure and temperature for the separation and the composition of the flow into the system. The code for this is given in Figure 5-4.

```
Input data
                       _ _ _ _
Tflash=25;
                                %Temp in [C]
T=Tflash+273.15;
                                %Temp in [K]
pflash=0.0;
                                %Pressure in [barg]
p=(1+pflash)*1e5;
                                %Pressure in [Pa]
%Composition [mol]
z(1)=0.10; %Water
z(2)=0.34;
            %TEG
           %Decane
z(3)=0.56;
%Normalizing for z-values
zsum=0;
for c=1:C
    zsum=zsum+z(c);
end
for c=1:C
    z(c)=z(c)/zsum;
end
%Start guess for liquid-liquid split
psi=0.913;
                   %Start guess for LL split (L1/(L1+L2))
```

Figure 5-4: Input data for the MATLAB program.

Because the calculations in the MATLAB program are done on mole fraction basis, the sum of the flow into the system must equal one. To achieve this, z-values are normalized. The code in Figure 5-4 also contains an initial estimate for the liquid-liquid split. This is required before the value can be updated in the Rachford-Rice equation.

Phase split

The first step in the algorithm is to update the phase split ψ , this is done with the Rachford-Rice equation using the Newton-Raphson method. This is shown in Figure 5-5.

```
Update of psi
error=1;
mb=0;
MB=100;
while error > 1e-6
   %Sum of f(1) og fdot(1)
   f=0;
   fdot=0;
    for c=1:C
        f=f+z(c)*(1-K(c))/(1+psi*(K(c)-1));
        fdot=fdot+z(c)*(K(c)-1)^2/(psi*(K(c)-1)+1)^2;
    end
   psi=psi-f/fdot;
    if psi<0 || psi>1
        disp('New initial value for psi')
        fprintf('psi %.4f\n',psi)
        return
    end
    error = abs(f);
   mb=mb+1;
    if mb>=MB
       disp('max no. of iterations in loop b')
        return
    end
end
```

Figure 5-5: Calculation of the phase split.

There are two stop function built into the phase split calculations. The first function stops the calculations if the ψ -value comes outside the interval between zero and one. This function is added to stop the program in case of a result that is not physically possible.

Because the calculation is done in a **while** loop, it will continue until a result is reached. The second stop function has therefore been inserted to stop the calculation after 100 iterations. Without this second stop function the loop would continue indefinitely if no result could be reached.

Phase composition

The phase composition is calculated from the flow into the system with the new phase split. Besides the phase split, the phase composition calculations also requires the value of the equilibrium constant K. The calculations are shown in Figure 5-6.

```
Phase composition ----
C
mposition of water
```

```
for c=1:C
    %Composition of water
    x(1,c)=z(c)/(1+psi*(K(c)-1));
    %Composition of oil
    x(2,c)=K(c)*x(1,c);
end
```

Figure 5-6: Calculation of the phase composition.

When the composition of the phases is known, the fugacity coefficient of the individual components in the different phases can be calculated.

Fugacity calculations

The fugacity calculations are conducted with PR, the first step is to calculate a and b for the individual components, this is illustrated in Figure 5-7.

```
%---- a and b in Peng-Robinson ----
for c=1:C
    kappa=0.37464+1.54226*omega(c)-0.26992*omega(c)^2;
    alfa=(1+kappa*(1-sqrt(T/Tc(c))))^2;
    a(c,c)=0.45724*R^2*Tc(c)^2/pc(c)*alfa;
    b(c)=0.07780*R*Tc(c)/pc(c);
end
for c=2:C
    for n=1:(c-1)
        a(c,n)=sqrt(a(c,c)*a(n,n));
        a(n,c)=a(c,n);
    end
end
```

Figure 5-7: Calculation of a and b for PR.

Besides calculating the a-values for the pure components, it is also necessary to calculate the a-values for the interaction between the components.

The a- and b-values are necessary for the calculation of a_m , b_m , A and B. These calculations are conducted for each phases as shown in Figure 5-8.

```
%---- Fugacity coefficients in water ---
%am and bm in water
am=0;
bm=0;
for c=1:C
    for n=1:C
        am=am+x(1,c)*x(1,n)*a(c,n);
    end
        bm=bm+x(1,c)*b(c);
end
%A and B for cubic equation
A=am*p/(R*T)^2;
B=bm*p/(R*T);
```

Figure 5-8: Calculation of a_m, b_m, A and B.

When the A and B values for the phase have been calculated, they are inserted into the cubic PR, so this can be solved. This is shown in Figure 5-9.

```
%Peng-Robinson cubic EOS
Z1=B;
error=1;
mc=0;
MC=100;
while error > 1e-6
    q=Z1^3+(-1+B)*Z1^2+(A-3*B^2-2*B)*Z1+(-A*B+B^2+B^3);
    gdot=3*Z1^2+2*(-1+B)*Z1+(A-3*B^2-2*B);
    Z1=Z1-g/gdot;
    error=abs(g);
    mc=mc+1;
    if mc>=MC
        disp('max no. of iterations in loop c')
        return
    end
end
```

Figure 5-9: Calculation of Z with PR.

Because the calculation in Figure 5-9 is for a liquid phase the initial guess for Z is equal to the B-value, for a gas phase the initial estimate for Z is one. Because the cubic EOS is solved with the Newton-Raphson method, it is inside a **while** loop, therefore the code include a stop function.

The Z-value for the phase is used for the fugacity calculations in Figure 5-10.

```
%Fugacity coefficients
for c=1:C
    xa=0;
    for n=1:C
        xa=xa+x(1,n)*a(c,n);
    end
    phi(1,c)=exp(b(c)/bm*(Z1-1)-log(Z1-bm*p/(R*T))-...
    am/(2*sqrt(2)*bm*R*T)*(2*xa/am-b(c)/bm)*...
    log((Z1+(1+sqrt(2))*bm*p/(R*T))/(Z1+(1-sqrt(2))*bm*p/(R*T))));
end
```

Figure 5-10: Calculation of the fugacity coefficient.

When the fugacity coefficient of all components in all phases is determined, the equilibrium constants for the components can be updated.

Update of the K-values

The loop to update the K-values is the main algorithm that runs the program. Besides the code for the input data, all the remaining calculations described in this section is inside this loop. The K-values are also used to determine if the program has reached a solution or if the calculations must continue. The loop and the calculation of the new K-values are described in Figure 5-11.

```
Calculations
%Loop to update the K values
flag=1;
ma=0;
MA=100; %Max no. of iterations in loop a
while flag
           Update of psi
    8----
                            ____
    %---- Phase composition
                               ____
    %---- a and b in Peng-Robinson
                                       ____
    %---- Fugacity coefficients in water
                                             ____
    %---- Fugacity coefficients in oil ----
    %---- Update K and check for convergence
                                                 ____
    flag=0;
    for c=1:C
        if abs(K(c) - phi(1,c)/phi(2,c)) > K(c)*1e-3+1e-6
            flag=1;
            K(c) = phi(1,c)/phi(2,c);
        else
            K(c) = phi(1,c)/phi(2,c);
        end
    end
    ma=ma+1;
    if ma>=MA
       disp('max no. of iterations for loop a')
        return
    end
end
```

Figure 5-11: Program loop and updating of the K-values.

The program is calculated inside a **while** loop, like the previous **while** loops it has a stop function. Unlike the previous **while** loops, that where controlled by the error function, this loop is controlled by the flag function.

The previous loops have used the Newton-Raphson method to reach zero, this result was then used to determine the error in the calculation. The loop for the K-values is controlled by the flag function that is determined by the difference in the K-values between the iterations.

Before the new K-values are calculated the flag constant is set equal to zero. If one or more of the new K-values differs more than accepted from the old K-value the flag constant is set equal to one again, and the **while** loops continues the iterations.

When the difference between the old and new K-values for all components are within the accepted limit, the separation has been solved and the results can be printed. The print commands are not given here, but the results given include the operation conditions, phase compositions and equilibrium constants. The print codes can be viewed alongside the entire code for case 1 in App. 2.

PRSV modification

When the program is calculated with PRSV, the described program must be modified slightly. In the critical data for the components the value for κ_1 must be added. The other difference is the calculation of the a-values for the pure components, the new calculations are shown in Figure 5-12.

```
%---- a and b in PRSV ----
for c=1:C
    Tr=T/Tc(c);
    kappa0=0.378893+1.4897153*omega(c)+...
        0.17131848*omega(c)^2+0.0196554*omega(c)^3;
    kappa=kappa0+kappa1(c)*(1+sqrt(Tr))*(0.7-Tr);
    alfa=(1+kappa*(1-sqrt(T/Tc(c))))^2;
    a(c,c)=0.45724*R^2*Tc(c)^2/pc(c)*alfa;
    b(c)=0.07780*R*Tc(c)/pc(c);
end
```

Figure 5-12: Calculation of a and b with PRSV.

In this program the only difference between the PR and PRSV calculations is the calculation of a. If departure calculations were a part of the program, these would also be modified because of the derivative da/dT.

Wong-Sandler modification

The Wong-Sandler mixing rule can be used instead of van der Waals one fluid mixing rule for different EOS; here it is combined with PR. The first difference when Wong-Sandler is introduced is in the calculation of the interaction between the components this is shown in Figure 5-13.

```
%---- a and b in Peng-Robinson ----
for c=1:C
    kappa=0.37464+1.54226*omega(c)-0.26992*omega(c)^2;
    alfa=(1+kappa*(1-sqrt(T/Tc(c))))^2;
    a(c,c)=0.45724*R^2*Tc(c)^2/pc(c)*alfa;
    b(c,c)=0.07780*R*Tc(c)/pc(c);
end
for c=1:C
    for n=1:C
        cross(c,n)=1/2*(b(c,c)+b(n,n))-sqrt(a(c,c)*a(n,n))/(R*T);
    end
end
```

Figure 5-13: Calculation of a, b and the cross term for Wong-Sandler.

The new cross term in Wong-Sandler also changes the calculation of a_m and b_m , the new calculations is conducted via two intermediate values Q and D. These new calculations are given in Figure 5-14.

Figure 5-14: Calculation of Q, D, a_m and b_m with Wong-Sandler.

The introduction of the Wong-Sandler mixing rule also changes the calculation of the fugacity coefficient, as shown in Figure 5-15.

```
%Fugacity coefficients
for c=1:C
    fug(4,c)=(a(c,c)/(b(c,c)*R*T))+(log(gamma(c))/Cstar);
    fusum=0;
    for n=1:C
        fusum=fusum+x(1,n)*cross(c,n);
end
    fug(3,c)=2*fusum;
    fug(1,c)=1/(1-D)*fug(3,c)-Q/(1-D)*2*(1-fug(4,c));
    fug(2,c)=R*T*D*fug(1,c)*(Z-1)-log(Z-B)+...
        am/(2*sqrt(2)*bm*R*T)*(1/am*fug(2,c)-1/bm*fug(1,c))*...
        log((Z+(1+sqrt(2))*B)/(Z+(1-sqrt(2))*B)));
end
```

Figure 5-15: Calculation of fugacity coefficient in Wong-Sandler.

The new fugacity coefficient calculation for Wong-Sandler depends on four derivatives, here named fug(1,c) to fug(4,c).

Three-phase modification

The main modification between the two- and three-phase programs is the calculation of the phase splits ψ and ξ . Because the three-phase program contains an additional phase, an additional K-value is required for each component in critical data.

The three-phase splits are also calculated with the Newton-Raphson method, this time the variant for two equations and two unknown. This gives two Newton-Raphson equations that depend on two equations and four derivatives, compared to one equation and one derivative for one unknown.

The phase split calculations for the three-phase program is given in Figure 5-16.

```
Calculation of psi and xi
error=1;
mb=0;
MB=100;
while error > 1e-6 && mb<MB
    %Sum of f(1)
    f(1)=0;
    for c=1:C
        f(1)=f(1)+z(c)*(1-K(1,c))/...
             (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+psi*K(1,c));
    end
    %Sum of f(2)
    %sum of fdot(1,1), fdot(1,1)=d(f(1))/d(psi)
    fdot(1,1)=0;
    for c=1:C
        fdot(1,1)=fdot(1,1)-z(c)*(1-K(1,c))*...
             (-xi-(1-xi)*K(1,c)/K(2,c)+K(1,c))/.
             (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+...
            psi*K(1,c))^2;
    end
    sum of fdot(1,2), fdot(1,2)=d(f(1))/d(xi)
    sum of fdot(2,1), fdot(2,1)=d(f(2))/d(psi)
    %sum of fdot(2,2), fdot(2,2)=d(f(2))/d(xi)
    %Newton Raphson equation for two equations and two unknown
    psi=psi-(f(1)*fdot(2,2)-f(2)*fdot(1,2))/...
        (fdot(1,1)*fdot(2,2)-fdot(1,2)*fdot(2,1));
    xi=xi-(fdot(1,1)*f(2)-fdot(2,1)*f(1))/...
        (fdot(1,1)*fdot(2,2)-fdot(1,2)*fdot(2,1));
    if psi<0 || psi >1 || xi<0 || xi>1
        disp('new initial values for xi or psi')
        fprintf('Vapor-Liquid split (psi) %.4f\n',psi);
fprintf('Oil-Water split (xi) %.4f\n',xi);
        return
    end
    error=abs(f(1))+abs(f(2));
    mb=mb+1;
    if mb>=MB
        disp('max no. iterations for loop b')
        %return
    end
end
```

Figure 5-16: Calculation of the phase split for three-phases.

Besides the changed phase split calculations in the three-phase algorithm, the phase composition and K-value calculations have been changed.

The result of the six cases is given in the following sections. The result contains the phase composition, both as molar percent and molar flow, along with the total size of the phases. The results do also contain the equilibrium constants and phase splits.

The results are commented and compared to the previous cases.

5.4.1 Case 1

In case 1 PR is used together with the Rachford-Rice algorithm, the system consists of three components, water, TEG and decane. The results of case 1 are given in Table 5-3.

Table 5-3: Phase composition in case 1							
	Water phase		Oil ph	ase	K		
	[mol %]	[mol %] [mol] [mol %] [mol]					
Water	100.00	0.087	1.44	0.013	$1.44 \cdot 10^{-2}$		
Glycol	0.00	0.000	37.23	0.340	$4.99 \cdot 10^{39}$		
Decane	0.00	0.000	61.33	0.560	$1.28 \cdot 10^{21}$		
Total	100.00	0.087	100.00	0.913	-		

• Liquid/liquid split (ψ): 0.913

As it can be seen in Table 5-3, PR is unable to simulate the water/glycol mixture. The water-phase only consists of water. From the K-values it can be seen that glycol are even more unwilling to mix with the water than decane.

The 1.44 % of water in the oil phase is approximately the same amount of water as in an oil-phase consisting solely of decane. This indicates that the glycol has no influence whatsoever in this case.

5.4.2 Case 2

In case 2 PRSV is used together with the Rachford-Rice algorithm, the system consists of three components, water, TEG and decane. The results of case 2 are given in Table 5-4.

Table 5-4: Phase composition in case 2							
	Water p	ohase	Oil ph	K			
	[mol %]	[mol %] [mol] [mol %] [mol]					
Water	100.00	0.083	1.90	0.017	$1.89 \cdot 10^{-2}$		
Glycol	0.00	0.000	37.06	0.340	$9.40 \cdot 10^{34}$		
Decane	0.00	0.000	61.04	0.560	$4.97 \cdot 10^{20}$		
Total	100.00	0.083	100.00	0.917	-		

• Liquid/liquid split (ψ): 0.917

As it can bee seen in Table 5-4, the results in case 2 is almost identical to case 1, indicating that PRSV is as insufficient as PR to simulate the water/glycol mixture.

There are small differences between case 1 and 2; there are a littler more water in the oil-phase and the K-values for glycol and decane are smaller than in case 1. These differences are however unable to change the overall result, that PRSV also yields an unsatisfactory result.

5.4.3 Case 3

In case 3 PR-WS is used together with the Rachford-Rice algorithm, the system consists of three components, water, TEG and decane.

The idea with using the Wong-Sandler mixing rule was to reach a more accurate result for the water/glycol mixture. Unfortunately this program was unable to reach a result, therefore this case have been subdivided into 6 subcases used to investigate the problem with this code.

Case 3a

This case is the origin case, all future subcases for case 3 are modifications based on this case. The first time this case was calculated the value for ψ was outside the boundaries from zero to one after the first iteration. The second parameter that seamed out of bound was the equilibrium constant K, here the values was close to 1, compared to the previous cases. The ψ and K values as a function of iteration are given in Table 5-5.

Table 5-5: ψ and K-values in Case 3a						
Iteration	Ψ	K(Water)	K(Glycol)	K(Decane)		
Start	0.917	$1.89 \cdot 10^{-2}$	$9.40 \cdot 10^{34}$	$4.97 \cdot 10^{20}$		
1	0.917	$2.18 \cdot 10^{0}$	$7.76 \cdot 10^{-3}$	$1.07 \cdot 10^{-1}$		
2	-0.655	-	-	-		

From Table 5-5 it can be seen that the K-values changes drastically, especially for glycol and decane. The theory behind the Rachford-Rice algorithm dictates that the Kvalue for some components must be higher than one, while others must be lower than one. The size of the K-value compared to one indicates which phase the components primarily will be in. After the first iteration all the K-values have changed their K-value compared to one, indicating that the phases change. In case 1 and 2 water was the first phase, here it seems the algorithm tries to make it the second phase.

Case 3b

The problem in case 3a is the calculation of the K-values, these influences the calculation of ψ , making the program collapse. Because ψ is calculated before the K-values, the initial estimate of the K-values influences the first iteration of the program. Case 3b is used to evaluate if the problem with the algorithm arises because the initial estimated of the K-values are to fare from the final estimates. The results from case 3b are given in Table 5-6.

Table 5-6: ψ and K-values in Case 3b							
Iteration	Ψ	K(Water)	K(Glycol)	K(Decane)			
Start	0.500	$1.89 \cdot 10^{-2}$	$9.40 \cdot 10^3$	$4.97 \cdot 10^2$			
1	0.917	$2.18 \cdot 10^{0}$	$7.90 \cdot 10^{-3}$	$1.08 \cdot 10^{-1}$			
2	-0.655	-	-	-			

The results of case 3b are almost identical to the results of case 3a, even though the initial estimate of the K-value for glycol and decane have been made significantly smaller, the result after one iteration is almost identical. This indicates that the problem is not the initial estimates of the K-value.

Case 3c

As indicated in the results from case 3a and b the algorithm tries to change the phases. Too investigate if this is the case, the initial estimates for the K-values have been changed compared to 1, this is illustrated in Table 5-7.

Table 5-7: ψ and K-values in Case 3c						
Iteration	Ψ	K(Water)	K(Glycol)	K(Decane)		
Start	0.083	$1.89 \cdot 10^2$	$9.40 \cdot 10^{-3}$	$4.97 \cdot 10^{-2}$		
1	0.099	$4.54 \cdot 10^{-1}$	$8.07 \cdot 10^{1}$	$7.24 \cdot 10^{0}$		
2	3036900.449	-	-	-		

In case 3c, the result is similar to case 3a and b namely that the components try to exchange phase. Like in case 3a the values of K becomes very close to 1, which indicates that there are no distinct phase separation. Because of the very equal K-values the ψ -value rises towards infinite.

Case 3d

The problem might originate in the introduction of the excess Gibbs energy and activity coefficient in the Wong-Sandler equations. This is combined with the properties of glycol, which might create problems in the simulation. To investigate if the problem is one of these factors, the system has in case 3d been simplified to water and decane. The excess Gibbs energy is set to 0 and the activity coefficient of the components are 1. The flow into the system is equal amounts of water and decane. This gives a simple simulation the results are given in Table 5-8.

Table 5-8: ψ and K-values in Case 3d						
Iteration	Ψ	K(Water)	K(Decane)			
Start	0.500	$1.89 \cdot 10^{-2}$	$4.97 \cdot 10^2$			
1	0.509	$2.37 \cdot 10^{0}$	$1.53 \cdot 10^{-1}$			
2	0.224	$6.18 \cdot 10^{-1}$	$3.28 \cdot 10^{0}$			
3	1.089	-	-			

In this case the program collapsed in the third iteration, the size of the K-values compared to one changes in the iterations. This indicates that the Wong-Sandler equations try to create one liquid phase, thus working against the two-phase algorithm.

Case 3e

The problems in case 3a to 3d spark the question where the problem arises in the calculations, therefore the different calculations are compared between PR (case 1) and the PR-WS calculations in case 3a. Both programs are run one iteration and the following values are compared: a_m , b_m , and ϕ for both phases and the overall K-values. The result from case 3e is given in Table 5-9.

	Table 5-9. LOS uata in Case Se					
			PR	PR-WS		
	a _m		14.717	14.609		
1	\mathbf{b}_{m}		$2.40 \cdot 10^{-4}$	$2.36 \cdot 10^{-4}$		
asi		Water	$2.69 \cdot 10^{-2}$	$4.29 \cdot 10^8$		
ЧЧ	φ	Glycol	$2.24 \cdot 10^{34}$	$1.99 \cdot 10^{10}$		
		Decane	$2.72 \cdot 10^{18}$	$1.14 \cdot 10^8$		
	a _m		0.987	0.982		
e 2	b_{m}		$1.89 \cdot 10^{-5}$	$1.89 \cdot 10^{-5}$		
asi		Water	$1.87 \cdot 10^{0}$	$9.35 \cdot 10^8$		
ЧЧ	φ	Glycol	$4.50 \cdot 10^{-6}$	$1.57 \cdot 10^{8}$		
		Decane	$2.13 \cdot 10^{-3}$	$1.23 \cdot 10^7$		
		Water	$1.44 \cdot 10^{-2}$	$2.18 \cdot 10^{0}$		
K		Glycol	$4.99 \cdot 10^{39}$	$7.90 \cdot 10^{-3}$		
_		Decane	$1.28 \cdot 10^{21}$	$1.08 \cdot 10^{-1}$		

Table	5-9:	EOS	data	in	Case	3

When the parameters from case 3a are compared to case 1 it is clear that the problem is in the fugacity calculations. In PR there is a noticeable difference in value of the fugacity coefficient between the two phases, in PR-WS however the value for the two phases is almost identical. The identical fugacity coefficients may indicate that the difference between the two phases is minimal, and the program is trying to create one phase.

Case 3f

In the previous subcases there have been indications that the PR-WS equations is trying to create only one liquid phase, thereby clashing with the purpose of the Rachford-Rice algorithm. Therefore the next logical step is to add methane, thus creating a distinct vapour phase; this will allow a single liquid phase to be created. To minimize the possibility of errors the excess energy is zero and the activity coefficient is one for all components. The components are added in equal amounts. The results from case are given in Table 5-10.

Table 5-10: ψ and K-values in Case 3f							
Iteration	Ψ	K(Water)	K(Decane)	K(Methane)			
Start	0.660	$1.89 \cdot 10^{-2}$	$4.97 \cdot 10^2$	$1.90 \cdot 10^2$			
1	0.678	$5.07 \cdot 10^{8}$	$1.18 \cdot 10^{7}$	$2.85 \cdot 10^4$			
2	2845130.341	-	-	-			

With methane added to the system the Rachford-Rice algorithm is allowed to create one single liquid phase. Unfortunately this does not happen, instead all K-values are distinctly larger than one, this collapses the ψ calculations, making the value of ψ go towards infinite. That all the values of K are larger than one indicates that the equations still tries to create one single phase, which now includes methane.

Case 3 conclusion

The problem with the PR-WS seems to be in the fugacity calculations. There is not a clear difference in the fugacity between the two phases. In case 3f, where methane is added to the system there are a little difference in the K-values, but this difference is insignificant compared to the fact that all K- values are higher than 1. This means that all three components move towards the same phase.

In [A1], [A3], [A4] and [B8] the Wong-Sandler equations are only used for dew- and bubble-point pressure calculations. Common for the described experiments are that they use the Wong-Sandler equation to calculate phase boundaries, and not the separation in the two-phase region.

The problem is either in the Wong-Sandler equations or in the combination of these equations with the Rachford-Rice algorithm.

5.4.4 Case 4

In case 4 PR is used together with the Henley-Rosen algorithm for a four component system consisting of water, TEG, methane and decane. The results of case 4 are given in Table 5-11.

	Table 5-11: Phase composition in case 4							
	Gas pł	nase	Oil ph	ase	Water phase		K1	K2
	[mol %]	[mol]	[mol %]	[mol]	[mol %]	[mol]		
Water	2.71	0.005	1.40	0.010	100.00	0.085	$1.93 \cdot 10^{0}$	$2.71 \cdot 10^{-2}$
Glycol	0.00	0.000	47.61	0.340	0.00	0.000	$4.75 \cdot 10^{-6}$	$2.17 \cdot 10^{34}$
Methane	97.18	0.196	0.60	0.004	0.00	0.000	$1.63 \cdot 10^2$	$1.03 \cdot 10^{6}$
Decane	0.11	0.000	50.38	0.360	0.00	0.000	$2.22 \cdot 10^{-3}$	$2.817 \cdot 10^{18}$
Total	100.00	0.201	100.00	0.714	100.00	0.085	-	-

- Gas/liquid split (ψ): 0.201
- Liquid/liquid split (ξ): 0.894

The result of the three-phase separation using PR is very similar to the result in the twophase separation; there are some small differences due to the additional phase. The additional phase influences the K2 value a little compared to the K value in case 1. The value for decane and glycol a slightly smaller, indicating that the gas phase have a small influence on the oil/water equilibrium.

The overall result is however still that PR is insufficiently accurate to simulate the water/glycol mixture, and the influence of the gas-phase is minimal in this problem.

5.4.5 Case 5

In case 5 PRSV is used together with the Henley-Rosen algorithm for a four component system consisting of water, TEG, methane and decane. The results of the simulation in case 5 are given in Table 5-12.

Table 5-12: Phase composition in case 5								
	Gas phase		Oil phase		Water phase		K1	K2
	[mol %]	[mol]	[mol %]	[mol]	[mol %]	[mol]		
Water	2.58	0.005	1.92	0.014	100.00	0.081	$1.34 \cdot 10^{0}$	$2.58 \cdot 10^{-2}$
Glycol	0.00	0.000	47.41	0.340	0.00	0.000	$1.91 \cdot 10^{-6}$	$1.65 \cdot 10^{27}$
Methane	97.36	0.197	0.48	0.003	0.00	0.000	$2.04 \cdot 10^2$	$1.24 \cdot 10^{6}$
Decane	0.06	0.000	50.18	0.360	0.00	0.000	$1.19 \cdot 10^{-3}$	$5.875 \cdot 10^{17}$
Total	100.00	0.202	100.00	0.717	100.00	0.081	-	-

- Gas/liquid split (ψ): 0.202
- Liquid/liquid split (ξ): 0.899

The relationship between case 4 and 5 are similar to that between case 1 and 2. In this case it is again demonstrated that PRSV offers small improvements over PR but is unable to simulate the glycol/water mixture.

5.4.6 Case 6

In case 6 PR is used together with the Wong-Sandler mixing rule and the Henley-Rosen algorithm for a four component system consisting of water, TEG, methane and decane.

Like in two-phase Wong-Sandler calculations in case 3, the three-phase calculation gives no result, because the algorithm collapses. This happens after the first iteration, the K-values after the first iteration is given in Table 5-13.

Table 5-13:	Equilibrium	constants in	n case 6

	K1	K2
Water	$1.87 \cdot 10^{7}$	$4.22 \cdot 10^7$
Glycol	$2.03 \cdot 10^{10}$	$2.02 \cdot 10^{8}$
Methane	$6.63 \cdot 10^2$	$3.50 \cdot 10^3$
Decane	$7.75 \cdot 10^7$	$9.60 \cdot 10^{6}$

Like in case 3f, where methane was added to the two phase calculations, all the K-values becomes significantly higher than one, indicating that all components tries to create one single phase.

5.5 Part discussion/conclusion

Six different cases to investigate the water/glycol mixture were created in MATLAB. Unfortunately none of these cases were able to simulate the water/glycol mixture satisfactorily. They did however all illustrate some of the problems associated with simulation of the water/glycol mixture.

In the cases with PR and PRSV, glycol was even more unwilling than decane to mix with water. This demonstrates that water and glycol form a very non-ideal liquid mixture. Therefore some means of describing the excess energy of the water/glycol mixture must be incorporated in the thermodynamic calculations. In this report the Wong-Sandler mixing rule was introduced, unfortunately without any result.

The problem with the PR-WS could be the use in this report, that there are some unknown difficulties when PR-SW is combined with the separation algorithms. The problem could also be in the PR-SW equations themselves.

6 Final aim

An alternative to simulating the water/glycol mixture with MATLAB is to use a commercial thermodynamic process simulation program. In these programs the non-ideal liquid behaviour has already been incorporated in the calculations. The simulations with process simulation programs do not have to be limited to the water/glycol mixture, but can include the entire dehydration process. This has resulted in the following final aim for the report:

How is glycol dehydration plant simulated with commercial thermodynamic process simulation programs?

To clarify the tasks associated with the final aim 5 subquestions have been asked:

- What are the advantages of using a commercial thermodynamic process simulation program?
- What thermodynamic equations are used in process simulation programs?
- What problems must be taken into consideration when creating a simulation model in a process simulation program?
- What are the problems with simulation of a dehydration plant in process simulation programs?
- What is the effect of using different thermodynamic packages for the simulation?

There are numerous aspects in process simulation; in this report the focus is the thermodynamic calculations. Because of this focus some aspects usually incorporated in the design and simulation of a dehydration unit is not considered in this report, it is:

- Unit efficiency.
- Energy consumption.
- Pressure loss in the units.

As a result of these assumptions the values of these parameters will be the default values in the simulation program. The exception is units where specific values are required, e.g. pressure loss in heat exchangers, in these cases the efficiency will be one and loss will be zero.

Because the energy consumption is not treated in the simulation, there will be no integration of heat exchangers. The streams are just cooled or heated to the required temperature.

7 Process simulation

When a process plant is simulated with process simulation programs there are several things that must be taken into account. This includes the settings for the simulation program, design and settings for the process plant. When the simulation is being created additional problems might arise, because variables needs to be defined or values estimated before the simulation can be calculated.

7.1 Process simulation programs

The purpose of section 5 was to simulate a flash separation with MATLAB. A flash separation is a relatively simple unit, which nevertheless required some hundred code lines. When entire process plants are to be simulated, this would require programs for each unit in the plant, programs that often would be more complicated than for a flash separation. Programming entire process plants in MATLAB would therefore be very impracticable.

Fortunately there are several thermodynamic process simulation programs available. They offer the user a wide range of opportunity when selecting:

- Chemical components.
- Process units.
- Thermodynamic packages.
- Measuring units.

The programs contain thermodynamic data for numerous chemical components. Besides the chemical components in the program it is also possible to create user defined components. This can be e.g. be used for an oil-phase consisting of an unknown mixture of heavy hydrocarbons.

The process units can be connected in various ways, thus creating almost any process plant. The settings for the process units can furthermore be adjusted in several ways (depending on the unit), thereby increasing the accuracy of the simulation.

The programs also offer several thermodynamic packages for the calculations. A package contains all the thermodynamic equations required to calculate the system. The packages available are the commonly used EOS like PR and SRK, but also more seldom or special packages depending on the program. The special packages may be modifications of the traditional EOS, or they are specially created for a specific process, like glycol dehydration.

The input data and results can be given in several measuring unit sets, depending on the users' choice. The individual measuring unit sets can further more be customized by the user, e.g. using SI units but with the temperature in centigrade and the pressure in bar.

The available options may differ between the different process simulation programs. There may also be differences in the thermodynamic calculations in different programs, depending on the composition of the packages. Which process simulation program that is used, is often specified beforehand, in this report the process simulation program used will be HYSYS. HYSYS has already been used as the source of the thermodynamic data used in the MATLAB program.

The version of HYSYS used for the process simulations in this report is Aspen HYSYS 2006.5. The thermodynamic packages available in HYSYS include a Glycol package specially designed for TEG dehydration. Besides the Glycol package, PR is the recommended EOS simulation of TEG dehydration [O1].

The PR equations have already been described in section 4.3, although there are some differences between which equations that are given in section 4.3 and the HYSYS documentation [O1].

The Glycol package is based on the Twu-Sim-Tassone EOS. This EOS is designed with a mixing rule which incorporates the Helmholtz excess energy to describe the non-ideal behaviour of the water/glycol mixture. The glycol package gives an improved accuracy in the simulation of glycol than PR, especially in the vapour-phase [A5].

The glycol package is applicable for the temperatures, pressures and gas compositions normally encountered in a glycol plant. This applies for contactor for temperatures from 15 to 50 $^{\circ}$ C and pressures between 10 and 100 bars. In the regenerator it is temperatures between 202 and 206 $^{\circ}$ C, for a pressure of 1.2 bars. Within these limits the glycol packages can accurately predict: [O2]

- Activity coefficients of the TEG/water solution within an average absolute deviation of 2%.
- Dew-point temperatures within an average error of ± 1 °C.
- Water content of gas within an average absolute deviation of 1 %.

The glycol package and PR will be used as the EOS in the HYSYS simulation of the glycol plant.

7.2 Simulation model

The elements and design guidelines for a glycol plant have already been given in section 3.3. In this section the final design of the glycol plant will be given along with the design specifications for the plant.

7.2.1 Dehydration simulation

When the process plant on an offshore platform is simulated the dehydration unit is often a problem. The dehydration plant solves a simple problem, removing the water, but it can be difficult to simulate. There is an alternative to simulating an entire dehydration plant, namely to insert a component splitter.

A component splitter is a non-thermodynamic separator, where the composition of the output streams is determined by the programmer. Before the component splitter can be used with some accuracy input data for the unit must be known. If data for an actual unit is not available the input data must be calculated by a simulation. With input data from a dehydration unit, the effect of the component splitter is equal to simulating the entire dehydration plant.

Using a component splitter instead of a dehydration unit is much faster. There may be a small error if the input data in the component splitter comes from calculations with a different gas composition.

The simulation of the dehydration plant is often neglected when the process plant on an offshore platform is simulated. It is also possible to create new dehydration plants from empirical data from already existing plant, without simulations. But simulations are necessary when dehydration plants are optimized, or the actual split needs to be calculated for a component splitter.

7.2.2 Dehydration plant specifications

The specifications consist of the composition, flow, temperature and pressure for the wet gas, and the required purity of the lean glycol. Finally the operation conditions for the contactor are given.

Gas

Oub						
٠	Flow:	$250 \text{ MMscfd} (6.698 \cdot 10^6 \text{ Nm}^3/\text{d})$				
•	Temperature:	37.9 °C				
٠	Pressure:	70 bara				
٠	Molar composition:					
	Methane:	0.88322				
	Ethane:	0.06755				
	Propane:	0.01995				
	i-Butane:	0.00688				
	n-Butane:	0.00769				
	i-Pentane:	0.00388				
	n-Pentane:	0.00183				
	n-Hexane:	0.00177				
	n-Heptane:	0.00132				
	n-Octane:	0.00050				
	n-Nonane:	0.00012				
	n-Decane:	0.00005				
	Water:	0.00121				
	Nitrogen:	0.00237				
	Carbon dioxide	0.00166				
	Hydrogen sulphide	0.00000				
Glyco	D					
•	Type:	TEG				
٠	Lean TEG purity:	\geq 99.5 wt%				
•	Lean TEG temperature:	80°C				
•	Lean TEG pressure:	1.2 bara				
Conta	actor					
٠	Pressure:	70 bara				
•	Gas temperature :	25 °C				
٠	Glycol Temperature:	30 °C				

The lean TEG purity must be equal to or higher than 99.5 wt%, to insure this demand is met at all times the value used in the dehydration plant will be 99.6 wt%. The specifications for the dehydration plant will be incorporated in the final plant design.

7.2.3 Dehydration plant design

The design of the dehydration plant has its origin in the processes described in section 3.3, and the requirements for the plant given in section 7.2.2. A description of the dehydration plant will be given in this here along with some plant specifications.

The gas inlet temperature differs from the temperature in the contactor. The gas is therefore cooled and the liquids removed in the inlet scrubber. The TEG temperature and pressure differs from the contactor specification, first the pressure is raised by a pump and then the temperature is reduced.

The contactor is a column with five trays. The TEG inlet and gas outlet is in the top of the column, the TEG outlet and gas inlet is in the bottom of the column. The TEG flow is dependent on the water contents in the wet gas, with a flow of 0.025 m^3 TEG/kg Water.

After the contactor the pressure of the now rich TEG is reduced, by a valve. The temperature is increased before the rich TEG is flashed to remove hydrocarbons dissolved in the TEG in the contactor. The temperature is increased once more before the regenerator.

The regenerator is a column with five trays plus a condenser and a boiler. The rich TEG enters the regenerator on the middle tray. Because a TEG purity of 99.6 wt% is required stripping gas is added to the boiler of the column. The stripping gas is pure nitrogen with a flow of 28.3 Nm^3 gas/m³ rich TEG. The temperatures given for the boiler and condenser are 204.4 °C (400 °F) and 98.9 °C (210 °F) respectably.

After the regenerator the once again lean TEG is cooled to 80 °C to save the pump. At this point there will also be some kind of TEG makeup system to replace the TEG lost to the gas phases in the dehydration plant. Finally the lean TEG is recycled to the start point.

The described dehydration plant is illustrated with operating conditions in Figure 7-1.



The operating conditions given in Figure 7-1 are the outlet conditions for the process units.

7.2.4 Creating the simulation model

When the design criteria for the simulation models have been determined, there are still some criteria that have to be determined during the creation of the model. An example is the flow of TEG and stripping gas that depends on the flow of other streams in the model.

The HYSYS simulation being created here uses the glycol thermodynamic package, and the EuroSI unit set.

Dehydration

The first step in the simulation is to create the gas streams into the contactor column. The gas is cooled and flashed to remove the condensed components. The water content in the flashed gas is used to calculate the TEG flow. In this case the water flow is 107.7 kg/h, thus making the TEG flow 2.69 m³/h.

Now the size of the TEG flow is known, the TEG pressure and temperature must be adjusted to required criteria for the contactor. When the gas and TEG outlet streams are added to the contactor column it can now be calculated.

Regeneration

The next step is to reduce the TEG pressure, and increase the temperature before it is flashed to remove dissolved water and hydrocarbons released at this new temperature and pressure. After the separator the temperature is increased before the TEG enters the regenerator.

The flow of stripping gas into the regenerator is dependent on the glycol flow. In this case the flow is calculated to $80 \text{ Nm}^3/\text{h}$. This flow goes into the reboiler, even though HYSYS illustrates the stream on the side of the regenerator column.

In the regenerator there are two degrees of freedom, which signify that two variables must be defined. This is perfect since two design criteria are given namely the condenser and reboiler temperature. When these criteria are entered into HYSYS the column control screen is shown in Figure 7-2.

Column: Regenvert Design Connections Monitor Specs Specs Summary Subcooling Notes	Prator / COL2 Fluid P Optional Checks Input Summary Iter Step Equi 448 0,0000 0 449 0,0000 0 450 0,0000 0 451 0,0000 0 452 0,0000 0 Specifications Temperature - 2 Temperature	kg: Basis-1 / Glycol Pa View Initial Estimates ilibrium Heat / Spec 000000 1.353639 000001 1.353624 000001 1.353624 000001 1.353624 000001 1.353624 000001 1.353624 0000011 1.353624 0000008 1.353624 Specified Value 98,89 C 204,4 C 204,4 C	Current Value	200,0 180,0 160,0 140,0 120,0 100,0 80,00 0 1 Wt. Error -0,0276 -0,0245	Active	Position from 3 4 Estimate	Top
Design Paramet	View A	Add Spec <u>G</u> roup A Worksheet Performanc R <u>u</u> n <u>R</u> eset	active Up; e Flowsheet Unconv	date Inactive Reactions _ D erged	Degra	ees of Free	dom 0

Figure 7-2: Regenerator control screen in HYSYS.

As shown in Figure 7-2 the HYSYS calculation of the column is unable to converge with only the condenser and reboiler temperature given. The design parameters can be supplemented by estimated start values that help to solve the calculations.

Alternatively other values may be defined that describes the distillation better than the temperatures. These new designed values could be some of the estimates previously used only to guide the calculation.

Column control conditions

Obvious estimates are values that already are known or can be easily estimated. The first estimate is the lean TEG purity, the TEG mass fraction in the liquid phase in the reboiler is set to 0.996.

A second estimate is the total vapour flow from the condenser. This estimate is calculated as the total flow of stripping gas into the regenerator plus the flow of water in the TEG. The total vapour estimate is calculated as mass flow, giving an estimated flow of 217 kg/h.

A third estimate is given by the reflux ratio from the condenser. This value is not estimated but calculated by HYSYS. The third estimate is actually redundant in most cases because the first two estimates are sufficient to describe the column. But in some rare cases when the column is reset, it will not converge again. To avoid these situations this third initial estimate are added to the column. This gives the column control screen in Figure 7-3.

Design Connections	Optional Checks	Profile	200.0				
fonitor ipecs ipecs Summary iubcooling lotes	Iter Step Equ 3 0,0001 2 4 1,0000 3 5 0,0046 968 6 1,0000 7 7 1,0000 7	ilibrium Heat / Spec 0,300382 0,260558 3,180229 0,964837 2,239936 1,159291 0,382854 0,407376 1,256205 0,080444	Temp Press Flows	180.0 160.0 140.0 120.0 100.0 80.00 0 1	2	3 4	5 6
		Specified Value	Current Value	Wt. Error	Active	Estimate	Current
Temperature - 2		98,89 C	98,9	0,0000	N	<u> </u>	<u></u>
	Reflux Batio	2,500e-002 217.0 kg/h	2.55e.002	0,0000		4 1 4	Ē
	Vapour Flow		217				
	Comp Fraction	0,9960	0,996	0,1250	Ē	<u>.</u>	
		Add Spec	sctive Upg	late Inactive		ees of Free	jom 0
				1976 - 17 - 18 - 18 - 18 - 18 - 18 - 18 - 18		_	

Figure 7-3: Regenerator control screen in HYSYS.

With these estimates the column now converges.

TEG makeup

The TEG from the regenerator is cooled and recycled back to the TEG inlet stream. To do this a logical recycle operator must be inserted between the two streams. There is a problem with the recycling of the TEG; this is that small amounts of TEG are lost from the system in the gas flow from the contactor, separator and regenerator. The lost TEG must be replaced, or less TEG than required is recycled.

To makeup the lost TEG the gas stream from the contactor, separator and regenerator is entered into a component splitter. In the component splitter the TEG is separated from the gas, creating a stream of pure TEG that is transferred back to the TEG stream. A mixer is required to mix the recovered TEG with the TEG from the regenerator.

The final HYSYS model

The final HYSYS model of the dehydration plant is illustrated in Figure 7-4



Figure 7-4: The HYSYS model of the dehydration.
The size of Figure 7-4 makes it a little difficult to see all the details, therefore a larger version of Figure 7-4 can be seen in App. 3.

The HYSYS model has been used in two cases, with different thermodynamic packages, thereby investigating the influence the thermodynamics.

7.3 Simulation results

The HYSYS simulation model developed in section 7.2 have been used for two cases. The two cases differ in the thermodynamic package used, in case 11 it is Glycol and case 12 uses PR.

The results given in the report are limited to the results of interest for the cases. All the results for the cases are given as pdf-files on the CD in the folder HYSYS.

7.3.1 Case 11

The efficiency of the dehydration simulation using the glycol thermodynamic package is evaluated by the water content in the dry gas, the water mass balance in the simulation model and the TEG regeneration.

Dry gas water content

The efficiency of the dehydration can be measured on the water contents in the dry gas. This is usually evaluated by the dry gas dew-point. Figure 3-2 gives the water dew-point temperature as function of the lean TEG purity and contactor temperature. For 99.6 wt% lean TEG and a contactor temperature of 30 °C, the expected water dew-point temperature is -30 °C.

In HYSYS it is however problematic to use this value to evaluate the efficiency of the dehydration. The dew-point curve for the dry gas (stream 5) in case 11 is given in Figure 7-5.



Figure 7-5: Dew-point curve for stream 5.

The problem with the dew-point curve in Figure 7-5 is that it is the hydrocarbon dewpoint curve. With the given gas composition the hydrocarbon dew-point is considerably higher than the water dew-point. Unfortunately only one dew-point curve is available in HYSYS, the total gas dew-point. The problem can not be resolved by removing all components except water and methane from the gas stream. In this case the dew-point curve stops at the critical point; 45.99 bars and -82.59 °C.

Instead the water content in the gas before and after the contactor is given in Table 7-1.

Table 7-1: water content in the gas before and after the contactor.							
	Mole fraction	Mass fraction	Mole flow	Mass flow			
	[mole%]	[wt%]	[kmole/h]	[kg/h]			
Stream 3	0.0481	0.0460	5.9782	107.6975			
Stream 5	0.0010	0.0009	0.1199	2.1604			
Difference	-	_	5.8583	105.5371			

 Table 7-1: Water content in the gas before and after the contactor.

From the data in Table 7-1 it can be concluded that the contactor removes 98% of all the water from the wet gas.

Water flow

When evaluating a dehydration process it is important to investigate the water flow in the simulation. The water mass balance is therefore given in Table 7-2.

Table 7-2: Water mass balance in case 11						
	Stream	Flow	Total			
		[kg/h]	[kg/h]			
u	1	271.4461	282 6427			
Ĥ	16	11.1966	282.0427			
	4	163.7485				
<u>ب</u>	5	2.1604				
)n	10	0.0874	282.6427			
Ŭ	13	11.2308				
	14	105.4156				

The two main exit points for the water is in the liquid flow from the inlet scrubber (58%) and the vapour flow from the regenerator (37%). The most cost efficient dehydration method is the inlet scrubber; this decreases the required water removal capacity for the dehydration plant. With the actual water flow in the inlet scrubber, it would perhaps be more appropriate to call it a separator. The main idea with the inlet scrubber is to remove liquid droplets from the gas.

Another interesting point is the water flow in the TEG stream into the system (16) and out of the regenerator (13). There is a small difference in flow; this indicates that HY-SYS accepts small differences between recycled streams.

Glycol purity

The last parameter to be investigated is the efficiency of the regenerator. The dehydration unit is design with a lean TEG purity of 99.6 wt%. The lean TEG purity in stream 13 is 99.62 wt%, after the makeup TEG is added the purity becomes 99.63 wt%.

The result of the TEG regeneration is as expected.

Glycol loss

The total loss of TEG in the dehydration plant equals the size of the makeup TEG stream. In case 11 the TEG loss is 0.5422 kg/h, compared to the total TEG flow of 3033.1 kg/h, giving a TEG loss of 0.018%.

7.3.2 Case 12

The parameters evaluated in case 12 are similar to those evaluated in case 11.

Dry gas water content

As explained in case 11, it is not possible to get the water dew-point curve in HYSYS. HYSYS does however have a special hydrate formation temperature curve feature in the PR thermodynamic package. The hydrate formation temperature is given in Figure 7-6.



Figure 7-6: Hydrate formation temperature.

Although the hydrate formation temperature is a useful to avoid hydrate formation in process equipment and pipelines, it can not be used to evaluate the water contents in the gas. This will therefore like in case 11 be evaluated by comparing the water content in the gas streams before and after the contactor, the values are given in Table 7-3.

Table 7-5. Water content in the gas before and after the contactor.							
	Mole fraction	Mass fraction	Mole flow	Mass flow			
	[mole%]	[wt%]	[kmole/h]	[kg/h]			
Stream 3	0.0600	0.0574	7.4610	137.0357			
Stream 5	0.0013	0.0012	0.1585	2.8553			
Difference	-	-	7.3025	134.1804			

Table 7-3: Water content in the gas before and after the contactor.

The water flows in Table 7-3 gives a water removal in the contactor of 98%.

Water flow

The mass balance for water in case 12 is given in Table 7-4.

	Stream	Flow	Total
		[kg/h]	[kg/h]
u	1	271.4461	281 7417
I	16	10.2956	201./41/
	4	137.0357	
<u>ب</u>	5	2.8553	
)u(10	0.2188	281.7417
0	13	10.2873	
	14	131.3446	

The two main exit points for the water are still the inlet scrubber with 49% and the regenerator with 47%.

Glycol purity

The last parameter to be investigated is the efficiency of the regenerator. The dehydration unit is design with a lean TEG purity of 99.6 wt%. The lean TEG purity in stream 13 is 99.63 wt%, after the makeup TEG is added the purity is still 99.63 wt%.

Glycol loss

The total TEG loss in case 12 is 0.6640 kg/h; with a total TEG flow of 3033.1 kg/h this gives a TEG loss of 0.022%

7.4 Part discussion/conclusion

When an entire production plant for a platform is designed the easiest way to simulate the dehydration plant is to insert a component splitter. A component splitter only requires data from an actual dehydration plant or a simulation of a plant, to dehydrate the gas as efficiently as simulating the entire dehydration plant. Including the dehydration plant in the simulation of the entire process plant is a time consuming task that just increases the complexity of the simulation.

When a dehydration plant is simulated it is important to know the operating parameters and units required in the dehydration plant. With this design determined, it is easy to create the HYSYS simulation, with the possible exception of the regenerator column.

Determining the correct parameters and estimates required in the regenerator is more difficult. It is often necessary to experiment a little with different parameters, before the column will converge.

When the dehydration simulation finally worked it became case 11. Case 12 was easily created by changing the thermodynamic package from glycol to PR.

There are some differences in the results of case 11 and 12. The largest difference is in the water flow, where 58% of the water was removed in the inlet scrubber in case 11, this percentage has fallen to 49% in case 12.

Even with the larger water flow in case 12, 98% of the water is still removed in the contactor. The water contents in the dry gas are in case 12 increased 0.6949 kg/h. But without any data for the water contents in dry gas form an actual dehydration unit, it can not be concluded if case 11 or 12 give the most accurate result.

The recovery rates in the two cases are very similar with a lean TEG purity of 99.63 wt%. Compared to the expected purity this is a fairly accurate result, because the given stripping gas flow of 28.3 Nm^3 gas/m³ rich TEG (4 scf/gal) is from a figure in [B5].

The recovery rate only fits the expected result at this gas stripping flow. In the literature the recovery rate does not increase with increasing stripping gas flows. This is however the case in the HYSYS simulations.

The glycol package gives a slightly smaller glycol loss than PR. But with a loss of 0.018% compared to 0.022% the difference is minimal.

8 Discussion

The main part of the report has been discussed during the report. This final discussion will mainly be a repetition of the main points in the part discussions.

There are four possible methods for gas dehydration, but glycol is chosen because it offers the best cost benefit choice. The efficiency of a dehydration plant is determined by the water dew-point for the gas. There are several design options for a glycol dehydration plant. The integration of heat exchangers is very important to minimize the energy consumption in the dehydration plant. This is due to the temperature differences between the contactor and regenerator columns.

There are several thermodynamic EOS that can be used for simulation where ideal liquid mixtures can be assumed. For non-ideal liquid mixtures it is however necessary to introduce equations like the Wong-Sandler mixing rule, that incorporated the excess Gibbs energy and the activity coefficients in the calculations.

The MATLAB simulations were unsuccessful because they could not simulate a water/glycol phase. With PR and PRSV the problem was the excess energy for the mixture, with PR-WS the problem was to reach a solution of the calculations.

In HYSYS the easiest way to simulate a dehydration plant is to insert a component splitter. But it is possible simulate the dehydration plant in HYSYS using the thermodynamic packages Glycol and PR. The result differs a little between the two thermodynamic packages; although it can not be determined which package yields the most accurate result.

9 Conclusion

There are several processes involved in processing the reservoir fluid into oil, gas and water. One of the most important processes offshore is gas dehydration, because wet gas increases corrosion and can course plugs from ice or gas hydrate.

Absorption with TEG offers the best cost benefit choice for the dehydration process. The dehydration process is divided into two parts, the dehydration and the regeneration. In the dehydration part, gas is dried by the glycol. In the regeneration part the water is removed from the glycol so it can be used for dehydration once more. There are several possibilities in the design of the dehydration plant. The design options include the integration of heat exchangers and the recovery rate of the glycol.

The main problem involved in simulation of the dehydration process is the non-ideal liquid behaviour of the water/glycol mixture. Process simulation calculations are conducted with thermodynamic equations, designed for ideal liquid mixtures. The problem can be solved by introducing thermodynamic equations that include the liquids excess parameters. The Wong-Sandler mixing rule is an example of equations that include the excess liquid parameters. The Wong-Sandler mixing rule can be combined with the classic equation of state like PR and PRSV.

It was not possible to simulate the water/glycol mixture adequately in MATLAB. The PR and PRSV simulations could not describe the excess liquid parameters of the mixture. The simulations with PR-WS were also unsuccessful because the MATLAB program could not calculate the problem. As long as the MATLAB code can not simulate the water/glycol mixture it is impossible to simulate a dehydration plant with MAT-LAB.

It is however possible to simulate the dehydration unit in HYSYS. In HYSYS the excess liquid behaviour has been incorporated into the thermodynamic equations. It vas therefore possible to create two cases, that simulated the dehydration unit. There are small differences between the two cases. Without any reliable data for an actual dehydration plant it is impossible to determine which thermodynamic package that gives the most accurate result.

10 References

The literature used is divided into four categories books, articles, web pages and other.

Books

- B1. <u>Offshore Book An introduction to the offshore industry</u>; Offshore Center Danmark; January 2008.
- B2. <u>Oil and Gas Production in Denmark 2006</u>; The Danish Energy Authority; 2007; ISBN: 978-87-7844-659-6.
- B3. <u>Natural Gas Production, Processing, Transport</u>; A. Rojey et.al; Editions Technip; 1997; ISBN: 2-7108-0693-2.
- B4. <u>Fundamentals of Natural Gas Processing</u>; Arthur J. Kidnay and William R. Parrish; Taylor & Francis Group 2006; ISBN: 0-8493-3406-3.
- B5. <u>Oilfield Processing of Petroleum Vol. 1 Natural Gas;</u> F. S. Manning and R. E. Thompson; 1st ed.; Pennwell Publishing Company 1991; ISBN: 0-87814-342-2.
- B6. <u>Perry's Chemical Engineers' Handbook</u>; Robert H. Perry and Don W. Green; 8th ed.; McGraw-Hill 2007; ISBN: 978-0-07-142294-9.
- B7. <u>Specification for Glycol-Type Gas Dehydration units API Specification</u> <u>12GDU</u>; American Petroleum Institute; 1990.
- B8. <u>Chemical and Engineering Thermodynamics</u>; Stanley I. Sandler; 3rd ed.; John Wiley & Sons Inc. 1999; ISBN: 0-471-18210-9.
- B9. DACHEMA Vol. I Part 1: Aqueous-organic Systems;
- B10. <u>Separation Process Principles</u>; J. D. Seader and E. J. Henley; 1st ed.; John Wiley & Sons Inc. 1998; ISBN: 0-471-58626-9.

Articles

All the other sources are on the CD in the path \SOURCES\ARTICLES\.

- A1. Equation of State Mixing Rule for Nonideal Mixtures Using Available Activity Coefficient Model Parameters and That Allows Extrapolation over Large Ranges of Temperature and Pressure; D. S. H. Wong, H. Orbey and S. I. Sandler; Ind. Eng. Chem. Res. 1992, 31, 2033-2039.
- A2. <u>Vapor-Liquid Equilibria of the Three Binary Systems: Water + Tetraethylene</u> <u>Glycol (TEG), Ethanol + TEG, and 2-Propanol + TEG</u>; T. Tsuji, T. Hiaki and M. Hongo; Ind. Eng. Chem. Res. 1998, 37, 1685 – 1691.
- A3. <u>A Theoreticaly Correct Mixing Rule for Cubic Equations of State</u>; D. S. H. Wong and S. I. Sandler; AIChE Journal; May 1992 Vol. 38, No. 5.
- A4. <u>Reformulation of Wong-Sandler Mixing Rule for Cubic Equations of State;</u> H. Orbey and S. I. Sandler; AIChE Journal; March 1995 Vol. 41, No. 3.
- A5. <u>Using a More Thermodynamically Adapted Model Improves Modelling of TEG</u> <u>Dehydration Units – Allows Better Operation and Efficiency</u>; W. van Wassenhove, C. Twu and J. A. Feliu; Aspentech.

Web pages

- W1. www.alaska-in-pictures.com/data/media/17/pipeline-pig_3240.jpg
- W2. <u>www.cspg.org/conventions/Gussow2007/images/tech-gas-hydrates.jpg</u>

Other

All the other sources are on the CD in the path \SOURCES\OTHER\.

- O1. AspenCOMThermo2006_5-Ref.
- O2. <u>AspenHYSYSSimBasis2006_5-Ref.</u>

Index

APPENDIX	X 1:	CRITICAL DATA	.2
A1.1	NRTL IN	IPUT DATA	.2
APPENDIX	X 2:	MATLAB CODE	.3
A2.1 A2.2	CASE 1 . CASE 4 .		.3 .8
APPENDIX	X 3:	HYSYS PFD	15

Appendix 1: Critical Data

The critical data for the MATLAB cases comes from HYSYS. They are also given in this appendix.

Table 1-1: Critical data.						
Component	Tc	p _c	ω	к 1	1	1
	[K]	[Pa]			Case 3	Case 6
Water	647.25	$22.12 \cdot 10^{6}$	0.3440	-0.0767	0.6605	9.8929
TEG	727.05	$1.42 \cdot 10^{6}$	0.6900	0.8175	0.8900	0.5415
Methane	190.70	$4.64 \cdot 10^{6}$	0.0115	-0.0193	-	7.6199
Decane	617.55	$2.108 \cdot 10^{6}$	0.4885	0.0643	1.0439	1.0372

The activity coefficient in Table 1-1, have been calculated in MATLAB with the multicomponent NRTL model. The MATLAB program for the NRTL calculations is on the attached CD in the folder MATLABACTIVITY.

Besides calculating the activity coefficients, the NRTL program also calculates the excess Gibbs energy:

- Case 3: -153.3433 J/mol
- Case 6: 170.6076 J/mol

A1.1 NRTL input data

The MATLAB program for the NRTL calculations also requires input data, like the critical data the NRTL data comes from HYSYS.

Table 1-2: a for NRTL.							
a [cal/gmol]	Water	TEG	Decane	Methane			
Water	-	-776.787	0.000	-142.234			
TEG	1914.666	-	0.000	0.000			
Decane	0.000	0.000	-	-14.928			
Methane	26.596	0.000	10.832	-			

Table 1-2: a for NRTL

|--|

b [cal/(gmol*K)	Water	TEG	Decane	Methane
Water	-	1.184	0.000	32.308
TEG	-3.340	-	0.000	0.000
Decane	0.000	0.000	-	4.899
Methane	-0.090	0.000	-0.011	-

Table 1-4: α for NRTL.

α	Water	TEG	Decane	Methane
Water	-	1.613	0.000	0.000
TEG	1.613	-	0.000	0.000
Decane	0.000	0.000	-	0.000
Methane	0.000	0.000	0.000	-

Appendix 2: MATLAB code

The MATLAB codes for case 1 and 4 are given in this appendix

A2.1 Case 1

```
%MATLAB script to find phase composition for Water/Glycol/Decane
%mixture by solving LL equilibrium, using PR - Peng-Robinson
%and the Rachford-Rice algorithm
clear
clc
%---- Global Variables
                            ____
global C; %Number of components
C=3;
global R;
                   %The Gas Constant
R=8.314;
                   %[J/(mol*K)]
global pc;
                   %Critical pressure [pa]
global Tc;
                   %Critical temperature [K]
global omega;
                   %Acentric factor
%---- Critical data for components
                                        ____
%Data is from HYSYS critical component data database - K-values are
%guessed
for c=1
                %Water
    Tc(c)=647.25;
                       %Critical Temperature [K]
    pc(c)=22.12e6;
                       %Critical Pressure [Pa]
    omega(c) = 0.344;
                       %Acentric factor
    K(c) = 1.439e - 2;
                       %Water in water
end
for c=2
                %TEGlycol
                    %Critical Temperature [K]
    Tc(c)=727.05;
    pc(c) = 1.419e6;
                       %Critical Pressure [Pa]
    omega(c)=0.690;
                       %Acentric factor
    K(c) = 4.985e39;
                       %TEG in water
end
for c=3
                %Decane
                   %Critical Temperature [K]
    Tc(c)=617.55;
    pc(c) = 2.108e6;
                       %Critical Pressure [Pa]
    omega(c)=0.4885;
                       %Acentric factor
    K(c) = 1.276e21;
                       %Decane in water
end
8____
        Input data
                       ____
                                %Temp in [C]
Tflash=25;
T=Tflash+273.15;
                                %Temp in [K]
                                %Pressure in [barg]
pflash=0.0;
p=(1+pflash)*1e5;
                               %Pressure in [Pa]
%Composition [mol]
z(1)=0.10;
           %Water
z(2)=0.34;
            %TEG
           %Decane
z(3)=0.56;
%Normalizing for z-values
zsum=0;
for c=1:C
    zsum=zsum+z(c);
end
```

```
for c=1:C
    z(c) = z(c) / zsum;
end
%Start guess for liquid-liquid split
                    %Start guess for LL split (L1/(L1+L2))
psi=0.913;
%---- Calculations
                       ____
%Loop to update the K values
flag=1;
ma=0;
MA=100; %Max no. of iterations in loop a
while flag
    %---- Update of psi
                           ____
    error=1;
    mb=0;
    MB=100;
    while error > 1e-6
        %Sum of f(1) og fdot(1)
        f=0;
        fdot=0;
        for c=1:C
            f=f+z(c)*(1-K(c))/(1+psi*(K(c)-1));
            fdot=fdot+z(c)*(K(c)-1)^2/(psi*(K(c)-1)+1)^2;
        end
        psi=psi-f/fdot;
        if psi<0 || psi>1
            disp('New initial value for psi')
            fprintf('psi
                            %.4f\n',psi)
            return
        end
        error = abs(f);
        mb=mb+1;
        if mb>=MB
            disp('max no. of iterations in loop b')
            return
        end
    end
    %---- Phase composition
                               ____
    for c=1:C
        %Composition of water
        x(1,c)=z(c)/(1+psi*(K(c)-1));
        %Composition of oil
        x(2,c) = K(c) * x(1,c);
    end
    %---- a and b in Peng-Robinson
                                      ____
    for c=1:C
        kappa=0.37464+1.54226*omega(c)-0.26992*omega(c)^2;
        alfa=(1+kappa*(1-sqrt(T/Tc(c))))^2;
        a(c,c)=0.45724*R^2*Tc(c)^2/pc(c)*alfa;
        b(c)=0.07780*R*Tc(c)/pc(c);
    end
```

```
for c=2:C
    for n=1:(c-1)
        a(c,n)=sqrt(a(c,c)*a(n,n));
        a(n,c)=a(c,n);
    end
end
8----
        Fugacity coefficients in water
                                          ____
%am and bm in water
am=0;
bm=0;
for c=1:C
    for n=1:C
        am=am+x(1,c)*x(1,n)*a(c,n);
    end
    bm=bm+x(1,c)*b(c);
end
%A and B for cubic equation
A=am*p/(R*T)^2;
B=bm*p/(R*T);
%Peng-Robinson cubic EOS
Z1=B;
error=1;
mc=0;
MC=100;
while error > 1e-6
    g=Z1^3+(-1+B)*Z1^2+(A-3*B^2-2*B)*Z1+(-A*B+B^2+B^3);
    gdot=3*Z1^2+2*(-1+B)*Z1+(A-3*B^2-2*B);
    Z1=Z1-g/gdot;
    error=abs(g);
    mc=mc+1;
    if mc>=MC
        disp('max no. of iterations in loop c')
        return
    end
end
%Fugacity coefficients
for c=1:C
    xa=0;
    for n=1:C
        xa=xa+x(1,n)*a(c,n);
    end
    phi(1,c) = exp(b(c)/bm*(Z1-1)-log(Z1-bm*p/(R*T))-...
    am/(2*sqrt(2)*bm*R*T)*(2*xa/am-b(c)/bm)*...
    log((Z1+(1+sqrt(2))*bm*p/(R*T))/(Z1+(1-sqrt(2))*bm*p/(R*T))));
end
8----
        Fugacity coefficients in oil
                                       ____
%am and bm in oil
am=0;
bm=0;
for c=1:C
    for n=1:C
        am=am+x(2,c)*x(2,n)*a(c,n);
    end
    bm=bm+x(2,c)*b(c);
end
```

```
%A and B for cubic equation
    A=am*p/(R*T)^2;
    B=bm*p/(R*T);
    %Peng-Robinson cubic EOS
    Z2=B;
    error=1;
    mc=0;
    MC=100;
    while error > 1e-6
        g=Z2^{3}+(-1+B)*Z2^{2}+(A-3*B^{2}-2*B)*Z2+(-A*B+B^{2}+B^{3});
        gdot=3*Z2^{2}+2*(-1+B)*Z2+(A-3*B^{2}-2*B);
        Z2=Z2-g/gdot;
        error=abs(g);
        mc=mc+1;
        if mc>=MC
            disp('max no. of iterations in loop c')
            return
        end
    end
    %Fugacity coefficients
    for c=1:C
        xa=0;
        for n=1:C
            xa=xa+x(2,n)*a(c,n);
        end
        phi(2,c) = exp(b(c)/bm*(Z2-1)-log(Z2-bm*p/(R*T))-...
        am/(2*sqrt(2)*bm*R*T)*(2*xa/am-b(c)/bm)*...
        log((Z2+(1+sqrt(2))*bm*p/(R*T))/(Z2+(1-sqrt(2))*bm*p/(R*T))));
    end
    8----
            Update K and check for convergence
                                                    ____
    flag=0;
    for c=1:C
        if abs(K(c) - phi(1,c)/phi(2,c)) > K(c)*1e-3+1e-6
            flag=1;
            K(c) = phi(1,c)/phi(2,c);
        else
            K(c) = phi(1,c)/phi(2,c);
        end
    end
    ma=ma+1;
    if ma>=MA
        disp('max no. of iterations for loop a')
        return
    end
end
L1=1-psi;
L2=psi;
```

```
%---- Printing of results
                             ____
print1=1;
print2=1;
print3=1;
if print1==1
    fprintf('LLE with PR EOS\n');
    fprintf('P= %.2e Pa\n',p);
    fprintf('P= \&.2f Barg n', pflash);
    fprintf('T= \&.2f K \setminus n', T);
    fprintf('T= %.2f C\n',Tflash);
    fprintf('\n');
    fprintf('Water-phase
                           %.3f mol\n',L1);
    fprintf('Oil-phase
                           %.3f mol\n',L2);
    fprintf('\n');
end
if print1==1 && print2==1
    fprintf(' Total flow [mol%%]\n');
    fprintf('Water
                        %.4f\n',z(1)*100);
    fprintf('TEGlycol
                       %.4f\n',z(2)*100);
    fprintf('Decane
                       %.4f\n',z(3)*100);
    fprintf(' \ n');
    fprintf(' Water-phase [mol%%]
                                         [mol] \ ;
    fprintf('Water
                              %.4f %.4f\n',x(1,1)*100,x(1,1)*L1);
    fprintf('TEGlycol
                               %.4f %.4f\n',x(1,2)*100,x(1,2)*L1);
                               .4f \ .4f \ .4f \ .1,3) + 100, x(1,3) + L1);
    fprintf('Decane
    fprintf('\n');
    fprintf(' Oil-phase
                            [mol%%]
                                        [mol] n');
    fprintf('Water
                               %.4f %.4f\n',x(2,1)*100,x(2,1)*L2);
                              .4f \ .4f \ x(2,2) \times 100, x(2,2) \times L2);
    fprintf('TEGlycol
                              %.4f %.4f\n',x(2,3)*100,x(2,3)*L2);
    fprintf('Decane
    fprintf('\n');
end
if print1==1 && print3==1
    fprintf('Liquid-Liquid split (psi) %.3f\n',psi);
    fprintf('\n');
    fprintf(' K-values \n')
    fprintf('Water
                     %.3e\n',K(1));
    fprintf('TEGlycol
                       %.3e\n',K(2));
    fprintf('Decane %.3e\n',K(3));
end
```

A2.2 Case 4

```
%MATLAB script to find phase composition for
%Water/Glycol/methane/Decane mixture by solving VLL equilibrium,
%using PR - Peng-Robinson and the Henley-Rosen algorithm
clear
clc
%---- Global Variables
                            ____
global C;
                   %Number of components
C = 4;
global R;
                    %The Gas Constant
                    %[J/(mol*K)]
R=8.314;
                    %Critical pressure [pa]
global pc;
                    %Critical temperature [K]
global Tc;
global omega;
                    %Acentric factor
%---- Critical data for components
Data is from HYSYS critical component data database - K-values are
%guessed
for c=1
                %Water
    Tc(c)=647.25;
                        %Critical Temperature [K]
    pc(c)=22.12e6;
                        %Critical Pressure [Pa]
    omega(c) = 0.3440;
                        %Acentric factor
                        %Water in oil
    K(1,c) = 1.674e0;
                        %Water in water
    K(2,c)=2.705e-2;
end
for c=2
                %TEGlycol
    Tc(c)=727.05; %Critical Temperature [K]
                        %Critical Pressure [Pa]
    pc(c)=1.42e6;
    omega(c)=0.6900; %Acentric factor
K(1,c)=8.836e0; %Glycol in oil
    K(2,c) = 4.052 = 16;
                       %Glycol in water
end
for c=3
                %Methane
    Tc(c)=190.70; %Critical Temperature
    pc(c) = 4.641e6;
                        %Critical Pressure
    omega(c)=0.0115;
                       %Acentric factor
    K(1,c) = 1.724e2;
                       %Methane in oil
    K(2,c)=1.033e6;
                        %Methane in water
end
for c=4
                %Decane
    Tc(c)=617.55; %Critical Temperature [K]
    pc(c)=2.108e6;
                       %Critical Pressure [Pa]
    omega(c)=0.4885;
                       %Acentric factor
                        %Decane in oil
    K(1,c)=2.154e-3;
    K(2,c) = 2.776 = 18;
                        %Decane in water
end
8----
        Input data
                       ____
Tflash=25;
                                 %Temp in [C]
T=Tflash+273.15;
                                 %Temp in [K]
pflash=0.0;
                                %Pressure in [barg]
p=(1+pflash)*1e5;
                                %Pressure in [Pa]
```

```
%Composition [mol]
z(1)=0.10;
             %Water
z(2)=0.34;
             %TEG
z(3)=0.20;
             %Methane
z(4)=0.36;
             %Decane
%Normalizing for z-values
zsum=0;
for c=1:C
    zsum=zsum+z(c);
end
for c=1:C
    z(c) = z(c) / zsum;
end
8----
        Start guess
                        ____
psi=0.555;
                  %Start guess for vapor-liquid split (V/F)
xi=0.822;
                  %Start guess for liquid-liquid split (L1/(L1+L2))
8----
       Calculations
                         _ _ _ _
%Loop to calculate K(1) & K(2)
flag=1;
ma=0;
MA=100;
         %Max no. of iterations
while flag && ma<MA
    8----
            Calculation of psi and xi
                                            ____
    error=1;
    mb=0;
    MB=100;
    while error > 1e-6 && mb<MB
        %Sum of f(1)
        f(1) = 0;
        for c=1:C
            f(1)=f(1)+z(c)*(1-K(1,c))/...
                (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+psi*K(1,c));
        end
        %Sum of f(2)
        f(2) = 0;
        for c=1:C
            f(2)=f(2)+z(c)*(1-K(1,c)/K(2,c))/...
                 (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+psi*K(1,c));
        end
        %sum of fdot(1,1), fdot(1,1)=d(f(1))/d(psi)
        fdot(1,1)=0;
        for c=1:C
            fdot(1,1)=fdot(1,1)-z(c)*(1-K(1,c))*...
                 (-xi-(1-xi)*K(1,c)/K(2,c)+K(1,c))/...
                (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+...
                psi*K(1,c))^2;
        end
        %sum of fdot(1,2), fdot(1,2)=d(f(1))/d(xi)
        fdot(1,2)=0;
        for c=1:C
            fdot(1,2)=fdot(1,2)-z(c)*(1-K(1,c))...
                *((1-psi)-(1-psi)*K(1,c)/K(2,c))/...
                (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+...
                psi*K(1,c))^2;
        end
```

```
%sum of fdot(2,1), fdot(2,1)=d(f(2))/d(psi)
    fdot(2,1)=0;
    for c=1:C
        fdot(2,1) = fdot(2,1) - z(c) * (1-K(1,c)/K(2,c)) * ...
            (-xi-(1-xi)*K(1,c)/K(2,c)+K(1,c))/...
            (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+...
            psi*K(1,c))^2;
    end
    %sum of fdot(2,2), fdot(2,2)=d(f(2))/d(xi)
    fdot(2,2)=0;
    for c=1:C
        fdot(2,2) = fdot(2,2) - z(c) * (1-K(1,c)/K(2,c)) * \dots
            ((1-psi)-(1-psi)*K(1,c)/K(2,c))/...
            (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+...
            psi*K(1,c))^2;
    end
    %Newton Raphson equation for two equations and two unknown
    psi=psi-(f(1)*fdot(2,2)-f(2)*fdot(1,2))/...
        (fdot(1,1)*fdot(2,2)-fdot(1,2)*fdot(2,1));
    xi=xi-(fdot(1,1)*f(2)-fdot(2,1)*f(1))/...
        (fdot(1,1)*fdot(2,2)-fdot(1,2)*fdot(2,1));
    if psi<0 || psi >1 || xi<0 || xi>1
        disp('new initial values for xi or psi')
        fprintf('Vapor-Liquid split (psi)
                                            %.4f\n',psi);
        fprintf('Oil-Water split
                                    (xi)
                                            %.4f\n',xi);
        return
    end
    error=abs(f(1))+abs(f(2));
   mb=mb+1;
    if mb>=MB
        disp('max no. iterations for loop b')
        %return
    end
end
8----
        Composition of the phases
                                      ____
%Composition of oil
for c=1:C
    x(1,c) = z(c) / ...
        (xi*(1-psi)+(1-xi)*(1-psi)*K(1,c)/K(2,c)+psi*K(1,c));
end
%Composition of gas
for c=1:C
    y(c) = K(1,c) * x(1,c);
end
%Composition of water
for c=1:C
    x(2,c) = y(c) / K(2,c);
end
      a and b in Peng-Robinson
8----
                                     ____
for c=1:C
    kappa=0.37464+1.54226*omega(c)-0.26992*omega(c)^2;
    alfa=(1+kappa*(1-sqrt(T/Tc(c))))^2;
    a(c,c)=0.45724*R^2*Tc(c)^2/pc(c)*alfa;
    b(c)=0.07780*R*Tc(c)/pc(c);
end
```

```
for c=2:C
    for n=1:(c-1)
        a(c,n)=sqrt(a(c,c)*a(n,n));
        a(n,c)=a(c,n);
    end
end
        Fugacity coefficients in gas
8----
                                          ____
%a and b for mixtures in gas
am=0; %a for mixture
bm=0; %b for mixture
for c=1:C
    for n=1:C
        am=am+y(c)*y(n)*a(c,n);
    end
    bm=bm+y(c)*b(c);
end
%A and B for cubic equation
A=am*p/(R*T)^2;
B=bm*p/(R*T);
%Peng-Robinson cubic equation
Z=1; %initial guess
error=1;
mc=0;
MC=100;
while error > 1e-6 && mc<MC
    g(1) = Z^{3} + (-1+B) * Z^{2} + (A-3*B^{2}-2*B) * Z + (-A*B+B^{2}+B^{3});
    gdot(1)=3*Z^2+2*(-1+B)*Z+(A-3*B^2-2*B);
    Z=Z-g(1)/gdot(1);
    error= abs(g(1));
    mc=mc+1;
    if mc>=MC
        disp('max no. iterations for loop c')
        return
    end
end
ZV=Z;
%Fugacity
for c=1:C
    ya=0; %summation of interaction for component C with other
    for n=1:C
        ya=ya+y(n)*a(c,n);
    end
    phi(1,c) = exp(b(c)/bm*(Z-1)-log(Z-bm*p/(R*T))-...
        am/(2*sqrt(2)*bm*R*T)*(2*ya/am-b(c)/bm)*...
        log((Z+(1+sqrt(2))*bm*p/(R*T))/...
        (Z+(1-sqrt(2))*bm*p/(R*T))));
end
8----
        Fugacity coefficients in oil
%a and b for mixtures in oil
am=0; %a for mixture
bm=0; %b for mixture
for c=1:C
    for n=1:C
        am=am+x(1,c)*x(1,n)*a(c,n);
    end
    bm=bm+x(1,c)*b(c);
end
```

```
%A and B for cubic equation
A=am*p/(R*T)^2;
B=bm*p/(R*T);
%Peng-Robinson cubic equation
Z=B; %initial guess
error=1;
md=0;
MD = 100;
while error > 1e-6 && md<MD
    q(2) = Z^{3} + (-1+B) * Z^{2} + (A-3*B^{2}-2*B) * Z + (-A*B+B^{2}+B^{3});
    qdot(2)=3*Z^2+2*(-1+B)*Z+(A-3*B^2-2*B);
    Z=Z-g(2)/gdot(2);
    error= abs(g(2));
    md=md+1;
    if md>=MD
        disp('max no. iterations for loop D')
         return
    end
end
ZO=Z;
%Fugacity
for c=1:C
    xa=0; %summation of interaction for component C with other
    for n=1:C
        xa=xa+x(1,n)*a(c,n);
    end
    phi(2,c) = exp(b(c)/bm*(Z-1)-log(Z-bm*p/(R*T))-...
         am/(2*sqrt(2)*bm*R*T)*(2*xa/am-b(c)/bm)*...
         log((Z+(1+sqrt(2))*bm*p/(R*T))/...
         (Z+(1-sqrt(2))*bm*p/(R*T))));
end
8----
        Fugacity coefficients in water
                                              _ _ _ _
%a and b for mixtures in water
am=0; %a for mixture
bm=0; %b for mixture
for c=1:C
    for n=1:C
         am=am+x(2,c)*x(2,n)*a(c,n);
    end
    bm=bm+x(2,c)*b(c);
end
%A and B for cubic equation
A=am*p/(R*T)^2;
B=bm*p/(R*T);
%Peng-Robinson cubic equation
Z=B; %initial guess
error=1;
me=0;
ME=100;
while error > 1e-6 && me<ME
    g\,(\,3\,)=\!Z^3+(\,-1\!+\!B\,)*Z^2+(\,A\!-\!3*B^2\!-\!2*B\,)*Z\!+(\,-A*B\!+\!B^2\!+\!B^3\,);
    gdot(3)=3*Z^2+2*(-1+B)*Z+(A-3*B^2-2*B);
    Z=Z-g(3)/gdot(3);
```

```
error= abs(g(3));
        me=me+1;
        if me>=ME
            disp('max no. iterations for loop e')
            return
        end
    end
    ZW = Z;
    %Fugacity
    for c=1:C
        xa=0; %summation of interaction for component C with other
        for n=1:C
            xa=xa+x(2,n)*a(c,n);
        end
        phi(3,c) = exp(b(c)/bm*(Z-1)-log(Z-bm*p/(R*T))-...
            am/(2*sqrt(2)*bm*R*T)*(2*xa/am-b(c)/bm)*...
            loq((Z+(1+sqrt(2))*bm*p/(R*T))/...
             (Z+(1-sqrt(2))*bm*p/(R*T))));
    end
    8----
            Update K and check for convergence
                                                      ____
    flag=0;
                 %Clear the flag
    for c=1:C
        if (abs(K(1,c) - phi(2,c)/phi(1,c)) > K(1,c)*1e-3+1e-6)
            flag=1; %more calculations needed
            K(1,c) = phi(2,c) / phi(1,c);
            K(2,c) = phi(3,c) / phi(1,c);
        elseif (abs(K(2,c) - phi(3,c)/phi(1,c)) > K(2,c)*1e-3+1e-6)
            flag=1; %more calculations needed
            K(2,c) = phi(3,c) / phi(1,c);
        else
            K(1,c) = phi(2,c) / phi(1,c);
            K(2,c) = phi(3,c) / phi(1,c);
        end
    end
    ma=ma+1;
    if ma>=MA
        disp('max no. iterations for loop a')
    end
end
V=psi;
L1=xi*(1-psi);
L2=L1/xi-L1;
        Printing of results
8____
print1=1;
print2=1;
print3=1;
if print1==1
    fprintf('VLLE with PR EOS\n');
    fprintf('P= %.2e Pa\n',p);
    fprintf('P= %.2f Barg\n',pflash);
    fprintf('T= \&.2f K \setminus n', T);
    fprintf('T= %.2f C\n',Tflash);
    fprintf('\n');
```

```
fprintf('Vapor-phase %.3f mol\n',V);
    fprintf('Decane-phase %.3f mol\n',L1);
    fprintf('Glycol-phase %.3f mol\n',L2);
    fprintf('\n');
end
if print2==1
    fprintf('
               Total flow [mol%%]\n');
    fprintf('Water
                      %.4f\n',z(1)*100);
    fprintf('TEGlycol
                      %.4f\n',z(2)*100);
                       .4f n', z(3)*100);
    fprintf('Methan
    fprintf('Decane
                       .4f n', z(4)*100);
    fprintf('\n');
    fprintf(' Gas-phase
                                        [mol] n');
                             [mol%%]
    fprintf('Water
                              %.4f %.4f\n',y(1)*100,y(1)*V);
    fprintf('TEGlycol
                              %.4f %.4f\n',y(2)*100,y(2)*V);
    fprintf('Methan
                              %.4f %.4f\n',y(3)*100,y(3)*V);
    fprintf('Decane
                              %.4f %.4f\n',y(4)*100,y(4)*V);
    fprintf('\n');
    fprintf(' Oil-phase
                             [mol%%]
                                        [mol] n');
    fprintf('Water
                              %.4f %.4f\n',x(1,1)*100,x(1,1)*L1);
    fprintf('TEGlycol
                              %.4f %.4f\n',x(1,2)*100,x(1,2)*L1);
    fprintf('Methan
                              %.4f %.4f\n',x(1,3)*100,x(1,3)*L1);
    fprintf('Decane
                              %.4f %.4f\n',x(1,4)*100,x(1,4)*L1);
    fprintf('\n');
    fprintf(' Water-phase
                            [mol%%]
                                        [mol] \ ;
    fprintf('Water
                              .4f \ .4f \ x(2,1)*100, x(2,1)*L2);
    fprintf('TEGlycol
                              %.4f %.4f\n',x(2,2)*100,x(2,2)*L2);
    fprintf('Methan
                             %.4f %.4f\n',x(2,3)*100,x(2,3)*L2);
    fprintf('Decane
                              %.4f %.4f\n',x(2,4)*100,x(2,4)*L2);
    fprintf('\n');
end
if print1==1 && print3==1
    fprintf('Vapor-Liquid split (psi)
                                       %.3f\n',psi);
    fprintf('Liquid-Liquid split (xi)
                                      %.3f\n',xi);
    fprintf(' n');
    fprintf(' K-values K1
                                       K2 n')
    fprintf('Water
                        %.3e
                                %.3e\n',K(1,1),K(2,1));
                                %.3e\n',K(1,2),K(2,2));
    fprintf('TEGlycol
                        %.3e
    fprintf('Methan
                                %.3e\n',K(1,3),K(2,3));
                        %.3e
                                %.3e\n',K(1,4),K(2,4));
    fprintf('Decane
                         %.3e
    fprintf('\n');
```

end

Appendix 3: HYSYS PFD

