NTNU
Norwegian University of
Science and Technology

Master's thesis

2020

Inger Anna Helmersen

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Inger Anna Helmersen

Simulation, optimization and mathematical modelling of a hybrid membrane system for natural gas dehydration and CO_2 removal

July 2020

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Simulation, optimization and mathematical modelling of a hybrid membrane system for natural gas dehydration and CO_2 removal

Inger Anna Helmersen

Chemical Engineering and Biotechnology
Submission date: July 2020
Supervisor: Magne Hillestad, IKP
Co-supervisor: Xuezhong He, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Abstract

Membranes for natural gas treatment are gaining increasing acceptance with the increasing natural gas demand and stricter environmental regulations. In this work a hybrid membrane system for natural gas dehydration and CO_2 removal was studied. A Spiral-wound MTR Pebax® co-polymer based composite membrane for dehydration was simulated, and a hollow fiber polyimide derived carbon membrane was simulated in series for removal of CO_2. The aim was to evaluate the technical end economical feasibility with respect to pipeline and sales specifications and specific cost of natural gas processing. Furthermore a membrane model was developed, compared to experimental data and analyzed for use in the process simulation to predict the membrane performance. Various design configurations were proposed with respect to the number and arrangement of compressors. Each design configuration was optimized using the Nelder-Mead simplex algorithm with a penalty function. The optimization routine was carried out externally by creating a COM automation server (ActiveX) to connect with the simulation interface. The set of membrane model equations were dicretized using the orthogonal collocation method to form a set of non-linear algebraic equations which was solved applying the Newton-Rhapson method. The membrane model was then integrated in the simulation environment with the optimal design configuration as a Cape-Open unit operation based on the COM technology.

The optimal design configuration contained two compressors in series between the dehydration unit and the sweetening unit, with compression of the recycled retentate from the sweetening unit, and a pressure relief valve instead of a turbine for expanding the sweetening unit permeate before entering the dehydration unit as sweep gas. The specific natural gas processing cost for the optimal design was $2.192 \cdot 10^{-3}$ $/m^3, and the pipeline and sales specifications were met. For the optimal design with the ChemBrane model the water dew point was $-52.5\,°C$, the methane purity was 97.5%, the CO_2 purity in the gas for enhanced oil or gas recovery was 95%, and the gas dew point was $-11.5\,°C$. The design with the model developed in this work was not optimized due to connection issues between the COM automation server and the simulation case. The model was found suitable for gas transport through hollow fiber membranes by one of two experiments with a relative deviation $\leq 3.65\%$. The other experiment deemed the model invalid due to large deviations in the permeate flow rate and permeate purity. Compared to the ChemBrane model, which was initially applied in the process design, the model showed similar but more realistic performance.

In conclusion, the study shows that it is technically and economically feasible to use membranes for gas treatment, but the dehydration unit must be improved in terms of design, module configuration or membrane performance. Testing of the membranes in plant scale is necessary to determine the membrane lifetime and for further model validation. Investigation of the COM-technology and Cape-Open unit operation in the simulation environment is necessary for external optimization of user-defined models. Lastly, a detailed economical evaluation including an economic analysis and a sensitivity analysis is required to improve the cost estimations.

Sammendrag

Membraner aksepteres i økende grad som metode for naturgassrensing ettersom etterspørselen etter naturgass øker og strengere klima- og utslippsregler innføres. I denne rapporten, har et hybrid membransystem for fjerning av vann og CO_2 i naturgass blitt studert. En spiralmodul med en MTR Pebax® co-polymerbasert sammensatt membran ble simulert for dehydrering, og to fibermoduler med polyimid-derivert karbonmembran ble simulert i serie for fjerning av CO_2. Målet var å evaluere kost-nytte-effekten som et forstudie for systemet. Videre ble en membranmodell utviklet, sammenlignet med eksperimentelle data og analysert for bruk i simuleringsverktøy for å forutsi prestasjonen til membraner for gasseparasjon. Et utvalg av ulike designkonfigurasjoner ble vurdert med hensyn til antallet og utformingen av kompressorer. Hver designkonfigurasjon ble optimalisert ved bruk av Nelder-Mead-metoden med en penalty-funksjon. Optimaliseringsprogrammet ble utført eksternt ved å opprette en COM automatasjonsserver (ActiveX) som koblet det eksterne programmet til brukergrensesnittet for simuleringsprogrammet. Likningssettet for membranmodellen ble diskretisert ved å bruke en ortogonal kollokasjonsmetode som transformerte likningene til ikke-lineære algebraiske likninger som videre ble løst ved bruk av Newton-Rhapson-metoden. Membranmodellen ble deretter integrert i simuleringsprogrammet med den optimale designkonfigurasjon som en Cape-Open enhetsoperasjon basert på COM-teknologi.

Det optimale designet besto av to kompressorer i serie mellom enheten for dehydrering og enheten for CO_2-fjerning, med kompresjon av den resirkulerte retentatstrømmen fra CO_2-enheten, og en trykkventil for å redusere trykket i den resirkulerte permeatstrømmen fra CO_2-enheten. Den spesifikke kostnaden for renset naturgass var $2.192 \cdot 10^{-3}$ $/m^3 og kravene til gassens sammensetning og utslipp ble innfridd. I det optimale designet med ChemBrane-modellen var duggpunktet til vann $-52.5\,°C$, den rensede naturgassen besto av 97.5% metan og det var 95% CO_2 i strømmen for videre bruk i EOR. Gassens duggpunkt var $-11.5\,°C$. Det optimale systemdesignet med membranmodellen utviklet i dette arbeidet ble ikke optimalisert på grunn av tilkoblingsproblemer med COM-serveren og simuleringsprogrammet. Membranmodellen ble validert av ett av to eksperimenter med et relativt avvik $\leq 3.65\%$, og kunne derfor brukes til simulering av gasseparasjon i fibermoduler. Sammenlignet med data fra det andre eksperimentet, ble membranmodellen funnet å være ugyldig på grunn av store relative avvik i permeatstrømmene og renheten. Membranmodellen viste derimot tilnærmet like resultater som ChemBrane-modellen, men ble vurdert å vise mer realistisk gasseparasjon i fibermoduler.

Studiet viser at membraner for naturgassresning er både teknisk og økonomisk mulig, men dehydreringsenheten må forbedres med hensyn til design, modulkonfigurasjon eller membranytelse. Testing av membranene i anleggsskala er nødvendig for å bestemme levetiden og for videre validering av membranmodellen. COM-teknologien og Cape-Open enhetsoperasjon i simuleringsprogrammet må utforskes for ekstern optimalisering av bruker-definerte modeller. Til slutt, en detaljert evaluering av kostnaden, inkludert en økonomisk analyse og en sensitivitetsanalyse, er nødvendig for å forbedre kostnadsberegningene.

Acknowledgements

This thesis is submitted as part of a Master's degree in collaboration with the Environmental Engineering and Reactor Technology Group at the Department of Chemical Engineering, NTNU. The work presented in this thesis is a continuation of the specialization project during the fall of 2019.

I would like to thank my family for supporting me and rooting for me through all my years as a student. I am particularly grateful for the support and care that my partner, Jørgen, and my sister, Signe Eva, have given me, and for their feedback on my thesis. I would like to thank my supervisor, Prof. Magne Hillestad, for giving me the opportunity to work on an exiting and important project, and for his valuable guidance throughout the project.

Declaration of Compliance

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

Inger A. Hel

Inger Anna Helmersen
Trondheim
July 13, 2020

Table of Contents

List of Figures

List of Tables

Abbreviations

CA	Cellulose acetate
Cape-Open	Computer Aided Process Engineering-Open
COM	Component Object Model
CPU	Central Processing Unit
DFM	Derivate-free method
EGR	Enhanced gas recovery
EOR	Enhanced oil recovery
FDM	Finite difference method
FEM	Finite element method
FVM	Finite volume method
LP	Linear programming
modEBJ	Extension of the initial membrane model
modEBJQ	Further extension of modEBJ, including heat of conduction
modMB	Initial membrane model
MTR	Membrane Technology Research
NG	Natural gas
Pebax®	polyether block amide
PEEK	Poly(ether ether ketone)
PI	Polyimide

PSF	Polysulfone
RD	Relative Deviation
RGM	Reduced gradient method
SLP	Successive linear programming
SNG	Sweet natural gas
SQP	Successive quadratic programming
STP	Standard Temperature and Pressure

Chapter 1

Introduction

Natural gas is one of the world's most important sources of energy and accounts for almost a quarter of the global energy consumption [2, 3]. The composition depends on the location for which the natural gas is extracted, but it mainly consist of methane, lighter hydrocarbons and some heavier hydrocarbons. Natural gas also contains numerous contaminants and undesired components such as water, CO_2 and H_2S. These components can lead to pressure drop in the distribution pipelines, reduced gas flow, and plugging of the pipelines along with corrosion [4, 5, 6, 7, 8, 9]. Consequently, natural gas needs to be treated by dehydration and CO_2 removal processes to meet pipeline and sales specifications and to comply with environmental regulations. Current methods for natural gas treatment are mostly conventional methods such as glycol dehydration and amine treatment. These methods require large facility areas and are highly energy intensive and therefore involve high operating costs. Furthermore, the current methods pose a concern due to increasing environmental constraints. Therefore, membranes for natural gas treatment is gaining acceptance.

1.1 Motivation

Methane loss in glycol dehydration can be as low as 1% [10]. However, the technology is not compatible for offshore application as the open flame in the desorption column poses a safety threat. The system operation is complex, and solvent storage replacement and disposal is required. Glycols absorb volatile organic compounds from the natural gas streams, particularly benzene, ethylbenzene and xylenes which can be emitted in the regeneration unit. This is becoming of increasing concern with respect to environmental regulations. For amine treatment, emission of amine based pollutants and loss of solvent due to high regeneration temperatures, corrosion problems in the operating units, and struggles achieving low dew points, are common challenges.

Membranes for gas separation are commercially available , and studies report promising performances of water-selective composite membranes for dehydration of natural gas and polymer derived carbon membranes for CO_2 removal. Some advantages related to membrane gas separations are the ease of installation and operation, there is no need for extra agents for separation, low energy demand, continuous mode of operation with partial or complete recycle, and possibility for integration with other separation units. While glycol dehydration and amine treatment requires large amounts of thermal and electrical energy as well as taking up large facility areas, membranes are cost-efficient, small and compact, and easy to scale up. As a result there is a high potential for application on-shore and off-shore, and it is especially promising for subsea natural gas processing compared to the conventional technology.

1.2 Objective

In this thesis a hybrid membrane system for dehydration and CO_2 removal in natural gas is simulated and studied. The performance of field-tested spiral-wound MTR Pebax® co-polymer based membranes, as reported by Lin et al. [11], is simulated for dehydration of natural gas. Hollow fiber carbon membranes made from polyimide precursors, as reported by Zhang et al. [12], is simulated for CO_2 removal. The aim is to evaluate the technical and economical feasibility of using membranes for natural gas treatment. This is accomplished by investigating the membrane performance with respect to pipeline and sales specifications, and evaluating the cost of the system design. As such, the process design is optimized with respect to the specific natural gas processing cost, and a membrane model is developed for use in the process simulation to predict the membrane performance.

1.3 Thesis structure

This report consists of 7 chapters and 5 appendices.

Chapter 1
Gives a brief introduction to the background, motivation and objective for the work presented in this report.

Chapter 2
Gives theoretical background for membrane gas separation, compressor op-

eration, numerical approach for solving differential equations, optimization methods and specifications on membranes in natural gas treatment.

Chapter 3
Presents a description of the hybrid membrane system, the simulation basis and the cost model.

Chapter 4
Describes the optimization problem and routine for solving the problem.

Chapter 5
Presents the governing equation for the membrane model and the numerical approach for solving the set of governing equations.

Chapter 6
The results from the optimization and the simulation of the membrane model is presented and discussed.

Chapter 7
Conclusion of the work in this thesis is presented followed by recommendations for future work.

Appendices
The initial design configuration and results from previous work are presented here. Detailed derivation of the membrane equations, an example of the software framework for the optimization, and code for the membrane model and optimization routine is presented.

Chapter 2

Preliminaries

This section will provide the theoretical background applied in this thesis. First some numerical methods for solving a set of differential equations are presented with emphasis on collocation methods relevant for solving mathematical models of physical systems. This is followed by methods for optimization, aspects on compressor operations, and principles of membrane gas separation such as membrane material, structure, flow pattern and configuration. Finally the specifications for natural gas treatment and membranes with promising performances are presented.

2.1 Numerical methods for solving differential equations

Modelling a physical system often results in a set of differential equations for which a solution can be obtained by many different numerical methods and schemes. However, the performance of the various methods differs in terms of consistency, stability and convergence. Consequently, the accuracy and computational requirement differ for the different methods. Some allow for high accuracy, but at a high computational cost, whereas for some methods it is the opposite, and some handle stiff systems poorly due to large gradients. For some systems the numerical accuracy can be critical, and for others the computational efficiency is of the essence. Therefore the numerical method to be chosen is dependent on the system at hand.

The principle of any numerical method is to transform the set of governing equations into a system of algebraic equations that can be solved iteratively [13].

$$Af = b \tag{2.1}$$

In general, any problem involving a system of differential equations can be presented as

$$\mathcal{L}f(z) = g(z), \quad \forall z \in \Omega \tag{2.2}$$

$$\mathcal{B}f(z) = f_{\Gamma}, \quad \forall z \in \Gamma \tag{2.3}$$

where $\mathcal{L}$ is a linear differential operator acting on the solution function f to yield g in the domain Ω, and $\mathcal{B}$ is the boundary operator acting on f in the domain Γ. Some methods for solving such a system are the finite difference method (FDM), finite element method (FEM) and finite volume method (FVM) which will be mentioned briefly, and weighted residual methods of which the orthogonal collocation method will be explained in more detail.

2.1.1 FDM, FEM and FVM

FDM is one of the methods that approximates the solution of differential equations [14]. The starting point is the differential form of the transport equation [13]. A finite set of discrete points is defined on a grid which is often chosen to be uniform. The differential equations are approximated in terms of the nodal values of the functions using for example Taylor series expansion. It is suited for low order approximations of the derivatives. Non-uniform grids pose a challenge and yield large errors, which makes it less suitable for complex geometries. FVM uses the integral form of the conservation equations as its starting point. The solution domain is divided into sub-regions referred to as cells. Quadrature formulas are chosen to approximate surface and volume integrals, and it is suitable for complex geometries. FEM is similar to the FVM in that it divides the domain into sub-regions referred to as elements. The solution is approximated by piece-wise linear functions in each element. Often polynomial functions are used to approximate the solution. The accuracy of the solution increases with increasing number of elements.

According to Jakobsen [13], the drawbacks of FDM and FVM are low convergence rate and low accuracy. FDM is further restricted to simple geometries. FVM is suitable for complex geometries, but it requires two levels of approximations. The drawback with FEM is the computational cost for achieving accuracy due to the amount of algebraic equations. A collocation method finds an optimal distribution of nodes and is suitable for higher order approximations.

2.1.2 Orthogonal collocation - A weighted residual method

Generally, the approach in any weighted residual method is to approximate $f(z)$ in Equations (2.5) and (2.6) by a trial function expansion $\tilde{f}(z)$ which is a linear combination of the basis coefficients a_i and basis functions ϕ_i [13].

$$f(z) \approx \tilde{f}(z; a_1, a_2, ..., a_N) = \sum_{i=1}^{N} a_i \phi_i(z) \tag{2.4}$$

where N is the number of trial functions used. The result of inserting the approximation of f into Equations (2.5) and (2.6) will never be identically $g(z)$, which represents the exact solution, but gives a residual error

$$\mathcal{L}\tilde{f}(z) - g(z) = R, \quad \forall z \in \Omega \tag{2.5}$$

$$\mathcal{B}\tilde{f}(z) - f_\Gamma = R, \quad \forall z \in \Gamma \tag{2.6}$$

In order to satisfy the boundary conditions, the unknown constants, a_i, must be determined by forcing the residual to zero over the domain

$$\int_\Omega W_i(z) R(z; a_1, a_2, ..., a_N) d\Omega + \int_\Gamma W_i(z) R(z; a_1, a_2, ..., a_N) d\Gamma = 0 \tag{2.7}$$

where W_i are the $i = 1, 2, ..., N$ weight functions. This gives a set of N algebraic equations for the unknown constants a_i.

The type of weighted residuals method depend on the choice of weight function W_i. For the orthogonal collocation method, the weight functions are translated Dirac delta functions which have the following quality

$$\int_a^b \delta(z - z_i) dz = 1 \tag{2.8}$$

Thus, Equation (2.7) becomes

$$\int_\Omega R(z; a_1, a_2, ..., a_N) \delta(x - x_i) d\Omega = 0 \quad i = 1, 2, ..., N \tag{2.9}$$

which results in driving the residuals to zero at N collocation points $z_1, z_2, ..., z_N$.

$$R(z; A) = 0 \tag{2.10}$$

Orthogonal collocation uses nodal basis functions in the solution function expansion Equation (2.4), thus the solution function is approximated at the collocation points [13]. Nodal basis functions are defined as the Lagrange polynomials

$$l_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{x - x_k}{x_i - x_k} \tag{2.11}$$

which is of order N and interpolating f in $N+1$ points. The polynomials meet the following condition

$$l_i(x) = \begin{cases} 1, & \text{for } i = k \\ 0, & \text{for } i \neq k \end{cases} \tag{2.12}$$

As a result, the basis coefficients, a_i, are identified as the solution function values themselves at the nodes. Inserting this into Equation (2.4) and differentiating, gives

$$\frac{df(x)}{dx} \approx \frac{d}{dx}\sum_{i=0}^{N} f(x_i) l_i(x) = \sum_{i=0}^{N} f(x_i)\frac{dl_i(x)}{dx} \tag{2.13}$$

The collocation points are commonly selected as the roots of one of the orthogonal polynomials in the family of Jacobi polynomials, $P_n^{(\alpha,\beta)(x)}$ [15]. For orthogonal polynomials, p_n and p_m, the following is true

$$\int_a^b W(x) p_n(x) p_m(x) dx = 0, \quad n \neq m \tag{2.14}$$

and the weight function for Jacobi polynomials is defined as

$$W(x) = (1-x)^{\alpha}(1+x)^{\beta}, \quad x = [-1, 1], \quad (\alpha, \beta) > -1 \tag{2.15}$$

where α and β are considered parameters for shifting the collocation points towards one of the endpoints. Depending on the polynomial, three different grids of collocation points are produced: Gauss, Gauss-Lobatto, and Gauss-Radau, shown in Figure 2.1. Gauss grid includes only the interior nodes, Gauss- Radau includes in addition one of the endpoints, and Gauss-Lobatto includes both endpoints in addition to the interior points. Therefore, in a two-point boundary value problem, the Gauss-Lobatto grid should be applied.

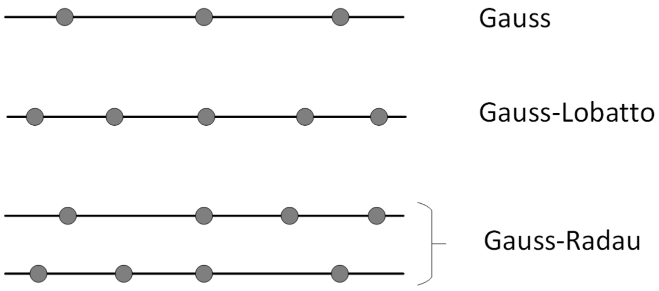


Figure 2.1: Gauss, Gauss-Lobatto, and Gauss-Radau grids. The Gauss grid consists of the interior nodes, while the Gauss-Lobatto grid includes the boundary points and the interior nodes, and the Gauss-Radau grid includes the interior nodes and one of the boundary points.

Moreover, all orthogonal polynomials satisfy a three term recurrence relationship

$$\begin{aligned} p_1(x) &= 0 \\ p_0(x) &= 1 \\ p_{i+1}(x) &= (a_i x + b_i)p_i(x) - c_i p_{i-1}(x), \quad -1 < i \in \mathbb{Z} \end{aligned} \tag{2.16}$$

where a_i, b_i and c_i are coefficients given by

$$a_i = \frac{(2i+\alpha+\beta+1)(2i+\alpha+\beta+2)}{2(i+1)(i+\alpha+\beta+1)} \tag{2.17}$$

$$b_i = \frac{(2i+\alpha+\beta+1)(\alpha^2-\beta^2)}{2(i+1)(i+\alpha+\beta+1)(2i+\alpha+\beta)} \tag{2.18}$$

$$c_i = \frac{(i+\alpha)(i+\beta)(2i+\alpha+\beta+2)}{(i+1)(i+\alpha+\beta+1)(2i+\alpha+\beta)} \tag{2.19}$$

which can be useful in estimating the Jacobi polynomial at the abscissa of a point in the interval [16]. A special case of the Jacobi polynomials are the Legendre polynomials where $\alpha = \beta = 0$ and the weight function is equal to unity, $W(x) = 1$ [17]. The Legendre polynomials simplifies the recurrence coefficients to

$$a_i = \frac{2i+1}{i+1} \tag{2.20}$$

$$b_i = 0 \tag{2.21}$$

$$c_i = \frac{i}{i+1} \tag{2.22}$$

There are several methods for calculating the zeros of Jacobi polynomials, e.g. an eigenvalue method using the three-term recurrence relation called Golub-Welsch algorithm [18], or other root finding algorithms such as the Newton-Rhapson method also in combination with the three-term recurrence relation.

2.2 Optimization

In optimization problems, the aim is to find the optimal solution to a problem [19]. The optimal solution is either a maximum or minimum of a quantity referred to as the objective. The need for optimization is found within many fields, and there are numerous optimization problems with various objectives to be solved, e.g. within chemical engineering it is important to optimize a process design for the customer. In many cases the objective is a measure of economic performance for which the aim is to minimize costs or maximize profits.

The objective is a function of a finite set of decision variables, and is therefore called the objective function. In the case of constraints, the constraints are a set of equations which bound the solution space. The optimization problem can be stated as

$$\begin{aligned} z &= f(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) &\leq 0 \\ \mathbf{h}(\mathbf{x}) &= 0 \end{aligned} \tag{2.23}$$

where z and $\mathbf{x}$ are the scalar objective and a vector of n decision variables, respectively, and f, $\mathbf{g}$ and $\mathbf{h}$ are the objective function, vector of inequality and equality constraints respectively [19].

There are several methods for solving an optimization problem for which a suitable method depends on the number of decision variables, whether the problem is linear or non-linear, convex and continuous. Additionally, the effort of solving the problem is important in choosing an optimization method. Some methods require numerous computations for certain problems, while others are efficient and have low computational cost. Generally, there is a trade-off between the solution quality and effort, as with increasing effort the solution quality increases [20].

If the optimization problem is linear, it is solved by linear programming which is widely used in operations such as optimization of petrochemical plants and oil refineries [19]. In the case of design problems, non-linear programming is applied to solve the problem as it generally contains many non-linear equations.

2.2.1 Linear Programming and the Simplex method

Linear programming (LP) refers to optimization methods that solve a linear problem. According to Rothlauf [20], a problem is linear if

- the objective function depends linearly on the decision variables and
- all relations among the variables are linear.

A linear problem can be stated as

$$f(x) = \sum_{i=1}^{n} c_i x_i, \quad x \in \mathbb{R}^n \tag{2.24}$$

If the problem also involves linear constraints, a linear minimization problem can be expressed in the following form

$$\begin{aligned} \min\ c^T x \\ \text{subject to } Ax \geq b \\ x_i \geq 0 \end{aligned} \tag{2.25}$$

A maximization problem can be obtained by multiplying the objective function in Equation (2.27) with −1. A global optimum is obtained with linear programming because the constraints always define a convex feasible region. If two points are within the feasible region, any point on a straight line between those two points is also within the feasible region. Thus the feasible region is convex. According to Sinnot et al. [19], a mathematical formulation of convexity can be stated as follows

$$\begin{aligned} x = \alpha x_a + (1-\epsilon)x_b \in F \\ \forall x_a, x_b \in F, \quad 0 < \epsilon < 1 \end{aligned} \tag{2.26}$$

where x_a, x_b are any two points within the feasible region, F. The feasible, convex region forms a simplex which is the simplest possible polytope in a space of size n. A convex polytope can be defined either as the convex hull of a feasible region, or as the intersection of a finite set of half-spaces.

Among the methods for solving linear problems, the Simplex method developed by Dantzig in 1946, is widely used [21]. The method systematically examine the vertices of the feasible region to determine the optimum. The Simplex method transforms the inequality constraints to equality constraints by introducing slack and surplus variables. Thus, the canonical form in Equation (2.27) is transformed to standard form.

$$\begin{aligned} \max\ c^T x \\ \text{subject to } Ax = b \\ x_i \geq 0 \end{aligned} \tag{2.27}$$

The set of equalities is solved to obtain a feasible solution. Some of the slack and surplus variables will be zero which corresponds to active constraint. An active constraint is one that has been converted from an inequality constraint to an equality constraint because the optimum might be at the boundary [19]. As such, the problem is made simpler because the degrees of freedom is reduced. The Simplex algorithm searches the vertices of the feasible region, increasing the objective at each step until the optimum is reached.

Linear programming can be used to solve very large problems, with thousands of variables and constraints. However, when the objective function and/or the constraints are non-linear, the optimization problem must be solved with non-linear programming (NLP) [19].

2.2.2 Non-linear programming

Non-linear programming refers to optimization methods that solve non-linear optimization problems. According to Sinnot et al., some common methods are successive linear programming (SLP), successive quadratic programming (SQP) and reduced gradient method (RGM) [19].

In short, SLP is an extension of LP as the objective function and the constraints are linearized and solved as in LP iteratively until convergence. SQP is similar to SLP, but the objective function is approximated as a quadratic function and quadratic programming methods are applied to solve the problem. It works well for highly non-linear problems with relatively few variables (n < 50). RGM is related to the Simplex method as it linearizes the constraints and introduces slack and surplus variables. The vector of n decision variables is partitioned into $n - m$ independent variables, where m is the number of constraints, and a quasi-Newton method is used to determine a solution. According to Sinnot et al. [19], it is suitable for sparse problems with a large number of variables, i.e. each constraint involves only a few of the variables. Common for all methods, is that there is no guarantee of convexity, convergence to local optimum or global optimum. Additionally, all the methods are gradient-based. Methods for finding the global optimum are pattern searches and probabilistic methods such as simulated annealing and genetic algorithms. Probabilistic method are useful when faced with a non-convex feasible region. Moreover, pattern-search methods and simulated-annealing are derivate-free methods (DFM), which means that these methods do not apply or even try to approximate the gradient, but use the sampled function values to determine the new iterate. Another DFM which is widely used is the Nelder-Mead method [22].

2.2.3 Nelder-Mead method

The Nelder–Mead simplex method is a popular DFM for solving unconstrained minimization problems [22]. As mentioned, it uses only the function values to determine new iterates and does not attempt to approximate the gradients. Hence it is a direct search method [23]. At any stage of the algorithm it keeps track of $n + 1$ points of interest in $\mathbb{R}^n$, whose convex hull forms a simplex, hence the name. It is not related to the Simplex method for linear problems described in Section 2.2.1. Consider a simplex S with vertices $x_1, ..., x_{n+1}$, an associated matrix $Y(S)$ can then be defined by taking the n edges along Y from one of its vertices e.g. x_1, as follows

$$Y(S) = [x_2 - x_1, ..., x_{n+1} - x_1] \tag{2.28}$$

The initial working simplex has to be non-degenerate or non-singular, which means that Y must be a non-singular matrix, i.e. the vertices must not be co-planar [22].

The aim of the Nelder–Mead simplex algorithm in one iteration, is to eliminate the vertex with the worst function value and replace it with another point with a better value. The new point is obtained by reflecting, expanding, or contracting the simplex along the line joining the worst vertex with the centroid of the remaining vertices. If a better point is not found, the vertex with the best function value is retained, and the simplex is shrunk by moving all other vertices toward this value. A pseudo-code of the the algorithm is provided by Nocedal [22].

Little is known about the convergence properties in the Nelder-Mead method, and there is no guarantee of reaching a global optimum. However, the algorithm typically requires only one or two function evaluations at each step, in contrast to many other direct search methods [23]. Consequently, the Nelder-Mead method is faster than other methods, and is a popular method to use for non-linear optimization problems [22].

2.2.4 Penalty method

To use the Nelder-Mead simplex method or any other optimization methods for unconstrained problems, on a constrained minimization problem, a penalty method is introduced [22]. It provides a measure of feasibility in the objective with respect to the constraints. A term called the penalty function, is added to the objective function and consists of a scalar penalty parameter and a measure of violation of the constraints. Thus the objective function in Equation (2.23) becomes

$$\min\, z = f(\mathbf{x}) + \mu \sum_i g(c_i(\mathbf{x})) \tag{2.29}$$

where f and c are the objective function and the constraints of $\mathbf{x}$ decision variables, respectively. μ is the penalty constant, usually taking a large value, and g is the penalty function, e.g. the quadratic penalty function defined as

$$g(c_i(\mathbf{x})) = \max(0, c_i(\mathbf{x}))^2 \tag{2.30}$$

2.3 Compression and expansion of gases

Derivation of the compressor and turbine work is given by Çengel et al. [24] and presented in the specialization project [25] as in the following section. For isentropic, polytropic and isothermal processes between the same pressure levels P_1 and P_2 with ideal gas behaviour, the compressor and turbine work is defined as

Isothermal (PV= constant):

$$W_{comp} = RTln\frac{P_2}{P_1} \tag{2.31}$$

Isentropic (PV^k = constant):

$$W_{comp} = \frac{kRT_1}{k-1}\left[\left(\frac{P_2}{P_1}\right)^{\frac{(k-1)}{k}} - 1\right] \tag{2.32}$$

Polytropic (PV^n = constant):

$$W_{comp} = \frac{nRT_1}{n-1}\left[\left(\frac{P_2}{P_1}\right)^{\frac{(n-1)}{n}} - 1\right] \tag{2.33}$$

where W$_{\mathrm{comp}}$ is the compression work, R, T, P, V is the universal gas constant, the fluid temperature, pressure and volume, respectively. k and n are the isentropic and polytropic exponents, respectively, and T_1 is the compressor inlet temperature. Of the three processes, the isentropic process (adiabatic compression) requires the maximum work, while the isothermal compression requires the minimum, and the polytropic work requirement is in between. By removing heat, the polytropic work decreases as n decreases, and at sufficient heat removal the value of n approaches 1 whereby the polytropic process becomes an isothermal process. Consequently, cooling a gas as it is compressed, reduces the required work for compression.

Cooling through the casing of the compressor is rarely sufficient, and it is therefore suggested to use multistage compression with intercooling, where the gas is compressed in stages, and cooled in between. Ideally, the cooling takes place at constant pressure and the gas is cooled to the initial temperature, T_1. However, in reality intercoolers cause pressure losses often in the range 5-15 psi (0.34-1.03 bar), in which a value of 10 psi (0.7 bar) can be used for preliminary design [26]. For multistage compression, the amount of saved work varies with the value of the intermediate pressures. The dependency is presented by Çengel et al. [24] for a two stage compression with intercooling. The intermediate pressure, P_x, that minimizes the total work is determined by differentiating the equation for the total amount of work with respect to P_x, which yields

$$\frac{P_x}{P_1} = \frac{P_2}{P_x} \tag{2.34}$$

$$\frac{P_x}{P_1} = \left(\frac{P_2}{P_1}\right)^{1/2} \tag{2.35}$$

where P_1, P_2 and P_x are the suction, discharge and intermediate pressures, respectively. In other words, the pressure ratio across each stage must be the same to minimize the compression work, and as a result, the compression work at each stage becomes the same. Moreover, as the number of compressor stages is increased, the compressor work approaches the value obtained for the isothermal case.
The interstage pressure ratio shown in Equation (2.35) can be rewritten for a general multistage compression system with n stages [25]

$$\frac{P_i}{P_{i-1}} = \left(\frac{P_n}{P_{i-1}}\right)^{1/n} \qquad i = 1, ..., n \tag{2.36}$$

Compressors, turbines and nozzles operate for long periods of time under the same conditions and steady operation is established and is therefore classified as steady-flow devices. The energy balance for a general steady-flow system

$$\dot{Q}_{in} + \dot{W}_{in} + \sum_{in} \dot{m}\theta = \dot{Q}_{out} + \dot{W}_{out} + \sum_{out} \dot{m}\theta \tag{2.37}$$

where $\dot{Q}_{in}$ and $\dot{Q}_{out}$ are the rate of heat transfer in and out of the system, $\dot{W}_{in}$ and $\dot{W}_{out}$ are the rate of work exerted in and out of the system, $\dot{m}$ is the mass flow rate and θ is the energy of a flowing fluid per unit mass

$$\theta = h + ke + pe = h + \frac{\nu^2}{2} + gz \tag{2.38}$$

where h is the enthalpy, ν, g, z, ke and pe are the fluid velocity, gravity, elevation of the system relative to some external reference point, kinetic energy and potential energy, respectively. For a single-stream device it can be assumed negligible changes in the fluid kinetic and potential energies. Moreover, for a system with unknown heat and work interactions, the general energy balance in Equation (2.37) can be written as

$$\dot{W} = \dot{m}(h_2 - h_1) \tag{2.39}$$

where heat transfer from the system is negligible for an adiabatic process, $\dot{Q} = 0$. $\dot{W}$ and $\dot{m}$ are the rate of work and mass flow into the system, respectively, and h_2 and h_1 are the discharge and suction enthalpy, respectively.

2.3.1 Compressor types

For gas transmissions it is common to use reciprocating or centrifugal compressors [26]. Reciprocating compressors are ideal for high pressure ratios, and low volume flow. However, it commonly generates pulsations and vibration which is avoided in centrifugal compressors, which in contrast are ideal for high volume flow and low head. The pressure ratio that a given centrifugal compressor can achieve depends primarily on gas composition and gas temperature. For reciprocating compressors the maximum pressure ratio of each stage is 10, and for centrifugal compressors it is in the range 3.5-4, but multistage compression is recommended if the pressure ratio exceeds a value of 3 [27, 26].

2.4 Basic principles of membrane gas separation

Although membrane technology is found in most industrial areas such as pharmaceutical, textile, biotechnology and chemical industry to mention a few, membrane technology is considered to be a new technology especially for separation processes [28]. Consequently, gas separation processes are currently dominated by conventional technology such as solvent absorption, cryogenic distillation and solid surface adsorption [29]. However, membranes are acclaimed for removal of acidic gases such as H_2S and CO_2 in which commercial membranes for CO_2 removal has been available since the 1980s [30]. Nevertheless, membranes for gas separation make up only a small part of the total market of which the majority are polymeric membranes [31]. Inorganic membranes with greater separation performance have been developed but are too expensive for commercial applications, whereas the current polymeric membranes are easily manufactured at low costs. Studies show that membranes are gaining increasing acceptance and finding new applications for gas separation [32]. This section describes the basic principles of membrane gas separation and membrane modelling.

Membrane separation occurs due to a difference in the chemical potential at the membrane surfaces [28]. In gas separation that difference is a result of the pressure difference and the gas flows from the high pressure side to the low pressure side. The transport of permeating species depends on the membrane material and structure.

2.4.1 Gas permeation through porous membranes

If the membranes for gas separation are asymmetric or composite membranes, the diffusion through the membrane can be described by several mechanisms [28]. The rate determining step is the solution-diffusion mechanism used to describe diffusion through the non-porous selective layer. However, other mechanisms can contribute to transport hence the membrane resistance depending on the structure of the supporting layer(s) [28].

Viscuous flow occurs for pores with a radius larger than 10 μm [28]. This means that gas molecules collide exclusively with each other and no separation is obtained. If the pores are smaller or the gas pressure is reduced, the mean free path of the gas molecules becomes larger than the pore size and collision between the molecules are less frequent than collision with the pore wall. The result is that the lighter molecules permeate through the pores. The mean free path is the average distance traversed by a molecule between collisions and depends on the pressure and temperature for which the mean free path increases with decreasing pressure. At constant pressure the mean free path is proportional to the temperature. Knudsen separation can be achieved for membranes having pore sizes smaller than 50 nm [33].

Surface diffusion is the mechanism in which gas molecules are absorbed at the pore walls of the membrane and traverse along the surface [33]. It increases the permeability of the more strongly absorbed component. On the other hand, the effective pore diameter is reduced and selectivity is increased. Capillary condensation can also occur for vapours where a condensed phase partially or completely blocks the pores for which only the soluble species can permeate through. Lastly, molecular sieving occurs when the pore sizes become very small (3-5.2 Å) for which the separation of species depend upon the kinetic diameter, thus only the smaller molecules permeate through the membrane.

There are several phenomena that can occur with time. Concentration polarization, adsorption gel formation and plugging of the pores are common phenomena which induce additional resistance to the transport across the membrane [28]. As there is a difference in the affinity to the gas components, there will be an accumulation of the less permeable species and a depletion of the more permeable ones. This leads to a growing concentration gradient in the boundary layer. This phenomena is called concentration polarization, which reduces the separation efficiency [33]. Gel layer occurs when the concentration becomes very high which then exerts a gel layer resistance. In porous membranes it is also possible for some solutes to block the pores causing a pore-blocking resistance. Lastly, resistance can occur due to adsorption i.e. adsorption can take place at the membrane surface as well as inside the pores. Concentration polarization can be prevented by manipulating the flux and the mass transfer coefficient [28]. The mass transfer coefficient is determined by the diffusion coefficient, the flow velocity and the module configuration. The diffusivities can only be changed by changing the temperature as shown by the relation in Equation (2.44). Changing the flow velocity and module configuration, i.e. decreasing the module length, increasing the hydraulic diameter or a complete different design, can also prevent concentration polarization. An increase in the feed temperature will increase the flux. Concentration polarization is however not severe in gas separation and may also be negligible [28].

2.4.2 Gas permeation through non-porous membranes

For non-porous membranes the solution-diffusion model is used to describe the transport of permeating species through the membrane [28]. First, the permeating species are absorbed at the high pressure membrane surface, and then diffusing through the membrane to the low pressure surface where the components are desorbed. The solution-diffusion mechanism is expressed in terms of the permeability, P which is a function of the diffusivity, D and the solubility, S

$$P = S \cdot D \tag{2.40}$$

The solubility is a thermodynamic parameter and gives a measure of the amount of penetrant absorbed by the membrane at equilibrium [28]. The diffusivity is a kinetic parameter which indicates the rate of the penetrant transportation through the membrane. Diffusivity is dependent on the penetrant geometry and size as well as the nature of the membrane material. For an ideal system, the diffusivity is assumed to be constant, and the solubility is assumed to follow Henry's law

$$c = S \cdot p \tag{2.41}$$

where c and p are the concentration and pressure at the membrane interface, respectively. Thus for an ideal system obeying Fick's law, transport of gases through membranes becomes

$$J_i = \frac{P_i}{\delta}(p_{f,i} - p_{p,i}) \tag{2.42}$$

where P_i and J_i are the permeability and flux of component i, respectively, δ is the membrane thickness, $p_{f,i}$ and $p_{p,i}$ are the partial pressure of component i on the feed side, f, and on the permeate side, p, respectively. Additionally, there is another important performance parameter obtained by comparing the permeabilities of the components which gives the selectivity, $\alpha_{(i/j)ideal}$ defined as

$$\alpha_{(i/j)ideal} = \frac{P_i}{P_j} \tag{2.43}$$

It should be noted that both the solubility and diffusivity for interacting systems become concentration dependent [28]. For organic vapours the solubility can be relatively high depending on the specific interaction, and the diffusion coefficient increase with increasing concentration. Similarly, with respect to the membrane material, glassy polymers are not in a true state of equilibrium and deviates from Henry's law as such. The permeability thus becomes dependent on the driving force. Interactions, especially at high vapour pressures, and use of glassy polymers, results in non-linear sorption isotherms which can be described by free-volume models and Flory-Huggins thermodynamics.

2.4.3 Rubbery and glassy polymers

Polymers are characterised as rubbery or glassy depending on the polymer glass transition temperature and crystallinity [28]. These parameters are further determined by structural factors such as chain flexibility, chain interaction and molecular weight. An amorphous polymer transforms from a rubbery state to a glassy state at a specific temperature referred to as the glass transition temperature. The mobility of the polymer is substantially restricted in the glassy state whereas in the rubbery state there is a high degree of chain mobility. The modulus, specific volume, specific heat, permeability and all physical properties of the polymer change at the glass transition temperature. In rubbery polymers the transport of gases is mainly due to the solubility and for glassy polymers the diffusivity is the determining factor for transportation [33]. Moreover, the permeability increases with permeant size in rubbery polymers whereas for glassy polymers, the permeability decreases with increasing permeant size. In general, rubbery polymers usually have higher permeability than glassy polymers, but the selectivity of glassy polymers is higher [28].

There are several models for describing molecular gas diffusion in polymers, but all apply the concept of free volume as diffusing channels and include the Arrhenius equation where the diffusivity and permeability is given by

$$D = D_0 exp(-E_D/RT) \tag{2.44}$$

$$P = P_0 exp(-E_P/RT) \tag{2.45}$$

where D_0 and P_0 are factors dependent on pressure and diffusion, and E_D and E_P are the activation energy [33]. Also for glassy polymers the Arrhenius equation can be applied to express the diffusivity, solubility and permeability. The permeant gas selectivity in most glassy polymers is dependent on the organic vapour partial pressure. With increasing partial pressure, the amount of vapour absorbed in the polymer increases. However, the vapour plasticisez the polymer, ie. the chain becomes more flexible and the free volume increases considerably, thus the polymer becomes rubbery. In the plasticized material the permeability of the smaller species increases, but the organic vapour permeability increases even more. Plasticization thus transform the permanent gas-selective glassy polymer to an organic vapour-selective rubbery polymer. If a polymer is quenched from a rubbery state to a glassy state, excess free volume is trapped. Over time though, free volume may be lost which is termed physical ageing. It is especially eminent in thin films, commonly used in gas separation, resulting in reduced gas transport properties.

2.4.4 Joule-Thomson effect

If a gas is expanded adiabatically across a membrane, the temperature may change to a great extent depending on the gas component and the applied pressure [28]. Consequently, the permeation properties may be affected, i.e. in general the selectivity increases and the permeation flux decreases if the temperature decreases. The temperature change is expressed by the temperature differential $(\partial T/\partial P)_H$ referred to as the Joule-Thomson coefficient, μ_J, which is defined as follows

$$\mu_J = \left(\frac{\partial T}{\partial P}\right)_H = -\left(\frac{\partial T}{\partial H}\right)_P \left(\frac{\partial H}{\partial P}\right)_T \tag{2.46}$$

The total differential of the enthalpy, H, as a function of temperature and pressure is

$$dH(P,T)_n = \left(\frac{\partial H}{\partial T}\right)_P dT + \left(\frac{\partial H}{\partial P}\right)_T dP \tag{2.47}$$

Consider the enthalpy change of a reversible process

$$dH = VdP + TdS \tag{2.48}$$

where S is the entropy. Differentiating with respect to pressure, P

$$\left(\frac{\partial H}{\partial P}\right)_T = V + T\left(\frac{\partial S}{\partial P}\right)_T \tag{2.49}$$

The following Maxwell relation and identity are used

$$-\left(\frac{\partial S}{\partial P}\right)_T = \left(\frac{\partial V}{\partial T}\right)_P \tag{2.50}$$

$$\left(\frac{\partial H}{\partial T}\right)_P = c_p \tag{2.51}$$

where c_p is the heat capacity. The following expression of the Joule-Thomson coefficient is then obtained

$$\mu_J = \left(\frac{\partial T}{\partial P}\right)_H = -\frac{1}{c_p}\left[V - T\left(\frac{\partial V}{\partial T}\right)_P\right] \tag{2.52}$$

Thus, the gas is either cooled or heated upon pressurizing. For carbon dioxide, the temperature may decrease at high pressure having a Joule-Thomson coefficient of 1.11 K/bar [28]. As such, the Joule-Thomson effect should be accounted for in CO_2 removal at high pressure [34].

2.4.5 Membrane design

In addition to the membrane material and structure, the membrane separation performance is dependent on the membrane configuration and the combination of module configurations for design with several modules [28]. The module configurations can be separated into two categories; flat, and tubular. The types of flat membranes are plate-and-frame modules and spiral-wound modules. Tubular modules involves tubular, capillary and hollow fiber modules mainly distinguished by the tube diameter. Which module configuration and arrangement of the modules to choose is based on several aspects such as economic considerations along with the design parameters, ease of operation, ease of cleaning, and compactness. For gas separation, hollow fiber modules and spiral-wound modules are suitable [28]. However, commercial membrane modules are often designed as hollow fiber membranes as the packing density is much higher than for spiral-wound membranes [35]. Furthermore, for hollow fiber modules it is preferable for the feed solution to flow from the outside-in as to avoid high pressure losses inside the fiber and to attain a large membrane area [28]. Hollow fiber modules are often comprised of asymmetric membranes, while spiral-wound are commonly made with composite membranes [35].

Related to the module design is the flow pattern which also affects the membrane performance [28]. In system design, there are two main categories of module operations; dead-end operation and cross-flow operation, of which cross-flow is preferred for industrial applications because of the lower fouling tendency. There are several types of cross-flow operations such as co-current, counter-current, cross-flow with perfect permeate mixing, and perfect mixing. Of the cross-flow operations, counter-current flow operation gives the best separation results. In order to predict membrane performance and costs in process simulation, theoretical models have been derived.

The governing equations are derived for the various flow patterns applying the principles of gas transportation in the membrane with the conservation equations as basis [36]. However, many simplifying assumptions are employed which often include isothermal conditions and negligible pressure drop in the feed stream and permeate stream [34]. It is also assumed that the effects of total pressure and composition of the gas are negligible and that the permeability of each component is constant [37]. Complete mixing both on the feed side and the permeate side is assumed for the simplest models, which is applicable in systems operating at low recovery. However, most gas separation systems resembles cross-flow conditions i.e. plug flow at the feed side and complete mixing at the permeate side. A selection of different models are presented in the literature by Geankoplis et al., Mulder et al. and Koros et al. [37, 28, 36].

Models in literature often assume co-current or counter-current flow pattern for hollow fiber modules, but as mentioned the counter-current design yields the best performance. For spiral-wound membranes, the available models differ mainly with respect to the flow pattern and the permeate-side pressure drop [38]. Pan [39] has presented a mathematical model for separation of a binary gas mixture which is applicable for both hollow fiber and spiral wound modules with feed flow outside the fiber and permeate pressure drop. Co-current and counter-current flow pattern is assumed for the hollow fiber while cross-flow is assumed for the spiral-wound membrane. Pan has also proposed a model for multicomponent permeation in hollow fiber modules [40]. Counter-current flow pattern has also been studied for spiral-wound modules by Lin et al. [11].

Membrane cost

The commercially available membranes are based on low cost polymers which can be fabricated as asymmetric membranes. New membranes are based on custom-made polymers with improved performance, but they can cost as much as 1000-10000 \$/kg to synthesise [41]. Composite membrane with an ultra thin selective layer however reduces the cost substantially. Hollow fiber productions are estimated to cost 2-5 \$/m^2. Spiral-wound membranes are in the range of 10-100 \$/m^2. This is due to a more efficient production process, and a much higher packing density is obtained in hollow fiber membranes as previously mentioned. The membrane pricing represents the cost of manufacturing the membrane and does not include the costs of housing and costs related to operation. It has been reported that these additional costs may lead to membrane skid cost of 500 \$/m^2 [11, 41], thus the membrane price for high pressure applications is small compared to the overall cost. The price of carbon membranes have been estimated to 15-45 \$/m^2, while zeolite based membranes have been estimated to cost 3000 \$/m^2. The capital cost of membrane gas separation processes is dominated by equipment costs. The membrane cost is dependent on the required membrane area which is further dependent on the membrane performance. The operating costs are dependent on the energy and utility requirements in the gas handling equipment such as compressors. The base case for many techno-economical feasibility studies has been a price of 50 \$/m^2.

2.5 Membranes for natural gas treatment

Natural gas from the wells consists primarily of methane, but also includes other components, for which the gas composition depends on the location of the reservoir, and it also varies from well to well [42]. In addition to methane, natural gas contains lighter hydrocarbons, heavier hydrocarbons, nitrogen, water, H_2S, CO_2, mercaptans, mercury and particulates. H_2S is very corrosive as well as being extremely toxic even at very low concentrations. Furthermore, it can induce hydrogen embrittlement of steel. CO_2 in the presence of water can be very corrosive, and it can lower the heating value of the natural gas. Similarly, nitrogen has no calorific value which can decrease the heating value of the natural gas. Heavier hydrocarbons also reduce the calorific value of the gas as well as increasing the gas dew point. Liquid water enhance corrosion in the pipelines and equipment, and increases the gas dew point. Moreover, the presence of liquid water increases the risk of hydrate formation which can plug valves, fittings and pipelines, in addition to reducing pipeline capacity as a result of accumulation [43]. Water vapour also increases the gas dew point which can condense at lower temperatures. As a result, these components must be removed to meet pipeline specifications and specifications set by customers for the natural gas properties.

Typical specifications are presented in the literature by Kidnay et al. and Poe et al. [44, 45] in which a normal limit for CO_2 is 2-4 vol-% and 4 ppmv for H_2S. Typical values for water content is 4-7 lb/MMscf which corresponds to a gas dew point of -18 °C and 0 °C at 1000 psi gas (69 bar), respectively. According to Poe et al. the gas dewpoint requirement is -10-0 °C at delivery pressures in the range 400-1200 psig (~27-82 bar) [45]. Furthermore, the water dew point should be lower than the lowest pipeline temperature to avoid condensation [44].

Membranes for dehydration are still in the early commercialisation stage [46]. Currently, there are just a few natural gas dehydration installations. However, there is increasing research on and development of membranes for dehydration. Hollow fiber PEEK composite membranes and spiral-wound MTR Pebax® co-polymer based composite membranes are examples of membranes that have been suggested [46, 10, 11]. For sweetening, there are several commercially available membranes such as polysulfone (PSF), polyimide (PI), and cellulose acetate (CA). However, low performances with respect to selectivity and permeance due to penetrant induced plasticization at higher pressures, makes these membranes exceedingly less cost efficient [47, 48]. According to Xiao et al. the most important identified challenges in membrane gas separation in general are obtaining both good selectivity and permeability, and maintaining a long-term separation performance by overcoming the problems of ageing and plasticization [49]. Moreover, for dehydration membranes specifically, the methane loss need to be reduced to be competitive [50].

Inorganic membranes have been investigated as they possess high thermal and chemical stability as well as high mechanical strength and a long lifetime [28], but they are very expensive compared to the commercial polymeric membranes. Some composite membranes which has shown promising performance on CO_2/CH_4 separation are mixed matrix zeolite/poly vinyl acetate [51], carbon nanotubes reinforced polyvinylamine/polyvinyl alcohol fixed-site carrier membranes, and cellulose acetate based hollow fiber carbon membranes [52, 53]. It has recently been reported by Zhang et al. good CO_2/CH_4 separation performance in carbon molecular sieve hollow fiber membranes made from cross-linked polyimide precursors, which were tested at pressures up to 120 bar [12]. The results of a 50% CO_2/50% CH_4 gas mixture showed a separation factor of ~ 60 and a CO_2 permeance of 0.27 m^3(STP)/m^2·h·bar (100 gpu) at 60-120 bar.

Membrane Technology & Research Inc. has constructed Pebax® block co-polymer based composite membranes which have been tested in lab scale and plant scale with auspicious H_2O/CH_4 separation performance. In lab scale the composite membrane had a performance of 5.4 m^3(STP)/m^2·h·bar (2000 gpu) and H_2O/CH_4 selectivity up to 1500 at a feed pressure of 2 bar [10]. In plant scale testing at feed pressures of 30-60 bar, the water permeance was 2.51 m^3(STP)/m^2·h·bar (930 gpu), and a selectivity of 47. Although, the field testing showed a lower performance, it shows a possibility for membrane dehydration of high-pressure natural gas. Moreover, the counter-current membrane design with dry gas as sweep was found to be the best combination of minimal membrane area requirement and CH_4 loss. However, spiral-wound modules may not provide perfect counter-current flow, which was also indicated by the field-test results.

Chapter 3

Process Design Basis

In this chapter, a description of a hybrid membrane system for natural gas dehydration and CO_2 removal is presented, followed by the simulation basis giving more detailed information on the feed properties and product specifications as well as membrane properties. Lastly, the different design configurations and the cost model subject to minimization are described and derived, respectively.

3.1 Process Description

A process design of a hybrid membrane system, as shown in Figure 3.1, was studied and adjusted with the aim of minimizing the annual specific cost of natural gas processing while meeting the specifications. The specifications for the sweet natural gas was a water content corresponding to a water dew point temperature below −40 °C, and a maximum amount of CO_2 of 2.5 mol-%, respectively. Additionally, the process should attain a CO_2 purity of minimum 95 mol-% in the produced CO_2 stream for enhanced oil or gas recovery. Lastly, it was desired to minimize the loss of methane and satisfy a gas dew point between -10-0 °C.

First, sand, dust, gum and other particulates are filtered out and heavier hydrocarbons and free liquid water are removed in a condenser. Then, the pretreated natural gas is sent to the dehydration unit with water-selective MTR Pebax® membranes to remove water vapour. Some hydrocarbons and CO_2 will also permeate through the membrane. Therefore, the removed water vapour and lean natural gas is separated in a condenser for reuse and burning respectively. From the dehydration unit, the dehydrated natural gas is sent to carbon membrane units in series for removal of CO_2 as to produce sweet natural gas with a minimum hydrocarbon purity of 97.5 mol-%. Some CO_2 rich permeate with a minimum purity of 95 mol-% is recycled as sweep gas to the dehydration unit as to increase the driving force and to prevent water from condensing, and the rest is compressed for possible use in enhanced oil and gas recovery (EOR/EGR) and carbon storage. The enhanced oil and gas recovery process was not further studied.

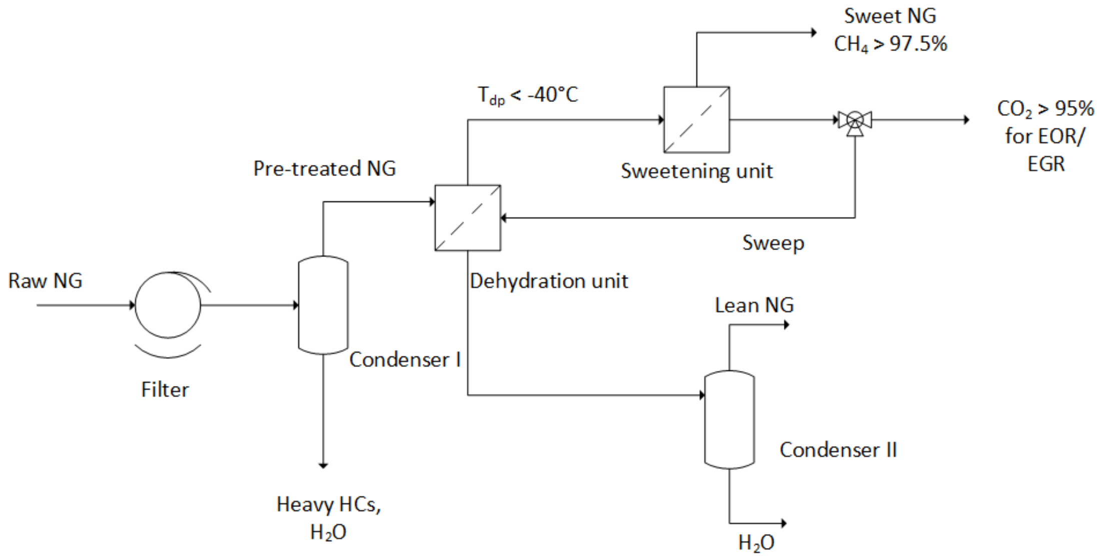


Figure 3.1: Principal illustration of the hybrid membrane system for natural gas dehydration and CO_2 removal.

3.2 Simulation basis

Raw natural gas free of particulates at ~ $5 \cdot 10^5$ m^3(STP)/h, 60 bar and 80 °C with a composition as shown in Table 3.1, was treated by Pebax® spiral-wound membranes in the dehydration unit and carbon hollow fiber membranes in the sweetening unit, both with counter-current flow pattern. The membrane performances reported by Haiqing et al. and Zhang et al. are presented in Table 3.2 [11, 12]. The targets for the simulation were a CO_2 purity of minimum 95 mol-% for the enhanced oil and gas recovery process, a CH_4 purity of minimum 97.5 mol-% and a water dew point of maximum -40 °C as listed in Table 3.1. Because the sweet natural gas will consist of small amounts of lighter hydrocarbons, the methane purity target was based on the total hydrocarbon purity.

Table 3.1: The feed parameters and process targets used as simulation basis.

Parameter	Value
Feed flow [m^3(STP)/h]	$5.258 \cdot 10^5$
Feed pressure [bar]	60
Feed temperature [°C]	80
Water content (dew point) [°C]	<-40
CH_4 purity in sweet natural gas [mol-%]	>97.5
CO_2 purity for EOR/EGR [mol-%]	>95
CH_4 loss [%]	<2
Compressor adiabatic efficiency [%]	75

Table 3.2: Composition of the feed gas and gas permeances in the MTR Pebax® membrane, and the carbon hollow fiber membranes.

Component	Concentration in the feed gas [mol-%]	Permeance in Pebax® membrane [mol(STP)/m^2·h·kPa]	Permeance in carbon membranes [mol(STP)/m^2·h·kPa]
Methane	77.22	0.023	0.002
Ethane	7	0.046	0.001**
Propane	3	0.071	0.001**
Butane	1.9	0.075	0.001**
C_{5+}	0.1	0.116	0.001**
H_2O	0.78	1.218	0.95**
CO_2	10	0.069	0.095

**estimated

3.3 Cost model

The objective function subject to optimization was the specific natural gas processing cost which is a function of annual operating cost, annual capital related cost and amount of produced sweet natural gas (SNG), $\dot{V}_{SNG}$.

$$C_s = \frac{C_{OPEX} + C_{CRC}}{\dot{V}_{SNG}} \quad \left[\frac{\$}{\text{year}}\right] \tag{3.1}$$

where C_s is the specific natural gas processing cost, C_{OPEX} is the annual operating cost, C_{CRC} is the annual capital related cost, and $\dot{V}_{SNG}$ is the produced amount of sweet natural gas. The cost categories with corresponding parameters are presented in Table 3.4.

Only the membranes and compressors/turbine were included in the cost calculations as the costs of the smaller equipment were expected to be much lower. Thus, the operating cost was estimated by the compressor power demand and turbine power generation, excluding labour cost and assuming no operating cost related to the membranes. The annual operating cost was calculated as a function of the total power consumption obtained from the optimization routine of the different design configurations, the electricity cost and annual operating hours which was assumed to be 0.04 \$/kWh and 8000 h/year, respectively [1].

The capital related cost was calculated by a preliminary cost estimation based on purchased cost of equipment at some base conditions depending on specific equipment type, system pressure and materials of construction as described by Turton et al. [54]. It involves estimation of the bare module cost, and the total module cost, whereby the total module cost refers to the cost of making small to moderate expansions or alterations to an existing facility. The bare module cost of the major equipment was calculated as follows

$$C_{BM} = C_p^0 F_{BM} \tag{3.2}$$

where C_{BM} is the bare module equipment cost, C_p^0 is the purchased cost for base conditions, and F_{BM} is the bare module cost factor.

The bare module equipment cost, C_{BM}, represents the sum of indirect and direct costs. The base conditions are equipment in carbon steel and operation at near-ambient pressure. The compressors used in this design are centrifugal compressors (450-3000 kW) in stainless steel operating at high pressures. Consequently, the purchased cost of compressors was fitted as shown in Equation (3.3). The bare module cost factor was adjusted to the material of construction, stainless steel, F_{BMSS}, obtained from CAPCOST© as presented in Table 3.3 [54]. Additionally, the purchased cost was adjusted to account for inflation using the cost index. For the purposes of comparing the optimization routine presented in this work with previous work, the index from 2017 was used as shown in Equation (3.4). The turbine cost was assumed to be approximately the same as the compressor cost.

$$log_{10} C_p^0 = K_1 + K_2 log_{10}(A) + K_3 log_{10}(A^2) \tag{3.3}$$

$$C_{2017} = C_{2001} \frac{I_{2017}}{I_{2001}} \tag{3.4}$$

A is the compressor duty, and K_i are constants obtained from CAPCOST© presented by Turton et al. and shown in Table 3.3 [54]. C_{2017} and C_{2001} are the purchased cost in 2017 and 2001, respectively, and I_{2017} and I_{2001} are the cost indexes in 2017 and 2001, respectively [55, 54]. Adding contingency and fee costs (15% and 3% of the bare module cost respectively) to the bare module costs gives the total module cost of the plant. It can be evaluated from

$$C_{TM} = \sum_{i=1}^{n} C_{TM,i} = 1.18 \sum_{i=1}^{n} C_{BM,i} \tag{3.5}$$

where C_{TM} is the total module cost of the plant, and $C_{TM,i}$ and $C_{BM,i}$ are the total module cost and bare module cost of each equipment, i, respectively.

The average membrane skid cost, C_M, for both membrane types was assumed to be 50 \$/m^2 in accordance with Section 2.4.5. Additionally, it was assumed a lifetime of 5 years. Finally, the annual capital related cost was assumed to be 20% of the total module cost.

$$C_{CRC} = 0.2(C_{TM} + C_M) \tag{3.6}$$

Table 3.3: Data for estimation of purchased cost of centrifugal compressors in stainless steel.

Compressor	K_1	K_2	K_3	F_{BMSS}	W_{min} [kW]	W_{max} [kW]
Centrifugal	2.2897	1.3604	-0.1027	8.0	450	3000

Table 3.4: Cost categories and parameters for natural gas processing [1].

Category	Parameter	Value
Capital related cost	Membrane skid cost, C_M [\$/m^2]	50
	Compressor cost, $C_{BM,i}$ [\$]	Equation (3.2)
	Total module cost, C_{TM} [\$]	Equation (3.5)
Operating cost	Electricity Cost, C_E [\$/kWh]	0.04
	Compressor Duty, P_C [kW]	Table 6.2
	Annual operating hours, t [h/year]	8000
Annual operating cost, C_{OPEX}	$C_E \cdot P_C \cdot t$ [\$/year]	
Annual Capital Related Cost, C_{CRC}	$0.2(C_M + C_{TM})$ [\$/year]	
Specific natural gas processing cost, C_s	$\frac{\text{CRC+OPEX}}{\text{Annual produced NG}}$, [\$/m^3]	
Assumptions	Membrane lifetime	5 years
	Compressor lifetime	25 years
	Annual operating hours	8000 h/year

3.4 Simulation

The hybrid membrane system was simulated in Aspen Hysys v10 as presented in Figure 3.2. The process flow diagram shows that filtered raw natural gas (NG) is cooled in a heat exchanger and a cooler, E-30 and E-31, respectively, to 10 °C. NG then enters a condenser, V-100, to knock out heavier hydrocarbons and liquid water. The gas is reheated to 60 °C, by heat exchanging in E-30 with the raw natural gas stream. The gas is further fed to the 1st stage dehydration unit to remove most of the water vapour and to reach the water dew point target in the dehydrated gas, R-1. Dehydrated natural gas in the retentate, R-1, is mixed with the NG rich retentate, R-3, and fed to the 2nd stage sweetening unit for CO_2 removal. Sweet natural gas in the retentate, R-2, is produced with a purity of minimum 97.5 mol-%. The permeate gas, P-2, is pressurized by K-1, representing a three-stage compressor as illustrated in Figure 3.3. It is then fed to the 3rd stage sweetening unit for further CO_2 removal as to attain a CO_2 purity of minimum 95 mol-% in the permeate, P-3. Some of the permeate is recycled as sweep gas at 1 bar to the dehydration unit, and some is used for enhanced oil and gas recovery, for which the pressure is 5 bar.

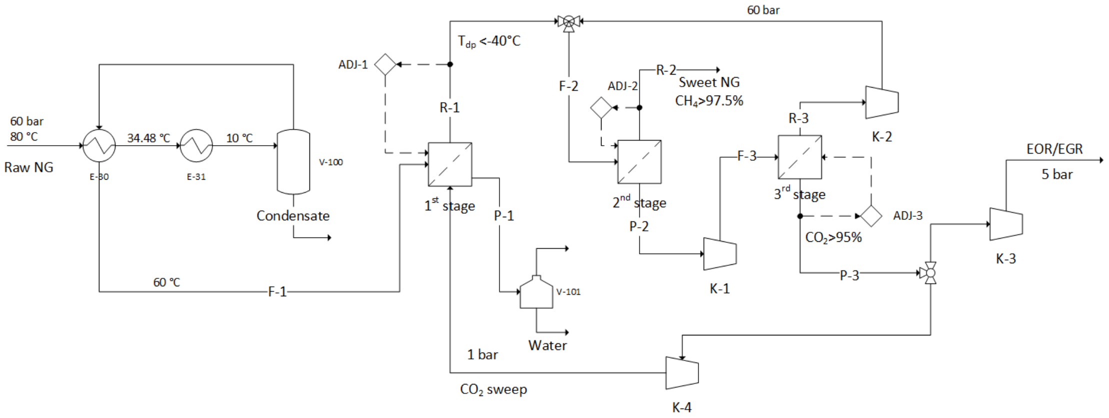


Figure 3.2: Process flow diagram of the initial design of the hybrid membrane system.

3.4.1 Multistage compression

A detailed representation of K-1 is given in Figure 3.3. The 2nd stage permeate, P-2, is compressed over three compressors in series, K-100, K-101 and K-102, respectively with equal pressure ratios. As the temperature of the permeate increases upon compression, interstage cooling is provided by coolers, E-40 and E-41, to reduce the temperature to initial suction temperature, $T_{\text{P-2}}$. This reduces the required work input to the compressors and avoids damage to compressor seals due to high temperatures. The cooling results in a pressure drop set to 0.05 bar over each cooler. As the 3rd stage feed must be at 60 °C a cooler, E-42, is placed prior to the stage.

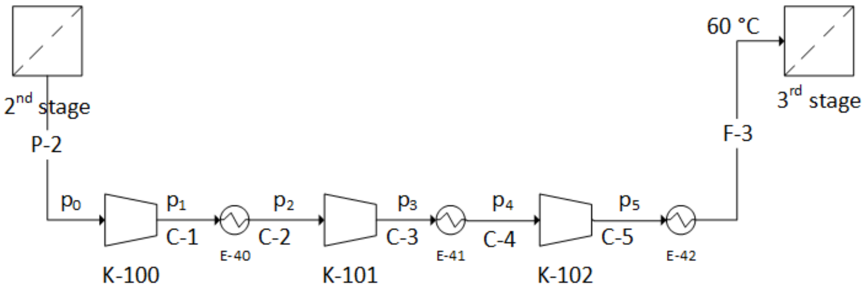


Figure 3.3: Three-stage compression with intercooling for compression of the 2nd stage permeate before entering as feed to the 3rd stage.

The simulation of the membrane units is initially based on a membrane model developed at the Department of Chemical Engineering at NTNU, which was integrated in Aspen Hysys through ChemBrane. The model assumes constant pressure and temperature on the shell side, and constant pressure on the fiber side in addition to permeances independent of composition. Moreover, the model uses the fourth-order Runge-Kutta method to calculate the flux along the fiber length, and then iterates over the permeate values to converge to a solution. The Aspen HYSYS flowsheet with the ChemBrane model is presented in Appendix C.2. The flowsheet shows that sweep gas is implemented for all stages with a temperature of 30 °C. The permeate in all membrane units leave at a temperature lower than 30 °C which is calculated in the ChemBrane model based on the enthalpies obtained from HYSYS as follows

$$H_p = \frac{H_f - H_r + H_s}{N_p} \tag{3.7}$$

where H_p, H_f, H_r, H_s and N_p are the permeate, feed, retentate and sweep gas enthalpies and permeate flow rate, respectively.

Other design configurations were also simulated and subject to optimization which is described in more detail in Chapter 5. However, the recently described design served as a basis for subsequent alterations, such as replacing the ChemBrane model with the membrane model presented in Chapter 4, and reducing the number of compressors.

Chapter 4

Membrane modelling

In this chapter a membrane model for simulation of the membrane units is presented, followed by the method for solving the set of model equations. One of the principles of modelling is to estimate the behaviour and performance of a system based on governing equations and phenomena. As such, one is able to perform preliminary feasibility studies which is valuable knowledge before executing physical experiments. Ideally the aim is to create a model which represents the reality to the best extent possible. However, such considerations often make for overly complex models that is challenging to solve, and involve high computational cost. In order to solve a model efficiently, assumptions that simplify the model are often taken. In other words, there is a trade-off between high-accuracy realistic models, and approximate models that are efficient and readily solved.

4.1 Model description

A hollow fiber membrane module with counter-current flow pattern was modelled. The aim was to create a model that includes more phenomena than the ChemBrane membrane model. More specifically, in addition to the differential changes in molar flow rate on the feed and permeate side, the aim was to include the differential changes in temperature on both sides and the pressure loss on the permeate side.

The high pressure multicomponent gas is fed to the shell side of a hollow fiber membrane module and is flowing in negative z direction. The components permeate through the membrane at different rates to the fiber bore, where the permeate flows counter-currently to the shell side flow in positive z direction, as illustrated in Figure 4.1.

The following assumptions are made for the model:

- The rates of permeation obey Fick's law (Equation (2.42)).
- The different velocity profiles at entrance region is neglected and a fully developed flow is assumed.
- Plug flow on shell side; Constant velocity and constant pressure because of the high feed pressure.
- Laminar flow on fiber side, due to the low sweep gas flow rate.
- Uniform temperature profile on both sides.
- Resistance only through the membrane. Both for mass transfer and heat transfer, any resistance in the boundary layers are neglected.
- Mass transfer by solution-diffusion mechanism.
- Gas viscosities are independent of pressure and assumed to not vary significantly with composition, i.e. constant viscosity.
- Permeance is assumed to be independent of driving force, i.e. constant permeances.
- The effective membrane thickness is constant along the length of the fiber.

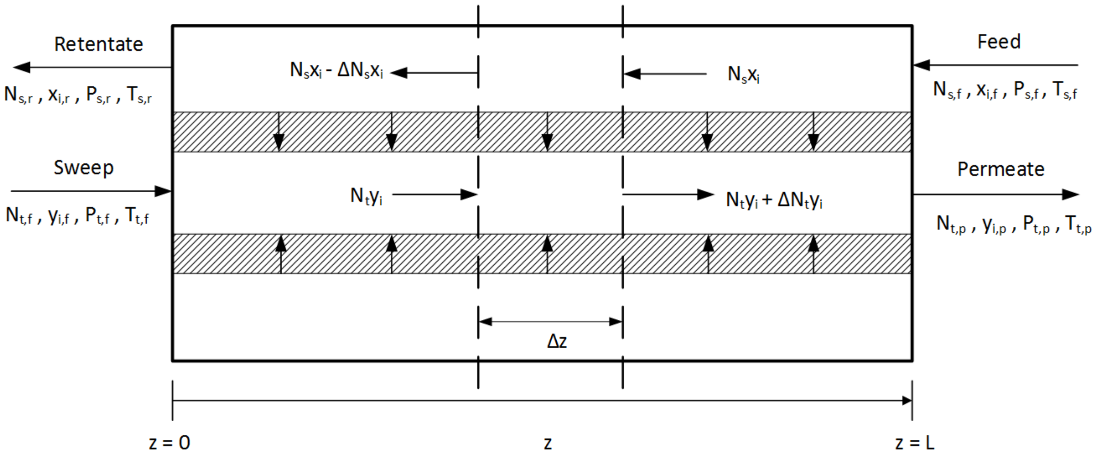

Retentate
$N_{s,r}$, $x_{i,r}$, $P_{s,r}$, $T_{s,r}$
N_sx_i - ΔN_sx_i
N_sx_i
Feed
$N_{s,f}$, $x_{i,f}$, $P_{s,f}$, $T_{s,f}$
Sweep
$N_{t,f}$, $y_{i,f}$, $P_{t,f}$, $T_{t,f}$
N_ty_i
$N_ty_i + \Delta N_ty_i$
Permeate
$N_{t,p}$, $y_{i,p}$, $P_{t,p}$, $T_{t,p}$
Δz
z = 0
z
z = L

Figure 4.1: A schematic representation of counter-current gas separation in a hollow fiber membrane with feed entering at $z = L$ on shell side, and sweep gas entering on fiber side at $z = 0$. The gas permeates through the membrane to the fiber side and leaves at $z = L$, and the reject leaves at $z = 0$.

4.2 Model equations

The model equations are presented in this section, while the complete derivation is presented in Appendix A.1. Conservation of mass, momentum and energy forms the basis for the model equations. The initial approach is to create a model with respect to conservation of mass and momentum, which is later evolved by including conservation of energy, with the assumptions listed in Section 4.1. Moreover, the initial model is later referred to as modMB, the model with the first measures of extensions is referred to as modEBJ and the final model is referred to as modEBJQ.

4.2.1 Initial model (modMB)

Making a balance over a small element Δz with respect to mass and pressure and assuming steady state, gives the following model equations for feed and permeate side molar flow rates, and pressure on the permeate side

$$\frac{dN_s x_i}{dz} = N_T \pi D_o J_i \tag{4.1}$$

$$\frac{dN_t y_i}{dz} = N_T \pi D_o J_i \tag{4.2}$$

$$\frac{dP_t}{dz} = -\frac{128 \mu R T_t}{N_T \pi D_c^4 P_t} \sum_{i=1}^{nc} N_t y_i \tag{4.3}$$

where N_s, N_t, P_t and J_i are the the molar flow rate on the feed side (shell side), the molar flow rate and pressure on the permeate side (fiber side), and the molar flux across the membrane, respectively. μ, T_t, N_T and nc are the viscosity of the permeate, the fiber side temperature, the number of fibers in the module, and the number of components permeating through the membrane, respectively. D_o and D_c are the fiber outer and inner diameter, respectively. The flux is defined as follows

$$J_i = \beta_i (P_s x_i - P_t y_i), \quad \beta_i = \frac{D_i S_i}{\delta} \tag{4.4}$$

where δ is the membrane thickness and β_i, D_i, and S_i are the permeance, diffusivity and selectivity for component i, respectively. The mole fractions on the feed side and permeate side, x_i and y_i are defined as follows

$$x_i = \frac{N_s x_i}{\sum_{i=1}^{nc} N_s x_i}; \qquad y_i = \frac{N_t y_i}{\sum_{i=1}^{nc} N_t y_i} \tag{4.5}$$

Dimensionless model equations

In order to solve the set of equations, dimensionless variables are introduced with every dimensionless variable,γ_i^*, satisfying the following condition $\gamma_i^* \in [0,1]$.

$$z^* = \frac{z}{L};\ (N_s x_i)^* = \frac{N_s x_i}{\sum\limits_{i=1}^{nc}(N_s x_i)_f};\ (N_t y_i)^* = \frac{N_t y_i}{\sum\limits_{i=1}^{nc}(N_s x_i)_f};\ P_s^* = \frac{P_s}{P_{sf}};\ P_t^* = \frac{P_t}{P_{sf}} \tag{4.6}$$

Inserting the dimensionless variables into the model equations in Equations (4.1) to (4.3) gives the following dimensionless model equations

$$\frac{dN_s x_i^*}{dz^*} = K_j \beta_i \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.7}$$

$$\frac{dN_t y_i^*}{dz^*} = K_j \beta_i \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.8}$$

$$\frac{dP_t^*}{dz^*} = K_{pt} \frac{\sum\limits_{i=1}^{nc}(N_t y_i)^*}{P_t^*} \tag{4.9}$$

where the coefficients K_j and K_{pt} are defined as follows

$$K_j = \frac{\pi L N_T D_o P_{sf}}{\sum\limits_{i=1}^{nc}(N_s x_i)_f} \tag{4.10}$$

$$K_{pt} = -\frac{128 L \mu R T_{tf} \sum\limits_{i=1}^{nc}(N_s x_i)_f}{P_{s,f}^2 N_T \pi D_i^4} \tag{4.11}$$

Boundary values

The boundary values for the system with dimensionless variables are presented in Table 4.1

Table 4.1: The system boundary values for the cases with and without sweep gas, respectively.

With sweep gas		Without sweep gas	
$z^* = 0$	$z^* = 1$	$z^* = 0$:	$z^* = 1$:
$N_t y_i^* = (N_t y_i)_f^*$	$N_s x_i^* = (N_s x_i)_f^*$	$N_t y_i^* = 0$	$N_s x_i^* = (N_s x_i)_f^*$
$P_t^* = P_{tf}^*$	$P_s^* = P_{sf}^*$		$P_t^* = P_{tp}^*$
			$P_t^* = P_{tp}^*$

4.2.2 Extension of the initial model (modEBJ)

An energy balance over a small element Δz assuming steady state is made. The kinetic and potential energy are assumed to be negligible, and the changes in the specific heat capacities are assumed to be very small as the model deals with relatively small temperature gradients at low temperatures. Thus the specific heat capacities are treated as constants with values at standard temperature and pressure as presented by Blackman [56]. The additional model equations are presented in Equations (4.12) and (4.13), which introduces two new dimensionless variables and additional boundary values as presented in Table 4.2, and additional dimensionless model equations as stated in Equations (4.14) to (4.16). Notice that the permeate side pressure is affected by the introduction of the temperatures as it is a function of the permeate side temperature.

$$\frac{dT_s}{dz} = N_T \pi D_o \left[\frac{\sum_{i=1}^{nc} J_i c_{pi} (T_s - T_t)}{\sum_{i=1}^{nc} N_s x_i c_{pi}} \right] \tag{4.12}$$

$$\frac{dT_t}{dz} = N_T \pi D_o \left[\frac{\sum_{i=1}^{nc} J_i c_{pi} (T_s - T_t)}{\sum_{i=1}^{nc} N_t y_i c_{pi}} \right] \tag{4.13}$$

Table 4.2: The additional system dimensionless variables, and the additional boundary values for the cases with and without sweep gas, respectively.

With sweep gas		Without sweep gas		Dimensionless variables
$z^* = 0$	$z^* = 1$	$z^* = 0$:	$z^* = 1$:	$T_s^* = \frac{T_s}{T_{sf}}$; $T_t^* = \frac{T_t}{T_{sf}}$
$T_t^* = T_{tf}^*$	$T_s^* = T_{sf}^*$		$T_s^* = T_{sf}^*$	

Dimensionless model equations

$$\frac{dT_s^*}{dz^*} = K_j \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_s x_i^* c_{pi}} \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.14}$$

$$\frac{dT_t^*}{dz^*} = K_j \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_t y_i^* c_{pi}} \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.15}$$

$$\frac{dP_t^*}{dz^*} = K_{ptc} \frac{T_t^* \sum\limits_{i=1}^{nc} (N_t y_i)^*}{P_t^*} \tag{4.16}$$

where K_{ptc} is defined as

$$K_{ptc} = \frac{T_{sf}}{T_{tf}} K_{pt} \tag{4.17}$$

4.2.3 Further extension - Including heat of conduction (modE-BJQ)

Including the heat of conduction through the membrane introduces two new equations to the set of model equations as shown in Equations (4.18) and (4.19). Inserting the dimensionless variables as presented in Equation (4.6) and Table 4.2, gives the additional dimensionless model equations shown in Equations (4.21) and (4.22).

$$\frac{dT_s}{dz} = N_T \pi \left[\frac{D_o \sum\limits_{i=1}^{nc} J_i c_{pi} (T_s - T_t) + Q_k}{\sum\limits_{i=1}^{nc} N_s x_i c_{pi}} \right] \tag{4.18}$$

$$\frac{dT_t}{dz} = N_T \pi \left[\frac{D_o \sum\limits_{i=1}^{nc} J_i c_{pi} (T_s - T_t) + Q_k}{\sum\limits_{i=1}^{nc} N_t y_i c_{pi}} \right] \tag{4.19}$$

where the heat of conduction across the membrane, Q_k, is defined as

$$Q_k = \frac{2\pi\kappa(T_s - T_t)}{ln\left(\frac{D_o}{D_c}\right)} \tag{4.20}$$

where κ is the thermal conductivity, and D_o and D_c are the outer and inner diameter of the fibers, respectively.

$$\frac{dT_s^*}{dz^*} = \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_s x_i^* c_{pi}} \left(K_j \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] + K_k \right) \tag{4.21}$$

$$\frac{dT_t^*}{dz^*} = \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_t y_i^* c_{pi}} \left(K_j \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] + K_k \right) \tag{4.22}$$

where K_k is defined as follows

$$K_k = \frac{2\pi\kappa L N_T}{\ln \frac{D_o}{D_c} \sum\limits_{i=1}^{nc} (N_s x_i)_f} \tag{4.23}$$

4.3 Numerical procedure

For solving the set of model equations, the orthogonal collocation method was chosen to discretize the problem. As described in Section 2.1, the method provides an optimal distribution of nodes and lower computational cost compared to the finite methods. Applying the orthogonal collocation method, the discrete approximation to the first order differential equations can be stated as

$$\left(\frac{dY}{dz^*}\right)_{z_i^*} = \sum_{j=1}^{nz} A_{ij} Y_j, \quad i = 1, 2, ..., nz \tag{4.24}$$

where A_{ij} is the set of first-order derivative weights, Y is the set of dimensionless function variables, $f(z^*)$, and the indexes i and j represent the collocation points. As described in Section 2.1.2, the residuals, R, of the initial model equations in Equations (4.7) to (4.9) can be stated as follows

$$R_{ij} = \sum_{j=1}^{nz} A_{ij}(N_s x_i^*)_j - K_j \beta_i \left[P_s^* \frac{(N_s x_i)^*}{\sum\limits_{i=1}^{nc} (N_s x_i)^*} - P_t^* \frac{(N_t y_i)^*}{\sum\limits_{i=1}^{nc} (N_t y_i)^*} \right] \tag{4.25}$$

$$R_{ij} = \sum_{j=1}^{nz} A_{ij}(N_t y_i^*)_j - K_j \beta_i \left[P_s^* \frac{(N_s x_i)^*}{\sum\limits_{i=1}^{nc} (N_s x_i)^*} - P_t^* \frac{(N_t y_i)^*}{\sum\limits_{i=1}^{nc} (N_t y_i)^*} \right] \tag{4.26}$$

$$R_i = \sum_{j=1}^{nz} A_{ij}(P_t^*)_j - K_{pt} \frac{\sum\limits_{i=1}^{nc} (N_t y_i)^*}{P_t^*} \tag{4.27}$$

Including the temperature differentials and altered pressure differential in Equations (4.14) to (4.16)

$$R_i = \sum_{j=1}^{nz} A_{ij}(T_s^*)_j - K_j \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_s x_i^* c_{pi}} \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.28}$$

$$R_i = \sum_{j=1}^{nz} A_{ij}(T_t^*)_j - K_j \frac{T_s^* - T_t^*}{\sum\limits_{i=1}^{nc} N_t y_i^* c_{pi}} \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum\limits_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum\limits_{i=1}^{nc} N_t y_i^*} \right] \tag{4.29}$$

$$R_i = \sum_{j=1}^{nz} A_{ij}(P_t^*)_j - K_{pt} \frac{T_t^* \sum\limits_{i=1}^{nc} (N_t y_i)^*}{P_t^*} \tag{4.30}$$

Including the heat of conduction, the temperature residuals in Equations (4.28) and (4.29) become as follows

$$R_i = \sum_{j=1}^{nz} A_{ij}(T_s^*)_j - \frac{T_s^* - T_t^*}{\sum_{i=1}^{nc} N_s x_i^* c_{pi}} \left(K_j \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum_{i=1}^{nc} N_t y_i^*} \right] + K_k \right) \tag{4.31}$$

$$R_i = \sum_{j=1}^{nz} A_{ij} - (T_t^*)_j - \frac{T_s^* - T_t^*}{\sum_{i=1}^{nc} N_t y_i^* c_{pi}} \left(K_j \sum_{i=1}^{nc} \beta_i c_{pi} \left[P_s^* \frac{N_s x_i^*}{\sum_{i=1}^{nc} N_s x_i^*} - P_t^* \frac{N_t y_i^*}{\sum_{i=1}^{nc} N_t y_i^*} \right] + K_k \right) \tag{4.32}$$

$$R = 0 \quad \forall R_i \in \mathfrak{R} \tag{4.33}$$

The model was specified and solved in MATLAB as presented in Appendix E.2. The collocation points and the first order derivative weights were calculated as to transform the set of first order differential equations into a set of algebraic equations. Then, `fsolve`, an optimization scheme based on the Newton-Rhapson method was applied to find the solution of the non-linear algebraic equations by seeking to obtain $R = 0$. The number of non-linear algebraic equations to solve was $2(nc+1)(nz)$ for the initial model and $2(nc+3)(nz)$ for the extended models, where nc is the number of chemical species permeating through the membrane and nz is the total number of collocation points.

4.4 Simulation

First, an evaluation of the initial membrane model (modMB) performance and numerical procedure for the carbon membranes were conducted. Next, both membrane types were compared to the ChemBrane model performance at process conditions obtained in the optimal design configuration. Lastly, the model was compared to experimental data and the mollocator membrane model reported by Yunhan et al. [57].

The aim for evaluating the carbon membranes was to find the conditions for which the solution was obtained efficiently and accurately for the purposes of integrating the membrane in Aspen Hysys. Accordingly, it had to be tested whether the membrane should be integrated with sweep gas or not, and whether the boundary condition regarding the fiber side pressure should be the sweep gas pressure or the permeate pressure. The model was simulated with the process conditions obtained in the optimal design configuration for the 2nd stage as presented in Table 6.2 and the permeances as listed in Table 3.2. Moreover, for the scenarios with sweep gas, the sweep gas flow rate was set very low because it was not attempted to simulate the membrane with actual sweep gas, but to study the numerical and computational differences. Consequently, there were three scenarios that were simulated and evaluated with 12 and 32 collocation points respectively. The optimization settings (function tolerance, step tolerance etc.) were equal for all scenarios, but the initial values for the third scenario differed slightly from the two others. A different set of initial points was applied for scenario 3 to reduce the number of iterations. The run time, number of iterations and the residual were obtained for simulations for each scenario, respectively. In addition, the output parameters such as the mass balances, sweep gas pressure and permeate pressure were obtained as to investigate the differences. The scenarios are as follows

Scenario 1: With sweep gas and defined permeate pressure

Scenario 2: With sweep gas and defined sweep gas pressure

Scenario 3: Without sweep gas and therefore defined permeate pressure

Next, having evaluated the different scenarios for the carbon membranes, the membrane model was compared to the ChemBrane model, experimental data and the mollocator model at conditions presented in Table 4.3. Case 1 represents the spiral-wound Pebax® co-polymer based membrane in the 1st stage in the process design. It was simulated with sweep gas on the fiber side flowing counter current to the feed on the the shell side, and the permeate pressure was defined. Case 2 represents the 2nd and 3rd stage polyimide derived hollow fiber carbon membranes for which scenario 3 was chosen for comparison. The membranes were simulated with the permeances as listed in Table 3.2 and at process conditions obtained in the optimal design configuration as listen in Table 6.2. Case 3 refers to a pilot-scale hollow fiber carbon membrane module with the effective area of 1.27 m^2 tested for a gas mixture of 10% CO_2/90% CH_4 fed to the shell side at 25 °C and 5 bar. There was no sweep gas applied and the permeate exited on the fiber side counter-currently to the feed. Case 4 refers to a lab-scale module with 106 hollow fiber carbon membranes tested for a gas mixture of 40% CO_2/60% CH_4 fed to the fiber side at 5 bar and a N_2 sweep gas was fed to the shell side at 1 bar flowing counter-currently to the feed. Consequently, the membrane model modMB was altered as to simulate fiber side feed and shell side permeate. The experimental error for the instruments employed in the experiments was expected to be 5% according to Yunhan et al. [57]. The relative deviation, RD [%], was calculated for comparison as follows

$$RD = 100\,\frac{\chi_{model} - \chi_j}{\chi_j} \tag{4.34}$$

where χ_{model} refers to either the modMB model or the mollocator model compared to the χ_j results obtained from the ChemBrane model, the mollocator model or the experimental data, respectively. As the majority of the permeate in case 2 and 3 was assumed to be CO_2, the permeate viscosity was estimated to be the viscosity of CO_2 at 25 °C and 1 atm [58]. In case 3, the majority of the permeate was assumed to be methane, thus the viscosity was estimated as the viscosity of methane at 25 °C and 1.08 bar [59]. For case 1, the viscosity was estimated as the average of the CO_2 and methane viscosities. Additionally, the heat capacities and thermal conducticity for the simulations of the extended models are presented in Appendix A.2 along with the viscosities.

Table 4.3: Operating conditions, membrane properties and feed parameters are given for simulation of four different simulations cases.

Parameter	Case			
	1	2	3	4
Membrane material	Pebax® co-polymers	PI-Carbon	CA-Carbon	CA-Carbon
Feed flow, N_f [mol/s]	$6.153 \cdot 10^3$	$5.516 \cdot 10^3$	$3.718 \cdot 10^{-4}$	$4.464 \cdot 10^{-4}$
Sweep flow, N_sg [mol/s]	3.595	-	-	$2.012 \cdot 10^{-5}$
Feed pressure, P_f [10^5 Pa]	60	60	5	5
Permeate pressure, P_p [10^5 Pa]	1	1.492	1	1
Temperature bore side, T_t [K]	303	303	298	298
Temperature shell side, T_s [K]	333	333	298	298
Sweep gas composition, y_{if} [mol-%]	Table 4.4	Table 4.4		
CO_2			-	-
N_2			-	1
Feed composition, x_{if} [mol-%]	Table 4.4	Table 4.4		
CO_2			10	40
CH_4			90	60
Permeances, β_i [mol/m^2 Pa s]	Table 3.2	Table 3.2		
CO_2 10^{-9}			1.749	8.405
CH_4 10^{-10}			1.227	1.323
N_2 10^{-10}				3.968
Fiber length, L [m]	0.6	0.6	0.8	0.3
Fiber outer diameter, D_o [μm]	250	250	180	200
Fiber inner diameter, D_c [μm]	200	200	126	150
Number of fibers, N_T	34,015,497	131,372,898	2,805	106
Viscosity, μ [Pa s]	13.0	14.9	14.9	11.05

Table 4.4: Feed and sweep gas compositions [mol-%] for the spiral-wound Pebax® co-polymer membrane in the 1st stage and the hollow fiber polyimide derived carbon membrane in the 2nd stage.

	1st stage Pebax® co-polymer		2ns stage carbon membrane
Component	Feed gas concentration [mol-%]	Sweep gas concentration [mol-%]	Feed gas concentration [mol-%]
Methane	77.81	4.79	80.54
Ethane	7.05	0.10	6.43
Propane	3.02	0.04	2.46
Butane	1.91	0.02	1.52
C_{5+}	0.10	0.001	0.07
CO_2	10.07	95.04	8.98
H_2O	0.03	0.005	0.0003

4.4.1 HYSYS integration

Aspen Hysys is a Cape-Open compliant simulation environment. Thus, after simulation in Matlab, the membrane model was integrated in Hysys as a Cape-Open COM unit operation with license provided by AmsterCHEM. Additionally, the COM-Thermo property package was selected as it provides the Peng-Robinson equation of states which is in agreement with the selected fluid package for the Hysys flowsheet. For more information on the Cape-Open standard and Matlab unit operation, the reader is referred to the websites of CO-LaN [60] and AmsterCHEM [61].

Chapter 5

Optimization

In this chapter the optimization problem is defined, and the strategy and method for solving the problem is presented.

5.1 Design configurations

To begin with, the different design configurations which are subject to optimization are introduced. The results from the specialization project showed that the capital related cost of the compressors and the turbine was much larger than the cost of membranes [25]. Therefore, it was suggested to replace the turbine with a pressure relief valve and to consider reducing the number of compressors. Moreover, the temperature dew point constraint was very strict; 3.5 ppm water, corresponding to a water vapour dew point of $-52\,^{\circ}\mathrm{C}$, and the pressure losses over the interstage coolers in the multistage compressor were set much lower than what is common as described in Section 2.3.1. The suggested changes were implemented in the design and optimized successively as to observe the effect on the specific SNG cost.

Configuration 1
The first design configuration to be optimized is the initial design presented in Chapter 3. This design is the same that was simulated in the specialization project, but differs in that a different approach for optimization was taken. The purpose is to compare the method of optimization. The optimization routine described in Chapter 5 is expected to give a better optimum because it optimizes the objective function with respect to all decision variables simultaneously and not in sequence. The successive method restricts the solution space by the previously calculated variable instead of finding the simultaneous combination of values that minimize the objective function.

Configuration 2
In the second configuration the turbine, K-4, is replaced with a pressure relief valve which is is expected to have negligible effect on the cost function as it costs substantially less.

Configuration 3
In the third configuration the compressor, K-2, is removed. The multistage compressor will require more power, and be larger than the previous configuration. However, the work that was required in K-2, will be evened out over the three stages. Therefore the operating cost is not expected to be considerably affected by this change. A reduction in the capital related cost can also be expected to be observed. Additionally, since the 3$^{\text{rd}}$ stage feed pressure must be 60 bar, the separation performance is expected to improve and the membrane area expected to be reduced.

Configuration 4
In the fourth configuration only two compressors in the multistage compressor, K-1, remain. The cost is expected to increase.

Configuration 5
In the fifth configuration, K-2 is added to configuration 4. The objective function value is expected to be lower than the previous one.

Configuration 6
Lastly, in the sixth configuration, the ChemBrane membrane model is replaced by the membrane model described in Chapter 4. It is expected to increase the optimum as the model is expected to demonstrate a more realistic performance, which means that the membrane areas will be larger to meet the requirements.

5.2 Problem definition

The optimization problem can be defined as

$$\begin{aligned} &\min_{\mathbf{x}} \; C_s(\mathbf{x}) \\ \textbf{s.t} \;\; &g_i(\mathbf{x}) \leq 0, \quad i \in \mathrm{I} \\ &g_i(\mathbf{x}) = 0, \quad i \in \mathrm{E} \end{aligned} \tag{5.1}$$

where g is the set of equality and inequality constraints, E and I, respectively, and $\mathbf{x}$ is the set of decision variables .

5.2.1 Objective function

The objective function is defined as the cost function

$$C_s = \frac{C_{OPEX} + C_{CRC}}{\dot{V}_{SNG}} \quad \left[\frac{\$}{\text{year}}\right] \tag{3.1 revisited}$$

where C_s is the annual specific cost of sweet natural gas processing, C_{OPEX} and C_{CRC} are the annual operating cost and annual capital related cost, respectively, and $\dot{V}_{SNG}$ is the amount of produced sweet natural gas (SNG).

5.2.2 Decision variables

The membrane and compressor/turbine annual capital related cost, C_{CRC} and annual operating cost, C_{OPEX} in Equation (3.1) depend on the size and utility of the equipment, which is further dependant on certain process parameters. Along with the membrane areas, the identified process parameters make up the set of decision variables, $\mathbf{x}$ as listed here

- 1st stage area [m^2]
- 2nd stage area [m^2]
- 3rd stage area [m^2]
- 2nd stage permeate pressure, $P_{\text{P-2}}$ [bar]
- 3rd stage feed pressure, $P_{\text{F-3}}$ [bar]
- 3rd stage permeate pressure, $P_{\text{P-3}}$ [bar]
- 3rd stage permeate recycle ratio, $\alpha_{\text{P-3}}$
- K-100 discharge pressure, p_1 [bar]
- K-101 discharge pressure, p_3 [bar]
- K-102 discharge pressure, p_5 [bar]

5.2.3 Constraints

The constraints have been described in Chapter 3, Section 3.4, but are recapitulated here. The water dew point in the 1st stage retentate, R-1, must be lower than −40 °C, the produced CO_2 for enhanced oil/gas recovery must have a purity of minimum 95 mol-%, and the CH_4 purity of the sweet natural gas in the 2nd stage retentate must be minimum 97.5 mol-%. Since there will be a small amount of lighter hydrocarbons present in the product, the methane purity is considered as the total amount of hydrocarbons. These constraints are identified as the inequality constraints, $\mathbf{g}_{\text{I}}(\mathbf{x})$. Additionally, the pressure ratio over each stage in the multistage compressor, K-1, must be equal. This is an equality constraint, $\mathbf{g}_{\text{E}}(\mathbf{x})$.

5.3 The optimization method

Both the objective function and constraints are non-linear, thus the derivative free Nelder-Mead simplex algorithm was chosen to solve the optimization problem defined in Equation (5.1). The MATLAB Optimization Toolbox™ contains several solvers for solving optimization problems of which the function `fminsearch` uses the Nelder-Mead simplex algorithm [62]. However, the optimization problem is constrained, and as stated in Section 2.2.3, the Nelder-Mead simplex algorithm solves unconstrained problems. Consequently, the function `fminsearchbnd` was used, which is a modification of the `fminsearch` solver. It allows for bounded variables and for exactly fixed variables. To allow for constraints, a penalty function, P, was introduced to translate the constrained optimization problem to an unconstrained problem

$$P(\mathbf{x}) = \mu \sum_i \max(0, c_i(\mathbf{x})) \tag{5.2}$$

where μ is a penalty constant set to $1 \cdot 10^6$ for this optimization problem, and $c_i(\mathbf{x})$ is the difference between the target value, t_i and the constraint, g_i

$$c_i(\mathbf{x}) = t_i(\mathbf{x}) - g_i(\mathbf{x}) \tag{5.3}$$

The objective function then becomes

$$f(\mathbf{x}) = C_s(\mathbf{x}) + P(\mathbf{x}) \tag{5.4}$$

During the optimization routine, it was discovered that the 1st stage area was increased by the optimizer to an extent where the amount of water in the retentate, R-1, was ~ 0 ppm according to the HYSYS flowsheet calculation. The increasingly large area, lead to major losses in the 1st stage permeate and a low production rate, which is in contrast to the objective for the process. Consequently, the constraint on the amount of water in the 1st stage retentate was formulated as an equality constraint with a small margin. For that particular constraint, the penalty function was defined as follows

$$P(\mathbf{x}) = \mu \max(0, c(\mathbf{x})^2) \tag{5.5}$$

The pattern-search algorithm, also a derivative free method, was tested for the first configuration by using the solver `patternsearch`. The solver accounts for the constraints through a separate constraint function. Thus the objective function for the pattern-search algorithm did not include the penalty function as defined in Equation (5.2). However, a penalty function is included and calculated in the solver algorithm. The objective function was otherwise identical to the objective function used in the Nelder-Mead method. The purpose was to compare the performance of the two algorithms.

5.3.1 The optimization program and HYSYS integration

The optimization program created in Matlab, takes the decision variables and the lower and upper bounds as arguments, and scales the variables so that $\mathbf{x} \in [0,1]$ which are then passed on to the solver `fminsearchbnd`. The solver evaluates the objective function value at that point (set of decision variables). If the optimum is reached, the program ends. If not, the decision variables are updated and new function values calculated. As long as the solver runs, an output function stores the results after each iteration. This is achieved by defining both the objective function and output function as nested functions.

The initial values of the decision variables, and the lower and upper bounds are exported from Hysys to the optimizer. The optimizer exports the updated values of these to Hysys. The constraints and the cost functions, except for the objective cost function, is calculated in and exported from Hysys to the optimizer to be used in the objective function. Furthermore, the interstage discharge pressures are calculated and the last stage discharge pressure (see Figure 3.3) is updated in the optimizer and exported to Hysys. The procedure is illustrated schematically in Figure 5.1 and the optimization program is available in Appendix E.1.

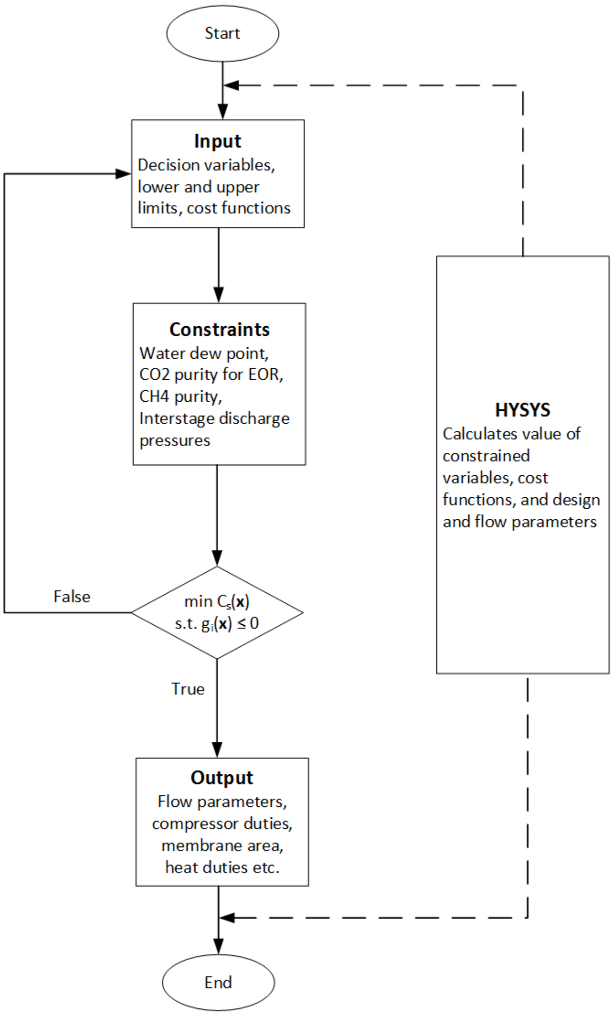

Start
Input
Decision variables, lower and upper limits, cost functions
Constraints
Water dew point, CO2 purity for EOR, CH4 purity, Interstage discharge pressures
min $C_s(\mathbf{x})$
s.t. $g_i(\mathbf{x}) \leq 0$
False
True
Output
Flow parameters, compressor duties, membrane area, heat duties etc.
End
HYSYS
Calculates value of constrained variables, cost functions, and design and flow parameters

Figure 5.1: A schematic representation of the optimization routine.

5.3.2 HYSYS set up

The cost functions, the decision variables and the constraints are all gathered in Hysys spreadsheets presented in Appendix C.1, which define the input/output structure between Hysys and the optimization program. Instead of importing and passing variables to each object (material streams, energy streams and unit operations), the variables are easily accessed through one object. Additionally, it offers a way of monitoring the changes during optimization that is comprehensible and structured.

The spreadsheets in Hysys are

Input
The decision variables are handled by this spreadsheet. It contains the updated values calculated in the optimizer, the initial values of the decision variables used as the starting point for the optimization routine, as well as the lower and upper bounds.

Constraints
This spreadsheet handles the constraints. The constraint values are set here and the targets calculated in Hysys are presented and imported to the optimizer for evaluation.

Cost
The costs are handled in this spreadsheet. All costs are calculated based on the plant simulation result of which the capital related cost and the operating cost are exported from Hysys to the optimizer where the objective cost function is calculated. It is also calculated in the spreadsheet for the purpose of monitoring in a graphical user interface.

The connection between Hysys and Matlab is handled by the Component Object Model (COM) in ActiveX which exposes the Hysys objects so that the Hysys simulation can be controlled from Matlab, and variables can be accessed and treated in the optimization program. The interested reader is referred to the Aspen Hysys Custumization Guide [63] for more information on customized unit operations in Hysys.

5.4 Optimization problem

The starting point, i.e. the initial values of the decision variables for optimization of design configuration 1 were set based on the results from the optimization in the specialization project [25] as presented in Table 5.2. The constraint on the water vapour dew point in the 1st stage retentate, was translated to a constraint on the amount of water vapour (mol-%). In addition, as the methane purity in the produced sweet natural gas was considered as the total hydrocarbon purity, the constraint was formulated as a constraint on the amount of CO_2 (mol-%) as shown in Table 5.1. In the calculations, the latter constraint was multiplied by -1 to be on the same form as the other constraints as in Equation (5.1)

Because the pressure drop over the cooler prior to the 3rd stage was set, the 3rd stage feed pressure became determined by the discharge pressure in K-102, p_5. Consequently, the feed pressure was eliminated as a decision variable. Moreover, the interstage discharge pressures, p_1 and p_3, were eliminated as decision variables by eliminating the equality constraint specified in Section 5.2. An equal pressure ratio was still attained by setting the pressures directly in the objective function, and updating the K-102 discharge pressure, p_5, subsequently. Eliminating constraints and decision variables simplified the optimization problem, with the purpose of potentially reducing the computational effort in solving the problem.

For design configurations 3-5, the K-102 discharge pressure, p_5, was eliminated as a decision variable because the pressure had to be set to a constant pressure at 60.05 bar. In addition, the lower bound was reduced to 5000 as the increased pressure ratio was expected to reduce the required membrane area. In design configuration 5, compression of the 3rd stage retentate, R-3, was implemented. Thus p_5 was reintroduced to the set of decision variables.

For design configuration 6, with the membrane model integrated in Hysys as Cape-Open unit operations, there were some challenges with the execution of the optimization routine. The optimizer failed to open the Hysys simulation case, for which it was attempted to catch errors to investigate the reason. However, it remained a challenge to successfully open the simulation case, and it required a greater understanding of the Aspen Hysys software in connection with Cape-Open unit operations and COM technology to resolve. Thus, design configuration 6 was not optimized in this work.

Table 5.1: Constraints in the optimization problem.

Description	Value
Water in 1st stage retentate	≤ 3.5 ppm
CO_2 in 2nd stage retentate	≤ 2.5 mol-%
CO_2 in 3rd stage retentate	≥ 95 mol-%

Table 5.2: Initial values, lower and upper bounds for the decision variables in the optimization problem.

Decision variable	Initial	Min.	Max.
2nd stage permeate pressure, $P_{\text{P-2}}$ [bar]*	2.65	1.00	20.0
K-102 discharge pressure, p_5 [bar]	13	2	60
3rd stage permeate pressure, $P_{\text{P-3}}$ [bar]*	2.2	1.1	4.9
3rd stage permeate recycle ratio, $\alpha_{\text{P-3}}$	0.11	$1 \cdot 10^{-4}$	$9.999 \cdot 10^{-1}$
1st stage area [m^2]	$1.57 \cdot 10^4$	$1.0 \cdot 10^4$	$5.0 \cdot 10^5$
2nd stage area [m^2]	$8.23 \cdot 10^4$	$1.0 \cdot 10^4$	$5.0 \cdot 10^5$
3rd stage area [m^2]	$3.88 \cdot 10^4$	$7.0 \cdot 10^3$	$5.0 \cdot 10^4$

*In the HYSYS simulation shown in Appendix B, the pressure is determined by the sweep gas pressure in S2 and S3, respectively. The sweep gas pressure and the permeate pressure over each stage is however equal for the ChemBrane model.

Chapter 6

Results and Discussion

In this chapter the results of the optimization of all the design configurations and the membrane model performance is presented and discussed.

6.1 Optimization

The aim of the optimization routine for every design configuration was to minimize the specific cost of sweet natural gas subject to a set of constraints. The objective function was dependent on the annual capital related cost and operating cost and the amount of sweet natural gas produced, which was further determined by the equipment costs and the process parameters. The results of the optimized design configurations are summarized in Figure 6.1 where the costs are presented. In Table 6.2, the decision variables, interstage pressures, and constrained process parameters are presented in addition to the compressor duties, costs and methane production and recovery. The results reveals a greater trend, that the removal of compressors reduces the costs and results in a better optimum. Overall, the constraints were met for all design configurations and the interstage pressure ratios were set correctly to give an equal pressure ratio over the multistage compressor.

Design configuration 1 and the optimizer

Design configuration 1 was equal to the previously studied design in the specialization project, and was optimized with the purpose of comparing the optimization method. The minimum cost of sweet natural gas was found to be $2.448 \cdot 10^{-3}$ $/m^3 which is lower than the optimum from previous work at $2.483 \cdot 10^{-3}$ $/m^3. It therefore suggests that the optimization routine described in this thesis is a better method for finding the optimum. This was expected as the design variables were optimized simultaneously and not successively. Another advantage is that

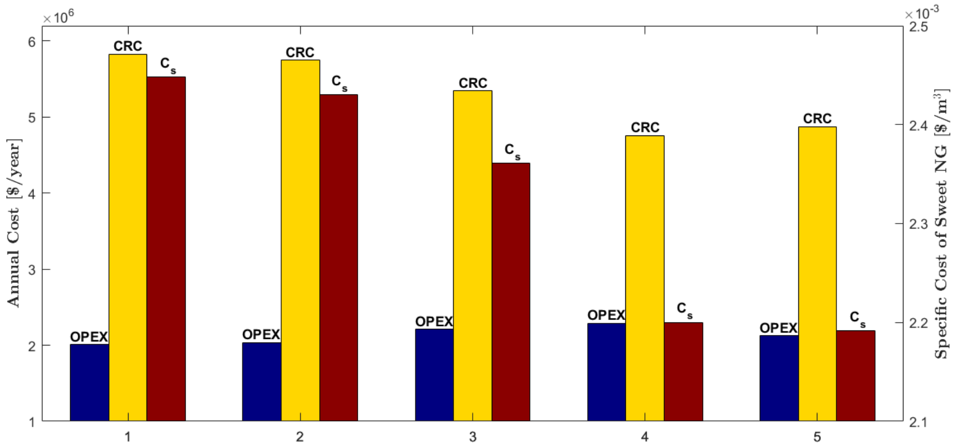


Figure 6.1: The annual capital related cost, operating cost and the specific cost of sweet natural gas processing in each of the optimized design configurations.

the optimizer is less time consuming. Solving the optimization problem for design configuration 1 using the optimizer took one hour. Another derivative-free method was tested for comparison, the pattern-search method, which showed a better optimum at $2.098 \cdot 10^{-3}$ \$/m^3, but the solver spent more than two hours to solve the optimization problem. Moreover it had a substantial larger amount of function evaluations per iteration. However, the pattern-search algorithm adjusted and modified the penalty function depending on the function evaluations, while in the penalty method applied in the Nelder-Mead algortihm, the penalty function remained the same throughout the computation. Neither of the methods guarantee a global optimum. The Nelder-Mead simplex algorithm was chosen for optimization of the subsequent design configurations as it was the most efficient.

Design configuration 2

Design configuration 2 was optimized having removed the turbine. As shown in Table 3.4 (Page 30), the effect on the annual operating cost was not significant, and the reduction in the compressor/turbine investment cost was greater than the increase in the annual operating cost. Additionally, the cost of the membranes was reduced and thus the specific cost of natural gas was reduced from configuration 1 to 2 at $2.430 \cdot 10^{-3}$ \$/m^3. The reduced compressor cost can be explained by the removal of the turbine and because there was little change in the required compressor sizes from design 1 to 2. Although the compressor cost decreased, the difference was not as large as for the other configurations. This is further explained by the turbine size, which based on the power generation in design configuration 1, was small compared to the total compressor power consumption. Thus the turbine size and cost was lower than the compressor sizes and cost. The membrane cost was also reduced because the total membrane area decreased due to an increased partial pressure gradient in the sweetening unit.

Design configuration 3

As mentioned in Section 5.4, the lower bound in configuration 3 to 5 was lower than for configuration 1 and 2. It should be noted that it was not necessary for configuration 5, but it did not affect the optimization routine. The reason for altering the lower bound was because the 3rd stage feed pressure was set to 60 bar, thus the pressure ratio and the driving force for permeation increased requiring a smaller membrane area. The optimization showed that the area was immediately set to be very close to the lower bound. To ensure that the lower bound did not further constrain the optimization problem, the lower bound was lowered. Finally, the optimum of design configuration 3 was found to be $2.361 \cdot 10^{-3}$ \$/m^3 which was lower than the optimum in design 2. The main reason is that one compressor was removed, thus the compressor cost and annual capital related costs were reduced. The total power consumption in configuration 3 was 548.9 kW more than in design configuration 2, which slightly increased the annual operating cost. However, the compressor investment cost was significantly reduced because the power consumption from the compressor that was removed, was distributed evenly over each stage in the multistage compressor. This further caused only a minor increase in the multistage compressor sizes and cost which was less than the size and cost of keeping the compressor that was removed. It should also be noted that the pressure ratio over each stage in the multistage compressor was 2.7, which is closer to recommended pressure ratios than for the previous designs. Additionally, the 3rd stage area was reduced so that the total membrane cost was almost 1 million \$ less than in design 2. The total annual capital related cost was therefore noticeably lower than in design configuration 2.

Design configuration 4 and 5

Removing yet another compressor, resulted in an even better optimum in design configuration 4 than in the previous ones. However, because of the 3rd stage feed pressure being at 60 bar, the pressure ratio over the remaining two compressors in series was 4.43 which is the highest pressure ratio attained of the different design configurations. It need not be a problem, it simply requires more intercooling, but it also means that the required power and size of the compressors are higher. Similar to design configuration 3, design configuration 4 proves that it is economically favourable to reduce the number of compressors.

In design configuration 5, the compressor after the 3rd stage was reinstalled, resulting in the optimal design configuration with respect to the objective. This contradicts the initial assumption that as few compressors as possible yields the best design. However, the compressor, K-3, reduces the required power with about 700 kW for each of the compressors in series, only using 892 kW itself. A substantial increase in the 3rd stage area is also observed, but a decrease in the 2nd stage area is simultaneously taking place because the associated permeate pressure is allowed to be lower as the compressor power requirements are relieved by the inserted compressor K3. The total membrane cost is the lowest observed for all the configurations, and the compressor cost next to lowest which explains the low annual capital related cost. Although the annual capital related cost is at its minimum in design 4, the annual operating cost is at its highest. Additionally, configuration 5 has the highest loss of methane which means that there is produced less sweet natural gas, However, the difference between configuration 5 and the other configuration is about 40 kmol/h which is very little and not of pronounced significance for the result. Revisiting Equation (3.1) (Page 27) and taking all this into account explains the final annual specific cost of natural gas processing in design configuration 5.

Design configuration 6 and gas dew point

The membrane model was successfully integrated in Hysys as shown in Appendix C.3, but it was not optimized. The results of the membrane modelling and design configuration 6 is presented in more detail in Section 6.2. Design configuration 6 was identical to design configuration 5 except for the membrane model and user operation interface. However, the constraint on the methane purity in the produced sweet natural gas was violated slightly being 97.3%, while the water dew point was −61.6 °C and the CO_2 purity in the gas for enhanced oil/gas recovery was 95.3%. It should be mentioned that the lower water dew point corresponded to a larger amount of water than in the previous design configurations. It was 6.2 ppm. Moreover, a gas dew point was not obtained at the sweet natural gas conditions.

For design configuration 5, the envelope of the produced sweet natural gas was obtained in the Aspen Hysys simulation case and is presented in Appendix C.4. From the envelope the gas dew point was obtained at −11.5 °C at 60 bar which is within the pipeline specification presented in Section 2.5. Moreover, the cricondentherm, cricondenbar and the two-phase critical temperature and pressure was obtained. The parameters as presented in Table 6.1

Table 6.1: The gas dew point and critical values for the produced sweet natural gas obtained in the HYSYS simulation case for design configuration 5.

Parameter	Value
Gas dew point [°C]	-11.5
Cricondentherm [°C]	-10.2 (@ 47.3 bar)
Cricondenbar [bar]	83.9 bar (@ −30.9 °C)
Two-pase critical temperature [°C]	-49.9
Two-pase critical pressure [bar]	73.6

Table 6.2: The results of the optimization for each design configuration.

Parameter	Design configuration				
	1	2	3	4	5
Decision variables					
1st stage area, [10^4 m^2]	1.573	1.574	1.579	1.572	1.603
2nd stage area, [10^4 m^2]	8.089	8.045	8.690	8.785	6.191
3rd stage area, [10^4 m^2]	3.460	3.324	0.7488	0.7678	2.482
2nd stage permeate pressure, p_{P-2} [bar]	2.562	2.547	3.064	3.072	1.492
3rd stage feed pressure, p_{F-3} [bar]	14.92	15.28	60.00	60.00	21.78
3rd stage permeate pressure, p_{P-3} [bar]	2.640	2.610	3.650	4.775	4.898
3rd stage recycle ratio (TEE-100) [%]	11.37	11.06	8.774	11.75	1.059
Interstage pressures					
K-100 discharge pressure, p_1 [bar]	4.640	4.659	8.284	13.61	5.733
K-101 discharge pressure, p_2 [bar]	8.315	8.431	22.26	-	-
Constrained variables					
Water content (dew point) [ppm]	3.50	3.50	3.50	3.50	3.49
CH_4 purity in sweet natural gas [mol-%]	95.01	95.07	95.00	95.00	95.04
CO_2 purity for EOR/EGR [mol-%]	2.50	2.50	2.50	2.50	2.50
Compressor/Turbine duty, [kW]					
K-100	1310	1326	2243	3595	2854
K-101	1325	1342	2235	-	-
K-102	1306	1322	2120	3517	2877
K-103	1668	1618	-	-	892.3
K-104	-68.57	-	-	-	-
K-105	741.2	756.5	315.4	43.49	24.13
Product and loss					
CH_4 loss [%]	12.49	12.50	12.54	12.49	12.70
Sweet natural gas [10^4 kmol/h]	1.787	1.787	1.786	1.787	1.783
Costs					
Membrane skid cost, C_M [10^6 \$]	6.561	6.471	5.509	5.562	5.138
Compressor/turbine cost, C_{TM} [10^7 \$]	2.258	2.226	2.122	1.822	1.924
Operating cost, C_{OPEX} [10^6 \$/year]	2.010	2.036	2.212	2.290	2.127
Capital related cost, C_{CRC} [10^6 \$/year]	5.829	5.747	5.346	4.757	4.876
Specific cost of sweet natural gas processing, C_s [10^{-3} \$/year]	2.448	2.430	2.361	2.200	2.192

Summary

In general, the results show that the 1st stage membrane area is dependent on the recycle ratio of the sweep gas as the feed and permeate pressure is held constant. The partial pressure gradients thus depend on the sweep gas flow rate, i.e. the sweep gas recycle ratio. Consequently, the area decreases with lower recycle ratio because the partial pressure gradient is reduced. Although the 2nd and 3rd stage areas are affected by the feed flow rate, the result shows the dependence on the feed and permeate pressure ratios. The reason is that the feed flow rates do not vary significantly from configuration to configuration, thus it is primarily the pressures of the feed and permeate that affect the membrane areas in the two sweetening stages. The methane loss is dependent on the 1st stage area. As presented in Table 6.2, the design which gives the highest methane loss also have the largest membrane area in the 1st stage, which is design configuration 5.

It was specified in the process description that a methane loss lower than 2% was desired. According to results from previous work [25], increasing the sweep gas flow rate would increase the partial pressure gradient and amount of permeated water. However, recycling most of the 3rd stage permeate as sweep gas, only reduced the methane loss to about 11.5%. The result suggested that to attain a better methane recovery, the membrane performance must be improved or another design must be investigated. Moreover, a larger recycle ratio would decrease the 1st stage area which would reduce the membrane cost. However, the optimizer did not go in the direction of increasing the recycle ratio. A possible explanation to that could be that the membrane cost was not reduced to a significant degree to affect the objective function. Furthermore, a low recycle ratio also meant a lower production of sweet natural gas, reducing the 2nd stage area, and the associated permeate flow rate to be compressed, thus affecting both the membrane cost and compressor costs. To better account for the methane loss in the optimization routine it is suggested to define the methane loss either as a constraint or as an objective function. The first would imply a different design than the one that has been studied in this work because the constraint of maximum 2% methane loss is unattainable in the hybrid system. The latter, defining the methane loss as an objective function would transform the optimization problem into a multiobjective problem.

An important remark is that the membrane cost was based on the effective membrane area and not the total area of the module. Yuhan et al. estimated a module area twice the effective area [57]. Accounting for this would lead to the membrane cost having a greater impact on the objective function and further result in a different optimal design. Additionally, the membrane cost is assumed to be equal for both the Pebax® co-polymer membrane and the carbon membrane which might not be realistic as the spiral-wound membranes are more expensive to manufacture as presented in Section 2.4.5. An increased membrane cost would increase the objective function and the membrane areas could possibly be more central in the optimization problem. Although it has not been considered in the cost model, the coolers, vessels and heat exchangers add to the cost of the plant. Consequently, for more detailed evaluation of the process design, all equipment must be included in the cost calculations.

6.2 Membrane model

The initial membrane model was solved using the Newton-Rhapson method in MATLAB with the function `fsolve`. First, an evaluation of the carbon membrane conditions with respect to sweep gas and fiber side boundary pressure was conducted. Second, the results of both the membrane in the dehydration unit and the sweetening unit were compared to the ChemBrane model at process conditions obtained in design configuration 5. Lastly, the membrane model performance was compared to experimental data and the mollocator membrane model.

6.2.1 Analysis of the carbon membrane conditions for process simulation

The results of the carbon membrane evaluation is presented in Table 6.3, which shows the output parameters with respect to the solver performance and the membrane performance. A graphical representation of the results are available in Appendix D. Solving the membrane with sweep gas and defined permeate pressure (scenario 1) showed that the model was solved with only 4 iterations, and a very small residual with the solver considering that the equation was solved. Similar results were obtained in the solution for the membrane with sweep gas and defined sweep gas pressure (scenario 2). For the membrane without sweep gas (scenario 3), the solution was obtained at 17 and 20 iterations at 12 and 32 collocation points, respectively. The residual was much larger compared to scenarios 1 and 2. The solver stopped because it met the criteria for stopping, but the equation was not considered as solved. However, for both cases with and without sweep gas, the computations were carried out quickly and there was not much difference in the run time for equal number of collocation points.

Generally, for all scenarios, increasing the number of collocations points increases the run time. However, for scenario 3, the effect of increasing the number of nodes was lower compared to the other scenarios. This can be explained by the lower accuracy and it is a possibility that it is partially due to the CPU and memory capacity at the time. For scenario 3, it is observed a larger residual and larger difference in the mass balances which is 10^3 and 10^4 times larger, respectively, for the case of 32 nodes compared to case of 12 nodes. As such, the accuracy decreases with increasing number of nodes. Moreover, the decreased accuracy affects the stream parameters for which it is observed a larger permeate flow, increased CH_4 purity in the retentate and increased CO_2 purity in the permeate from 12 to 32 nodes. For scenario 1 and 2, the residue is 10^3 and 10^2 times smaller, respectively, for the case of 32 nodes. Consequently, the accuracy increases with increasing number of collocation points. It is further observed that the difference in the mass balances is somewhat larger in the case of 32 nodes compared to the case of 12 nodes. However, the difference is negligible and the total mass balance, and the absolute difference between the shell side and fiber side mass balances remain significantly small at the order of 10^{-11}. For scenario 1 and 2, it is sufficient with 12 nodes with respect to accuracy and it is favourable in terms of efficiency.

Scenario 2 shows the best membrane performance as more is permeated and a higher purity of CH_4 and CO_2 is obtained in the retentate and permeate, respectively, compared to the other scenarios. This is due to the lower permeate pressure causing a greater partial pressure gradient. However, considering the scenarios with respect to the process design eliminates scenario 2 because there is a suction pressure on the permeate side for both the 2nd and 3rd stage. In addition, as mentioned in Chapter 4, the motivation for using sweep gas was for potential ease of numerical calculations in the process simulation, and not to simulate an actual flow of sweep gas. That is why the sweep gas flow rate was set to practically zero. Consequently, the permeate pressure should be defined as a boundary condition. Moreover, as described in Chapter 5, the permeate pressure affects the operating cost and the capital cost of the compressors which further affects the objective function. Thus it is identified as a decision variable in the optimization routine, which further reinforce the suggestion of defining the permeate pressure as a boundary condition.

Considering the scenarios with respect to integration in Aspen HYSYS, certainly the efficiency is crucial as numerous calculations and iterations take place within the HYSYS simulation case. Also, regarding the optimization routine, which is dependant on the HYSYS calculations, it is imperative that the model calculations are carried out as fast as possible. For the design configurations containing the ChemBrane model, the optimization scheme took about one hour, which could be significantly increased with the implementation of the membrane model. In that case, scenario 1 or 3 with 12 collocation points would be favourable. Regarding the accuracy, scenario 1 should be chosen for the carbon membranes. However, implementation of the membrane model as described in scenario 1 requires that the sweep gas pressure is set after calculating the pressure value. When the sweep gas pressure is set, the model is triggered to run again. Consequently, the membrane model at conditions described in scenario 1 require two runs rather than one. Therefore, scenario 3 with 12 nodes is a viable option. However, the inaccuracy in scenario 3 can cause strain on the calculation in Hysys as well.

The model was successfully integrated in Hysys as Cape-Open unit operations. However, there were some issues handling the flowsheet. First, there was an issue with opening the simulation case containing the Cape-Open unit operations. Therefore, optimization of the Hysys flowsheet containing the membrane model was not performed. Such a study would have been valuable in order to evaluate the computational cost of solving the the optimization problem, thus indicating the ease of calculations in Hysys for each of the viable scenarios (1 and 3). As a result, it would have been possible to further evaluate which scenario for the carbon membrane that is best suited for integration in the flowsheet and perhaps arrive at a conclusion. Second, attempting to use adjust blocks on the Cape-Open unit operations did not converge for different starting points and numerical settings. Third, it was not possible to set the permeate temperature in a similar manner as for the ChemBrane model as shown in Equation (3.7). Therefore, the handling of Cape-Open unit operations in Hysys came across as more challenging compared to the ease of handling the ChemBrane unit operation. Therefore, a further investigation on the Aspen Hysys software in relation to the Cape-Open unit operations and COM technology is necessary for future work.

Table 6.3: The results of simulating the carbon membrane in the 2nd stage at operating condition as specified in design configuration 5, with and without sweep gas and defined permeate pressure or sweep gas pressure.

Parameter	Scenario 1		Scenario 2		Scenario 3	
	nz=12	nz=32	nz=12	nz=32	nz=12	nz=32
Solver performance						
Iterations	4	5	4	5	17	20
Time elapsed [s]	0.35	1.20	0.35	1.17	0.35	0.97
Residual	$1.732 \cdot 10^{-22}$	$3.114 \cdot 10^{-26}$	$9.587 \cdot 10^{-25}$	$3.374 \cdot 10^{-27}$	$3.374 \cdot 10^{-7}$	$5.370 \cdot 10^{-4}$
Membrane performance						
Sweep gas pressure, [bar]	1.557	1.557	1.492	1.492	1.557	1.557
Permeate pressure [bar]	1.492	1.492	0.892	0.892	1.492	1.492
Retentate [kmol/h]	$1.790 \cdot 10^{4}$	$1.790 \cdot 10^{4}$	$1.782 \cdot 10^{4}$	$1.782 \cdot 10^{4}$	$1.790 \cdot 10^{4}$	$1.790 \cdot 10^{4}$
CH_4 in retentate [mol-%]	85.92	85.92	86.30	86.30	85.92	85.94
Permeate [kmol/h]	$1.959 \cdot 10^{3}$	$1.959 \cdot 10^{3}$	$2.043 \cdot 10^{3}$	$2.043 \cdot 10^{3}$	$1.959 \cdot 10^{3}$	$1.963 \cdot 10^{3}$
CO_2 in permeate [mol-%]	66.53	66.53	67.64	67.64	66.53	66.60
Mass balance shell side [kmol/h]	$-1.959 \cdot 10^{3}$	$-1.959 \cdot 10^{3}$	$-2.043 \cdot 10^{3}$	$-2.043 \cdot 10^{3}$	$-1.959 \cdot 10^{3}$	$-1.963 \cdot 10^{3}$
Mass balance fiber side [kmol/h]	$1.959 \cdot 10^{3}$	$1.959 \cdot 10^{3}$	$2.043 \cdot 10^{3}$	$2.043 \cdot 10^{3}$	$1.959 \cdot 10^{3}$	$1.963 \cdot 10^{3}$
Absolute difference [kmol/h]	$7.731 \cdot 10^{-12}$	$7.230 \cdot 10^{-11}$	$2.547 \cdot 10^{-11}$	$6.867 \cdot 10^{-11}$	$1.997 \cdot 10^{-9}$	$3.159 \cdot 10^{-5}$
Absolute total mass balance [kmol/h]	$7.072 \cdot 10^{-12}$	$7.256 \cdot 10^{-11}$	$2.526 \cdot 10^{-11}$	$6.892 \cdot 10^{-11}$	$1.997 \cdot 10^{-9}$	$3.159 \cdot 10^{-5}$

6.2.2 Model validation

For the purpose of comparing modMB with ChemBrane, scenario 3 was selected for the carbon membrane. The Pebax® co-polymer membrane and the carbon membrane was simulated with the membrane dimensions and operating conditions as specified in Tables 4.3 and 4.4 (Pages 43 and 44). The results of the four different simulation cases are presented in Table 6.4.

Case 1 and 2

For case 1 it is observed a larger retentate flow rate and smaller permeate flow rate in modMB than in ChemBrane. Consequently, the permeation flux is smaller which is due to the reduced partial pressure gradient. As modMB accounts for pressure drop in the fibers, the fiber side pressure at the retentate end is higher in modMB than in ChemBrane where the fiber side pressure is constant at 1 bar. In this case, the pressure at the retentate end in modMB was calculated to be 1.48 bar corresponding to a pressure loss of 0.48 bar along the fiber length. The absolute difference between the ChemBrane and modMB retentate flow rate and permeate flow rate is of the same order and close to equal, but the relative deviation is larger for the permeate side parameters. This is because the flow rate is lower on the permeate side, i.e. a smaller amount of the less permeable species have permeated. The observations in case 2 are similar to that of case 1. The sweep pressure is calculated to be 1.56 bar corresponding to a pressure loss of 0.07 bar along the fiber length. Consequently, the pressure gradient in modMB is closer to the pressure gradient in ChemBrane which can explain the smaller relative deviations in case 2 compared to case 1 where the pressure drop is larger. For both case 1 and 2 the relative deviation is ≤ 5%, suggesting that the modMB model is a valid model compared to ChemBrane. It should be noted that the deviation might be larger or smaller dependent on the fiber side pressure drop. The fiber side pressure drop is further dependent on the module dimensions such as number of fibers, the inner diameter and fiber length, as shown in Equation (4.3).

Case 3

For case 3 it is observed that the retentate flow rate and the permeate flow rate is lower and higher, respectively, compared to the experimental data. The permeate flow rate is almost twice as large as the permeate flow rate obtained in the experiment. Additionally, the methane purity in the retentate and the CO_2 purity in the permeate are higher than the experimental data. Consequently, the performance obtained in the modMB is much better than that obtained in the experimental data. There are several possible explanations for the deviations. First, observing the deviation between the mollocator and the experimental data shows that the mollocator is closer to the experimental data than modMB. The mollocator accounts for the pressure loss on the shell side as well as on the fiber side. Consequently, the partial pressure gradient is smaller than in the modMB, reducing the permeate flux and thus the purities. Second, it could be that there was negligible pressure drops on both sides due to the low flow rates and that the real performance was not as good as the initial observations due to ageing effects which is prominent in thin films, as described in Section 2.4, or it could be a combination of ageing effects and pressure drop. In the case of constant pressures, the performance should have been closer to or better than the modMB and mollocator. In the case of ageing effects, more information about the tested module is required. Moreover, there is an experimental error which was expected to be 5%. This indicates that there is expected a small deviation between the model and the experimental data. Additionally, as shown in the previous section, the accuracy in the case of no sweep gas is low, which introduces an error in the results obtained in the modMB. However, the large relative deviation in case 3 suggests that the modMB is not suited for simulating gas transport behaviour through hollow fiber membranes.

Case 4

For case 4, the modMB yields a better membrane performance compared to the experimental data, but the differences are small corresponding to a relative deviation of ≤ 3.65%. This is somewhat larger than the deviation between the mollocator and the experimental data. As mentioned, some deviation is expected due to the experimental error, but the relative deviations are small, suggesting that the modMB is suitable for simulation of gas transportation through hollow fiber membranes.

Generally, there are uncertainties related to the instrumental measurements referred to as experimental error. Second, the differential model equations are based on approximations and assumptions. Furthermore, approximations are made in the discretization process, and the set of non-linear equations are solved iteratively, which propagates uncertainty. Lastly, errors are also introduced to the solution by the limiting machine accuracy and the approximate convergence criteria (function tolerance, step tolerance, optimality tolerance etc.) set to stop the iterative process. Certainly, a difference in the experimental data and the model solution is expected.

In deriving the model equations for the modMB model, a solution-diffusion mechanism for both the selective layer and the support layer was assumed, and no further resistances in the boundary layers were included. As there is a difference in the affinity to the gas components, there will be an accumulation of the less permeable species and a depletion of the more permeable ones. This leads to a growing concentration gradient in the boundary layer, i.e. concentration polarization, which reduces the separation efficiency. Often a higher flow rate prevents this phenomena, or flushing is provided occasionally. Although, the effect of concentration polarization is not severe in gas separation, it poses a resistance in the boundary layer. Moreover a uniform membrane thickness was assumed, when in reality it can vary along the fiber length and for each fiber. Pore-blockage might also occur in the porous support layer and other transport mechanisms contributing to the membrane resistance as described in Section 2.4. The membrane performance can also be reduced with the membrane age as previously mentioned. The Joule-Thomson effect is also affecting the separation performance for separation of CO_2 at high pressures. Therefore it is suggested that the model is further extended to account for the Joule-Thomson effect, and it should be evaluated whether the resistance in the boundary layers should be included. To conclude, there are several factors and phenomena that can cause deviation in experimental measurements, of which some should be accounted for in the model.

Table 6.4: The simulation results of the modMB compared to the ChemBrane model, the mollocator and the experimental results.

Case	Source	Retentate [mol/s]	Permeate [mol/s]	CH_4 in retentate [mol-%]	H_2O in permeate [mol-%]
1	ChemBrane	$5.293 \cdot 10^3$	$8.643 \cdot 10^2$	80.85	0.209
	modMB	$5.339 \cdot 10^3$	$8.177 \cdot 10^2$	80.66	0.219
	RD [%]	0.87	-5.39	-0.24	4.78
		Retentate [mol/s]	Permeate [mol/s]	CH_4 in retentate [mol-%]	CO_2 in permeate [mol-%]
2	ChemBrane	$4.953 \cdot 10^3$	$5.630 \cdot 10^2$	86.07	65.95
	modMB	$4.972 \cdot 10^3$	$5.443 \cdot 10^2$	85.92	66.53
	RD [%]	0.38	-3.32	-0.17	0.88
3	Experimental	$3.287 \cdot 10^{-4}$	$4.313 \cdot 10^{-5}$	92.09	25.96
	Mollocator	$3.292 \cdot 10^{-4}$	$4.258 \cdot 10^{-5}$	91.95	25.04
	modMB	$2.865 \cdot 10^{-4}$	$8.530 \cdot 10^{-5}$	95.69	29.11
	RD (Mollocator,Experimental) [%]	0.15	-1.28	-0.15	-3.54
	RD (modMB,Experimental) [%]	-12.84	97.77	3.91	12.13
	RD (modMB,Mollocator) [%]	-0.13	100.33	4.07	16.25
4	Experimental	$4.219 \cdot 10^{-4}$	$4.464 \cdot 10^{-5}$	63.30	54.40
	Mollocator	$4.209 \cdot 10^{-4}$	$4.567 \cdot 10^{-5}$	63.45	55.37
	modMB	$4.202 \cdot 10^{-4}$	$4.627 \cdot 10^{-4}$	63.54	55.84
	RD (Mollocator,Experimental) [%]	-0.25	2.31	0.24	1.78
	RD (modMB,Experimental) [%]	-0.40	3.65	0.38	2.65
	RD (modMB,Mollocator) [%]	-0.17	1.31	0.14	0.85

6.2.3 Extended models

The solver is able to find a solution for the extended models that meet the stopping criteria. However, the results for the temperatures are highly unstable, suggesting that there is an issue with convergence when energy conservation is included. Several approaches were tested, such as different variants of the temperature equations as shown in Appendix A.1.2, but the results remained unstable. It should be noted that the resistance to heat transfer in the boundary layer was neglected in the extended models. For future work it should be included.

Chapter 7

Conclusion

With the increasing demand for natural gas and stricter environmental regulations, it becomes increasingly important to introduce new technologies or improve existing technology in order to comply with the emission regulations as well as the pipeline and sales specifications. Membranes for gas separation is gaining acceptance as a viable alternative for natural gas treatment with membranes showing promising separation performance for water and CO_2 removal. Membranes are small and compact, easy to scale up, require little maintenance, have low operating cost and very low or no emissions. In this work, a hybrid membrane system was simulated, optimized and modelled to investigate the technical and economical feasibility. For dehydration of natural gas, a spiral-wound MTR Pebax® co-polymer based composite membrane was simulated and for CO_2 removal a hollow fiber polyimide derived carbon membrane in series was simulated. Various design configurations were evaluated and optimized, and a membrane model was developed for prediction of membrane performance in process simulation.

The optimal design was found to be with two compressors in series between the dehydration unit and the sweetening unit, with compression of the recycled retentate from the sweetening unit, and a pressure relief valve instead of a turbine for expanding the sweetening unit permeate before entering the dehydration unit as sweep gas. This design resulted in a specific natural gas processing cost of 2.192 $\$/m^3$. However, the methane loss was at 12.7%, which is mainly lost in the dehydration unit. With the current membrane design and performance, the minimum obtainable methane loss is 11.5%. Consequently, a different design, module configuration or improved performance for the dehydration unit must be employed to meet the desired methane recovery of 2%. Additionally, it is suggested that the methane loss is included in the optimization problem for future work, either as a constraint or as an additional objective function. Moreover, the membrane cost should be adjusted to account for the module size and not just the effective membrane area.

The relative deviation between the membrane model developed in this work and one of the experiments was ≤3.65% proving that the model is valid for gas transport in hollow fiber carbon membranes. However, for the other experiment the deviations were larger proving the model invalid. In this case, the mollocator was more in agreement with the experimental data, which suggested that the pressure drop on the shell side should be included. However, other explanations for the deviation is possible such as ageing phenomena. The membrane model should therefore be compared to several experiments and at preferably higher pressures and flow rates. Comparing the model with the ChemBrane model, showed a small relative deviation of ≤ 5.4% for the spiral-wound composite membrane and ≤ 3.3% for the hollow fiber carbon membrane. The deviation is due to the lower pressure gradient, thus the model developed in this work yields more realistic gas transport than the ChemBrane model. For the purposes of integrating the membrane model in Hysys, where the efficiency is essential, the carbon membrane conditions were analysed. Two conditions proved to be efficient, but differed in accuracy. The conditions proving to be most efficient and accurate were a defined permeate pressure and a relatively small sweep gas solved with 12 collocation points. However, integrated in Hysys the model is run twice as many times, making the case of zero sweep gas a viable option despite lower accuracy. To determine the best condition for process simulation it is suggested to compare the computational effort by running an optimaztion scheme as the one presented in this work.

In reality, concentration polarization is likely to occur, and there is a possibility that other transport mechanisms take place through the porous support layer besides the solution-diffusion mechanism. Therefore, the model should be evaluated with the boundary layer resistance and other transport mechanisms in future work. It should further be investigated with respect to experimental data and computational effort. Moreover, the Joule-Thomson effect should be accounted for in the sweetening unit. Including the conservation of energy in the membrane model showed unstable results for the temperatures, but also the fiber side pressure. Therefore, it is suggested to further investigate the temperature equations and to consider other methods than the Newton-Rhapson method for solving the set of non-linear algebraic equations. It could also be useful to test other discretization methods. Similar to the mass transfer equations, the temperature equations should evaluate the thermal resistance in the boundary layer at the membrane surface.

There is some uncertainty regarding the membrane life time for both membranes. Consequently, further testing must be conducted in plant scale over time to determine the membrane life time and for further model validation. Moreover, the COM-technology and the Cape-Open unit operation in the HYSYS simulation case, should be investigated for external optimization routines. Lastly, the economic evaluation in this work was based on a preliminary cost estimation. In future work, a more detailed economic evaluation should be conducted including the sizing and cost calculation of all equipment along with a sensitivity analysis.

Bibliography

[1] He, X. Techno-economic feasibility analysis on carbon membranes for hydrogen purification. *Separation and Purification Technology*, 186:117–124, 2017.

[2] Enerdata. World Energy Consumption Statistics. https://yearbook.enerdata.net/total-energy/world-consumption-statistics.html, 2019. Retrieved July 13, 2020.

[3] BP p.l.c. Statistical Review of World Energy. https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2019-full-report.pdf, 2019. Retrieved July 13, 2020.

[4] Speight, J.G. *Water Removal*, chapter 6.1, page 137. Gulf Publishing Company, 2007.

[5] Speight, J.G. *Acid Gas Removal*, chapter 6.4, page 146. Gulf Publishing Company, 2007.

[6] Guo, B., Ghalambor, A. *Water Content of Natural Gas Streams*, chapter 8.2.1, pages 123–124. Gulf Publishing Company, 2005.

[7] Guo, B., Ghalambor, A. *Removal of Acid Gases*, chapter 8.3, pages 147–148. Gulf Publishing Company, 2005.

[8] Bahadori, A. *Natural Gas Dehydration*, chapter 9, pages 441–481. Gulf Professional Publishing, Boston, 2014.

[9] Bahadori, A. *Natural Gas Sweetening*, chapter 10, pages 483–518. Gulf Professional Publishing, Boston, 2014.

[10] Lin, H., Thompson, S.M., Serbanescu-Martin, A., Wijmans, J.G., Amo, K.D., Lokhandwala, K.A., Merkel, T.C. Dehydration of natural gas using membranes. Part I: Composite membranes. *Journal of Membrane Science*, 413-414:70–81, 2012.

[11] Lin, H. Thompson, S.M., Serbanescu-Martin, A., Wijmans, J.G. Amo, K.D., Lokhandwala, K.A., Low, B.T., Merkel, T.C. Dehydration of natural gas using membranes. Part II: Sweep/countercurrent design and field test. *Journal of Membrane Science*, 432:106–114, 2013.

[12] Zhang, C., Wenz, G.B., Williams, J., Mayne, J.M., Liu, G., Koros, W.J. Purification of aggressive supercritical natural gas using carbon molecular sieve hollow fiber membranes. *Industrial & Engineering Chemistry Research*, 56(37):10482–10490, 2017.

[13] Jakobsen, H.A. *Chemical reactor modeling: multiphase reactive flows.* Springer, 2008.

[14] Zhou, P. *Finite Difference Method*, pages 63–94. Springer Berlin Heidelberg, 1993.

[15] Iserles, A. *A first course in the numerical analysis of differential equations.* Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, 1996.

[16] Mancha, N. Zeros of Jacobi polynomials and associated inequalities. Master's thesis, University of the Witwatersrand, Johannesburg, 2015.

[17] Szegö, G. *Orthogonal polynomials.* American Mathematical Soc., 1974.

[18] Golub, G.H., Welsch, J.H. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969.

[19] Sinnott, R., Towler, G. *Chemical engineering design.*, pages 20–38. Coulson & Richardson's chemical engineering series. Elsevier, Amsterdam, 5th edition, 2009.

[20] Rothlauf, F. *Optimization Methods.*, pages 45–102. Springer Berlin Heidelberg, 2011.

[21] Larson, R., Edwards, B.H., Falvo, D.C. *Elementary linear algebra.* Houghton Mifflin Harcourt, Boston, 6th edition, 2009.

[22] Nocedal, J. *Numerical optimization.* Springer series in operation research and financial engineering. Springer, New York, 2nd edition, 2006.

[23] Singer, S., Nelder, J. Nelder-Mead algorithm. *Scholarpedia*, 4(7):2928, 2009.

[24] Çengel, Y.A., Boles, M.A. *Thermodynamics : an engineering approach.* McGraw-Hill series in mechanical engineering. Mc-Graw-Hill, Singapore ; New York, 7th edition, 2010.

[25] Helmersen, I.A. Modelling, simulation and optimization of a membrane system for natural gas dehydration and CO_2 removal. 2019.

[26] Mokhatab, S., Poe, W.A., Mak, J.Y. *Natural Gas Compression*, chapter 14, pages 433–461. Gulf Professional Publishing, 4th edition, 2019.

[27] Bahadori, A. *Gas Compressors*, chapter 5, pages 223–227. Gulf Professional Publishing, Boston, 2014.

[28] Mulder, M. *Basic principles of membrane technology.* Kluwer Academic Publishers, Dordrecht, 2nd edition, 1996.

[29] Sridhar, S., Bee, S., Bhargava, S.K. Membrane-based gas separation: Principle, applications and future potential. *Chemical Engineering Digest*, 2014.

[30] Schell, W.J., Wensley, C.G., Chen, M.S.K., Venugopal, K.G., Miller, B.D., Stuart, J.A. Recent advances in cellulosic membranes for gas separation and pervaporation. *Gas Separation & Purification*, 3(4):162 – 169, 1989.

[31] Scholes, C.A., Stevens, G.W., Kentish, S.E. Membrane gas separation applications in natural gas processing. *Fuel*, 96:15–28, 2012.

[32] Bernardo, P., Clarizia, G. 30 years of membrane technology for gas separation. volume 32, pages 1999–2004, 2013.

[33] Ismail, A.F., Khulbe, K., Matsuura, T. *Gas Separation Membranes : Polymeric and Inorganic.* Springer International Publishing : Imprint: Springer, Cham, 1st edition, 2015.

[34] Lock, S.S.M., Lau, K.K., Shariff, A.M., Yeong, Y.F. *Joule Thomson Effect in a Two-dimensional Multi-component Radial Crossflow Hollow Fiber Membrane Applied for CO_2 Capture in Natural Gas Sweetening*, chapter 14, pages 371–397. John Wiley & Sons, Ltd, 2017.

[35] Hägg, M., Deng, L. *Membranes in Gas Separation*, chapter 7, page 143–180. CRC Press, 2nd edition, 2015.

[36] Koros, W.J., Chern, R.T. *Separation of Gaseous Mixtures Using Polymer Membranes*, chapter 20, pages 862–953. John Wiley & Sons, Inc., New York, 1987.

[37] Geankoplis, C.J. *Transport processes and separation process principles : (includes unit operations).* Pearson Education, Essex, 4th edition, 2014.

[38] Qi, R. *Modeling and optimal design of spiral-wound membrane systems for gas separations.* PhD thesis, LSU Historical Dissertations and Theses, 1998.

[39] Pan, C.Y. Gas separation by permeators with high-flux asymmetric membranes. *AIChE Journal*, 29(4):545–552, 1983.

[40] Pan, C.Y. Gas separation by high-flux, asymmetric hollow-fiber membrane. *AIChE Journal*, 32(12):2020–2027, 1986.

[41] Scholes, C.A. *Cost Competitive Membrane Processes for Carbon Capture*, volume 1, chapter 8, pages 216–241. The Royal Society of Chemistry, 2018.

[42] Speight, J.G. *Properties Processing of Gas From Tight Formations*, chapter 7, pages 307–347. Gulf Professional Publishing, Boston, 2017.

[43] Stewart, M.I. *Overview of Gas-Handling, Conditioning, and Processing Facilities*, volume 2, chapter 1, pages 1–8. Gulf Professional Publishing, Boston, 3rd edition, 2014.

[44] Kidnay, A.J., Parrish, W.R., McCartney, D.G. *Fundamentals of natural gas processing*, volume 200 of *Mechanical engineering.* CRC/Taylor & Francis, Boca Raton, Fla, 2006.

[45] Poe, W.A., Mokhatab, S. *Introduction to Natural Gas Processing Plants*, chapter 1, pages 1–72. Gulf Professional Publishing, Boston, 2017.

[46] Mitariten, M., Mokhatab, S. Efficient membrane systems for natural gas dehydration. https://www.digitalrefining.com/article/1001543/efficient-membrane-systems-for-natural-gas-dehydration, 2018. Retrieved July 13, 2020.

[47] Ahmad, A.L., Adewole, J.K., Leo, C.P., Sultan, A.S., Ismail, S. Preparation and gas transport properties of dual-layer polysulfone membranes for high pressure CO_2 removal from natural gas. *Journal of Applied Polymer Science*, 131(20), 2014.

[48] He, X., Kim, T., Hägg, M. Hybrid fixed-site-carrier membranes for CO_2 removal from high pressure natural gas: Membrane optimization and process condition investigation. *Journal of Membrane Science*, 470:266–274, 2014.

[49] Xiao, Y., Low, B.T., Hosseini, S.S., Chung, T.S., Paul, D.R. The strategies of molecular architecture and modification of polyimide-based membranes for CO_2 removal from natural gas—A review. *Progress in Polymer Science*, 34(6):561 – 580, 2009.

[50] Baker, R.W. Future directions of membrane gas separation technology. *Industrial & Engineering Chemistry Research*, 41(6):1393–1411, 2002.

[51] Adams, R.T., Lee, J.S., Bae, T., Ward, J.K., Johnson, J.R., Jones, C.W., Nair, S., Koros, W.J. CO_2–CH_4 permeation in high zeolite 4A loading mixed matrix membranes. *Journal of Membrane Science*, 367(1):197–203, 2011.

[52] He, X. Lie, J.A., Sheridan, E., Hägg, M. CO_2 capture by hollow fibre carbon membranes: Experiments and process simulations. *Energy Procedia*, 1(1):261–268, 2009. Greenhouse Gas Control Technologies 9.

[53] He, X., Hägg, M. Hybrid fixed–site–carrier membranes for CO_2/CH_4 separation. *Procedia Engineering*, 44:118–119, 2012. Euromembrane Conference 2012.

[54] Turton, R., Bailie, R.C., Whiting, W.B., Shaeiwitz, J.A. *Analysis, synthesis, and design of chemical processes.* Prentice Hall international series in the physical and chemical engineering sciences. Prentice Hall PTR, Upper Saddle River, N.J, 2nd edition, 2003.

[55] Jenkins, S. Chemical engineering plant cost index: 2018 annual value - chemical engineering: Page 1, 2019.

[56] Blackman, A.G. *Aylward and Findlay's SI Chemical Data.* John Wiley & Sons, Milton, 7th edition, 2014.

[57] Chu, Y., Lindbråthen, A., Lei, L., He, X., Hillestad, M. Mathematical modeling and process parametric study of CO_2 removal from natural gas by hollow fiber membranes. *Chemical Engineering Research and Design*, 148:45 – 55, 2019.

[58] Kestin, J., Ro, S.T., Wakeham, W.A. Viscosity of carbon dioxide in the temperature range 25–700°C. *The Journal of Chemical Physics*, 56(8):4114–4118, 1972.

[59] Van Der Gulik, P.S., Mostert, R., Van Den Berg, H.R. The viscosity of methane at 25°C up to 10 kbar. *Physica A: Statistical Mechanics and its Applications*, 151(1):153 – 166, 1988.

[60] The CAPE-OPEN Laboratories Network. Specifications.
https://www.colan.org/specifications/. Retrieved July 13, 2020.

[61] AmsterCHEM. Cape-open software design and development.
https://www.amsterchem.com/capeopen.html. Retrieved July 13, 2020.

[62] The MathWorks, Inc. Optimization toolbox™ user's guide. https://se.mathworks.com/help/pdf_doc/optim/index.html?s_cid=doc_ftr, 2020. Retrieved July 13, 2020.

[63] Aspen Technologies Inc. Aspen hysys v7.3 - customization guide. https://esupport.aspentech.com/S_Article?id=000053479, 2017. Retrieved July 13, 2020.

[64] Bird, R.B., Stewart, W.E., Lightfoot, E.N. *Transport phenomena.*, pages 51,178 – 182. John Wiley & Sons, Inc., New York, 2nd edition, 2002.

[65] Ahsan, M., Hussain, A. Mathematical modeling of helium recovery from a multicomponent fuel gas with polymeric membrane. *International Journal of Chemical Engineering and Applications*, 6:173–178, 2015.

[66] Thundyil, W.J, Koros, W.J. Mathematical modeling of gas separation permeators — for radial crossflow, countercurrent, and cocurrent hollow fiber membrane modules. *Journal of Membrane Science*, 125(2):275 – 291, 1997.

[67] Kaldis, S.P., Kapantaidakis, G.C.,Papadopoulos, T.I., Sakellaropoulos, G.P. Simulation of binary gas separation in hollow fiber asymmetric membranes by orthogonal collocation. *Journal of Membrane Science*, 142(1):43 – 59, 1998.

[68] Zhao, S., Li, Z., Liu, Y., Wang, L. Simulation of binary gas separation in hollow fiber membrane-acetylene dehydration. *Desalination*, 233(1):310 – 318, 2008. The Fourth Conference of Aseanian Membrane Society: Part 1.

[69] Makaruk, A., Harasek, M. Numerical algorithm for modelling multicomponent multipermeator systems. *Journal of Membrane Science*, 344(1):258 – 265, 2009.

[70] MatWeb LLC. DuPont™ Kapton® 500VN Polyimide Film, 125 Micron Thickness. http://www.matweb.com/search/datasheet_print.aspx?matguid-=338573ad1bdf4586aa17fab95f3a57d7. Retrieved July 13, 2020.

Appendix A

Membrane model derivations

A.1 Derivation of model equations

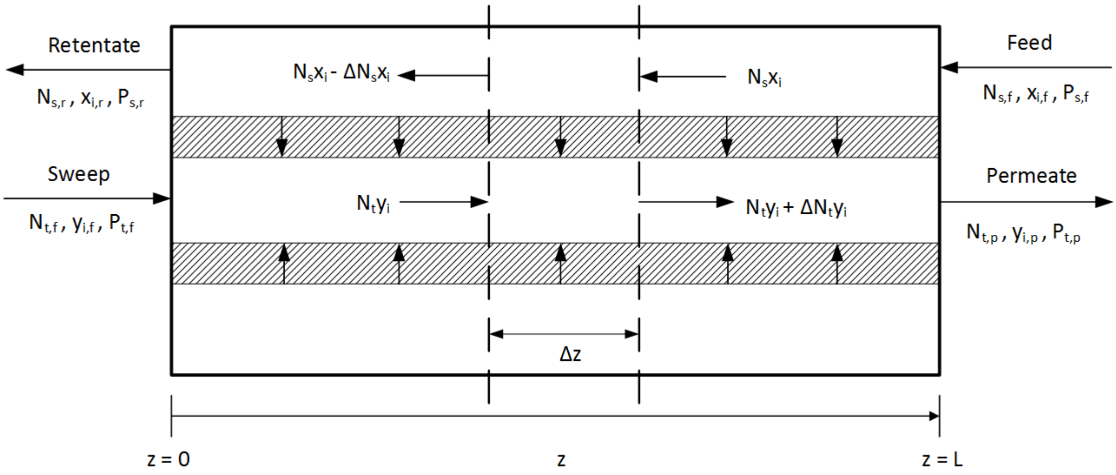

Retentate
N_{s,r} , x_{i,r} , P_{s,r}
Sweep
N_{t,f} , y_{i,f} , P_{t,f}
N_sx_i - ΔN_sx_i
N_sx_i
N_ty_i
N_ty_i + ΔN_ty_i
Δz
Feed
N_{s,f} , x_{i,f} , P_{s,f}
Permeate
N_{t,p} , y_{i,p} , P_{t,p}
z = 0
z
z = L

Figure A.1: The counter-current flow pattern for hollow fiber membrane module with sweep gas.

Boundary conditions
At feed entry:

$$z = L, \quad N_s x_i = (N_s x_i)_f, \quad P_s = P_{sf}$$

At sweep gas entry:

$$z = 0, \quad N_t y_i = (N_t y_i)_f, \quad P_t = P_{tf}$$

The model equations are derived as balance equations over a small element Δz, assuming steady state, generally expressed as

$$In - Out = 0 \tag{A.1.1}$$

A.1.1 Mass balances and pressure drop

Mass balance on shell side over a small element Δz

$$N_s x_i|_{z+\Delta z} - (N_s x_i|_z + N_T \pi D_o J_i (z - z + \Delta z)) = 0 \tag{A.1.2}$$

$$\frac{dN_s x_i}{dz} = \lim_{\Delta z \to 0} \frac{\Delta N_s x_i}{\Delta z} = N_T \pi D_o J_i \quad \left[(kmol\ h^{-1} m^{-2})(m)\right] \tag{A.1.3}$$

where J_i is the flux through the membrane for each chemical component. Assuming that the permeation abides Fick's law and assuming an ideal system which yields permeation by a solution-diffusion mechanism, the permeating mass flux can be described as follows

$$J_i = \beta_i (P_s x_i - P_t y_i)$$

$$= \beta_i \left(P_s \frac{N_s x_i}{\sum_{i=1}^{nc} N_s x_i} - P_t \frac{N_t y_i}{\sum_{i=1}^{nc} N_t y_i} \right), \quad \left[\frac{(kmol\ kPa^{-1} h^{-1} m^{-2})(kPa)(kmol\ h^{-1})}{(kmol\ h^{-1})} \right] \tag{A.1.4}$$

where β_i is the permeance of component $i = 1, 2, ..., nc$.

Mass balance on fiber side over a small element Δz

$$(N_t y_i|_z + N_T \pi D_o J_i (z - z + \Delta z)) - N_t y_i|_{z+\Delta z} = 0 \tag{A.1.5}$$

$$\frac{dN_t y_i}{dz} = \lim_{\Delta z \to 0} \frac{\Delta N_t y_i}{\Delta z} = N_T \pi D_o J_i, \quad \left[(kmol\ h^{-1} m^{-2})(m)\right] \tag{A.1.6}$$

Pressure drop in fiber

Assumption: Incompressible Newtonian fluid flowing in one-dimensional, steady state, laminar flow which is fully developed. The friction force, F_f, is proportional to some characteristic area, A, and kinetic energy per unit volume, K [64].

$$F_f = fKA \tag{A.1.7}$$

where f is a friction factor. For flow through a fiber, i.e. a circular tube, the characteristic area and the kinetic energy per unit volume are defined as

$$A = \pi D_c z \qquad K = \frac{1}{2}\rho v^2 \tag{A.1.8}$$

where D_c, z and ρ is the fiber inner diameter, fiber length and fluid density, respectively. Accounting for the number of fibers in the membrane module, N_T, the total friction force in Equation (A.1.7) becomes

$$F_f = f\frac{1}{2}\rho v^2 N_T \pi D_c z \tag{A.1.9}$$

Inserting Equation (A.1.9) into the definition for pressure gives the following expression for pressure inside a tube with crossectional area S

$$\begin{aligned} P_t &= \frac{F_f}{S} \\ &= f\frac{1}{2}\rho v^2 \frac{N_T \pi D_c z}{N_T \pi \left(\frac{D_c}{2}\right)^2} \end{aligned} \tag{A.1.10}$$

Setting the friction factor equal to the Fanning friction factor, f_F, for laminar flow

$$f = f_F = \frac{16}{N_{Re}}, \qquad N_{Re} = \frac{Dv\rho}{\mu} < 2100 \tag{A.1.11}$$

and inserting Equation (A.1.11) into Equation (A.1.10) gives

$$\begin{aligned} P_t &= \frac{16\mu}{D_c v \rho}\frac{1}{2}\rho v^2 \frac{4z}{D_c} \\ &= \frac{32\mu v z}{D_c^2} \end{aligned} \tag{A.1.12}$$

which is known as the Hagen-Poiseuille equation. Moreover, the velocity, v, can be expressed as a function of the volumetric flow per cross-sectional area with the molar volume, V_m, as follows

$$V_m = \frac{V}{n} = \frac{RT}{P} \tag{A.1.13}$$

$$
\begin{aligned}
v &= \frac{q}{S} \\
&= \frac{\sum\limits_{i=1}^{nc} N_t y_i V_m}{N_T \pi \left(\frac{D_c}{2}\right)^2} \\
&= \frac{4RT_t}{N_T \pi D_c^2 P_t} \sum_{i=1}^{nc} N_t y_i \qquad \text{(A.1.14)}
\end{aligned}
$$

Inserting Equation (A.1.14) into the pressure equation, Equation (A.1.10) gives

$$P_t = \frac{128\mu z R T_t}{N_T \pi D_c^4 P_t} \sum_{i=1}^{nc} N_t y_i \qquad \text{(A.1.15)}$$

Taking the balance over a small element Δz

$$P_t|_z - P_t|_{z+\Delta z} = \frac{128\mu(z - z + \Delta z) R T_t}{N_T \pi D_c^4 P_t} \sum_{i=1}^{nc} N_t y_i \qquad \text{(A.1.16)}$$

$$\frac{dP_t}{dz} = \lim_{\Delta z \to 0} \frac{\Delta P_t}{\Delta z} = -\frac{128\mu R T_t}{N_T \pi D_c^4 P_t} \sum_{i=1}^{nc} N_t y_i \qquad \text{(A.1.17)}$$

$$\left[\frac{(kPa\ h)(m^3 kPa\ K^{-1} kmol^{-1})(K)(kmol\ h^{-1})}{(m)^4 (kPa)}\right]$$

According to literature [65, 66, 67, 68, 69, 37], the pressure drop is expressed as the differential form of Hagen-Poseuille

$$
\begin{aligned}
\frac{dP_t^2}{dz} &= -\frac{256\mu R T_t}{N_T \pi D_c^4} \sum_{i=1}^{nc} N_t y_i \\
2P_t \frac{dP_t}{dz} &= -\frac{256\mu R T_t}{N_T \pi D_c^4} \sum_{i=1}^{nc} N_t y_i \\
\frac{dP_t}{dz} &= -\frac{128\mu R T_t}{N_T \pi D_c^4 P_t} \sum_{i=1}^{nc} N_t y_i \qquad \text{(A.1.18)}
\end{aligned}
$$

A.1.2 Energy balance

Total energy is a function of internal energy, kinetic energy, potential energy and heat and work to the system

$$E = E_U + E_K + E_P + Q - W \qquad \text{(A.1.19)}$$

It is assumed that kinetic energy and potential energy is negligible. As a first extension of the initial model described in Section 4.2.1, it is assumed no heat flow to or from the system, and the only work done on the system is the flow work

$$W = W_{flow} = pV \tag{A.1.20}$$

The energy is then expressed as

$$E = U + PV = H \tag{A.1.21}$$

The total differential of enthalpy, H, is defined as

$$dH(p,T)_n = \left(\frac{\partial H}{\partial T}\right)_P dT + \left(\frac{\partial H}{\partial P}\right)_T dP \tag{2.47 revisited}$$

where the partial derivatives have the following identities

$$\left(\frac{\partial H}{\partial T}\right)_P = c_p \tag{A.1.22}$$

$$\left(\frac{\partial H}{\partial P}\right)_T = V - T\left(\frac{\partial V}{\partial T}\right) \tag{A.1.23}$$

Inserting this into Equation (2.47) and integrating gives the enthalpy

$$H = nc_p \int_{T_{ref}}^{T} dT + \int_{P_{ref}}^{P} \left[V - T\left(\frac{\partial V}{\partial T}\right)\right] dP \tag{A.1.24}$$

Assuming ideal gas the latter expression in Equation (A.1.24) becomes 0

$$\left[V^{ig} - T\left(\frac{\partial V^{ig}}{\partial T}\right)\right] = \frac{nRT}{p} - \frac{nRT}{p} = 0 \tag{A.1.25}$$

Hence, the enthalpy is independent of pressure for ideal gases, and the expression in Equation (A.1.24) reduces to

$$H = n\hat{H} = n\hat{c}_p(T - T_{ref}) \tag{A.1.26}$$

where $\hat{H}$ is the specific enthalpy, and $\hat{c}_p$ is the specific heat capacity.

Energy balance on shell side over a small element Δz

$$\sum_{i=1}^{nc} N_s x_i \hat{H}_i|_{z+\Delta z} - \sum_{i=1}^{nc} N_s x_i \hat{H}_i|_z - N_T \pi D_o \Delta z \sum_{i=1}^{nc} J_i \hat{H}_i = 0 \tag{A.1.27}$$

Inserting Equation (A.1.26), where T_{ref} cancels out, into the balance equation and letting Δz approach 0 gives the following differential equation of the feed side temperature T_s with respect to the fiber length, z

$$\frac{dT_s}{dz} = \lim_{\Delta z \to 0} \frac{\Delta T_s}{\Delta z} = \frac{N_T \pi D_o \sum_{i=1}^{nc} J_i c_{pi}(T_s - T_t)}{\sum_{i=1}^{nc} N_s x_i c_{pi}} \tag{A.1.28}$$

Energy balance on fiber side over a small element Δz

$$\sum_{i=1}^{nc} N_t y_i \hat{H}_i|_z - \sum_{i=1}^{nc} N_t y_i \hat{H}_i|_{z+\Delta z} + N_T \pi D_o \Delta z \sum_{i=1}^{nc} J_i \hat{H}_i = 0 \tag{A.1.29}$$

$$\frac{dT_t}{dz} = \lim_{\Delta z \to 0} \frac{\Delta T_t}{\Delta z} = \frac{N_T \pi D_o \sum_{i=1}^{nc} J_i c_{pi}(T_s - T_t)}{\sum_{i=1}^{nc} N_t y_i c_{pi}} \tag{A.1.30}$$

Implementing heat of conduction

Accounting for heat of conduction changes the definition of energy for the system as presented in Equation (A.1.21) to

$$E = H + Q \tag{A.1.31}$$

where Q is the heat of conduction, Q_k, defined as

$$Q_k = A_S \kappa \frac{dT}{dr} \tag{A.1.32}$$

and A_S is the cross-sectional area, $A_S = 2\pi \Delta z r$. An approximation with respect to the radius, r, is expressed as follows

$$Q_k \int_{r_c}^{r_o} \frac{1}{r} dr = 2\pi \Delta z \kappa \int_{T_c}^{T_o} dT \tag{A.1.33}$$

$$Q_k = \frac{2\pi \Delta z \kappa (T_o - T_c)}{\ln\left(\frac{r_o}{r_c}\right)} \tag{A.1.34}$$

where the index c and o symbolises the temperature and radius at inner and outer membrane wall, respectively and κ is the thermal conductivity of the membrane material.

Energy balance on shell side

$$\sum_{i=1}^{nc} N_s x_i \hat{H}_i|_{z+\Delta z} - \sum_{i=1}^{nc} N_s x_i \hat{H}_i|_z - N_T \pi D_o \Delta z \sum_{i=1}^{nc} J_i \hat{H}_i - N_T Q_k = 0 \tag{A.1.35}$$

$$\frac{dT_s}{dz} = \lim_{\Delta z \to 0} \frac{\Delta T_s}{\Delta z} = \frac{(T_s - T_t) N_T \pi}{\sum_{i=1}^{nc} N_s x_i c_{pi}} \left(D_o \sum_{i=1}^{nc} J_i c_{pi} + \frac{2\kappa}{\ln\left(\frac{D_o}{D_c}\right)} \right) \tag{A.1.36}$$

Energy balance on fiber side

$$\sum_{i=1}^{nc} N_t y_i \hat{H}_i|_z - \sum_{i=1}^{nc} N_t y_i \hat{H}_i|_{z+\Delta z} + N_T \pi D_o \Delta z \sum_{i=1}^{nc} J_i \hat{H}_i + N_T Q_k = 0 \tag{A.1.37}$$

$$\frac{dT_t}{dz} = \lim_{\Delta z \to 0} \frac{\Delta T_t}{\Delta z} = \frac{(T_s - T_t) N_T \pi}{\sum_{i=1}^{nc} N_t y_i c_{pi}} \left(D_o \sum_{i=1}^{nc} J_i c_{pi} + \frac{2\kappa}{\ln\left(\frac{D_o}{D_c}\right)} \right) \tag{A.1.38}$$

Different versions of the temperature equations

A simulation of the model was performed with the following equations for the temperatures

$$\frac{dT_s}{dz} = \frac{\kappa A_t (T_s - T_t)}{\sum_{i=1}^{nc} N_s x_i c_{pi}} \tag{A.1.39}$$

$$\frac{dT_t}{dz} = \frac{A_t (T_s - T_t)}{\sum_{i=1}^{nc} N_t y_i c_{pi}} \left(\kappa + \sum_{i=1}^{nc} J_i c_{pi} \right) \tag{A.1.40}$$

where three scenarios was tested for which A_t was consider to be

1. $A_t = N_T \pi D_0$ for both the flux and the heat of conduction.

2. $A_t = 2N_T \pi / ln(D_o/D_c)$ for both the flux and the heat of conduction.

3. $A_t = N_T \pi D_0$ for the flux, and $A_t = 2N_T \pi / ln(D_o/D_c)$ for the heat of conduction.

Additionally, the enthalpy related to the flux giving the expression in Equations (A.1.36) and (A.1.38) is actually the enthalpy change across the membrane. A different approach was tested as well, where the enthalpy of the flux was assumed to be the average temperature across the membrane, T_m, defined as follows

$$T_m = \frac{T_s + T_t}{2} \tag{A.1.41}$$

A.2 Physical data

The viscosities, heat capacities and thermal conductivity applied in the model simulations are presented here. For the higher hydrocarbons, C_{5+}, the value for n-hexane is applied. The thermal conductivity for the carbon membrane was estimated to be that of the DuPont Kapton polyimide film at 0.12 $Wm^{-1}K^{-1}$ [70].

Table A.1: Component specific heat capacities and viscosities at standard temperature and pressure (298 K, 1 bar).

Component	Specific heat capacity[56] $[JK^{-1}mol^{-1}]$	Viscosity [μPa s]
CO_2	37	14.9 [58]
Methane	36	11.05 [59]
Ethane	53	
Propane	74	
Butane	97	
n-Hexane (C_{5+})	143	
Water	34	

Appendix B

Specialization project results

The hybrid membrane system was simulated as described in Chapter 3. The Aspen Hysys flowsheet is presented in Figure C.5.

B.1 Results

All the decision variables were optimized in Aspen Hysys by performing case studies for each of the variables subsequently [25]. The identified decision variables and their optimum is presented in Table B.1. Additionally, the optimal annual specific NG processing cost was $2.486 \cdot 10^{-3}$ $/m^3, and the corresponding compressor/turbine duties and annual capital related cost of the equipment are shown in Table B.2.

Table B.1: Initial values, lower and upper bounds for the decision variables in the optimization problem.

Decision variable	Value
2nd stage permeate pressure, $P_{\text{P-2}}$ [bar]	2.65
3rd stage feed pressure, $P_{\text{F-3}}$ [bar]	12.95
3rd stage permeate pressure, $P_{\text{P-3}}$ [bar]	2.2
3rd stage permeate recycle ratio, $\alpha_{\text{P-3}}$	0.11
1st stage area [m^2]	15735
2nd stage area [m^2]	82257
3rd stage area [m^2]	38751
Compressor discharge pressure in K-100, p_1 [bar]	4.49
Compressor discharge pressure in K-101, p_3 [bar]	7.0

Table B.2: Optimal compressor/turbine duties and costs as a result of the optimized process parameters.

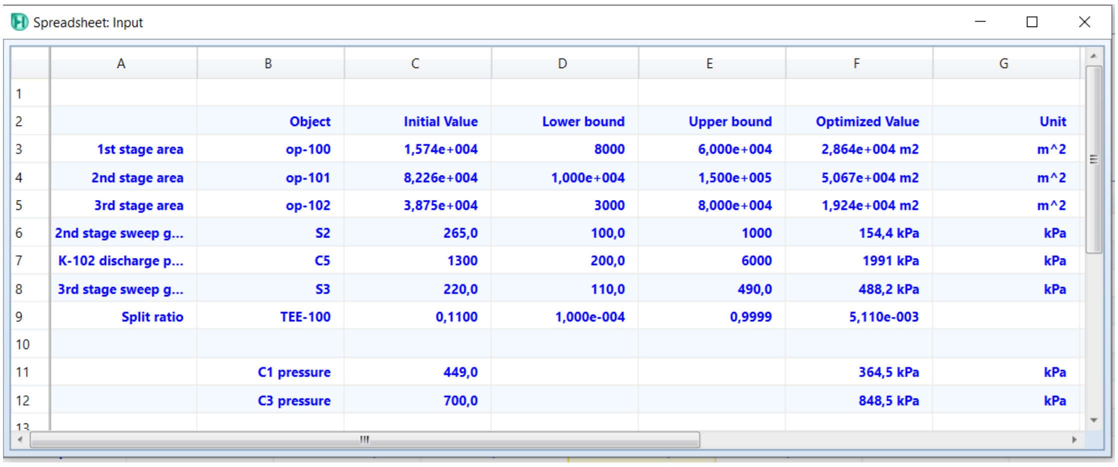
Decision variable	Value
K-100 power [kW]	1172
K-101 power [kW]	1187
K-102 power [kW]	1193
K-2 power [kW]	1888
K-3 power [kW]	979.0
K-4 power [kW]	-56.20
Sweet NG flow [m^3(STP)/h]	$4.2 \cdot 10^5$
Membrane skid cost, C_M [MM$]	6.84
Compressor cost, C_{TM} [MM$]	22.7

Appendix C

HYSYS simulation

The Hysys flowsheet and the spreadsheets used in the optimization routine are presented here.

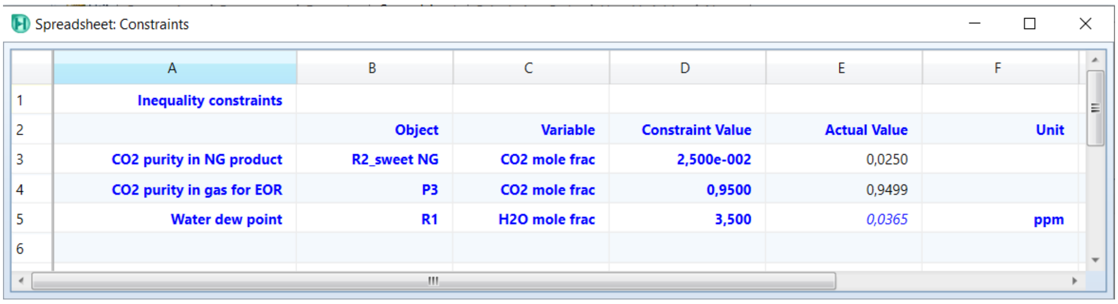
C.1 Spreadsheets



Spreadsheet: Input

	A	B	C	D	E	F	G
1							
2		Object	Initial Value	Lower bound	Upper bound	Optimized Value	Unit
3	1st stage area	op-100	1,574e+004	8000	6,000e+004	2,864e+004 m2	m^2
4	2nd stage area	op-101	8,226e+004	1,000e+004	1,500e+005	5,067e+004 m2	m^2
5	3rd stage area	op-102	3,875e+004	3000	8,000e+004	1,924e+004 m2	m^2
6	2nd stage sweep g...	S2	265,0	100,0	1000	154,4 kPa	kPa
7	K-102 discharge p...	C5	1300	200,0	6000	1991 kPa	kPa
8	3rd stage sweep g...	S3	220,0	110,0	490,0	488,2 kPa	kPa
9	Split ratio	TEE-100	0,1100	1,000e-004	0,9999	5,110e-003	
10							
11		C1 pressure	449,0			364,5 kPa	kPa
12		C3 pressure	700,0			848,5 kPa	kPa
13							

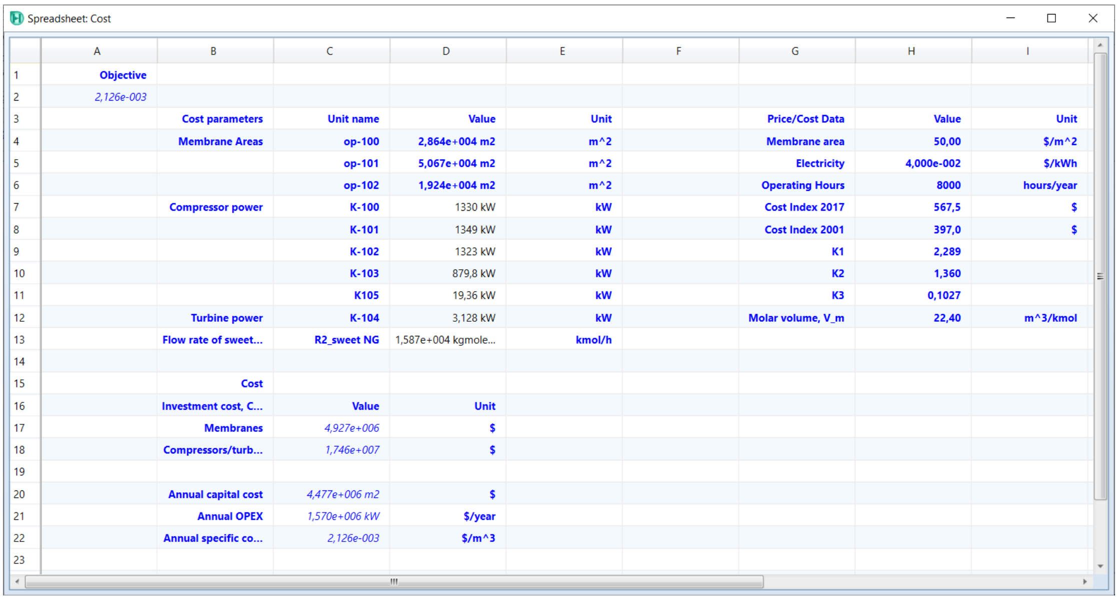
Figure C.1: The spreadsheet *Input* consists of the decision variables with the initial values used as starting point, the lower and upper bounds and the optimized values which is the output from the optimizer.



Spreadsheet: Constraints

	A	B	C	D	E	F
1	Inequality constraints					
2		Object	Variable	Constraint Value	Actual Value	Unit
3	CO2 purity in NG product	R2_sweet NG	CO2 mole frac	2,500e-002	0,0250	
4	CO2 purity in gas for EOR	P3	CO2 mole frac	0,9500	0,9499	
5	Water dew point	R1	H2O mole frac	3,500	0,0365	ppm
6						

Figure C.2: The spreadsheet *Constraints* consists of the constraints and the values of the targeted variables.



Spreadsheet: Cost

	A	B	C	D	E	F	G	H	I
1	Objective								
2	2,126e-003								
3		Cost parameters	Unit name	Value	Unit		Price/Cost Data	Value	Unit
4		Membrane Areas	op-100	2,864e+004 m2	m^2		Membrane area	50,00	$/m^2
5			op-101	5,067e+004 m2	m^2		Electricity	4,000e-002	$/kWh
6			op-102	1,924e+004 m2	m^2		Operating Hours	8000	hours/year
7		Compressor power	K-100	1330 kW	kW		Cost Index 2017	567,5	$
8			K-101	1349 kW	kW		Cost Index 2001	397,0	$
9			K-102	1323 kW	kW		K1	2,289	
10			K-103	879,8 kW	kW		K2	1,360	
11			K105	19,36 kW	kW		K3	0,1027	
12		Turbine power	K-104	3,128 kW	kW		Molar volume, V_m	22,40	m^3/kmol
13		Flow rate of sweet...	R2_sweet NG	1,587e+004 kgmole...	kmol/h				
14									
15		Cost							
16		Investment cost, C...	Value	Unit					
17		Membranes	4,927e+006	$					
18		Compressors/turb...	1,746e+007	$					
19									
20		Annual capital cost	4,477e+006 m2	$					
21		Annual OPEX	1,570e+006 kW	$/year					
22		Annual specific co...	2,126e-003	$/m^3					
23									

Figure C.3: The spreadsheet *Cost* consists of the cost functions and cost data.

C.2 Initial HYSYS process flowsheet diagram

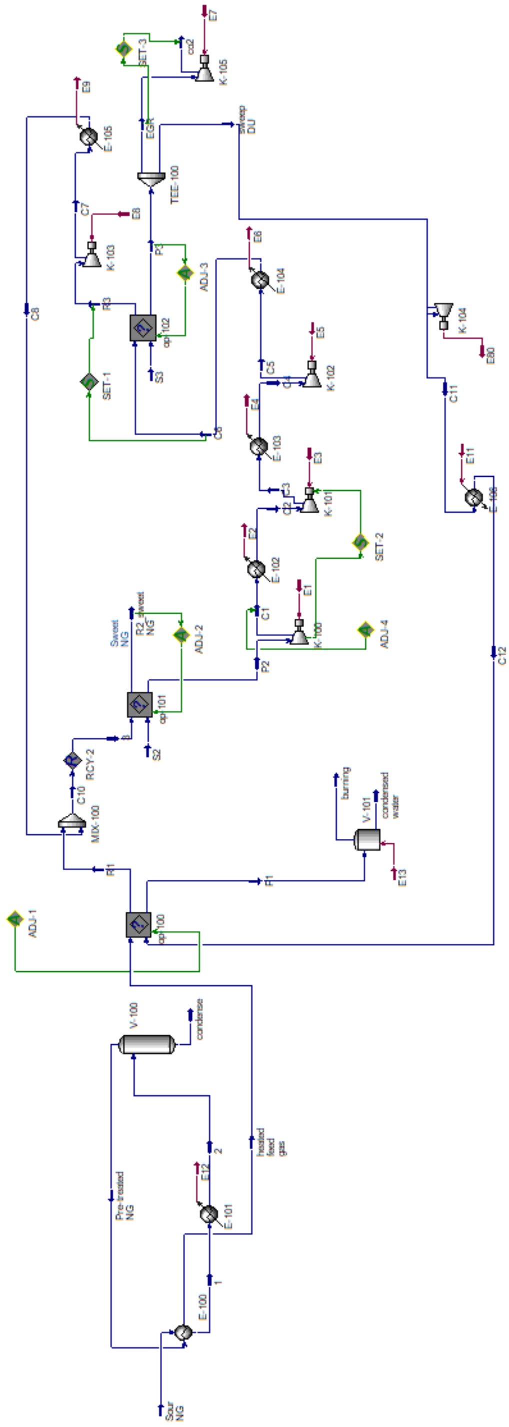


Figure C.4: The Hysys flowsheet for design configuration 1 with the ChemBrane model.

C.3 Final HYSYS process flowsheet diagram

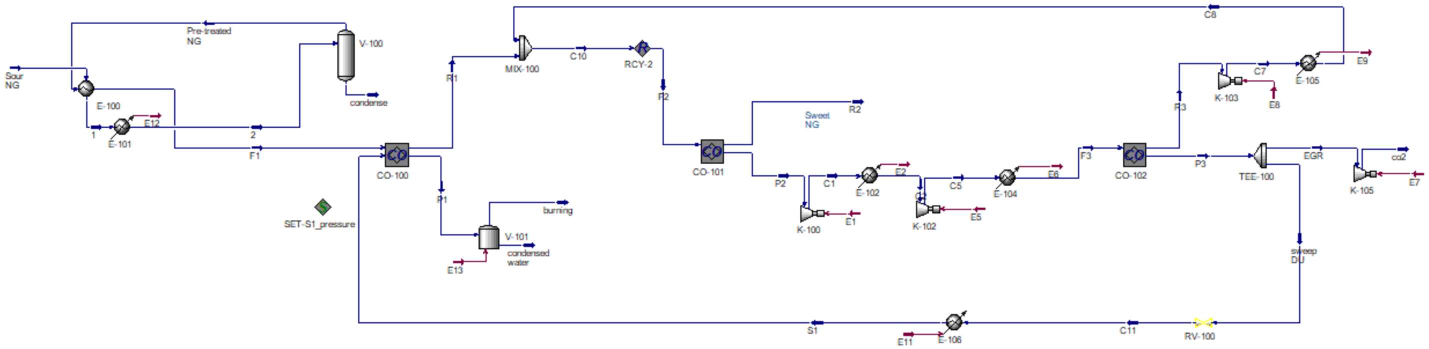


Figure C.5: The Hysys flowsheet for design configuration 6 with the modMB model integrated as Cape-Open unit operations.

C.4 Gas dew point

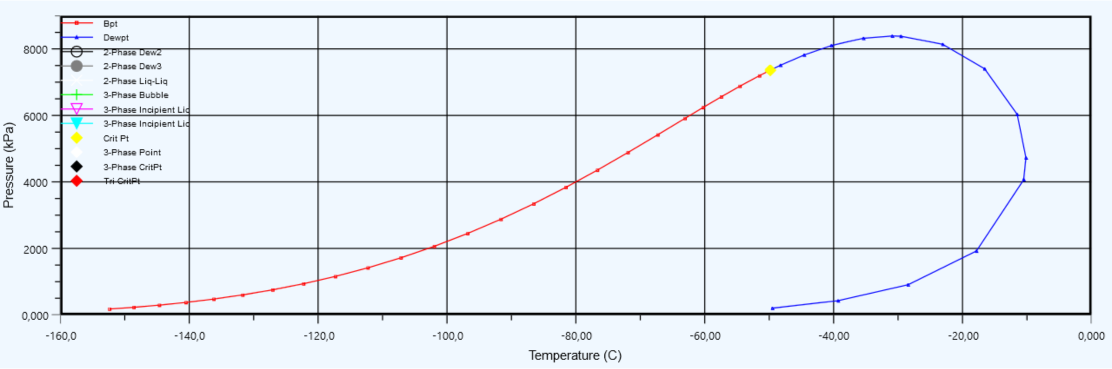


Figure C.6: The envelope of the produced sweet natural gas obtained in design configuration 5.

Appendix D

Membrane model simulation results

Graphical representation of the simulation results for case 1-4 described in Section 4.4, are presented here.

D.1 Case 1

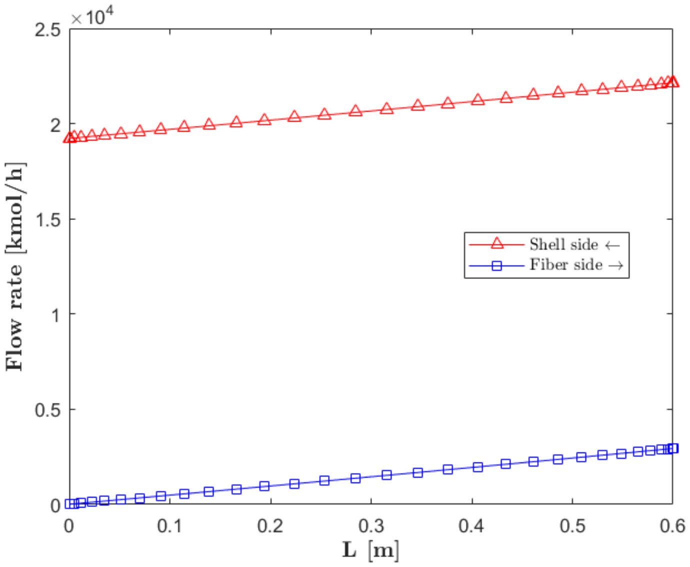


Figure D.1: Shell side and feed side total flow rates.

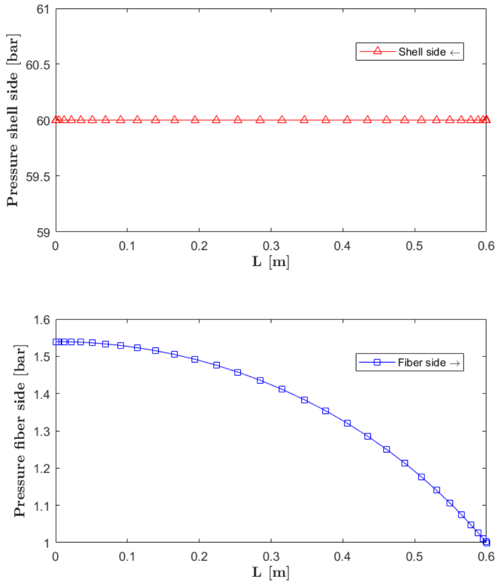


Figure D.2: Shell side and feed side pressures.

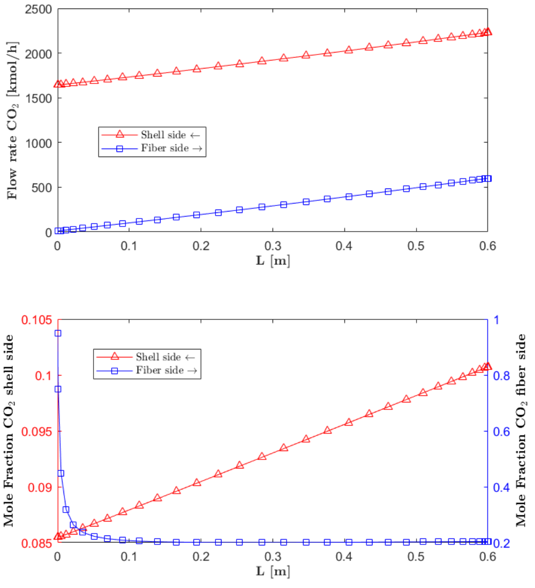


Figure D.3: Shell side and feed side CO_2 flow rates and mole fractions.

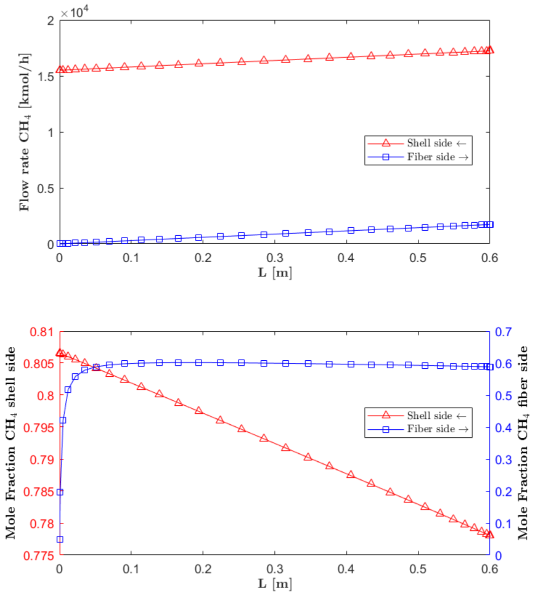


Figure D.4: Shell side and feed side methane flow rates and mole fractions.

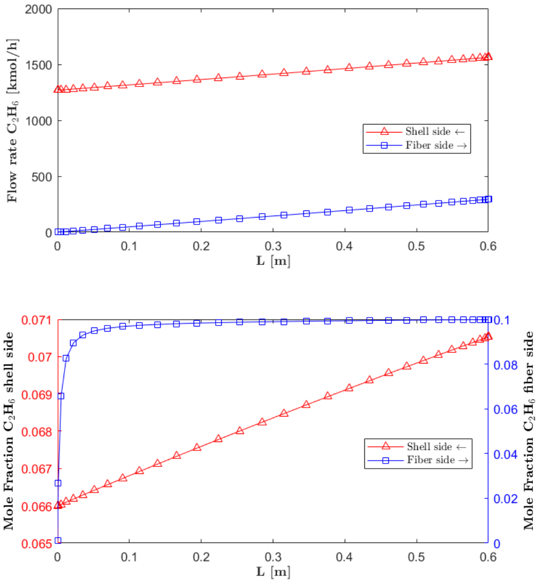


Figure D.5: Shell side and feed side ethane flow rates and mole fractions.

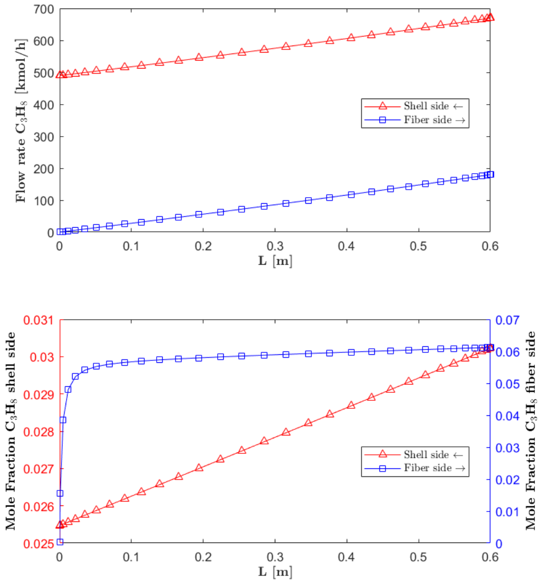


Figure D.6: Shell side and feed side propane flow rates and mole fractions.

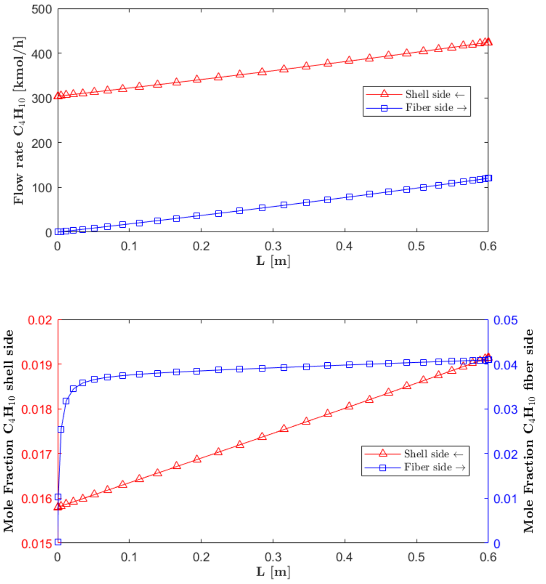


Figure D.7: Shell side and feed side butane flow rates and mole fractions.

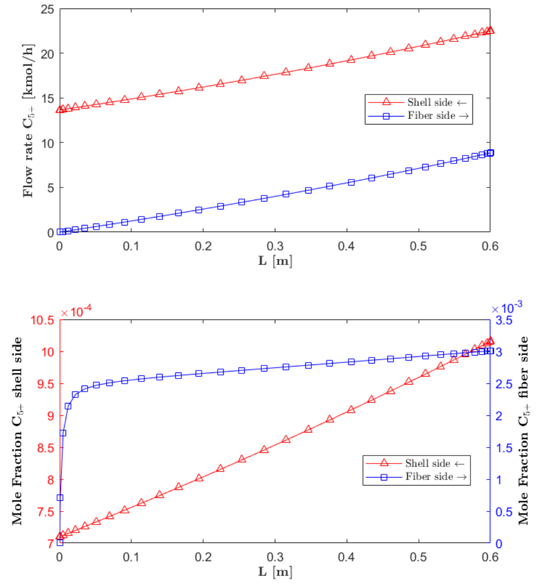


Figure D.8: Shell side and feed side C_{5+} flow rates and mole fractions.

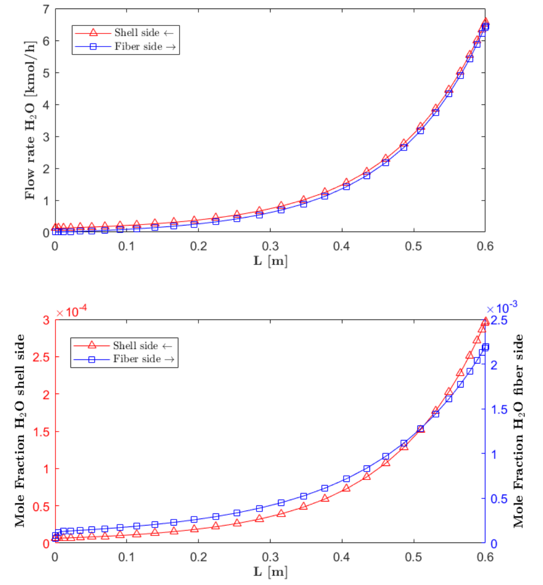


Figure D.9: Shell side and feed side water flow rates and mole fractions.

D.2 Case 2

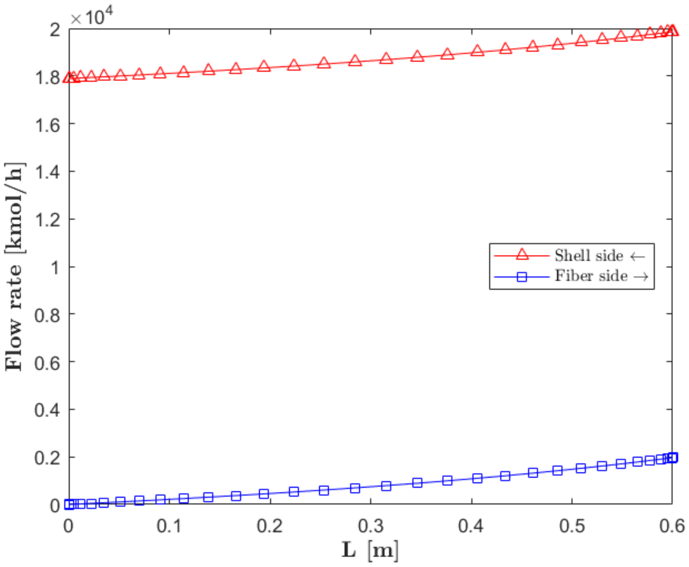


Figure D.10: Shell side and fiber side total flow rates.

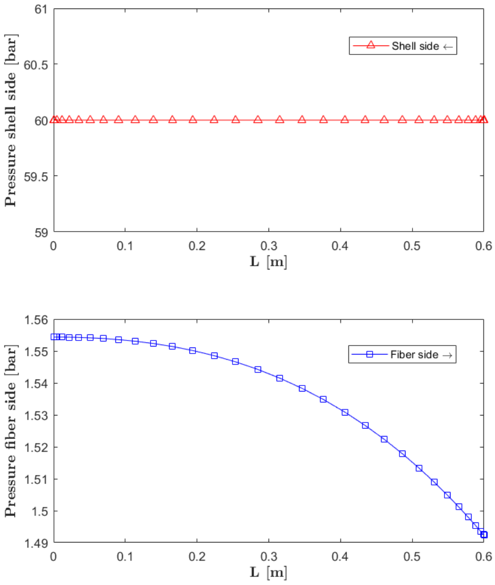


Figure D.11: Shell side and feed side pressures.

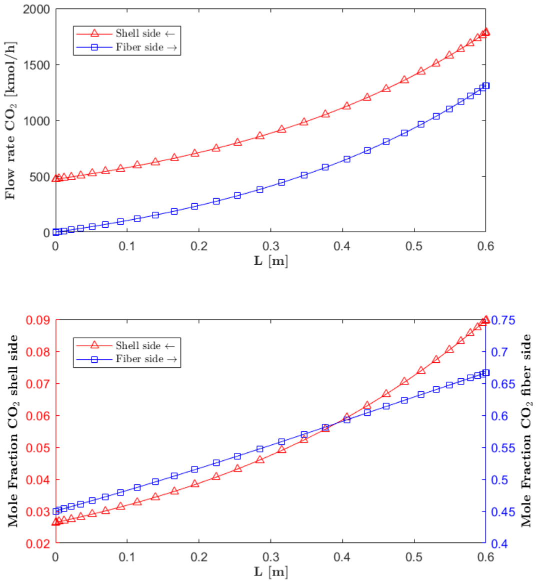
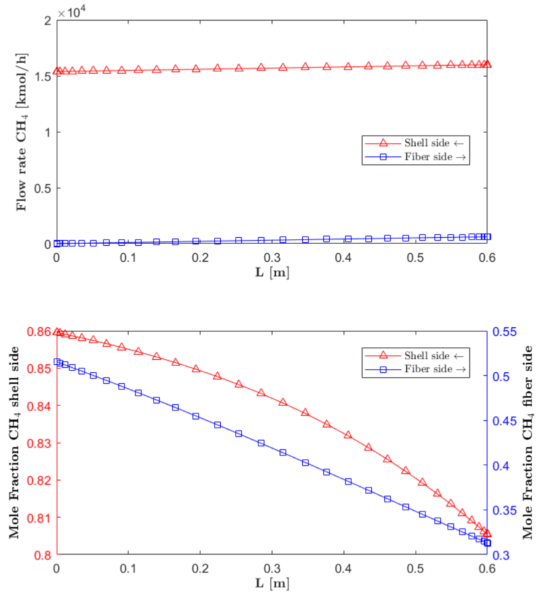


Figure D.12: Shell side and feed side CO_2 flow rates and mole fractions.



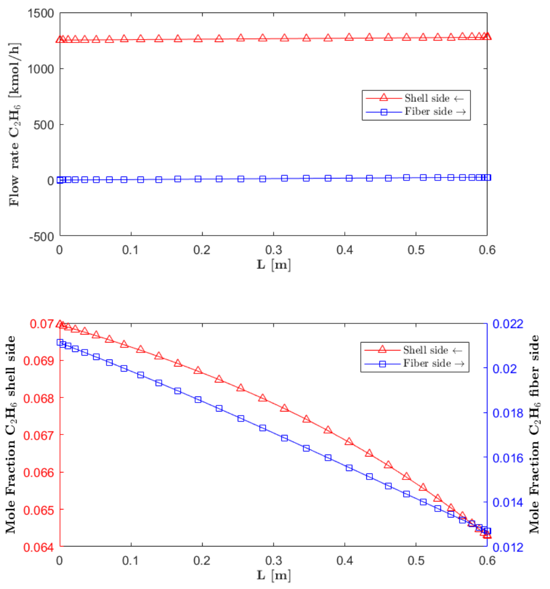
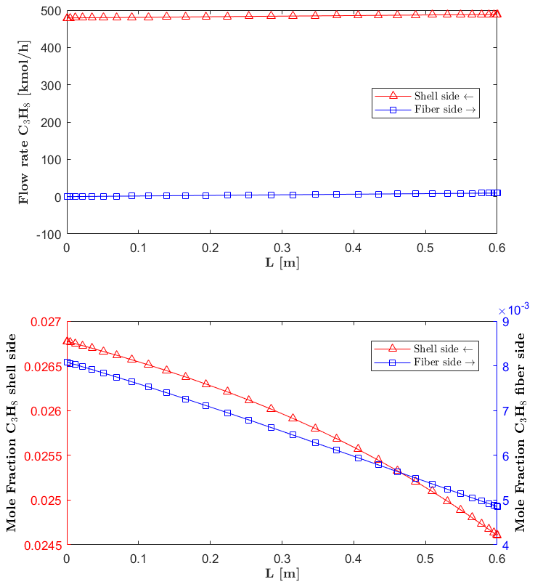


Figure D.13: Shell side and feed side methane flow rates and mole fractions.

Figure D.14: Shell side and feed side ethane flow rates and mole fractions.



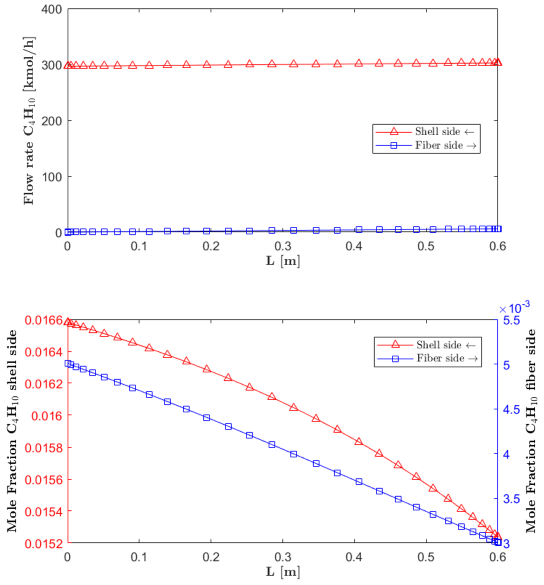
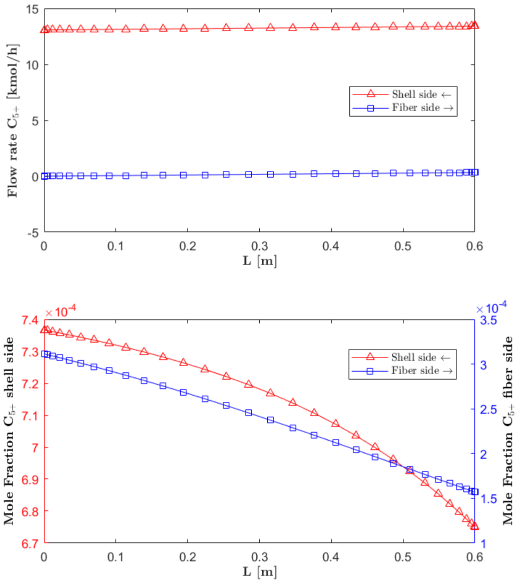


Figure D.15: Shell side and feed side propane flow rates and mole fractions.

Figure D.16: Shell side and feed side butane flow rates and mole fractions.



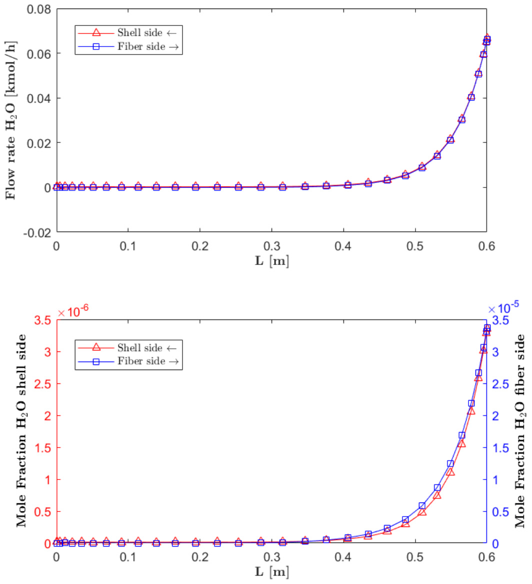


Figure D.17: Shell side and feed side C_{5+} flow rates and mole fractions.

Figure D.18: Shell side and feed side water flow rates and mole fractions.

D.3 Case 3

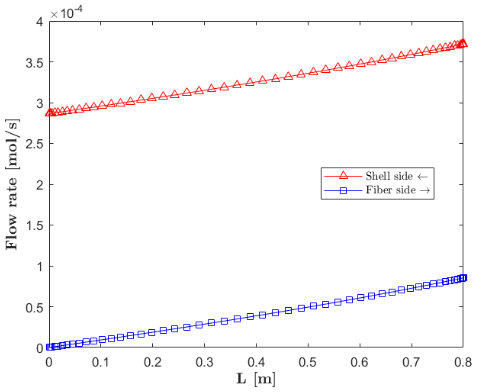


Figure D.19: Shell side and fiber side total flow rates.

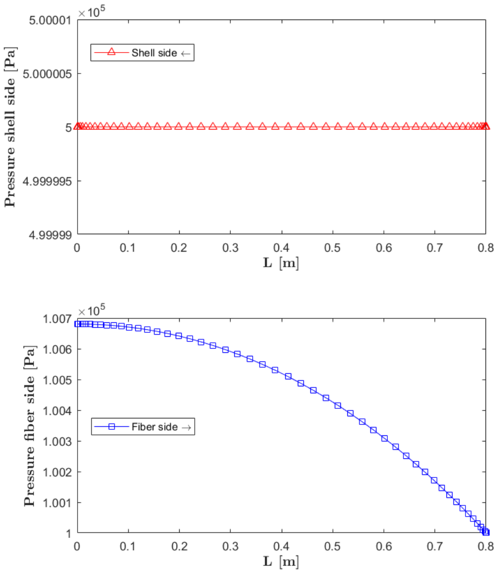
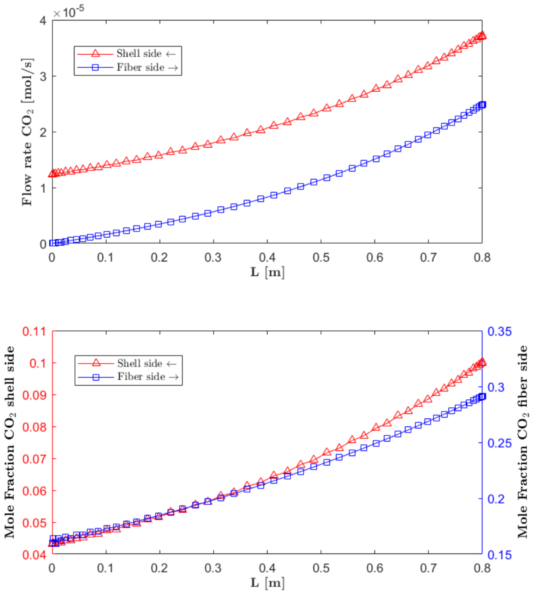


Figure D.20: Shell side and feed side pressures.



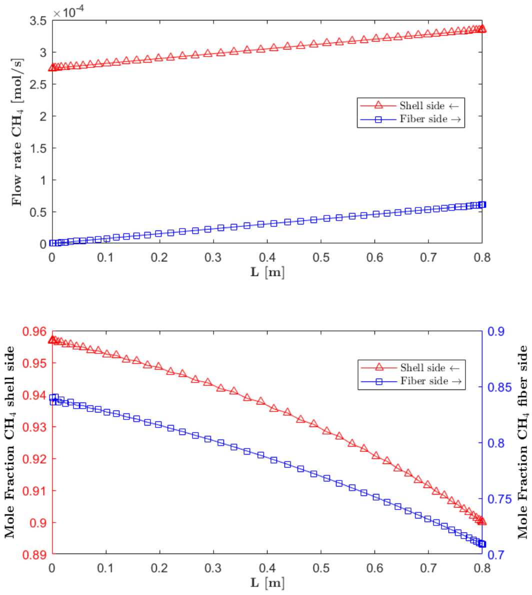


Figure D.21: Shell side and feed side CO_2 flow rates and mole fractions.

Figure D.22: Shell side and feed side methane flow rates and mole fractions.

D.4 Case 4

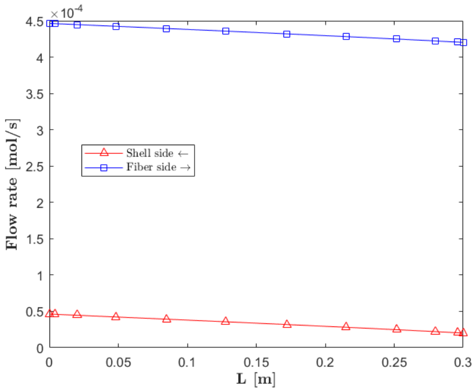


Figure D.23: Shell side and fiber side total flow rates.

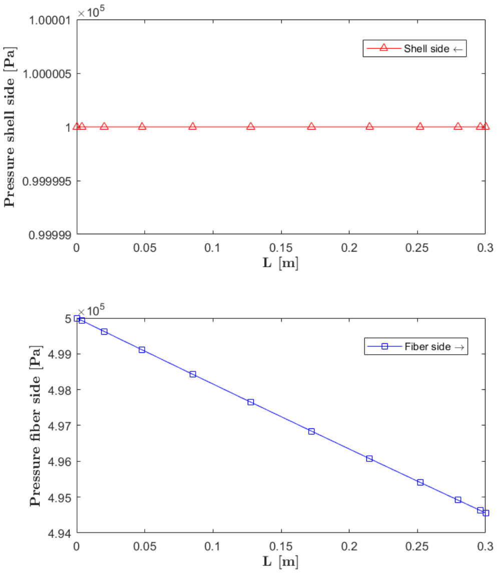


Figure D.24: Shell side and feed side pressures.

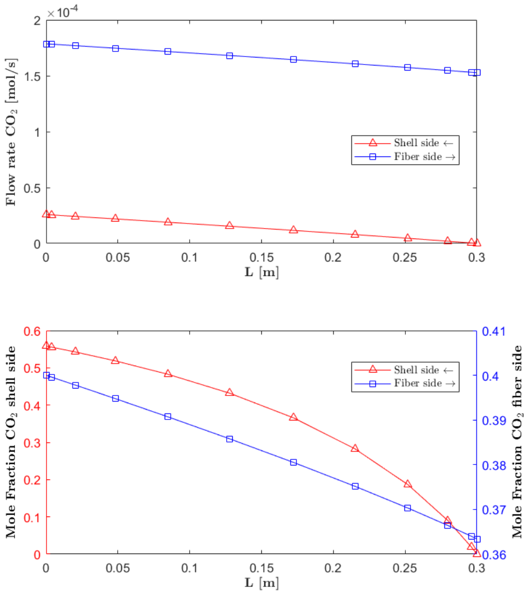


Figure D.25: Shell side and feed side CO_2 flow rates and mole fractions.

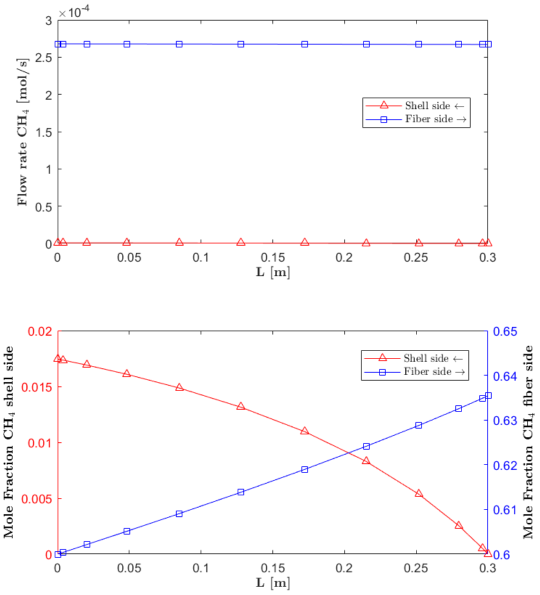


Figure D.26: Shell side and feed side methane flow rates and mole fractions.

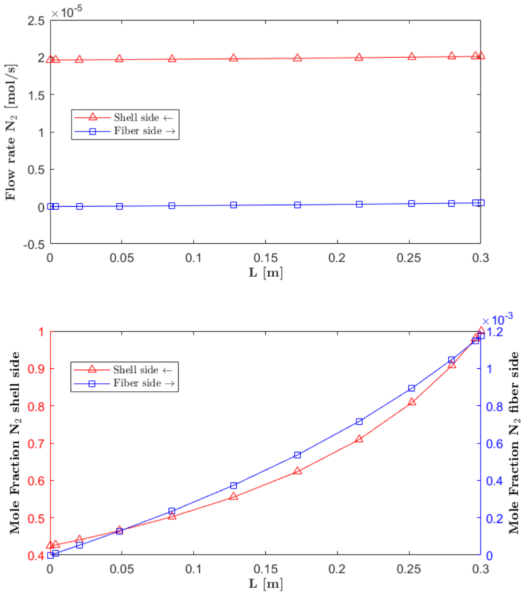


Figure D.27: Shell side and feed side nitrogen flow rates and mole fractions.

Appendix E

MATLAB code

E.1 Optimization program

The optimization program contains three Matlab functions. First, the program creates an automation server for connection with the Hysys interface. The initial decision variables and lower and upper bounds are obtained, and the solver is selected. Second, the objective function is calculated and obtained by the solver, while storing the output at each iteration with an output function.

E.1.1 The optimizer (`Optimize.m`)

This script creates a connection to Hysys, imports initial values for the decision variables and the lower and upper bounds, then scales the variables and passes them on to the selected solver.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%      OPTIMIZATION OF HYSYS SIMULATION      %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Inger Anna Helmersen
% Date (started): january 2020
% Description:  This script creates a connection to HYSYS, imports
%               initial values for the decision variables and the
%               lower and upper bounds, and scales the variables
%               then passes it on to the selected solver.
% Notation:
% hysys     :: COM automation server connected to HYSYS
% fID       :: The complete directory of the simulation case
% simcase   :: HYSYS simulation object
% fs        :: Flowsheet object
% op        :: Flowsheet operations object
% input     :: Flowsheet named 'Input' in HYSYS
% cost      :: Flowsheet named 'Cost' in HYSYS
% con       :: Flowsheet named 'Constraints' in HYSYS
% x0        :: Vector of the decision variables initial values
% lb        :: Vector of lower bounds
% ub        :: Vector of upper bounds
% x0s       :: Scaled decision variables
% lbs       :: Scaled lower bounds
% ubs       :: Scaled dupper bounds
% solver    :: The solver name
% opt       :: A struct of optimization options for the solver
%              fmincon
% history   :: A struct of optimization result from
%              runfminsearchbnd.m using the fminsearchbnd solver
% options   :: A struct of optimization options for the solver
%              patternsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all
format long

t0 = tic;
%% Create connection with HYSYS
%Create a COM automation server for connection between HYSYS and
    matlab
global simcase
hysys = actxserver('Hysys.Application');
hysys.Visible = 1;
%Open and invoke a hysys simulation case and assign object to
    variable
...names
fID='C:\Users\inger\Documents\H2019\Masterprosjekt\Hysys\config1.hsc';
simcase = hysys.SimulationCases.Open(filename);
simcase.invoke('Activate');
fs = simcase.get('flowsheet');
op = fs.get('Operations');

%Assign HYSYS spreadsheets to variables
global input cost con lb ub

input = op.Item('Input');
cost = op.Item('Cost');
con = op.Item('Constraints');

%% Decision Variables initial values
%Preallocates memory
x0 = zeros(1,7);

%Get initial values, lower and upper bounds from Input spreadsheet
for i = 1:7
```

```
    num = num2str(i+2);
    x0(i) = input.Cell(join(['C',num])).CellValue;
    lb(i) = input.Cell(join(['D',num])).CellValue;
    ub(i) = input.Cell(join(['E',num])).CellValue;
end

%Scale decision variables, and upper and lower bounds so that x0=[0,1]
x0s = xscale(lb,ub,x0);
lbs = zeros(size(lb));
ubs = ones(size(ub));

%% Solve optimization problem
solver = 'Nelder-Mead';

switch solver
    case 'Nelder-Mead'
        disp('You chose Nelder-Mead');
        history = runfminsearchbnd(x0s, lbs, ubs);

        %Plot the cost functions
        optimPlot(history);
    case 'Pattern-Search'
        disp('You chose Pattern Search');
        options = optimoptions('patternsearch','Display','iter','PlotFcn',...
            @psplotbestf);
        %The problem is well scaled so ScaleMesh is set to false.
        options.ScaleMesh = false;
        [x,fval,exitflag,output] = patternsearch(@objfunc,x0s,[],[],[],...
            [],lbs,ubs,@constraints,options);
    otherwise
        return;
end

%save optimized solution as new Hysys file:
date=datestr(now,'dd-mmm-yyyy-HH-MM-SS');
oldName = 'H2019\Masterprosjekt\Hysys\config1';
newdir = 'V2020\HYSYS\Result';
newName = sprintf('%s_%s_%s',newdir,date,solver);
newFile = strrep(simcase.FullName, oldName, newName);
simcase.SaveAs(newFile);
%Release hysys
release(hysys);
%save matlab workspace
fname = sprintf('Result_%s_%s.mat',date,solver);
save(fname)
tf = toc(t0);
```

```
disp(['Time elapsed (hh:mm:ss): ' datestr(datenum(0,0,0,0,0,tf),...
    'HH:MM:SS')]);
```

E.1.2 The objective and output function (runfminsearchbnd.m)

The following script defines the objective function and an output function for saving the output at each solver iteration.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%      SOLVE OPTIMIZATION PROBLEM      %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Inger Anna Helmersen
% Date: January 2020
% Description:  The nested objective function, objfun, calculates the
%               objective function value which is minimized by the
%               solver fminsearchbnd by changing the set of decision
%               variables. The results at each iteration is stored by
%               the nested output function, outfun.
%
% Input:
% x0s   :: The scaled decision variables
% lbs   :: The scaled lower bounds
% ubs   :: The scaled upper bounds
%
% Ouput:
% history :: A struct consisting of
%   x - decision variables
%   OPEX - Annual operating cost
%   CRC - Annual capital related cost
%   fval - Objective function value; the annual specific cost of
%          sweet natural gas processing.
%   iteration - number of iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function history = runfminsearchbnd(x0s, lbs, ubs)
% Set up shared variables with OUTFUN in a struct to store results
% from the solver at each iteration
history.x = [];
history.OPEX = [];
history.CRC = [];
history.fval = [];
history.iteration = [];

x0 = x0s;

options = optimset('fminsearch');
options.OutputFcn = @outfun;
options.Display = 'iter';
options.Diagnostics = 'on';
options.FunValCheck = 1;
options.TolFun = 1e-06;
options.MaxFunEvals = 1e4;
options.MaxIter = 4000;

[x, fval, exitflag, output] = fminsearchbnd(@objfun, x0, lbs, ubs,
   options);
history.exitflag = exitflag;
    function stop = outfun(x,optimValues,state)
        global lb ub cost
        stop = false;
        switch state
            case 'init'
                hold on
            case 'iter'
                % Concatenate current results with previous results
                % in history
                history.OPEX = [history.OPEX;
                   cost.Cell('C21').CellValue];
                history.CRC = [history.CRC;
                   cost.Cell('C20').CellValue];
                history.fval = [history.fval; optimValues.fval];
                history.iteration = [history.iteration;...
                    optimValues.iteration];
                x = xdescale(lb, ub, x);
                history.x = [history.x; x];
            case 'done'
                hold off
            otherwise
        end
    end
    function f = objfun(x)
        global simcase input cost con lb ub
        %Descale the decision values before calculations
        x = xdescale(lb, ub, x);

        %Set HYSYS solver as inactive
        simcase.Solver.Cansolve = 0;

        %Get suction and discharge pressures
        p0 = x(4);
        p5 = x(5);

        %Set interstage pressure ratio
        PR = (p5/p0)^(1/3);
        p1 = PR*p0;
        p3 = PR*(p1-5);
        p5 = PR*(p3-5);

        %update values in HYSYS
        input.Cell('F11').CellValue = p1;
        input.Cell('F12').CellValue = p3;
        x(5) = p5;
        for i=1:7
            num = num2str(i+2);
            input.Cell(join(['F',num])).CellValue = x(i);
        end

        simcase.Solver.CanSolve = 1;
        while simcase.Solver.issolving ~= 0
            %Do nothing
        end
        %Import the constraints and constrained variables (targets)
        targets = [con.Cell('E3').CellValue,
           -1*con.Cell('E4').CellValue,...
            con.Cell('E5').CellValue];
        constraints = [con.Cell('D3').CellValue,...
            -1*con.Cell('D4').CellValue, con.Cell('D5').CellValue];
        c = targets - constraints;

        %Setting the constraint on water dew point as an equality
        %constraint with some slack
        if(abs(c(3))>=0.01)
            c(3) = c(3)^2;
        end
        penalty = 0;
        for i=1:length(c)
            %All constraints larger than 0 gives penalty.
            penalty = penalty + (max(c(i),0));
        end
        penalty = penalty*1e6;

        %The annual operating cost, annual capital cost and the
        %produced sweet natural gas flow rate are imported
        %from HYSYS
        OPEX = cost.Cell('C21').CellValue;
        CRC = cost.Cell('C20').CellValue;
        sweetNG = cost.Cell('D13').CellValue;

        %Objective function
        f = (OPEX+CRC)/22.4/(sweetNG*3600)/8000 + penalty;

    end
end
```

E.1.3 The optimization solver (`fminsearchbnd.m`)

This script represent the solver algorithm which minimizes the objective function.

```
function [x,fval,exitflag,output] = fminsearchbnd(fun,x0,LB,UB,options,varargin)
% FMINSEARCHBND: FMINSEARCH, but with bound constraints by transformation
% usage: x=FMINSEARCHBND(fun,x0)
% usage: x=FMINSEARCHBND(fun,x0,LB)
% usage: x=FMINSEARCHBND(fun,x0,LB,UB)
% usage: x=FMINSEARCHBND(fun,x0,LB,UB,options)
% usage: x=FMINSEARCHBND(fun,x0,LB,UB,options,p1,p2,...)
% usage: [x,fval,exitflag,output]=FMINSEARCHBND(fun,x0,...)
%
% arguments:
%  fun, x0, options - see the help for FMINSEARCH
%
%  LB - lower bound vector or array, must be the same size as x0
%
%       If no lower bounds exist for one of the variables, then
%       supply -inf for that variable.
%
%       If no lower bounds at all, then LB may be left empty.
%
%       Variables may be fixed in value by setting the corresponding
%       lower and upper bounds to exactly the same value.
%
%  UB - upper bound vector or array, must be the same size as x0
%
%       If no upper bounds exist for one of the variables, then
%       supply +inf for that variable.
%
%       If no upper bounds at all, then UB may be left empty.
%
%       Variables may be fixed in value by setting the corresponding
%       lower and upper bounds to exactly the same value.
%
% Notes:
%
%  If options is supplied, then TolX will apply to the transformed
%  variables. All other FMINSEARCH parameters should be unaffected.
%
%  Variables which are constrained by both a lower and an upper
%  bound will use a sin transformation. Those constrained by
%  only a lower or an upper bound will use a quadratic
%  transformation, and unconstrained variables will be left alone.
%
```

```
%  Variables may be fixed by setting their respective bounds equal.
%  In this case, the problem will be reduced in size for FMINSEARCH.
%
%  The bounds are inclusive inequalities, which admit the
%  boundary values themselves, but will not permit ANY function
%  evaluations outside the bounds. These constraints are strictly
%  followed.
%
%  If your problem has an EXCLUSIVE (strict) constraint which will
%  not admit evaluation at the bound itself, then you must provide
%  a slightly offset bound. An example of this is a function which
%  contains the log of one of its parameters. If you constrain the
%  variable to have a lower bound of zero, then FMINSEARCHBND may
%  try to evaluate the function exactly at zero.
%
%
% Example usage:
% rosen = @(x)  (1-x(1)).^2 + 105*(x(2)-x(1).^2).^2;
%
% fminsearch(rosen,[3 3])     % unconstrained
% ans =
%    1.0000    1.0000
%
% fminsearchbnd(rosen,[3 3],[2 2],[])     % constrained
% ans =
%    2.0000    4.0000
%
% See test_main.m for other examples of use.
%
%
% See also: fminsearch, fminspleas
%
%
% Author: John D'Errico
% E-mail: woodchips@rochester.rr.com
% Release: 4
% Release date: 7/23/06

% size checks
xsize = size(x0);
x0 = x0(:);
n=length(x0);

if (nargin<3) || isempty(LB)
  LB = repmat(-inf,n,1);
else
  LB = LB(:);
end
if (nargin<4) || isempty(UB)
```

```
  UB = repmat(inf,n,1);
else
  UB = UB(:);
end

if (n~=length(LB)) || (n~=length(UB))
  error 'x0 is incompatible in size with either LB or UB.'
end

% set default options if necessary
if (nargin<5) || isempty(options)
  options = optimset('fminsearch');
end

% stuff into a struct to pass around
params.args = varargin;
params.LB = LB;
params.UB = UB;
params.fun = fun;
params.n = n;
% note that the number of parameters may actually vary if
% a user has chosen to fix one or more parameters
params.xsize = xsize;
params.OutputFcn = [];

% 0 --> unconstrained variable
% 1 --> lower bound only
% 2 --> upper bound only
% 3 --> dual finite bounds
% 4 --> fixed variable
params.BoundClass = zeros(n,1);
for i=1:n
  k = isfinite(LB(i)) + 2*isfinite(UB(i));
  params.BoundClass(i) = k;
  if (k==3) && (LB(i)==UB(i))
    params.BoundClass(i) = 4;
  end
end

% transform starting values into their unconstrained
% surrogates. Check for infeasible starting guesses.
x0u = x0;
k=1;
for i = 1:n
  switch params.BoundClass(i)
    case 1
      % lower bound only
      if x0(i)<=LB(i)
        % infeasible starting value. Use bound.
        x0u(k) = 0;
      else
        x0u(k) = sqrt(x0(i) - LB(i));
      end

      % increment k
      k=k+1;
    case 2
      % upper bound only
      if x0(i)>=UB(i)
        % infeasible starting value. use bound.
        x0u(k) = 0;
      else
        x0u(k) = sqrt(UB(i) - x0(i));
      end

      % increment k
      k=k+1;
    case 3
      % lower and upper bounds
      if x0(i)<=LB(i)
        % infeasible starting value
        x0u(k) = -pi/2;
      elseif x0(i)>=UB(i)
        % infeasible starting value
        x0u(k) = pi/2;
      else
        x0u(k) = 2*(x0(i) - LB(i))/(UB(i)-LB(i)) - 1;
        % shift by 2*pi to avoid problems at zero in fminsearch
        % otherwise, the initial simplex is vanishingly small
        x0u(k) = 2*pi+asin(max(-1,min(1,x0u(k))));
      end

      % increment k
      k=k+1;
    case 0
      % unconstrained variable. x0u(i) is set.
      x0u(k) = x0(i);

      % increment k
      k=k+1;
    case 4
      % fixed variable. drop it before fminsearch sees it.
      % k is not incremented for this variable.
  end

end
% if any of the unknowns were fixed, then we need to shorten
% x0u now.
```

```
if k<=n
  x0u(k:n) = [];
end

% were all the variables fixed?
if isempty(x0u)
  % All variables were fixed. quit immediately, setting the
  % appropriate parameters, then return.

  % undo the variable transformations into the original space
  x = xtransform(x0u,params);

  % final reshape
  x = reshape(x,xsize);

  % stuff fval with the final value
  fval = feval(params.fun,x,params.args{:});

  % fminsearchbnd was not called
  exitflag = 0;

  output.iterations = 0;
  output.funcCount = 1;
  output.algorithm = 'fminsearch';
  output.message = 'All variables were held fixed by the applied bounds';

  % return with no call at all to fminsearch
  return
end

% Check for an outputfcn. If there is any, then substitute my
% own wrapper function.
if ~isempty(options.OutputFcn)
  params.OutputFcn = options.OutputFcn;
  options.OutputFcn = @outfun_wrapper;
end

% now we can call fminsearch, but with our own
% intra-objective function.
[xu,fval,exitflag,output] = fminsearch(@intrafun,x0u,options,params);

% undo the variable transformations into the original space
x = xtransform(xu,params);

% final reshape to make sure the result has the proper shape
x = reshape(x,xsize);

% Use a nested function as the OutputFcn wrapper
```

```
  function stop = outfun_wrapper(x,varargin);
    % we need to transform x first
    xtrans = xtransform(x,params);

    % then call the user supplied OutputFcn
    stop = params.OutputFcn(xtrans,varargin{1:(end-1)});

  end

end % mainline end

% ======================================
% ========= begin subfunctions =========
% ======================================
function fval = intrafun(x,params)
% transform variables, then call original function

% transform
xtrans = xtransform(x,params);

% and call fun
fval = feval(params.fun,reshape(xtrans,params.xsize),params.args{:});

end % sub function intrafun end

% ======================================
function xtrans = xtransform(x,params)
% converts unconstrained variables into their original domains

xtrans = zeros(params.xsize);
% k allows some variables to be fixed, thus dropped from the
% optimization.
k=1;
for i = 1:params.n
  switch params.BoundClass(i)
    case 1
      % lower bound only
      xtrans(i) = params.LB(i) + x(k).^2;

      k=k+1;
    case 2
      % upper bound only
      xtrans(i) = params.UB(i) - x(k).^2;

      k=k+1;
    case 3
      % lower and upper bounds
      xtrans(i) = (sin(x(k))+1)/2;
      xtrans(i) = xtrans(i)*(params.UB(i) - params.LB(i)) +
          params.LB(i);
      % just in case of any floating point problems
      xtrans(i) = max(params.LB(i),min(params.UB(i),xtrans(i)));

      k=k+1;
    case 4
      % fixed variable, bounds are equal, set it at either bound
      xtrans(i) = params.LB(i);
    case 0
      % unconstrained variable.
      xtrans(i) = x(k);

      k=k+1;
  end
end

end % sub function xtransform end
```

E.2 Membrane model

The following section presents the code describing and solving the membrane model. A set of non-linear algebraic model equations are discretized by the orthogonal collocation method and solved with the Newton-Rhapson method.

E.2.1 Membrane model solver (`solveModel.m`)

This script defines the feed and membrane parameters, the number of collocation points, initial values of the dimensionless model variables, and solves the model equations for a hollow fiber membrane with counter-current flow pattern.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%       Solve membrane model      %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Inger Anna Helmersen
% Date (started): january 2020
% Description:  This script defines feed and membrane parameters,
%               the number of collocation points, initial values of
%               the dimensionless model variables, and solves the
%               model equations for a hollow fiber membrane with
%               counter-current flow pattern.
%
% Variables:
% HFperm    :: Permeances for the hollow fiber carbon membrane for
%              [CO2 CH4 C2 C3 C4 C5 H2O]
% SWperm    :: Permeances for the spiral-wound Pebax membrane for
%              [CO2 CH4 C2 C3 C4 C5 H2O]
% param     :: Struct consisting of the following membrane parameters
%   Perm - matrix of the permeances HFperm and SWperm
%          [kmol/(kPa*h*m2)]
%   Area - The effective membrane area [m2]
%   ODiam - Fiber outer diameter [m]
%   IDiam - Fiber inner diameter [m]
%   Visc - Permeate viscosity [kPa*h]
%   FiberLength - Fiber length [m]
%   Fibers - Number of fibers
%   R - Universal gas constant [m3*kPa/(K*kmol)]
%
% Feed      :: Struct consisting of the following feed parameters
%   Ns - Shell side flow rate for [CO2 CH4 C2 C3 C4 C5 H2O]
%   Nt - Fiber side flow rate for [CO2 CH4 C2 C3 C4 C5 H2O]
%   Ps - Feed pressure
%   Pt - Permeate pressure
%
% n         :: number of collocation points
% Y0         :: Initial values for the dimensionless variables
% Y          :: Solution - Approximated values for the dimensionless
%               variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Clear command window and workspace
clc
clear all
%% Creating a param struct where membrane parameters are stored.
HFperm = 1e-03*[9.5e-02 2e-03 1e-03 1e-03 1e-03 1e-03 0.95];
SWperm = 1e-03*[6.937e-02 2.31e-02 4.62e-02 7.06e-02 7.55e-02 0.1156 1.218];
param.Perm = [HFperm;SWperm];
param.Area = 61908.0195282997;
param.ODiam = 250e-06;
param.IDiam = 200e-06;
param.Visc = 14.9e-09/3600;
param.FiberLength = 0.6;
param.Fibers = round(param.Area/(pi*param.ODiam*param.FiberLength));
param.R = 8.314;
param.k = 0.12*3600;
param.Cp = 1e+03*[37 36 53 74 97 120 34];

%% Creating a feed struct where known values for the feed at shell and tube side are stored.
feed.Ns = 19858.1778560708*[8.97599390264413e-002 0.805434459540757 6.42866037442694e-002...
    2.46038563678811e-002 1.52368384720815e-002 6.74953859787872e-004...
    3.34898878201739e-006];
feed.Nt = zeros(1,7);
feed.Ps = 6000;
feed.Pt = 149.227577384370;
feed.Tt = 303.15;
feed.Ts = 273.15+59.9910189442990;

%% Orthogonal collocation method and FSOLVE
% Discretize and approximate model equation residuals

n=30;
[z, A, B, q] = colloc( n ,1 ,1);

Y0=ones(n+2,15);
%Initial values for 2nd stage membrane without sweep gas
Y0(:,1:2) = 1e-1;
Y0(:,3) = 1e-3;
Y0(:,4:5) = 1e-4;
Y0(:,6) = 1e-5;
Y0(:,7) = 1e-6;
```

```
Y0(:,8) = 1e-2;
Y0(:,9) = 0.7;
Y0(:,10:12) = 1e-4;
Y0(:,13) = 1e-4;
Y0(:,14) = 1e-6;
Y0(:,15) = 1e-2;

%Optimization settings for the solver
maxIter = 2000;
opt = optimoptions(@fsolve,'Display','iter-detailed',
    'FunValCheck',...
     'on','MaxFunctionEvaluations',10000000, 'MaxIterations',
        maxIter,...
     'StepTolerance', 1e-6, 'FunctionTolerance', 1e-6);

t0= tic;
[Y, fval, exitflag, output]=fsolve(@(Y) model_MB(Y, A, n, param,
    feed),Y0,opt);
tf = toc(t0);
disp(tf);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Plotting the results
L = param.FiberLength*z;
Res = sum(feed.Ns)*Y(:,1:14);
Res(:,15) = feed.Ps*Y(:,15);
Res(:,16) = feed.Ps;
Res(:,17:18) = zeros(n+2,2);
modelPlot(L, Res);

%% Some testing
% Total retentate and permeate flow rates, and sweep pressure
Ret = sum(transpose(sum(feed.Ns)*Y(1,8:14)));
Perm = sum(transpose(sum(feed.Ns)*Y(n+2,1:7)));
sweepPressure = feed.Ps*Y(1,15);

%Mole fractions in retentate and permeate
%If molefraction is negative, set it to zero
RetentateFrac = sum(feed.Ns)*Y(1,8:14)/Ret;
PermeateFrac = sum(feed.Ns)*Y(n+2,1:7)/Perm;
k = RetentateFrac < 0;
RetentateFrac(k) = 0;
k = PermeateFrac < 0;
PermeateFrac(k) = 0;

%Total feed and sweep flow rates
Feed = sum(transpose(Res(n+2,8:14)));
Sweep = sum(transpose(Res(1,1:7)));

```

```
%Checking the mass balance on shell/feed side and fiber/permeate side
output.dNs = Feed - Ret;
output.dNt = Perm - Sweep;
output.dJ = [Feed-Perm; Ret-Sweep; ((Feed+Sweep)-(Ret+Perm))];
```

E.2.2 Membrane model (`model_MB.m`)

This script calculates the set of dimensionless residual functions at each collocation point.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%       Membrane model      %%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Inger Anna Helmersen
% Date (started): january 2020
% Description:  This script calculates the set of dimensionless
%               residual functions at each collocation point.
%
% Input:
% Y     :: Matrix of dimensionless variables
% A     :: Matrix of first derivative weights
% n     :: Number of collocation points
% param :: Struct consisting of membrane parameters defined in
%          solveModel.m
% feed  :: Struct consisting of feed parameters defined in
%          solveModel.m
%
% Ouput:
% F     :: Matrix of residuals for each variable
%
% Notation:
% Nty   :: Fiber side flow rate for each component
% Nsx   :: Shell side flow rate of each component
% Pt    :: Fiber side pressure
% Ps    :: Shell side pressure
% Kj    :: Dimensionless constant
% Kpt   :: Dimensionless constant
% dJ    :: Permeate flux
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function F = model_MB(Y,A,n, param, feed)

%Dimensionless variables
Nty = Y(:,1:7);
Nsx = Y(:,8:14);
Pt = Y(:,15);
Ps = feed.Ps/feed.Ps;

%Dimensionless constants
Kj = param.FiberLength*pi*param.Fibers*param.ODiam*feed.Ps*...
    param.Perm(1,:)/(sum(feed.Ns));
Kpt = -128*param.Visc*param.FiberLength*param.R*sum(feed.Ns)*...
    feed.Tt/(param.Fibers*feed.Ps^2*pi*param.IDiam^4);
```

```

%Preallocates memory for the set of residuals
F = zeros(n+2,15);
%% Boundary conditions at z = 0
dJ1 = (Ps*(Nsx(1,:)/sum(Nsx(1,:)))-Pt(1)*(Nty(1,:)/sum(Nty(1,:))));

F(1,1:7) = Nty(1,:)-feed.Nt/sum(feed.Ns);
F(1,8:14) = A(1,:)*Nsx - Kj.*dJ1;
F(1,15) = A(1,:)*Pt - Kpt*sum(Nty(1,:))/Pt(1);
%% Inner collocation points:
for i=2:n+1
    dJi = (Ps*Nsx(i,:)/sum(Nsx(i,:))-Pt(i)*Nty(i,:)/sum(Nty(i,:)));

    F(i,1:7) = A(i,:)*Nty - Kj.*dJi;
    F(i,8:14) = A(i,:)*Nsx - Kj.*dJi;
    F(i,15) = A(i,:)*Pt - Kpt*sum(Nty(i,:))/Pt(i);
end
%% Boundary conditions at last collocation point, z = n+2;
dJn =
    (Ps*Nsx(n+2,:)/sum(Nsx(n+2,:))-Pt(n+2)*Nty(n+2,:)/sum(Nty(n+2,:)));

F(n+2,1:7) = A(n+2,:)*Nty - Kj.*dJn;
F(n+2,8:14) = Nsx(n+2,:) - feed.Ns/sum(feed.Ns);
F(n+2,15) = Pt(n+2) - feed.Pt/feed.Ps;
end
```

E.2.3 Orthogonal collocation (`colloc.m`)

The following script calculates the collocation weights for discretization of the model equations.

```
function [r, A, B, q]=colloc(n,left,right)
% colloc: Calculate collocation weights
%         [r, A, B, q] = colloc( n [,'left'] [,'right'])
% inputs:
%               n - number of interior node points
%          'left' - include left boundary
%         'right' - include right bounary also
% outputs:
%          r - vector of roots
%          A - Matrix of first derivative weights
%          B - Matrix of second derivative weights
%          q - Quadrature weights.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Copyright (C) 1996, 1997 John W. Eaton
%
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 2, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
% General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with Octave; see the file COPYING.  If not, write to the Free
% Software Foundation, 59 Temple Place - Suite 330, Boston, MA
% 02111-1307, USA.
%
% Adapted from Octave's colloc.cc by Steve Swinnea.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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

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

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

```

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

```
% -- ADD INTERPOLATION POINTS AT X = 0 AND/OR X = 1
%
if (n0 ~= 0)
    root = [ 0 ; root(1:nt-1) ];
end
if (n1 == 1)
    root(nt) = 1;
end
[dif1 dif2 dif3] = dif( root );

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

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

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